UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

ABDUCCIÓN EN LÓGICA DE PRIMER ORDEN: RAZONAMIENTO AUTOMÁTICO A TRAVÉS DE TABLAS SEMÁNTICAS

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRA EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A :

ARACELI LILIANA REYES CABELLO

DIRECTORA DE TESIS:
DRA. ATOCHA ALISEDA LLERA

México, D.F. 2005

Un desengaño a tiempo es un don inapreciable del cielo.
Antonio San Martín

Für alle Nächte, in denen ich in deinen Armen geschlafen habe.

Abducción en Lógica de Primer Orden: Razonamiento Automático a través de Tablas Semánticas

Araceli Liliana Reyes Cabello.

A Elo.

Por enseñarme a luchar por mis sueños y por tu inagotable amor. Te amo.

A Ponchis.

Por tu amor y alegría. Por enseñarme a vivir la vida conforme a lo que creo, pienso y siento. Te amo.

A Monty.

Por tu peculiar manera de convertir cada momento "catastrófico" en un chiste.

A Favio.

Por creer en mi.

A Canek.

Por el tiempo en el que fuiste oasis y espejismo.

Agradecimientos

Lo mejor de terminar este trabajo es poder agradecerles a aquellas personas que hicieron posible que pasara. Gracias a todos aquellos seres vivos[1] con los cuales compartí un instante de esta etapa, momentos de alegría y de tristeza, de paz y de tensión, de trabajo y de descanso, de enseñar y de aprender, de ilusión y desilusión... Especialmente a cada una de las siguientes personas:

A Dios, por este hermoso regalo llamado vida. A mis padres; por dejarme decidir libremente como vivirla, pero siempre permaneciéndo a mi lado apoyándome y amándome.

Al Dr. David Rosenblueth y al Dr.Francisco Hernández por el tiempo que dedicaron a revisar este trabajo. Muy especialmente a los Drs. José Alfredo Amor y Favio Miranda, por todo el trabajo y entusiasmo que invirtieron en la realización de este proyecto. Por el espacio que cada uno por su parte[2] me brindó para poder realizarlo. Al Dr.Ángel Nepomuceno por el tiempo que pude trabajar contigo y aprender de ti. A la persona que me enseño que la abducción no sólo tiene que ver con marcianos... ;). Mil gracias Atocha por confiar en mi.

A María Eugenia Valdés; por todo tu cariño y apoyo, por tus consejos, por tu tiempo, por tu casa y por todo lo demás. Gracias por esa alegria tan contagiosa con que encaras cada día.

A Lula, Juanjo[3], Guso, Karola, Yaz y Erick por su amistad y cariño. A los inges, Olivia y Roberto; por todas las cosas que pasamos juntos, sin ustedes no hubiera sido lo mismo.[4] A Isma, por todas tus muestras de cariño.

A todos mis maestros pero en especial a Elisa Viso por todos tus consejos, por tu cariño y sobre todo, por ceer en mi. A Jorge Urrutía por ser siempre un libro abierto, siempre dispuesto a compartir tus conocimientos. Gracias por el tiempo que me dedicaste y por ayudarme a tomar las decisiones correctas.

A la que fue mi familia en este tiempo, a mi má Lulú; por las charlas por la tarde, por tus consejos, por los clips, por las hojas, por los cafés, por las galletas... en fin, por tantas cosas. A Ceci; por preocuparte por mi. A Juanita, por tus regaños. A las tres, mil gracias por todo su cariño. Las quiero mucho.

A mis alumnos de la Facultad de Ciencias por permitirme vivir la maravillosa experiencia de enseñar.

A Valeria, por ayudarme a atravesar el espejo mágico.

[1] Escribo seres vivos porque está incluido el pasto del IIMAS... :)

[2] Lease como cubículo y casa, respectivamente.

[3] Gracias por los meses que compartimos casa. Eres la nobleza con pies...te quiero mucho.

[4] Ahh!! se me olvida, gracias por ayudarme a pasar Arquitectura.

A la Universidad Nacional Autónoma de México (UNAM) por ser mi otro hogar. A las instalaciones del Instituto de Investigaciones de Matemáticas Aplicadas y Sistemas (IIMAS) por todos sus servicios, en especial el de la biblioteca (gracias Juanita). Al Consejo Nacional de Ciencias y Tecnología (CONACYT) por su apoyo.

Creo que Dios no sólo nos permite coincidir con personas especiales con las cuales compartimos momentos inolvidables. Creo que también hay momentos en los que Dios, además, nos da la oportunidad de conocer a nuestros ángeles; los cuales, nos conocen a fondo y nos aman, nos sostienen cuando estamos cansados, nos escuchan y aconsejan. Pero sobre todo, a su lado siempre podemos ser plenamente nosotros mismos y encontrar un lugar donde no importa lo que esté pasando nos sentimos felices y en paz. En esta etapa fui muy afortunada porque me encontré con seis de ellos.

Este último párrafo lo reserve especialemente para mis ángeles para agradecerles con todo mi corazón todo su cariño, cuidado y todo su apoyo. A Ci; por esa manera que tienes de darlo todo y por estar tan cerca en momentos tan difíciles, por comprenderme. A Kiki; por tu ejemplo de fortaleza, por el tiempo que has invertido en conocerme, por escucharme, pero sobre todo, por dejarme conocerte. A Pily; por ser como el chapulín colorado, siempre te apareces en los momentos más difícilcs. A Clemente; por tu permanente estar aunque estés ausente. Gracias por todo tu amor. Te ... A Favio; porque aunque nos separaba (la mayor parte del tiempo) un gran charco de agua siempre estuviste cerca, regañándome, aconsejándome, impulsándome, pero sobre todo, sin importar de color estuviera pintado el día siempre fuiste un motivo para sonreir.[5] Por último, a quién decir que estuvo un instante junto a mi sería una "gran" mentira, porque estuvo muchas muchas horas de casi todo los días de los últimos tres años de mi vida, compartiendo momentos buenos y malos. A Monty. Definitivamente, si tuvieramos que ponerles nombre a las etapas de nuestra vida ésta se llamaría como tú, gracias chiquis; por cada uno de los momentos que compartimos juntos, gracias por todo tu cariño y confianza, por hacerme reír tanto con tus locas ocurrencias, por las interesantes pláticas en el puente y en el pasto. Por París.

[5]Se me olvidaba, por los partidos de scrabble y por solaparme.

Índice general

Y que importa que el mundo diga, con tal que me deje que haga.
Y que al mundo yo satisfaga, no es una ley que me obliga.

1
Introducción

La "abducción" es una forma de razonamiento cuya característica principal es la inferencia de premisas a partir de una conclusión dada. Tales razonamientos pueden ser encontrados en diversas disciplinas de estudio. La filosofía, lógica, inteligencia artificial y programación lógica, son sólo algunas de ellas.

Hablando en el ámbito lógico, un razonamiento abductivo consiste en inferir una explicación α a partir de las premisas $\alpha \rightarrow \varphi$ y φ. Tal procedimiento es claramente una regla de inferencia lógica incorrecta que, sin embargo, refleja algunas formas de razonamiento como el diagnóstico y el sentido común, en los que se intenta suponer cuales son las causas que originan un evento dado. Más formalmente, el razonamiento abductivo es un procedimiento lógico que consiste en generar una hipótesis o explicación α que permita explicar un hecho φ, el cual no es consecuencia lógica de una teoría adoptada Θ (véase [MP93]).

En lo que se refiere al estudio de la abducción proposicional, se han realizado diversos trabajos, tanto en la teoría como en la implementación de procedimientos que encuentren soluciones a un problema abductivo [Al97], [St91], [MP93]. Con respecto a la abducción de primer orden, el campo de investigación ha sido explorado muy escuetamente. La razón principal de ello es la indecidibilidad de la lógica de primer orden, ya que la definición aceptada de problema abductivo requiere como prerrequisito que $\Theta \not\models \varphi$ y $\Theta \not\models \neg\varphi$. De manera que, dado que la consecuencia lógica es indecidible, ni siquiera seríamos capaces de verificar si se cumplen los prerrequisitos recien mencionados, por lo que el problema resultaría ser intratable.

Por otro lado, el problema de la indecidibilidad no ha sido un obstáculo para atacar el problema de consecuencia lógica desde el ámbito computacional en fragmentos decidibles de la lógica de primer orden (de aquí en adelante $\mathcal{LPO}$). Existen diversos programas de razonamiento automático, no triviales tanto en su implementación como en sus aplicaciones, los cuales deciden el problema de consecuencia lógica en diversos fragmentos de

$\mathcal{LPO}$. Un ejemplo de ello es OTTER.[1]

De lo anterior surge la siguiente pregunta:

¿Existirán conjuntos de fórmulas (teorías y hechos) de $\mathcal{LPO}$ donde sea posible definir y resolver el problema de la abducción?

Adelantando que la respuesta a esta pregunta debería ser afirmativa, en este trabajo nos interesa plantear el problema abductivo pensando en conjuntos de fórmulas con modelos finitos, de la manera más general posible, aunque pensando en fórmulas pertenecientes a fragmentos decidibles de $\mathcal{LPO}$.

Un primer estudio sobre la abducción de primer orden aparece en [MP93]; en este artículo se analizan por vez primera los conceptos involucrados en la abducción de primer orden. Además se plantea un método teórico para obtener soluciones, utilizando para ello las tablas semánticas de Beth y el cálculo de secuentes.

En [Ne02] y [AN03] se plantea el problema de la abducción de primer orden por medio de una variación a las tablas semánticas de Beth, denominada DB-tableaux ([Bo84], [Di93]). Se utiliza una noción restringida de satisfacción referente a dominios finitos introducida en [Ne99], para introducir el concepto de *problema n-abductivo* a través de los DB-tableaux. La modificación hecha a los tableaux de Beth permite transformar un tableaux con ramas infinitas a uno con ramas acabadas, las cuales proporcionan información para construir explicaciones literales a problemas abductivos. El estudio [AN03], elaborado por Aliseda y Nepomuceno, sobre problemas abductivos de primer orden se restringe a teorías satisfactibles en modelos finitos y cuyos enunciados son de la forma $\forall x \exists y \psi$, con ψ un fórmula libre de cuantificadores. Además, la conclusión debe ser un enunciado sin cuantificadores y las explicaciones que se obtienen mediante los DB-tableaux son sólo literales.

Inspirados en los articulos anteriores, en este trabajo nos ocuparemos de analizar el problema de la abducción en la lógica de primer orden en conjuntos sencillos de fórmulas que tengan la propiedad del modelo finito.[2]

Para finalizar esta introducción describimos brevemente el contenido de nuestro trabajo. El capítulo dos se ocupa de presentar los conceptos básicos de la sintaxis y semántica de la lógica de primer orden que utilizaremos a lo largo de este trabajo. Además, se realizará un refinamiento al concepto de n-satisfacción y n-consecuencia, con el fin de caracterizar problemas n-abductivos por medio de estos conceptos.

El capítulo tres expone una breve introducción a las tablas semánticas de Beth además de presentar un estudio detallado de la correctud de los DB-tableaux, enfantizando la relación que existe entre ellos y la noción de n-consecuencia lógica.

El tema de la abducción de primer orden en conjuntos de fórmulas decidibles y satisfactibles en dominios finitos se aborda en el capítulo cuatro, donde se definen los problemas n-abductivos en términos de la noción de n-satisfacción. Además se caracterizan las

[1] Ver `http://www-unix.mcs.anl.gov/AR/otter`

[2] Una clase $X \subseteq \mathcal{LPO}$ tiene la propiedad del modelo finito, si para cualquier fórmula φ en X que tiene un modelo tiene un modelo finito [BGG01] (p.240)

soluciones de éstos por medio del concepto de la n-consecuencia lógica y se propone una nueva modificación al método de tablas semánticas, los N-tableaux, con miras a obtener un procedimiento efectivo para la búsqueda de soluciones a problemas n-abductivos. Finalmente se demuestra la correctud de los N-tableaux con base en la noción de n-satisfacción.

En el último capítulo presentamos algunas conclusiones, así como líneas abiertas de investigación a desarrollar en un futuro.

Aprende del ayer, vive para hoy, espera el mañana. Lo más importante es nunca dejar de preguntar.

Albert Einstein.

2

Preliminares: Lógica de primer orden

En este capítulo describiremos la sintaxis y semántica del lenguaje de primer orden que utilizaremos a lo largo de este trabajo revisaremos algunas de las definiciones de la lógica de primer orden así como los teoremas que necesitaremos. Además presentaremos un nuevo concepto de satisfacción restringida, el cual depende de la cardinalidad del dominio de discurso. Esta noción fue propuesta por Ángel Nepomuceno en [Ne99] y es por medio de ella que podremos abordar el problema de la abducción en la lógica de primer orden.

2.1. Sintaxis

Sea $\mathscr{L}$ un lenguaje formal de primer orden, el cual está definido sobre el alfabeto formado por:

- Un conjunto infinito numerable de variables $Var = \{x_1, x_2, \ldots\}$.
- Las constantes lógicas: $\bot, \top$.
- Los operadores o conectivos lógicos: $\neg, \wedge, \vee$. [1]
- Los cuantificadores: $\exists, \forall$.
- Símbolos auxiliares: (,), ","

La signatura del lenguaje está dada por:

[1]Tomaremos como hecho que cualquier fórmula puede ser representada por una fórmula equivalente que sólo contiene estos tres conectivos lógicos, apelando a que forman un conjunto de conectivos completo.

- Un conjunto no vacío de símbolos o letras de predicado subconjunto de $Pred = \{P_m^n \mid n, m \in \mathbb{N}, n \geq 1\}$, donde cada $n \geq 1$ denota el número de argumentos que recibe el predicado P_m^n. El número n se conoce como el *índice* o *aridad* del símbolo P_m.

- Un conjunto de constantes subconjunto de $Cons = \{c_1, c_2, \ldots\}$

Observemos que el lenguaje no contiene igualdad ni símbolos de función. Además, en la práctica usaremos metavariables $P, Q, R, \ldots$ para denotar símbolos de predicados.

Definición 2.1 (término) *Un término del lenguaje $\mathscr{L}$ es una variable o una constante. El conjunto de términos de $\mathscr{L}$ se denota por $TERM$ ($TERM = Cons \cup Var$).*

Definición 2.2 (fórmula atómica) *Sean P_m^n un símbolo de predicado con índice $n \geq 1$ y $t_1, t_2, \ldots, t_n$ términos en $TERM$ (no necesariamente distintos), entonces $P_m^n(t_1, t_2, \ldots, t_n)$ es una fórmula atómica de $\mathscr{L}$. El conjunto de fórmulas atómicas se denota por $ATOM$.*

Observación 2.1 *En la práctica omitiremos paréntesis y comas en fórmulas atómicas si tal omisión no genera ambigüedades. Por ejemplo, la fórmula atómica $P(x, y)$ se puede escribir como Pxy.*

Definición 2.3 (literal) *Una literal es una fórmula atómica o la negación de ésta. Decimos que dos literales l y l^* son complementarias, en los siguientes casos:*

- $l = P_m^n(t_1, \ldots, t_n)$ *y* $l^* = \neg P_m^n(t_1, \ldots, t_n)$, *o bien,*
- $l = \neg P_m^n(t_1, \ldots, t_n)$ *y* $l^* = P_m^n(t_1, \ldots, t_n)$.

El conjunto de fórmulas de $\mathscr{L}$ se define inductivamente de la siguiente manera.

Definición 2.4 (fórmula) *El conjunto de fórmulas de $\mathscr{L}$ es el mínimo conjunto que cumple con las siguientes condiciones:*

1. *Todas las fórmulas atómicas son fórmulas de $\mathscr{L}$.*
2. *Si φ es una fórmula de $\mathscr{L}$, entonces $\neg\varphi$ es una fórmula de $\mathscr{L}$.*
3. *Si φ y ψ son fórmulas de $\mathscr{L}$, entonces $\varphi \wedge \psi$ y $\varphi \vee \psi$ son fórmulas de $\mathscr{L}$.*
4. *Si φ es una fórmula de $\mathscr{L}$ y x es una variable, entonces $\exists x\varphi$ y $\forall x\varphi$ son fórmulas de $\mathscr{L}$.*

El conjunto de fórmulas de $\mathscr{L}$ lo denotaremos como $FORM$.

Definición 2.5 (complejidad de una fórmula) *La complejidad de una fórmula φ se define como el número de conectivos y cuantificadores que figuran en φ.*

En el caso de las fórmulas que contienen cuantificadores, convenimos que la parte de la fórmula a la cual afecta el cuantificador es la mínima fórmula a la derecha de éste. Dicha caracterización es llamada el *alcance* del cuantificador y se define formalmente de la siguiente manera:

Definición 2.6 (alcance de un cuantificador) *Sea φ una fórmula de $\mathscr{L}$. El alcance del cuantificador existencial (universal) en la fórmula $\exists x\varphi$ ($\forall x\varphi$) es la fórmula φ.*

Definición 2.7 (variable libre y variable ligada) *Sean x una variable y φ una fórmula de $\mathscr{L}$. Una presencia de x en φ es ligada si es la variable de un cuantificador o figura dentro del alcance de un cuantificador de x. Si una presencia de x en la fórmula φ no está ligada, decimos que está libre en φ.*

A continuación definimos recursivamente el conjunto de variables libres de una fórmula φ, denotado por $Vl(\varphi)$ y el conjunto de subfórmulas de φ, al cual denotaremos por $Sub(\varphi)$.

Definición 2.8 (conjunto de variables libres) *Sea φ, ψ y χ fórmulas de $\mathscr{L}$, el conjunto de variables libres de φ es:*

1. *Si $\varphi \in ATOM$, entonces $Vl(\varphi) = \{x_i \in Var \mid x_i \text{ figura en } \varphi\}$.*
2. *Si $\varphi = \neg\psi$, entonces $Vl(\varphi) = Vl(\psi)$*
3. *si $\varphi = \psi \wedge \chi$, entonces $Vl(\varphi) = Vl(\psi) \cup Vl(\chi)$. Análogamente si $\varphi = \psi \vee \chi$.*
4. *Si $\varphi = \exists x\ \psi$, entonces $Vl(\varphi) = Vl(\psi) \setminus \{x\}$. Análogamente si $\varphi = \forall x\ \psi$.*

Definición 2.9 (conjunto de subfórmulas) *Sean φ, ψ y χ fórmulas de $\mathscr{L}$. El conjunto de subfórmulas de φ es:*

1. *Si $\varphi \in ATOM$, entonces $Sub(\varphi) = \{\varphi\}$*
2. *si $\varphi = \neg\psi$, entonces $Sub(\varphi) = \{\varphi\} \cup Sub(\psi)$*
3. *si $\varphi = \psi\wedge\chi$, entonces $Sub(\varphi) = \{\varphi\}\cup Sub(\psi)\cup Sub(\chi)$. Análogamente si $\varphi = \psi\vee\chi$.*
4. *si $\varphi = \exists y\ \psi$, entonces $Sub(\varphi) = \{\varphi\} \cup Sub(\psi)$. Análogamente si $\varphi = \forall y\ \psi$.*

Definición 2.10 (fórmula cerrada) *Una fórmula φ es cerrada si no contiene variables libres, es decir, si $Vl(\varphi) = \varnothing$. Una fórmula cerrada también es llamada enunciado o sentencia.*

Son justamente las fórmulas cerradas o enunciados en los cuales estamos interesados para estudiar la abducción. Por este motivo de aquí en adelante restringimos las definiciones y resultados a enunciados, excepto que las definiciones o demostraciones se apliquen indistintamente.

Definición 2.11 (sustitución) *Sean φ una fórmula, x una variable y t un término. $\varphi(t/x)$ representa la fórmula obtenida a partir de φ al reemplazar todas las presencias libres de la variable x por el término t. Siempre que x sea libre (sustituible) para t en $\varphi(x)$ siempre y cuando se evite la captura de variables libres de t en φ.*

2.2. Semántica

Como mencionamos anteriormente nuestro trabajo no estará basado en la semántica usual de los lenguajes de primer orden. Sin embargo, nuestra base sigue siendo la semántica tarskiana, por lo que será necesario revisar algunos de los conceptos más importantes para resultados posteriores.

