

ABDUCCIÓN EN MODELOS FINITOS

FERNANDO SOLER TOSCANO
Departamento de Filosofía, Lógica y Filosofía de la Ciencia
Universidad de Sevilla
fsoler@us.es

ÁNGEL NEPOMUCENO FERNÁNDEZ
Departamento de Filosofía, Lógica y Filosofía de la Ciencia
Universidad de Sevilla
nepomuce@us.es

RESUMEN: Este artículo presenta un acercamiento a la resolución de problemas abductivos en C -estructuras, estructuras que tienen un universo de discurso finito y cada uno de sus elementos es la interpretación de una constante conocida. Empleando una variante del cálculo de tablas semánticas y resolución dual, construimos un procedimiento efectivo para encontrar soluciones abductivas minimales dentro de la semántica propuesta.

PALABRAS CLAVE: lógica de primer orden, resolución dual, tablas semánticas

SUMMARY: We present an approach to abductive reasoning on C -structures, that is, structures with a finite domain such that each of its elements is the interpretation of a known constant of the language. We use a modification of the tableaux calculus and dual resolution to make a process which finds minimal abductive solutions within the proposed semantics.

KEY WORDS: first order logic, dual resolution, semantic tableaux

1. Introducción

El razonamiento abductivo se caracteriza por su proceder inverso a la deducción, de la *conclusión* a las *premisas*. Dado cierto hecho sorprendente \mathcal{C} , para el que no tenemos explicación, si sabemos que desde \mathcal{A} se obtiene \mathcal{C} , podemos suponer \mathcal{A} como *hipótesis plausible*:

$$\frac{\mathcal{C}, \mathcal{A} \rightarrow \mathcal{C}}{\mathcal{A}}$$

Esta regla, incorrecta para la lógica clásica (se trataría de la falacia de afirmación del consecuente), refleja el carácter *retroductivo* (deducción hacia atrás) de la abducción. Son numerosas las aplicaciones que la abducción encuentra en campos tan diversos como la Filosofía de la Ciencia, la Lingüística o la Inteligencia Artificial.

Cuando se intenta desarrollar formalismos para la resolución de problemas abductivos en lógica de primer orden, aparece siempre el

problema de la indecidibilidad: en general, no es posible determinar si, para cierto *problema abductivo*, una hipótesis propuesta es o no una *solución abductiva* correcta. Sin embargo, en campos como el de la Programación Lógica (por ejemplo, en *Answer Set Programming*), es frecuente la reducción de ciertos problemas de satisfacibilidad en primer orden a dominios de cardinalidad finita. En este trabajo seguimos una estrategia similar, con la particularidad de que el acercamiento a la abducción que presentamos integra dos sistemas ya acreditados dentro del panorama lógico del razonamiento explicativo, pero que hasta ahora sólo habían sido considerados por separado. Se trata de los sistemas basados en tablas semánticas y en resolución. Así, empleamos cierta versión del cálculo de tablas semánticas para la reducción de los problemas abductivos a dominios de cardinalidad finita y entonces, en un marco proposicional, usamos resolución dual para la búsqueda de soluciones abductivas.

A continuación, introducimos las nociones fundamentales que se usarán a lo largo de este trabajo. Por una parte, los conceptos de δ -cláusula y forma δ -clausal, que serán básicos tanto para la composición de soluciones abductivas como para las demostraciones mediante δ -resolución. Además, presentamos la semántica de *C-estructuras* finitas en la que trabajaremos. Finalmente, daremos las nociones de *problema abductivo* y *solución C-abductiva*. En la sección 2 introduciremos el cálculo de *C-tablas* y probaremos sus propiedades fundamentales. Hacemos lo mismo en la sección 3 para el cálculo de δ -resolución. Finalmente, en la sección 4 mostramos los resultados principales de este trabajo y componemos un procedimiento que, dado un problema abductivo, obtiene todas sus soluciones *C-abductivas* por medio de la construcción de *C-tablas* y de la demostración mediante δ -resolución.

Usamos \mathcal{L} para denotar un lenguaje de primer orden sin identidad ni funtores, definido con las convenciones habituales. En cuanto a la semántica, no contemplaremos asignaciones de valores a las variables libres, con lo que restringimos la interpretación a las sentencias de \mathcal{L} . Por ello, cuando nos refiramos a un elemento de \mathcal{L} , o una fórmula, sobrentenderemos que se trata de una sentencia. Dada la secuencia t_1, \dots, t_n de términos (constantes y variables) y el predicado n -ádico P , decimos que P_{t_1, \dots, t_n} y $\neg P_{t_1, \dots, t_n}$ son literales complementarios (positivo y negativo, respectivamente). Dado cualquier literal λ (positivo o negativo), representamos mediante $\bar{\lambda}$ su complementario (negativo o positivo, respectivamente). En adelante, usaremos letras grie-

gas minúsculas para representar fórmulas y letras griegas mayúsculas para conjuntos de fórmulas.

Definición 1 (δ -cláusula). Una δ -cláusula $\Sigma = \{\lambda_1, \dots, \lambda_n\}$ es un conjunto finito de literales de \mathcal{L} tales que todos sus términos son constantes. Dada una estructura \mathcal{M} , $\mathcal{M} \models \Sigma$ syss (si y sólo si) $\mathcal{M} \models \lambda_i$ para todo literal λ_i , $1 \leq i \leq n$. Mediante \diamond representamos la δ -cláusula vacía, universalmente válida.

Definición 2 (forma δ -clausal). Una forma δ -clausal $A = \{\Sigma_1, \dots, \Sigma_n\}$ es un conjunto finito de δ -cláusulas. Dada una estructura \mathcal{M} , $\mathcal{M} \models A$ syss $\mathcal{M} \models \Sigma_i$ para al menos una δ -cláusula Σ_i , $1 \leq i \leq n$. La forma δ -clausal vacía es no satisfacible.

Las definiciones anteriores son duales a las de cláusula y forma clausal del cálculo de resolución estándar. Ahora bien, una δ -cláusula equivale a una conjunción (no disyunción) elemental de literales, y una forma δ -clausal a una forma normal disyuntiva (no conjuntiva). Dada esta dualidad con las definiciones estándar, podremos obtener resultados como el teorema 16 (sección 3), en el que veremos que las δ -cláusulas que se obtienen en el cálculo que definiremos no son consecuencia lógica del conjunto original de δ -cláusulas, como ocurre en los cálculos deductivos. Al contrario, la forma δ -clausal que hará las veces de *premisas*, será consecuencia lógica de cada una de las δ -cláusulas que se obtengan a partir de ella. De aquí el carácter abductivo de nuestro cálculo.

La noción principal en la semántica que usaremos es la de C -estructura, una \mathcal{L} -estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ con un dominio finito \mathcal{D} , de la misma cardinalidad que el conjunto de constantes \mathcal{C} para el que se define, y donde cada una de dichas constantes recibe, por interpretación, un elemento diferente de \mathcal{D} . En cuanto a la interpretación del resto de constantes de \mathcal{L} , no hay ninguna restricción. Con ello añadimos, a la semántica de modelos finitos, la posibilidad de contar con un *designador* para cada elemento del universo de discurso. La noción de C -satisfacibilidad será la propia de la semántica habitual de \mathcal{L} , pero ahora restringida a C -estructuras. El resto de conceptos semánticos se define a partir de éste.

Definición 3 (C -estructura). Dado el conjunto $C = \{c_1, \dots, c_n\}$ finito y no vacío de constantes de \mathcal{L} , definimos una C -estructura como una estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, con \mathcal{D} como universo de discurso e \mathcal{I} como función interpretación, tal que, además de los requisitos habituales, se verifican:

- $|\mathcal{D}| = n, y$
- Si $i \neq j, 1 \leq i, j \leq n$, entonces $\mathcal{I}(c_i) \neq \mathcal{I}(c_j)$.

