UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE MATEMÁTICAS

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y PROGRAMACIÓN

MÉTODOS DE TABLEAUX PARA LÓGICAS CON DECLARACIONES DE TÉRMINOS, DOMINIOS PREORDENADOS Y OPERACIONES MONÓTONAS

PEDRO JESÚS MARTÍN DE LA CALLE

MADRID, Mayo 2000

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE MATEMÁTICAS
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y PROGRAMACIÓN

MÉTODOS DE TABLEAUX PARA LÓGICAS CON DECLARACIONES DE TÉRMINOS, DOMINIOS PREORDENADOS Y OPERACIONES MONÓTONAS

Memoria que para optar al grado de
DOCTOR EN MATEMÁTICAS
presenta

PEDRO JESÚS MARTÍN DE LA CALLE

DIRECTOR: ANTONIO GAVILANES FRANCO

Madrid, Mayo 2000

Agradecimientos

"*¡Joder, a ver si para de llover!, ¡trabajando todo el día y encima no ves el sol!*" pensaba a menudo durante la elaboración de esta, *mi tesis*. Ahora que está acabada, rondan por la cabeza muchas preguntas, quizá demasiadas. Por una parte, en una tesis tan teórica, uno duda de la relevancia de sus resultados; quiero creer que el trabajo no sólo es novedoso, de lo cual estoy seguro, sino que su incidencia será significativa, o al menos interesante, y su aplicación, inmediata en el progreso de la informática. Además, uno no sabe cuándo detener el proceso de investigación, y ponerse a escribir este gran resumen. En este sentido, entiendo que la tesis es sólo un trámite formal que acredita el esfuerzo realizado, aunque sienta que podía haber redondeado más los temas, llegando a obtener más resultados.

Por otra parte, uno se contagia de la alegría que supone el haber desarrollado una investigación seria, de forma continua, durante tanto tiempo, y que detrás quede una labor dura *bien hecha*. Resulta un orgullo no sólo llegar hasta aquí, sino el haberlo intentado con tanta ilusión. Ojalá el tiempo depare momentos tan gozosos como éste...

Quiero agradecer a todos los que han ayudado en la elaboración de esta tesis, tanto directa como indirectamente. Entre los que han intervenido de forma explícita, debo agradecer en primer lugar la colaboración de mi *jefe* Antonio, no sólo por su dedicación, apoyo, consejos y orientación técnica -a veces comparaba nuestras improvisadas reuniones con una visita al siquiatra-, sino por su valiosa amistad. Esta tesis no existiría como tal sin su dirección. También agradezco profundamente la colaboración especial de Susana y Javier por sus provechosos comentarios y sus consejos. . .

Otros compañeros del departamento han hecho que me sintiera como en casa durante mi estancia aquí; así agradezco el buen rollo de mis amigos Puri, Jesús, Ana, Jaime, Rafa, Eva...

Otras personas que nada tienen que ver con mi trabajo en la Facultad han colaborado indirectamente durante estos años. En primer lugar Conchi, que me ha aguantado y animado desde siempre, y con la que también he celebrado los buenos momentos. Mi familia, Pedro, Lola, María y Carlos, también me ha apoyado incondicionalmente y por eso les doy las gracias que se merecen. A Conchi y a mis padres dedico esta tesis.

También agradezco a Nieves y Malicia por dejarme saborear la música de forma tan intensa y a los chicos del Kimura por sus buenos momentos de disfrute. Gracias a mis amigos Chuchi y Manolito Delgado, *el bueno*, a mis abuelas Teresa y *Agustín*, a mis primos Víctor, Laura y Pablo, a mis gatos Rin-Ran, Galois, Clamoxyl, Frenadol, Malena, Ariel y Grillo, y por último a mi *Villamanrique*, el lugar donde todo es posible...

Gracias

Índice

Capítulo 1

Introducción

Como su título indica, el objetivo de esta tesis consiste en el desarrollo de demostradores automáticos basados en tableaux para determinadas extensiones de la lógica clásica de predicados. En concreto, estudiamos la incorporación, en la lógica de primer orden, de declaraciones de términos y preórdenes monótonos. A continuación presentamos una pequeña introducción de cada uno de los elementos que configuran este trabajo.

El método de tableaux

Una lógica puede caracterizarse de distintas formas. Tradicionalmente se ha hecho mediante alguna de las tres aproximaciones siguientes:

1. Una motivación intuitiva, quizá ligada a alguna aplicación en algún área.

2. Una interpretación semántica.

3. Un sistema formal de pruebas.

De entre todas las metodologías existentes en el campo de la demostración automática, hemos elegido los *métodos de tableaux* porque reúnen, en un único marco, las características propias de las aproximaciones orientadas al estudio de pruebas (3) y aquellas fundamentadas semánticamente (2). Además, los tableaux presentan los sistemas lógicos de una forma intuitiva (1), clara y concisa, por lo que actualmente suelen ser utilizados como el primer mecanismo deductivo que aprenden los estudiantes de lógica en la mayoría de las universidades. Por otra parte, en los últimos años se han desarrollado gran cantidad de diversos sistemas lógicos, originados por multitud de aplicaciones, para los que la metodología basada en tableaux resulta ser el mecanismo apropiado, debido a que se adaptan perfectamente a las distintas exigencias específicas.

Por estos motivos, los tableaux se han extendido ampliamente y son el foco de atención de una gran cantidad de investigadores.

CARACTERÍSTICAS DEL MÉTODO DE TABLEAUX

Bajo la denominación *método de tableaux* se encuentran distintos sistemas deductivos que comparten características comunes:

1. Los sistemas basados en tableaux son procedimientos formales sintácticos que generan pruebas por refutación. Esto significa que para probar que una fórmula φ es válida, empezamos con alguna expresión sintáctica $\overline{\varphi}$ que establezca que no lo es. Obviamente, esta expresión depende de la lógica sobre la que trabajemos. Hecho esto, descomponemos sintácticamente $\overline{\varphi}$, distinguiendo casos cuando sea necesario. Esta parte del procedimiento de tableaux *-etapa de expansión-* puede entenderse como una generalización del proceso de transformación a forma normal disyuntiva. Una característica propia de los métodos de tableaux es el *principio de subfórmula*: "las fórmulas introducidas en esta etapa son subfórmulas de las ya existentes".

   Finalmente, estos métodos disponen de reglas para cerrar aquellos casos que impliquen condiciones sintácticas imposibles de satisfacer. Si todos los casos están cerrados, el tableau está cerrado y tenemos una prueba de que la fórmula $\overline{\varphi}$ no es satisfactible.

2. Desde el punto de vista semántico, un tableau representa la búsqueda de un modelo que verifique ciertas condiciones. En este sentido, cada rama abierta de un tableau puede considerarse como una descripción parcial de un modelo. De hecho, si estas ramas satisfacen determinadas propiedades de saturación entonces definen un modelo de la fórmula inicial $\overline{\varphi}$. Por este motivo, los métodos de tableaux pueden usarse para generar contraejemplos, ya que cualquier modelo de $\overline{\varphi}$ es un contraejemplo de la validez de φ.

Para profundizar en las nociones anteriores, necesitamos mecanismos sintácticos para establecer que una fórmula no es válida, analizar los distintos casos y cerrar aquellos casos imposibles. Dado que estos mecanismos dependen de la lógica en cuestión, usamos la lógica proposicional como ejemplo.

Resulta obvio que la negación lógica $\neg\varphi$ es el mecanismo natural para expresar que φ no es válida. Para presentar de un modo sencillo los mecanismos que descomponen sintácticamente una fórmula y analizan los casos correspondientes suponemos, sin pérdida de generalidad, que los únicos conectivos admitidos son la negación $\neg$ y la conjunción $\wedge$. Las reglas de expansión descomponen toda fórmula que no sea una variable proposicional o su negación. Por este motivo sólo necesitamos tres reglas:

doble negación $\dfrac{\neg\neg\varphi}{\varphi}$

conjunción $\dfrac{\varphi \wedge \psi}{\begin{array}{c}\varphi \\ \psi\end{array}}$

conjunción negada $\dfrac{\neg(\varphi \wedge \psi)}{\neg\varphi \mid \neg\psi}$

Estas reglas han de entenderse como sigue: si la premisa es válida, entonces la conclusión también lo es. En este sentido, las reglas de la doble negación y de la conjunción resultan sencillas. Para el caso de la *conjunción negada* sabemos, por las tablas veritativas, que si $\neg(\varphi \wedge \psi)$ es cierta, entonces $\neg\varphi$ o $\neg\psi$ son ciertas, por lo que bifurcamos en dos casos.

La representación habitual de los tableaux consiste en un árbol cuyos nodos son fórmulas y que se prolonga o bifurca mediante la aplicación de las reglas anteriores. De esta forma la representación de un tableau varía con el tiempo; de hecho deberíamos hablar de secuencias de tableaux. Desde el punto de vista semático, debemos entender una rama como la conjunción de todas las fórmulas que contiene y a un tableau, como la disyunción de sus ramas. Obsérvese que un nodo puede pertenecer a distintas ramas aunque sólo aparezca escrito una única vez. En este sentido, los tableaux no son redundantes pues reutilizan información en lugar de duplicarla.

Una característica propia de los sistemas de tableaux, y en concreto del que nos venimos ocupando, es el indeterminismo con el que aplicamos las reglas de expansión. Es cierto que se pueden emplear distintas estrategias para guiar la expansión, pero éstas no forman parte de la propia metodología basada en tableaux.

En la Figura 1.1 aparece una expansión que comienza por la fórmula $\varphi = p \wedge \neg(p \wedge \neg(q \wedge \neg(p \wedge q)))$.

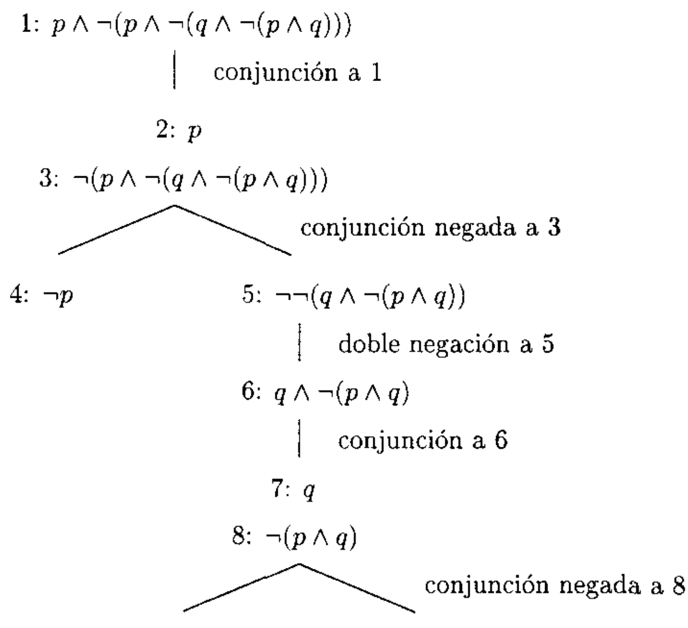

Figura 1.1: Expansión para φ.

Finalmente, las condiciones sintácticas que cierran una rama son sencillas: una rama está cerrada si contiene φ y $\neg\varphi$, para alguna fórmula φ. De esta forma, el tableau de la

Figura 1.1 está cerrado ya que sus cuatro ramas lo están.

En ocasiones puede ocurrir que no exista un tableau cerrado, por lo que podemos usar el método de tableaux como un generador de contraejemplos. Así, si intentamos probar la validez de $\psi = \neg(p \wedge \neg(q \wedge p))$ podemos construir el tableau de la Figura 1.2. Este tableau no está cerrado, ya que la rama de la izquierda no lo está, y su etapa de expansión ha concluido. De hecho, la etapa de expansión para cualquier fórmula de la lógica proposicional resulta ser un proceso finito debido a que podemos descartar las fórmulas que ya han sido usadas. Por este motivo, decimos que la lógica proposicional es decidible. Volviendo al ejemplo, podemos usar las fórmulas de la rama abierta para construir un modelo de la fórmula 1 y, por tanto, un contraejemplo de ψ. En efecto, usamos 3 y 5 para definir la valoración $v(p) = \mathit{true}$, $v(q) = \mathit{false}$. Obviamente v satisface 3 y 5, por lo que, retrocediendo en la construción del tableau, también es modelo de 4, 2 y 1.

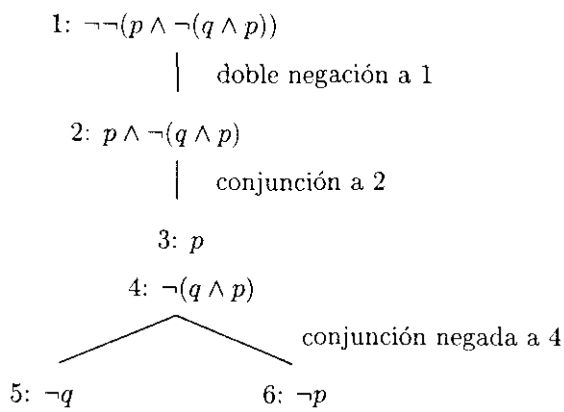

Figura 1.2: Expansión para ψ.

No todos los métodos de tableaux siguen una presentación basada en árboles como la que acabamos de ver en nuestro ejemplo para la lógica proposicional. Desde el punto de vista metodológico, e independientemente de la lógica escogida, todos estos sistemas comparten las características anteriores, aunque empleen distintas notaciones: Manna y Waldinger usan una notación distinta para su particular versión de los tableaux [MW 90] [MW 93], el método de conexión puede entenderse como un sistema basado en tableaux en el que los árboles se reemplazan por grafos más generales [BKAHS 87], etc. De hecho, la presentación original basada en árboles difiere de la que hoy consideramos habitual.

Ventajas del método de tableaux

Desde el punto de vista práctico, los tableaux entrañan un indeterminismo *acotado* ya que satisfacen el principio de subfórmula, es decir, las fórmulas introducidas son subfórmulas de las ya existentes. De esta forma, en cada nueva expansión optamos entre un número finito de posibilidades y la complejidad de las fórmulas no aumenta. Esto no ocurre, por ejemplo, en los antiguos sistemas axiomáticos del tipo Hilbert debido al *modus ponens* (para probar φ buscamos cualquier fórmula ψ tal que $\psi \rightarrow \varphi$).

En el caso proposicional, este indeterminismo no influye dado que si construimos los tableaux de forma *justa*, todas las posibles construcciones concluirán en un tableau cerrado, si alguna de ellas lo hace.

En el caso de la lógica de primer orden, los cuantificadores universales introducen un nuevo problema debido a que hay infinitas formas de aplicar la regla de expansión correspondiente. Para solucionar este problema eficientemente, los tableaux hacen uso de las variables libres y de la unificación sintáctica. No obstante, como las fórmulas cuantificadas universalmente no pueden descartarse una vez usadas, el procedimiento de expansión no acaba y el método de tableaux resulta ser un procedimiento de semidecisión (acorde con la indecidibilidad de primer orden). Todas estas razones hacen de los tableaux un mecanismo adecuado para su implementación.

Desde el punto de vista semántico, los tableaux destacan como método pedagógico ideal para iniciarse en el mundo de la lógica. Por otra parte, la cohesión entre las reglas sintácticas del método y la semántica asociada a la lógica en cuestión, hace que los tableaux sean atractivos para una gran mayoría de los investigadores dedicados al estudio de nuevas lógicas.

HISTORIA DEL MÉTODO DE TABLEAUX

La historia de los métodos de tableaux se inicia con Gentzen en 1935, año en el que publicó su famoso cálculo de secuentes [Gent 35]. Este cálculo, orientado a la generación de pruebas formales, separa por primera vez las propiedades estructurales deductivas de la lógica, de las reglas de conectivos y cuantificadores. Menos su *regla de corte* (¡que resultó eliminable!), todas sus reglas parten de premisas más sencillas que la conclusión por lo que, en cierto sentido, satisfacen el principio de subfórmula común a todos los métodos de tableaux. De hecho, éste es el primer cálculo que se ajusta a este principio, puesto que los anteriores sistemas axiomáticos de tipo Hilbert disponían de la regla del modus ponens. Por otra parte, si razonamos hacia atrás sobre las pruebas que el cálculo de Gentzen construye, podemos comparar tales pruebas con refutaciones basadas en tableaux que usan una representación del tipo *conjunto de conjuntos*.

Después de Gentzen, aparecieron los trabajos de Beth y Hintikka en los años 50, que estuvieron movidos por cuestiones semánticas en vez de por razones sintácticas (construcción de pruebas formales) como las que guiaron a su predecesor Gentzen. En 1955 Beth introdujo el término *tableau semántico* por primera vez, y pensó en los tableaux como métodos que trataban de encontrar contraejemplos de forma sistemática [Beth 55]. La representación que usó separaba las fórmulas negadas de las no negadas distribuyéndolas en dos columnas.

Los estudios de Hintikka coincidieron con los de Beth en espíritu semántico y en época. Así, en 1955, aparece su trabajo más relevante [Hint 55], en el que considera los tableaux como mecanismos de refutación que buscan contraejemplos. La metodología usada en este artículo para demostrar la completitud separa las propiedades abstractas de satisfactibilidad, de los detalles específicos del proceso de construcción de los tableaux. Estas ideas han perdurado hasta nuestros días debido a que la prueba de completitud resulta sencilla, si usamos los conocidos *conjuntos de Hintikka*. La representación usada por Hintikka es la de un árbol cuyos nodos son conjuntos de fórmulas. Como los nodos pueden compartir fórmulas, esta representación resulta redundante, pues la aparición de una fórmula puede repetirse innecesariamente.

La representación actual de los tableaux, como árboles cuyos nodos están etiquetados por fórmulas, fue introducida por Smullyan. Tras otros trabajos previos, en 1968 Smullyan publica su libro *First-Order Logic* [Smul 68], al que debemos la gran difusión que los tableaux han experimentado desde entonces. La presentación de las reglas de expansión se convirtió en una tarea cómoda a partir de que introdujera su famosa *Notación Uniforme*, consistente en una partición del conjunto de todas las fórmulas. Por primera vez Smullyan maneja el término *tableau análitico* para expresar que la principal característica común a todos estos métodos es el principio de subfórmula.

Tras Smullyan, Fitting ha afianzado las bases del método de tableaux en publicaciones más recientes [Fitt 96], siendo este libro constante objeto de referencia.

MÉTODOS DE TABLEAUX PARA LÓGICAS NO CLÁSICAS

En la mayoría de los casos, tras el diseño de métodos de tableaux para las lógicas clásicas de proposiciones y predicados, los investigadores se preguntaron qué pasaría si las propiedades deductivas de la lógica fueran modificadas. De esta forma, el estudio de métodos de tableaux para estas nuevas lógicas ha venido desarrollándose desde los trabajos de Gentzen. Veamos algunos de los casos en los que los sistemas de tableaux se han adaptado cómodamente a estas modificaciones.

Lógica intuicionista. Para un matemático intuicionista necesitamos pruebas fehacientes para poder afirmar que una fórmula es válida. En este sentido, $\neg\varphi$ es cierta si no disponemos de una prueba fehaciente de φ, que no es lo mismo que tener una prueba de $\neg\varphi$. Esto repercute en las reglas de los otros conectivos, por ejemplo $\neg(\varphi \to \psi)$ es válida si no disponemos de una prueba de $\varphi \to \psi$, lo que a su vez significa que no tenemos un mecanismo que transforme pruebas de φ en pruebas de ψ. En esta situación, disponemos de una prueba de φ (φ es cierto) y ninguna de ψ ($\neg\psi$ es cierto). Habitualmente, se usan las etiquetas T y F para expresar la disponibilidad actual o no de una prueba fehaciente. Así la regla de expansión aplicable a la fórmula etiquetada $F : \varphi \to \psi$ prolonga la rama con $T : \varphi$ y $F : \psi$, pero como ahora disponemos de una prueba de φ, debemos modificar el resto de las fórmulas de la rama de acuerdo con esta información. Debido a que las pruebas resisten el paso del tiempo, respetamos todas las fórmulas positivas (etiquetadas con T); sin embargo, eliminamos todas las fórmulas negativas de la forma $F : \varphi'$, porque la falta de una prueba para φ' en el pasado puede paliarse ahora con la prueba de φ.

Los tableaux para la lógica intuicionista fueron introducidos por Gentzen [Gent 35] y Beth [Beth 56] [Beth 59], y luego tratados por Fitting [Fitt 69].

Lógicas multivaloradas. En este caso disponemos de un conjunto finito de valores semánticos ordenados y etiquetamos las fórmulas para expresar el valor veritativo que esperamos de ellas. Por ejemplo, en la lógica trivalorada fuerte de Kleene, disponemos del nuevo valor semántico $\perp$ para denotar lo indefinido, y usamos las etiquetas T, F y ? para denotar aquellas fórmulas que se evalúan a *true*, *false* y $\perp$, respectivamente. Además, en esta lógica, el orden entre los valores veritativos es $false < \perp < true$ y la semántica asociada, por ejemplo, al conectivo disyuntivo $\vee$ viene expresada por la siguiente tabla:

φ	$\underline{t}$	$\underline{t}$	$\underline{t}$	$\underline{f}$	$\underline{f}$	$\underline{f}$	$\perp$	$\perp$	$\perp$
ψ	$\underline{t}$	$\underline{f}$	$\perp$	$\underline{t}$	$\underline{f}$	$\perp$	$\underline{t}$	$\underline{f}$	$\perp$
$\varphi \vee \psi$	$\underline{t}$	$\underline{t}$	$\underline{t}$	$\underline{t}$	$\underline{f}$	$\perp$	$\underline{t}$	$\perp$	$\perp$

donde hemos abreviado *true* y *false* a $\underline{t}$ y $\underline{f}$, respectivamente.

Entonces la regla de expansión disyuntiva aplicada, por ejemplo, sobre la fórmula etiquetada ?: $\varphi \vee \psi$ debería bifurcar el tableau de la siguiente forma:

$$?\!: \varphi \vee \psi$$

$$\begin{array}{ccc} ?\!: \varphi & ?\!: \varphi & F\!: \varphi \\ F\!: \psi & ?\!: \psi & ?\!: \psi \end{array}$$

En efecto, si el valor semántico de $\varphi \vee \psi$ es $\perp$, entonces uno de esos tres casos debe darse, como se comprueba, de forma inmediata, examinando la tabla.

Los tableaux para lógicas multivaloradas aparecen por primera vez en [Such 74] y luego en [Surma 84], [Carn 87] y [Carn 91].

Lógicas modales y temporales. Fueron introducidas por Kanger [Kang 57], Hintikka [Hint 61] [Hint 62] y Kripke [Krip 59]. En ellas se usan dos nuevos operadores unarios, $\Box$ y $\Diamond$, que, aunque pueden tener distintas interpretaciones, tradicionalmente se entiende que expresan "*necesidad*" y "*posibilidad*", respectivamente. La aplicación de estos operadores sobre fórmulas permite establecer diversas propiedades temporales o modales. La veracidad de las mismas se consigue teniendo en cuenta *mundos* y una relación de accesibilidad S entre ellos. Entonces, la fórmula $\Box\varphi$ es cierta en un mundo ω, si para todo mundo ω' accesible desde ω, escrito $S(\omega, \omega')$, es cierto φ; análogamente, $\Diamond\varphi$ es cierta en un mundo ω, si existe un mundo ω' tal que $S(\omega, \omega')$, en el que φ es cierto. Por ejemplo, según esto, la validez de la fórmula $\Box\varphi \longrightarrow \Diamond\varphi$ expresa que para todo mundo ω existe otro ω' tal que $S(\omega, \omega')$.

Para construir tableaux modales/temporales podemos optar, además de considerar fórmulas con signo T, F, por etiquetarlas con mundos. Podemos entonces entender una regla como la siguiente:

$$\frac{F\omega : \ \Box\varphi}{F\omega' : \ \varphi}$$

con ella se permite extender una rama con la fórmula etiquetada de abajo, para algún mundo ω' tal que $S(\omega, \omega')$. La corrección de la regla se sigue de la interpretación del operador $\Box$ y el sentido de los mundos que acabamos de ver.

El tema de los tableaux para lógicas modales y temporales es amplísimo y puede consultarse una visión general del mismo en el capítulo correspondiente de [DGHP 99].

Implementación de métodos de tableaux

Antes de acabar el estudio de los métodos de tableaux para otras lógicas, ya se empezó a considerar su implementación. Resulta curioso comparar los objetivos iniciales de los tableaux con los de su gran competidor: *la resolución*. Aunque su introducción en el tiempo está próxima, Robinson en 1965 [Rob 65] diseñó la resolución como un mecanismo de demostración automática, mientras que el objetivo inicial de los tableaux era puramente semántico. Esta es la razón por la que, históricamente, se ha dedicado más atención a la resolución en el campo de la demostración automática.

La implementación de los tableaux para la lógica de primer orden presenta dificultades asociadas con el manejo de los cuantificadores. La regla de expansión existencial correspondiente a $\exists x \varphi(x)$ introduce en la rama la fórmula $\varphi(c)$, donde c es una constante nueva en la rama; mientras que la regla de expansión universal correspondiente a $\forall x \varphi(x)$ extiende la rama con $\varphi(t)$, para cualquier término t. De esta manera disponemos de un número infinito de posibles expansiones universales, por lo que la expansión de los tableaux resulta ser un proceso que no acaba. En esta situación, la implementación de los tableaux para la lógica de primer orden debía superar este difícil problema.

En 1960 Wang [Wang 60] construyó un sistema de decisión basado en tableaux para un subconjunto decidible de la lógica de primer orden, las llamadas *AE-fórmulas*. En estas fórmulas, los cuantificadores existenciales siempre preceden a los universales, por lo que la regla universal sólo necesita usar el conjunto finito de las constantes introducidas previamente por la regla existencial.

Más tarde, en [CTW 74], [Bowen 80] y [Broda 80] se introdujeron las variables libres y la unificación sintáctica de Robinson para evitar el indeterminismo incontrolado que se derivaba de la regla de expansión universal. Así, esta regla aplicada a $\forall x \varphi(x)$ introduce la fórmula $\varphi(y)$, donde y es una variable libre nueva que denota un elemento arbitrario. Estas variables se instancian convenientemente a la hora de cerrar el tableau sin más que aplicar un mecanismo de unificación sintáctica. Esta forma de proceder aparecerá abundantemente en este trabajo.