Definición 2.12 ($\mathscr{L}$-estructura) *Una $\mathscr{L}$-estructura o $\mathscr{L}$-interpretación es un par $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, donde $\mathcal{D}$ es un conjunto no vacío llamado* dominio *o* universo *de $\mathcal{M}$ e $\mathcal{I}$ es una función denominada función de interpretación definida en las constantes y símbolos de predicados de $\mathscr{L}$ de la siguiente manera:*

1. *Si $c \in Cons$, entonces $\mathcal{I}(c) \in \mathcal{D}$,*

2. *Si $P^n_m \in Pred$, entonces $\mathcal{I}(P^n_m) \subseteq \mathcal{D}^n$.*

Observación 2.2 *Notemos que esta definición de interpretación es exclusiva de lenguajes de primer orden cuyas fórmulas son enunciados, dado que no mencionamos cómo interpretar a las variables que figuran libres. El hecho de pedir que la función sea inyectiva es para evitar que un elemento del dominio tenga más de un nombre, lo cual no restringe la clase de modelos.*

Definición 2.13 (c-equivalencia de $\mathscr{L}$-estructuras) *Sean $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ y $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ dos $\mathscr{L}$-estructuras y c una constante de $\mathscr{L}$. Decimos que $\mathcal{M}$ y $\mathcal{M}'$ son c-equivalentes si $\mathcal{D} = \mathcal{D}'$ y las funciones de interpretación $\mathcal{I}$ e $\mathcal{I}'$ difieren a lo más en la constante c, esto es, $\mathcal{I} \restriction_{\mathscr{L}\setminus\{c\}} = \mathcal{I}' \restriction_{\mathscr{L}\setminus\{c\}}$. En símbolos $\mathcal{M} =_c \mathcal{M}'$.*

Definición 2.14 (satisfacción) *Dada una $\mathscr{L}$-estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ y un enunciado φ de $\mathscr{L}$, la relación de $\mathcal{M}$ satisface a φ, en símbolos $\mathcal{M} \models \varphi$, se define de acuerdo con las siguientes reglas:*

1. *Si $\varphi = P^n_m(c_1, c_2, \ldots, c_n)$ con $n \geq 1$, $\mathcal{M} \models \varphi$ syss*[2] *$(\mathcal{I}(c_1), \ldots, \mathcal{I}(c_n)) \in \mathcal{I}(P^n_m)$.*

2. *Si $\varphi = \neg\psi$, $\mathcal{M} \models \varphi$ syss no se cumple que $\mathcal{M} \models \psi$, en símbolos $\mathcal{M} \not\models \psi$.*

3. *Si $\varphi = \psi \vee \chi$, $\mathcal{M} \models \varphi$ syss $\mathcal{M} \models \psi$ o $\mathcal{M} \models \chi$.*

4. *Si $\varphi = \psi \wedge \chi$, $\mathcal{M} \models \varphi$ syss $\mathcal{M} \models \psi$ y $\mathcal{M} \models \chi$.*

5. *Si $\varphi = \exists x\ \psi$, $\mathcal{M} \models \varphi$ syss existe una $\mathscr{L}$-estructura $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$, tal que, $\mathcal{M} =_c \mathcal{M}'$ donde c es una constante que no figura en φ y $\mathcal{M}' \models \psi(c/x)$.*

6. *Si $\varphi = \forall x\ \psi$, $\mathcal{M} \models \varphi$ syss para cualquier $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ tal que $\mathcal{M}' =_c \mathcal{M}$, donde c es una constante que no figura en φ y $\mathcal{M}' \models \psi(c/x)$.*

[2]En adelante abreviaremos la frase "si y sólo si" con "syss".

Dado un conjunto $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ *de enunciados de* $\mathscr{L}$ *y una* $\mathscr{L}$*-estructura* $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$. *Decimos que* $\mathcal{M}$ *satisface a* Γ *si* $\mathcal{M}$ *satisface a cada una de los enunciados de* Γ, *es decir,* $\mathcal{M} \models \varphi_i$ *para toda* $1 \leq i \leq n$. *En símbolos (abusando de la notación)* $\mathcal{M} \models \Gamma$.

Observación 2.3 *Cabe hacer notar que cuando hablamos de fórmulas cerradas las nociones de satisfacción y verdad son equivalentes.*

Definición 2.15 (modelo) *Dadas una* $\mathscr{L}$*-estructura* $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ *y* φ *un enunciado de* $\mathscr{L}$. *Decimos que* $\mathcal{M}$ *es un modelo de* φ *si* $\mathcal{M} \models \varphi$.

Definición 2.16 (satisfactibilidad, validez) *Un enunciado* φ *de* $\mathscr{L}$ *es satisfactible si* φ *tiene un modelo. Decimos que* φ *es válido si toda* $\mathscr{L}$*-estructura* $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ *es un modelo de* φ.

Definición 2.17 (consecuencia lógica) *Sean* Γ *un conjunto finito de enunciados de* $\mathscr{L}$ *y* φ *un enunciado de* $\mathscr{L}$. *Decimos que* φ *es consecuencia lógica de* Γ *si todo modelo de* Γ *es modelo de* φ. *En símbolos,* $\Gamma \models \varphi$.

Proposición 2.1 (principio de refutación) *Sean* Γ *un conjunto finito de enunciados y* φ *un enunciado de* $\mathscr{L}$. $\Gamma \models \varphi$ *syss* $\Gamma \cup \{\neg\varphi\}$ *es no satisfactible.*

Demostración.
$\Longrightarrow$) (Contrapositiva) Supongamos que $\Gamma \cup \{\neg\varphi\}$ es satisfactible, entonces existe una $\mathscr{L}$-estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $\mathcal{M} \models \Gamma \cup \{\neg\varphi\}$, esto es, $\mathcal{M} \models \Gamma$ y $\mathcal{M} \models \neg\varphi$. Como $\mathcal{M}$ es un modelo para $\neg\varphi$, entonces $\mathcal{M}$ no es un modelo para φ. Por lo que $\Gamma \not\models \varphi$.
$\Longleftarrow$) Supongamos que $\Gamma \cup \{\neg\varphi\}$ es insatisfactible, esto significa que no existe un modelo $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $\mathcal{M} \models \Gamma \cup \{\neg\varphi\}$. De lo anterior tenemos que para toda $\mathscr{L}$-estructura $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ si $\mathcal{M}' \models \Gamma$ entonces $\mathcal{M}' \not\models \neg\varphi$, por lo tanto, $\mathcal{M}' \models \varphi$. De donde concluimos que $\Gamma \models \varphi$. $\dashv$

2.2.1. Interpretaciones de Herbrand

Definición 2.18 (universo de Herbrand) *Sea* Γ *un conjunto finito de enunciados de* $\mathscr{L}$. *El universo de Herbrand de* Γ *son todas las constantes que aparecen en los enunciados de* Γ, *denotado por* $\mathcal{D}_\Gamma$. *Si no aparece ninguna constante en los enunciados de* Γ *entonces definimos* $\mathcal{D}_\Gamma = \{c_1\}$.

Definición 2.19 (interpretaciones de Herbrand) *Una interpretación de Herbrand para un conjunto finito de enunciados* Γ *de* $\mathscr{L}$ *es una* $\mathscr{L}$*-estructura* $\mathcal{H} = \langle \mathcal{D}_\Gamma, \mathcal{I}_\mathcal{H} \rangle$ *tal que:*

- $\mathcal{D} = \mathcal{D}_\Gamma$. *Es decir el universo de* $\mathcal{M}$ *es el universo de Herbrand de* Γ.
- *para cualquier* $c \in \mathcal{D}_\Gamma$, $\mathcal{I}_\mathcal{H}(c) = c$. *Es decir, los símbolos de constante se interpretan como ellos mismos.*

2.2.2. Satisfacción restringida

Ya antes habíamos mencionado que presentaríamos conceptos de satisfacción y de consecuencia lógica restringida. Estos conceptos son relativos a la cardinalidad del universo de discurso y por lo tanto los llamamos n-satisfacción y n-consecuencia lógica.

Definición 2.20 (n-satisfacción) *Sea φ un enunciado de $\mathscr{L}$, decimos que φ es n-satisfactible si existe una $\mathscr{L}$-estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $|\mathcal{D}| = n$ con $n \geq 1$ y $\mathcal{M} \models \varphi$. En símbolos, $\mathcal{M} \models_n \varphi$.*

El siguiente resultado fue publicado en [Ne02], la siguiente demostración detallada es aportación nuestra.

Teorema 2.1 *Si un enunciado φ es n-satisfactible, para $n \geq 1$, entonces φ es m-satisfactible para toda $m \geq n$.*

Demostración. Sean $\mathcal{D}$ un universo de cardinalidad $n \geq 1$ y $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ una $\mathscr{L}$-estructura tal que $\mathcal{M} \models_n \varphi$. Sea $\mathcal{D}^*$ un dominio con $|\mathcal{D}^*| = m \geq n$, y $\mathcal{D}' \subseteq \mathcal{D}^*$ tal que $|\mathcal{D}'| = n$. Sea $f : \mathcal{D}' \to \mathcal{D}$ una función biyectiva. Definimos la función $g : \mathcal{D}^* \to \mathcal{D}$, de la siguiente manera:

1. Si $r \in \mathcal{D}'$, entonces $g(r) = f(r)$,

2. $g(r) = s$, en otro caso (donde s es un elemento distinguido de $\mathcal{D}$).

Ahora bien, definimos $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ tal que $\mathcal{I}^*$ cumple con las siguientes condiciones: Sea c una constante de $\mathscr{L}$, entonces,

$$\mathcal{I}^*(c) = d' \in \mathcal{D}' \text{ si } \mathcal{I}(c) = g(d')$$

Observación 2.4 *Notemos que $\mathcal{I}(c) = g(\mathcal{I}^*(c))$ para toda $c \in Cons$.*

Además, dados $(s_1, \ldots, s_k) \in \mathcal{D}^{*k}$ y $P^k_m \in Pred$, definimos $\mathcal{I}^*(P^k_m)$ como:

$$(s_1, \ldots, s_k) \in \mathcal{I}^*(P^k_n) \text{ syss } (g(s_1), \ldots, g(s_k)) \in \mathcal{I}(P^k_m).$$

Una vez definida $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ de la manera anterior, podemos demostrar que $\mathcal{M} \models_n \varphi$ syss $\mathcal{M}^* \models_m \varphi$. Para realizar esta demostración antes probaremos el siguiente resultado:

Lema 2.1 *Sean $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ $\mathscr{L}$-estructuras y c una constante de $\mathscr{L}$. Sean $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ y $\mathcal{M}'^* = \langle \mathcal{D}'^*, \mathcal{I}'^* \rangle$ $\mathscr{L}$-estructuras definidas a partir de $\mathcal{M}$ y $\mathcal{M}'$, respectivamente, con la construcción antes descrita. Entonces,*

$$\mathcal{M} =_c \mathcal{M}' \textit{ syss } \mathcal{M}^* =_c \mathcal{M}'^*$$

Demostración. Si se cumple $\mathcal{M} =_c \mathcal{M}'$, por definición tenemos que

$$\mathcal{I}(c') = \mathcal{I}'(c') \text{ para toda } c' \neq c.$$

Y por la observación 2.4

$$g(\mathcal{I}^*(c')) = g(\mathcal{I}'^*(c')) \text{ para toda } c' \neq c.$$

Recordemos que $\mathcal{I}^*(c') \in D'$ e $\mathcal{I}^*(c') \in D$, además de que $g \restriction_{\mathcal{D}'} = f$ con f una función biyectiva, por lo que,

$$\mathcal{I}^*(c') = \mathcal{I}'^*(c')$$

De lo cual concluimos, $\mathcal{M}'^* =_c \mathcal{M}^*$. ⊣

Una vez demostrado este lema, continuemos con la demostración del Teorema 2.1, la cual haremos por inducción sobre la construcción de φ.

- $\varphi = P^k_m(c_1, \ldots, c_k)$.
  $$\begin{array}{lll} \mathcal{M} \models \varphi & \text{syss} & (\mathcal{I}(c_1), \ldots, \mathcal{I}(c_k)) \in \mathcal{I}(P^k_m) \\ & \text{syss} & (g(\mathcal{I}^*(c_1)), \ldots, g(\mathcal{I}^*(c_k))) \in \mathcal{I}(P^k_m) \\ & \text{syss} & (\mathcal{I}^*(c_1), \ldots, \mathcal{I}^*(c_k)) \in \mathcal{I}^*(P^k_m) \\ & \text{syss} & \mathcal{M}^* \models P^k_m(c_1, \ldots, c_k) \end{array}$$

- En el caso de los conectivos lógicos, la demostración se sigue de la definición de satisfacción y de la hipótesis inductiva. Los casos más interesantes son cuando el enunciado es una cuantificación. Sea $\varphi = \exists x \psi$.
  $\mathcal{M} \models \varphi$ syss $\mathcal{M}' \models \psi(c/x)$ para alguna $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ tal que $\mathcal{M}' =_c \mathcal{M}$.
  Por hipótesis inductiva y por el lema anterior,

  $$\mathcal{M}' \models \psi(c/x) \text{ syss } \mathcal{M}'^* \models \psi(c/x) \text{ para alguna } \mathcal{M}'^* =_c \mathcal{M}^*.$$

  Por lo tanto, $\mathcal{M}^* \models \exists x\, \psi$.
  El caso $\varphi = \forall x\, \psi$ se demuestra de manera similar.

De lo anterior queda demostrado que $\mathcal{M} \models_n \varphi$ syss $\mathcal{M}^* \models_m \varphi$. ⊣

2.2.3. n-consecuencia lógica

La noción de n-consecuencia lógica se obtiene de manera inmediata a partir de la noción de n-satisfacción de la manera usual.

Definición 2.21 (n-consecuencia lógica) *Dados un conjunto finito de enunciados Γ y un enunciado φ. Decimos que φ es n-consecuecia lógica de Γ si para toda estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ con $|\mathcal{D}| = n$, si $\mathcal{M} \models \Gamma$, entonces $\mathcal{M} \models \varphi$. En símbolos, $\Gamma \models_n \varphi$.*

Teorema 2.2 *Sea Γ un conjunto de enunciados y φ un enunciado. $\Gamma \models_n \varphi$ syss $\Gamma \cup \{\neg\varphi\}$ no es n-satisfactible.*

Demostración.
$\Longrightarrow$)(Por contradicción) Supongamos que $\Gamma \models_n \varphi$ y que $\Gamma \cup \{\neg\varphi\}$ es n-satisfactible, es decir, existe un modelo $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ con $|\mathcal{D}'| = n$ tal que,

$$\mathcal{M}' \models \Gamma \cup \{\neg\varphi\}$$

esto es,

$$\mathcal{M}' \models_n \Gamma \;\text{ y }\; \mathcal{M}' \models_n \neg\varphi$$

Pero por hipótesis tenemos que cualquier modelo de cardinalidad n de Γ es modelo de φ, en particular $\mathcal{M}'$. De donde,

$$\mathcal{M}' \models \varphi \;\text{ y }\; \mathcal{M}' \models \neg\varphi$$

Lo cual es absurdo. Por lo tanto, $\Gamma \cup \{\neg\varphi\}$ no es n-satisfactible.
$\Longleftarrow$) Supongamos que $\Gamma \cup \{\neg\varphi\}$ no es n-satisfactible, por lo tanto, no existe un modelo de cardinalidad n para $\Gamma \cup \{\neg\varphi\}$. Por otro lado, sea $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $\mathcal{M} \models_n \Gamma$. Ahora bien, dado que φ es un enunciado tenemos las siguientes dos opciones:

$$\mathcal{M} \models_n \varphi \;\text{ y }\; \mathcal{M} \models_n \neg\varphi$$

Pero por hipótesis tenemos que no existe un modelo de cardinalidad n que satisfaga a Γ y satisfaga a $\neg\varphi$. Por lo tanto, $\mathcal{M} \models_n \varphi$.

$\dashv$

Corolario 2.1 *Sea Γ un conjunto de enunciados y φ un enunciado. Si $\Gamma \models_n \varphi$, entonces $\Gamma \models_m \varphi$ para toda m con $1 \leq m < n$.*

Demostración. Supongamos que $\Gamma \not\models_m \varphi$, entonces $\Gamma \cup \{\neg\varphi\}$ es m-satisfactible y por el teorema 2.1 tendríamos que $\Gamma \cup \{\neg\varphi\}$ es n-satisfactible, contradiciendo que $\Gamma \models_n \varphi$. Por lo tanto, $\Gamma \models_m \varphi$ para toda m con $1 \leq m < n$. $\dashv$

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

Isaac Newton.

3
Tablas semánticas

3.1. Introducción

Las tablas semánticas de Beth fueron concebidas como un procedimiento de decisión para responder si $\Gamma \models \varphi$. Este método de demostración por refutación, inherentemente semántico, pero definido según la sintaxis de las fórmulas, proporciona un procedimiento efectivo para decidir la consecuencia lógica en el caso de la lógica proposicional, el cual se pierde en el caso de la lógica de primer orden al existir consecuencias válidas que generan tablas con ramas infinitas, impidiendo así la terminación del proceso.

Una solución a este problema fue presentada por George Boolos ([Bo84]) y Emilio Díaz ([Di93]) de manera independiente. Tal solución consiste en realizar una modificación a las tablas semánticas de Beth cambiando la regla correspondiente a las fórmulas cuantificadas existencialmente. Con esta modificación se garantiza la construcción finita de una refutación, si ésta existe, siempre y cuando el dominio en discurso sea finito.
En este capítulo estudiaremos las tablas semánticas de Beth así como la modificación de Díaz y Boolos y la relación que hay entre la noción de n-satisfactibilidad expuesta en el capítulo anterior y las tablas semánticas modificadas.

3.2. Tablas semánticas para enunciados de $\mathscr{L}$

Las *tablas semánticas* son un método de demostración por refutación, el cual fue introducido por Hintikka en [Hi53] y por Beth en [Be55], quien les dio el nombre de *tableaux*. La idea principal de los tableaux es tratar de construir sistemáticamente un modelo para una fórmula satisfactible o bien, si lo que se quiere es demostrar una relación de consecuencia lógica, $\Gamma \models \varphi$. El método consiste en intentar construir un contraejemplo para el conjunto $\Gamma \cup \{\neg\varphi\}$ extendiendo el tableau correspondiente a dicho conjunto.

Años después, Raymond Smullyan reinicia las investigaciones realizadas en torno a las tablas semánticas para estudiar la lógica clásica a través de los tableaux. Como resultado de su trabajo publica [Sm63], [Sm65],[Sm66] y finalmente en 1968 publica el libro *First-Order Logic* [Sm68]. Las contribuciones más relevantes de Smullyan en este trabajo fueron: la estandarización de la sintaxis de las tablas semánticas, la introducción de tablas semánticas con signo y sin signo y la clasificación de fórmulas según su forma. La clasificación de las fórmulas dada por Smullyan depende del conectivo principal que éstas tengan, una fórmula compuesta es vista como la negación de una fórmula, o bien, como la conjunción o disyunción de dos fórmulas; definiendo con ello subfórmulas de la fórmula original, las cuales conforman la principal propiedad del procedimiento de demostración de los tableaux. Por la razón anterior, Smullyan llama a las tablas semánticas, *tablas analíticas.*

Las tablas analíticas con signo son principalmente utilizadas en la lógica de proposiciones más que en la lógica de predicados, y su nombre es dado por el hecho de que a cada fórmula que aparece en el tableau se le asigna una etiqueta para indicar si la fórmula es falsa o verdadera. A lo largo de este trabajo utilizaremos la notación propuesta por Smullyan aunque las tablas semánticas que ocuparemos serán sin signo, si una fórmula aparece en el tableau asumimos que es verdadera.