Definición 4 (C-satisfacibilidad, C-consecuencia). Dado cualquier conjunto de constantes C y cualesquiera fórmulas $\alpha, \beta \in \mathcal{L}$, decimos que:

- α es C-satisfacible syss existe una C-estructura \mathcal{M} tal que $\mathcal{M} \models \alpha$.
- α es C-válida syss para cualquier C-estructura \mathcal{M} se verifica $\mathcal{M} \models \alpha$; en símbolos $\models_C \alpha$.
- β es C-consecuencia lógica de α syss para cualquier C-estructura \mathcal{M} se verifica que si $\mathcal{M} \models \alpha$ entonces $\mathcal{M} \models \beta$; en símbolos $\alpha \models_C \beta$.
- α y β son C-equivalentes syss $\alpha \models_C \beta$ y $\beta \models_C \alpha$.

Las nociones anteriores se pueden extender, de manera natural, a conjuntos de fórmulas, así como a δ -cláusulas y formas δ -clausales, teniendo en cuenta las condiciones de satisfacibilidad que imponen las definiciones 1 y 2.

Las dos definiciones siguientes introducen las nociones de *problema abductivo* y *solución abductiva*. Son definiciones habituales dentro del tratamiento lógico del razonamiento abductivo (Aliseda 2006; Kakas *et al.* 1998), aunque adaptadas a nuestra semántica. Dados el conjunto de fórmulas Θ , que hace las veces de teoría, y la fórmula φ , la observación, diremos que $\langle \Theta, \varphi \rangle$ es un *problema abductivo* syss ni φ ni su negación son consecuencia de Θ .

Definición 5 (problema abductivo). Dados el conjunto $\Theta \subset \mathcal{L}$ y $\varphi \in \mathcal{L}$, decimos que el par $\langle \Theta, \varphi \rangle$ es un problema abductivo si se verifican $\Theta \not\models \varphi$ y $\Theta \not\models \neg\varphi$.

Obsérvese que definimos la noción de *problema abductivo* en relación con la semántica estándar, esto es, usando la relación de consecuencia lógica clásica. Sin embargo, definiremos la noción de *solución abductiva* dentro del marco de la semántica de C-estructuras que hemos presentado. Esto no es ninguna contradicción, pues nuestro acercamiento está orientado a la solución en modelos finitos de problemas abductivos que aparecen en lógica clásica de primer orden.

Definición 6 (solución C -abductiva). Dado el problema abductivo $\langle \Theta, \varphi \rangle$, decimos que la δ -cláusula Σ es una solución C -abductiva al mismo si:

1. $\Theta \cup \Sigma \models_C \varphi$.
2. $\Theta \cup \Sigma$ es C -satisfacible.
3. $\Sigma \not\models_C \varphi$.
4. No existe ninguna δ -cláusula $\Sigma' \subset \Sigma$ tal que $\Theta, \Sigma' \models_C \varphi$.

Mediante $\text{Abd}(\Theta, \varphi)_C$ denotamos el conjunto de soluciones C -abductivas al problema abductivo $\langle \Theta, \varphi \rangle$.

Los requisitos que la definición 6 impone a una solución C -abductiva son la adaptación a las C -estructuras de requisitos habituales en las definiciones formales de solución abductiva. Concretamente, 1 es el requisito que Aliseda (2006) llama de *abducción plana*, es decir, φ es consecuencia lógica (C -consecuencia) de la unión de la teoría Θ y la explicación Σ . Siguiendo la misma terminología, 2 se corresponde con el requisito de *abducción consistente*, al exigir que sea C -satisfacible la extensión de la teoría con la explicación. Del mismo modo, el requisito 3 es la adaptación del requisito de *abducción explicativa*, ya que la observación no será C -consecuencia de la explicación por sí sola, sino que es necesaria la teoría. Finalmente, 4 impone un criterio de *minimalidad*, al seleccionar sólo los conjuntos de literales tales que ningún subconjunto propio suyo sea una explicación C -abductiva.

2. El cálculo de C -tablas

El cálculo de C -tablas parte de una modificación del método de las tablas semánticas de Beth (1955) introducida paralelamente por Boolos (1984) y Díaz (1993), con objeto de obtener modelos finitos para fórmulas que con las tablas semánticas estándar producirían ramas infinitas. Estas tablas modificadas ya han sido aplicadas al tratamiento formal de la abducción en (Nepomuceno 2002; Reyes *et al.* 2006), como extensión del procedimiento abductivo de (Aliseda 1997; Cialdea Mayer *et al.* 1993). En este trabajo definimos una variante que toma como referencia, en vez de la cardinalidad de los modelos buscados, el conjunto C de constantes que define la clase de C -estructuras para la que trataremos de comprobar si cierto conjunto de fórmulas es C -satisfacible. Con respecto a las tablas

α	α_1	α_2
$\lambda \wedge \eta$	λ	η
$\neg(\lambda \vee \eta)$	$\neg\lambda$	$\neg\eta$
$\neg(\lambda \rightarrow \eta)$	λ	$\neg\eta$

β	β_1	β_2
$\lambda \vee \eta$	λ	η
$\neg(\lambda \wedge \eta)$	$\neg\lambda$	$\neg\eta$
$\lambda \rightarrow \eta$	$\neg\lambda$	η

Regla α : $\frac{\alpha}{\alpha_1 \mid \alpha_2}$
--

Regla β : $\frac{\beta}{\beta_1 \mid \beta_2}$
--

Figura 1: Clasificación de fórmulas y reglas α y β

semánticas estándar, sólo modificaremos las reglas γ y δ . El resto no cambia. En la figura 1 se muestra la clasificación habitual de fórmulas proposicionales en los tipos α y β , así como sus subfórmulas correspondientes. La regla α se aplica a fórmulas de tipo α y añade a la rama correspondiente cada una de sus subfórmulas. La regla β , por su parte, divide la rama en dos nuevas y añade a cada una de ellas una de las subfórmulas de la fórmula tipo β a la que se aplica. La regla de doble negación toma una fórmula tipo $\neg\neg\lambda$ y añade λ a la misma rama. Por último, la regla de cierre clausura todas las ramas en las que ocurra un literal y su complementario. Para el tratamiento de los cuantificadores \forall y \exists se introducen, respectivamente, las reglas γ y δ . Serán éstas las que modificaremos para construir C -tablas.

Definición 7 (C -tabla). *Dado un conjunto finito $\Theta \subset \mathcal{L}$ y un conjunto finito y no vacío de constantes $C = \{c_1, \dots, c_n\}$ entre las que aparecen todas las de Θ , una C -tabla de Θ , que denotamos mediante $\mathcal{T}(\Theta)_C$, es una tabla semántica que difiere de las tablas de Beth únicamente en las reglas γ y δ :*

$$\begin{array}{c}
 \frac{\forall x\varphi}{\varphi(x/c_1)} \\
 \vdots \\
 \varphi(x/c_n) \\
 \hline
 \exists x\varphi \\
 \varphi(x/c_1) \mid \dots \mid \varphi(x/c_n)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\neg\exists x\varphi}{\neg\varphi(x/c_1)} \\
 \vdots \\
 \neg\varphi(x/c_n) \\
 \hline
 \neg\forall x\varphi \\
 \neg\varphi(x/c_1) \mid \dots \mid \neg\varphi(x/c_n)
 \end{array}$$

Ahora bien, cuando encontramos una fórmula de tipo γ como $\forall x\varphi$ (o $\neg\exists x\varphi$), añadimos a la rama todas las instancias de $\varphi(x/c)$

(respectivamente, de $\neg\varphi(x/c)$) para cada una de las constantes $c \in C$. En el caso de las fórmulas de tipo δ como $\exists x\varphi$ (o $\neg\forall x\varphi$), se divide la rama en la que ocurra dicha fórmula en $|C|$ nuevas ramas y a cada una de ellas se añade una instancia $\varphi(x/c)$ (respectivamente, de $\neg\varphi(x/c)$) con una constante $c \in C$ diferente en cada caso.