Los sistemas de tableaux para la lógica de primer orden no necesitan ningún tipo de precomputación adicional. No obstante, el proceso de *Skolemización* es reconocido como una herramienta interesante que se encarga de la eliminación de los cuantificadores existenciales. De hecho, este proceso debe incorporarse implícitamente en la regla de expansión existencial cuando manejamos variables libres. En 1987, Schmitt [Schm 87] presentó una nueva regla que, a partir de $\exists x \varphi(x)$, introducía en la rama $\varphi(f(x_1, \ldots, x_n))$, donde f es un nuevo símbolo de función y $x_1, \ldots, x_n$ son las variables libres que aparecen en la rama. Años después, se demostró que este conjunto de variables libres podía reducirse aún más y restringirse a las que aparecen en la propia fórmula $\exists x \varphi(x)$ [HS 94].

Entre las implementaciones recientes más populares debemos citar *THOT* [Schm 87], *HARP* [OS 88], *3TAP* [BGHK 92] y *leanTAP* [BP 94]. Actualmente el interés en el diseño de nuevas implementaciones se ha incrementado notablemente, quizá animado por la aparición de diversos *workshops* y conferencias dedicadas al tema.

Declaraciones de términos

Una lógica heterogénea es una lógica en la que el universo de discurso está dividido en subconjuntos llamados *géneros*. Estas lógicas disponen de un conjunto de *símbolos de género* S que se interpretan como dominios específicos en cualquier modelo. Por ejemplo, podemos dividir el universo de discurso en diferentes clases (géneros) de animales y plantas. Las variables de estas lógicas tienen asociado un género que restringe su dominio de acción, y existen diversos mecanismos para describir el comportamiento, con respecto a los géneros, de los símbolos de función y predicado. En definitiva, las lógicas heterogéneas disponen de herramientas específicas para el manejo de información taxonómica.

El interés de la demostración automática en estas lógicas estriba principalmente en su eficiencia computacional: el espacio de búsqueda puede reducirse porque es fácil detectar ciertas inferencias que no conducen a la construcción de una prueba; en concreto, aquellas inferencias que producen fórmulas *mal construidas*. Por ejemplo, la regla de expansión de los tableaux aplicada a $\forall x^s \varphi(x^s)$, donde x^s indica que la variable x tiene género s, debe introducir en la rama la fórmula $\varphi(t)$, para cualquier término t del que podamos asegurar sintácticamente que tiene género s, disminuyendo así el indeterminismo inherente a dicha regla de expansión.

Esta idea puede extenderse a otros métodos deductivos en los que la unificación tiene un papel relevante, como es el caso de la resolución o los tableaux con variables libres. En esos casos, las sustituciones deben ser *correctas* en el sentido de que deben respetar el género de cada variable sustituida. En el mundo de la resolución, los géneros son aceptados como un mecanismo adecuado para incrementar la eficiencia [Walt 87], [SS 89], [BHPSS 92], [Cohn 92], [Weid 93]; sin embargo, para los tableaux, sólo conocemos los trabajos de Schmitt y Wernecke [SW 89] y Weidenbach [Weid 95]. Goguen y Meseguer estudian las lógicas con géneros, pero en un marco más abstracto [GM 92].

Por otra parte, el uso de géneros supone un "incremento del poder expresivo" de la lógica. Por incremento del poder expresivo no debemos entender que los géneros permiten expresar conceptos que antes estaban formalmente fuera del alcance de la lógica (como es el caso de la lógica de predicados, que extiende la de proposiciones), sino que resulta más sencillo expresar ciertos conceptos (dándose una relación similar a la que se da entre los lenguajes de alto nivel y el código máquina). De hecho, los géneros actúan como predicados unarios y es posible probar resultados que aseguren la equivalencia expresiva entre lógicas de primer orden con y sin géneros. Sin embargo, la importancia de esta equivalencia es meramente teórica, dado que las expresiones que resultan cuando transformamos otras que sí consideran géneros, son muchos más complejas y, por ende, menos legibles y las pruebas involucradas más difíciles de automatizar.

No todas las lógicas heterogéneas actualmente diseñadas incorporan los géneros del mismo modo. Para hacerlo es importante conseguir una combinación adecuada entre los siguientes dos elementos:

- La información taxonómica relativa a los géneros, tales como las jerarquías entre los mismos y las declaraciones de los símbolos de función y predicado.
- Los datos dinámicos del lenguaje, es decir, las fórmulas.

De esta integración dependerá el poder expresivo de la lógica y la eficacia de los mecanismos de inferencia. Aunque existen otros enfoques mixtos, las dos grandes tendencias seguidas en la integración de los géneros difieren en el modo de manejar la información taxonómica. Tradicionalmente, la información sobre los géneros se fija de forma *estática* en la signatura. En este sentido, los géneros se parecen a los *tipos* en programación convencional; de hecho, comparten los mismos problemas de un lenguaje de programación fuertemente tipado, como es el caso de Pascal en el que la noción de *polimorfismo* resulta inaccesible.

Por otro lado, ciertas aplicaciones en el área del procesamiento del lenguaje natural y de la representación del conocimiento en ingeniería del software, indican que resulta interesante extraer la información relativa a los géneros de los propios datos [Her+ 86]. Así aparecen las lógicas que manejan los géneros de forma dinámica. Podemos tratar la relación de subgénero, y por tanto definir la jerarquía de géneros, de forma dinámica [GLMN 96], o incluso llegar a incorporar las declaraciones de las operaciones dentro de las expresiones del lenguaje. Con esta idea, Weidenbach [Weid 91] introdujo las lógicas con declaraciones de términos como aquellas en las que la información relativa a los géneros coexiste con las expresiones del lenguaje; y ello mediante el constructor $t \in s$, que expresa que el término t tiene género s.

La principal ventaja del enfoque dinámico, y por tanto de las declaraciones de términos, sobre el estático consiste en el incremento del poder expresivo: podemos razonar sobre las propiedades de los géneros al nivel mismo de las fórmulas; por ejemplo, podemos condicionar nuestros datos bajo ciertas suposiciones sobre el comportamiento de los géneros. De esta forma, la signatura no depende de las sentencias sobre las que trabaje nuestro sistema de inferencia. Además, desde el punto de vista de los tableaux, podemos disponer de una jerarquía de géneros distinta por cada rama del tableau.

Preórdenes monótonos

El razonamiento sobre determinadas teorías es una técnica importante para incrementar la eficacia de los sistemas de demostración automática. En concreto, podemos aprovechar las peculiaridades específicas de un dominio determinado (o teoría) a la hora de diseñar sistemas de inferencia que razonen sobre dicho dominio. Por otra parte, las aplicaciones reales en el campo de la inteligencia artificial atañen a dominios bien definidos, por lo que el estudio de estas técnicas tiene un alto valor.

La teoría que desde siempre ha despertado más interés es la inducida por la igualdad. La igualdad sirve, por ejemplo, para verificar el software involucrado en la especificación de *tipos abstractos de datos* y para plantear otros problemas matemáticos implicados en ciertas teorías algebraicas como las teorías AC, las de grupos, etc.

El aumento de la expresividad que se produce al incorporar la igualdad en una lógica es de sobra conocido, pero también es sabido que esta incorporación supone un aumento considerable del espacio de búsqueda de los sistemas de inferencia completos. En la lógica de primer orden, la igualdad no puede caracterizarse mediante un conjunto finito de axiomas, y es necesario diseñar técnicas específicas para integrarla en los sistemas de deducción si no queremos enfrentarnos con un conjunto infinito. No obstante, no basta con crear reglas de inferencia para este predicado que recojan implícitamente sus propiedades, sino que hay que desarrollar estrategias de control que reduzcan el amplio espacio de búsqueda que estas reglas suelen producir.

Existen sistemas de inferencia que consideran la igualdad, para la mayoría de los paradigmas conocidos en demostración automática (pueden consultarse los compendios [BFP 92], [Furb 94] y [Baum 98]). Por ejemplo, para resolución tenemos las populares reglas de paramodulación y superposición, más todas las variantes que conllevan [Rusi 91], [BGLS 92], [NR 92], [BGLS 95].

El uso de los métodos de tableaux para razonar sobre dominios concretos, y por tanto sobre la igualdad, resulta interesante porque podemos separar la complejidad propia de la lógica de primer orden, de la correspondiente a la teoría en cuestión. Una de las aproximaciones más estudiadas a la hora de incorporar la igualdad en los métodos de tableaux con variables libres, consiste en dividir la construcción del tableau en dos etapas:

1. La etapa de expansión, que no considera las peculiaridades de la igualdad, sino sólo el significado de los conectivos y cuantificadores de primer orden.

2. La etapa de cierre, en la que usamos las características propias de la igualdad para instanciar las variables y conseguir el cierre de las ramas.

De las dos etapas anteriores, sólo la etapa 2 considera las propiedades específicas de la igualdad, por lo que hemos separado la complejidad inherente a la igualdad de la del resto de la lógica. Desde la invención de esta técnica, que implica la resolución de problemas de *E-unificación rígida*, han aparecido gran número de publicaciones sobre el tema [GRS 87], [GNPS 90], [Baum 92], [BH 92]. Algunos de estos artículos estaban basados en la conjetura de que la E-unificación rígida simultánea era decidible; sin embargo, Degtyarev y Voronkov probaron en 1995 que este problema era en realidad indecidible [DV 96].

Además de esta técnica, se han usado otras para integrar la igualdad en los métodos de tableaux:

- La transformación previa del conjunto de sentencias inicial Φ en otro $\overline{\Phi}$ de forma que la satisfactibilidad en un modelo con igualdad de Φ equivalga a la satisfactibilidad de $\overline{\Phi}$ en cualquier modelo (sin igualdad) [Brand 75].

- El uso de refinamimentos basados en la paramodulación, como la superposición básica rígida [DV 94], [DV 98]. En este último artículo, Degtyarev y Voronkov prueban que basta un cálculo terminante, pero no completo, para la E-unificación rígida simultánea, a la hora de diseñar un método de tableaux correcto y completo.

- La técnica de *minus-normalization* [Wang 60], [Kang 63], [Matu 63], [Masl 67], que puede entenderse también como un cálculo incompleto para la E-unificación rígida simultánea, pero que integrado en los tableaux produce un método completo. Esta técnica consiste en saturar la aplicación de la regla de expansión, introduciendo todos los términos básicos que ya aparecían en el tableau. Dada su ineficiencia, su valor es sólo teórico.

Algunas investigaciones recientes han señalado la necesidad de extender el estudio de la igualdad al de otras relaciones transitivas (no simétricas). Este es el caso, por ejemplo, de CLP [JM 94], donde la resolución de restricciones se añade a la programación lógica. Las teorías preordenadas también se utilizan en el área de las álgebras de procesos [Inv 94] como una herramienta útil a la hora de establecer relaciones entre las distintas descripciones del sistema, y en concreto se han usado para especificaciones parciales [Walk 90] y divergencias entre modelos [AH 92].

En el campo de la demostración automática, esta situación ya ha propiciado el desarrollo de demostradores con reglas específicas para la relación que se estudia. Esta línea se sigue por ejemplo en [BKS 85] y [Hines 92] y, más recientemente, en [LA 93], [Levy 94], [BG 94], [BG 95] y [LA 96] aplicando técnicas de reescritura que evitan el cálculo del cierre transitivo de la teoría. Estas y otras aproximaciones están basadas en la resolución como método de inferencia.

En esta tesis estudiamos, por primera vez, la integración de los preórdenes (relaciones binarias transitivas y reflexivas) en los métodos de tableaux y usamos técnicas de reescritura para restringir el espacio de búsqueda. También extendemos los preórdenes para considerar monotonía, ocupándonos del diseño de técnicas especializadas. Como veremos, la falta de simetría supone un nuevo problema cuando permitimos el uso de variables no-lineales (aquellas que aparecen repetidas), ya que debemos adivinar ciertos contextos. Al igual que ocurre con la igualdad [Brand 75], probamos que los, por otra parte, ineficientes axiomas de reflexividad funcional, que caracterizan la igualdad como una relación de congruencia, son eliminables.

Estructura de la tesis

Primero introducimos las principales nociones de la lógica de predicados y de sus métodos de tableaux habituales, en el Capítulo 2. En el Capítulo 3 extendemos esta lógica con todos los ingredientes deseados: *géneros dinámicos*, dominios preordenados y operaciones monótonas y antimonótonas. Sin embargo, los géneros dinámicos no aprovechan todo el poder expresivo de las declaraciones de términos, puesto que sólo expresan la relación de subgénero dinámicamente. Por otra parte, los mecanismos de inferencia diseñados para manejar los preórdenes y la (anti)monotonía en los métodos de tableaux correspondientes resultan ineficientes debido a que usan axiomas de reflexividad funcional. Es por ello que en los capítulos siguientes especializamos nuestros métodos de tableaux en ambas direcciones. En el Capítulo 4 estudiamos la incorporación de las declaraciones de términos y diseñamos cálculos de unificación *tipada* correctos, completos y terminantes.

Finalmente, en el Capítulo 5 nos centramos en el estudio de los preórdenes monótonos y presentamos cálculos de unificación *preordenada y monótona*, correctos y completos, pero que sólo son terminantes cuando no consideramos monotonía.

Capítulo 2

Métodos de tableaux para la lógica de primer orden

En este capítulo introducimos, en primer lugar, la lógica de primer orden (LPO), sin duda la más usada y estudiada entre todas las conocidas. Como cualquier lógica, LPO está determinada por un lenguaje y su semántica. La sintaxis del lenguaje especifica el conjunto de expresiones gramaticalmente correctas, construidas a partir de un conjuto de símbolos Σ, que representa el alfabeto del lenguaje. La semántica, en cambio, se encarga de dotar de significado a dichas expresiones. Presentamos resultados sobradamente conocidos que relacionan sintaxis y semántica en LPO; resultados que serán frecuentemente citados en los siguientes capítulos.

Después de esto introducimos los tableaux semánticos como mecanismo de demostración automática utilizado a lo largo este trabajo. Se trata de un método refutacional del que presentamos sus versiones básica y con variables libres para LPO, así como referencias a los resultados de su corrección y completitud.

2.1 Sintaxis

Los símbolos permitidos en LPO son de dos tipos:

1. Símbolos lógicos con un significado predefinido -como los conectivos $\neg, \wedge$ y el cuantificador $\exists$- y un conjunto numerable de variables $X = \{x_1, x_2, \ldots\}$.

2. Una signatura o alfabeto Σ que consta de conjuntos finitos de símbolos de constante $\mathcal{C} = \{a, b, c, \ldots\}$, de función $\mathcal{F} = \{f, g, \ldots\}$ y de predicado $\mathcal{P} = \{P, Q, \ldots\}$. Cada uno de estos símbolos lleva un número asociado que denota su aridad; las funciones de aridad 0 representan en realidad constantes.

Con estos símbolos podemos construir términos y fórmulas como sigue.

Definición 2.1.1 *El conjunto $T(\Sigma, X)$ de Σ-términos se define mediante las siguientes reglas de formación:*

$$t ::= x(\in X) \mid c(\in C) \mid f(t_1, \ldots, t_n) \; (f \in \mathcal{F} \textit{de aridad } n;\; t_i \in T(\Sigma, X))$$

El conjunto $F(\Sigma, X)$ de Σ-fórmulas se define mediante las siguientes reglas de formación:

$$\varphi ::= P(t_1, \ldots, t_n) \; (P \in \mathcal{P} \textit{de aridad } n; t_i \in T(\Sigma, X)) \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x \varphi$$

Una fómula atómica o átomo es una Σ-fórmula del tipo $P(\ldots)$. Llamamos literal a una fórmula atómica o su negación.

En lo que sigue, utilizaremos las abreviaturas $\forall x\varphi$, $\varphi \vee \varphi'$, $\varphi \to \varphi'$ en lugar de las Σ-fórmulas $\neg\exists x(\neg\varphi)$, $\neg(\neg\varphi \wedge \neg\varphi')$ y $\neg\varphi \vee \varphi'$, respectivamente.

Para referirnos a posiciones concretas dentro de un término usaremos la siguiente notación.

Definición 2.1.2 *Dado un término t, $Pos(t)$ denota el conjunto de posiciones de t definido recursivamente por:*

$$Pos(t) = \begin{cases} \{\varepsilon\} & \textit{si } t \textit{ es una variable o constante} \\ \{\varepsilon\} \cup \{i.p | p \in Pos(t_i), 1 \leq i \leq n\} & \textit{si } t \textit{ es } f(t_1, \ldots, t_n). \end{cases}$$

Dos posiciones se comparan diciendo que $p \leq q$, si existe una posición r tal que $p.r = q$, donde . es la operación de concatenación de secuencias; en este caso $q - p$ denotará r. Dada $p \in Pos(t)$, $t|_p$ representará el subtérmino de t en la posición p, y $t[t']_p$ el resultado de sustituir $t|_p$ por t' en t. Posiciones, subtérminos y reemplazamientos pueden ser extendidos de forma natural a literales.

A continuación introducimos la notación correspondiente a la aparición de variables en un término o fórmula.

Definición 2.1.3

1. *$var(t)$ denota el conjunto de variables que aparecen en un término t. Un término t es básico si $var(t) = \emptyset$. Representamos el conjunto de términos básicos mediante $T(\Sigma)$.*

2. *Una variable está ligada en una fórmula φ si está afectada por un cuantificador con el mismo nombre de variable. En caso contrario, decimos que la variable aparece libre. $var(\varphi)$ denota el conjunto de variables libres que tiene la fórmula φ. Una fórmula φ es una sentencia si $var(\varphi) = \emptyset$. Representamos el conjunto de sentencias mediante $F(\Sigma)$.*

De forma natural, extendemos la notación $var(\ldots)$ a conjuntos de términos y de fórmulas.

2.2 Sustituciones y unificación

Las variables libres de las fórmulas pueden reemplazarse por términos más complejos.

Definición 2.2.1 *Dada una signatura Σ, una sustitución θ es una aplicación del conjunto de variables X en el conjunto de términos $T(\Sigma, X)$:*

$$\theta : X \longrightarrow T(\Sigma, X)$$

El dominio de una sustitución θ viene dado por

$$dom(\theta) = \{x \in X | \theta(x) \neq x\}$$

Decimos que una sustitución es básica si $\theta(x)$ es básico, para toda variable $x \in dom(\theta)$. Suponemos que el dominio de una sustitución θ es siempre finito $dom(\theta) = \{x_1, \ldots, x_n\}$ y representamos θ como:

$$\theta = [t_1/x_1, \ldots, t_n/x_n]$$

donde t_i es la imagen mediante θ de x_i.
El rango de una sustitución θ viene dado por:

$$ran(\theta) = \{\theta(x) | x \in dom(\theta)\}$$

El codominio de una sustitución θ se define como:

$$codom(\theta) = var(ran(\theta))$$

La sustitución identidad id es aquella que tiene dominio vacío.

La aplicación de una sustitución puede extenderse fácilmente a cualquier término; en efecto, usando la notación $t\theta$ en lugar de $\theta(t)$ tenemos:

1. $x\theta = \theta(x)$, para cualquier variable $x \in X$.

2. $f(t_1, \ldots, t_n)\theta = f(t_1\theta, \ldots, t_n\theta)$, para cualquier símbolo de función $f \in \mathcal{F}$ de aridad n.

También podemos extender la acción de una sustitución sobre fórmulas, aunque sin sustituir las variables ligadas y no permitiendo la captura de las nuevas variables introducidas. Es decir:

1. $P(t_1, \ldots, t_n)\theta = P(t_1\theta, \ldots, t_n\theta)$, para cualquier símbolo de predicado $P \in \mathcal{P}$ de aridad n.

2. Si $\varphi, \psi \in F(\Sigma, X)$ entonces $(\neg\varphi)\theta = \neg(\varphi\theta)$ y $(\varphi \wedge \psi)\theta = \varphi\theta \wedge \psi\theta$.

3. Si $\varphi \in F(\Sigma, X)$ entonces $(\exists x\varphi)\theta = \exists y((\varphi[y/x])\theta)$, donde y es una variable nueva que no aparece en φ tal que $y \notin dom(\theta)$ e $y \notin codom(\theta)$.

Definición 2.2.2 *Dada una sustitución θ y un conjunto de variables $V \subseteq X$, la sustitución θ restringida a V, escrito $\theta|_V$, se define como:*

$$\theta|_V(x) = \begin{cases} \theta(x) & si\ x \in V \\ x & en\ otro\ caso. \end{cases}$$

Definición 2.2.3 *La composición de dos sustituciones θ y σ es otra sustitución que escribimos $\theta\sigma$ y que definimos por $x\theta\sigma = \sigma(\theta(x))$, para toda variable $x \in X$.*

Definición 2.2.4 *Una sustitución θ es idempotente cuando $\theta\theta = \theta$.*

Podemos comparar sustituciones entre sí estudiando si una de ellas es un caso particular de la otra, en el siguiente sentido.

Definición 2.2.5 *Una sustitución θ es más general que otra σ si existe una tercera sustitución τ tal que $\sigma = \theta\tau$.*

Definición 2.2.6 *Dos términos $s, t \in T(\Sigma, X)$ son unificables si existe una sustitución θ tal que $s\theta = t\theta$. En tal caso decimos que θ es un unificador de la ecuación $s \simeq t$.*

La definición de unificador de una ecuación formada por términos puede extenderse de forma natural a átomos y a conjuntos de ecuaciones de términos o átomos.

De entre todos los unificadores de una ecuación entre términos, quizá infinitos, sólo nos interesan los más sencillos, en el sentido que se precisa a continuación.

Definición 2.2.7 *Un unificador θ de dos términos s, t es de máxima generalidad (umg) si para cualquier unificador σ de s, t se verifica $\sigma = \theta\sigma$.*

De entre todas las diferentes definiciones corespondientes a la noción de unificador de máxima generalidad, la definición que nosotros manejamos concuerda con la de unificador canónico debida a Robinson [Rob 65]. Obsérvese que con ésta podemos asegurar las siguientes propiedades.

Lema 2.2.8 *Sea θ un unificador de máxima generalidad de los términos s y t, entonces se verifica:*

1. *θ es más general que cualquier otro unificador*
2. *θ es idempotente*
3. *$dom(\theta) \subseteq var(s) \cup var(t)$*
4. *$codom(\theta) \subseteq var(s) \cup var(t)$.*

Demostración. (1) y (2) se deducen trivialmente a partir de la definición de unificador de máxima generalidad. Para (3), supongamos que existe $x \in dom(\theta)$ tal que $x \notin var(s) \cup var(t)$. Entonces la sustitución τ que se comporta como θ excepto que $x \notin dom(\tau)$ unifica t y s, pero no verifica $\theta\tau = \tau$, contra lo supuesto. En efecto, si $\theta\tau = \tau$ entonces $x\theta\tau = x\tau = x$; es decir, existe una variable $y \neq x$ tal que $x\theta = y$ y $y\tau = x$. Pero entonces $y \in dom(\theta)$ e $y\theta = y\tau$, por lo que obtenemos contradicción ya que $y = x\theta = y\theta\theta$ (porque $x = y\tau = y\theta) = y\theta$ (por (2)) $= y\tau = x$.

Para (4), supongamos que existe $x \in codom(\theta)$ tal que $x \notin var(s) \cup var(t)$, por tanto $x \notin dom(\theta)$, por (3). Entonces existe $y \in var(s) \cup var(t)$ tal que $x \in var(y\theta)$. Sin pérdida de generalidad supongamos que $y\theta = u[x]_p$, para algún término u con una posición p. Sean v_1, v_2 dos variables nuevas distintas y π la sustitución con $dom(\pi) = dom(\theta) \cup \{x\}$ definida por:

$$\pi(z) = \begin{cases} z\theta[v_1/x] & \text{si } z \in dom(\theta) \\ v_2 & z = x. \end{cases}$$

Entonces π unifica s, t, ya que $s\pi = s\theta[v_1/x] = t\theta[v_1/x] = t\pi$. En consecuencia, como θ es unificador de máxima generalidad, obtenemos que $\theta\pi = \pi$, lo que resulta falso. En efecto, $y\theta\pi = (u[x]_p)\pi = (u\pi)[v_2]_p \neq (u[v_1/x])[v_1]_p = y\theta[v_1/x] = y\pi$. ■

La unificación sintáctica, esto es, el problema correspondiente a determinar si dos términos son unificables, es decidible. En 1965, Robinson diseñó un algoritmo [Rob 65] que permite decidir si dos términos dados son unificables o no, y en caso de serlo calcula un umg, que además es único salvo renombramiento [LMM 86].

2.3 Semántica

Para interpretar términos y fórmulas en un contexto determinado usamos Σ-estructuras.

Definición 2.3.1 *Una Σ-estructura $\mathcal{D}$ es un triple compuesto por:*

1. *Un dominio no vacío D.*
2. *Un conjunto de interpretaciones de los símbolos de constante $\mathcal{C}^{\mathcal{D}} = \{c^{\mathcal{D}} \in D | c \in \mathcal{C}\}$*
3. *Un conjunto de interpretaciones de los símbolos de función $\mathcal{F}^{\mathcal{D}} = \{f^{\mathcal{D}} : D^n \to D | f \in \mathcal{F}$ de aridad $n\}$*
4. *Un conjunto de interpretaciones de los símbolos de predicado $\mathcal{P}^{\mathcal{D}} = \{P^{\mathcal{D}} : D^n \to \{\underline{f}, \underline{t}\} | P \in \mathcal{P}$ de aridad $n\}$.*

Una Σ-estructura nos permite interpretar términos básicos y sentencias. Para manejar variables libres usamos valoraciones.

Definición 2.3.2 *Una valoración ρ para la Σ-estructura $\mathcal{D}$ es una aplicación $\rho : X \to D$. Dado $d \in D$, $\rho[d/x]$ denota la valoración que coindice con ρ en todas las variables, excepto en x donde vale $\rho[d/x](x) = d$.*

El par formado por una Σ-estructura y una valoración es el contexto donde interpretamos términos y fórmulas.