3.2.1. Clasificación de los enunciados de primer orden

Los enunciados que no son literales del lenguaje de primer orden $\mathscr{L}$ se clasifican en cuatro clases dependiendo de su forma. Sea φ un enunciado que no es literal de $\mathscr{L}$.

1. φ es de *tipo* α si es la conjunción de dos enunciados de $\mathscr{L}$, es decir $\varphi = \psi \wedge \chi$, donde ψ y χ son llamadas las *α-subfórmulas* de φ.

2. φ es de *tipo* β si es la disyunción de dos enunciados de $\mathscr{L}$, esto es, $\varphi = \psi \vee \chi$, donde ψ y χ son las *β-subfórmulas* de φ.

3. φ es de *tipo* γ si es la cuantificación universal de una fórmula de $\mathscr{L}$, digamos $\varphi = \forall x \psi$, con ψ una fórmula de $\mathscr{L}$ cuya única variable libre es x, por lo tanto, el resultado de sustituir las apariciones de la variable x por una constante c en ψ es un enunciado. A los enunciados resultado de esta sustitución se les denomina *γ-subfórmulas* de φ.

4. φ es de *tipo* δ si es la cuantificación existencial de una fórmula de $\mathscr{L}$, es decir, $\varphi = \exists x \psi$ tal que ψ es una fórmula de $\mathscr{L}$ con $Vl(\psi) = \{x\}$. A los enunciados resultado de sustituir x por una constante c, $\psi(c/x)$, se les llama *β-subfórmulas* de φ.

Podemos observar que cualquier enunciado no literal de $\mathscr{L}$ pertenece a uno y sólo uno de estos cuatro tipos. Si un enunciado no literal tiene como conectivo principal la negación, entonces podemos dar un enunciado lógicamente equivalente cuyo símbolo lógico principal no sea la negación.

El método de demostración propuesto por Smullyan se basa en la semántica de los enunciados de $\mathscr{L}$ y en la clasificación antes mencionada. La idea es la siguiente:
Sean φ un enunciado de $\mathscr{L}$ y $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ un modelo de φ.

1. Si $\varphi = \alpha_1 \wedge \alpha_2$ es de tipo α, entonces $\mathcal{M}$ debe ser modelo de las dos α-subfórmulas α_1 y α_2.

2. Si $\varphi = \beta_1 \vee \beta_2$ es de tipo β, entonces $\mathcal{M}$ debe satisfacer al menos a una de las dos β-subfórmulas β_1 o β_2.

3. Si $\varphi = \forall x\psi$ es de tipo γ, entonces se cumple que para toda $\mathscr{L}$-estructura $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ tal que $\mathcal{M}' =_c \mathcal{M}$, $\mathcal{M}'$ satisface a la γ-subfórmula $\psi(c/x)$ donde c es una constante que no aparece en φ.

4. Si $\varphi = \exists x\psi$ es de tipo δ, entonces debe existir una $\mathscr{L}$-estructura $\mathcal{M}' = \langle \mathcal{D}', \mathcal{I}' \rangle$ tal que $\mathcal{M}' =_c \mathcal{M}$ que satisface a la γ-subfórmula $\psi(c/x)$ donde c es una constante que no aparece en φ.

Es justamente esta idea la que da origen a las reglas de extensión de las tablas semánticas. Estas se definen a continuación.

3.2.2. Reglas de extensión de las tablas semánticas

La tabla semántica correspondiente a un conjunto finito Γ de enunciados de $\mathscr{L}$ puede ser representada por medio de un árbol que tiene las siguientes propiedades:

a) Cada uno de los nodos está etiquetado con un enunciado de $\mathscr{L}$.

b) Nos referiremos a una rama de un tableau como el conjunto de enunciados que aparecen en ella, por lo cual las denotaremos con letras griegas mayúsculas al igual que lo hemos venido haciendo con los conjuntos de fórmulas. Los nodos de la rama inicial están etiquetados con los enunciados de $\Gamma \neq \varnothing$, a este conjunto le llamaremos *conjunto inicial.*

c) Las relaciones de subárbol derecho y subárbol izquierdo están dadas por las siguientes reglas de expansión: Sea Φ una rama del tableau de Γ y φ un enunciado en Ψ, extendemos la rama Φ de acuerdo a las siguientes reglas:

   1. Si φ es de tipo α la regla de expansión correspondiente es la *α-regla*, la cual opera de la siguiente forma:

$$\frac{\alpha_1 \wedge \alpha_2}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}}$$

   La aplicación de esta regla agrega dos nuevos nodos a la rama Φ, etiquetados con las α-subfórmulas α_1 y α_2. En símbolos: $\Phi + \alpha_1 + \alpha_2$.

2. Si φ es de tipo β, la regla correspondiente es la *β-regla* la cual se define de la manera siguiente:
$$\frac{\beta_1 \vee \beta_2}{\beta_1 \mid \beta_2}$$
La β-regla agrega dos nuevas ramas, cada una etiquetada con una de las β-subfórmulas β_1 y β_2. En símbolos: $\Phi + \beta_1$ y $\Phi + \beta_2$.
3. Si φ es de tipo γ, la regla que se aplicará es la *γ-regla*, la cual opera como sigue:
$$\frac{\forall x\ \psi}{\begin{array}{c}\varphi(c_1/x)\\ \varphi(c_2/x)\\ \vdots\\ \varphi(c_n/x)\end{array}}$$
Esta regla agrega tantos nodos como constantes aparecen en la rama Φ, cada uno de los cuales está etiquetado con una γ-subfórmula $\varphi(c_i/x)$ con c_i constante de Φ. Es decir, si $c_1, c_2, \ldots, c_n$ (con $n \geq 1$) son las constantes que aparecen en la rama Φ (en caso de que ninguna constante aparezca, $n = 1$), entonces el resultado de aplicar la regla γ es el siguiente: $\Phi + \varphi(c_1/x) + \varphi(c_2/x) + \ldots + \varphi(c_n/x)$.
Esta regla tiene la peculiaridad de que permanece activa, con lo cual queremos decir que siempre que una nueva constante c aparezca en Φ como resultado de la aplicación de una regla de expansión, debemos agregar un nuevo nodo etiquetado con la γ-subfórmula correspondiente a c, la cual es $\varphi(c/x)$.
4. Si φ es de tipo δ, la regla que debe aplicarse es la *δ-regla*, la cual opera como:
$$\frac{\exists x\ \psi}{\psi(c/x)}$$
La δ-regla agrega un nuevo nodo etiquetado con la δ-subfórmula $\psi(c/x)$, donde c es una constante que no aparece antes en Φ. La extensión es la siguiente: $\Phi + \psi(c/x)$.

La tabla semántica correspondiente a un conjunto Γ de enunciados de $\mathscr{L}$ no es única, pues un tableau válido del conjunto Γ es el que está formado únicamente por los enunciados de Γ y también es válido cualquier tableau resultante de aplicar una regla de extensión a cualquiera de los enunciados de Γ, o cualquiera de los enunciados de un tableau extendido de Γ. Formalmente, la definición de un tableau válido de Γ es la siguiente:

Definición 3.1 (tableau de un conjunto finito de enunciados de $\mathscr{L}$) *Sea $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ un conjunto finito de enunciados de $\mathscr{L}$, un tableau de Γ, denotado por $\mathcal{T}(\Gamma)$, se define como:*

1. *Cualquier árbol con una sola rama cuyos nodos están etiquetados, respectivamente, con los enunciados $\varphi_1, \ldots, \varphi_n$ es un tableau de Γ.*

2. *Si $\mathcal{T}_1$ es un tableau de Γ, entonces la extensión que se obtiene después de aplicarle alguna de las reglas α, β, γ y δ también es un tableau de Γ.*

3. *Únicamente son tableaux de Γ aquellos que se obtienen con 1 y 2.*

Los tableaux son un método basado en la descomposición de las fórmulas. Si en algún momento de este proceso de descomposición, llamado extensión de tableaux, aparece en una rama un enunciado (atómico) y su negación, tendremos que el conjunto inicial de fórmulas no es satisfactible. Lo anterior motiva la siguiente definición y la propiedad fundamental de las tablas semánticas de Beth.

Definición 3.2 (tableau cerrado) *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$ y $\mathcal{T}(\Gamma)$ un tableau de Γ. Decimos que una rama Φ de $\mathcal{T}(\Gamma)$ es* cerrada (atómicamente) *si un enunciado (atómico) φ y su negación aparecen en Φ. Un tableau es* cerrado (atómicamente) *si todas las ramas que lo forman son cerradas (atómicamente).*

Definición 3.3 (tableau abierto) *Una rama Φ de un tableau es* abierta *si no es cerrada. Un tableau se considera* abierto *si contiene al menos una rama abierta.*

Lema 3.1 *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$ y $\mathcal{T}(\Gamma)$ un tableau de Γ. Para cualquier rama Φ de $\mathcal{T}(\Gamma)$ tal que Φ es cerrada se cumple que es posible construir una rama cerrada atómicamente a partir de Φ.*

Demostración. Sea Φ una rama cerrada, por definición Φ contiene un enunciado y su negación, llamémosles φ y $\neg\varphi$, respectivamente. La demostración se hará por inducción sobre la complejidad de φ.

- Si φ es una fórmula atómica, se cumple el lema.

- Si φ es de tipo α, $\varphi = \varphi_1 \wedge \varphi_2$, entonces $\neg\varphi$ es de tipo β, $\neg\varphi = \neg(\varphi_1 \wedge \varphi_2)$. Así que al aplicar la α-regla a φ obtenemos la siguiente rama

$$\Phi + \varphi_1 + \varphi_2$$

  y si después aplicamos la β-regla a $\neg\varphi$ obtenemos dos ramas:

$$\Phi_1 = \Phi + \varphi_1 + \varphi_2 + \neg\varphi_1$$

$$\Phi_2 = \Phi + \varphi_1 + \varphi_2 + \neg\varphi_2$$

  Por hipótesis inductiva, es posible constuir a partir de cualquiera de ellas, una rama cerrada atómicamente.

- Si φ es tipo β, la demostración es similar al caso anterior.

- Si φ es tipo γ, $\varphi = \forall x\psi$, entonces $\neg\varphi$ es tipo δ, $\neg\varphi = \neg\forall x\psi$. Aplicando primero la regla δ obtenemos las siguiente rama:

  $$\Phi_1 = \Phi + \neg\psi(c/x)$$

  Con c es una constante nueva en Φ. Ahora bien, al aplicar a esta rama la γ-regla de extensión al enunciado φ, obtenemos:

  $$\Phi_2 = \Phi_1 + \psi(c_1/x) + \ldots + \psi(c_m/x) + \psi(c/x)$$

  Donde $c_1, c_2, \ldots, c_m$ son las constantes que aparecían en Φ. Por lo tanto, Φ_2 contiene un enunciado y su negación, tal enunciado es de menor complejidad que φ. Así que, por hipótesis inductiva es posible construir una rama cerrada atómicamente a partir de Φ_2.

- Si φ es tipo δ, la demostración es similar al caso anterior.

De lo anterior, podemos concluir que siempre que una rama es cerrada es posible cerrarla atómicamente. ⊣

El método de tablas semánticas es un método de refutación completo y correcto. Antes de presentar las demostraciones correspondientes, introducimos la noción de *conjunto descendientemente saturado*, el cual será de gran utilidad.

Definición 3.4 (conjunto descendientemente saturado) *Sea Γ un conjunto finito de enunciados de la lógica de primer orden y $\mathcal{D}_\Gamma$ el universo de Herbrand de Γ. El conjunto Γ se llama* descendientemente saturado *si cumple con las siguientes reglas:*

1. *Si Γ contiene un enunciado α, entonces Γ contiene las dos α-subfórmulas.*

2. *Si Γ contiene un enunciado β, entonces Γ contiene al menos una de las β-subfórmulas.*

3. *Si Γ contiene un enunciado $\gamma = \forall x\ \psi$, entonces para cada $c_i \in \mathcal{D}_\Gamma$ la γ-subfórmula $\psi(c_i/x)$ está en Γ.*

4. *Si Γ contiene un enunciado $\delta = \exists x\ \psi$, entonces Γ contienen al menos una δ-subfórmula $\psi(c/x)$ con $c \in \mathcal{D}_\Gamma$.*

De manera similar, definimos un *tableau saturado* de un conjunto de enunciados Γ como un tableau de Γ para el cual ya no es posible realizar ninguna extensión.

Definición 3.5 (tableau saturado) *Sea $\mathcal{T}(\Gamma)$ un tableau de un conjunto finito de enunciados Γ. Y Φ una rama de $\mathcal{T}(\Gamma)$. Decimos que Φ es saturada si el conjunto formado por todas las fórmulas que aparecen en Φ, es un conjunto descendientemente saturado. $\mathcal{T}(\Gamma)$ es saturado si cada una de sus ramas es saturada.*

Definición 3.6 (conjunto de Hintikka) *Un* conjunto de Hintikka (atómico) *es un conjunto descendientemente saturado el cual no contiene ningún enunciado (atómico) y su negación.*

Lema 3.2 *Sea* Γ *un conjunto de Hintikka, entonces* Γ *tiene un modelo de Herbrand.*

Demostración. Definimos una estructura de Herbrand $\mathcal{H} = \langle \mathcal{D}_\Gamma, \mathcal{I}_\mathcal{H} \rangle$ para Γ de acuerdo con las siguientes reglas:

1. $\mathcal{D}_\Gamma = \{c \mid$ con c es una constante que figura en $\Gamma\}$
2. $\mathcal{I}_H(c_i) = c_i$ para todo $c_i \in \mathcal{D}_\Gamma$.
3. $\mathcal{I}_H(P^k_m) = \{(c_1, \ldots, c_k) \in \mathcal{D}^k_\Gamma \mid P^n_m(c_1, \ldots, c_k) \in \Gamma\}$

Una vez definida $\mathcal{H} = \langle \mathcal{D}_\Gamma, \mathcal{I}_\mathcal{H} \rangle$ únicamente nos falta demostrar que $\mathcal{H} \models \varphi$ para toda $\varphi \in \Gamma$, para lo cual proseguiremos por inducción sobre el grado de complejidad de φ que: Si $\varphi \in \Gamma$ entonces $\mathcal{H} \models \varphi$.

El caso base es obvio. Sea $\varphi = P^k_m(c_1, \ldots, c_k) \in \Gamma$, entonces de la definición de $\mathcal{H}$ se sigue directamente que $\mathcal{H} \models P^k_m(c_1, \ldots, c_k)$.

- Si $\varphi = \alpha_1 \wedge \alpha_2$ (tipo α) por hipótesis tenemos que $\alpha_1, \alpha_2 \in \Gamma$ y como el grado de complejidad de α_1 y α_2 es menor que el de α, entonces $\mathcal{H} \models \alpha_1$ y $\mathcal{H} \models \alpha_1$. Por lo tanto, $\mathcal{H} \models \alpha_1 \wedge \alpha_2$.

- Si $\varphi = \beta_1 \vee \beta_2$ (tipo β), entonces tenemos que $\beta_1 \in \Gamma$ o $\beta_2 \in \Gamma$. Sin pérdida de generalidad, supongamos que $\beta_1 \in \Gamma$, dado que el grado de complejidad de β_1 es menor que el grado de complejidad de φ, por hipótesis inductiva tenemos que $\mathcal{H} \models \beta_1$. De lo cual concluimos que $\mathcal{H} \models \beta_1 \vee \beta_2$.

- Si $\varphi = \forall x \psi$ (tipo γ) se cumple que $\psi(c_i/x) \in \Gamma$ para toda $c_i \in \mathcal{D}_\Gamma$. Y como el grado de complejidad de los enunciados $\psi(c_i/x)$ es menor que el de φ, por hipótesis inductiva tenemos $\mathcal{H} \models \psi(c_i/x)$ para toda $c_i \in \mathcal{D}_\Gamma$. Por lo tanto podemos concluir, $\mathcal{H} \models \forall x \psi$.

- Si $\varphi = \exists x \psi$ (tipo δ) se cumple que $\psi(c/x) \in \Gamma$ para alguna $c \in \mathcal{D}_\Gamma$; ya que que el grado de complejidad de $\psi(c/x)$ es menor que el de φ, por hipótesis inductiva tenemos que $\mathcal{H} \models \psi(c/x)$. Por lo tanto, $\mathcal{H} \models \exists x \psi$.

De lo anterior concluimos que $\mathcal{H} \models \Gamma$. $\dashv$

3.3. Tablas semánticas modificadas (DB-tableaux)

El método de tablas de Beth para la lógica de primer orden presenta el defecto de que la combinación de las reglas de extensión γ y δ puede generar ramas infinitas en un árbol aun cuando la fórmula analizada sea satisfactible en dominios finitos. Veamos el siguiente ejemplo:

$$
\begin{array}{c}
\forall x \exists y Pxy \\
| \\
\exists y Pay \\
| \\
Pab \\
| \\
\exists y Pby \\
| \\
Pbc \\
| \\
\exists y Pcy \\
| \\
Pcd \\
\vdots
\end{array}
$$

Podemos ver que la única rama posible de construir es infinita, aun cuando la fórmula es satisfactible en un dominio de cardinalidad uno, por lo que el método de tablas de Beth resulta inútil en este caso para construir un modelo.

A continuación presentamos una modificación del método de Beth, introducida en [Bo84] y [Di93], que resuelve este problema al transformar cualquier árbol infinito de un conjunto de fórmulas con modelo finito obtenido con las reglas de Beth en un árbol con ramas finitas y acabadas, las cuales permiten construir el modelo finito en cuestión.

Definición 3.7 (DB-tableaux) *Un DB-*tableau *es un tableau construido a través de la α-regla, la β-regla y la γ-regla, además de una nueva regla denominada la δ'-regla cuya definición es la siguiente:*

$$\frac{\exists x\varphi}{\varphi(c_1/x)|\ldots|\varphi(c_{m+1}/x)}$$

donde c_i con $1 \leq i \leq m$ son todas las constantes que ocurren en la rama Φ a la cual el enunciado $\exists x\varphi$ pertenece y c_{m+1} es una nueva constante. Los enunciados $\varphi(c_i/x)$ son una δ'-subfórmula para cada $1 \leq i \leq m+1$. La rama Φ se extiende de la siguiente forma:

$$\Phi + \varphi(c_1/x); \quad \Phi + \varphi(c_2/x); \ \ldots; \ \Phi + \varphi(c_m/x) \quad ; \Phi + \varphi(c_{m+1}/x).$$

La regla δ' genera en el ejemplo anterior una rama abierta y acabada, a saber:

$$\{\forall x \exists y Rxy, \exists y Rc_1y, Rc_1c_1\}$$

la cual genera un modelo de cardinalidad uno para $\forall x \exists y Rxy$, a saber $\langle \{c_1\}, \{c_1, c_1\} \rangle$.

Los siguientes conceptos acerca de los DB-tableau serán útiles más adelante.

Definición 3.8 (peso) *Una rama de un DB-tableau tiene* peso *n, para $n \geq 1$, si n es el número de constantes (no repetidas) que figuran en la rama.*

Definición 3.9 (rama saturada) *Sean Γ un conjunto finito de enunciados, $\mathcal{T}_{DB}(\Gamma)$ un DB-tableau de Γ y Φ una rama de $\mathcal{T}_{DB}(\Gamma)$. Decimos que Φ es saturada si el conjunto formado por todas las fórmulas que aparecen en Φ es un conjunto descendientemente saturado.*

Observación 3.1 *Notemos que cualquier rama Φ abierta y saturada de un DB-tableau de Γ forma un conjunto de Hintikka.*

3.3.1. Correctitud de los DB-tableaux

En esta sección presentamos la correctitud del método de tablas modificado mediante la propiedad fundamental de los DB-tableaux. Aunque estos resultados aparecen en [Ne99], ese artículo no contiene demostraciones detalladas, las cuales presentamos aquí. Esta aportación nuestra se obtuvo en colaboración directa con el autor de [Ne99].