A modo de ejemplo, en la figura 2 mostramos la $\{a, b\}$ -tabla de $\forall x\exists y(Pxy \wedge \neg Pxx)$, es decir, $\mathcal{T}(\{\forall x\exists y(Pxy \wedge \neg Pxx)\})_{\{a,b\}}$. La siguiente definición introduce la idea de *forma δ -clausal de una C -tabla*, una forma δ -clausal que tiene una δ -cláusula por cada rama abierta de la C -tabla correspondiente. Para la $\{a, b\}$ -tabla de la figura 2, como sólo tiene una rama abierta, su forma δ -clausal será $\{\{Pab, \neg Paa, Pba, \neg Pbb\}\}$ que tiene una sola δ -cláusula con los literales de dicha rama.

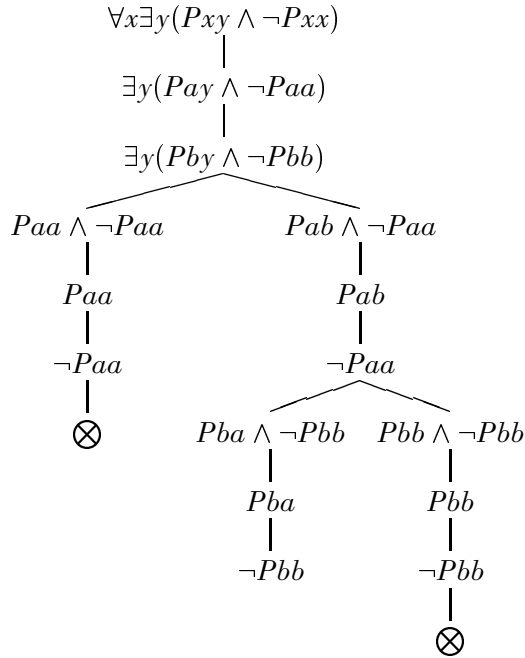


Figura 2: $\{a, b\}$ -tabla de $\forall x\exists y(Pxy \wedge \neg Pxx)$

Definición 8 (forma δ -clausal de una C -tabla). La forma δ -clausal de $\mathcal{T}(\Theta)_C$, llamada $\mathcal{C}\delta(\mathcal{T}(\Theta)_C)$, es la más pequeña que contiene, por cada rama abierta de la C -tabla $\mathcal{T}(\Theta)_C$, una δ -cláusula con sus literales.

El resto de esta sección se dedica a demostrar que toda fórmula $\eta \in \mathcal{L}$ es C -equivalente a la forma δ -clausal de su C -tabla. Esto es, que η y $\mathcal{C}\delta(\mathcal{T}(\{\eta\})_C)$ son C -equivalentes. Esto se demostrará en el teorema 13. Antes, debemos probar varios resultados intermedios.

Lema 9. *Dado un conjunto de literales Θ que no contiene ningún par de literales complementarios, si todos los términos que aparecen en Θ están contenidos en el conjunto de constantes C , entonces Θ es C -satisfacible.*

Prueba: Sea Θ un conjunto de literales tal como indica el enunciado. Como no contiene literales complementarios, puede definirse una C -estructura $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ que satisface todos sus literales. Basta tomar $\mathcal{D} = C$ y $\mathcal{I}(c) = c$ para toda $c \in C$. Por otra parte, para cada predicado P de \mathcal{L} , $\mathcal{I}(P)$ es el conjunto más pequeño tal que para cada literal positivo $P(c_1, \dots, c_n) \in \Theta$, $\langle c_1, \dots, c_n \rangle \in \mathcal{I}(P)$. Fácilmente se comprueba que $\mathcal{M} \models \Theta$. ■

Corolario 10. *El conjunto de los literales que pertenecen a cualquier rama abierta de una C -tabla es C -satisfacible.*

Prueba: Sea Θ el conjunto de literales de una rama abierta de una C -tabla. Por la definición 7, todos los términos de Θ son constantes de C , y en Θ no hay literales complementarios. Entonces, por el lema 9, Θ es C -satisfacible. ■

Lema 11. *Si Σ es el conjunto de literales que pertenecen a una rama abierta de la C -tabla $\mathcal{T}(\{\theta_1, \dots, \theta_m\})_C$, entonces:*

$$\Sigma \models_C \theta_1 \wedge \dots \wedge \theta_m$$

Prueba: Sea $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ una C -estructura que satisface Σ , conjunto de literales de una rama abierta de $\mathcal{T}(\{\theta_1, \dots, \theta_m\})_C$. Tomemos $C = \{c_1, \dots, c_n\}$. Probemos que \mathcal{M} satisface todas las fórmulas de dicha rama, por inducción sobre su grado lógico. En el caso base son literales que, por hipótesis, son satisfechos por \mathcal{M} . Supongamos que \mathcal{M} satisface todas las fórmulas de la rama hasta las de grado i . Sea λ una fórmula de grado $i + 1$. No consideraremos el caso en que λ es un literal negativo, pues ya sabemos que todos son satisfechos por \mathcal{M} . Por tanto, γ sólo puede ser:

- Una doble negación $\neg\neg\epsilon$. Entonces, como para completar la construcción de la C -tabla se debió aplicar la regla de doble negación a λ , tenemos que ϵ está en la rama, y por ser su grado lógico menor o igual que i , $\mathcal{M} \models \epsilon$, y por evaluación de \neg , $\mathcal{M} \models \gamma$.
- Una fórmula de tipo α . Entonces, sus dos componentes deben encontrarse en la rama, y por ser ambas de grado lógico menor o igual que i , ambas deben ser satisfechas por \mathcal{M} . Por evaluación de los signos lógicos de las fórmulas de tipo α , tenemos que $\mathcal{M} \models \lambda$.
- Una fórmula de tipo β . Entonces, una de sus componentes debe estar en la rama, y por ser de grado lógico menor o igual que i , es satisfecha por \mathcal{M} . Por evaluación de los signos lógicos de las fórmulas de tipo β , $\mathcal{M} \models \lambda$.
- Una fórmula de tipo γ , como $\forall x\varphi$ (el caso en que λ es $\neg\exists\varphi$ es similar). Entonces, como se debió aplicar la regla γ durante la construcción de la C -tabla, todas las subfórmulas tipo $\varphi(x/c_j)$, $1 \leq j \leq n$, deben estar en la rama, y por ser de grado menor o igual que i son satisfechas por \mathcal{M} . Además, como el valor de la función \mathcal{I} para las constantes c_j recorre todo el dominio \mathcal{D} se verifica, por evaluación de \forall , $\mathcal{M} \models \lambda$.
- Una fórmula de tipo δ , como $\exists x\varphi$ (es similar el caso en que λ es $\neg\forall x\varphi$). Entonces, por la aplicación de la regla δ , la rama debe contener cierta subfórmula $\varphi(x/c_j)$, $1 \leq j \leq n$, que por hipótesis de inducción es satisfecha por \mathcal{M} . Por lo tanto, por evaluación de \exists , $\mathcal{M} \models \lambda$.