Definición 2.3.3 *Una Σ-interpretación es un par $\langle \mathcal{D}, \rho \rangle$, donde $\mathcal{D}$ es una Σ-estructura y ρ es una valoración para $\mathcal{D}$.*

Definición 2.3.4 *El valor semántico de un Σ-término t en una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$ es un elemento $[\![t]\!]_\rho^\mathcal{D} \in D$ definido por inducción estructural sobre t como:*

1. $[\![x]\!]_\rho^\mathcal{D} = \rho(x)$
2. $[\![f(t_1, \ldots, t_n)]\!]_\rho^\mathcal{D} = f^\mathcal{D}([\![t_1]\!]_\rho^\mathcal{D}, \ldots, [\![t_n]\!]_\rho^\mathcal{D})$, *para cualquier símbolo de función $f \in \mathcal{F}$ de aridad n.*

Definición 2.3.5 *El valor veritativo de una Σ-fórmula φ en una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$ es un valor booleano $[\![\varphi]\!]_\rho^\mathcal{D} \in \{\underline{f}, \underline{t}\}$ definido por:*

1. $[\![P(t_1, \ldots, t_n)]\!]_\rho^\mathcal{D} = P^\mathcal{D}([\![t_1]\!]_\rho^\mathcal{D}, \ldots, [\![t_n]\!]_\rho^\mathcal{D})$.
2. *Si $\varphi, \psi \in F(\Sigma, X)$ entonces* $[\![\neg\varphi]\!]_\rho^\mathcal{D} = not\ [\![\varphi]\!]_\rho^\mathcal{D}$ *y* $[\![\varphi \wedge \psi]\!]_\rho^\mathcal{D} = [\![\varphi]\!]_\rho^\mathcal{D}\ and\ [\![\psi]\!]_\rho^{\mathcal{D}*}$.
3. *Si $\varphi \in F(\Sigma, X)$ entonces* $[\![\exists x\varphi]\!]_\rho^\mathcal{D} = \begin{cases} \underline{t} & \text{si existe } d \in D \text{ tal que } [\![\varphi]\!]_{\rho[d/x]}^\mathcal{D} = \underline{t} \\ \underline{f} & \text{en otro caso.} \end{cases}$

Definición 2.3.6 *Decimos que una fórmula $\varphi \in T(\Sigma, X)$ es satisfactible en una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$, escrito $\langle \mathcal{D}, \rho \rangle \models \varphi$, si $[\![\varphi]\!]_\rho^\mathcal{D} = \underline{t}$.*

Una fórmula φ es satisfactible si existe una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$ tal que $\langle \mathcal{D}, \rho \rangle \models \varphi$. En caso contrario, decimos que φ es insatisfactible.

Decimos que una Σ-estructura $\mathcal{D}$ es modelo de una fórmula φ, escrito $\mathcal{D} \models \varphi$, si para toda valoración ρ para $\mathcal{D}$ se tiene $\langle \mathcal{D}, \rho \rangle \models \varphi$. Decimos que una fórmula es cierta si tiene un modelo.

Una fórmula φ es válida si cualquier Σ-estructura es modelo de φ.

Las anteriores definiciones pueden extenderse de forma natural a un conjunto finito de fórmulas Φ; esto es, entendiendo Φ como la conjunción de sus fórmulas. Obsérvese que si trabajamos sobre sentencias, tener un modelo es equivalente a ser satisfactible. Haremos uso de este hecho repetidas veces a lo largo del trabajo.

Generalmente, en demostración automática, estamos interesados en probar la validez de una fórmula φ. El siguiente lema (cfr. [EFT 94] para su demostración) nos permite transformar el problema y pasar a preocuparnos únicamente de la insatisfactibilidad de sentencias.

Lema 2.3.7 *Dada una signatura Σ y una fórmula $\varphi \in F(\Sigma, X)$, entonces:*

*Las operaciones *not*, *and* entre los valores semánticos $\underline{t}, \underline{f}$ corresponden a las tablas veritativas clásicas para los conectivos $\neg, \wedge$.

1. *Para cualquier Σ-interpretación $\langle\mathcal{D},\rho\rangle$ tenemos: $\langle\mathcal{D},\rho\rangle \models \varphi$ si y sólo si $\langle\mathcal{D},\rho\rangle \not\models \neg\varphi$.*

2. *φ es válida si y sólo si $\neg\varphi$ es insatisfactible.*

3. *Si $\varphi, \neg\varphi \in \Phi$ entonces Φ es insatisfactible.*

4. *φ es válida si y sólo si $\forall x_1 \ldots \forall x_n \varphi$ es válida, donde $var(\varphi) = \{x_1, \ldots, x_n\}$.*

Definición 2.3.8 *Decimos que una fórmula $\varphi \in F(\Sigma, X)$ es consecuencia lógica de un conjunto de fórmulas $\Phi \subseteq F(\Sigma, X)$, escrito $\Phi \models \varphi$, si para toda Σ-estructura $\mathcal{D}$ tal que $\mathcal{D} \models \Phi$, se verifica $\mathcal{D} \models \varphi$.*

Si Φ y φ son sentencias, resulta obvio que $\Phi \models \varphi$ si y sólo si el conjunto $\Phi \cup \{\neg\varphi\}$ es insatisfactible. Sin embargo, este resultado no es cierto en presencia de variables libres, porque la definición anterior las considera universalmente cuantificadas. Por ejemplo, si $\Phi = \{P(x)\}$ y $\varphi = P(y)$ entonces obviamente $\Phi \models \varphi$, pero $\Phi \cup \{\neg\varphi\}$ es satisfactible. Dado que algunos métodos de tableaux razonan con variables libres, resulta interesante conseguir que esta propiedad, que relaciona los conceptos semánticos de consecuencia lógica y satisfactibilidad, resulte cierta. Para ello, consideraremos las variables de $var(\Phi \cup \{\varphi\})$ existencialmente cuantificadas, es decir, las variables libres denotan un único elemento arbitrario. Así definimos la noción de *consecuencia lógica rígida* como sigue.

Definición 2.3.9 *Decimos que una fórmula $\varphi \in F(\Sigma, X)$ es consecuencia lógica rígida de un conjunto de fórmulas $\Phi \subseteq F(\Sigma, X)$, escrito $\Phi \models_r \varphi$, si el conjunto $\Phi \cup \{\neg\varphi\}$ es insatisfactible.*

Los siguientes lemas establecen propiedades clásicas de LPO que resultan cruciales a la hora de probar la corrección y completitud de los métodos de tableaux que estudiaremos después. Sus demostraciones pueden encontrarse en [EFT 94].

Lema 2.3.10 (Sustitución) *Dada una signatura Σ, una Σ-interpretación $\langle\mathcal{D},\rho\rangle$ y un término $t' \in T(\Sigma, X)$, se tiene:*

1. *Para cualquier término $t \in T(\Sigma, X)$, $[\![t[t'/x]]\!]^{\mathcal{D}}_{\rho} = [\![t]\!]^{\mathcal{D}}_{\rho[[\![t']\!]^{\mathcal{D}}_{\rho}/x]}$.*

2. *Para cualquier fórmula $\varphi \in F(\Sigma, X)$, $[\![\varphi[t'/x]]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]^{\mathcal{D}}_{\rho[[\![t']\!]^{\mathcal{D}}_{\rho}/x]}$.*

Lema 2.3.11 (Coincidencia) *Dada una signatura Σ:*

1. *Si $t \in T(\Sigma, X)$ (resp. $\varphi \in F(\Sigma, X)$) y $\mathcal{D}$ y $\mathcal{D}'$ son dos Σ-estructuras que coinciden en el dominio y en cualquier interpretación de los símbolos que aparecen en t (resp. φ) entonces $[\![t]\!]^{\mathcal{D}}_{\rho} = [\![t]\!]^{\mathcal{D}'}_{\rho}$ (resp. $[\![\varphi]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]^{\mathcal{D}'}_{\rho}$), para toda valoración ρ para $\mathcal{D}$.*

2. *Si $t \in T(\Sigma, X)$ (resp. $\varphi \in F(\Sigma, X)$) y ρ y ρ' son dos valoraciones para una Σ-estructura $\mathcal{D}$ que coinciden en $var(t)$ (resp. $var(\varphi)$) entonces $[\![t]\!]^{\mathcal{D}}_{\rho} = [\![t]\!]^{\mathcal{D}}_{\rho'}$ (resp. $[\![\varphi]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'}$).*

2.4 Tableaux semánticos para LPO

Como ya hemos mencionado en el capítulo anterior, el método de los tableaux fue introducido por E. W. Beth [Beth 55] y K. J. J. Hintikka [Hint 55] en los años 50, aunque la versión más popular se debe a R. Smullyan [Smul 68] quien introdujo la elegante notación que hoy conocemos. Existen distintas versiones de tableaux para *LPO* de entre las que presentaremos los métodos estándar de tableaux básicos y de variables libres.

Un tableau para un conjunto de fórmulas es un árbol, con fórmulas en sus nodos, que crece y se ramifica mediante la aplicación de unas reglas de expansión que dependen del tipo de las fórmulas que etiquetan sus nodos. Atendiendo a dicho tipo, Smullyan dividió el conjunto de fórmulas $F(\Sigma, X)$ en cinco clases:

1. Fórmulas del tipo *Alfa*. Son las conjuntivas $\varphi_1 \wedge \varphi_2$ y las doblemente negadas $\neg\neg\varphi$.

2. Fórmulas del tipo *Beta*. Son las disyuntivas $\neg(\varphi_1 \wedge \varphi_2)$.

3. Fórmulas del tipo *Gamma*. Son las universalmente cuantificadas $\neg\exists x\varphi$.

4. Fórmulas del tipo *Delta*. Son las existencialmente cuantificadas $\exists x\varphi$.

5. Literales.

Obsérvese que esta clasificación de las fórmulas representa una partición del conjunto $F(\Sigma, X)$.

2.4.1 Tableaux básicos

Supongamos que hemos ampliado la signatura del lenguaje Σ con un conjunto infinito de constantes (de Skolem) auxiliares, y que representamos tal extensión como $\overline{\Sigma}$.

Para cada una de las anteriores clases de fórmulas -excepto la de los literales- existe una regla que expande el tableau según el carácter lógico de ésta:

1. Regla α. Añadimos uno o dos nuevos nodos a la rama donde se encuentra la fórmula *Alfa*. Esquemáticamente:

$$\frac{\varphi_1 \wedge \varphi_2}{\begin{array}{c}\varphi_1\\ \varphi_2\end{array}} \qquad\qquad \frac{\neg\neg\varphi}{\varphi}$$

2. Regla β. Bifurcamos la rama donde se encuentra la fórmula *Beta* añadiendo un nuevo nodo en cada nueva rama:

$$\frac{\neg(\varphi_1 \wedge \varphi_2)}{\neg\varphi_1 | \neg\varphi_2}$$

3. Regla γ. Añadimos un nuevo nodo a la rama donde se encuentra la fórmula *Gamma*, sustituyendo la variable universalmente cuantificada por un término básico cualquiera $t \in T(\overline{\Sigma})$:

$$\frac{\neg\exists x\varphi}{\neg\varphi[t/x]}$$

4. Regla δ. Añadimos un nuevo nodo a la rama donde se encuentra la fórmula *Delta*, sustituyendo la variable existencialmente cuantificada por una constante de Skolem c que no aparezca en la rama:

$$\frac{\exists x\varphi}{\varphi[c/x]}$$

Definición 2.4.1 *Una secuencia de tableaux para un conjunto finito no vacío Φ de Σ-sentencias es cualquier secuencia $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2 \ldots$ donde:*

1. *$\mathcal{T}_0$ es un árbol lineal con una sóla rama con tantos nodos como sentencias tiene Φ. Además estos nodos están etiquetados con las sentencias de Φ.*

2. *$\mathcal{T}_{k+1}$ proviene de $\mathcal{T}_k$ mediante la aplicación de alguna de las reglas de expansión α, β, γ y δ sobre los nodos de sus ramas.*

Un tableau infinito para un conjunto Φ de Σ-sentencias se define como el límite de alguna secuencia de tableaux.

Según las reglas de expansión, cada rama de un tableau puede verse como la conjunción de todas las sentencias que etiquetan sus nodos, y el tableau entero, como la disyunción de todas sus ramas.

Desde el punto de vista semántico, una secuencia de tableaux básicos para Φ representa una búsqueda sistemática de un modelo para Φ. A cada rama de estos tableaux les corresponde un posible modelo parcial en el que las fórmulas de la rama se hacen ciertas. Cada vez que se aplica una regla sobre un nodo de una rama, damos un paso más en la construcción de un modelo para dicha rama. Obviamente, cuando en una rama aparecen un par de nodos etiquetados con fórmulas *complementarias* (φ y $\neg\varphi$), el conjunto de fórmulas de dicha rama será insatisfactible. Dicha complementariedad puede restringirse a fórmulas atómicas.

Definición 2.4.2 *Decimos que una rama B en un tableau está cerrada, y entonces no se prolonga ni bifurca más, cuando se detecta una contradicción atómica entre sus etiquetas. Esto significa que existen en B dos fórmulas φ y $\neg\varphi$, siendo ambas atómicas. En otro caso B está abierta. Un tableau básico está cerrado si todas sus ramas están cerradas.*

La corrección y la completitud del método de tableaux básicos queda establecida por el siguiente resultado.

Teorema 2.4.3 (Corrección y Completitud) *[Fitt 96] Para todo conjunto finito Φ de Σ-sentencias, Φ tiene un tableau cerrado si y sólo si Φ es insatisfactible.*

Obsérvese que, aunque este teorema sólo hace referencia a la insatisfactibilidad de las sentencias, ésto no es un problema por el Lema 2.3.7. En este sentido decimos que el método de tableaux es un sistema de *refutación* ya que estudia la insatisfactibilidad de la sentencia $\neg\varphi$ en lugar de la validez de φ.

En cuanto a la eficacia del método, si Φ es satisfactible y además contiene alguna subfórmula del tipo γ, el proceso de expansión puede no acabar nunca -por esta razón se dice que el sistema de tableaux básicos es *parcialmente decidible* para *LPO*. Por otra parte, la implementación que usemos debe ser, en teoría, capaz de encontrar una expansión cerrada para Φ siempre que Φ sea insatisfactible -lo que se conoce como estrategia de expansión *justa* [Fitt 96]. En este sentido, incluso las implementaciones justas de tableaux básicos no son del todo completas, ya que al trabajar de forma finita es posible que no lleguen a alcanzar una expansión cerrada por limitaciones físicas de espacio o tiempo.

2.4.2 Tableaux con variables libres

A continuación presentamos un método de tableaux con variables libres para *LPO* en el que las fórmulas de clase *Gamma* introducen variables libres, en lugar de términos básicos. Estas variables son instanciadas a la hora de cerrar una rama, usando algún algoritmo de unificación sintáctica.

El sistema de tableaux con variables libres está compuesto por las anteriores reglas α, β y por las nuevas γ', δ'. Para δ', supondremos además que la signatura extendida $\overline{\overline{\Sigma}}$ también contiene infinitos símbolos de función de Skolem, para cualquier aridad.

1. Regla γ'. Añadimos un nuevo nodo a la rama donde se encuentra la fórmula *Gamma*, sustituyendo la variable universalmente cuantificada por una variable libre y, nueva en el tableau:

$$\frac{\neg\exists x\varphi}{\neg\varphi[y/x]}$$

2. Regla δ'. Añadimos un nuevo nodo a la rama donde se encuentra la fórmula *Delta*, sustituyendo la variable existencialmente cuantificada por el término $f(x_1, \ldots, x_n)$, donde f es un símbolo de Skolem, nuevo en el tableau, y las variables $x_1, \ldots, x_n$ son las variables libres que aparecen en la rama:

$$\frac{\exists x\varphi}{\varphi[f(x_1, \ldots, x_n)/x]}$$

Definición 2.4.4 (Cierre de máxima generalidad) *Un tableau con variables libres $\mathcal{T}$ y ramas $B_1, \ldots, B_k$ está cerrado si existe un conjunto unificable de ecuaciones de átomos $\{\varphi_1 \simeq \psi_1, \ldots, \varphi_k \simeq \psi_k\}$, donde φ_i y $\neg\psi_i$ son dos literales que aparecen en B_i, $1 \leq i \leq k$.*

Obsérvese que, en la anterior definición, hablamos de un conjunto en vez de un multiconjunto. Esto es así porque suponemos que hemos eliminado las redundancias que derivan de aquellas ecuaciones repetidas. Esta precisión atañe a las reglas de cierre que aparecen a lo largo del trabajo.

La corrección y completitud del método de tableaux con variables libres se establece como sigue.

Teorema 2.4.5 (Corrección y Completitud) *Para todo conjunto finito Φ de Σ-sentencias, Φ tiene un tableau con variables libres cerrado si y sólo si Φ es insatisfactible.*

La prueba de este teorema puede encontrarse en [Fitt 96]. No obstante, comentemos que la demostración de la completitud se basa en un lema que eleva cualquier tableau básico cerrado a otro con variables libres cerrado.

Como sucedía con los tableaux básicos, el método con variables libres es un sistema de refutación parcialmente decidible. Sin embargo, el sistema con variables libres no introduce términos básicos aleatoriamente cuando aplica la regla γ', sino que usa variables libres que son instanciadas a posteriori.

Existen varias mejoras que preservan la corrección y la completitud del método de tableaux con variables libres. Por ejemplo, Hähnle y Schmitt [HS 94] restringieron el conjunto de variables libres involucradas en la regla δ. Así su regla δ^+ sustituye la variable existencialmente cuantificada x por el término $f(x_1, \ldots, x_n)$, donde las variables $x_1, \ldots, x_n$ son únicamente las variables libres que aparecen en $\exists x\varphi$.

Ejemplo 2.4.6 *Demostramos que si "hay un hombre en la ciudad que afeita a todos los que no se afeitan a sí mismos", entonces "existe un hombre en la ciudad que se afeita a sí mismo". Formalizando la relación "x afeita a y" mediante $A(x, y)$, se trata de probar la siguiente consecuencia lógica:* $\{(\psi =)\exists x\forall y(\neg A(y, y) \to A(x, y))\} \models (\varphi =)\exists x A(x, x)$.

Usamos el tableau básico de la Figura 2.1 para demostrar que el conjunto $\{\psi, \neg\varphi\}$ es insatisfactible.

También podemos usar el sistema con variables libres para construir el tableau cerrado de la Figura 2.2. Obsérvese que este tableau está cerrado sin más que aplicar la sustitución $\theta = [c/x, c/y]$ que corresponde al umg del conjunto de ecuaciones $\{A(x, x) \simeq A(y, y), A(x, x) \simeq A(c, y)\}$.

$$
\begin{array}{c}
1\colon \exists x \forall y(\neg A(y,y) \longrightarrow A(x,y)) \\
2\colon \neg \exists x A(x,x) \\
\mid \ \delta \text{ a } 1 \\
3\colon \forall y(\neg A(y,y) \longrightarrow A(c,y)) \\
\mid \ \gamma \text{ a } 2 \\
3\colon \neg A(c,c) \\
\mid \ \gamma \text{ a } 3 \\
4\colon \neg A(c,c) \longrightarrow A(c,c) \\
\diagup \quad \diagdown \quad \beta \text{ a } 4 \\
\begin{array}{cc}
5\colon \neg\neg A(c,c) & 6\colon A(c,c) \\
\mid \ \alpha \text{ a } 5 & \\
7\colon A(c,c) &
\end{array}
\end{array}
$$

Figura 2.1: Expansión para $\{\psi, \neg\varphi\}$.

$$
\begin{array}{c}
1\colon \exists x \forall y(\neg A(y,y) \longrightarrow A(x,y)) \\
2\colon \neg \exists x A(x,x) \\
\mid \ \delta \text{ a } 1 \\
3\colon \forall y(\neg A(y,y) \longrightarrow A(c,y)) \\
\mid \ \gamma \text{ a } 2 \\
3\colon \neg A(x,x) \\
\mid \ \gamma \text{ a } 3 \\
4\colon \neg A(y,y) \longrightarrow A(c,y) \\
\diagup \quad \diagdown \quad \beta \text{ a } 4 \\
\begin{array}{cc}
5\colon \neg\neg A(y,y) & 6\colon A(c,y) \\
\mid \ \alpha \text{ a } 5 & \\
7\colon A(y,y) &
\end{array}
\end{array}
$$

Figura 2.2: Expansión para $\{\psi, \neg\varphi\}$.

Capítulo 3

Preórdenes y géneros dinámicos

En este capítulo presentamos una lógica especialmente diseñada para permitir el razonamiento con preórdenes y géneros ordenados, en la que las funciones y los predicados se comportan de forma monótona o antimonótona en sus argumentos y donde la información sobre la relación entre géneros está incorporada explícitamente en la sintaxis del lenguaje.

Para esta lógica proponemos dos métodos deductivos basados en tableaux, uno básico y otro con variables libres, de los cuales probaremos su corrección y completitud. Al final del capítulo se esbozan las bases de una implementación del núcleo del método con variables libres, basada en programación lógica.

3.1 Introducción

El estudio de métodos eficientes para el tratamiento de la igualdad se ha considerado tradicionalmente una importante línea de trabajo en diferentes áreas de la informática teórica. No obstante, algunas investigaciones recientes han mostrado la necesidad de extender este estudio a relaciones transitivas distintas de las de equivalencia o de la igualdad, en particular. Este es el caso, por ejemplo, de CLP [JM 94], donde la resolución de restricciones se añade a la programación lógica.

En el campo de la deducción automática, esta situación ha propiciado el desarrollo de demostradores autómaticos con reglas específicas que caracterizan la relación que se estudia. Esta línea fue seguida por ejemplo en [BKS 85] y [Hines 92] y, más recientemente, en [LA 93], [Levy 94], [BG 94], [BG 95] y [LA 96], aplicando técnicas basadas en reescritura. Estas y otras aproximaciones están basadas en resolución como método refutacional de prueba.

Por otra parte, el uso de géneros puede servir de gran ayuda para aproximar la programación al mundo real y, en particular, el uso de géneros ordenados sirve para incorporar, de un modo simple y elegante, funciones parciales, representación múltiple y constructores y selectores en datos estructurados [GM 92]. En demostración automática, además de lo dicho, el uso de géneros conlleva una reducción significativa del espacio de búsqueda.

Cuando se razona dentro de una jerarquía de géneros ordenados, es habitual mantener la información de los géneros separada de los datos. En este sentido, se dice que esta información se declara de forma estática. Pero en algunas situaciones, resulta necesario el razonamiento bajo ciertas hipótesis sobre los géneros, usando lo que se conoce con el nombre de géneros dinámicos [Weid 91].

En este capítulo, presentamos una lógica especialmente diseñada para permitir el razonamiento con preórdenes y géneros ordenados, en la que las funciones y los predicados se comportan de forma monótona o antimonótona en sus argumentos y donde la incorporación de la información de géneros en el lenguaje, está permitida por el uso de géneros dinámicos.

Para esta lógica, presentamos dos métodos correctos y completos basados en tableaux, uno básico y otro con variables libres. Este último sigue la línea de [Fitt 96] a la hora de añadir igualdad al método de tableaux clásico, sin olvidar las versiones de tableaux con variables libres para el tratamiento de la igualdad o de preórdenes, que aparecen en [BH 92] y en [GLN 96], respectivamente.

La organización del capítulo es la siguiente. En la sección 2 repasamos muy brevemente algunas nociones y notaciones usadas en lo sucesivo. En las tres secciones siguientes presentamos la lógica *LPGD*, su sintaxis y las sustituciones adecuadas para el manejo de géneros. La sección 6 presenta un método de tableaux básico que es extendido, en la sección 7, a una versión con variables libres. Para ambos métodos se demostrará su corrección y completitud. La sección 8 contiene algunos ejemplos y, finalmente, se esbozan las bases de una implementación del núcleo de *LPGD*, basada en Prolog.

3.2 Preliminares

Antes de nada, definimos los conceptos de preorden y (anti)monotonía.

Definición 3.2.1 (Preorden) *Un preorden es un par $(D, \sqsubseteq_D)$ donde D es un conjunto no vacío y $\sqsubseteq_D$ es una relación binaria sobre D, reflexiva y transitiva.*

Como casos particulares de preórdenes tenemos los preórdenes simétricos (relaciones de equivalencia; por ejemplo, la igualdad).

Definición 3.2.2 (Monotonía y antimonotonía) *Dados los preórdenes $(D_i, \sqsubseteq_{D_i})$, $1 \leq i \leq n$, y $(D, \sqsubseteq_D)$, una aplicación $f : D_1 \times \ldots \times D_n \to D$ se dice monótona (resp. antimonótona) en el i-ésimo argumento, si para todos $d_i, d'_i \in D_i$,*

$$d_i \sqsubseteq_{D_i} d'_i \Rightarrow f(d_1, \ldots, d_i, \ldots, d_n) \sqsubseteq_D f(d_1, \ldots, d'_i, \ldots, d_n)$$

$$(\text{resp. } d_i \sqsubseteq_{D_i} d'_i \Rightarrow f(d_1, \ldots, d'_i, \ldots, d_n) \sqsubseteq_D f(d_1, \ldots, d_i, \ldots, d_n)).$$

En este caso diremos que i es un argumento monótono (resp. antimonótono) de f.

El concepto de monotonía para funciones puede ser extendido a predicados, sin más que considerarlos como funciones booleanas y ordenar los valores booleanos $\{\underline{t}, \underline{f}\}$ de forma que $\underline{f} \sqsubseteq \underline{t}$.

3.3 Sintaxis de LPGD

En esta sección presentamos la que llamamos Lógica con Preórdenes usando Géneros Dinámicos (*LPGD* en lo que sigue), una lógica que extiende *LPO* con las siguientes características:

1. Es una lógica heterogénea en la que los géneros están ordenados. Además la información del orden entre géneros está incorporada dentro del propio lenguaje. La posibilidad de la declaración dinámica del orden entre los géneros resulta ser una gran mejora con respecto a las lógicas con géneros ordenados habituales, en que las relaciones entre géneros se declaran estáticamente, incluyéndolas de algún modo en la signatura.

2. Es una lógica preordenada, esto es, los dominios donde interpretamos los términos son preórdenes. Además, las operaciones (funciones y predicados) siempre se comportan en cualquiera de sus argumentos, de forma monótona o antimonótona.

Las signaturas Σ para *LPGD* constan de un conjunto finito S de géneros s, y una familia de conjuntos de constantes $\mathcal{C}^s$, símbolos de función $\mathcal{F}^{s_1,\ldots,s_l\to s}$ y símbolos de predicado $\mathcal{P}^{s_1,\ldots,s_r}$. Además, Σ contiene información sobre el comportamiento (anti)monótono de los símbolos de función y predicado mediante una aplicación m, de manera que para un símbolo de función f de aridad n, $m(f) : \{1\ldots n\} \longrightarrow \{0,1\}$, siendo $m(f)(i)$ 0 ó 1, $1 \leq i \leq n$, dependiendo de si f es monótona o antimonótona respectivamente, en el i-ésimo argumento; información análoga se tiene para los símbolos de predicado.

El uso de géneros nos permite restringir el rango de acción de las variables; así disponemos de una familia S-indexada de conjuntos de variables $X = (X^s)_{s\in S}$.