Lema 3.3 *Sea Γ un conjunto finito de enunciados, se cumple que para cualquier $\mathcal{T}(\Gamma)$ existe $\mathcal{T}_{DB}(\Gamma)$ tal que $\mathcal{T}(\Gamma) \subseteq \mathcal{T}_{DB}(\Gamma)$.*

Demostración. (Por inducción sobre la aplicación de las reglas)
El caso base es obvio, el tableau inicial de Γ es exactamente el mismo que el DB-tableau de Γ.
Sean $\mathcal{T}(\Gamma)$ y $\mathcal{T}_{DB}(\Gamma)$ un tableau y un DB-tableau de Γ, respectivamente, tales que $\mathcal{T}(\Gamma) \subseteq \mathcal{T}_{DB}(\Gamma)$. Si $\mathcal{T}'(\Gamma)$ se obtiene a partir de $\mathcal{T}(\Gamma)$ utilizando una de las α, β o γ reglas. Es claro que esta misma regla puede ser utilizada en $\mathcal{T}_{DB}(\Gamma)$ obteniendo la DB-tableau $\mathcal{T}'_{DB}(\Gamma)$ extensión de $\mathcal{T}_{DB}(\Gamma)$, de tal manera que $\mathcal{T}'(\Gamma) \subseteq \mathcal{T}'_{DB}(\Gamma)$. El caso interesante ocurre cuando la regla que se emplea para extender el tableau es la δ-regla. Sea

$$\mathcal{T}(\Gamma) = \{\Phi_1, \ldots, \Phi_i, \ldots, \Phi_m\}$$

donde Φ_i es la rama que contiene el enunciado $\varphi = \exists x\phi$ sobre el que se aplicará la δ-regla. Si $a_1, a_2, \ldots, a_n$ son todas las constantes que aparecen en Φ_i, entonces el resultado de dicha extensión es agregarle el nodo $\phi(a_{n+1}/x)$ a la rama Φ_i, obteniendo con ello el siguiente tableau:

$$\mathcal{T}'(\Gamma) = \{\Phi_1, \ldots, \Phi_i + \phi(a_{n+1}/x), \ldots, \Phi_m\}$$

Por hipótesis inductiva, tenemos que existe $\mathcal{T}_{DB}(\Gamma)$ tal que $\mathcal{T}(\Gamma) \subseteq \mathcal{T}_{DB}(\Gamma)$. Entonces, el resultado de aplicar la regla δ' a φ en el DB-tableau es agregar las ramas

$$\Phi_i + \phi(a_1/x), \ldots, \Phi_i + \phi(a_n/x), \Phi_i + \phi(a_{n+1}/x)$$

Por lo tanto, si

$$\mathcal{T}_{DB}(\Gamma) = \{\Phi_1, \ldots, \Phi_i, \Phi_m, \ldots, \Phi_k\}$$

con $k \geq m$, entonces el resultado de dicha extensión es:

$$\mathcal{T}'_{DB}(\Gamma) = \{\Phi_1, \ldots, \Phi_i + \phi(a_1/x), \ldots, \Phi_i + \phi(a_n/x), \Phi_i + \phi(a_{n+1}/x), \ldots, \Phi_m, \ldots, \Phi_k\}$$

De donde es claro que $\mathcal{T}'(\Gamma) \subseteq \mathcal{T}'_{DB}(\Gamma)$. $\dashv$

Lema 3.4 *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$. Si existe un $\mathcal{T}_{DB}(\Gamma)$ con una rama abierta y saturada de peso n, entonces Γ tiene un modelo de cardinalidad n.*

Demostración. Sea Φ una rama abierta y saturada de peso n de $\mathcal{T}_{DB}(\Gamma)$. Φ forma un conjunto de Hintikka y por el Lema 3.2 Φ tiene un modelo de Herbrand $\mathcal{H} = \langle \mathcal{D}_\Phi, \mathcal{I}_\mathcal{H} \rangle$ constuido de la siguiente manera:

1. $\mathcal{D}_\Phi = \{c \mid$ con c es una constante que figura en $\Phi\}$

2. $\mathcal{I}_H(c_i) = c_i$ para todo $c_i \in \mathcal{D}_\Phi$.

3. $\mathcal{I}_H(P^k_m) = \{(c_1, \ldots, c_k) \in \mathcal{D}^k_\Phi \mid P^n_m(c_1, \ldots, c_k) \in \Phi\}$

Como se cumple que, $\mathcal{H} \models \varphi$ para toda $\varphi \in \Phi$ y $\Gamma \subseteq \Phi$, entonces $\mathcal{H}$ también es modelo de Γ. Obsérvese que $|\mathcal{D}_\Phi| = n$ pues Φ es de peso n.

⊣

Definición 3.10 (conjunto de enunciados minimalmente n-satisfactible) *Un conjunto finito de enunciados Γ es minimalmente n-satisfactible si Γ tiene un modelo de cardinalidad n y no existe ningún modelo de Γ de cardinalidad menor a n.*

Lema 3.5 *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$. Si Γ es minimalmente n-satisfactible ($n \geq 1$), entonces existe un $\mathcal{T}_{DB}(\Gamma)$ que cumple con las siguientes condiciones:*

1. *$\mathcal{T}_{DB}(\Gamma)$ tiene una rama abierta y saturada de peso n.*

2. *Todas las ramas de peso m, con $m < n$, son cerradas.*

Demostración. *(Contrapositiva)* Dado que Γ es minimalmente n-satisfactible, existe $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $|\mathcal{D}| = n$ y $\mathcal{M} \models_n \Gamma$. Por otro lado, de la propiedad fundamental de los tableaux[1] y el Lema 3.3 se sigue que existe un $\mathcal{T}_{DB}(\Gamma)$ con una rama abierta. Ahora sólo nos falta demostrar que dicha rama es de peso n y que todas las ramas de peso menor son cerradas.

Sea Φ una rama abierta de peso mínimo, Φ no puede tener peso menor que n, ya que por el Lema 3.4 tendríamos un modelo de Γ de menor cardinalidad contradiciendo con ello que Γ es minimalmente n-satisfactible.

Por otro lado, si el peso de Φ fuera mayor que n significaría que todas las ramas de peso n son cerradas. Para demostrar que esto no puede ocurrir basta con ver que las reglas son correctas.

Por hipótesis tenemos que $\mathcal{M} \models_n \Gamma$. Entonces, el resultado de aplicar cualquier regla de extensión a uno de los enunciados de Γ provoca que al menos una rama siga abierta, ya que las reglas son correctas y cada aplicación da lugar a nuevas fórmulas satisfactibles por $\mathcal{M}$, en el caso de las β y δ' reglas, se agregan nuevas ramas garantizando que al menos una de ellas permanece abierta. Por lo tanto, no puede ser que todas las ramas de peso n sean cerradas, porque tendríamos que $\mathcal{M}$ satisface a dos literales complementarias.

De lo anterior, podemos concluir que Φ es de peso n. ⊣

Teorema 3.1 (propiedad fundamental de las DB-tableaux) *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$. Γ es minimalmente n-satisfactible si y sólo si existe un DB-tableau de Γ con una rama Φ abierta y saturada de peso n tal que toda rama de peso menor que n es cerrada.*

Demostración. La demostración es inmediata de los Lemas 3.4 y 3.5. ⊣

En la siguiente sección se presenta un algoritmo para constuir un modelo, dado un conjunto finito de enunciados en *forma normal prenex.*

[1] Γ es un conjunto satisfactible syss cualquier $\mathcal{T}(\Gamma)$ contiene una rama abierta.

3.3.2. Algoritmo para encontrar un modelo de un conjunto finito de enunciados n-satisfactibles

El algoritmo $buscaModelo(\Gamma, n)$ recibe como entrada un conjunto finito de enunciados Γ en forma normal prenex y un número entero positivo n. Si Γ es minimalmente m-satisfactible, con $1 \leq m \leq n$, regresa un modelo de cardinalidad m. En caso contrario, reporta que no es satisfactible en un modelo de cardinalidad menor o igual a n.

Algoritmo $buscaModelo(\Gamma, n)$

1. Si $(peso(\Phi) + numExistenciales(\Phi) < n)$ reportamos lo siguiente:

   "No existe un modelo para Γ de cardinalidad menor o igual a n".

2. En caso contrario, realizar los siguientes pasos:

   a) $\Phi = construyeRamaAS(\Gamma, n)$.

   b) Si $\Phi \neq \varnothing$ definimos la interpretación de $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, de acuerdo a las siguientes reglas:

      1) $\mathcal{D}_\Gamma = \{c \mid$ con c es una constante que figura en $\Phi\}$
      2) $\mathcal{I}(c_i) = c_i$ para todo $c_i \in \mathcal{D}_\Gamma$.
      3) $\mathcal{I}(P^k_m) = \{(c_1, \ldots, c_k) \in \mathcal{D}^k_\Gamma \mid P^n_m(c_1, \ldots, c_k) \in \Phi\}$

   c) En caso contrario reportamos que,

      Γ *no tiene un modelo de cardinalidad menor o igual a n.*

La correctitud del algoritmo $buscaModelo(\Gamma, n)$ depende directamente del algoritmo $construyeRamaAS(\Gamma, n)$. La entrada de este algoritmo es exactamente la misma que el algoritmo $buscaModelo(\Gamma, n)$.

El algoritmo $construyeRamaAS(\Gamma, n)$ utiliza una cola de prioridades para almacenar las ramas abiertas que aún no son saturadas, a la cual llamaremos *ABIERTAS*. Las ramas están ordenas de acuerdo con su peso en orden ascendente (la de menor peso está al inicio). Además, cada uno de los nodos que forma cada rama tienen una marca llamada *etiqueta*, para indicar si ya fue utilizado o no. De esta manera, una rama es saturada si cada uno de los nodos etiquetados con un enunciado distinto de una literal ha sido utilizado.

La idea intuitiva del algoritmo es la siguiente: primero inicializa la cola de prioridades *ABIERTAS* con el conjunto incial de fórmulas Γ. Mientras que no encuentra una rama abierta y saturada, o bien, *ABIERTAS* contenga al menos una rama, el algortimo extrae una rama de *ABIERTAS* y la extiende utilizando las reglas α, γ, β y δ, en este orden. Dependiendo de la regla de extensión utilizada y del peso de la rama una vez realizada la extensión, los pasos a seguir son distintos. En general, cuando una rama se cierra o su peso es mayor que n se descarta. Cuando se crean nuevas ramas por la aplicación de una de las reglas β o δ, las ramas son introducidas a la cola *ABIERTAS*, siempre y cuando, no sean cerradas ni saturadas, ni tengan peso mayor a n. El algoritmo termina cuando encuentra una rama abierta y saturada de peso menor o igual a n, o bien, cuando no hay ninguna rama abierta de peso menor o igual a n, es decir, cuando *ABIERTAS* está vacía.

Algoritmo *construyeRamaAS*(Γ, n)

Inicializar *ABIERTAS* con la rama inicial formada por todos los enunciados de Γ, cada uno de estos nodos se etiqueta con *noUsado*.

Mientras que $ABIERTAS \neq \varnothing$ (i.e., existe una rama abierta no saturada de peso menor o igual a n).

(1) Actualizar $\Phi = obtenPrimerElemento(ABIERTAS)$.[2]

(2) Si $peso(\Phi) < n$

(2.1) Si Φ contiene un nodo φ etiquetado como *noUsado* de tipo α, esto es, $\varphi = \alpha_1 \wedge \alpha_2$, ejecutar los siguientes pasos:

$$etiqueta(\varphi) = usado$$
$$\Phi = \Phi + \alpha_1 + \alpha_2$$

b) Si Φ es abierta y saturada, terminar y regresar Φ.
a) Si Φ es cerrada, ir al paso (1).
c) En otro caso, ir al paso (2.1).[3]

(2.2) Si Φ contiene un enunciado tipo γ ($\varphi = \forall x \psi$). Actualizamos Φ de la siguiente manera:

$$\Phi = \Phi + \psi(c_1/x) + \ldots + \psi(c_k/x)$$

donde c_i con $1 \leq i \leq k$ son las constantes que ocurren en Φ.[4] Se etiqueta como *usado* siempre y cuando los otros nodos no literales sean etiquetados como *usados*.

a) Si Φ es abierta y saturada, terminar y regresar Φ.
b) Si Φ es cerrada, ir a (1).
c) En caso contrario, $etiqueta(\psi(c_i/x)) = noUsado$ para toda $1 \leq i \leq n$. Ir al paso (2.1).[5]

(2.3) Si Φ contiene un enunciado φ tipo β ($\varphi = \beta_1 \vee \beta_2$).

$$etiqueta(\varphi) = usado$$

Utilizando la β-regla extender el árbol obteniendo dos nuevas ramas, $\Phi_1 = \Phi + \beta_1$ y $\Phi_2 = \Phi + \beta_2$.

a) Si alguna de las ramas Φ_1 o Φ_2 es abierta y saturada, terminar y regresarla.
b) Si Φ_1 y Φ_2 son cerradas, ir al paso (1).

---

[2]Esta función obtiene el primer elemento de *ABIERTAS* eliminándolo de la misma.

[3]Esto es porque la regla α no incrementa el peso de la rama y ésta era la de mayor peso, era la primera de *ABIERTAS*.

[4]Sólo se agregan los enunciados $\psi(c_i/x)$ que no aparecían en Φ. Además, no marcamos el enunciado φ como *noUsado* ya que si aparece una nueva constante entonces φ deberá instanciase con ésta.

[5]Φ sigue abierta y no se modifico el peso.

c) En otro caso, suponiendo que Φ_1 es la rama abierta actualiza $\Phi = \Phi_1$ e introducir la rama Φ_2 a *ABIERTAS* si ésta es abierta. Ir al paso (2.1).

(2.4) Si Φ contiene un enunciado φ tipo δ ($\varphi = \exists x\psi$).

$$etiqueta(\varphi) = usado$$

Utilizando la δ'-regla extender la rama Φ obteniendo $k+1$ nuevas ramas, si k es el número de constantes que ocurren en Φ, $\Phi_i = \Phi + \psi(c_i/x)$ para cada $1 \leq i \leq k+1$.[6]

a) Si alguna Φ_i ($1 \leq i \leq k+1$) es abierta y saturada, terminar y regresar Φ_i.

c) En cualquier otro caso, actualiza Φ con la primer rama Φ_i que sea abierta.[7] Insertar todas las ramas abiertas a *ABIERTAS*. Ir al paso (2.1).

(3) Si $peso(\Phi) = n$ y Φ no es saturada, realizar lo siguiente:

(3.1) Si Φ contiene un nodo φ etiquetado como *noUsado* de tipo α, esto es, $\varphi = \alpha_1 \wedge \alpha_2$, ejecutar los siguientes pasos:

$$etiqueta(\varphi) = usado$$
$$\Phi = \Phi + \alpha_1 + \alpha_2$$

a) Si Φ es abierta y saturada, terminar y regresar Φ.

b) Si Φ es cerrada, ir al paso (1).

c) En caso contrario,

$$etiqueta(\alpha_1) = etiqueta(\alpha_2) = noUsado$$

Regresar al paso (3.1).

(3.2) Si Φ contiene un enunciado tipo γ ($\varphi = \forall x\psi$) etiquetado como *noUsado*:

$$etiqueta(\varphi) = usado\ ^{8}$$
$$\Phi = \Phi + \psi(c_1/x) + \ldots + \psi(c_n/x)$$

donde c_i con $1 \leq i \leq n$ son las constantes que ocurren en Φ.

a) Si Φ es abierta y saturada, terminar y regresar Φ.

b) Si Φ es cerrada, ir a (1).

c) En caso contrario, $etiqueta(\psi(c_i/x)) = noUsado$ para toda $1 \leq i \leq n$. Ir al paso (3.1).[9]

(3.3) Si Φ contiene un enunciado φ tipo β ($\varphi = \beta_1 \vee \beta_2$).

$$etiqueta(\varphi) = usado$$

[6] La constante c_{k+1} es nueva para Φ.

[7] Por lo menos hay una ya que Φ antes era abierta y la rama Φ_{k+1} se extendio agregando un enunciado instanciado con una variable nueva en Φ, por lo tanto, no puede cerrar la rama Φ_{k+1}.

[8] Dado que no exploraremos ramas con peso mayor a n, y ya se aplico la regla γ con cada una de las constantes que pueden ocurrir.

[9] Φ sigue abierta y tiene peso n.

Utilizando la β-regla extender el árbol obteniendo dos nuevas ramas, $\Phi_1 = \Phi + \beta_1$ y $\Phi_2 = \Phi + \beta_2$.

a) Si alguna de las ramas Φ_1 o Φ_2 es abierta y saturada, terminar y regresarla.

b) Si Φ_1 y Φ_2 son cerradas, ir al paso (1).

c) En otro caso, suponiendo que Φ_1 es la rama abierta actualiza $\Phi = \Phi_1$ e introducir la rama Φ_2 a *ABIERTAS* si ésta es abierta. Ir al paso (3.1).

(3.4) Si Φ contiene un enunciado φ tipo δ ($\varphi = \exists x\psi$).

$$etiqueta(\varphi) = usado$$

Utilizando la δ'-regla extender la rama Φ obteniendo n nuevas ramas, $\Phi_i = \Phi + \psi(c_i/x)$ para cada $1 \leq i \leq n$.[10]

a) Si alguna Φ_i ($1 \leq i \leq n$) es abierta y saturada, terminar y regresar Φ_i.

b) Si todas las ramas Φ_i (con $1 \leq i \leq n$) son cerradas, ir al paso (1).

c) En cualquier otro caso, actualiza Φ con la primer rama Φ_i que sea abierta e insertar todas las ramas abiertas a *ABIERTAS*. Ir al paso (3.1).

A continuación demostraremos la correctitud del algoritmo:

Lema 3.6 *El algoritmo constryeRamaAS*(Γ, n) *siempre termina.*

Demostración. Las dos razones por las cuales el algoritmo *construyeRamaAS*(Γ, n) podría no terminar son las siguientes:

1. agregar un número infinito de veces ramas a *ABIERTAS*. O bien,

2. procesar una rama infinita.

Observemos que al inicio del algoritmo *ABIERTAS* sólo contiene una rama, a saber la rama que contiene los enunciados de Γ. En el paso (1) eliminamos esta rama de *ABIERTAS* almacenandola en Φ, quedando *ABIERTAS* vacía. En los siguientes pasos se procesa la rama Φ, si *peso*(Φ) es menor a n (paso (2)), tenemos cuatro opciones:

- Si se ejecuta el paso (2.1) se etiqueta el nodo correspondiente como *usado* y en el peor de los casos (es decir, la rama no es cerrada ni saturada) se agregan dos nuevos nodos etiquetados a Φ, pero con menor complejidad estructural. Ninguna rama es agregada a *ABIERTAS*.

- Si se ejecuta el paso (2.2) el nodo no es marcado como *usado* a menos que todos los nodos no literales restantes estén marcados como *usados* lo cual significa que el peso ya no puede incrementar y ya se agregaron a la rama todas las α-subfórmula instanciadas con cada una de las constantes de la rama. El peor de los casos, surge cuando el nodo no es etiquetado como *usado* y se agregan nuevos nodos, pero éstos tienen complejidad menor. Ninguna rama es insertada a *ABIERTAS*.

[10] No tomamos en cuenta a la rama $\Phi + \varphi(c_{n+1}/x)$ ya que tiene peso mayor a n.

- Si se ejecuta el paso (2.3) un nodo es etiquetado como *usado*. Se crean dos nuevas ramas similares a Φ cada una de ellas con un nuevo nodo etiquetado como *noUsado* pero de menor complejidad. En el peor de los casos, únicamente una rama es insertada a *ABIERTAS*, pero como ya mencionamos ésta tiene en nodo de la rama originial etiquetado como *usado* y el nuevo nodo recién agregado que está etiquetado como *noUsado* tiene complejidad estructural menor al ahora etiquetado como *usado*.