Por consiguiente, \mathcal{M} satisface todas las fórmulas de la rama, y por ello también $\mathcal{M} \models \theta_k$, para cada valor de k entre 1 y m . Por evaluación de \wedge , $\mathcal{M} \models \theta_1 \wedge \dots \wedge \theta_m$. ■

Lema 12. *Para cualquier conjunto finito C -satisfacible $\Theta \subset \mathcal{L}$ se cumple:*

1. *Cualquier C -estructura que satisfaga Θ satisface todas las fórmulas de al menos una de las ramas de $\mathcal{T}(\Theta)_C$.*
2. *$\mathcal{T}(\Theta)_C$ es abierta.*

Prueba: Sea Θ un conjunto finito de fórmulas C -satisfacible, para cierto conjunto de constantes C , y $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ una C -estructura que satisface Θ . Demostraremos que \mathcal{M} satisface todas las fórmulas de al menos una rama de $\mathcal{T}(\Theta)_C$, por inducción en el número de veces que se ha aplicado alguna regla de construcción de C -tablas. En el caso base, con 0 aplicaciones, la única rama de la C -tabla tiene sólo las fórmulas de Θ , que son satisfechas por \mathcal{M} . Ahora supongamos que \mathcal{M} satisface todas las fórmulas de cierta rama hasta la i -ésima aplicación de reglas. Consideremos la $(i+1)$ -ésima regla que se aplica a cierta fórmula λ . En caso de que dicha regla no afecte a la rama que estamos estudiando, es trivial que la rama sigue siendo satisfecha por \mathcal{M} tras su aplicación. En otro caso, veamos qué ocurre según dicha regla sea:

- La regla de doble negación. Entonces, λ es $\neg\neg\lambda'$, y se añade a la rama λ' . Como por hipótesis $\mathcal{M} \models \lambda$, entonces, por evaluación de \neg , $\mathcal{M} \models \lambda'$, con lo que \mathcal{M} sigue satisfaciendo todas las fórmulas de la rama.
- La regla α . Entonces, λ tiene la forma $\lambda_1 \wedge \lambda_2$, o bien $\neg(\lambda_1 \vee \lambda_2)$, etc., y se añaden a la rama las subfórmulas λ_1 y λ_2 o bien $\neg\lambda_1$ y $\neg\lambda_2$, etc. En cualquier caso, como por hipótesis \mathcal{M} satisface λ , por evaluación de los signos lógicos, \mathcal{M} satisface las dos nuevas subfórmulas, por lo que \mathcal{M} sigue satisfaciendo todas las fórmulas de la rama tras la aplicación de la $(i+1)$ -ésima regla.
- La regla β . Entonces λ tiene la forma $\lambda_1 \vee \lambda_2$, o bien $\neg(\lambda_1 \wedge \lambda_2)$, etc., y entonces se divide la rama en dos y a cada una se añade una de las subfórmulas λ_1 y λ_2 o bien $\neg\lambda_1$ y $\neg\lambda_2$, etc. Como por hipótesis $\mathcal{M} \models \lambda$, por evaluación de los signos lógicos, \mathcal{M} satisface al menos una de las dos nuevas subfórmulas, por lo que \mathcal{M} satisface al menos una de las dos nuevas ramas tras la aplicación de la $(i+1)$ -ésima regla.
- La regla γ . Entonces λ es $\forall x\varphi$ (si λ es $\neg\exists x\varphi$ la prueba es similar). Por lo tanto, al aplicar la regla γ se añaden a la rama todas las subfórmulas $\varphi(x/c_j)$, $1 \leq j \leq n$. Pero como por hipótesis $\mathcal{M} \models \forall x\varphi$ entonces, por evaluación de \forall se verifica que $\mathcal{M} \models \varphi(x/c_j)$ para cada constante c_j , $1 \leq j \leq n$. Por tanto, \mathcal{M} satisface la rama tras la aplicación de la regla γ .
- La regla δ . En este caso, λ es $\exists x\varphi$ (si λ es $\neg\forall x\varphi$ la prueba es similar), y por hipótesis de inducción $\mathcal{M} \models \lambda$, es decir,

$\mathcal{M} \models \exists x\varphi$. Como \mathcal{M} asigna a cada constante de $\mathcal{T}(\{\eta\})_C$ un elemento diferente de \mathcal{D} , se verifica por evaluación de \exists , que para al menos una constante c_j , $1 \leq j \leq n$, $\mathcal{M} \models \varphi(x/c_j)$. Pero una de las nuevas ramas que surgen tras la aplicación de la regla δ contiene $\varphi(x/c_j)$, por lo que sigue habiendo una rama cuyas fórmulas son todas satisfechas por \mathcal{M} .

Al finalizar la construcción de la C -tabla habrá, en consecuencia, al menos una rama tal que todas sus fórmulas son satisfechas por \mathcal{M} , con lo que resulta probado el punto primero del lema 12. Pero, además, entre las fórmulas de la rama satisfecha por \mathcal{M} no puede haber literales complementarios. Por ello, la C -tabla es abierta, lo que prueba la segunda parte del lema. ■

Teorema 13. Sean $\eta \in \mathcal{L}$ y $\mathcal{T}(\{\eta\})_C$ una C -tabla de $\{\eta\}$. Entonces,

$$\eta \quad y \quad \mathcal{C}\delta(\mathcal{T}(\{\eta\})_C)$$

son C -equivalentes.

Prueba: Sea \mathcal{M} una C -estructura que satisface η . Entonces, por el lema 12 se verifica que \mathcal{M} satisface todas las fórmulas de al menos una rama de $\mathcal{T}(\{\eta\})_C$. Por lo tanto, \mathcal{M} satisface todos los literales de dicha rama, entre los cuales no puede haber literales complementarios, con lo que son literales de una rama abierta de $\mathcal{T}(\{\eta\})_C$. Por la definición 8, tales literales constituyen una δ -cláusula de $\mathcal{C}\delta(\mathcal{T}(\{\eta\})_C)$. Por las definiciones 1 y 2 tenemos que $\mathcal{M} \models \mathcal{C}\delta(\mathcal{T}(\{\eta\})_C)$.

Ahora, sea \mathcal{M} una C -estructura que satisface $\mathcal{C}\delta(\mathcal{T}(\{\eta\})_C)$. Por las definiciones 1 y 2, \mathcal{M} satisface todos los literales de al menos una rama abierta de $\mathcal{T}(\{\eta\})_C$. Por el lema 11, $\mathcal{M} \models \eta$. ■

3. Búsqueda mediante δ -resolución

El cálculo de resolución (Robinson 1965) es posiblemente el procedimiento lógico más empleado en demostración automática de teoremas. En esta sección presentamos el cálculo de δ -resolución que nos servirá para obtener hipótesis explicativas para cierto conjunto de fórmulas. Comenzamos presentando la regla de δ -resolución que, aunque sintácticamente idéntica, es semánticamente dual a la regla de resolución clásica, al ser las δ -cláusulas equivalentes a la conjunción

de sus literales. Puesto que trabajaremos con formas δ -clausales sin variables (tras la reducción a modelos finitos que haremos con las C -tablas), no empleamos en las definiciones nociones como renombramiento de variables, unificación, etc. La idea de un cálculo dual a la resolución no es nueva. Ya Quine (1955) define $\phi\psi$ como el *consenso* de los *implicantes* $\alpha\phi$ y $\bar{\alpha}\psi$ —que interpreta conjuntivamente, como las δ -cláusulas—. Sin embargo, hasta donde sabemos es (Soler *et al.* 2006) donde por primera vez se aplica la resolución dual a la formalización del razonamiento abductivo.

Definición 14 (regla de δ -resolución). Dadas dos δ -cláusulas $\Sigma_1 \cup \{\lambda\}$ y $\Sigma_2 \cup \{\neg\lambda\}$, la regla de δ -resolución produce su δ -resolvente $\Sigma_1 \cup \Sigma_2$:

$$\frac{\Sigma_1 \cup \{\lambda\} \quad \Sigma_2 \cup \{\neg\lambda\}}{\Sigma_1 \cup \Sigma_2}$$

Definición 15 (demostración mediante δ -resolución). Una δ -cláusula Σ es demostrable mediante δ -resolución a partir de la forma δ -clausal A , lo que representamos como $A \vdash_\delta \Sigma$, si existe una secuencia de δ -cláusulas tal que:

- Cada δ -cláusula de la secuencia o bien pertenece a A o es un δ -resolvente de δ -cláusulas anteriores.
- La última δ -cláusula de la secuencia es Σ .