Los términos y las fórmulas de esta lógica se definen como sigue.

Definición 3.3.1 *El conjunto $T(\Sigma,X)$ de Σ-términos se define mediante las siguientes reglas de formación:*

$$t ::= x^s (\in X^s) \mid c^s\ (\in \mathcal{C}^s) \mid f(t_1,\ldots,t_n)\ (f \in \mathcal{F}^{s_1,\ldots,s_n\to s};\ t_i \in T(\Sigma,X))$$

El conjunto $F(\Sigma,X)$ de Σ-fórmulas se define mediante las siguientes reglas de formación:

$$\varphi ::= t \sqsubseteq t' \mid s \sqsubseteq_{@} s' \mid P(t_1,\ldots,t_n)\ (P \in \mathcal{P}^{s_1,\ldots,s_n}; t_i \in T(\Sigma,X)) \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x^s \varphi.$$

Antes de nada, obsérvese que a pesar de considerar una lógica heterogénea, estamos permitiendo la construcción de términos y predicados mal tipados con respecto a la información de géneros dada en la signatura. La razón de este aparente mal uso de los símbolos es dotar a la lógica de un mayor poder expresivo; concretamente, podrá tratar propiedades sobre datos, que estén condicionadas a hipótesis sobre los géneros. Este es el caso de la fórmula $s \sqsubseteq_{@} s' \to \forall x^s \exists x^{s'} (x^s \sqsubseteq x^{s'})$ que será interpretada como cierta, ya que la segunda desigualdad expresa una relación cierta entre datos, suponiendo que es cierta la relación de contenido entre géneros dada por la primera desigualdad. A la hora de hacer razonamientos, la combinación de la información del orden entre géneros, ofrecida por las fórmulas, y la información sobre la heterogeneidad de los símbolos dada por la signatura, reduce significativamente el espacio de búsqueda.

Una vez fijada una jerarquía mediante un conjunto de fórmulas de la forma $s \sqsubseteq_{@} s'$, podemos definir el conjunto de términos bien tipados con respecto a éste. Obviamente, este concepto puede ser extendido definiéndolo con respecto a cualquier conjunto de fórmulas, pero sólo teniendo en cuenta relaciones de jerarquía entre géneros como las anteriores.

Definición 3.3.2 *Dado un conjunto de Σ-fórmulas Φ, definimos la familia de conjuntos Φ-bien tipados, o bien tipados con respecto a Φ, de Σ-términos $\mathcal{T}_\Sigma^\Phi$ como la menor familia S-indexada de conjuntos $\mathcal{T}_\Sigma^\Phi(s)$ de Σ-términos tal que:*

1. $C^s \subset \mathcal{T}_\Sigma^\Phi(s)$.
2. $X^s \subset \mathcal{T}_\Sigma^\Phi(s)$.
3. *Si* $s' \sqsubseteq_{@} s \in \Phi$ *entonces* $\mathcal{T}_\Sigma^\Phi(s') \subseteq \mathcal{T}_\Sigma^\Phi(s)$.
4. *Si* $f \in \mathcal{F}^{s_1,\ldots,s_n \to s}$ *y* $t_i \in \mathcal{T}_\Sigma^\Phi(s_i)$, *para todo* $1 \leq i \leq n$, *entonces* $f(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma^\Phi(s)$.

Para cualquier término t, usaremos $sort(t)$ para representar el género *estático* de t deducido a partir de la signatura Σ; esta operación se define por:

- $sort(c^s) = sort(x^s) = s$.
- $sort(f(t_1, \ldots, t_n)) = s$, si $f \in \mathcal{F}^{s_1,\ldots,s_n \to s}$.

La definición del conjunto de términos bien tipados $\mathcal{T}_\Sigma^\Phi$ usa implícitamente el cierre reflexivo y transitivo de $\sqsubseteq_{@}$. Por este motivo definimos el concepto de *subgénero* de forma explícita.

Definición 3.3.3 *Dado un conjunto de Σ-fórmulas Φ, decimos que el género s es subgénero de s', escrito $s \ll^\Phi s'$, si $s' = s$ o existe en Φ una secuencia de la forma*

$$s \sqsubseteq_{@} s_1, s_1 \sqsubseteq_{@} s_2, \ldots s_k \sqsubseteq_{@} s'.$$

Si Φ es un conjunto finito, la relación $\ll^\Phi$ es decidible y puede computarse en $O(card(S)^3)$ sin más que usar el algoritmo de Floyd [Floyd 62].

Los términos bien tipados cumplen las siguientes propiedades.

Lema 3.3.4 1. $t \in \mathcal{T}_\Sigma^\Phi(s) \iff (sort(t) \ll^\Phi s$ *y* $t \in \mathcal{T}_\Sigma^\Phi(sort(t)))$.

2. *Si* $f \in F^{s_1,\ldots,s_n \to s}$ *y* $t = f(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma^\Phi(s)$ *entonces* $t_i \in \mathcal{T}_\Sigma^\Phi(s_i), 1 \leq i \leq n$.

Demostración. (1) El sentido $\Leftarrow$ es inmediato. Para $\Rightarrow$, supongamos que $sort(t) \not\ll^\Phi s$. Sea $\mathcal{X}$ la familia S-indexada de términos definida para todo $r \in S$ como:

$$\mathcal{X}^r = \begin{cases} \mathcal{T}_\Sigma^\Phi(r) - \{t\} & \text{si } sort(t) \not\ll^\Phi r \\ \mathcal{T}_\Sigma^\Phi(r) & \text{en otro caso} \end{cases}$$

Llegaremos a contradicción probando que $\mathcal{X}$ es, en realidad, la familia S-indexada de Σ-términos bien tipados con respecto a Φ, en contra de la minimalidad de $\mathcal{T}_\Sigma^\Phi$, cuando $r = s$; vamos a probar, por tanto, que $\mathcal{X}$ verifica la Definición 3.3.2. Las condiciones 1, 2 y 4 pueden ser comprobadas fácilmente ya que sólo eliminamos t cuando $sort(t) \not\ll^\Phi r$ (luego $sort(t) \neq r$). Para la condición 3, sean s', s'' dos géneros tales que $t \in \mathcal{X}^{s'}$, $s' \sqsubseteq_@ s'' \in \Phi$ y $t \notin \mathcal{X}^{s''}$; esto último ha de ser porque $sort(t) \not\ll^\Phi s''$. Pero esto es imposible pues entonces $sort(t) \not\ll^\Phi s'$ y en consecuencia $t \notin \mathcal{X}^{s'}$.

Una vez probado que $sort(t) \ll^\Phi s$, supongamos $t \notin \mathcal{T}_\Sigma^\Phi(sort(t))$. Razonamos como antes usando la familia $\mathcal{X}$ definida para todo $r \in S$ como:

$$\mathcal{X}^r = \mathcal{T}_\Sigma^\Phi(r) - \{t\}$$

En estas condiciones obtenemos de nuevo contradicción demostrando que $\mathcal{X}$ es $\mathcal{T}_\Sigma^\Phi$, en contra de la minimalidad de $\mathcal{T}_\Sigma^\Phi$, cuando $r = s$. En efecto, $\mathcal{X}$ verifica las condiciones 1, 2 y 4 de la Definición 3.3.2 ya que $t \notin \mathcal{T}_\Sigma^\Phi(sort(t))$. Además 3 es inmediato porque $t \notin \mathcal{X}$.

(2) Razonamos por contradicción. Supongamos que $t_i \notin \mathcal{T}_\Sigma^\Phi(s_i)$, para algún $1 \leq i \leq n$, y sea $\mathcal{X}$ la familia definida para todo $r \in S$ como:

$$\mathcal{X}^r = \mathcal{T}_\Sigma^\Phi(r) - \{t\}$$

De nuevo demostramos que $\mathcal{X}$ es $\mathcal{T}_\Sigma^\Phi$ contra la minimalidad de $\mathcal{T}_\Sigma^\Phi$ en s. En efecto, $\mathcal{X}$ verifica las condiciones 1, 2, 3 de la Definición 3.3.2 de forma inmediata. Además 4 es cierto porque $t_i \notin \mathcal{T}_\Sigma^\Phi(s_i)$. ∎

El anterior Lema asegura la decidibilidad del problema ¿$t \in \mathcal{T}_\Sigma^\Phi(s)$? De hecho puede resolverse en $O(card(S)^3 + card(Pos(t)))$, si Φ es finito. En efecto, primero computamos la relación de subgénero $\ll^\Phi$ y luego descomponemos el término t progresivamente. Por ejemplo, si $f \in F^{s_1 \to s_2}$ y $t = f(t')$, basta comprobar si $s_2 \ll^\Phi s$ y si $t' \in \mathcal{T}_\Sigma^\Phi(s_1)$.

A continuación, extendemos el concepto de *buena tipificación* de términos a fórmulas.

Definición 3.3.5 *La buena tipificación de una fórmula φ con respecto a un conjunto de Σ-fórmulas Φ, lo que escribimos como $WS(\varphi, \Phi)$, es una propiedad definida inductivamente sobre la estructura de φ como sigue:*

1. $WS(s \sqsubseteq_@ s', \Phi) \Leftrightarrow s \sqsubseteq_@ s' \in \Phi$.

2. $WS(t_1 \sqsubseteq t_2, \Phi) \Leftrightarrow$ *existe* $s \in S$ *tal que* $t_i \in \mathcal{T}_\Sigma^\Phi(s)$, $i = 1, 2$.

3. $WS(P(t_1, \ldots, t_n), \Phi) \Leftrightarrow t_i \in \mathcal{T}_\Sigma^\Phi(s_i)$, $1 \leq i \leq n$, $P \in \mathcal{P}^{s_1 \ldots s_n}$.

4. $WS(\neg\varphi, \Phi) \Leftrightarrow WS(\varphi, \Phi)$.

5. $WS(\varphi_1 \wedge \varphi_2, \Phi) \Leftrightarrow WS(\varphi_i, \Phi), i = 1, 2$.

6. $WS(\exists x^s\varphi, \Phi) \Leftrightarrow WS(\varphi, \Phi)$.

Para un conjunto de fórmulas Φ' *definimos* $WS(\Phi', \Phi) \Leftrightarrow$ *para todo* $\varphi \in \Phi'$, $WS(\varphi, \Phi)$.

Como sucedía para los términos, podemos usar de nuevo el Lema 3.3.4 para asegurar que el problema ¿$WS(\varphi, \Phi)$? también es decidible.

3.4 Sustituciones para LPGD

Las sustituciones para *LPGD* son las mismas que para *LPO* por lo que pueden no respetar el género de alguna de las variables sustituidas. Para evitar estas situaciones, introducimos el concepto de sustituciones bien tipadas con respecto a cierta información sobre géneros. Esta noción será utilizada a la hora de cerrar un tableau con variables libres, pues las variables deben ser sustituidas por términos bien tipados con respecto a la información contenida en toda rama del tableau donde aparezcan.

Definición 3.4.1 *Una sustitución* τ *está bien tipada con respecto a un conjunto de* Σ-*fórmulas* Φ, *escrito* $WS(\tau, \Phi)$, *si* $\tau(x^s) \in \mathcal{T}_\Sigma^\Phi(s)$, *para toda variable* $x^s \in dom(\tau)$.

Las sustituciones bien tipadas verifican las siguientes buenas propiedades.

Lema 3.4.2 *Dados un término* t, *un conjunto de fórmulas* $\Phi \cup \{\varphi\}$ *y una sustitución* τ *tal que* $WS(\tau, \Phi)$, *se tienen las siguientes propiedades:*

1. *Si* $t \in \mathcal{T}_\Sigma^\Phi(s)$ *entonces* $t\tau \in \mathcal{T}_\Sigma^\Phi(s)$.
2. *Si* $WS(\varphi, \Phi)$ *entonces* $WS(\varphi\tau, \Phi)$.

Demostración. (1) Procedemos por inducción sobre la estructura de $t \in \mathcal{T}_\Sigma^\Phi(s)$. Si t es una constante o una variable $x^{s'} \notin dom(\tau)$, la prueba es inmediata. Si $t = x^{s'}$ y $x^{s'} \in dom(\tau)$, usamos el Lema 3.3.4(1) para obtener que $s' \ll^\Phi s$ y $x^{s'} \in \mathcal{T}_\Sigma^\Phi(s')$. Además tenemos que $x^{s'}\tau \in \mathcal{T}_\Sigma^\Phi(s')$, porque $WS(\tau, \Phi)$. Por tanto concluimos que $t\tau \in \mathcal{T}_\Sigma^\Phi(s)$ sin más que usar $s' \ll^\Phi s$.

Si $t = f(t_1, \ldots, t_n)$ y $f \in F^{s_1,\ldots,s_n \to s'}$, aplicamos el Lema 3.3.4(1) y (2) para obtener que $s' \ll^\Phi s$ y $t_i \in \mathcal{T}_\Sigma^\Phi(s_i)$. Usando hipótesis de inducción, tenemos que $t_i\tau \in \mathcal{T}_\Sigma^\Phi(s_i)$ y, por tanto, $t\tau \in \mathcal{T}_\Sigma^\Phi(s') \subseteq \mathcal{T}_\Sigma^\Phi(s)$.

(2) Procedemos por inducción sobre la estructura de la fórmula φ que verifica $WS(\varphi, \Phi)$. La demostración es inmediata para el caso $s \sqsubseteq_@ s'$. Si $\varphi = t \sqsubseteq t'$ entonces debe existir un género s tal que $t, t' \in \mathcal{T}_\Sigma^\Phi(s)$. Aplicando (1), obtenemos que $t\tau, t'\tau \in \mathcal{T}_\Sigma^\Phi(s)$ y, por tanto, $WS(\varphi\tau, \Phi)$. El caso $\varphi = P(t_1, \ldots, t_n)$ se prueba de forma análoga. Para el caso de los conectivos usamos hipótesis de inducción. ∎

Los recíprocos de los anteriores lemas sólo son ciertos cuando la sustitución τ respeta el género de cada variable instanciada, de forma *estática*, en el sentido siguiente.

Definición 3.4.3 *La sustitución τ respeta los géneros de forma estática si $sort(\tau(x^s)) = s$, para toda variable $x^s \in dom(\tau)$.*

Cuando demostremos la completitud del método de variables libres sólo utilizaremos estas sustituciones porque verifican las siguientes propiedades.

Lema 3.4.4 *Dados dos términos t, t', un conjunto de fórmulas $\Phi \cup \{\varphi\}$ y una sustitución τ que respeta los géneros de forma estática, se tiene:*

1. *Si $t\tau \in \mathcal{T}_\Sigma^\Phi(s)$ entonces $t \in \mathcal{T}_\Sigma^\Phi(s)$.*
2. *Si $WS(\varphi\tau, \Phi)$ entonces $WS(\varphi, \Phi)$.*
3. *Si $t\tau = t'\tau$, $WS(\tau, \Phi)$ y $\theta = umg(t, t')$ entonces $WS(\theta, \Phi)$.*

Demostración. (1) Procedemos por inducción sobre la estructura del término t tal que $t\tau \in \mathcal{T}_\Sigma^\Phi(s)$. Si t es una constante o una variable $x^{s'} \notin dom(\tau)$, la prueba es inmediata. Si $t = x^{s'}$ y $x^{s'} \in dom(\tau)$ entonces, por hipótesis, $x^{s'}\tau \in \mathcal{T}_\Sigma^\Phi(s)$ y $sort(x^{s'}\tau) = s'$. Usando el Lema 3.3.4(1), obtenemos que $s' \ll^\Phi s$ y $x^{s'}\tau \in \mathcal{T}_\Sigma^\Phi(s')$. En consecuencia, $x^{s'} \in \mathcal{T}_\Sigma^\Phi(s)$. Para el caso $t = f(t_1, \ldots, t_n)$ con $f \in F^{s_1, \ldots, s_n \to s'}$, aplicamos el Lema 3.3.4(1) y (2) para obtener que $s' \ll^\Phi s$ y $t_i\tau \in \mathcal{T}_\Sigma^\Phi(s_i)$. Usando hipótesis de inducción concluimos que $t_i \in \mathcal{T}_\Sigma^\Phi(s_i)$ y, por tanto, $t \in \mathcal{T}_\Sigma^\Phi(s') \subseteq \mathcal{T}_\Sigma^\Phi(s)$.

(2) Procedemos por inducción sobre la estructura de la fórmula φ tal que $WS(\varphi\tau, \Phi)$. La demostración es inmediata para el caso $s \sqsubseteq_@ s'$. Si $\varphi = t \sqsubseteq t'$ entonces debe existir un género s tal que $t\tau, t'\tau \in \mathcal{T}_\Sigma^\Phi(s)$. Aplicando (1) obtenemos que $t, t' \in \mathcal{T}_\Sigma^\Phi(s)$ y, por tanto, $WS(\varphi, \Phi)$. El caso $\varphi = P(t_1, \ldots, t_n)$ se prueba de forma análoga y para los conectivos usamos hipótesis de inducción.

(3) Sea $\theta = umg(t, t')$ con $\theta = [t_1/{x_1}^{s_1}, \ldots, t_n/{x_n}^{s_n}]$. Por la generalidad de θ, $t_i\tau = {x_i}^{s_i}\tau$, para todo $1 \leq i \leq n$. Usando $WS(\tau, \Phi)$ obtenemos $t_i\tau \in \mathcal{T}_\Sigma^\Phi(s_i)$ y, aplicando (1), $t_i \in \mathcal{T}_\Sigma^\Phi(s_i)$. ∎

Obsérvese que si no exigimos que τ respete los géneros de forma estática, las anteriores propiedades no tienen por qué ser ciertas. Por ejemplo, para (3) obtenemos contradicción si tomamos $t = x^s, t' = y^{s'}, \Phi = \{s'' \sqsubseteq_@ s, s'' \sqsubseteq_@ s'\}$ y $\tau = [z^{s''}/x^s, z^{s''}/y^{s'}]$, donde $z^{s''} \in X^{s''}$, ya que $WS(\tau, \Phi), t\tau = t'\tau$, pero $\theta = umg(t, t') = [y^{s'}/x^s]$ no verifica $WS(\theta, \Phi)$.

3.5 Semántica de LPGD

Para interpretar términos y fórmulas, las Σ-estructuras de *LPGD* contienen dominios preordenados para cada género, así como un elemento especial $\perp$ para representar el valor de cualquier término sintáctico que haga referencia a un elemento inexistente en el dominio. Obsérvese que esta falta de definición en un término depende de la relación que existe entre los dominios de la estructura.

Aunque estamos siguiendo una aproximación similar a otras usuales en el tratamiento de funciones parciales, en nuestro caso la falta de definición no expresa parcialidad; de hecho, si un término es interpretado como indefinido entonces es que contiene un subtérmino mal tipado con respecto a la signatura. Por supuesto, la implicación contraria es, en general, falsa porque, como ya se dijo, el valor semántico de un término depende de la relación entre los dominios de la estructura.

Definición 3.5.1 *Una Σ-estructura $\mathcal{D}$ está compuesta por un sistema $\{(D^s, \sqsubseteq_s^{\mathcal{D}}) | s \in S\} \cup \{\bot\}$ y por las interpretaciones de las constantes $\{c^{\mathcal{D}} \in D^s | c \in C^s\}$, los símbolos de función $\{f^{\mathcal{D}} : D^{s_1} \times \ldots \times D^{s_l} \to D^s | f \in \mathcal{F}^{s_1,\ldots,s_l \to s}\}$ y los símbolos de predicado $\{P^{\mathcal{D}} : D^{s_1} \times \ldots \times D^{s_r} \to \{\underline{f}, \underline{t}\} | P \in \mathcal{P}^{s_1,\ldots,s_r}\}$ de manera que:*

1. *$\{(D^s, \sqsubseteq_s^{\mathcal{D}}) | s \in S\}$ es una familia de preórdenes* transitiva *en el sentido siguiente: si $d \sqsubseteq_s^{\mathcal{D}} d'$, $d' \sqsubseteq_{s'}^{\mathcal{D}} d''$, y $d, d'' \in D^{s''}$ entonces $d \sqsubseteq_{s''}^{\mathcal{D}} d''$.*

2. *Si $m(f)(i) = 0$ (resp. 1) entonces $f^{\mathcal{D}}$ es monótona (resp. antimonótona) en el i-ésimo argumento. Análogamente para símbolos de predicado.*

Una valoración ρ para $\mathcal{D}$ es una familia S-indexada de aplicaciones $\rho^s : X^s \to D^s$. Una Σ-interpretación es un par $\langle \mathcal{D}, \rho \rangle$ tal que $\mathcal{D}$ es una Σ-estructura y ρ es una valoración para $\mathcal{D}$.

Ya que la lógica está valorada de forma clásica, no podemos conseguir una verdadera aproximación funcional. Por tanto, las variables en las valoraciones y en la semántica de cuantificadores se moverán sobre individuos del respectivo dominio, y no tomarán el valor $\bot$.

La transitividad de los preórdenes es necesaria para asegurar la corrección de algunas reglas de expansión del método de tableaux (por ejemplo, las reglas ζ_1, ζ_2, véase más adelante). En particular, con esta condición se puede probar algo tan lógico como que dados los géneros s, s', si $d, d' \in D^s \cap D^{s'}$ y $d \sqsubseteq_s d'$ entonces $d \sqsubseteq_{s'} d'$.

En la semántica de los términos expresamos que las funciones son estrictas. Por otra parte, para predicados y desigualdades entre datos, la semántica asigna el valor falso cuando aparece alguna vez el elemento indefinido.

Definición 3.5.2 *El valor semántico de un Σ-término t^s en una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$ es un elemento $[\![t^s]\!]_\rho^{\mathcal{D}} \in D^s \cup \{\bot\}$ definido por inducción sobre t como:*

- $[\![c]\!]_\rho^{\mathcal{D}} = c^{\mathcal{D}}$
- $[\![x]\!]_\rho^{\mathcal{D}} = \rho(x)$
- $[\![f(t_1, \ldots, t_n)]\!]_\rho^{\mathcal{D}} = \begin{cases} f^{\mathcal{D}}([\![t_1]\!]_\rho^{\mathcal{D}}, \ldots, [\![t_n]\!]_\rho^{\mathcal{D}}) & \textit{si } [\![t_i]\!]_\rho^{\mathcal{D}} \in D^{s_i}, 1 \le i \le n \\ \bot & \textit{en otro caso.} \end{cases}$

  para cualquier $f \in \mathcal{F}^{s_1,\ldots,s_n \to s}$.

El valor veritativo de una Σ-fórmula φ en una Σ-interpretación $\langle \mathcal{D}, \rho \rangle$ es un valor booleano $[\![\varphi]\!]_\rho^{\mathcal{D}} \in \{\underline{f}, \underline{t}\}$ definido por inducción sobre φ como:

- $[\![s \sqsubseteq_{@} s']\!]_{\rho}^{\mathcal{D}} = \begin{cases} \underline{t} & \text{si } D^s \subseteq D^{s'} \\ \underline{f} & \text{en otro caso.} \end{cases}$

- $[\![t_1 \sqsubseteq t_2]\!]_{\rho}^{\mathcal{D}} = \begin{cases} \underline{t} & \text{si existe } s \text{ tal que } ([\![t_i]\!]_{\rho}^{\mathcal{D}} \in D^s, i = 1, 2, \text{ y } [\![t_1]\!]_{\rho}^{\mathcal{D}} \sqsubseteq_s^{\mathcal{D}} [\![t_2]\!]_{\rho}^{\mathcal{D}}) \\ \underline{f} & \text{en otro caso.} \end{cases}$

- $[\![P(t_1, \ldots, t_n)]\!]_{\rho}^{\mathcal{D}} = \begin{cases} P^{\mathcal{D}}([\![t_1]\!]_{\rho}^{\mathcal{D}}, \ldots, [\![t_n]\!]_{\rho}^{\mathcal{D}}) & \text{si } [\![t_i]\!]_{\rho}^{\mathcal{D}} \in \mathcal{D}^{s_i}, 1 \leq i \leq n \\ \underline{f} & \text{en otro caso.} \end{cases}$

  para cualquier $P \in \mathcal{P}^{s_1, \ldots, s_n}$.

- *El valor semántico para* $\neg, \wedge$ *y* $\exists$ *se define como es usual en LPO.*

Con respecto al cuantificador existencial, sólo hay que tener en cuenta que el rango de una variable queda restringido por el género de la misma.

Los conceptos de modelo, satisfactibilidad y consecuencia lógica se definen como en *LPO*.

Acabamos esta sección recogiendo algunas propiedades semánticas importantes de *LPGD*. Previamente hacemos observar que los términos funcionales y los predicados se comportan monótonamente (resp. antimonótonamente), cuando incrementamos el valor de un subtérmino que aparece dentro de un número par (resp. impar) de argumentos antimonótonos. Este hecho será crucial cuando se definan algunas reglas de expansión de tableaux. Veámoslo más formalmente.

Definición 3.5.3 *Para* $p \in Pos(t)$, *la función* $\mathcal{F}_{t,p} : \{q \in Pos(t) | q < p\} \to N$, *donde* $\mathcal{F}_{t,p}(q)$ *es el primer dígito de* $p - q$, *devuelve el lugar que ocupa el argumento de* $t|_q$ *en el que* $t|_p$ *aparece como subtérmino. En adelante, escribiremos* $\mathcal{F}$ *en lugar de* $\mathcal{F}_{t,p}$ *cuando no haya posible confusión.*

Definición 3.5.4 *Dado un término* t *y* $p \in Pos(t)$, *decimos que* p *es una posición monótona de* t *si el número* $\sum_{q<p} m(raiz(t|_q))(\mathcal{F}(q))$ *es par, donde* $raiz(t)$ *es el símbolo que aparece en la raíz de* t *cuando éste es visto como un árbol. En otro caso* p *es una posición antimonótona de* t.

Dados $P(t_1, \ldots, t_n)(= A)$, $1 \leq i \leq n$, *y* $p \in Pos(t_i)$, *decimos que* p *es una posición monótona de* A *en* i, *si* $\sum_{q<p} m(raiz(t_i|_q))(\mathcal{F}(q)) + m(P)(i)$ *es par. En otro caso* p *es una posición antimonótona de* A *en* i.

Lema 3.5.5 *1. Interpretación de términos bien tipados.*

Dados un conjunto de Σ*-fórmulas* Φ *y un término* t, *si* $\langle \mathcal{D}, \rho \rangle \models \Phi$ *y* $t \in \mathcal{T}_{\Sigma}^{\Phi}(s)$ *entonces* $[\![t]\!]_{\rho}^{\mathcal{D}} \in D^s$.