- Si se ejecuta el paso (2.3) un nodo es etiquetado como *usado*. Este caso es parecido al anterior, en el peor de los casos, se agregan un número finito de ramas a *ABIERTAS*, a lo más n, cada una de ellas con un nodo de menor complejidad estructural etiquetado como *noUsado* mientras que uno de mayor complejidad fue etiquetado como *usado*.

Cuando $peso(\Phi)$ es igual a n, los pasos a seguir son similares a los anteriores, salvo cuando se aplica la regla γ (paso (3.2)) pues el nodo correspondiente siempre es etiquetado como *usado*.

De lo anterior, podemos ver que en cada paso se agregan nuevos nodos con complejidad menor, y se etiqueta un nodo con mayor complejidad que los agregados como *usado*. El único caso en el cual no etiquetamos un nodo de mayor complejidad como *usado* es cuando aplicamos una regla γ, pero como el peso máximo de las ramas está acotado por n, entonces, no hay forma de crear una rama infinita, ya que significaría que ésta tiene peso infinito (no hay fórmulas con complejidad infinita).

Por otro lado, el número de ramas que se agregan a *ABIERTAS* siempre es finito y con un nodo de mayor complejidad etiquetado como *usado*. Además de que una vez que una rama es eliminada de *ABIERTAS* nunca se vuelve a introducir. Por lo tanto, el algoritmo termina.

$\dashv$.

Lema 3.7 *Sea Γ un conjunto finito de enunciados minimalmente n-satisfactible cuyo peso es m. Entonces, el número de cuantificadores existenciales de Γ debe ser:*

1. *mayor o igual a $n - m$ si $m \geq 1$ ó si $m = 0$ y Γ no contiene ningún enunciado tipo γ, o*

2. *al menos $n - 1$, en cualquier otro caso.*

Demostración. Las únicas reglas que incrementan el peso de una rama son las reglas γ (siempre y cuando no aparezca ninguna constante en la rama) y δ'.

Luego entonces, por cada existencial podemos incrementamos en uno el peso de de una de las ramas garantizando que esta no es cerrada.

Si $peso(\Gamma) \geq 1$ o bien $peso(\Gamma) = 0$ y Γ no contiene ningún enunciado tipo γ, entonces la única regla que incrementa el peso de un tableau extendido de Γ es la δ'-regla. Por lo tanto, si el número de existenciales que aparecen en Γ es k con $k < n - m$, entonces la rama con mayor peso tiene $m + k$ constantes, y esto es menor que n. Si existiera una rama abierta sería posible constuir un modelo de cardinalidad menor a n por el Lema 3.4, contradiciendo la minimalidad de la n-satisfactibilidad. En caso contrario, cuando todas las ramas son cerradas contradecimos que Γ sea satisfactible.

En caso de que el $peso(\Gamma) = 0$ y Γ contenga un enunciado tipo γ, entonces primero aplicamos la γ-regla incrementando el peso de la rama. Por lo tanto, el peso de esta extensión es 1, lo cual nos lleva al caso anterior, pero el número de existenciales debe ser al menos $n-1$ existenciales. $\dashv$

Lema 3.8 (correctitud del algoritmo) *Sea* Γ *un conjunto finito de enunciados en forma normal prenex.* Γ *es minimalmente* m*-satisfactible, con* $1 \leq m \leq n$ *si y sólo si el algoritmo*
$construyeRamaAS(\Gamma, n)$ *encuentra una rama abierta y saturada de peso* m.

Demostración.
$\Longrightarrow$) Si Γ es un conjunto minimalmente m-satisfactible, entonces debe contener una rama Φ abierta y saturada de peso m por la propiedad fundamental de las DB-tableaux. Ya que $construyeRamaAS(\Gamma, n)$ examina todas las posibles extensiones de la rama inicial en orden ascendente de acuerdo al peso (la de peso menor está al inicio), en algún punto debe examinar a Φ, finalizando y regresándola. No puede regresar una rama de menor peso que m ya que contradice la hipótesis de que es minimalmente m-satisfactible y tampoco puede suceder que no encuentre ninguna rama de peso m abierta y saturada. Ya que las reglas son correctas y al menos una rama de peso m debe ser abierta.
$\Longleftarrow$) El algoritmo $construyeRamaAS(\Gamma, n)$ extiende las ramas en orden ascendente de acuerdo al peso ya que *ABIERTAS* es una cola de prioridades, así que si encuentra una rama de peso m abierta y saturada, es porque ya examinó todas las ramas de menor peso que m. Y por el Lema 3.4 es posible construir un modelo de peso m, demostrando con ello que Γ es minimalmente m-satisfactible. $\dashv$

Teorema 3.2 (Correctitud del algoritmo $buscaModelo(\Gamma, n)$**)** *Sea* Γ *un conjunto finito de enunciados en forma normal prenex.* Γ *es minimalmente* m*-satisfactible, con* $1 \leq m \leq n$ *si y sólo si el algoritmo* $buscaModelo(\Gamma, n)$ *encuentra un modelo de cardinalidad* m.

Demostración. El algoritmo explora las ramas en orden ascendente, ya que *ABIERTAS* es una cola de prioridades.
$\Longrightarrow$) Sea Γ un conjunto m-satisfactible. Entonces $peso(\Gamma)+numExistenciales(\Gamma)$ debe ser mayor o igual a n, por el Lema 3.7, por lo tanto, no se cumple la condición del paso 1 del algoritmo. Asi es que se ejecuta el algoritmo $construyeRamaAS(\Gamma, n)$ con los argumentos Γ y n. Y dado que el algoritmo $construyeRamaAS(\Gamma, n)$ es correcto es posible construir un modelo de cardinalidad menor o igual a n.
$\Longleftarrow$). En caso de que $buscaModelo(\Gamma, n)$ devuelva un modelo de cardinalidad m, entonces por la correctitud del algoritmo $construyeRamaAS(\Gamma, n)$ todas las ramas de menor peso que m son cerradas. Así que por la propiedad fundamental de las DB-tableaux; Γ es un conjunto minimalmente m-satisfactible.

$\dashv$

El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos lo desconocido. Para los valientes es la oportunidad.

Victor Hugo.

4

Abducción en lógica de primer orden

En este capítulo estudiaremos la abducción como inferencia lógica en conjuntos de enunciados de $\mathcal{LPO}$ que son satisfactibles en modelos finitos.

La abducción es una forma de razonamiento lógico que nos permite inferir hipótesis que expliquen un evento dado. Es decir, la abducción es un proceso lógico que nos permite explicar un hecho φ que no se puede deducir de la teoría Θ dada.

4.1. Problemas abductivos

Un "problema abductivo" surge cuando no es posible explicar un evento φ a partir de una teoría adoptada Θ, es decir, $\Theta \not\models \varphi$ y $\Theta \not\models \neg\varphi$ [MP93]. Por lo tanto, es necesario agregar nuevas premisas a Θ. Formalmente,

Definición 4.1 (problema abductivo) *Sea Θ un conjunto finito de enunciados y φ un enunciado en $\mathscr{L}$, tales que $\Theta \not\models \varphi$ y $\Theta \not\models \neg\varphi$. Por tanto, se requiere explicar a φ. Decimos entonces que Θ y φ forman un problema abductivo, denotado por $\langle\Theta, \varphi\rangle$.*

Una solución a un problema abductivo es cualquier enunciado α que junto con la teoría explique el hecho. Notemos que la solución a un problema abductivo no es única. El conjunto de todas las posibles soluciones se define formalmente como:

Definición 4.2 (conjunto solución) *Sea $\langle\Theta, \varphi\rangle$ un problema abductivo, definimos el conjunto solución de $\langle\Theta, \varphi\rangle$, en símbolos $Sol(\langle\Theta, \varphi\rangle)$, de la siguiente manera:*

$$Sol(\langle\Theta, \varphi\rangle) = \{\psi \in \mathscr{L} \mid \Theta \cup \{\psi\} \models \varphi\}$$

Notemos que el conjunto anterior contiene a todas las soluciones posibles, incluso a las "indeseables". Por ejemplo, las triviales[1] y las inconsistentes con la teoría. Es claro que

[1] Soluciones que contienen a φ.

tales soluciones no aportan información que permita explicar a φ, por ello únicamente adoptaremos como soluciones a aquellas que si lo hagan.

En [Al97] se distinguen cinco tipos de soluciones abductivas, las cuales se clasifican de acuerdo con las condiciones que verifican:

- Requerimiento de inferencia. Esta condición establece que la solución obtenida para explicar el hecho a partir de cierta teoría adoptada sea tal, que el hecho sea consecuencia lógica de la teoría y la solución dada. Este requerimiento es el mínimo requesito que debe cumplir cualquier solución.

- Consistente. Del hecho que cualquier contradicción implica lógicamente cualquier fórmula, todas las soluciones que sean inconsistentes por sí mismas o inconsistentes con la teoría deben ser abandonadas.

- Explicativa. La exigencia de que el hecho no sea consecuencia lógica de la teoría no excluye aquellas soluciones que sean triviales, es decir, aquellas que por sí solas explican el hecho. Es decir, si α es la explicación y φ el hecho, entonces, $\alpha \not\models \varphi$.

- Mínima. Una solución se considera mínima si es la menor explicación, es decir, provee la menor información requerida para explicar el hecho en conjunción con la teoría adoptada.

- Preferencial. Una solución es preferancial si es la mejor explicación de acuerdo con un orden de prioridad dado.

Nosotros únicamente nos enfocaremos en soluciones que cumplan las tres primeras condiciones, por lo cual defimos una solución a un problema abductivo de la siguiente manera:

Definición 4.3 (solución de un problema abductivo) *Sea $\langle\Theta, \varphi\rangle$ un problema abductivo y α un enunciado de $\mathscr{L}$; α es una solución o explicación de $\langle\Theta, \varphi\rangle$ si cumple con las siguientes condiciones:*

1. $\alpha \in Sol(\langle\Theta, \varphi\rangle)$, *es decir,* $\Theta \cup \{\alpha\} \models \varphi$.

2. $\Theta \cup \{\alpha\}$ *es consistente* ($\Theta \cup \{\alpha\} \not\models \bot$).

3. $\alpha \not\models \varphi$.

Observación 4.1 *Notemos que $\alpha \not\models \neg\varphi$ es consecuencia de las condiciones anteriores. Si $\alpha \models \neg\varphi$ entonces $\Theta \cup \{\alpha\} \models \neg\varphi$ y por 1 $\Theta \cup \{\alpha\} \models \varphi$, entonces $\Theta \cup \{\varphi\}$ es inconsistente contradiciendo con ello a 2. Por lo tanto, $\alpha \not\models \neg\varphi$.*

Definimos formalmente cuándo una solución es no redundante y cuándo es mínima.

Definición 4.4 (solución no redundante) *Dado un problema abductivo $\langle\Theta, \varphi\rangle$ y una solución α del mismo. Decimos que α es no redundante si ninguna subfórmula propia de α es una solución.*

Definición 4.5 (solución mínimal) *Dado un problema abductivo $\langle\Theta,\varphi\rangle$ y una solución α para éste. Decimos que α es minimal si no existe ninguna solución β de $\langle\Theta,\varphi\rangle$ tal que $\alpha \models \beta$ y $\beta \not\models \alpha$.*

4.2. Problemas n-abductivos

Una vez que se han definido formalmente un problema abductivo y una solución del mismo, es claro ver por qué ha sido tan poco explorado el tema de la abducción de primer orden. Tanto la definición del problema abductivo como la de la solución se enfrentan con el problema de indecidibilidad de $\mathcal{CPO}$. No obstante, como ya hemos mencionado anteriormente, este hecho no ha sido motivo para abandonar por completo el estudio de la relación de consecuencia lógica de enunciados de primer orden.

En esta sección estudiaremos la noción de problemas n-abductivos, los cuales son definidos sobre teorías que son satisfactibles en dominios finitos. Además, plantearemos problemas n-abductivos en términos de una modificación a las tablas semánticas.

En los trabajos [Ne02] y [AN03] se propone conectar la noción de un problema abductivo, relativa a dominios de cardinalidad finita con los DB-tableaux. En la siguiente sección presentamos un breve estudio de este planteamiento.

4.2.1. Problemas n-abductivos y DB-tableaux

Las siguientes definiciones son tomadas literalmente de los trabajos antes citados [[Ne02], [AN03]].

Definición 4.6 (problema n-abductivo)[2] *Dado un conjunto finito de enunciados Θ y un enunciado φ libre de cuantificadores en $\mathscr{L}$, para los cuales existe un DB-tableau, $\mathcal{T}_{DB}(\Theta \cup \{\neg\varphi\})$, con una rama abierta de peso n, decimos que $\langle\Theta,\varphi\rangle_n$ representa un problema n-abductivo.*

La solución a un problema n-abductivo se define a través de los DB-tableaux.

Definición 4.7 (solución n-abductiva) *Dado un problema n-abductivo $\langle\Theta,\varphi\rangle_n$, con $1 \leq n < \omega$, α es una solución n-abductiva si y sólo si*

1. *Existe un DB-tableau de $\Theta \cup \{\alpha\} \cup \{\neg\varphi\}$ tal que todas las ramas de peso m (con $1 \leq m \leq n$) son cerradas.*

2. *Existe un DB-tableau de $\Theta \cup \{\alpha\}$ con una rama abierta y acabada[3] de peso n.*

Observemos que tal definición depende en su totalidad de las DB-tableaux. Por lo tanto, si quisiéramos utilizar un método de búsqueda de explicación a problemas n-abductivos distinto a éste, las definiciones expuestas no podrían ser utilizadas.

[2] Versión Aliseda y Nepomuceno.

[3] La condición de que la rama sea acabada es equivalente a decir que la rama es saturada.

Por otra parte, la forma de los enunciados que forman la teoría y el hecho, se restringe de manera que sólo es posible tratar con problemas n-abductivos cuya solución sea una literal.

4.2.2. Problemas n-abductivos y n-consecuencia lógica

Inspirados en las definiciones presentadas en la sección anterior, proponemos una terminología para explorar el tema de abducción de primer orden en conjuntos de enunciados que son satisfactibles en modelos finitos, esta noción es independiente de un método de decisión.

A continuación planteamos la definición de problemas n-abductivos para lo cual nos serviremos de la noción de n-satisfactibilidad.

Definición 4.8 (problema n-abductivo) *Sea Θ un conjunto finito de enunciados de $\mathscr{L}$ minimalmente n-satisfactible ($1 \leq n < \omega$) y φ un enunciado de $\mathscr{L}$. Decimos que $\langle \Theta, \varphi \rangle$ es un problema n-abductivo, denotado por $\langle \Theta, \varphi \rangle_n$, si verifica:*

$$\Theta \not\models_n \varphi \;\; y \;\; \Theta \not\models_n \neg\varphi$$

En contraste con la Definición 4.6, la definición que acabamos de introducir no está sujeta a un procedimiento específico. Sin embargo, el siguiente resultado es válido.

Teorema 4.1 *Si $\langle \Theta, \varphi \rangle_n$ es un problema n-abductivo con $1 \leq n < \omega$, entonces existe un DB-tableau $\mathcal{T}_{DB}(\Theta \cup \{\neg\varphi\})$ correspondiente al problema tal que $\mathcal{T}_{DB}(\Theta \cup \{\neg\varphi\})$ contiene una rama saturada y abierta de peso n.*

Demostración. La demostración es directa de la definición de problema n-abductivo y del principio de refutación para la n-consecuencia lógica. ⊣

De manera similar, la solución a un problema n-abductivo se plantea en términos del concepto de n-consecuencia lógica.

Definición 4.9 (solución n-abductiva) *Sea $\langle \Theta, \varphi \rangle_n$ un problema n-abductivo y α un enunciado de $\mathscr{L}$; α es una solución n-abductiva de $\langle \Theta, \varphi \rangle_n$ si verifica:*

1. $\Theta \cup \{\alpha\} \models_n \varphi$.
2. $\Theta \cup \{\alpha\}$ *es minimalmente n-satisfactible* ($\Theta \cup \{\alpha\} \not\models_n \bot$).
3. $\alpha \not\models_n \varphi$.

Observación 4.2 *Notemos que $\alpha \not\models_n \neg\varphi$ es consecuencia lógica de 1 y 2 (verificar Observación 4.1).*

Esta definición no depende de ningún procedimiento de decisión y tampoco restringe la forma sintáctica que deben tener los enunciados que conforman un problema n-abductivo. El único prerrequisito que se debe cumplir es que la teoría Θ sea n-satisfactible. Lo cual es decidible para diversos fragmentos de enunciados de $\mathcal{LPO}$, ver [BGG01].

Una vez introducido el concepto de problema n-abductivo surge la siguiente pregunta: ¿existirá un procedimiento mecánico para encontrar soluciones a tales problemas?. Dado que en la historia de la abducción las tablas semánticas han sido de gran utilidad para encontrar soluciones a problemas abductivos; en la siguiente sección proponemos una modificación a los tableaux de Beth para conseguir un procedimiento efectivo para resolver problemas n-abductivos.

4.3. Una nueva extensión de los tableaux (N-tableaux)

En esta sección presentamos un refinamiento a las tablas semánticas de Beth para conjuntos de enunciados minimalmente n-satisfactibles. La modificación que proponemos está inspirada en los DB-tableaux que consiste, nuevamente, en una variación a la definición de la δ-regla de extensión.

Las interpretaciones que se toman en cuenta en el desarrollo de un N-tableau tienen como característica principal que cada elemento del dominio recibe un nombre único, es decir, existe una y sólo una constante para cada elemento del dominio. Nos referiremos a estas interpretaciones como *interpretaciones canónicas.*

Definición 4.10 (interpretaciones canónicas) *Sean $\mathscr{L}$ un lenguaje de primer orden y $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ una interpretación de $\mathscr{L}$. $\mathcal{M}$ es una interpretación canónica si para cualesquiera $c, c' \in Cons(\mathscr{L})$ y para cualquier $a \in \mathcal{D}$, si $\mathcal{I}(c) = \mathcal{I}(c') = a$ entonces $c = c'$. Es decir, si la función de interpretación $\mathcal{I}$ restringida a las constantes es inyectiva.*

Observemos que dado un conjunto de enunciados Γ de $\mathscr{L}$ minimalmente n-satisfactible con modelo $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ siempre es posible realizar una transformación de tal forma que obtengamos un conjunto de enunciados minimalmente n-satisfactible Γ^* de un lenguaje $\mathscr{L}^*$ y un $\mathscr{L}^*$-modelo canónico $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ tal que $\mathcal{M} \models \Gamma$ syss $\mathcal{M}^* \models \Gamma^*$. A esta transformación le denominamos n-estandarización.

Definición 4.11 (n-estandarización) *Sean Γ un conjunto de enunciados de $\mathscr{L}$ con m constantes $c_1, c_2, \ldots, c_m$ y $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ un $\mathscr{L}$-modelo de Γ, tal que $|D| = n$, $D = \{a_1, \ldots, a_n\}$. Una n-estandarización de $\langle \mathscr{L}, \Gamma, \mathcal{M} \rangle$ consta de un lenguaje $\mathscr{L}^*$, un conjunto de enunciados Γ^* del lenguaje $\mathscr{L}^*$ y un $\mathscr{L}^*$-modelo $\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ tales que:*

- *$\mathscr{L}^*$ tiene n constantes $d_1, \ldots, d_n$ donde $d_j \in N_j$ con $N_j = \{c_i \mid \mathcal{I}(c_i) = a_j\}$ si $N_j \neq \varnothing$, en caso contrario ($N_j = \varnothing$) d_j es una constante nueva.*
- *$\mathcal{M}^* = \langle \mathcal{D}^*, \mathcal{I}^* \rangle$ donde*

$$\begin{array}{lcl} D^* & = & D \\ \mathcal{I}^* \restriction_{Pred(\mathscr{L})} & = & \mathcal{I} \\ \mathcal{I}^*(d_j) & = & a_j \end{array}$$

- *$\Gamma^* = \{\varphi^* \mid \varphi \in \Gamma\}$ donde φ^* se define recursivamente de la siguiente manera:*
  - $P(t_1, \ldots, t_n)^* = P(t_1^*, \ldots, t_n^*)$.