Como puede apreciarse, no hay ninguna diferencia sintáctica entre la regla de resolución habitual y la de δ -resolución. Sin embargo, como trabajamos con δ -cláusulas que equivalen a conjunciones de literales —y no a disyunciones, como en el cálculo de resolución— tendremos que el δ -resolvente $\Sigma_1 \cup \Sigma_2$ no es consecuencia lógica de las δ -cláusulas $\Sigma_1 \cup \{\lambda\}$ y $\Sigma_2 \cup \{\neg\lambda\}$, sino al revés: toda forma δ -clausal que contenga estas dos últimas δ -cláusulas será consecuencia lógica de $\Sigma_1 \cup \Sigma_2$. Esto es lo que se demuestra en el siguiente teorema 16 de corrección. El teorema 17 prueba la completud del cálculo, es decir, que a partir de toda forma δ -clausal C -válida podremos demostrar \diamond mediante δ -resolución. El teorema 18 prueba la *completud abductiva* del cálculo, es decir, que desde toda forma δ -clausal A será posible obtener, mediante δ -resolución, las δ -cláusulas más pequeñas de las que A es C -consecuencia.

Teorema 16 (corrección). Dadas la forma δ -clausal A y la δ -cláusula Σ , si $A \vdash_\delta \Sigma$ entonces $\Sigma \models A$.

Prueba: Hacemos la prueba por inducción sobre el número de aplicaciones de la regla de δ -resolución. En el caso base, con 0 aplicaciones, $\Sigma \in A$, y por la definición 2, $\Sigma \models A$. Consideremos que hasta la n -ésima aplicación, A es consecuencia lógica de cada δ -cláusula resultante. Sea $\Sigma = \Sigma_1 \cup \Sigma_2$ la $n + 1$ -ésima δ -cláusula obtenida, un δ -resolvente de $\Sigma_1 \cup \{\lambda\}$ y $\Sigma_2 \cup \{\neg\lambda\}$. Entonces, cada estructura \mathcal{M} que satisfaga Σ satisface una de las δ -cláusulas anteriores $\Sigma_1 \cup \{\lambda\}$ o $\Sigma_2 \cup \{\neg\lambda\}$, puesto que \mathcal{M} satisface cada literal de $\Sigma_1 \cup \Sigma_2$ y $\mathcal{M} \models \lambda$ o $\mathcal{M} \models \neg\lambda$. Finalmente, por hipótesis de inducción, \mathcal{M} satisface A . Por lo tanto, $\Sigma \models A$. ■

Teorema 17 (completud). *Sea A una forma δ -clausal cuyas constantes pertenecen a C . Entonces, si A es C -válida, $A \vdash_\delta \diamond$.*

Prueba: Sea A una forma δ -clausal C -válida, s la cardinalidad de A y t la suma de las cardinalidades de todas las δ -cláusulas de A . Tomemos $k = t - s$. Procedemos por inducción sobre el valor de k , considerando que $\diamond \notin A$ (para evitar el caso trivial, donde es obvio que $A \vdash_\delta \diamond$). Si $k = 0$, la única posibilidad es que A sea un conjunto de δ -cláusulas unitarias (de un solo literal). Entonces, como la interpretación que toda C -estructura hace de cada constante de A es diferente (por ser todas constantes de C), si A es C -válida, sólo puede ocurrir que contenga dos δ -cláusulas $\{\lambda\}$ y $\{\neg\lambda\}$. Con sólo una aplicación de la regla de δ -resolución, obtenemos \diamond .

Supongamos que el teorema se verifica para $k \leq n$, así que lo probaremos para $k = n + 1$. En este caso, debe haber en A una δ -cláusula $\Sigma = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$, donde $m \geq 2$. Entonces, definimos $\Sigma' = \{\lambda_1\}$, y $\Sigma'' = \{\lambda_2, \dots, \lambda_m\}$. Pero si A es C -válida, también lo son $(A - \Sigma) \cup \{\Sigma'\}$ y $(A - \Sigma) \cup \{\Sigma''\}$ (esto es una conclusión directa a partir de las definiciones 1 y 2), y para tales conjuntos, $k \leq n$, por lo que podemos obtener \diamond desde ellos, mediante dos pruebas (secuencias de δ -cláusulas) que llamaremos Dem' y Dem'' , respectivamente. Sea Dem una prueba a partir de A , construida de forma parecida a Dem'' , pero cada vez que Σ'' se usa en Dem'' , se usa Σ en Dem . Entonces, como la única diferencia entre Σ'' y Σ es que Σ contiene λ_1 , la última δ -cláusula en Dem es o bien \diamond (en este caso $A \vdash_\delta \diamond$, y la prueba termina) o $\{\lambda_1\}$. En el último caso, ya han aparecido en la demostración todas las δ -cláusulas de $(A - \Sigma) \cup \{\Sigma'\}$ (puesto que Σ' es $\{\lambda_1\}$), y podemos completarla como Dem' para obtener \diamond . ■

Teorema 18 (completud abductiva). *Sea A una forma δ -clausal C -equivalente a $\alpha \in \mathcal{L}$, que sólo contiene constantes de C . Entonces, $A \vdash_\delta \Sigma$ para cada δ -cláusula Σ tal que:*

1. Σ es C -satisfacible.
2. $\Sigma \models_C \alpha$.
3. No existe ninguna δ -cláusula $\Sigma' \subset \Sigma$ tal que $\Sigma' \models_C \alpha$.

Prueba: Sean $\alpha \in \mathcal{L}$, A una forma δ -clausal C -equivalente a α que sólo contiene constantes de C y $\Sigma = \{\lambda_1, \dots, \lambda_n\}$ una δ -cláusula con las propiedades que indica el enunciado del teorema. Probemos que $A \vdash_\delta \Sigma$. Como $\Sigma \models_C \alpha$, tenemos que $\models_C \lambda_1 \wedge \dots \wedge \lambda_n \rightarrow \alpha$, ya que cada C -estructura o bien no satisface Σ o satisface α (por ser α C -consecuencia lógica de Σ). Por evaluación de \rightarrow y \neg , tenemos que $\models_C \overline{\lambda_1} \vee \dots \vee \overline{\lambda_n} \vee \alpha$. Entonces, como esta fórmula es C -equivalente a la forma δ -clausal $A \cup \{\{\overline{\lambda_1}\}, \dots, \{\overline{\lambda_n}\}\}$ (definiciones 1 y 2), es posible obtener \diamond desde ella (teorema 17), mediante una prueba, que llamamos Dem . De modo similar a como hicimos al probar el teorema 17, construimos una prueba paralela a Dem , que llamamos Dem' , que sólo emplea δ -cláusulas de A . Ahora bien, cuando una δ -cláusula de tipo $\{\overline{\lambda_i}\}$, $1 \leq i \leq n$, se usa en Dem , no se hace nada en Dem' . Fácilmente observamos que la última δ -cláusula de Dem' debe ser una $\Sigma' \subseteq \Sigma$, pues cada δ -cláusula $\{\overline{\lambda_i}\}$ sólo puede eliminar el literal λ_i en la prueba. Pero si $\Sigma' \subset \Sigma$, como por el teorema 16, $\Sigma' \models A$, entonces $\Sigma' \models_C A$, y por lo tanto $\Sigma' \models_C \alpha$, lo cual es contrario a lo que sabemos de Σ . En consecuencia, $\Sigma' = \Sigma$, de modo que $A \vdash_\delta \Sigma$. ■

Es conocido que el cálculo de resolución estándar emplea ciertas técnicas con objeto de optimizar la búsqueda. A continuación adaptamos dos de dichas técnicas al cálculo de δ -resolución. En primer lugar, definimos la relación de *subsunción* entre δ -cláusulas, y demostramos, en el corolario 20, que cada δ -cláusula $\Sigma \cup \Lambda$ que resulte subsumida por otra δ -cláusula Σ durante una prueba mediante δ -resolución puede ser eliminada. Después, el corolario 21 prueba que las δ -cláusulas contradictorias —aquellas que como $\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}$, $n \geq 0$, contienen un literal y su negación— pueden ser también desechadas.