2. Lema de sustitución para términos y fórmulas.

Dados un conjunto de Σ*-fórmulas* Φ *y una sustitución* $\tau = [t_1/x_1, \ldots, t_n/x_n]$, *si* $\langle \mathcal{D}, \rho \rangle \models \Phi$ *y* $WS(\tau, \Phi)$ *entonces se verifica:*

- $[\![t\tau]\!]_{\rho}^{\mathcal{D}} = [\![t]\!]_{\rho'}$

- $[\![\varphi\tau]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]_{\rho'}$

  donde $\rho' = \rho[[\![t_1]\!]^{\mathcal{D}}_{\rho}/x_1, \ldots, [\![t_n]\!]^{\mathcal{D}}_{\rho}/x_n]$.

3. *(Anti)Monotonía de términos.*

   Sea $p \in Pos(t)$ *una posición monótona (resp. antimonótona) de* t. *Si* $\langle \mathcal{D}, \rho\rangle \models t_1 \sqsubseteq t_2$ *y existe* s *tal que* $[\![t[t_i]_p]\!]^{\mathcal{D}}_{\rho} \in D^s$, $i = 1, 2$, *entonces* $[\![t[t_1]_p]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}}_s [\![t[t_2]_p]\!]^{\mathcal{D}}_{\rho}$ *(resp.* $[\![t[t_2]_p]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}}_s [\![t[t_1]_p]\!]^{\mathcal{D}}_{\rho}$*).*

4. *(Anti)Monotonía de predicados.*

   Sea $P(t_1, \ldots, t_n)(= A)$ *y* $p \in Pos(t_i)$ *una posición monótona (resp. antimonótona) de* A *en* i. *Si* $\langle \mathcal{D}, \rho\rangle \models t^1 \sqsubseteq t^2$ *y* $[\![t_i[t^j]_p]\!]^{\mathcal{D}}_{\rho} \in D^{s_i}$, $j = 1, 2$, *entonces:*

$$[\![P(t_1, \ldots, t_i[t^1]_p, \ldots, t_n)]\!]^{\mathcal{D}}_{\rho} \Rightarrow [\![P(t_1, \ldots, t_i[t^2]_p, \ldots, t_n)]\!]^{\mathcal{D}}_{\rho}$$

$$(\textit{resp. } [\![P(t_1, \ldots, t_i[t^2]_p, \ldots, t_n)]\!]^{\mathcal{D}}_{\rho} \Rightarrow [\![P(t_1, \ldots, t_i[t^1]_p, \ldots, t_n)]\!]^{\mathcal{D}}_{\rho}).$$

Demostración. (1) Por inducción sobre t. En el caso básico, si $t = c^s$ entonces $[\![t]\!]^{\mathcal{D}}_{\rho} = c^{\mathcal{D}} \in D^s$, mientras que si $t = x^s$ entonces $[\![t]\!]^{\mathcal{D}}_{\rho} = \rho(x) \in D^s$.

Para el paso inductivo, supongamos que $t = f(t_1, \ldots, t_n) \in \mathcal{T}^{\Phi}_{\Sigma}(s)$, con $f \in \mathcal{F}^{s_1, \ldots, s_n \to s'}$. Por el Lema 3.3.4(1) tenemos $t \in \mathcal{T}^{\Phi}_{\Sigma}(s')$ y $s' \ll^{\Phi} s$. Aplicando el Lema 3.3.4(2) obtenemos $t_i \in \mathcal{T}^{\Phi}_{\Sigma}(s_i), 1 \leq i \leq n$, por lo que usando hipótesis de inducción obtenemos $[\![t_i]\!]^{\mathcal{D}}_{\rho} \in D^{s_i}$. En estas condiciones $[\![t]\!]^{\mathcal{D}}_{\rho} = f^{\mathcal{D}}([\![t_1]\!]^{\mathcal{D}}_{\rho}, \ldots, [\![t_n]\!]^{\mathcal{D}}_{\rho}) \in D^{s'} \subseteq D^s$.

(2) Obsérvese que ρ' está bien definida por (1). Procedemos por inducción estructural primero sobre t y luego sobre φ. Si t es una variable o una constante la prueba resulta inmediata. Si $t = f(t'_1, \ldots, t'_q)$ entonces $[\![t\tau]\!]^{\mathcal{D}}_{\rho} = [\![f(t'_1\tau, \ldots, t'_q\tau)]\!]^{\mathcal{D}}_{\rho} = [\![f(t'_1, \ldots, t'_q)]\!]^{\mathcal{D}}_{\rho'}$ ya que, por hipótesis de inducción, $[\![t'_i\tau]\!]^{\mathcal{D}}_{\rho} = [\![t'_i]\!]^{\mathcal{D}}_{\rho'}$, $1 \leq i \leq q$.

Si $\varphi = s \sqsubseteq_{@} s'$, la prueba es inmediata. Si $\varphi = t \sqsubseteq t'$, acudimos al resultado correspondiente a los términos para obtener que $[\![t\tau]\!]^{\mathcal{D}}_{\rho} = [\![t]\!]^{\mathcal{D}}_{\rho'}$ y $[\![t'\tau]\!]^{\mathcal{D}}_{\rho} = [\![t']\!]^{\mathcal{D}}_{\rho'}$. En consecuencia, $[\![\varphi\tau]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'}$. El caso $\varphi = P(t_1, \ldots, t_n)$ se prueba de forma análoga. Para el caso de los conectivos usamos hipótesis de inducción.

(3) Sea $t'_i = t[t_i]_p$, $i = 1, 2$, y supongamos que p es una posición monótona de t (para el caso antimonótono procedemos análogamente). Razonamos por inducción sobre p. Si $p = \varepsilon$ entonces $t'_i = t_i$, $i = 1, 2$, y la prueba es immediata por la semántica y la transitividad de la estructura. Si $p = j.q$ entonces $t = f(u_1, \ldots, u_n)$, $f \in \mathcal{F}^{s_1 \ldots s_n \to s}$ y $1 \leq j \leq n$.

Por la interpretación de los símbolos de función, tenemos que $[\![u_j[t_i]_q]\!]^{\mathcal{D}}_{\rho} \in D^{s_j}$, $i = 1, 2$. Aplicando hipótesis de inducción resulta

$$[\![u_j[t_1]_q]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}}_{s_j} [\![u_j[t_2]_q]\!]^{\mathcal{D}}_{\rho}$$

y concluimos aplicando la monotonía de $f^{\mathcal{D}}$.

(4) Esta prueba es similar a la del caso (3). ∎

3.6 Tableaux básicos para LPGD

En esta sección presentamos una extensión del método de los tableaux de primer orden, que permite el uso de preórdenes para datos y que maneja los géneros adecuadamente. Para ello, no añadiremos los axiomas necesarios para caracterizar cualquier estructura preordenada, sino que formularemos reglas de expansión dentro del propio método.

Como es costumbre en los sistemas de tableaux para primer orden, extendemos Σ con constantes auxiliares. En concreto, introducimos la familia S-indexada de conjuntos numerables AC^s, $s \in S$. La clase de los literales en *LPGD* contiene las $\overline{\Sigma}$-fórmulas $P(t_1, \ldots, t_n)$, $t \sqsubseteq t', s \sqsubseteq_{@} s'$, y sus respectivos complementarios $\neg P(t_1, \ldots, t_n)$, $\neg t \sqsubseteq t', \neg s \sqsubseteq_{@} s'$.

Definición 3.6.1 *Una secuencia de tableaux para un conjunto finito no vacío de Σ-sentencias Φ es cualquier secuencia $\mathcal{T}_0$, $\mathcal{T}_1$, $\mathcal{T}_2, \ldots$ donde:*

1. *$\mathcal{T}_0$ es un árbol lineal con una rama con tantos nodos como sentencias tiene Φ. Además estos nodos están etiquetados con las sentencias de Φ.*

2. *$\mathcal{T}_{k+1}$ proviene de $\mathcal{T}_k$ mediante la aplicación de alguna de las reglas de expansión (α—ξ_2) que abajo se detallan.*

Un tableau infinito para un conjunto Φ de Σ-sentencias se define como el límite de alguna secuencia de tableaux.

Las reglas de expansión α y β son las reglas usuales en LPO, a saber α aplicada sobre $\varphi \wedge \psi$ y $\neg\neg\varphi$, y β aplicada sobre $\neg(\varphi \wedge \psi)$. Para el resto suponemos que $\mathcal{T}$ es un tableau finito para Φ y B es una rama cualquiera de $\mathcal{T}$, entonces:

(γ) *Si $\neg\exists x^s\varphi \in B$ entonces B se extiende con un nodo etiquetado con $\neg\varphi[t/x^s]$, donde t es un término básico tal que $t \in \mathcal{T}^B_{\overline{\Sigma}}(s)$.*

(δ) *Si $\exists x^s\varphi \in B$ entonces B se extiende con un nodo etiquetado con $\varphi[c^s/x^s]$, donde $c^s \in AC^s$ es nueva en B.*

(*Ref*) *B se extiende con un nuevo nodo etiquetado con $t \sqsubseteq t$, donde t es un término básico tal que $t \in \mathcal{T}^B_{\overline{\Sigma}}(sort(t))$.*

(ζ_1) *Si $t \sqsubseteq t'$, $t_1 \sqsubseteq t_2[t]_p \in B$, donde p es una posición monótona de t_2, entonces B se extiende con un nuevo nodo etiquetado con $t_1 \sqsubseteq t_2[t']_p$, si $WS(t_1 \sqsubseteq t_2[t']_p, B)$.*

(ζ_2) *Si $t' \sqsubseteq t$, $t_1 \sqsubseteq t_2[t]_p \in B$, donde p es una posición antimonótona de t_2, entonces B se extiende con un nuevo nodo etiquetado con $t_1 \sqsubseteq t_2[t']_p$, si $WS(t_1 \sqsubseteq t_2[t']_p, B)$.*

(ξ_1) *Si $P(t_1, \ldots, t_i[t]_p, \ldots, t_n)(= A)$, $t \sqsubseteq t' \in B$, donde p es una posición monótona de A en i, entonces B se extiende con un nuevo nodo etiquetado con $P(t_1, \ldots, t_i[t']_p, \ldots, t_n)$, supuesto que $WS(P(t_1, \ldots, t_i[t']_p, \ldots, t_n), B)$.*

(ξ_2) *Si $P(t_1, \ldots, t_i[t]_p, \ldots, t_n)(= A)$, $t' \sqsubseteq t \in B$, donde p es una posición antimonótona de A en i, entonces B se extiende con un nuevo nodo etiquetado con $P(t_1, \ldots, t_i[t']_p, \ldots, t_n)$, supuesto que $WS(P(t_1, \ldots, t_i[t']_p, \ldots, t_n), B)$.*

Obsérvese que γ es aplicada de tal modo que sólo se incluyen en el tableau términos bien tipados con respecto a la rama. De la misma forma operan las llamadas reglas para desigualdades entre datos *Ref*, ζ_1, ζ_2, ξ_1, ξ_2.

Definición 3.6.2 *Decimos que una rama B en un tableau está cerrada, y entonces no se prolonga ni bifurca más, cuando se detecta una contradicción atómica entre sus etiquetas. Esto significa que existen en B dos literales de la forma φ y $\neg\varphi$. En otro caso B está abierta. Un tableau está cerrado si todas sus ramas están cerradas.*

3.6.1 Corrección y completitud

La corrección del método de tableaux expresa que la insatisfactibilidad de un conjunto de fórmulas Φ es consecuencia de la existencia de un tableau cerrado para él. Como con los tableaux básicos para *LPO*, la prueba de este resultado se basa en el hecho de que la satisfactibilidad de un tableau $\mathcal{T}$, es decir la existencia en $\mathcal{T}$ de una rama satisfactible, se preserva cuando $\mathcal{T}$ es extendido por cualquiera de las reglas de expansión.

Lema 3.6.3 *La satisfactibilidad de un tableau es preservada por las reglas de expansión.*

Demostración. Para las reglas α, β, δ la demostración es análoga a *LPO*.

Para la regla γ, supongamos que $\neg\exists x^s\varphi \in B$, donde $t \in \mathcal{T}_\Sigma^B(s)$, y sea $\mathcal{D}$ un modelo de B. Por el Lema 3.5.5(1), $[\![t]\!]^\mathcal{D} \in D^s$, luego $[\![\neg\varphi[t/x^s]]\!]^\mathcal{D} = [\![\neg\varphi]\!]^\mathcal{D}_{[[\![t]\!]^\mathcal{D}/x^{s'}]} = \underline{t}$, de acuerdo con la semántica y el Lema 3.5.5(2).

El caso de la regla *Ref* es trivial, por el Lema 3.5.5(1).

Con respecto a la regla ζ_1, sean $\varphi_1 = t \sqsubseteq t'$, $\varphi_2 = t_1 \sqsubseteq t_2[t]_p$, $\psi = t_1 \sqsubseteq t_2[t']_p$, $WS(\psi, B)$ y supongamos que $\varphi_1, \varphi_2 \in B$, siendo $\mathcal{D}$ un modelo de B. Si $p = \epsilon$, es inmediato; en otro caso usamos la semántica y los Lemas 3.3.4 y 3.5.5(1) para deducir que existe un género s tal que $[\![t_2[t]_p]\!]^\mathcal{D}, [\![t_2[t']_p]\!]^\mathcal{D} \in D^s$. Ahora aplicando el Lema 3.5.5(3), obtenemos que $[\![t_2[t]_p]\!]^\mathcal{D} \sqsubseteq_s^\mathcal{D} [\![t_2[t']_p]\!]^\mathcal{D}$. Por otra parte, $WS(\psi, B)$ implica la existencia de otro género s' tal que $[\![t_1]\!]^\mathcal{D}, [\![t_2[t']_p]\!]^\mathcal{D} \in D^{s'}$. Finalizamos aplicando la transitividad de la Σ-estructura $\mathcal{D}$ a φ_2 y $[\![t_2[t]_p]\!]^\mathcal{D} \sqsubseteq_s^\mathcal{D} [\![t_2[t']_p]\!]^\mathcal{D}$.

Los casos para las reglas ζ_2, ξ_1 y ξ_2 son análogos. ∎

Teorema 3.6.4 (Corrección) *Para todo conjunto de Σ-sentencias Φ, si Φ tiene un tableau básico cerrado entonces Φ es insatisfactible.*

La dirección opuesta del teorema previo expresa la completitud del método, pero para *LPGD* se requiere una hipótesis adicional. Aquí demostramos que si un conjunto de fórmulas Φ no tiene ningún tableau cerrado, podemos construir de forma sistemática un tableau en el que cada rama abierta B tenga ciertas propiedades de saturación que nos permitan definir un modelo de $\{\varphi \in B / WS(\varphi, B)\}$. Obsérvese que las fórmulas pueden dejar de estar mal tipadas según avancemos en la expansión de la rama, pero una vez bien tipadas, no pueden dejar de estarlo. Evitamos las fórmulas mal tipadas porque no tenemos garantías de poder hacerlas satisfactibles, ni siquiera en las estructuras que verifican estas propiedades de saturación. Las propiedades de saturación pueden ser definidas *à la Hintikka* como sigue.

Definición 3.6.5 (Conjunto de Hintikka) *Un conjunto de $\overline{\Sigma}$-sentencias H es un conjunto de Hintikka si satisface las siguientes condiciones:*

1. *Si $\varphi_1 \wedge \varphi_2 \in H$ entonces $\varphi_1, \varphi_2 \in H$.*

2. *Si $\neg(\varphi_1 \wedge \varphi_2) \in H$ entonces $\neg\varphi_1 \in H$ o $\neg\varphi_2 \in H$.*

3. *Si $\neg\neg\varphi \in H$ entonces $\varphi \in H$.*

4. *Si $\exists x^s \varphi \in H$ entonces existe una constante $c^s \in AC^s$ tal que $\varphi[c^s/x^s] \in H$.*

5. *Si $\neg\exists x^s \varphi \in H$ entonces $\neg\varphi[t/x^s] \in H$, para todo término básico $t \in \mathcal{T}_{\overline{\Sigma}}^H(s)$.*

6. *Para todo término básico $t \in \mathcal{T}_{\overline{\Sigma}}^H(sort(t))$, $t \sqsubseteq t \in H$.*

7. *Sean $t, t', t_1, t_2 \in T(\overline{\Sigma})$:*

   (i) si $t \sqsubseteq t'$ (resp. $t' \sqsubseteq t$), $t_1 \sqsubseteq t_2[t]_p \in H$, $WS(t_1 \sqsubseteq t_2[t']_p, H)$ y p es una posición monótona (resp. antimonótona) de t_2, entonces $t_1 \sqsubseteq t_2[t']_p \in H$

   (ii) si $t \sqsubseteq t'$ (resp. $t' \sqsubseteq t$), $P(t_1, \ldots, t_i[t]_p, \ldots, t_n)(= A) \in H$, $WS(P(t_1, \ldots, t_i[t']_p, \ldots, t_n), H)$ y p es una posición monótona (resp. antimonótona) de A en i, entonces $P(t_1, \ldots, t_i[t']_p, \ldots, t_n) \in H$.

8. *H es coherente, es decir ningún átomo y su negación pertenecen a H.*

Lema 3.6.6 *Si H es un conjunto de Hintikka entonces el sistema $\mathcal{D} = \{(D^s, \sqsubseteq_s^{\mathcal{D}}) | s \in S\} \cup \{\perp\}$, donde $D^s = \{t | t \in \mathcal{T}_{\overline{\Sigma}}^H(s), t \text{ básico}\}$ y $t \sqsubseteq_s^{\mathcal{D}} t' \Leftrightarrow_{def} t \sqsubseteq t' \in H$, para todo $t, t' \in D^s$, con las interpretaciones*

- *$f^{\mathcal{D}} : D^{s_1} \times \ldots \times D^{s_n} \to D^s$, $f^{\mathcal{D}}(t_1, \ldots, t_n) =_{def} f(t_1, \ldots, t_n)$. En particular, $c^{\mathcal{D}} =_{def} c$.*

- *$P^{\mathcal{D}} : D^{s_1} \times \ldots \times D^{s_n} \to \{\underline{t}, \underline{f}\}$, $P^{\mathcal{D}}(t_1, \ldots, t_n) =_{def} \begin{cases} \underline{t} & \text{si } P(t_1, \ldots, t_n) \in H \\ \underline{f} & \text{en otro caso} \end{cases}$*

es una $\overline{\Sigma}$-estructura.*

Demostración. En primer lugar, obsérvese que D^s no es vacío ya que todo género tiene infinitas constantes. Por otra parte, $\sqsubseteq_s^{\mathcal{D}}$ es un preorden porque:

1. $t \sqsubseteq_s^{\mathcal{D}} t$, para todo $t \in D^s$, por el Lema 3.3.4 y la Definición 3.6.5(6).

2. Si $t \sqsubseteq_s^{\mathcal{D}} t'$ y $t' \sqsubseteq_s^{\mathcal{D}} t''$ entonces $t, t', t'' \in \mathcal{T}_{\overline{\Sigma}}^H(s)$ y por tanto $WS(t \sqsubseteq t'', H)$. En estas condiciones $t \sqsubseteq t'' \in H$, por la Definición 3.6.5(7)(i), y en consecuencia $t \sqsubseteq_s^{\mathcal{D}} t''$.

*Usualmente llamada de *Herbrand*.

El sistema además es transitivo porque si $t, t'' \in D^{s''}$, $t \sqsubseteq_s^{\mathcal{D}} t', t' \sqsubseteq_{s'}^{\mathcal{D}} t''$ entonces $t \sqsubseteq t'$, $t' \sqsubseteq t'' \in H$ y $WS(t \sqsubseteq t'', H)$. Por tanto, de la Definición 3.6.5(7)(i), concluimos $t \sqsubseteq t'' \in H$ luego $t \sqsubseteq_{s''}^{\mathcal{D}} t''$.

Finalmente supongamos $m(f)(i) = 0$, $f \in F^{s_1,\ldots,s_n \to s}$. Sean $t_j \in D^{s_j}$, $j \in \{1, \ldots, n\}$, y $t_i \sqsubseteq_{s_i}^{\mathcal{D}} t_i'$. Por la Definición 3.6.5(6) tenemos $f(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq f(t_1, \ldots, t_i, \ldots, t_n) \in H$. Además $WS(f(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq f(t_1, \ldots, t_i', \ldots, t_n), H)$ ya que $f(t_1, \ldots, t_i, \ldots, t_n)$, $f(t_1, \ldots, t_i', \ldots, t_n) \in D^s$. En estas condiciones llegamos a que $f(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq f(t_1, \ldots, t_i', \ldots, t_n) \in H$, y entonces $f^{\mathcal{D}}(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq_s^{\mathcal{D}} f^{\mathcal{D}}(t_1, \ldots, t_i', \ldots, t_n)$. Análogamente para $m(f)(i) = 1$ y la interpretación de predicados $P^{\mathcal{D}}$. ■

Teorema 3.6.7 *Para todo conjunto de Hintikka H existe un modelo del conjunto de sentencias $\{\varphi \in H / WS(\varphi, H)\}$.*

Demostración. Sea $\mathcal{D}$ la $\overline{\Sigma}$-estructura que acabamos de definir. Es fácil probar que $t \in D^{sort(t)} \Longleftrightarrow [\![t]\!]^{\mathcal{D}} = t$, para todo término básico $t \in T(\overline{\Sigma})$. En estas condiciones probamos el teorema por inducción sobre φ. Veamos algunos casos :

- $\varphi = t_1 \sqsubseteq t_2$. Ya que $WS(\varphi, H)$, existe un s tal que $[\![t_i]\!]^{\mathcal{D}} = t_i \in D^s$, $i = 1, 2$; por lo que $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$, por la definición de $\sqsubseteq_s^{\mathcal{D}}$.

- $\varphi = P(t_1, \ldots, t_n)$. Ya que $WS(\varphi, H)$, tenemos $[\![t_i]\!]^{\mathcal{D}} = t_i \in D^{s_i}$, $i = 1, \ldots, n$, $P \in P^{s_1,\ldots,s_n}$; por lo tanto, $[\![\varphi]\!]^{\mathcal{D}} = P^{\mathcal{D}}(t_1, \ldots, t_n) = \underline{t}$, por la definición de $P^{\mathcal{D}}$.

- $\varphi = \neg(s \sqsubseteq_{@} s')$. $WS(\varphi, H)$ no es posible porque H es coherente.

- $\varphi = \neg(t_1 \sqsubseteq t_2)$. Por la coherencia de H, $(t_1 \sqsubseteq t_2) \notin H$, por lo que, por la definición de $\mathcal{D}$, $[\![t_1 \sqsubseteq t_2]\!]^{\mathcal{D}} = \underline{f}$.

- $\varphi = \exists x^s \psi$. Ya que $\varphi \in H$, existe $c^s \in AC^s$ tal que $\psi[c^s/x^s] \in H$, por la Definición 3.6.5(4). Es más, tenemos $WS(\psi[c^s/x^s], H)$, por ser $WS(\varphi, H)$ y el Lema 3.4.2(2). Por hipótesis de inducción $[\![\psi[c^s/x^s]]\!]^{\mathcal{D}} = \underline{t}$ y por tanto $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$, por el Lema 3.5.5(2).

- $\varphi = \neg\exists x^s \psi$. Sea $t \in D^s$, entonces tenemos que $\neg\psi[t/x^s] \in H$, por la Definición 3.6.5(5). Además $WS(\neg\psi[t/x^s], H)$, ya que $WS(\varphi, H)$. En consecuencia:

$$
\begin{aligned}
[\![\neg\psi]\!]^{\mathcal{D}}_{[[\![t]\!]^{\mathcal{D}}/x^s]} &= [\![\neg\psi[t/x^s]]\!]^{\mathcal{D}} \text{ (por el Lema 3.5.5(2))} \\
&= \underline{t} \text{ (por hipótesis de inducción).} \quad \blacksquare
\end{aligned}
$$

Ahora definimos la noción de regla de construcción de tableaux *justa* como un procedimiento de construcción de tableaux tal que toda rama no cerrada llega a contener un conjunto de Hintikka.

Definición 3.6.8 *Una regla de construcción de tableaux básicos es un procedimiento $\mathcal{R}$ tal que, dado un tableau finito $\mathcal{T}$, produce como respuesta una refutación, si no hay continuación posible, o un nuevo tableau $\mathcal{T}'$ que resulte de la aplicación de una regla de expansión de tableaux básicos a $\mathcal{T}$. Una regla de construcción de tableaux básicos $\mathcal{R}$ es justa si, para cualquier secuencia de tableaux $\mathcal{T}_0$, $\mathcal{T}_1, \ldots$ para un conjunto de Σ-sentencias Φ, construida de acuerdo con $\mathcal{R}$, se verifican las siguientes propiedades:*

1. *A toda sentencia que no sea un literal y que aparezca en una rama abierta de cualquier $\mathcal{T}_i$, se le aplicará en algún momento la regla de expansión adecuada.*
2. *A cualquier aparición de una sentencia de clase Gamma en una rama abierta de cualquier $\mathcal{T}_i$, se le aplicará la regla γ un número cualquiera de veces.*
3. *$\mathcal{R}$ en algún momento introduce $t \sqsubseteq t$ en toda rama abierta B de cualquier $\mathcal{T}_i$, para todo $\overline{\Sigma}$-término básico t tal que $t \in \mathcal{T}_{\Sigma}^{B}(sort(t))$.*
4. *A cualquier aparición de un par de átomos en una rama abierta de cualquier $\mathcal{T}_i$, para el que las reglas ζ_1, ζ_2, ξ_1 o ξ_2 puedan ser utilizadas, se le aplicará la regla adecuada en algún momento.*

Para probar la completitud de una regla de construcción de tableaux básicos justa, pediremos al conjunto de fórmulas Φ, para el cual se está construyendo el tableau, que esté bien tipado con respecto a cualquier rama que contenga un conjunto de Hintikka. Esta condición se consigue, obviamente, si Φ está bien tipado con respecto a la información del orden entre géneros estáticamente obtenida de Φ, donde este concepto se define como sigue.