- $(\neg\varphi)^* = \neg\varphi^*$
- $(\varphi \wedge \psi)^* = \varphi^* \wedge \psi^*$
- $(\varphi \vee \psi)^* = \varphi^* \vee \psi^*$
- $(\forall x\varphi)^* = \forall x\varphi^*$
- $(\exists x\varphi)^* = \exists x\varphi^*$

En el caso de los términos:

- $x^* = x$ *con* x *una variable.*
- $c^* = d_j$ *si* $\mathcal{I}(c) = a_j$.

Teorema 4.2 *Sea* φ *un enunciado de* $\mathscr{L}$, $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ *una* $\mathscr{L}$*-interpretación y* $\langle \mathscr{L}^*, \varphi^*, \mathcal{M}^* \rangle$ *una* n*-estandarización de* $\langle \mathscr{L}, \varphi, \mathcal{M} \rangle$. *Entonces,* $\mathcal{M} \models \varphi$ *syss* $\mathcal{M}^* \models \varphi^*$.

Demostración. Inducción sobre φ. ⊣

El teorema anterior nos permite suponer de ahora en adelante que todos los modelos son canónicos. Además, al considerar un modelo de tamaño n podemos suponer que el lenguaje subyacente tiene n constantes, las cuales denotan a cada uno de los elementos del dominio. Por lo tanto, de aquí en adelante cada vez que hablemos de un modelo supondremos que se trata de un modelo canónico y que el número de constantes del lenguaje es igual a la cardinalidad del dominio.

Definición 4.12 (N-tableau) *Sea* Γ *un conjunto finito de enunciados de* $\mathscr{L}$ *minimalmente* n*-satisfactible, con* n *conocida y* $1 \leq n < \omega$. *Un* N*-tableau de* Γ, *denotado por* $\mathcal{T}_N(\Gamma)$, *es un tableau construido por medio de las* α, β *y* γ *reglas definidas anteriormente y una nueva regla* δ_N, *la cual opera de la siguiente forma:*

$$\frac{\exists x\varphi}{\varphi(c_1/x) | \ldots | \varphi(c_n/x)}$$

donde c_i $(1 \leq i \leq n)$ *son las* n *constantes del lenguaje* $\mathscr{L}$. *A los enunciados* $\varphi(c_i/x)$ $(1 \leq i \leq n)$ *se les denomina* δ_N*-subfórmulas de* $\exists x\varphi$.

Observación 4.3 *Es trivial verificar que esta nueva regla de extensión es correcta ya que por hipótesis tenemos que el conjunto de enunciados* Γ *es minimalmente* n*-satisfactible. Así,* Γ *tiene un modelo canónico* $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ *de cardinalidad* n, *por lo tanto,* $\mathcal{M} \models \varphi(c_i/x)$ *para alguna* $1 \leq i \leq n$, *pues* $\mathcal{M}$ *interpreta* n *constantes y cada una de ellas nombra a uno y sólo uno de los objetos del universo* $\mathcal{D}$.

Las definiciones correspondientes a una *rama abierta, cerrada* y *saturada* son exactamente las mismas que en los DB-tableaux.

Observación 4.4 *Notemos que cualquier rama* Φ *abierta y saturada de un* N*-tableau de* Γ, $\mathcal{T}_N(\Gamma)$, *forma un conjunto de Hintikka.*

En el caso de los N-tableaux sí tiene sentido hablar de un N-tableau saturado, ya que a diferencia de los DB-tableaux los N-tableaux no tienen ramas infinitas. La razón de ello es que la δ_N-regla de extensión siempre abre el mismo número de ramas, pues el número de constantes es fijo desde el inicio de la construcción. Así, no nos enfrentamos con el problema de introducir nuevas constantes cada vez que un enunciado tipo δ es utilizado para extender el N-tableau, provocando con ello que tengamos que instanciar aquellos enunciados tipo γ que aparezcan en la rama con la nueva constante, pudiendo caer en un proceso infinito.

Definición 4.13 (N-tableau saturado) *Un N-tableau de un conjunto finito de enunciados Γ minimalmente n-satisfactible de $\mathscr{L}$ es saturado si todas sus ramas son saturadas (abiertas o cerradas).*

4.3.1. Correctitud de las N-tableaux

Al igual que los DB-tableaux los N-tableaux también pueden ser utilizados para encontrar modelos para conjuntos de enunciados minimalmente n-satisfactibles. Aunque en el caso particular de los N-tableaux es prerrequisito conocer el valor de n, ya que es una condición necesaria para aplicar la δ_N-regla de extensión.

Lema 4.1 *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$ minimalmente n-satisfactible. Si $\mathcal{T}_N(\Gamma)$ es un N-tableau saturado de Γ con una rama saturada y abierta Φ de peso n. Entonces, es posible construir un modelo de Γ de cardinalidad n a partir de Φ.*

Demostración. Sea Φ una rama abierta y saturada de peso n de $\mathcal{T}_N(\Gamma)$. Φ forma un conjunto de Hintikka y por el Lema 3.2 podemos asegurar que Φ tiene un modelo de Herbrand $\mathcal{H} = \langle \mathcal{D}_\Phi, \mathcal{I}_\mathcal{H} \rangle$ de cardinalidad n, es decir, $\mathcal{H} \models_n \varphi_i$ para todo φ_i enunciado de Φ, en particular para aquellos que están en Γ. Por lo tanto, $\mathcal{H}$ es un modelo de Γ.

$\dashv$

Lema 4.2 *Sea Γ un conjunto finito de enunciados de $\mathscr{L}$. Si Γ es minimalmente n-satisfactible, entonces cualquier $\mathcal{T}_N(\Gamma)$ saturado cumple con las siguientes condiciones:*

1. *Todas las ramas de peso m, con $1 \leq m < n$, son cerradas.*

2. *$\mathcal{T}_N(\Gamma)$ tiene una rama abierta de peso n.*

Demostración. Sea Γ un conjunto finito de enunciados minimalmente n-satisfactible. Sea $\mathcal{T}_N(\Gamma)$ un N-tableau saturado. Supongamos que $\mathcal{T}_N(\Gamma)$ contiene una rama abierta y saturada de peso m ($1 \leq m < n$). Por el Lema 4.1, es posible constuir un modelo de cardinalidad m, contradiciendo con ello que Γ es minimalmente n-satisfactible. Por lo tanto, se cumple la primera condición, todas las ramas de peso menor a n son cerradas.

Ahora supongamos que $\mathcal{T}_N(\Gamma)$ no contiene ninguna rama de peso n abierta. Por hipótesis sabemos que existe una estructura canónica $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ tal que $|\mathcal{D}| = n$ y $\mathcal{M} \models_n \Gamma$, dado que el resultado de aplicar cualquiera de las reglas de extensión de los N-tableaux a un conjunto de enunciados satisfactibles, provoca que al menos una de las ramas permanezca abierta[4] agregándose nuevos enunciados que son satisfactibles por $\mathcal{M}$. Por lo tanto, no puede ocurrir que todas las ramas sean cerradas,[5] así que al menos una rama de peso n debe estar abierta. $\dashv$

Teorema 4.3 *Un conjunto finito de enunciados Γ es minimalmente n-satisfactible si y sólo si existe un $\mathcal{T}_N(\Gamma)$ saturado con una rama abierta y saturada de peso n y todas las ramas de peso menor a n son cerradas.*

Demostración. La demostración de este teorema es inmediata del Lema 4.2 y la observación 4.4.

4.4. Problemas n-abductivos y N-tableaux

En esta sección caracterizaremos las N-tableaux como un procedimiento efectivo para buscar soluciones a problemas n-abductivos.

Teorema 4.4 *Si $\langle \Theta, \varphi \rangle_n$ es un problema n-abductivo con $1 \leq n < \omega$, entonces existe un N-tableau, $\mathcal{T}_N(\Theta \cup \{\neg\varphi\})$, saturado que contiene una rama abierta de peso n.*

Demostración. Por definición de $\langle \Theta, \varphi \rangle_n$ tenemos que $\Theta \not\models_n \varphi$ y que Θ es un conjunto minimalmente n-satisfactible, por lo tanto, por el principio de refutación tenemos que $\Theta \cup \{\neg\varphi\}$ es un conjunto minimalmente n-satisfactible, así que por el Teorema 4.3, cualquier N-tableau, $\mathcal{T}_N(\Theta \cup \{\neg\varphi\})$, saturado tiene una rama abierta de peso n. $\dashv$

Teorema 4.5 *Sea $\langle \Theta, \varphi \rangle_n$ un problema n-abductivo y α un enunciado en $\mathscr{L}$, α es una solución n-abductiva de $\langle \Theta, \varphi \rangle_n$ si y sólo si se cumplen las siguientes condiciones:*

1. *Cualquier N-tableau saturado de $\Theta \cup \{\neg\varphi\} \cup \{\alpha\}$ es cerrado, es decir, todas sus ramas son cerradas.*

2. *Cualquier N-tableau saturado de $\Theta \cup \{\alpha\}$ tiene al menos una rama de peso n abierta y cualquier rama de peso m, con $1 \leq m < n$, es cerrada.*

3. *Cualquier N-tableau saturado de $\{\alpha\} \cup \{\neg\varphi\}$ tiene al menos una rama abierta de peso m, con $1 \leq m \leq n$, y todas las ramas de menor peso son cerradas.*

Demostración.

1. Por definición de solución n-abductiva tenemos que $\Theta \cup \{\alpha\} \models_n \varphi$ esto ocurre syss el conjunto $\Theta \cup \{\alpha\} \cup \{\neg\varphi\}$ no es minimalmente n-satisfactible. Por el Teorema 4.3, esto se cumple syss cualquier $\mathcal{T}_N(\Theta \cup \{\neg\varphi\} \cup \{\alpha\})$ saturado es cerrado.

2. Por definición, $\Theta \cup \{\alpha\}$ es minimalmente n-satisfactible, nuevamente esto ocurre syss cualquier $\mathcal{T}_N(\Theta \cup \{\alpha\})$ saturado contiene una rama de peso n abierta y todas las ramas de menor peso son cerradas, Teorema 4.3.

[4]Consecuencia de la correctitud de las reglas de extensión.

[5]Notemos que ninguna rama tiene peso mayor a n pues el número máximo de constante que introdujimos es fijo (n).

3. Por definición de solución n-abductiva tenemos que $\alpha \not\models_n \varphi$, esto ocurre syss $\{\alpha\} \cup \{\neg\varphi\}$ es n-satisfactible. Sea m con $1 \leq m \leq n$ tal que $\{\alpha\} \cup \{\neg\varphi\}$ es minimalmente m-satisfactible, el Teorema 2.1 garantiza que tal m existe. Por lo tanto, todo N-tableau saturado de $\{\alpha\} \cup \{\neg\varphi\}$ contiene una rama abierta de peso m.

   Ahora bien, sea $\mathcal{T}_N(\{\alpha\} \cup \{\neg\varphi\})$ un N-tableau saturado con una rama abierta de peso m, con m mínima, entonces $\{\alpha\} \cup \{\neg\varphi\}$ es un conjunto minimalmente m-satisfactible, por el Teorema 4.3. Dado que n es mayor que m, entonces $\{\alpha\} \cup \{\neg\varphi\}$ es un conjunto n-satisfactible, de donde es consecuencia que $\alpha \not\models_n \varphi$.

$\dashv$

Como ya mencionamos anteriormente, las N-tableaux son un primer prototipo de procedimiento efectivo para resolver un problema n-abductivo $\langle\Theta, \varphi\rangle_n$. El proceso consiste en construir un N-tableau saturado de $\Theta \cup \{\varphi\}$ y tratar de cerrarlo, para lo cual es necesario obtener el conjunto cerradura de cada una de las ramas abiertas.

Definición 4.14 (conjunto cerradura) *Sea Φ una rama de un N-tableau o de un DB-tableau, y $Lit(\Phi)$ el conjunto de literales que aparecen en Φ. El conjunto cerradura de Φ, denotado por $CC(\Phi)$, se define como el conjunto formado por cada una de las literales complementaria que constituyen $Lit(\Phi)$.*

Una vez definido formalmente el conjunto cerradura, un proceso para encontrar soluciones a un problema n-abductivo $\langle\Theta, \varphi\rangle_n$ a partir de un N-tableau saturado de $\Theta \cup \{\neg\varphi\}$ cuyas ramas abiertas son $\Phi_1, \ldots, \Phi_k$, es el siguiente:

1. Para cada rama Φ_i, con $1 \leq i \leq k$, se obtiene $CC(\Phi_i)$.

2. Si la intersección de todos los $CC(\Phi_k)$, con $1 \leq i \leq k$, es distinta del vacío, entonces, verificar si alguna de las literales es una solución n-abductiva. En tal caso, terminar.

3. En caso de no encontar soluciones literales, verificar si existen soluciones conjuntivas, las cuales se forman a patir de la conjunción de literales pertenecientes a los conjuntos cerradura. Verificar si tal conjunción es una solución n-abductiva.

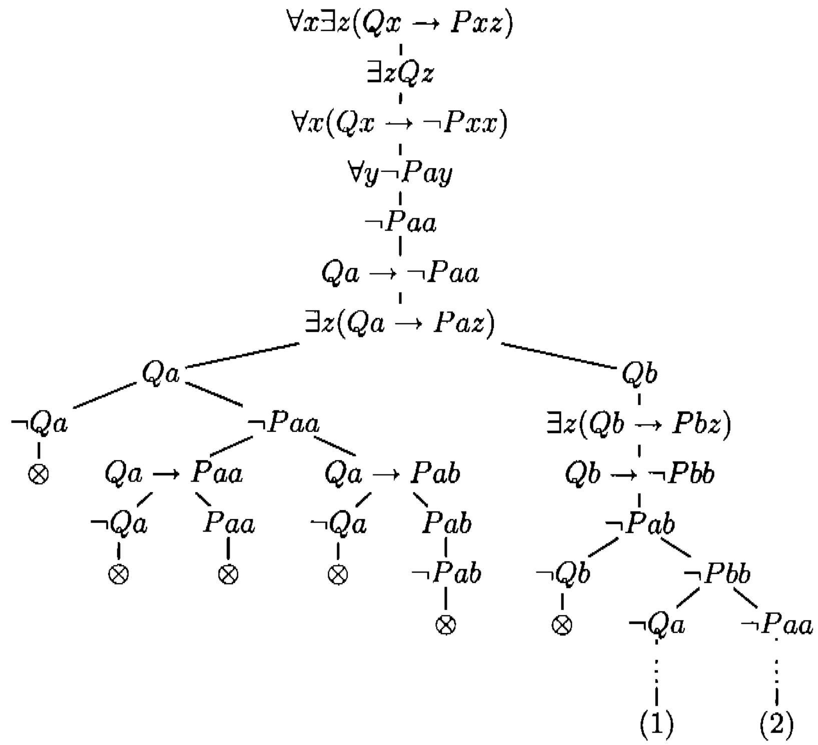
Observemos el siguiente hecho, si alguna solución es una conjunción de la forma:

$$\varphi(c_1/x) \wedge \varphi(c_2/x) \wedge \ldots \wedge \varphi(c_n/x)$$

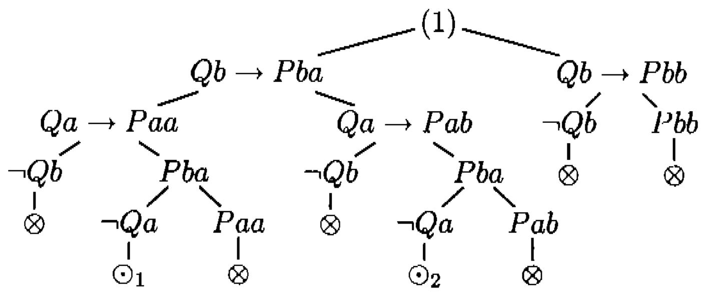
Entonces la solución n-abductiva es equivalente con el enunciado $\forall x\varphi$.

Ejemplo 4.1 *Sea $\Theta = \{\forall x \exists z(Qx \rightarrow Pxz),\ \exists zQz,\ \forall x(Qx \rightarrow \neg Pxx)\}$ una teoría minimalmente 2-satisfactible y el hecho $\varphi = \exists yPay$. Encontrar una solución para el problema 2-abductivo, $\langle\Theta, \varphi\rangle_2$.*

Solución. Construyamos el N-tableau correspondiente a $\Theta \cup \{\neg\varphi\}$.



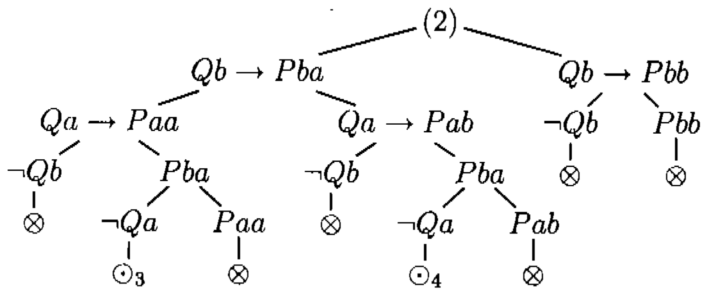
Antes de continuar con la construcción del N-tableau, queremos hacer notar que todas las ramas de peso 1 se cerraron aplicando únicamente las reglas de extensión a enunciados de Θ. Extenderemos el árbol a profundidad de izquierda a derecha. Además, utilizaremos los símbolos $\otimes$ y $\odot$ para representar que una rama ha sido cerrada o, saturada y abierta, respectivamente.



De la extensión de la rama (1) obtenemos dos ramas abiertas y saturadas ($\odot_1$ y $\odot_2$), y el conjunto de literales de cada una de ellas es el mismo. A saber,

$$\{\neg Paa, Qb, \neg Pab, \neg Pbb, \neg Qa, Pba\}$$

Ahora bien, el resultado de extender la rama (2) del N-tableau inicial, es el siguiente:



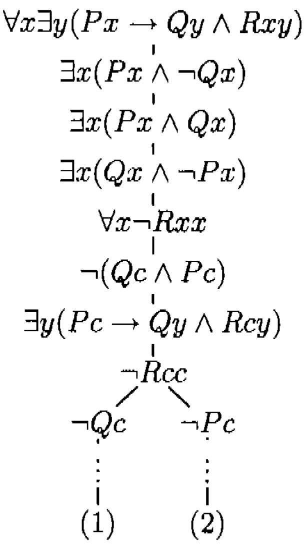
De donde obtenemos nuevamente dos ramas abiertas y saturadas ($\odot_3$ y $\odot_4$) , tales que el conjunto de literales es exactamente el mismo que en las ramas abiertas y saturadas obtenidas de (1). Por lo tanto, el conjunto de cierre de cada una de ellas es el mismo, a saber:

$$\{Paa, \neg Qb, Pab, Pbb, Qa, \neg Pba\}$$

Ahora sólo nos falta verificar cuáles de ellas son soluciones al problema $\langle \Theta, \varphi \rangle_n$. Ni Pab ni Paa pueden ser explicaciones ya que son soluciones triviales. Pero cualquiera de las literales $\neg Qb$ y Qa son soluciones a $\langle \Theta, \varphi \rangle_n$. Además, por el hecho de que tenemos Pba y Pbb en el conjunto cerradura y el problema es 2-abductivo, entonces también podemos construir la solución $\forall x Pbx$.

Ejemplo 4.2 *Dada la teoría* $\Theta = \{\forall x \exists y (Px \to Qy \wedge Rxy),\ \exists x(Qx \wedge \neg Px),\ \exists x(Px \wedge Qx),\ \exists x(Px \wedge \neg Qx)\ \forall x \neg Rxx\}$, *la cual es minimalmente 3-satisfactible, y el hecho* $\varphi = Qc \wedge Pc$. *Encuentra una solución para el problema 3-abductivo* $\langle \Theta, \varphi \rangle_3$.