Definición 19 (subsunción). *Decimos que la δ -cláusula Σ' subsume la δ -cláusula Σ si $\Sigma' \subset \Sigma$.*

Corolario 20. *Para todas las δ -cláusulas Γ , Σ y Λ , y cada forma δ -clausal A , si $A \cup \{\Sigma\} \cup \{\Sigma \cup \Lambda\} \vdash_\delta \Gamma$ y Γ es C -satisfacible, para cierto conjunto de constantes C , entonces existe una δ -cláusula Γ' tal que $A \cup \{\Sigma\} \vdash_\delta \Gamma'$ y $\Gamma' \subseteq \Gamma$.*

Prueba: Si eliminamos $\Sigma \cup \Lambda$ de $A \cup \{\Sigma\} \cup \{\Sigma \cup \Lambda\}$, la forma δ -clausal resultante, $A \cup \{\Sigma\}$, es C -equivalente a la anterior, como puede observarse fácilmente mediante un simple razonamiento semántico basado en las definiciones 1 y 2. Por el teorema 16, $\Gamma \models A \cup \{\Sigma\} \cup \{\Sigma \cup \Lambda\}$, con lo que $\Gamma \models_C A \cup \{\Sigma\} \cup \{\Sigma \cup \Lambda\}$, y por la C -equivalencia mencionada, $\Gamma \models_C A \cup \{\Sigma\}$. Pero sea $\alpha \in \mathcal{L}$ una fórmula C -equivalente a $A \cup \{\Sigma\}$ (basta con que α sea la disyunción de las conjunciones elementales formadas a partir de cada δ -cláusula de $A \cup \{\Sigma\}$). Entonces, $\Gamma \models_C \alpha$, por lo que debe existir una δ -cláusula satisfacible $\Gamma' \subseteq \Gamma$ tal que $\Gamma' \models_C \alpha$ y para cualquier $\Gamma'' \subset \Gamma'$, $\Gamma'' \not\models_C \alpha$. Entonces, por el teorema 18, $A \cup \{\Sigma\} \vdash_\delta \Gamma'$. ■

Corolario 21. *Para toda δ -cláusula C -satisfacible Σ , y cada forma δ -clausal A y literales $\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n, n \geq 0$, si $A \cup \{\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}\} \vdash_\delta \Sigma$, entonces hay una δ -cláusula $\Sigma' \subseteq \Sigma$ tal que $A \vdash_\delta \Sigma'$.*

Prueba: Por el teorema 16, $\Sigma \models A \cup \{\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}\}$, y por lo tanto $\Sigma \models_C A \cup \{\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}\}$, pero como $\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}$ no es C -satisfacible, por las definiciones 1 y 2, $A \cup \{\{\lambda, \neg\lambda, \gamma_1, \dots, \gamma_n\}\}$ es C -equivalente a A , con lo que $\Sigma \models_C A$. Sea α una fórmula C -equivalente a A . Entonces, $\Sigma \models_C \alpha$. Sea $\Sigma' \subseteq \Sigma$ una δ -cláusula tal que $\Sigma' \models_C \alpha$ pero para toda $\Sigma'' \subset \Sigma'$, $\Sigma'' \not\models_C \alpha$. Entonces, por el teorema 18, $A \vdash_\delta \Sigma'$. ■

Definición 22 (saturación por δ -resolución). *Dada la forma δ -clausal A , el conjunto saturación por δ -resolución de A , que llamamos A^δ , es el más pequeño que contiene cada δ -cláusula Σ tal que:*

- $A \vdash_\delta \Sigma$.
- Σ es satisfacible.
- No existe ninguna $\Sigma' \subset \Sigma$ tal que $A \vdash_\delta \Sigma'$.

En el siguiente corolario probamos que A^δ es el conjunto de las δ -cláusulas minimales de las que cierta fórmula es C -consecuencia.

Este conjunto podrá obtenerse siempre en un número finito de pasos para conjuntos finitos de δ -cláusulas A . Basta con aplicar δ -resolución hasta que no se genera ninguna δ -cláusula nueva que no esté subsumida por alguna de las anteriores, y eliminar del conjunto resultante todas las δ -cláusulas subsumidas o no C -satisfacibles. Sin embargo, dada la complejidad del proceso, cobra sentido el empleo de estrategias que hagan la búsqueda más eficiente, como es habitual en la resolución estándar. Por ejemplo, los corolarios 20 y 21 nos permiten eliminar las δ -cláusulas subsumidas o contradictorias en cualquier momento, sin esperar al final del proceso.

Corolario 23. *Sea A una forma δ -clausal C -equivalente a α , para cierto conjunto C de constantes que contiene todos los términos de A y de α . Entonces, A^δ es el conjunto de δ -cláusulas Σ tales que:*

- $\Sigma \models_C \alpha$.
- Σ es C -satisfacible.
- Para toda $\Sigma' \subset \Sigma$, $\Sigma' \not\models_C \alpha$.

Prueba: El teorema 18 asegura que cada Σ que tenga las propiedades del enunciado de este corolario es demostrable mediante δ -resolución a partir de A . Por lo tanto, por la definición 22, Σ debe pertenecer a A^δ , puesto que si Σ es C -satisfacible también es satisfacible y no hay ninguna $\Sigma' \subset \Sigma$ tal que $A \vdash_\delta \Sigma'$ (pues entonces, por el teorema 16, $\Sigma' \models A$, y también $\Sigma' \models_C A$, lo que contradice las propiedades de Σ). Por lo tanto, cada δ -cláusula con las propiedades de Σ pertenece a A^δ . Probemos ahora que toda δ -cláusula Σ de A^δ tiene las propiedades que enuncia este corolario. En primer lugar, como $A \vdash_\delta \Sigma$, entonces $\Sigma \models A$ (teorema 16) y por tanto $\Sigma \models_C A$ y $\Sigma \models_C \alpha$. Además, por la definición 22 tenemos que Σ es satisfacible, lo que implica que no tiene literales complementarios. Como además todos los términos de Σ son constantes de C , por el lema 9 sabemos que Σ es C -satisfacible. Por último, si fuera el caso de que existiera una δ -cláusula $\Sigma' \subset \Sigma$ tal que $\Sigma' \models_C \alpha$, entonces habría una δ -cláusula $\Sigma'' \subset \Sigma$ tal que $\Sigma'' \models_C \alpha$ y para toda $\Sigma^* \subset \Sigma''$, $\Sigma^* \not\models_C \alpha$. Pero entonces, por el teorema 18, $A \vdash_\delta \Sigma''$, lo que contradice lo que la definición 22 afirma sobre Σ . Por consiguiente, Σ tiene todas las propiedades que indica el teorema. ■

4. *Obtención de soluciones C-abductivas*

El teorema 24 contiene el resultado principal de este trabajo. Dado cierto problema abductivo $\langle \Theta, \varphi \rangle$, si tenemos las formas δ -cláusulas C -equivalentes a la negación de Θ y de φ , resultará posible obtener todas las soluciones C -abductivas a partir de operaciones de δ -resolución exclusivamente. Recordemos que el teorema 13 nos daba un procedimiento para obtener formas δ -cláusulas C -equivalentes a cualquier fórmula a través de las C -tablas. Estos dos resultados serán la base que nos permitirá definir el proceso abductivo que proporcionamos en la siguiente sección.