Definición 3.6.9 *La información estática positiva de géneros de una fórmula φ, escrito $\mathcal{P}(\varphi)$, se define recursivamente por:*

1. $\mathcal{P}(s \sqsubseteq_{@} s') = \{s \sqsubseteq_{@} s'\}$.
2. $\mathcal{P}(\varphi) = \emptyset$, *para cualquier otro literal* φ.
3. $\mathcal{P}(\varphi \wedge \psi) = \mathcal{P}(\varphi) \cup \mathcal{P}(\psi)$.
4. $\mathcal{P}(\neg(\varphi \wedge \psi)) = \emptyset$.
5. $\mathcal{P}(\exists x\varphi) = \mathcal{P}(\varphi)$.
6. $\mathcal{P}(\neg\exists x\varphi) = \mathcal{P}(\neg\varphi)$.
7. $\mathcal{P}(\neg\neg\varphi) = \mathcal{P}(\varphi)$.

Para un conjunto de fórmulas Φ definimos $\mathcal{P}(\Phi) = \bigcup_{\varphi\in\Phi} \mathcal{P}(\varphi)$.

Teorema 3.6.10 (Completitud) *Sea $\mathcal{R}$ una regla de construcción de tableaux básicos justa. Si Φ es un conjunto finito de Σ-sentencias insatisfactible y $WS(\Phi, \mathcal{P}(\Phi))$ entonces Φ tiene un tableau básico cerrado construido de acuerdo con $\mathcal{R}$.*

Demostración. Supongamos que Φ no tiene un tableau cerrado. Sea $\mathcal{T}$ el tableau para Φ construido de acuerdo con $\mathcal{R}$. En estas condiciones, $\mathcal{T}$ no es cerrado y por tanto tiene una rama B que es un conjunto de Hintikka y que contiene a Φ. Ahora bien, gracias a que $WS(\Phi, \mathcal{P}(\Phi))$ y al Teorema 3.6.7, podemos concluir que Φ tiene un modelo. ∎

3.7 Tableaux con Variables Libres

En esta sección presentamos un método de tableaux con variables libres para *LPGD* en el que las fórmulas de la clase *Gamma* introducen variables libres, en lugar de términos bien tipados. Estas variables libres son *correctamente* instanciadas a la hora de cerrar una rama, o de expandirla aplicando reglas de desigualdad entre datos. Aquí la palabra *correctamente* significa que la sustitución debe asignar a cada variable libre x^s, un término $t \in \mathcal{T}_\Sigma^B(s)$, para toda rama B donde x^s aparezca. Las sustituciones *correctas* son necesarias para mantener la corrección del método, en otro caso el valor semántico de las variables libres podría quedar indefinido, lo cual no está permitido en nuestra aproximación.

Definición 3.7.1 *Una sustitución σ se dice bien tipada con respecto a un tableau $\mathcal{T}$, escrito $WS(\sigma, \mathcal{T})$, si $WS(\sigma|_{var(B)}, B)$, para toda rama B del tableau $\mathcal{T}$.*

El nuevo método cuenta con las reglas α, β ya conocidas más las siguientes. Para δ', supondremos que la signatura extendida ya con constantes $\overline{\Sigma}$, también contiene una colección de conjuntos numerables $SF^{s_1,\ldots,s_l \to s}$, $s_1, \ldots, s_l, s \in S$, de símbolos de función de Skolem, monótonos en todos sus argumentos. Sea $\mathcal{T}$ un tableau para *LPGD* con variables libres y B una rama de $\mathcal{T}$ entonces:

(γ') Si $\neg\exists x^s \varphi \in B$ y $s' \ll^B s$ entonces B se extiende con un nuevo nodo etiquetado con $\neg\varphi[y^{s'}/x^s]$, donde $y^{s'}$ es una nueva variable libre en el tableau.

(δ') Si $\exists x^s \varphi \in B$ entonces B se extiende con un nuevo nodo etiquetado con la fórmula $\varphi[f(x_1^{s_1}, \ldots, x_n^{s_n})/x^s]$, donde $f \in SF^{s_1,\ldots,s_n \to s}$ no ha sido usado antes en B y $x_1^{s_1}, \ldots, x_n^{s_n}$ son las variables libres de B.

(ζ_1', ζ_2') Si $t \sqsubseteq t'$ ($t' \sqsubseteq t$ para ζ_2'), $t_1 \sqsubseteq t_2[t'']_p \in B$, σ es un unificador de máxima generalidad de t y t'', $WS(\sigma, \mathcal{T})$ y p es una posición monótona (antimonótona para ζ_2') de t_2, entonces se aplica σ a $\mathcal{T}$ y se extiende $B\sigma$ con un nuevo nodo etiquetado con $(t_1 \sqsubseteq t_2[t']_p)\sigma$, si $WS((t_1 \sqsubseteq t_2[t']_p)\sigma, B)$.

(ξ_1', ξ_2') Si $t \sqsubseteq t'$ ($t' \sqsubseteq t$ para ξ_2'), $P(t_1, \ldots, t_i[t'']_p, \ldots, t_n)(= A) \in B$, σ es un unificador de máxima generalidad de t y t'', $WS(\sigma, \mathcal{T})$ y p es una posición monótona (antimonótona para ξ_2') de A en i, entonces se aplica σ a $\mathcal{T}$ y se extiende $B\sigma$ con un nuevo nodo etiquetado con $P(t_1, \ldots, t_i[t']_p, \ldots, t_n)\sigma$, si $WS(P(t_1, \ldots, t_i[t']_p, \ldots, t_n)\sigma, B)$.

(*Refv*) Se extiende B con un nuevo nodo etiquetado con $x^s \sqsubseteq x^s$, donde x^s es una variable nueva en $\mathcal{T}$.

(*Reff*) Se extiende B con un nuevo nodo etiquetado con $f(x_1^{s_1},\ldots,x_n^{s_n}) \sqsubseteq f(x_1^{s_1},\ldots,x_n^{s_n})$, donde f es un símbolo de función de $\overline{\Sigma}$, $x_1^{s_1},\ldots,x_n^{s_n}$ son variables nuevas en B y $WS(f(x_1^{s_1},\ldots,x_n^{s_n}) \sqsubseteq f(x_1^{s_1},\ldots,x_n^{s_n}), B)$.

Para cerrar un tableau calculamos un unificador de máxima generalidad y nos preguntamos si está bien tipado con respecto a todo el tableau.

Definición 3.7.2 *Un tableau $\mathcal{T}$ con variables libres y ramas $B_1,\ldots,B_k$ está cerrado si existe un conjunto de ecuaciones de fórmulas atómicas $M = \{\varphi_1 \simeq \psi_1,\ldots,\varphi_k \simeq \psi_k\}$ tal que M es unificable, $WS(umg(M),\mathcal{T})$ y $\varphi_i, \neg\psi_i \in B_i, 1 \le i \le k$.*

Obsérvese que, dada una sustitución θ y un tableau finito $\mathcal{T}$, el problema ¿$WS(\theta,\mathcal{T})$? es decidible porque la cuestión ¿$t \in \mathcal{T}_{\overline{\Sigma}}^B(s)$? puede también resolverse, para toda rama B de $\mathcal{T}$.

3.7.1 Corrección

La corrección se basa en que la existencia de un modelo para un tableau $\mathcal{T}$ se mantiene cuando éste es extendido, en el sentido siguiente.

Definición 3.7.3 *Un tableau $\mathcal{T}$ es básicamente satisfactible si existe una Σ-estructura $\mathcal{D}$ tal que $\mathcal{D} \models \mathcal{T}\tau$, para toda sustitución τ básica que cumpla $var(\mathcal{T}) \subseteq dom(\tau)$ y verifique $WS(\tau,\mathcal{T})$. En este caso decimos que $\mathcal{D}$ es un modelo básico de $\mathcal{T}$.*

Dado que muchas de las reglas del método con variables libres aplican unificadores de máxima generalidad bien tipados con respecto al tableau, demostramos previamente el siguiente lema.

Lema 3.7.4 *Dados una Σ-estructura $\mathcal{D}$, que es modelo básico de un tableau $\mathcal{T}$, y una sustitución τ tal que $WS(\tau,\mathcal{T})$, entonces $\mathcal{D}$ es modelo básico de $\mathcal{T}\tau$.*

Demostración. Si θ es una sustitución básica tal que $var(\mathcal{T}\tau) \subseteq dom(\theta)$ y $WS(\theta,\mathcal{T}\tau)$, basta con probar que $\tau\theta$ es una sustitución básica para $var(\mathcal{T})$ tal que $WS(\tau\theta,\mathcal{T})$. Resulta inmediato que $\tau\theta$ es básica para $var(\mathcal{T})$. Para demostrar la buena tipificación, sea $x^s \in B$, siendo B una rama de $\mathcal{T}$, entonces $t = x^s\tau \in \mathcal{T}_\Sigma^B(s)$, ya que $WS(\tau,\mathcal{T})$. Además deducimos $WS(\theta|_{var(t)}, B)$, debido a que $WS(\theta,\mathcal{T}\tau)$. En estas condiciones, $t\theta = t\theta|_{var(t)} \in \mathcal{T}_\Sigma^B(s)$, por el Lema 3.4.2(1). ■

Lema 3.7.5 *La existencia de un modelo básico para un tableau es preservada por las reglas de expansión.*

Demostración. Para las reglas α, β, δ', *Refv* y *Reff*, la demostración es inmediata.

Para la regla γ, supongamos que $\neg\exists x^s\varphi \in B, s' \ll^B s$ y que hemos extendido B con $\neg\varphi[y^{s'}/x^s]$ para formar $\mathcal{T}'$. Sea θ una sustitución básica que cumple $var(\mathcal{T}') \subseteq dom(\theta)$

y verifica $WS(\theta, \mathcal{T}')$, entonces θ es básica para $var(\mathcal{T})$ y $WS(\theta, \mathcal{T})$. Sea $\mathcal{D}$ el modelo básico de $\mathcal{T}$, entonces $\mathcal{D} \models \mathcal{T}\theta$. Supongamos directamente que $\mathcal{D} \models B\theta$ entonces $\mathcal{D} \models (\neg\exists x^s \varphi)\theta$. En estas condiciones, si $t = \theta(y^{s'})$ tendremos $[\![(\neg\varphi[y^{s'}/x^s])\theta]\!]^{\mathcal{D}} = [\![\neg\varphi\theta[t/x^s]]\!]^{\mathcal{D}} = [\![(\neg\varphi)\theta]\!]^{\mathcal{D}}_{[[\![t]\!]^{\mathcal{D}}/x^s]}$ (por Lema 3.5.5(2)) $= \underline{t}$ (por la semántica).

Con respecto a la regla ζ_1', sean $\varphi_1 = t \sqsubseteq t'$, $\varphi_2 = t_1 \sqsubseteq t_2[t'']_p$ y $\psi = t_1 \sqsubseteq t_2[t']_p$. Sea $\sigma = umg(t, t'')$ tal que $WS(\sigma, \mathcal{T})$. Aplicando el Lema 3.7.4 tenemos que existe $\mathcal{D}$, modelo básico de $\mathcal{T}\sigma$. Sea θ una sustitución básica para $var(\mathcal{T}\sigma)$ y supongamos que $\mathcal{D} \models B\sigma\theta$. Usando el Lema 3.5.5(3) y la transitividad de $\mathcal{D}$ concluimos que $\mathcal{D} \models \psi\sigma\theta$, como en el caso básico.

Los casos para las reglas ζ_2', ξ_1' y ξ_2' son análogos. ∎

Teorema 3.7.6 (Corrección) *Para todo conjunto Φ de Σ-sentencias, si Φ tiene un tableau con variables libres cerrado entonces Φ es insatisfactible.*

Demostración. Supongamos que Φ es satisfactible. Sea $\mathcal{T}$ el tableau cerrado y σ el unificador de máxima generalidad, con $WS(\sigma, \mathcal{T})$, que cierra $\mathcal{T}$. Entonces, aplicando el Lema 3.7.5, obtenemos que $\mathcal{T}\sigma$ admite un modelo básico, lo cual resulta imposible. ∎

3.7.2 Completitud

En esta sección probamos la completitud del método de tableaux con variables libres mostrando que los tableaux básicos cerrados pueden ser elevados a tableaux con variables libres cerrados. Antes de presentar el correspondiente lema de elevación, demostramos que el resultado del Lema 3.4.4(3) puede extenderse al caso de los tableaux de forma natural.

Lema 3.7.7 *Sean t, t' dos términos, $\mathcal{T}$ un tableau, φ una fórmula y τ una sustitución que respeta los géneros de forma estática tal que $WS(\tau, \mathcal{T})$. Si $t\tau = t'\tau$ y $\theta = umg(t, t')$ entonces $WS(\theta, \mathcal{T})$.*

Demostración. Sea $\theta = [t_1/x_1{}^{s_1}, \ldots, t_n/x_n{}^{s_n}]$. Supongamos que $x_i{}^{s_i} \in var(B)$, para algún $1 \leq i \leq n$ y una rama B de $\mathcal{T}$. Por la generalidad de θ obtenemos que $t_i\tau = x_i{}^{s_i}\tau$. Además, usando $WS(\tau, \mathcal{T})$, obtenemos $t_i\tau \in \mathcal{T}_\Sigma^B(s_i)$, por lo que, aplicando el Lema 3.4.4(1), concluimos $t_i \in \mathcal{T}_\Sigma^B(s_i)$. ∎

Lema 3.7.8 (Elevación) *Sean $\mathcal{T}$ un tableau con variables libres y τ, una sustitución básica tal que $var(\mathcal{T}) \subseteq dom(\tau)$, $WS(\tau, \mathcal{T})$ y τ respeta los géneros de forma estática. Si $\mathcal{T}\tau$ puede cerrarse mediante las reglas Ref, ζ_1, ζ_2, ξ_1 y ξ_2, entonces $\mathcal{T}$ puede cerrarse usando Refv, Reff, ζ_1', ζ_2', ξ_1' y ξ_2'.*

Demostración. Por inducción sobre el número n de aplicaciones de *Ref*, ζ_1, ζ_2, ξ_1 y ξ_2 usadas para cerrar $\mathcal{T}\tau$. Si $n = 0$ entonces $\mathcal{T}\tau$ está cerrado y $\mathcal{T}$ está también cerrado por el Lema 3.7.7.

Para el paso inductivo, sean $B_1, \ldots, B_l$ las ramas de $\mathcal{T}$, y B^* la rama obtenida al extender, digamos $B_i\tau$, con la primera aplicación de *Ref*, ζ_1, ζ_2, ξ_1 y ξ_2. La idea de la demostración es extender $\mathcal{T}$ a un tableau $\mathcal{T}'$ con ramas $B'_1, \ldots, B'_l$, usando las reglas permitidas en el lema, y extender τ a una sustitución básica τ' que respete los géneros de forma estática, tal que $var(\mathcal{T}') \subseteq dom(\tau')$ y $WS(\tau', \mathcal{T}')$, y tal que $B'_j\tau' = B_j\tau$, $1 \leq j \leq l$, $j \neq i$, y $B'_i\tau' \supseteq B^*$. En estas condiciones la demostración se completa aplicando hipótesis de inducción a $\mathcal{T}'\tau'$.

Analicemos entonces los distintos casos posibles, dependiendo de la regla utilizada para obtener B^*. En lo que sigue, omitiremos el subíndice i para simplificar la notación.

El caso de la regla *Ref* es inmediato, usando *Refv* en B con una variable nueva en $\mathcal{T}$. Veamos el caso de ξ_1, probándose de forma similar el resto de los casos. Sin pérdida de generalidad supongamos que $B\tau$ es extendida como sigue :

$$
\begin{array}{ccc}
\underline{B} & \underline{B\tau} & \underline{B^*} \\
\vdots & \vdots & B\tau \\
t \sqsubseteq t' & t\tau \sqsubseteq t'\tau & | \\
P(t_1, \ldots, t_k) & P(t_1\tau, \ldots, t_k\tau) & P(t_1\tau, \ldots, t_i\tau[t'\tau]_p, \ldots, t_k\tau)
\end{array}
$$

donde $t_i\tau|_p = t\tau$, p es una posición monótona de $P(t_1\tau, \ldots, t_k\tau)$ en i, y $WS(P(t_1\tau, \ldots, t_i\tau[t'\tau]_p, \ldots, t_k\tau), B)$. Como se prueba en el siguiente lema, existe $q \in Pos(t_i)$, $q \leq p$, tal que B puede ser extendida a una rama $B'' \in \mathcal{T}''$, usando sólo *Reff*, ζ'_1 y ζ'_2, tal que $B \subseteq B''$ y B'' contiene también una desigualdad $a_q \sqsubseteq a'_q$ ($a'_q \sqsubseteq a_q$, si q es una posición antimonótona de $P(t_1, \ldots, t_n)$ en i), y existe una sustitución básica τ' que extiende τ tal que:

i) $a'_q\tau' = (t_i\tau|_q)[t'\tau]_{p-q}$

ii) τ' unifica a_q y $t_i|_q$.

iii) $var(\mathcal{T}'') \subseteq dom(\tau')$, $WS(\tau', \mathcal{T}'')$ y τ' respeta los géneros de forma estática.

De *ii)* y *iii)* deducimos la existencia de $\sigma = umg(a_q, t_i|_q)$ y $WS(\sigma, \mathcal{T}'')$, por el Lema 3.7.7.

Además tenemos $WS(P(t_1, \ldots, t_i[a'_q]_q, \ldots, t_k)\sigma, B)$. En efecto:

$$
\begin{array}{rcl}
WS(P(t_1\tau, \ldots, t_i\tau[t'\tau]_p, \ldots, t_k\tau), B) & \Rightarrow & WS(P(t_1\sigma\tau', \ldots, (t_i[a'_q]_q)\sigma\tau', \ldots, t_k\sigma\tau'), B) \\
 & & \text{(por } i) \text{ y } ii)) \\
 & \Rightarrow & WS(P(t_1, \ldots, t_i[a'_q]_q, \ldots, t_k)\sigma, B) \\
 & & \text{(por el Lema 3.4.4(2))}
\end{array}
$$

En esta situación podemos aplicar ξ'_1 a $a_q \sqsubseteq a'_q$ (o ξ'_2 a $a'_q \sqsubseteq a_q$) y $P(t_1, \ldots, t_k)$ para obtener $B' = B''\sigma \cup P(t_1\sigma, \ldots, (t_i[a'_q]_q)\sigma, \ldots, t_k\sigma)$. Para cualquier otra rama B'_j distinta de B se tiene que $B'_j = B_j\sigma$.

Ahora veamos que hemos llegado a lo que buscábamos; en efecto, para $j \neq i$ se tiene que $B'_j\tau' = B_j\sigma\tau' = B_j\tau$, y para B', que $B'\tau' \supseteq B^*$ pues:

- $B''\sigma\tau' = B''\tau'$ (por *ii)*) $\supseteq B\tau' = B\tau$ (ya que τ' coincide con τ en B).

- $P(t_1\sigma\tau', \ldots, (t_i[a'_q]_q)\sigma\tau', \ldots, t_k\sigma\tau') = P(t_1\tau, \ldots, t_i\tau[a'_q\tau']_q, \ldots, t_k\tau)$
  (ya que $q \in Pos(t_i)$, τ' coincide con τ en B, y $ii)$)
  $= P(t_1\tau, \ldots, t_i\tau[(t_i\tau|_q)[t'\tau]_{p-q}]_q, \ldots, t_k\tau)$
  (por $i)$)
  $= P(t_1\tau, \ldots, t_i\tau[t'\tau]_p, \ldots, t_k\tau)$

Además, la inducción puede seguir aplicándose porque $WS(\tau', \mathcal{T}')$ se deduce de $WS(\tau', \mathcal{T}'')$. En efecto, sea B_1 una rama cualquiera de $\mathcal{T}'$ y $x^s \in var(B_1)$; sea B_2 la rama correspondiente a B_1 en $\mathcal{T}''$. Si $x^s \in var(B_2)$ hemos acabado. Si no, existe $y^{s'} \in var(B_2)$ tal que $x^s \in var(\sigma(y^{s'}))$. Entonces, por la generalidad de σ, tenemos que $y^{s'}\sigma\tau' = y^{s'}\tau' \in \mathcal{T}_\Sigma^{B_2}(s')$, ya que $WS(\tau', \mathcal{T}'')$. Aplicando el Lema 3.3.4(1) y (2), se puede probar que si $u \in \mathcal{T}_\Sigma^{B_1}(sort(u))$ entonces $u|_q \in \mathcal{T}_\Sigma^{B_1}(sort(u|_q))$, para todo término u y toda posición q de u; así usamos que τ' respeta los géneros de forma estática para concluir que $x^s\tau' \in \mathcal{T}_\Sigma^{B_1}(s)$.
■

El anterior lema prueba que la extensión de $\mathcal{T}\tau$ se puede elevar incluso en el caso de que la regla de extensión haya sido aplicada en una nueva posición introducida por τ. Esto es posible sin más que construir la inecuación deseada $a_q \sqsubseteq a'_q$ mediante sucesivas aplicaciones de ζ'_1, ζ'_2 y *Reff*.

Lema 3.7.9 *Usando la notación y condiciones del lema anterior, si $t_i\tau|_p = t\tau$ entonces existe una posición q de t_i, $q \leq p$, tal que $\mathcal{T}$ puede expandirse usando únicamente Reff, ζ'_1 y ζ'_2, de tal modo que sólo cambia la rama B que es extendida a una rama $B' \in \mathcal{T}'$ tal que $B' \supseteq B$, y B' contiene también, para cada posición r, $q \leq r \leq p$, una desigualdad $a_r \sqsubseteq a'_r$ o $a'_r \sqsubseteq a_r$, según sea r una posición monótona o antimonótona de $P(t_1\tau, \ldots, t_k\tau)$ en i, respectivamente, y existe una sustitución básica τ' que extiende τ tal que:*

1. *$a_r\tau' = (t_i\tau|_r)[t\tau]_{p-r}, a'_r\tau' = (t_i\tau|_r)[t'\tau]_{p-r}$*
2. *τ' es un unificador de a_q y $t_i|_q$.*
3. *$var(\mathcal{T}') \subseteq dom(\tau')$, $WS(\tau', \mathcal{T}')$ y τ' respeta los géneros de forma estática.*

Demostración. Sea q la mayor posición de t_i tal que $q \leq p$. La existencia de B' y τ' verificando (3) y (1) se prueba por inducción sobre la longitud de r. A partir de esto se deduce (2) de forma inmediata.

Si $r = p$ entonces $t \sqsubseteq t'$ puede usarse como $a_r \sqsubseteq a'_r$ y no es necesaria ninguna extensión de B o τ. En este caso, (3) es trivial y (1) se tiene porque $a_p\tau' = t\tau = (t_i\tau|_p)[t\tau]_\varepsilon$ y $a'_p\tau' = t'\tau = (t_i\tau|_p)[t'\tau]_\varepsilon$.

Para el paso de inducción, asumamos que tenemos B'' y τ'' para $q < r \leq p$. Veamos cómo extender B'' para incorporar una desigualdad para la posición q. Sea $p - q = j.r'$ para alguna $j \in N$ y alguna cadena r'; supongamos que $l = q.j$ es una posición monótona de $P(t_1\tau, \ldots, t_k\tau)$ en i, por lo que, por hipótesis de inducción, $a_l \sqsubseteq a'_l$ (†)

está en $B'' \in \mathcal{T}''$ y se tiene (1) para l. Ya que $l \in Pos(t_i\tau)$, existe un símbolo de función f de aridad m ($\geq j$), tal que $t_i\tau|_q = f(u_1, \ldots, u_m)$, para ciertos términos básicos $u_1, \ldots, u_m$ con géneros estáticos $s_1, \ldots, s_m$ (además $u_i \in \mathcal{T}_\Sigma^B(s_i)$, por ser $WS(\tau, \mathcal{T})$, por el Lema 3.3.4(1) y (2)). Extendemos B'' aplicando *Reff* con la fórmula $f(z_1^{s_1}, \ldots, z_m^{s_m}) \sqsubseteq f(z_1^{s_1}, \ldots, z_m^{s_m})$ (‡), siendo $z_1^{s_1}, \ldots, z_m^{s_m}$ variables nuevas en $\mathcal{T}''$. La regla puede ser aplicada porque $WS(f(z_1^{s_1}, \ldots, z_m^{s_m}) \sqsubseteq f(z_1^{s_1}, \ldots, z_m^{s_m}), B)$ es consecuencia de la buena tipificación de τ y por tanto de $t_i\tau|_q$ con respecto a B.

Supongamos ahora que $m(f)(j) = 0$ y consideremos $\sigma' = [a_l/z_j]$. Entonces tenemos:

- $\sigma = umg(a_l, z_j)$.
- $WS(\sigma, \mathcal{T}'')$. En efecto, puesto que z_j es nueva en el tableau, es suficiente probar que $a_l \in \mathcal{T}_\Sigma^B(s_j)$. Tenemos que $a_l\tau'' = u_j[t\tau]_{p-l} = u_j$, porque $t_i\tau|_p = t\tau$. Recordemos que $u_j \in \mathcal{T}_\Sigma^B(s_j)$; así acabamos usando el Lema 3.4.4(1) para concluir $a_l \in \mathcal{T}_\Sigma^B(s_j)$. Por otra parte, $a'_l \in \mathcal{T}_\Sigma^B(s_j)$ porque $a_l\tau'' = u_j[t'\tau]_{p-l}$, $u_j[t'\tau]_{p-l} \in \mathcal{T}_\Sigma^B(s_j)$ y el Lema 3.4.4(1).
- $WS(f(z_1, \ldots, z_{j-1}, a_l, z_{j+1}, \ldots, z_m) \sqsubseteq f(z_1, \ldots, z_{j-1}, a'_l, z_{j+1}, \ldots, z_m), B)$ como consecuencia de los puntos anteriores.

En estas condiciones ζ'_1 puede ser aplicada a (†) y (‡) mediante σ, produciendo la siguiente rama B':

$$\begin{array}{c} B'' \\ | \\ f(z_1, \ldots, z_{j-1}, a_l, z_{j+1}, \ldots, z_m) \sqsubseteq f(z_1, \ldots, z_{j-1}, a_l, z_{j+1}, \ldots, z_m) \\ | \\ f(z_1, \ldots, z_{j-1}, a_l, z_{j+1}, \ldots, z_m) \sqsubseteq f(z_1, \ldots, z_{j-1}, a'_l, z_{j+1}, \ldots, z_m) \ (\S) \end{array}$$

El resto de las ramas no cambian pues z_j es nueva. Ahora si tomamos $\tau' = \tau''[u_1/z_1^{s_1}, \ldots, u_{j-1}/z_{j-1}^{s_{j-1}}, u_{j+1}/z_{j+1}^{s_{j+1}}, \ldots, u_m/z_m^{s_m}]$, se puede probar directamente que (§) es la fórmula que buscábamos. Obsérvese que (3) se tiene inmediatamente por la construcción de B' y la definición de τ'.