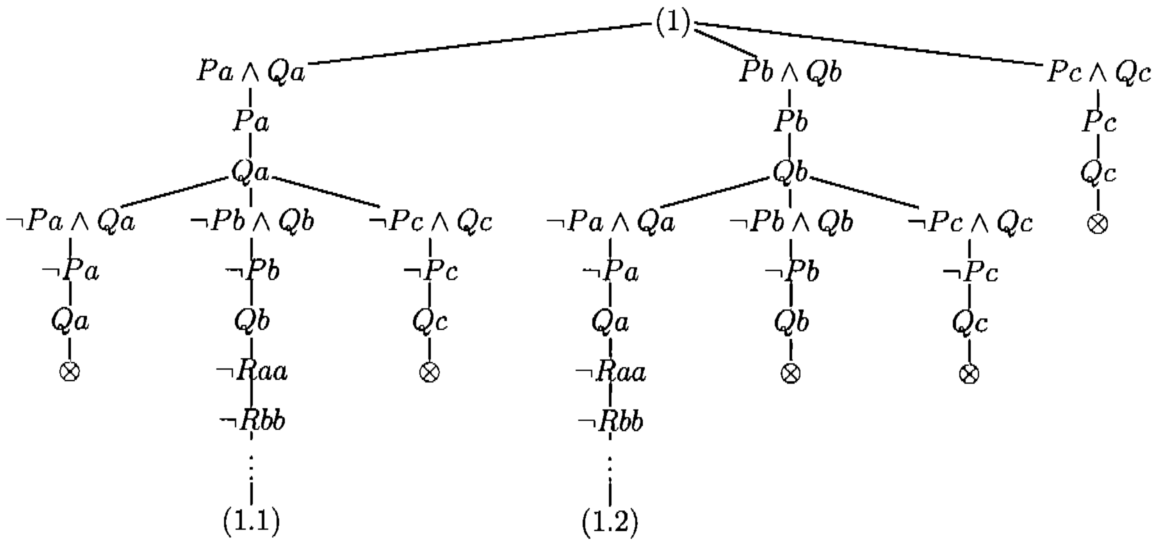
Solución. La parte inicial del N-tableau saturado de $\Theta \cup \{\neg\varphi\}$ es la siguiente:



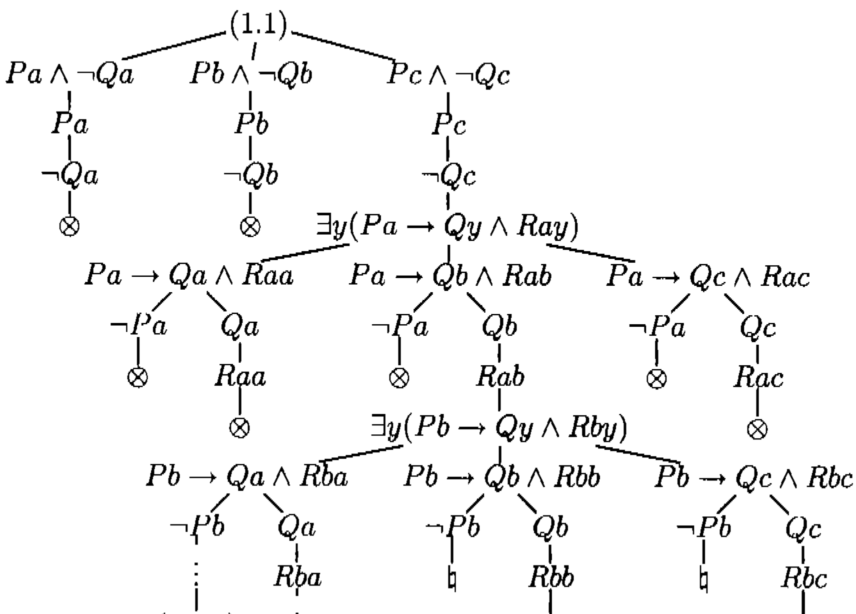
Observación 4.5 *Notemos que cualquier extensión que realicemos sobre el N-tableau debe ser utilizando la δ_N-regla. Esto implica que deberemos abrir tres ramas, una por cada elemento del dominio. En cambio, con los DB-tableau el resultado de aplicar la δ'-regla abre sólo dos ramas: una con la constante c y otra con una nueva constante. Lo*

anterior provoca la pérdida de información en la búsqueda de una solución a problemas n-abductivos si el procedimiento se realiza con DB-tableaux y por eso hemos propuesto los N-tableaux.

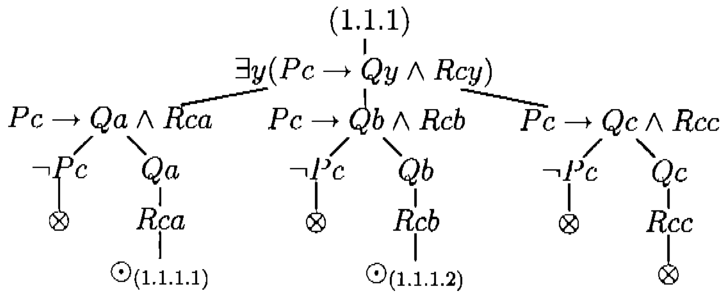
Advirtiendo que de aquí en adelante podaremos aquellas ramas cuyos enunciados que las conforman sean los mismos, unicamente escribiremos una de ellas. El resultado es claramente el mismo; para resaltar aquellas ramas que podaremos utilizaremos el símbolo ♮. Prosiguiendo con la extensión de la rama (1) obtenemos:



Primero extenderemos la rama abierta **(1.1)**.

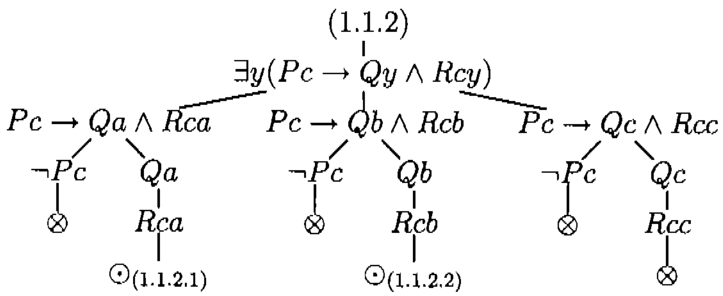


Extendiendo las dos ramas que aún continúan abiertas, obtenemos:



Por lo tanto, de la rama (1.1.1) obtenemos dos ramas abiertas y saturadas: $\odot_{(1.1.1.1)}$ y $\odot_{(1.1.1.2)}$, las cuales tienen los siguientes conjuntos de literales:

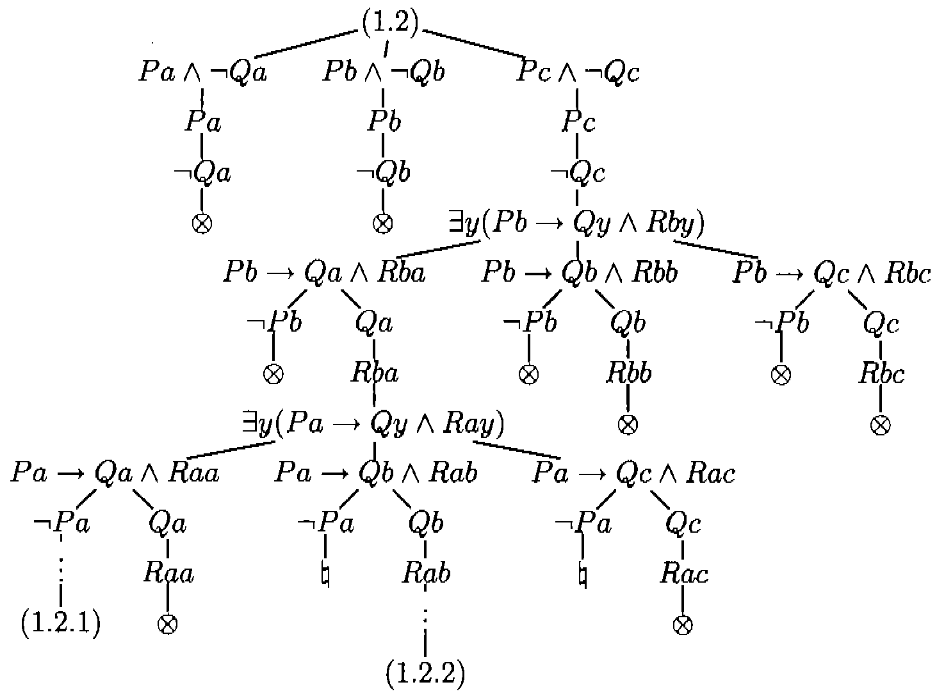
$$\begin{array}{rcl} Lit(\odot_{(1.1.1.1)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rca,\ Rab\} \\ Lit(\odot_{(1.1.1.2)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rcb,\ Rab\} \end{array}$$



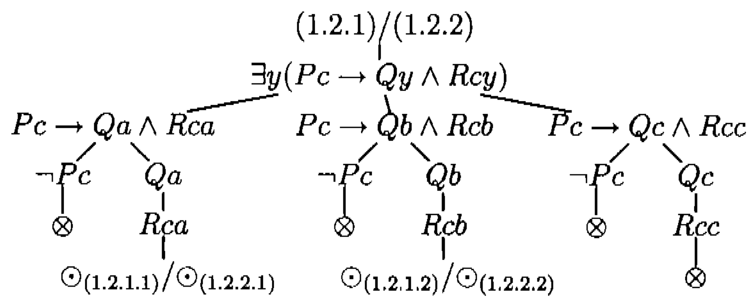
La extensión de (1.1.2) resultó exactamente igual, con la única diferencia que en todas las subramas de (1.1.2) aparece la literal *Rba*. El conjunto de literales de las ramas abiertas y saturadas $\odot_{(1.1.2.1)}$ y $\odot_{(1.1.2.2)}$ son:

$$\begin{array}{rcl} Lit(\odot_{(1.1.2.1)}) & = & \{\neg Qc\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rba,\ Rca,\ Rab\} \\ Lit(\odot_{(1.1.2.2)}) & = & \{\neg Qc,\ Pa,\ Qa,\ \neg Pb,\ Qb,\ Pc,\ Rba,\ Rcb,\ Rab\} \end{array}$$

Ahora regresemos a extender la rama abierta **(1.2)**.



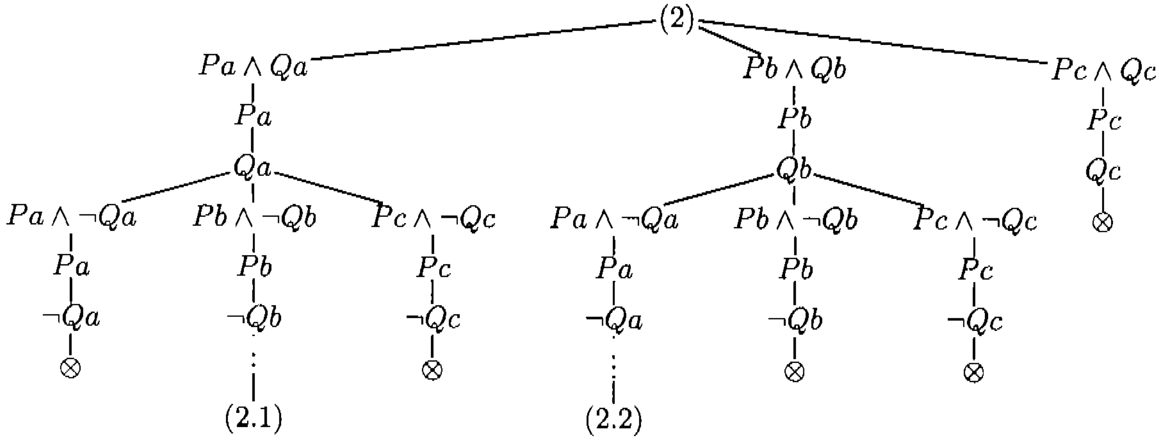
Dado que únicamente nos falta extender el γ-enunciado $\forall x \exists y(Px \rightarrow Qy \wedge Rxy)$ en las ramas (1.2.1) y (1.2.2), el resto de la extensión de tales ramas es exactamente igual. Por lo tanto, únicamente lo realizaremos una vez.



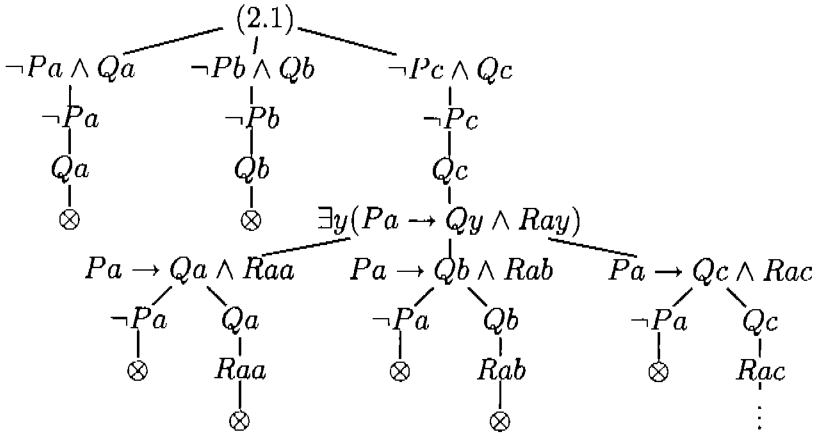
De lo anterior, concluimos que la rama **(1.2)** tiene cuatro ramas abiertas y saturadas $\odot_{(1.2.1.1)}$, $\odot_{(1.2.1.2)}$, $\odot_{(1.2.2.1)}$ y $\odot_{(1.2.2.2)}$. Cuyos conjuntos de literales son los siguientes:

$$
\begin{array}{rcl}
Lit(\odot_{(1.2.1.1)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rca\} \\
Lit(\odot_{(1.2.1.2)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rcb\} \\
Lit(\odot_{(1.2.2.1)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rab,\ Rca\} \\
Lit(\odot_{(1.2.2.2)}) & = & \{\neg Qc,\ Pb,\ Qb,\ \neg Pa,\ Qa,\ Pc,\ Rab,\ Rcb\}
\end{array}
$$

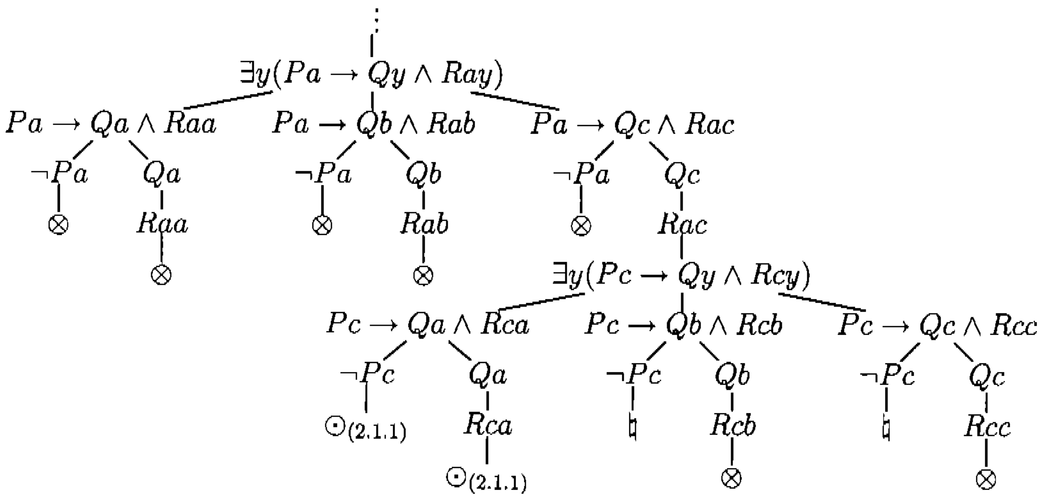
Una vez completa la extensión de la rama **(1)**, seguimos con la extensión que dejamos pendiente de la rama **(2)**.



Extendiendo la rama **(2.1)**.



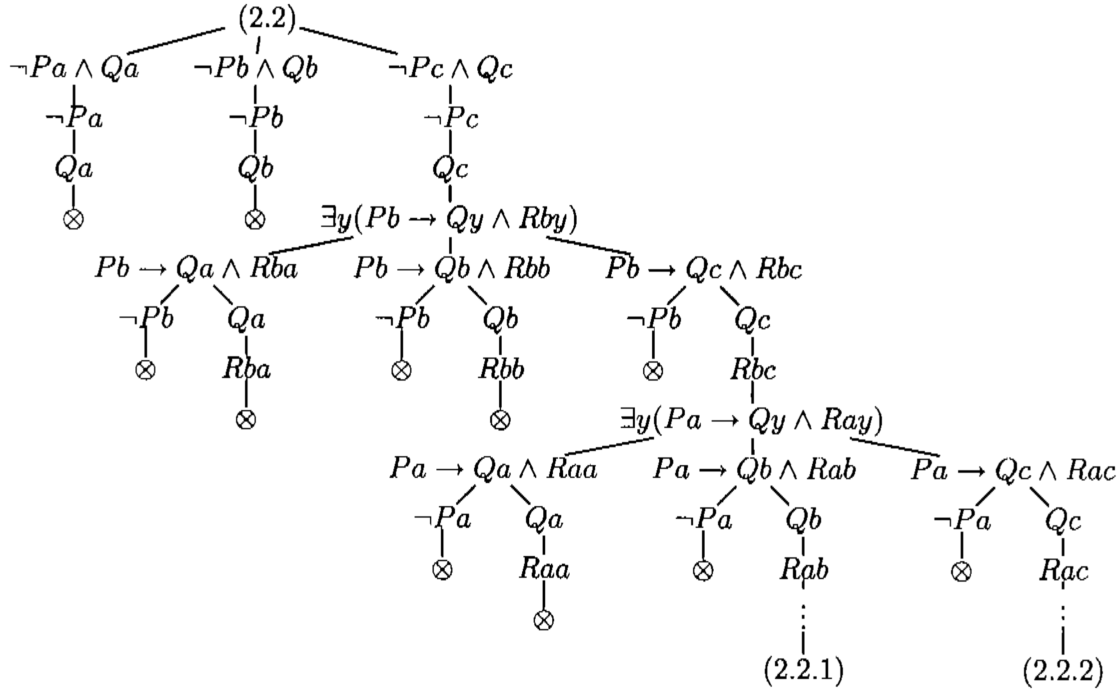
Continuamos con la extensión de la única rama que permanece abierta de **(2.1)**.



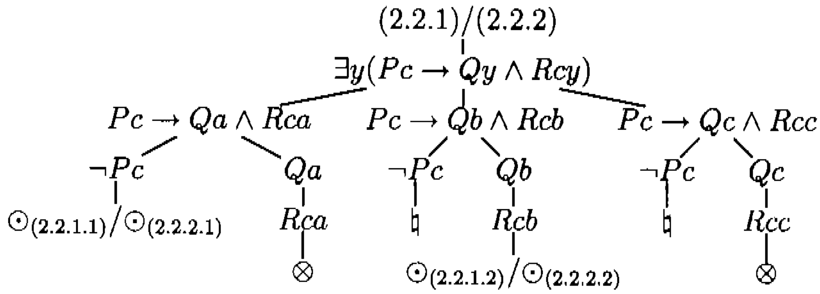
Por lo tanto, las únicas ramas abiertas y saturadas de la rama **(2.1)** son $\odot_{(2.1.1)}$ y $\odot_{(2.1.2)}$ y los conjuntos de literales son:

$$\begin{aligned} Lit(\odot_{(2.1.1)}) &= \{\neg Pc,\ Pa,\ Qa,\ Pb,\ \neg Qb,\ Qc,\ Rac\} \\ Lit(\odot_{(2.1.2)}) &= \{\neg Pc,\ Pa,\ Qa,\ Pb,\ \neg Qb,\ Qc,\ Rac,\ Rca\} \end{aligned}$$

Por último, extendemos la rama **(2.2)**.