Teorema 24. *Dado el problema abductivo $\langle \{\theta_1, \dots, \theta_n\}, \varphi \rangle$, si N_Θ y O son formas δ -clausales C -equivalentes, respectivamente, a $\neg(\theta_1 \wedge \dots \wedge \theta_n)$ y φ , y C contiene todos los términos que aparecen en N_Θ y O , entonces:*

$$Abd(\Theta, \varphi)_C = \left(N_\Theta^\delta \cup O^\delta \right)^\delta - \left(N_\Theta^\delta \cup O^\delta \right)$$

Prueba: Tomando $\Theta = \{\theta_1, \dots, \theta_n\}$, $\Sigma \in Abd(\Theta, \varphi)_C$ equivale, según la definición 6, a que se verifica:

1. $\Theta, \Sigma \models_C \varphi$. Pero esto equivale a $\Sigma \models_C \neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$ y a $\Sigma \models_C N_\Theta \cup O$, por ser N_Θ y O , respectivamente, C -equivalentes a $\neg(\theta_1 \wedge \dots \wedge \theta_n)$ y φ .
2. $\Theta \cup \Sigma$ es C -satisfacible, que equivale a $\Sigma \not\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n)$. Desde aquí también se infiere que Σ es C -satisfacible.
3. $\Sigma \not\models_C \varphi$.
4. No existe ninguna $\Sigma' \subset \Sigma$ tal que $\Theta, \Sigma' \models_C \varphi$. Esto equivale a que para toda $\Sigma' \subset \Sigma$, $\Sigma' \not\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$.

Por lo tanto, comenzaremos suponiendo $\Sigma \in Abd(\Theta, \varphi)_C$ y dado que entonces se verifican los puntos de la definición anterior, trataremos de obtener con ellos que $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta$ y $\Sigma \notin (N_\Theta^\delta \cup O^\delta)$. A continuación supondremos que $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta - (N_\Theta^\delta \cup O^\delta)$ y probaremos que se verifican los cuatro puntos de la enumeración anterior, por lo que $\Sigma \in Abd(\Theta, \varphi)_C$.

En primer lugar, consideremos que $\Sigma \in Abd(\Theta, \varphi)_C$. Entonces:

- Desde 1, 2 y 4 (enumeración anterior), teniendo en cuenta que $N_\Theta \cup O$ es C -equivalente a $\neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$, obtenemos, por el corolario 23, $\Sigma \in (N_\Theta \cup O)^\delta$, y por lo tanto $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta$ (el orden de aplicación de la regla de δ -resolución en un conjunto A no altera el resultado A^δ).
- Desde 2 y 3 llegamos, por el corolario 23, a $\Sigma \notin N_\Theta^\delta$ y $\Sigma \notin O^\delta$, respectivamente, con lo que $\Sigma \notin (N_\Theta^\delta \cup O^\delta)$.

De los resultados obtenidos concluimos $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta - (N_\Theta^\delta \cup O^\delta)$. Ahora, supongamos esto último, para probar $\Sigma \in \mathcal{A}bd(\Theta, \varphi)_C$. Tenemos:

- $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta$. Como se ha observado, esto es equivalente a $\Sigma \in (N_\Theta \cup O)^\delta$, y por ser C -equivalentes $N_\Theta \cup O$ y $\neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$, por el corolario 23, $\Sigma \models_C \neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$ y para toda $\Sigma' \subset \Sigma$, $\Sigma' \not\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$.
- $\Sigma \notin N_\Theta^\delta$. Supongamos que $\Sigma \models_C \neg(\theta_1 \wedge \dots \wedge \theta_n)$. Entonces existirá una $\Sigma' \subseteq \Sigma$ tal que $\Sigma' \models_C \neg(\theta_1 \wedge \dots \wedge \theta_n)$ y para toda $\Sigma^* \subset \Sigma'$, $\Sigma^* \not\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n)$. Pero entonces, puesto que Σ es C -satisfacible (lema 9), por el corolario 23 tendríamos que $\Sigma' \in N_\Theta^\delta$, con lo que no puede ser que $\Sigma' = \Sigma$, pero tampoco que $\Sigma' \subset \Sigma$, ya que esto contradice $\Sigma \in (N_\Theta^\delta \cup O^\delta)^\delta$, puesto que Σ queda subsumida por Σ' . Por lo tanto, $\Sigma \not\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n)$.
- $\Sigma \notin O^\delta$. En este caso, mediante un razonamiento por reducción al absurdo, paralelo al del apartado anterior, concluimos que $\Sigma \not\models_C \varphi$.

Los resultados obtenidos son los mismos que al comienzo de la demostración comentamos que equivalen a $\Sigma \in \mathcal{A}bd(\Theta, \varphi)$. ■

Corolario 25. Sean $\alpha, \beta \in \mathcal{L}$, C -equivalentes, respectivamente, a las formas δ -clausales A y B , de modo que cada término de A y B pertenece a C . Entonces $\alpha \models_C \beta$ si y solo si para cada $\Sigma \in A^\delta$ existe una $\Sigma' \in B^\delta$ tal que $\Sigma' \subseteq \Sigma$.

Prueba: Supongamos que $\alpha \models_C \beta$ y $\Sigma \in A^\delta$. Entonces, por el corolario 23, Σ es C -satisfacible y $\Sigma \models_C \alpha$. Pero $\alpha \models_C \beta$, y por lo tanto $\Sigma \models_C \beta$. Por ello, existe una δ -cláusula $\Sigma' \subseteq \Sigma$ tal que

$\Sigma' \models_C \beta$, Σ' es satisficible y para toda $\Sigma^* \subset \Sigma'$, $\Sigma^* \not\models_C \beta$. Entonces, por el corolario 23, $\Sigma' \in B^\delta$.

Ahora, supongamos que para cada $\Sigma \in A^\delta$ existe una $\Sigma' \in B^\delta$ tal que $\Sigma' \subseteq \Sigma$. Probaremos que $\alpha \models_C \beta$. Sea \mathcal{M} una C -estructura tal que $\mathcal{M} \models \alpha$. Entonces, por el lema 12, $\mathcal{T}(\{\alpha\})_C$ tiene al menos una rama abierta cuyos literales son satisfechos por \mathcal{M} . Sea Σ^* la δ -cláusula compuesta por tales literales. Por el lema 11, $\Sigma^* \models_C \alpha$. Entonces existe una δ -cláusula $\Sigma \subseteq \Sigma^*$ tal que $\Sigma \models_C \alpha$ y para cada $\Sigma'' \subset \Sigma$, $\Sigma'' \not\models_C \alpha$. En consecuencia, por el corolario 23, $\Sigma \in A^\delta$. Entonces, existe una $\Sigma' \subseteq \Sigma$ tal que $\Sigma' \in B^\delta$. Por el corolario 23, $\Sigma' \models_C \beta$. Entonces, como $\mathcal{M} \models \Sigma'$ (ya que $\Sigma' \subseteq \Sigma \subseteq \Sigma^*$ y $\mathcal{M} \models \Sigma^*$), $\mathcal{M} \models \beta$. Por lo tanto, $\alpha \models_C \beta$. ■

Presentaremos ahora el procedimiento para la obtención de soluciones C -abductivas mediante la combinación de C -tablas y δ -resolución. Las dos operaciones principales que se realizarán son, por una parte, la obtención de formas δ -cláusulas a partir de C -tablas, y, por la otra, la saturación mediante δ -resolución. En el corolario 27 demostramos la corrección del proceso a partir de los teoremas de las secciones previas.