El resto de los casos (k es posición antimonótona o $m(f)(j) = 1$) se prueban de manera similar usando ζ'_2 cuando sea necesario. ■

Para probar la completitud del método, enumeramos el conjunto de variables X como $x_1, x_2, \ldots$ de modo que γ' es aplicada a una fórmula φ en una rama B, instanciando φ con la primera variable libre x_i^s todavía no usada en el tableau. Análogamente se procede con *Refv* y *Reff*. Además consideramos un límite $l \geq 1$ para el número de γ'-aplicaciones de cualquier aparición de una fórmula de clase *Gamma* en una rama.

Definición 3.7.10 *La definición de regla de construcción de tableaux con variables libres es similar a la del caso básico. Una regla de construcción de tableaux con variables libres es justa con respecto a un límite l para el número de γ'-aplicaciones si, para toda secuencia de tableaux $\mathcal{T}_0, \mathcal{T}_1, \ldots$ para un conjunto de Σ-sentencias Φ construida de acuerdo con $\mathcal{R}$, se verifican las condiciones siguientes:*

- *A toda fórmula que no sea un literal y aparezca en una rama abierta de cualquier $\mathcal{T}_i$, se le aplicará en algún momento la regla de expansión adecuada.*

- *A cualquier aparición de una fórmula de clase Gamma en una rama abierta de cualquier tableau $\mathcal{T}_i$ se le aplicará γ' hasta l veces con variables nuevas en el tableau.*

- *$\mathcal{R}$ introduce en algún momento $x \sqsubseteq x$ y $f(x_1,\ldots,x_n) \sqsubseteq f(x_1,\ldots,x_n)$, en cada rama abierta B de cualquier tableau, para toda variable nueva en el tableau x, y cada símbolo de función f de Σ y variables nuevas en el tableau $x_1,\ldots,x_n$ tales que $WS(f(x_1,\ldots,x_n) \sqsubseteq f(x_1,\ldots,x_n), B)$, respectivamente.*

- *A cualquier aparición, en una rama abierta de un tableau $\mathcal{T}_i$, de un par de átomos a los que se les pueda aplicar alguna de las reglas ζ_1', ζ_2', ξ_1' o ξ_2', se le aplicará en algún momento la regla adecuada.*

Teorema 3.7.11 (Completitud) *Sea Φ un conjunto finito insatisfactible de Σ-sentencias tal que $WS(\Phi, \mathcal{P}(\Phi))$. Existe un límite $l \geq 1$ para el cual, si $\mathcal{R}$ es una regla de construcción de tableaux con variables libres, justa con respecto a l, tal que:*

(i) la aplicación de todas las reglas de expansión de primer orden se hace al principio

(ii) toda aplicación posible de las reglas de expansión de desigualdad entre datos se hace después

entonces Φ tiene un tableau con variable libres cerrado construido de acuerdo con $\mathcal{R}$.

Demostración. Consideremos una regla justa de construcción de tableaux básicos $\mathcal{R}'$ similar a $\mathcal{R}$[†], pero en la que no se limite el uso de γ-aplicaciones. Por el Teorema 3.6.10 existe un tableau básico cerrado $\mathcal{T}_g$ para Φ construido de acuerdo con $\mathcal{R}'$. Sea l el mayor número de γ-aplicaciones a una fórmula en cualquier rama de $\mathcal{T}_g$. Extendamos $\mathcal{T}_g$ aplicando l veces la regla γ a toda aparición de fórmula de clase *Gamma* en $\mathcal{T}_g$ y movamos las aplicaciones de las reglas de desigualdad entre datos al final de la rama, lo cual es posible porque las fórmulas resultantes de estas reglas son literales. Sea $\mathcal{T}_g'$ la parte de este tableau transformado, antes de aplicar las reglas de desigualdad entre datos. Consideremos el tableau con variables libres $\mathcal{T}$ construido de acuerdo con $\mathcal{R}$, simulando cada γ-expansión que introduce el término básico $t \in \mathcal{T}_{\Sigma}^{B}(s)$ en $\mathcal{T}_g'$, mediante la γ'-expansión que introduce la variable libre $x^{sort(t)}$ en $\mathcal{T}$. Sea σ la sustitución básica que relaciona ambos tableaux ($\mathcal{T}\sigma = \mathcal{T}_g'$). Obsérvese que $WS(\sigma, \mathcal{T})$ y que σ respeta los géneros de forma estática. En estas condiciones podemos utilizar el Lema de Elevación 3.7.8 para concluir que $\mathcal{T}$ puede cerrarse. ■

[†] $\mathcal{R}'$ modifica ligeramente el concepto de regla de construcción de tableaux básicos ya que δ se aplica a φ con un término funcional de Skolem $f(t_1',\ldots,t_n')$ en lugar de con una constante auxiliar, siendo $t_1',\ldots,t_n'$ los términos previamente introducidos en φ por alguna regla de expansión.

3.8 Ejemplos

En esta sección presentamos algunos tableaux en *LPGD* usando los métodos previamente introducidos.

Ejemplo 3.8.1 *Sean s, s' dos géneros sobre los que están definidas las relaciones binarias $R^{s,s'}$, ${R_s}^{s,s}$ y ${R_{s'}}^{s',s'}$. Supongamos que R_s y $R_{s'}$ son reflexivas. Demostramos que si s'' es un subgénero de s y s' tal que para todo elemento a de s y b de s', aRb se verifica cuando existe un c en s'' con aR_sc y $cR_{s'}b$, entonces la restricción de R a s'' es reflexiva.*

El problema puede plantearse en *LPGD* como la derivación de $\varphi = \forall z^{s''}(z^{s''}Rz^{s''})$ a partir de los axiomas $\Phi = \{1 : s'' \sqsubseteq_@ s, 2 : s'' \sqsubseteq_@ s', 3 : \forall x^s \forall y^{s'}(\exists z^{s''}(x^s R_s z^{s''} \wedge z^{s''}R_{s'}y^{s'}) \longrightarrow x^s R y^{s'}), 4 : \forall x^s(x^s R_s x^s), 5 : \forall y^{s'}(y^{s'}R_{s'}y^{s'})\}$.

En la Figura 3.1 mostramos el tableau básico cerrado $\mathcal{T}_b$ para el conjunto formado por $\Phi \cup \{6 : \neg\varphi\}$. En este y los siguientes tableaux, hemos condensado la notación para facilitar su lectura; así cada γ-expansión conlleva la eliminación implícita de la doble negación mediante la regla α.

Usamos la demostración del Teorema 3.7.11 para construir el tableau con variables libres $\mathcal{T}$ de la Figura 3.2. Además dicho tableau está cerrado por el Lema de Elevación 3.7.8. En efecto, $\theta = umg(\{16 : u^{s''}R_s z^{s''} \simeq 9 : x^{s''}R_s x^{s''}, 17 : z^{s''}R_{s'}v^{s''} \simeq 10 : y^{s''}R_{s'}y^{s''}, 14 : u^{s''}Rv^{s''} \simeq 8 : c^{s''}Rc^{s''}\}) = [c^{s''}/x^{s''}, c^{s''}/y^{s''}, c^{s''}/u^{s''}, c^{s''}/v^{s''}, c^{s''}/z^{s''}]$ verifica $WS(\theta, \mathcal{T})$.

A veces es posible construir tableaux cerrados con variables libres sin seguir los pasos de la demostración del Teorema 3.7.11. Este es el caso de este ejemplo, ya que podemos introducir variables libres del mismo género que la variable universalmente cuantificada en cada γ'-expansión para obtener el tableau cerrado $\mathcal{T}'$ de la Figura 3.3.

Obsérvese que $\mathcal{T}'$ está cerrado porque $\theta' = [c^{s''}/x^s, c^{s''}/y^{s'}, c^{s''}/u^s, c^{s''}/v^{s'}, c^{s''}/z^{s''}] = umg(\{16 \simeq 9, 17 \simeq 10, 14 \simeq 8\})$ verifica $WS(\theta', \mathcal{T}')$. Nótese que este cierre es viable porque la información del orden entre géneros es común a todas las ramas; podemos pensar, por tanto, que dicha información está estáticamente fijada en la signatura.

Ejemplo 3.8.2 *Sea una signatura con géneros $S = \{s_1, s_2, s_3, s_4\}$, una constante d^{s_4} y un símbolo de predicado $P \in P^{s_2,s_1}$, monótono en sus dos argumentos. Sea $\Phi = \{1 : s_1 \sqsubseteq_@ s_2 \to \forall x^{s_2}\forall x^{s_1}P(x^{s_2}, x^{s_1}), 2 : (s_4 \sqsubseteq_@ s_3) \vee (s_4 \sqsubseteq_@ s_1), 3 : (s_1 \sqsubseteq_@ s_2) \vee (s_3 \sqsubseteq_@ s_2), 4 : \forall x^{s_2}\neg P(x^{s_2}, d), 5 : \forall x^{s_3}P(x^{s_3}, d)\}$ y $\varphi = s_4 \sqsubseteq_@ s_3$. Entonces $\Phi \models \varphi$.*

Probamos que $\Phi \models \varphi$ con el tableau con variables libres $\mathcal{T}$ de la Figura 3.4.

Obsérvese que $\mathcal{T}$ está cerrado porque $\theta = umg(\{6 : s_4 \sqsubseteq_@ s_3 \simeq 7 : s_4 \sqsubseteq_@ s_3, 11 : s_1 \sqsubseteq_@ s_2 \simeq 9 : s_1 \sqsubseteq_@ s_2, 17 : P(y^{s_2}, d) \simeq 16 : P(x^{s_2}, x^{s_4}), 15 : P(y^{s_3}, d) \simeq 13 : P(x^{s_3}, d)\}) = [y^{s_3}/x^{s_3}, d/x^{s_4}, y^{s_2}/x^{s_2}]$ verifica $WS(\theta, \mathcal{T})$.

Obsérvese que si en la obtención de 13 hubiéramos introducido la variable libre z^{s_2} en lugar de x^{s_3}, entonces la buena tipificación del *umg* correspondiente dependería de elegir z^{s_2}/y^{s_3} (mal tipado) o y^{s_3}/z^{s_2} (bien tipado). Esta situación es posible porque con el uso de x^{s_3} estamos elevando en realidad un cierre básico, siguiendo los pasos de la demostración del Teorema 3.7.11 y por eso cualquier *umg* está bien tipado. Nótese que esto no ocurre si introducimos la variable libre z^{s_2} en el tableau.

$$\mid \quad \alpha \text{ a } 6$$

$$7{:}\ \exists z^{s''} \neg (z^{s''} R z^{s''})$$

$$\mid \quad \delta \text{ a } 7$$

$$8{:}\ \neg c^{s''} R c^{s''}$$

$$\mid \quad \gamma \text{ a } 4$$

$$9{:}\ c^{s''} R_s c^{s''}$$

$$\mid \quad \gamma \text{ a } 5$$

$$10{:}\ c^{s''} R_{s'} c^{s''}$$

$$\mid \quad \gamma \text{ a } 3$$

$$11{:}\ \forall y^{s'} (\exists z^{s''} (c^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} y^{s'}) \rightarrow c^{s''} R y^{s'})$$

$$\mid \quad \gamma \text{ a } 11$$

$$12{:}\ (\exists z^{s''} (c^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} c^{s''}) \rightarrow c^{s''} R c^{s''})$$

$$\beta \text{ a } 12$$

$$13{:}\ \neg \exists z^{s''} (c^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} c^{s''}) \qquad\qquad 14{:}\ c^{s''} R c^{s''}$$

$$\mid \quad \gamma \text{ a } 13$$

$$15{:}\ \neg (c^{s''} R_s c^{s''} \wedge c^{s''} R_{s'} c^{s''})$$

$$16{:}\ \neg c^{s''} R_s c^{s''} \qquad\qquad 17{:}\ \neg c^{s''} R_{s'} c^{s''}$$

Figura 3.1: El tableau básico cerrado $\mathcal{T}_b$.

α a 6

7: $\exists z^{s''} \neg(z^{s''} R z^{s''})$

δ' a 7

8: $\neg c^{s''} R c^{s''}$

γ' a 4, 1

9: $x^{s''} R_s x^{s''}$

γ' a 5, 2

10: $y^{s''} R_{s'} y^{s''}$

γ' a 3, 1

11: $\forall y^{s'} (\exists z^{s''} (u^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} y^{s'}) \rightarrow u^{s''} R y^{s'})$

γ' a 11, 2

12: $(\exists z^{s''} (u^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} v^{s''}) \rightarrow u^{s''} R v^{s''})$

β a 12

13: $\neg\exists z^{s''} (u^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} v^{s''})$ 14: $u^{s''} R v^{s''}$

γ' a 13

15: $\neg(u^{s''} R_s z^{s''} \wedge z^{s''} R_{s'} v^{s''})$

16: $\neg u^{s''} R_s z^{s''}$ 17: $\neg z^{s''} R_{s'} v^{s''}$

Figura 3.2: El tableau con variables libres cerrado $\mathcal{T}$.

$\mid$ α a 6

7: $\exists z^{s''} \neg(z^{s''} R z^{s''})$

$\mid$ δ' a 7

8: $\neg c^{s''} R c^{s''}$

$\mid$ γ' a 4

9: $x^s R_s x^s$

$\mid$ γ' a 5

10: $y^{s'} R_{s'} y^{s'}$

$\mid$ γ' a 3

11: $\forall y^{s'} (\exists z^{s''} (u^s R_s z^{s''} \wedge z^{s''} R_{s'} y^{s'}) \rightarrow u^s R y^{s'})$

$\mid$ γ' a 11

12: $(\exists z^{s''} (u^s R_s z^{s''} \wedge z^{s''} R_{s'} v^{s'}) \rightarrow u^s R v^{s'})$

β a 12

13: $\neg \exists z^{s''} (u^s R_s z^{s''} \wedge z^{s''} R_{s'} v^{s'})$ 14: $u^s R v^{s'}$

$\mid$ γ' a 13

15: $\neg(u^s R_s z^{s''} \wedge z^{s''} R_{s'} v^{s'})$

16: $\neg u^s R_s z^{s''}$ 17: $\neg z^{s''} R_{s'} v^{s'}$

Figura 3.3: El tableau con variables libres cerrado $\mathcal{T}'$.

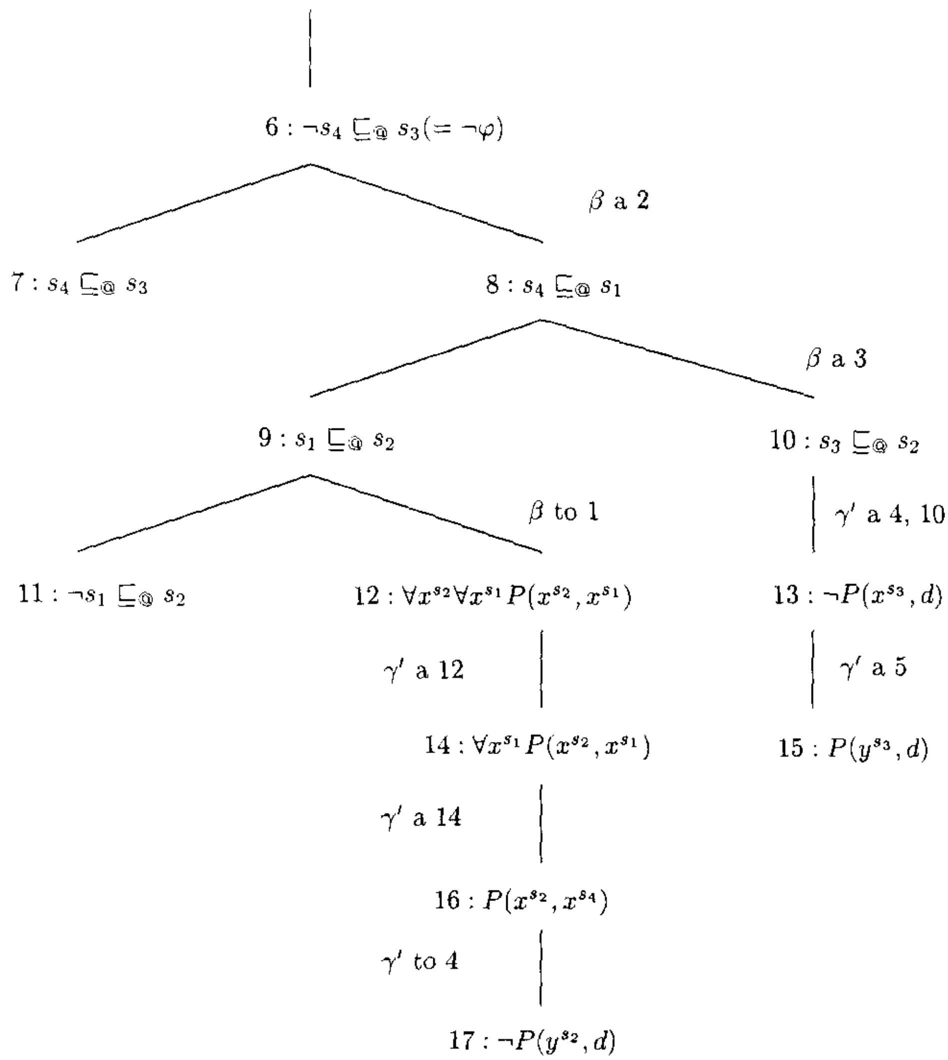

Figura 3.4: El tableau con variables libres cerrado $\mathcal{T}$.

Ejemplo 3.8.3 *Para motivar la necesidad del Lema 3.7.9, veamos cómo reconstruimos los términos involucrados en un tableau básico cerrado. Sea una signatura con géneros* $S = \{s, s_1, s_2\}$, *dos constantes* a^{s_2}, b^{s_2} *y dos símbolos de función* $f \in F^{s_1, s_1, s_1 \longrightarrow s}$, $g \in F^{s_2 \longrightarrow s_1}$, *monótonos en todos sus argumentos. Probaremos* $\{a^{s_2} \sqsubseteq b^{s_2}\} \models \exists x^{s_1} \exists y^{s_1} f(g(b), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a))$.

Usando el método de los tableaux básicos podemos construir el tableau cerrado $\mathcal{T}_b$ de la Figura 3.5.

$$1 : a^{s_2} \sqsubseteq b^{s_2}$$

$$2 : \neg \exists x^{s_1} \exists y^{s_1} f(g(b^{s_2}), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))$$

$$\Big| \quad \gamma \text{ a } 2$$

$$3 : \neg \exists y^{s_1} f(g(b^{s_2}), g(a^{s_2}), g(a^{s_2})) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))$$

$$\Big| \quad \gamma \text{ a } 3$$

$$4 : \neg f(g(b^{s_2}), g(a^{s_2}), g(a^{s_2})) \sqsubseteq f(g(b^{s_2}), g(b^{s_2}), g(a^{s_2}))$$

$$\Big| \quad \mathit{Ref}$$

$$5 : f(g(b^{s_2}), g(a^{s_2}), g(a^{s_2})) \sqsubseteq f(g(b^{s_2}), g(\mathbf{a}^{s_2}), g(a^{s_2}))$$

$$\Big| \quad \zeta_1 \text{ a } 5 \text{ y } 1$$

$$6 : f(g(b^{s_2}), g(a^{s_2}), g(a^{s_2})) \sqsubseteq f(g(b^{s_2}), g(b^{s_2}), g(a^{s_2}))$$

Figura 3.5: El tableau básico cerrado $\mathcal{T}_b$.

Para elevar $\mathcal{T}_b$ necesitamos aplicar las reglas *Reff* y ζ_1' para construir las nuevas posiciones introducidas en $\mathcal{T}_b$. El tableau resultante $\mathcal{T}$ aparece en la Figura 3.6.

Obsérvese que en la obtención de la fórmula 6 de $\mathcal{T}$ se verifica que σ está bien tipado con respecto al tableau correspondiente y que $WS(g(a^{s_2}) \sqsubseteq g(a^{s_2}), B)$. Además en la regla ζ_1' aplicamos σ a todo el tableau por lo que la fórmula 5 pasa a ser en realidad $g(a^{s_2}) \sqsubseteq g(a^{s_2})$. Lo mismo sucede con la obtención de la fórmula 8. Para finalizar nótese que $\mathcal{T}$ está cerrado porque $\theta = umg(\{8 : f(u^{s_1}, g(a^{s_2}), w^{s_1}) \sqsubseteq f(u^{s_1}, g(b^{s_2}), w^{s_1}) \simeq 4 : f(g(b^{s_2}), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))\}) = [g(b^{s_2})/u^{s_1}, g(a^{s_2})/x^{s_1}, g(a^{s_2})/w^{s_1}, g(b^{s_2})/y^{s_1}]$ verifica $WS(\theta, \mathcal{T})$.

3.9 Implementación

Entre todos los lenguajes de programación, parece que Prolog es especialmente adecuado para dar soporte a una implementación del método de los tableaux con variables libres. De hecho, a la no existencia de la secuencialidad explícita, podemos añadir la ventaja de usar las variables lógicas para implementar las variables libres, y así conseguir de una manera automática el efecto de unificación en todo el tableau.

$$1 : a^{s_2} \sqsubseteq b^{s_2}$$

$$2 : \neg\exists x^{s_1}\exists y^{s_1} f(g(b^{s_2}), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))$$

$$\mid \quad \gamma' \text{ a } 2$$

$$3 : \neg\exists y^{s_1} f(g(b^{s_2}), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))$$

$$\mid \quad \gamma' \text{ a } 3$$

$$4 : \neg f(g(b^{s_2}), x^{s_1}, x^{s_1}) \sqsubseteq f(y^{s_1}, y^{s_1}, g(a^{s_2}))$$

$$\mid \quad \textit{Reff}$$

$$5 : g(z^{s_2}) \sqsubseteq g(\mathbf{z}^{\mathbf{s_2}})$$

$$\mid \quad \zeta_1' \text{ a } 5 \text{ y } 1;\ \sigma = [a^{s_2}/z^{s_2}]$$

$$6 : g(a^{s_2}) \sqsubseteq g(b^{s_2})$$

$$\mid \quad \textit{Reff}$$

$$7 : f(u^{s_1}, v^{s_1}, w^{s_1}) \sqsubseteq f(u^{s_1}, \mathbf{v}^{\mathbf{s_1}}, w^{s_1})$$

$$\mid \quad \zeta_1' \text{ a } 7 \text{ y } 6;\ \sigma' = [g(a^{s_2})/v^{s_1}]$$

$$8 : f(u^{s_1}, g(a^{s_2}), w^{s_1}) \sqsubseteq f(u^{s_1}, g(b^{s_2}), w^{s_1})$$

Figura 3.6: El tableau con variables libres cerrado $\mathcal{T}$.

En la construcción del demostrador que aquí presentamos, se ha considerado un diseño modular bien estructurado para conseguir una versión fácilmente extendible y modificable. Con este propósito, su desarrollo pasa por tres etapas: (1) *Chus1*, que implementa un método de tableaux con variables libres para *LPO*, (2) *Chus2*, que extiende *Chus1*, incluyendo reglas para el tratamiento del orden entre géneros, y (3) *Chus3*, un sistema todavía en desarrollo, que considerará las reglas de extensión de desigualdad entre datos.

3.9.1 *Chus1*

Chus1 es un demostrador de teoremas de primer orden basado en el método de los tableaux con variables libres. Aunque admite como *input* cualquier sentencia, sólo trabaja sobre las denominadas *Skolemized negation normal forms* (*SNNFs* en adelante), reduciendo de esta forma la cantidad de reglas aplicables, ya que las reglas α (aplicada a doble negación) y δ' no son necesarias. Así, lo primero que *Chus1* hace es transformar la sentencia de partida en *SNNF*. El principal predicado de *Chus1* es `test/2` y se define por:

```
test(X, Qdepth) :-
        reset,
        snnf(neg X, [], SNNF, _),
        expand(nb(SNNF, [], [] # [], Qdepth), Tree\[]), !,
        closed(Tree).
```

Para resolver el objetivo `:- test(X, Qdepth)`, primero se construye la *SNNF* de ¬`X`, mediante la llamada a `snnf(neg X,[],SNNF,_)`, donde la variable `SNNF` se instancia a la sentencia deseada. Ya que se pueden necesitar nuevos símbolos de función, usamos un contador global[‡] y lo inicializamos con `reset`. Para `SNNF`, se construye un tableau *completo* con `expand(nb(SNNF,[],[]#[],Qdepth),Tree\[])`; completo significa que las γ'-expansiones están permitidas en cada rama con límite `Qdepth`, por lo que un fallo puede deberse a que la expansión conseguida no sea satisfactoria. El constructor `nb/4` representa una rama anotada, cuyos parámetros establecen la fórmula bajo expansión, la lista de fórmulas todavía no expandidas, la lista de los literales positivos y negativos presentes, y el límite de γ'-expansiones. El parámetro `Tree\[]` recoge la expansión y es implementado como una estructura de datos incompleta.

Hasta este punto, el demostrador trabaja de forma determinista, por lo que el corte elimina las posibles alternativas pendientes. Una vez que se obtiene la expansión, se intenta cerrar cada rama de `Tree` encontrando una sustitución adecuada. Esto se realiza con el predicado `closed(Tree)`, usando el predicado oculto `unify/2`[§]. Como hemos dicho, el proceso de unificación se beneficia de la unificación de Prolog, en el sentido de que si una variable libre es instanciada, será instanciada en todo el tableau. Obsérvese que en el proceso de cierre, una sustitución puede ser descartada si no permite cerrar otra rama, por lo que el *backtraking* puede resultar necesario.

El diseño modular correspodiente a *Chus1* comprende los 6 módulos de la Figura 3.7. Los módulos *snnf* y *expand* contienen las reglas correspondientes a los predicados `snnf` y `expand`, respectivamente. Como tanto en el proceso de skolemización, como en el de γ-expansión, necesitamos instanciar las variables que aparecen en determinadas fórmulas, estos dos módulos acuden al predicado `instance/3` del módulo *instance*. En el módulo *closure* se encuentra el predicado de cierre `closed/1`, que requiere a su vez el predicado `unify/2` del módulo *unify*.

3.9.2 *Chus2*

Considerar géneros dinámicos implica varios cambios en *Chus1*. En primer lugar, es necesaria la declaración estática de cualquier constante o símbolo de función que aparezca en la sentencia de partida. Obsérvese que los símbolos de función de Skolem tienen que ser declarados dinámicamente. Todo ésto se consigue usando predicados dinámicos.