Para las dos ramas que permanecen abiertas $\odot_{(2.2.1)}$ y $\odot_{(2.2.2)}$, únicamente falta aplicar la γ-regla al enunciado $\forall x \exists y (Px \rightarrow Qy \wedge Rxy)$ utilizando la constante c. Así, la extensión de ambas ramas utilizando la γ-regla es idéntica para ambas ramas.



Por lo tanto, las literales de las subramas abiertas y saturadas de la rama **(2.2)** son:

$$\begin{aligned} Lit(\odot_{(2.2.1.1)}) &= \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rab,\} \\ Lit(\odot_{(2.2.1.2)}) &= \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rab,\ Rcb\} \\ Lit(\odot_{(2.2.2.1)}) &= \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rac,\ \} \\ Lit(\odot_{(2.2.2.2)}) &= \{\neg Pc,\ Pb,\ Qb,\ Pa,\ \neg Qa,\ Qc,\ Rbc,\ Rcb,\ \} \end{aligned}$$

Una vez que hemos dado un N-tableau saturado para el problema 3-abductivo $\langle\Theta,\varphi\rangle_3$ planteado en el Ejemplo 2. Únicamente nos falta encontrar una solución a partir de los conjuntos de cierre de las ramas abiertas y saturadas. Los listamos a continuación:

$$\begin{array}{lcl}
CC(\odot_{(1.1.1.1)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rca,\ \neg Rab\} \\
CC(\odot_{(1.1.1.2)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rcb,\ \neg Rab\} \\
CC(\odot_{(1.1.2.1)}) & = & \{Qc\, \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rba,\ \neg Rca,\ \neg Rab\} \\
CC(\odot_{(1.1.2.2)}) & = & \{Qc,\ \neg Pa,\ \neg Qa,\ Pb,\ \neg Qb,\ \neg Pc,\ \neg Rba,\ \neg Rcb,\ \neg Rab\} \\
CC(\odot_{(1.2.1.1)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rca\} \\
CC(\odot_{(1.2.1.2)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rcb\} \\
CC(\odot_{(1.2.2.1)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rab,\ \neg Rca\} \\
CC(\odot_{(1.2.2.2)}) & = & \{Qc,\ \neg Pb,\ \neg Qb,\ Pa,\ \neg Qa,\ \neg Pc,\ \neg Rab,\ \neg Rcb\} \\
CC(\odot_{(2.1.1)}) & = & \{Pc,\ \neg Pa,\ \neg Qa,\ \neg Pb,\ Qb,\ \neg Qc,\ Rac\} \\
CC(\odot_{(2.1.2)}) & = & \{Pc,\ \neg Pa,\ \neg Qa,\ \neg Pb,\ Qb,\ \neg Qc,\ \neg Rac\, \neg Rca\} \\
CC(\odot_{(2.2.1.1)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rab\} \\
CC(\odot_{(2.2.1.2)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rab,\ \neg Rcb\} \\
CC(\odot_{(2.2.2.1)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rac\ \} \\
CC(\odot_{(2.2.2.2)}) & = & \{Pc,\ \neg Pb,\ \neg Qb,\ \neg Pa,\ Qa,\ \neg Qc,\ \neg Rbc,\ \neg Rcb\ \}
\end{array}$$

La intersección de estos conjuntos es vacía. De lo que concluimos que no hay soluciones que sean literales. Es necesario constuir explicaciones conjuntivas.

Utilizando el procedimiento antes propuesto, es posible construir las siguientes explicaciones:

$$\neg Qb \wedge \neg Pa$$
$$\neg Qa \wedge \neg Pb$$

Cabe destacar que podemos encontrar muchas más, pero éstas tienen la propiedad de ser minimales.

Un inconveniente que presenta este método de búsqueda de explicaciones es que como prerrequisito es necesario conocer la cardinalidad del modelo de Herbrand que satisface la teoría Θ. Además, en términos computacionales, tal procedimiento es muy costoso. El primer problema puede ser solucionado fácilmente al existir programas que construyen de manera autómatica modelos finitos para conjuntos de enunciados de $\mathcal{LPO}$. En cuanto al segundo problema, se puede tratar de desarrollar heurísticas de búsqueda, de tal forma que se restrinja la extensión completa del árbol.

Las dificultados aumentan conforme se aproxima uno al fin.

Goethe.

5 Conclusiones

El problema de abducción en la lógica de primer orden podría considerarse como imposible de tratar. La razón principal de tal aseveración se encuentra en la definición misma del problema abductivo $\langle\Theta, \varphi\rangle$, la cual requiere que $\Theta \cup \{\neg\varphi\}$ sea satisfactible o equivalentemente que $\Theta \not\models \varphi$, relación indecidible en $\mathcal{LPO}$. Sin embargo, la indecidibilidad de la lógica de primer orden no ha sido un obstáculo para estudiar el problema de consecuencia lógica desde el punto de vista algorítmico, con lo cual se han descubierto diversos fragmentos de $\mathcal{LPO}$ que tienen un problema de decisión de consecuencia lógica decidible (véase [BGG01]). Por otra parte los avances en velocidad y capacidad de cálculo que proporcionan las computadoras de nuestros días han llevado a un florecimiento del razonamiento automático mediante la implementación de diversos asistentes basados no sólo en lógica de primer orden sino en lógica de orden superior (OTTER e ISABELLE por mencionar un par de ejemplos).

De esta manera, en nuestra opinión, el problema de la indecidibilidad no debe ser un impedimento para plantear e intentar resolver problemas abductivos en $\mathcal{LPO}$. Aún más, sabiendo que existen diversos fragmentos decidibles de $\mathcal{LPO}$ que tienen la propiedad de modelo finito, es decir, el hecho de que tengan un modelo implica que tienen un modelo finito. Una breve descripción de tales fragmentos se presenta en la siguiente sección.

Por todo lo anterior y basados en [Ne02] y [AN03] enunciamos el problema de abducción en lógica de primer orden de manera que tenga sentido en aquellos fragmentos decidibles de $\mathcal{LPO}$ con la propiedad del modelo finito. Para realizar este planteamiento hemos propuesto un refinamiento a las nociones tradicionales de satisfacción y consecuencia lógica, a las cuales denominamos n-satisfacción y n-consecuencia lógica, ya que son relativas a dominios con cierta cardinalidad finita n.

Enfocándonos en conjuntos de enunciados de $\mathcal{LPO}$ satisfactibles en modelos finitos, utilizamos el método de refutación conocido como DB-tableaux, introducido de manera independiente en [Bo84] y [Di93], como una herramienta mecánica para decidir si un conjunto de enunciados es n-satisfactible para $n \geq 1$. Con este objetivo propusimos el algoritmo *buscaModelo*(Γ, n).

Además demostramos algunas propiedades interesantes de la relación existente entre la noción de n-satisfactibilidad y los DB-tableaux. El resultado más destacado es la propiedad fundamental de los DB-tableaux: *"Un conjunto de enunciados Γ de $\mathscr{L}$ es minimalmente n-satisfactible syss existe un $\mathcal{T}_{DB}(\Gamma)$ con una rama abierta de peso n y cualquier rama de peso menor es cerrada"*. Para la demostración de este teorema hemos contado con la colaboración del autor de [Ne02]. Este artículo contiene un resultado análogo al nuestro, aunque con ciertas diferencias esenciales como el hecho de que nosotros usamos modelos de cardinalidad mínima.

Con respecto a la abducción, caracterizamos este tipo de razonamiento en conjuntos de enunciados de $\mathcal{LPO}$ decidibles y con modelos finitos a través de la noción de n-satisfactibilidad; definimos qué es un problema n-abductivo por medio de la n-consecuencia lógica. Una versión de ello se encuentra en [Ne02], donde se utilizan los DB-tableaux como procedimiento mecánico para obtener explicaciones a problemas n-abductivos siempre y cuando el problema $\langle\Theta, \varphi\rangle_n$ cumpla con las siguientes condiciones: tanto φ como α (la explicación) deben ser literales. En contraste nuestra propuesta no requiere de tales restricciones, además de que nosotros definimos la noción de n-consecuencia lógica de manera detallada.

Influenciados nuevamente por [Ne02] proponemos una modificación al método de tablas semánticas de Beth con miras a obtener un procedimiento efectivo que nos permita encontrar soluciones a problemas n-abductivos. Realizamos un cambio a la definición de la δ-regla de extensión de las tablas de Beth, obteniendo una nueva regla de extensión para los enunciados tipo δ a la que llamamos δ_N-regla. Esta regla también depende de la cardinalidad del dominio. A los tableaux construidos a través de esta nueva regla de extensión y las reglas α, β y γ usuales los denominamos N-tableaux. Al igual que para los DB-tableaux demostramos también la completud y correctud de los N-tableaux, la cual está en función de la noción de n-satisfactibilidad.

Desde el punto de vista teórico el trabajo que hemos realizado hasta este punto presenta varias limitaciones, como el hecho de que trabajamos con conjuntos de enunciados satisfactibles en modelos finitos, lo cual podría llevar a la conclusión de que el uso de enunciados de $\mathcal{LPO}$ es una simple abreviatura para disyunciones y conjunciones finitas. Sin embargo, recordemos que desde el punto de vista computacional, finito dista mucho de ser sinónimo de pequeño o manejable. Además, en este ámbito la gran mayoría de problemas solubles están definidos en modelos finitos.

Otro inconveniente podría ser que el procedimiento que proponemos para encontrar una explicación a un problema n-abductivo $\langle\Theta, \varphi\rangle_n$, tiene como prerrequisito conocer la cardinalidad del modelo de Herbrand mínimo de Θ. Dicho problema encuentra soluciones en la rama del razonamiento automático conocida como construcción automática de modelos. En los casos sencillos bastará solicitar a algún constructor automático de modelos, como MACE[1], la búsqueda de modelos finitos de Herbrand para Θ.

Finalmente queremos resaltar que la terminología expuesta a lo largo de este trabajo, la

[1]Ver `http://www-unix.mcs.anl.gov/AR/mace2`

cual se sirve de todo el poder expresivo de la lógica de primer orden, abre una nueva puerta para explorar el problema de abducción en (fragmentos de) $\mathcal{LPO}$, como analizaremos en la siguiente sección.

5.1. Fragmentos decidibles de $\mathcal{LPO}$

En esta sección describimos algunos fragmentos de la $\mathcal{LPO}$ que son decidibles y además gozan de la propiedad del modelo finito. Nos basamos totalmente en [BGG01] y fijamos nuestra atención en aquellos conjuntos de enunciados de $\mathcal{LPO}$ en forma normal prenex, lo cual no es restricción alguna puesto que existe un algoritmo simple para transformar cualquier enunciado de $\mathcal{LPO}$ en dicha forma normal.
Tales fragmentos se caracterizan poniendo restricciones a la forma de los prefijos de cuantificadores y/o al lenguaje.
Entenderemos por *prefijo* aquellas cadenas formadas en el alfabeto $\{\forall, \exists\}$. Para representar prefijos utilizaremos expresiones regulares. Una *secuencia de índice* es una función

$$f : \mathbb{N}^+ \to \mathbb{N} \cup \{\omega\},$$

.

Definición 5.1 *Para cualquier conjunto de prefijos* Π *y cualesquiera dos secuencias de índice* p *y* f. $[\Pi, p, f]$ *(respectivamente* $[\Pi, p, f]_=$*) denota a la colección de todos los enunciados* φ *en forma normal prenex de* $\mathcal{LPO}$ *sin igualdad (respectivamente, con igualdad), tales que:*

- *El prefijo de* φ *pertenece a* Π.
- *El número de símbolos de predicado de índice* n *que figuran en* φ *es menor o igual que* $p(n)$.
- *El número de símbolos de función de índice* n *que figuran en* φ *es menor o igual que* $f(n)$.
- φ *no tiene símbolos de proposición (es decir, símbolos de predicado de índice cero), excepto las constantes lógicas* $\top$ *y* $\bot$ *y tampoco tiene símbolos de constante (es decir, símbolos de función de índice cero).*

Una vez presentada la notación anterior, listamos las clases de enunciados en forma normal prenex de $\mathcal{LPO}$ que cumplen con la propiedad de modelo finito y son decidibles:

$[\exists^*\forall^*, all, (0)]_=$	(Ramsey 1930)
$[\exists^*\forall^2\exists^*, all, (0)]$	(Gödel 1932, Kalmár 1933, Schütte 1934)
$[all, (\omega), (\omega)]$	(Löb 1967, Gurevich 1969)
$[\exists^*\forall\exists^*, all, all]$	(Maslov-Orevkov 1972, Gurevich 1973)
$[\exists^*, all, all]_=$	(Gurevich 1976)

donde la sucesión p (respectivamente f) denotada *all* representa a la secuencia constante ω, es decir, en los enunciados de dichas clases el número de símbolos de predicado (respectivamente de función) es ilimitado.

Para un estudio profundo con respecto al problema de la decisión véase [BGG01].

5.2. Trabajo relacionado

En esta sección mencionamos brevemente algunas investigaciones relacionadas con nuestro trabajo.

5.2.1. Abducción en $\mathcal{LPO}$ por medio de tableaux

Mayer y Pirri [MP93] presentan un modelo para calcular abduccion tanto en lógica de proposiciones como en lógica de primer orden. En el caso de $\mathcal{LPO}$ tratan con problemas abductivos mediante skolemización en reversa, unificación y un procedimiento al cual nombran "herbrandización dinámica" de fórmulas. Para dar una idea de como generan explicaciones, mencionamos los siguientes pasos. (1) se construyen los "conjuntos cerrados mínimos", (2) se buscan soluciones abductivas que sean literales las cuales cierren todas las ramas correspondientes a los conjuntos anteriores y (3) se eliminan soluciones inconsistentes.
Se presentan dos métodos para formalizar el razonamiento abductivo: uno de ellos utiliza tablas semánticas de Beth y el otro un cálculo de secuentes y se muestra la relación que existe entre estos dos métodos. Más aún, se propone una extensión a la lógica de primer orden, la cual probablemente es el estudio más significativo realizado hasta ese momento en abducción de $\mathcal{LPO}$, pues va más allá del marco usual de resolución.
Un punto importante de [MP93] es que muestra claramente el contraste que existe entre la abducción proposicional y la abducción de primer orden. Mientras que la primera es fácil de calcular, aún considerando explicaciones mínimas, la segunda es inherentemente indecidible.

5.2.2. Abducción en $\mathcal{LPO}$ a través de DB-tableaux

En [Ne02] se presenta por vez primera una noción de satisfactibilidad y consecuencia lógica restringida a conjuntos de enunciados con modelos finitos. Además, en [AN03] se aborda la abducción de primer orden para clases de enunciados de la forma $\forall x \exists y \varphi$, con φ una fórmula libre de cuantificadores. En ambos trabajos se define un problema n-abductivo por medio de los DB-tableaux. Cabe mencionar que [AN03] se encuentra aún en fase de revisión por parte de los autores.

5.3. Trabajo futuro

Para terminar nuestro trabajo mencionamos algunas lineas futuras de investigación:

- Estudiar qué otras propiedades de interés cumplen la noción de n-satisfactibilidad así como la relación de n-consecuencia lógica.
- Desarrollar heurísticas que permitan encontrar soluciones a problemas n-abductivos a través N-tableaux o DB-tableaux, de tal forma que sea posible implementar los resultados expuestos aquí con un costo razonable (no exponencial) en cuanto a tiempo y espacio.
- Explorar los fragmentos de enunciados decidibles de $\mathcal{LPO}$ con la propiedad de modelo finito descritos en la Sección 5.1, con el propósito de encontrar algoritmos o heurísticas basadas en la forma que tienen los enunciados que conforman un problema n-abductivo.

Bibliografía

[Al97] Aliseda Llera, A. Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence. Institute for Logic, Language and Computation, Amsterdam. 1997.

[AN03] Aliseda, A., Nepomuceno A. Abduction in First Order Semantic Tableaux. *Sin publicar.* Abril 2003.

[Be55] Beth E.W. Semantic Entailment and Formal Derivability. *Mededelingen der Kon. Ned. Akad. v. Wet.* Vol 18, no.13. pp. 309-342. 1955

[Be56] Beth, E.W. Semantic Construction of Intuitionistic Logic.*Mededelingen der Kon. Ned. Akad. v. Wet.* Vol. 19, no.11. pp. 357-388. 1956.

[Bo84] Boolos, G. 1984., Trees and finite satisfiability: Proof of a conjecture of Burgess. *Notre Dame Journal of Formal Logic* Vol. 25 pp. 193-197. 1984.

[BGG01] Börger, E., Grädel, E., Gurevich, Y. The Classical Decision Problem. Second printing. Springer Verlag, 2001.

[Ch36] Church, A., An Unsolvable Problem of Elementary Number Theory, *American Journal of Mathematics*, Vol.58, pp. 345-363, 1936.

[DGH99] D'Agustino M., Gabbay D.M, Hähnle R. and Posegga, J. Handbook of Tableaux Methods. Kluwer Academic Publishers 1999.

[Di93] Díaz E. 1993. Árboles Semánticos y Modelos Mínimos". En E.Pérez (editor): *Actas del I Congreso de la Sociedad de Lógica, Metodología y Filosofía de la Ciencias en España.* Universidad Complutense de Madrid, pp. 40-43. 1993.

[Ge35] Gentzen G., Untersuchungen über das logische Schliessen, *Mathematische Zeitschrift* Vol. 39 pp. 176-210, 405-431,1935.

[Go30] Gödel K., Die Vollständigkeit der axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik* Vol. 37, pp. 349-360, 1930.

[Hi53] Hintikka J., A New Approach to Sentencial Logics. *Societas Scientiarum Fennica Commentationes Physico-Mathematicae.* Vol. 17, No. 3. 1953.

[MP93] Mayer M.C. and Pirri F., First Order Abducction via Tableaux and Sequent Calculi. *Bulletin of The I.G.P.L.*, Vol. 1. pp.99-117. 1993.

[Ne99] Nepomuceno A., Tablas semánticas y Metalógica (el caso de la lógica de segundo orden). *Crítica.* Vol. XXXXI, No. 93, pp.21-47.

[Ne02] Nepomuceno A., Scientific Explanation and Modified Semantic Tableaux. En Magnani. L., Nerssessian, N.J., Pizzi, C. (Eds). Logical and Computational Aspects of Model-Based Reasoning. Applied Logic Series. pp.181-198. Kluwer 2002.

[Sm63] Smullyan R., A Unifying Principle in Quantification Theory. *Proceedings of the National Academy of Sciences 49*, pp. 828-832, 1963.

[Sm65] Smullyan R., Analytic Natural Deduction. *Journal of Symbolic Logic* Vol. 30 pp. 123-139. 1965.

[Sm66] Smullyan R., Trees and Nest Structures. *Journal of Symbolic Logic* Vol. 31 pp. 303-321. 1966.

[Sm68] Smullyan R., First-Order Logic. Springer-Verlag. 1968. (Revised edition, Dover Press, NY, 1994).

[St91] Stickel M. Upside-Down Meta-Interpretation of the Model Elimination Theorem-Proving Procedure for Deduction and Abduction. Technical Report. Institute for New Generation Computer Technology. (ICOT) TR-664. 1991.

[Sz69] Szabo M.E. (Ed). The Collected Papers of Gerhard Gentzen. Studies in Logic and the Foundations of Mathematics. North-Holland. 1969.

[Tu36] Turing A .M., On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* Vol. 42 pp. 230-265. 1936.

[Wo99] Wolenski J., Logic from a metalogical point of view. En *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa* Orlowska E. (ed.), pp. 25-35. Springer-Verlag 1999.

Índice