Definición 26 (proceso abductivo). *Definimos el proceso de búsqueda de soluciones C -abductivas de $\langle \Theta, \varphi \rangle$, con $\Theta = \{\theta_1, \dots, \theta_n\}$, donde $\Theta \subset \mathcal{L}$, $\varphi \in \mathcal{L}$, y todos los términos de Θ y φ están contenidos en el conjunto de constantes C , de la siguiente forma:*

Paso 1: Análisis de la teoría. *Se obtiene $N_\Theta = C\delta(\mathcal{T}(\{\neg(\theta_1 \wedge \dots \wedge \theta_n)\})_C)$. Es decir, N_Θ será la forma δ -clausal de la C -tabla de $\neg(\theta_1 \wedge \dots \wedge \theta_n)$. Entonces:*

- Si $N_\Theta = \emptyset$, entonces Θ es C -válida, y el proceso acaba.
- Si no, se obtiene N_Θ^δ mediante saturación por δ -resolución. Si $\diamond \in N_\Theta^\delta$, entonces Θ no es C -satisficible, y el proceso acaba. En otro caso,

Paso 2: Análisis de la observación. *Sea $O = C\delta(\mathcal{T}(\{\varphi\})_C)$. Como en el caso anterior, O será la forma δ -clausal obtenida a partir de la C -tabla de φ . Entonces:*

- Si $O = \emptyset$, entonces φ no es C -satisficible, y el proceso acaba.

- Si no, se obtiene O^δ . Si $\diamond \in O^\delta$, entonces φ es C -válida, y el proceso acaba. En otro caso,

Paso 3: Búsqueda de refutaciones. Si para cada δ -cláusula $\Sigma \in O^\delta$ existe una $\Sigma' \in N_\Theta^\delta$ tal que $\Sigma' \subseteq \Sigma$, entonces $\Theta \models_C \neg\varphi$, y el proceso termina. En otro caso,

Paso 4: Búsqueda de explicaciones. Desde N_Θ^δ y O^δ , se obtienen $(N_\Theta^\delta \cup O^\delta)$ y luego $(N_\Theta^\delta \cup O^\delta)^\delta$. Entonces, si $\diamond \in (N_\Theta^\delta \cup O^\delta)^\delta$, $\Theta \models_C \varphi$ y el proceso acaba. En otro caso, $\langle \Theta, \varphi \rangle$ es un problema abductivo, y el proceso devuelve:

$$Abd(\Theta, \varphi)_C = \left(N_\Theta^\delta \cup O^\delta \right)^\delta - \left(N_\Theta^\delta \cup O^\delta \right)$$

Corolario 27. Para cualesquiera $\Theta \subset \mathcal{L}$ y $\varphi \in \mathcal{L}$, el proceso de búsqueda de soluciones C -abductivas (para cualquier conjunto C de constantes que contenga todos los términos de Θ y φ) que sigue la definición 26 para $\langle \Theta, \varphi \rangle$ es correcto.

Prueba: Dada una fórmula $\alpha \in \mathcal{L}$ que sea C -equivalente a la forma δ -clausal A , $\diamond \in A^\delta$ $\text{syss} \models_C \alpha$ (por los teoremas 16 y 17). Igualmente, α no es C -satisfacible $\text{syss } \mathcal{C}\delta(\mathcal{T}(\{\alpha\})_C)$ tampoco lo es (teorema 13), es decir, $\text{syss } \mathcal{C}\delta(\mathcal{T}(\{\alpha\})_C) = \emptyset$ (puesto que, por el corolario 9 y el lema 11, $\mathcal{T}(\{\alpha\})_C$ no tendrá ramas abiertas). Con estas observaciones, la prueba de corrección de los pasos 1 y 2 de la definición 26 es directa. El paso 3 es una consecuencia del corolario 25. Para el paso 4, si $\diamond \in (N_\Theta^\delta \cup O^\delta)^\delta$, entonces $N_\Theta^\delta \cup O^\delta$ es C -válida (teorema 16). Pero esta forma δ -clausal es C -equivalente a $\neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$, por lo que $\models_C \neg(\theta_1 \wedge \dots \wedge \theta_n) \vee \varphi$, lo cual equivale a $\Theta \models_C \varphi$. En otro caso, la forma en que el paso 4 construye $Abd(\Theta, \varphi)_C$ es correcta, por el teorema 24. ■

5. Conclusiones

Aunque el problema de la abducción en primer orden es, por lo general, indecible, cobra sentido la búsqueda de soluciones abductivas en estructuras de cardinalidad finita, como son las C -estructuras, dado que muchos de los ámbitos que actualmente requieren una automatización del razonamiento abductivo pueden caracterizarse a través de estructuras con dominios finitos.

El procedimiento presentado en la definición 26 devuelve, para cierto problema abductivo $\langle \Theta, \varphi \rangle$, todas las soluciones C -abductivas que verifican los requisitos de la definición 6. Téngase en cuenta que estos criterios corresponden con lo que muchos autores como Aliseda (2006) consideran que es la forma de razonamiento abductivo más interesante desde el punto de vista de sus aplicaciones, especialmente en lo que se refiere a la filosofía de la ciencia y la epistemología.

En Soler y Nepomuceno 2006, hemos presentado un sistema, escrito en Prolog, basado igualmente en tablas semánticas para modelos finitos y resolución dual que muestra que el uso combinado de estos dos formalismos no sólo es más eficiente que su uso por separado (tal como se hace en Cialdea Mayer y Pirri 1993, al desarrollar todo un proceso abductivo exclusivamente mediante tablas semánticas, o como lo hacen los acercamientos tradicionalmente más próximos a la Programación Lógica (Kakas *et al.* 1998) con la resolución), sino que, a pesar de reducir a modelos finitos las teorías en primer orden, son numerosas y variadas las aplicaciones que encuentra.

BIBLIOGRAFÍA

- Aliseda, A., 2006, *Abductive Reasoning: Logical Investigations into Discovery and Explanation*, Springer, Dordrecht.
- , 1997, *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence*, Universidad de Amsterdam, Amsterdam.
- Beth, E.W., 1955, “Semantic Entailment and Formal Derivability”, *Koninklijke Nederlandse Akademie van Wetenschappen, Proceedings of the Section of Sciences*, vol. 18, pp. 309–342.
- Boalos, G., 1984, “Trees and Finite Satisfiability”, *Notre Dame Journal of Formal Logic*, vol. 25, pp. 110–115.
- Carnielli, W., 2006, “Surviving Abduction”, *Logic Journal of the IGPL*, vol. 14, no. 2, pp. 237–256.
- Cialdea Mayer, M. y F. Pirri, 1993, “First Order Abduction Via Tableau and Sequent Calculi”, *Bulletin of the IGPL*, vol. 1, pp. 99–117.
- Díaz, E., 1993, “Árboles semánticos y modelos mínimos”, en *Actas del I Congreso de la Sociedad de Lógica, Metodología y Filosofía de la Ciencia en España*, Universidad Complutense de Madrid, Madrid.
- Kakas, A., R. Kowalski y F. Toni, 1998, “The Role of Abduction in Logic Programming”, en D.M. Gabbay, C.J. Hogger y J.A. Robinson (comps.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 5, Oxford University Press, Oxford, pp. 235–324.
- Nepomuceno, A., 2002, “Scientific Explanation and Modified Semantic Tableaux”, en L. Magnani, N.J. Nersessian y C. Pizzi (comps.), *Logical and*

- Computational Aspects of Model-Based Reasoning*, Kluwer, Dordrecht, pp. 181–198.
- Quine, W.V.O., 1955, “A Way to Simplify Truth Functions”, *The American Mathematical Monthly*, vol. 62, pp. 627–631.
- Reyes, A.L., A. Aliseda y A. Nepomuceno, 2006, “Abductive Reasoning in First Order Logic”, *Logic Journal of the IGPL*, vol. 14, no. 2, pp. 287–304.
- Robinson, J.A., 1965, “A Machine-Oriented Logic Based on the Resolution Principle”, *Journal of the ACM*, vol. 12, pp. 23–41.
- Soler, F. y A. Nepomuceno, 2006, “TARFA: Tableaux and Resolution for Finite Abduction”, *Lecture Notes in Computer Science*, vol. 4160, pp. 511–514.
- Soler, F., A. Nepomuceno y A. Aliseda, 2006, “Model-Based Abduction Via Dual Resolution”, *Logic Journal of the IGPL*, vol. 14, no. 2, pp. 305–319.

Recibido el 14 de junio de 2007; aceptado el 4 de abril de 2008.