En el proceso de expansión, tenemos que modificar la regla γ' para que admita géneros ordenados y, además, tenemos que implementar la relación de subgénero $\ll^B$. Para lo último, construiremos el cierre transitivo de la información de orden entre géneros, cada

[‡]Este contador es implementado como predicado dinámico en el módulo *instance*.

[§]`unify/2` respeta el occur check.

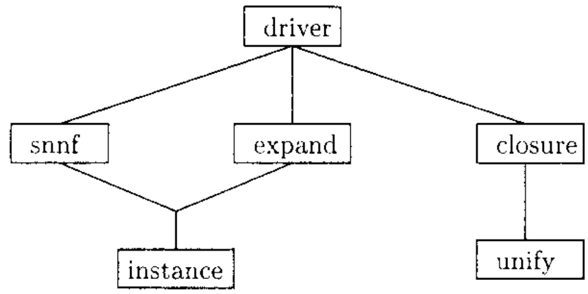

Figura 3.7: Jerarquía de módulos de *Chus1*.

vez que se detecte una nueva desigualdad. De esta forma, la regla γ' se implementa de manera que, en una única expansión, se introducen todas las posibles instancias con respecto a la información actual de orden entre géneros.

Además el género de cada variable libre debe declararse, pues será necesario en el proceso de cierre. Para poder asegurar que una sustitución bien tipada con respecto a la rama esté bien tipada con respecto a todo el tableau, asociaremos a cada variable libre la información de orden entre géneros que existía en la rama cuando la variable fue introducida.

En el proceso de cierre, tenemos que manejar las desigualdades de orden entre géneros, que pueden cerrar una rama de forma trivial, y asegurarnos de que una sustitución de cierre está bien tipada con respecto a todo el tableau.

En el proceso de unificación, dos variables pueden ser identificadas por lo que su información de orden entre géneros puede cambiar. Como el *backtracking* está permitido, no podemos implementar la declaración de las variables libres como predicados dinámicos; esta es la razón de la aparación de nuevos parámetros en los predicados `expand/4` y `closed/3` de más abajo. Así, el parámetro `A\A` de `expand/4` contiene la lista del género y de la jerarquía correspondiente a cada variable del tableau. Como la información de cada variable libre se genera en la expansión del tableau, usamos el parámetro `Vs\[]` de `expand/4` como acumulador y, entonces, utilizamos y/o modificamos `Vs` en el proceso de cierre.

En estas condiciones, el predicado principal `test/2` se convierte en:

```
test(X, Qdepth) :-
        reset,
        snnf(neg X, [], SNNF, _),
        expand(nb(SNNF, [], []#[], []#[], Qdepth), Tree\[], A\A, Vs\[]),!,
        closed(Tree, Vs, _).
```

El diseño modular correspodiente a *Chus2* comprende los 7 módulos de la Figura 3.8. El nuevo módulo *declaration* se encarga de generar los nuevos símbolos de función de Skolem y de gestionar sus declaraciones en cuanto a géneros, durante el proceso de skolemización. Además, se dedica a la gestión de las declaraciones de las variables introducidas en cada γ-expansión.

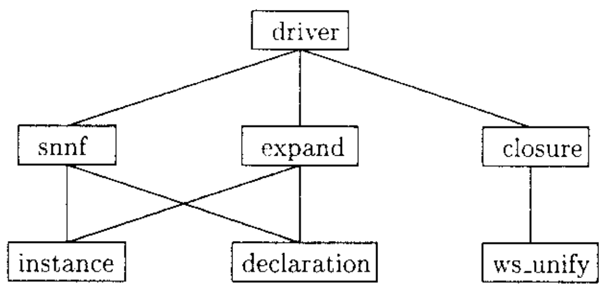

Figura 3.8: Jerarquía de módulos de *Chus2*.

3.10 Conclusiones y trabajos relacionados

Hemos integrado géneros dinámicos y preórdenes (anti)monótonos en la lógica *LPGD* para la que hemos diseñado métodos de deducción correctos y completos basados en los tableaux semánticos.

Como en *LPO*, los tableaux básicos resultan ser un mecanismo ineficiente de demostración ya que introducen términos bien tipados de forma aleatoria. Para evitar este problema, los tableaux con variables libres usan variables que son instanciadas para cerrar o extender un tableau.

A propósito de la eficacia del método de tableaux con variables libres cabe decir lo siguiente:

1. Con respecto a los géneros dinámicos, este método sólo necesita unificación sintáctica, si bien debemos comprobar que el *umg* correspondiente está bien tipado con respecto a todo el tableau. En este sentido, parece que los géneros dinámicos no añaden complejidad a *LPO*, pues la unificación sintáctica y la comprobación de buena tipificación son problemas decidibles. Sin embargo, para asegurar completitud, la regla γ' puede instanciar la variable universalmente cuantificada x^s con variables de la forma $y^{s'}$, donde s' es cualquier género tal que $s' \ll^B s$, por lo que el método es ineficiente ya que la elección de s' es aleatoria.

   En el capítulo 4, construiremos cálculos terminantes de unificación que separen el problema de los géneros de la indecidibilidad de *LPO*. Sin embargo no usaremos *LPGD*, sino que extenderemos el poder expresivo de esta lógica mediante lo que denominaremos *declaraciones de términos*.

2. Con respecto a los preórdenes (anti)monótonos, el gran problema reside en la necesidad de los ineficientes *axiomas de reflexividad funcional Reff*, ya que el símbolo de función elegido en ellos es aleatorio.

En el caso de la igualdad se ha demostrado que tales axiomas son innecesarios [Brand 75], [BGV 97], [DV 94], [DV 98]. Nosotros estudiaremos si estas técnicas pueden aplicarse a preórdenes monótonos en el capítulo 5.

Los resultados que aparecen en el presente capítulo son el fruto de diversos trabajos previos. En [MG 96] y [Mart 96] se estudia la lógica *LPGD*, pero los métodos de tableaux no son los mismos. La diferencia estriba en que la relación $\ll$ se trata allí de forma explícita mediante la regla de expansión:

$$Ref_{@})\quad \frac{}{s \sqsubseteq_{@} s} \qquad\qquad Tran_{@})\quad \frac{\begin{array}{c} s_1 \sqsubseteq_{@} s_2 \\ s_2 \sqsubseteq_{@} s_3 \end{array}}{s_1 \sqsubseteq_{@} s_3}$$

Además, las sustituciones permitidas en ellos siempre respetan el género de forma estática, por lo que los resultados son similares, pero sobre una base diferente.

Más diferencias existen con respecto a [GLMN 96], donde *LPGD* se hace trivalorada. La interpretación de los términos es la misma que la que aparece en este capítulo, pero las fórmulas mal tipadas se evalúan a $\underline{u}$ (un tercer valor booleano que expresa incertidumbre) en vez de interpretarse como $\underline{f}$. Esta aproximación introduce nuevos cambios en el planteamiento de las reglas de los tableaux. Por ejemplo, en las reglas básicas ζ_1, ζ_2, ξ_1 y ξ_2 exigimos que las premisas estén bien tipadas en lugar de pedirlo en la conclusión. En cuanto al cierre de un tableau básico, exigimos que los átomos complementarios estén bien tipados para que así no se evalúen a $\underline{u}$ y, por tanto, forzar a que la contradicción semántica exista. Los tableaux con variables libres en [GLMN 96] también se basan en unificación sintáctica.

Capítulo 4

Declaraciones de términos

En este capítulo solventamos las carencias del sistema de tipos usado en *LPGD*. Para ello, nos olvidamos de los preórdenes, y nos centramos en el diseño de mecanismos más especializados en el manejo de los géneros. Sin embargo, no partimos del sistema de géneros dinámicos incluido en *LPGD*, sino que extendemos el poder expresivo de ésta usando *declaraciones de términos*. Para esta nueva lógica construimos métodos de deducción correctos y completos basados en tableaux. Además, incluimos un cálculo de unificación tipada dentro del método de tableaux con variables libres, que se usa para cerrar tableaux en un solo paso. La importancia de este cálculo reside en que es terminante y, por ello, puede usarse como un procedimiento de decisión. En este sentido, a diferencia de lo hecho para *LPGD*, hemos separado la complejidad de los géneros de la indecidibilidad de la lógica de primer orden.

4.1 Introducción

En el contexto de los sistemas lógicos, los géneros han sido ampliamente aceptados como un mecanismo útil para incrementar la eficacia, reduciendo el espacio de búsqueda y permitiendo representaciones más naturales. Dos aproximaciones principales se han seguido en la incorporación de los géneros a las lógicas. Por una parte, se dice que los géneros se comportan *estáticamente* cuando las propiedades de éstos -tales como las jerarquías y las declaraciones del género de las operaciones- se fijan en la signatura [Cohn 87], [Walt 87], [SS 89].

Por otra parte, se dice que los géneros se comportan *dinámicamente* cuando la información sobre géneros e individuos coexiste dentro del mismo marco formal [Fris 91], [GLMN 96]. En este caso, la mayor expresividad se consigue cuando las declaraciones del género de las operaciones se expresan a través de un nuevo contructor de fórmulas. Así aparecen las llamadas *lógicas con declaraciones de términos* [Weid 91] como un sistema lógico que incluye, en un único formalismo, una lógica heterogénea clásica junto con toda la información que las declaraciones de términos conllevan (relaciones entre géneros y declaraciones del género de los símbolos de función).

Este capítulo sigue una línea de investigación orientada a la construcción de métodos de tableaux para lógicas con declaraciones de términos [GLMN 97], [MGL 98], [MG 99], [MGL 00]. En lugar de definir nuevas reglas de deducción, separamos los géneros de *LPO* usando un cálculo de unificación tipada. Este cálculo sirve para unificar un conjunto de ecuaciones obtenidas de literales potencialmente complementarios que aparecen en las ramas del tableau. Las variables libres presentan dos dificultades a tener en cuenta cuando diseñamos el cálculo tipado. En primer lugar, tienen un género asociado estáticamente que restringe su dominio [Weid 91], [Fris 91], [GLMN 96], de modo que sólo podemos aplicar sustituciones que estén *bien tipadas*. Esto significa que el género (estático) de la variable sustituida debe coincidir con el género (dinámico) del correspondiente término introducido. En segundo lugar, las variables libres se comportan rígidamente y por ello sólo pueden ser instanciadas una única vez [DV 98]. Por este motivo, tenemos que tener en cuenta la información de géneros que aparece en todo el tableau, incluso cuando cerramos una única rama.

El capítulo esta organizado como sigue. Las secciones siguientes presentan la Lógica con Declaración de Términos. La Sección 4 introduce la noción de sustitución *bien tipada*. A continuación diseñamos métodos deductivos basados en tableaux. En la Sección 5 presentamos un sistema de tableaux básicos y en la Sección 6, diseñamos una primera versión con variables libres. Este último sistema no aprovecha todo el poder de la variables libres, por lo que en la Sección 7 presentamos un cálculo de unificación tipada que cierra ramas independientes en un tableau. En la Sección 8 extendemos este cálculo de unificación para cerrar un tableau de forma simultánea. Finalmente, en la Sección 9, integramos este último cálculo en un nuevo sistema de tableaux con variables libres.

4.2 Sintaxis de LDT

La Lógica con Declaraciones de Términos (*LDT* en lo que sigue) amplía la lógica de primer orden, introduciendo un nuevo constructor de fórmulas $t \in s$ para expresar que el término t tiene género s. Llamaremos *declaraciones de términos* a dichas fórmulas. Las operaciones de *LDT* no tienen asociado ningún género estático. De hecho, una signatura en *LDT* se compone de un conjunto finito S de géneros s, y conjuntos *no tipados* $\mathcal{C}$, $\mathcal{F}$ y $\mathcal{P}$ de símbolos de constante, función y predicado, respectivamente, los dos últimos con elementos de aridad fija. Sólo las variables están restringidas estáticamente a un género fijo; pertenecen a uno de los conjuntos numerables de la familia S-indexada $X = (X^s)_{s \in S}$.

Los conjuntos de Σ-términos y Σ-fórmulas se definen como en *LPO*, pero incluyendo declaraciones de términos.

Definición 4.2.1 *El conjunto de Σ-términos $T(\Sigma, X)$ se define mediante las siguientes reglas de formación:*

$$t ::= x^s (\in X^s) \mid c\ (\in \mathcal{C}) \mid f(t_1, \ldots, t_n)\ (f \in \mathcal{F} \text{ de aridad } n;\ t_1, \ldots, t_n \in T(\Sigma, X))$$

El conjunto de Σ-fórmulas $F(\Sigma, X)$ se define mediante las siguientes reglas de formación:

$$\varphi ::= t \in s \mid P(t_1, \ldots, t_n)\ (P \in \mathcal{P} \text{ de aridad } n; t_1, \ldots, t_n \in T(\Sigma, X)) \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x^s \varphi.$$

LDT incrementa el poder expresivo de *LPGD* ya que podemos simular $s \sqsubseteq_{\mathbb{Q}} s'$ con la fórmula $\forall x^s(x^s \in s')$. Además, en *LDT* podemos declarar el comportamiento de cualquier símbolo de función o predicado de forma dinámica; por ejemplo, $\forall x^s(f(x^s) \in s')$ expresa que el rango de la función f en el s-dominio es un conjunto de s'-elementos. Incluso podemos sobrecargar la declaración de las funciones y predicados con fórmulas del tipo $\forall x^s(f(x^s) \in s') \wedge \forall x^{s''}(f(x^{s''}) \in s)$.

4.3 Semántica para LDT

Las estructuras para *LDT* contienen una familia de dominios, más las interpretaciones de los símbolos de la signatura.

Definición 4.3.1 *Una Σ-estructura $\mathcal{D}$ es una tupla compuesta por:*

1. *Un dominio total D que contiene una familia de dominios $\{D^s | s \in S\}$.*
2. *Conjuntos de interpretaciones $\{c^{\mathcal{D}} \in D | c \in \mathcal{C}\}, \{f^{\mathcal{D}} : D^n \to D | f \in \mathcal{F}$ de aridad $n\}$, $\{P^{\mathcal{D}} : D^n \to \{\underline{t}, \underline{f}\} | P \in \mathcal{P}$ de aridad $n\}$, para los símbolos de Σ.*

Considerando que no tenemos declaraciones de género en la signatura, los dominios pueden llegar a estar vacíos; sólo se sabe que $\cup D^s \subseteq D$. Además obsérvese que, como los símbolos de función o predicado no están declarados en la signatura, sus interpretaciones se comportan de forma homogénea, esto es, actúan sobre todos los elementos del dominio total D.

Una valoración para $\mathcal{D}$ es una familia S-indexada $\rho = (\rho^s)_{s\in S}$ de aplicaciones finitas $\rho^s : X^s \to D^s$ de la forma $[\rho^s(x_1^s)/x_1^s, \ldots, \rho^s(x_n^s)/x_n^s]$; $dom(\rho^s) = \{x_1^s, \ldots, x_n^s\}$ es el dominio de ρ^s, y $dom(\rho) = \bigcup_{s\in S} dom(\rho^s)$ es el dominio de ρ. Obsérvese que $dom(\rho^s) = \emptyset$ si $D^s = \emptyset$. Como es usual, $\rho[d/x^s]$ representará la valoración que asigna d a x^s y se comporta como ρ sobre el resto de las variables de $dom(\rho)$.

El valor semántico $[\![t]\!]_\rho^{\mathcal{D}}$ de un término t en una Σ-interpretación $\langle \mathcal{D}, \rho\rangle$ se define como en *LPO* y existe siempre que $var(t) \subseteq dom(\rho)$. El valor booleano $[\![\varphi]\!]_\rho^{\mathcal{D}}$ de una fórmula φ en $\langle \mathcal{D}, \rho\rangle$ también existe cuando $var(\varphi) \subseteq dom(\rho)$ y se define como en *LPO*, excepto para:

- $[\![\exists x^s \varphi]\!]_\rho^{\mathcal{D}} = \begin{cases} \underline{t} & \text{si existe } d \in D^s \text{ tal que } [\![\varphi]\!]_{\rho[d/x^s]}^{\mathcal{D}} = \underline{t} \\ \underline{f} & \text{en otro caso.} \end{cases}$
- $[\![t \in s]\!]_\rho^{\mathcal{D}} = \begin{cases} \underline{t} & \text{si } [\![t]\!]_\rho^{\mathcal{D}} \in D^s \\ \underline{f} & \text{en otro caso.} \end{cases}$

En adelante, cuando escribamos $[\![t]\!]_\rho^{\mathcal{D}}$ (resp. $[\![\varphi]\!]_\rho^{\mathcal{D}}$), asumimos que $var(t) \subseteq dom(\rho)$ (resp. $var(\varphi) \subseteq dom(\rho)$), lo que trivialmente se cumple para los términos básicos (resp. sentencias).

Obsérvese que si $\mathcal{D}$ es una Σ-estructura con $D^s = \emptyset$, las sentencias $\forall x^s\varphi$ y $\exists x^s\varphi$ siempre se evalúan, respectivamente, a $\underline{t}$ y a $\underline{f}$, independientemente de la forma de φ.

En *LDT*, los conceptos de *modelo* y *consecuencia lógica* se adaptan al modo en que manejamos las valoraciones; así $\mathcal{D} \models \varphi$ es cierto si y sólo si $\langle \mathcal{D}, \rho \rangle \models \varphi$, para toda valoración ρ que verifique $var(\varphi) \subseteq dom(\rho)$.

LDT no es más expresiva que la lógica de primer orden (los géneros pueden ser expresados como predicados unarios [Weid 95]), pero permite representaciones y deducciones más naturales.

Ejemplo 4.3.2 *Sea Σ una signatura con géneros s, s', la constante a, el símbolo de función unitaria f y el símbolo de predicado binario P. Para entender las siguientes sentencias más directemente, nos referiremos al género s como el de los* seres humanos, *s', el de la* gente amable, *$f(\Box)$ como el* padre de $\Box$, *y $P(\Box, \Diamond)$ el predicado que expresa que* $\Box$ se lleva bien con $\Diamond$. *Supongamos que 1:* a es un ser humano ($a \in s$), *2:* que no se lleva bien con su padre ($\neg P(a, f(a))$), *3:* todo humano amable se lleva bien con todo el mundo ($\forall x^s (x^s \in s' \rightarrow \forall y^s P(x^s, y^s))$) *y 4:* el padre de todo ser humano es un ser humano ($\forall x^s f(x^s) \in s$). *De todo ello deducimos que* algunos seres humanos no son amables ($\exists x^s (\neg x^s \in s')$).

La formulación del ejemplo anterior puede expresarse en *LPO*, pero usando fórmulas más complejas. Por ejemplo, la tercera fórmula podría transformarse en $\forall x(S(x) \rightarrow (S'(x) \rightarrow \forall y(S(y) \rightarrow P(x, y))))$, donde S y S' serían predicados unarios que representan los géneros s y s', respectivamente.

Las declaraciones de términos también mejoran el poder expresivo de las lógicas heterogéneas con géneros ordenados estáticamente.

Ejemplo 4.3.3 *Podemos probar que la sentencia $\exists x^{s''} (x^{s''} \in s')$, que expresa que la intersección de los géneros s' y s'' no es vacía, es consecuencia lógica del conjunto de sentencias $\{a \in s, f(a) \in s'', \forall x^s \ (f(x^s) \in s')\}$.*

Obsérvese que, en una lógica heterogénea sin igualdad, la sentencia $\exists x^{s''} (x^{s''} \in s')$ no puede ser expresada cómodamente, ya que no podemos representar la identificación de dos elementos (de distinto género). Una solución sin igualadad necesitaría el género $s \cap s'$, haciendo que la signatura dependa del problema.

Ejemplo 4.3.4 *Las estructuras con géneros vacíos se pueden caracterizar sintácticamente mediante la fórmula $\varphi = \forall x^s \ (x^s \notin s)$, ya que para toda Σ-estructura $\mathcal{D}$, $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$ si y sólo si $\mathcal{D}^s = \emptyset$.*

4.4 Sustituciones para LDT

Las sustituciones para *LDT* son reemplazamientos finitos de variables por términos, como en *LPO*. Dado que las variables tienen asociado un género (estático) debemos asegurar que el género (dinámico) del término introducido coincide con el de la variable sustituida. Los contextos sintácticos que garantizan esto son los conjuntos de declaraciones de términos que llamaremos desde ahora $\in$-*teorías*, o simplemente *teorías*. Obsérvese que nuestro concepto de teoría difiere del sentido habitualmente usado, ya que no hacemos referencia a ninguna noción de cierre bajo derivabilidad. Representaremos las teorías mediante la letra $\mathcal{L}$.

Definición 4.4.1 (Sustituciones bien tipadas) *Una sustitución $[t_1/x_1^{s_1}, \ldots, t_n/x_n^{s_n}]$ está bien tipada con respecto a una teoría $\mathcal{L}$ si $(t_i \in s_i) \in \mathcal{L}$, para todo $1 \leq i \leq n$.*

Las sustituciones bien tipadas verifican las siguientes propiedades.

Lema 4.4.2 *Sea $\mathcal{L}$ una teoría, $\langle \mathcal{D}, \rho \rangle$ una Σ-interpretación que satisface $\mathcal{L}$ y $\tau = [t_1/x_1^{s_1}, \ldots, t_n/x_n^{s_n}]$ una sustitución bien tipada con respecto a $\mathcal{L}$, entonces $[\![t_i]\!]_\rho^\mathcal{D} \in D^{s_i}$, para todo $1 \leq i \leq n$.*

Demostración Usando que τ está bien tipada con respecto a $\mathcal{L}$ obtenemos que $(t_i \in s_i) \in \mathcal{L}$, para todo $1 \leq i \leq n$. En consecuencia tenemos que $[\![t_i]\!]_\rho^\mathcal{D} \in D^{s_i}$ porque $\langle \mathcal{D}, \rho \rangle \models \mathcal{L}$. ■

Lema 4.4.3 (Substitutividad para términos y fórmulas) *Sea $\mathcal{L}$ una teoría, $\langle \mathcal{D}, \rho \rangle$ una Σ-interpretación que satisface $\mathcal{L}$ y $\tau = [t_1/x_1^{s_1}, \ldots, t_n/x_n^{s_n}]$ una sustitución bien tipada con respecto a $\mathcal{L}$. Para todo término t (resp. fórmula φ) tal que $var(t) - \{x_1^{s_1}, \ldots, x_n^{s_n}\}$ (resp. $var(\varphi) - \{x_1^{s_1}, \ldots, x_n^{s_n}\}$) está incluido en $dom(\rho)$, se verifica:*

1. $[\![t\tau]\!]_\rho^\mathcal{D} = [\![t]\!]^\mathcal{D}_{\rho[[\![t_1]\!]_\rho^\mathcal{D}/x_1^{s_1}, \ldots, [\![t_n]\!]_\rho^\mathcal{D}/x_n^{s_n}]}$

2. $[\![\varphi\tau]\!]_\rho^\mathcal{D} = [\![\varphi]\!]^\mathcal{D}_{\rho[[\![t_1]\!]_\rho^\mathcal{D}/x_1^{s_1}, \ldots, [\![t_n]\!]_\rho^\mathcal{D}/x_n^{s_n}]}$

Demostración. Por inducción sobre la estructura de t y φ. Obsérvese que $\rho' = \rho[[\![t_1]\!]_\rho^\mathcal{D}/x_1^{s_1}, \ldots, [\![t_n]\!]_\rho^\mathcal{D}/x_n^{s_n}]$ es una valoración por el Lema 4.4.2.

1. En el caso básico, si $t = c$ o $t = y$, $y \notin \{x_1, \ldots, x_n\}$, la demostración es trivial (obsérvese que, por hipótesis, $y \in dom(\rho)$); si $t = x_i$ entonces $[\![t\tau]\!]_\rho^\mathcal{D} = [\![t_i]\!]_\rho^\mathcal{D} = [\![x_i]\!]_{\rho'}^\mathcal{D}$.

   Para el paso inductivo, sea $t = f(r_1, \ldots, r_m)$ entonces, por la semántica, $[\![t\tau]\!]_\rho^\mathcal{D} = f^\mathcal{D}([\![r_1\tau]\!]_\rho^\mathcal{D}, \ldots, [\![r_m\tau]\!]_\rho^\mathcal{D})$ y, por hipótesis de inducción, $[\![r_j\tau]\!]_\rho^\mathcal{D} = [\![r_j]\!]_{\rho'}^\mathcal{D}$, $1 \leq j \leq m$. Por tanto, $f^\mathcal{D}([\![r_1\tau]\!]_\rho^\mathcal{D}, \ldots, [\![r_m\tau]\!]_\rho^\mathcal{D}) = f^\mathcal{D}([\![r_1]\!]_{\rho'}^\mathcal{D}, \ldots, [\![r_m]\!]_{\rho'}^\mathcal{D}) = [\![t]\!]_{\rho'}^\mathcal{D}$.

2. Si $\varphi = (t \in s)$ entonces $[\![\varphi\tau]\!]_\rho^\mathcal{D} = [\![t\tau \in s]\!]_\rho^\mathcal{D} = [\![t \in s]\!]_{\rho'}^\mathcal{D}$, por la semántica y (1); si $\varphi = P(r_1, \ldots, r_m)$ entonces $[\![\varphi\tau]\!]_\rho^\mathcal{D} = P^\mathcal{D}([\![r_1\tau]\!]_\rho^\mathcal{D}, \ldots, [\![r_m\tau]\!]_\rho^\mathcal{D})$. Por (1) tenemos $[\![r_j\tau]\!]_\rho^\mathcal{D} = [\![r_j]\!]_{\rho'}^\mathcal{D}$, $1 \leq j \leq m$, por tanto $[\![\varphi\tau]\!]_\rho^\mathcal{D} = [\![\varphi]\!]_{\rho'}^\mathcal{D}$.

   Los casos inductivos para los conectivos $\neg$ y $\wedge$ son sencillos de probar. Si $\varphi = \exists y^s \varphi'$, supongamos sin pérdida de generalidad que $y^s \neq x_i{}^{s_i}$ y que $y^s \notin var(t_i)$, $1 \leq i \leq n$; de lo contrario renombramos la variable cuantificada y^s. En estas condiciones, $[\![\varphi\tau]\!]_\rho^\mathcal{D} = [\![\exists y^s(\varphi'\tau)]\!]_\rho^\mathcal{D}$. Sea d un elemento de D^s, entonces:

$$\begin{aligned}
[\![\varphi'\tau]\!]^{\mathcal{D}}_{\rho[d/y]} &= [\![\varphi']\!]^{\mathcal{D}}_{\rho[d/y][[\![t_1]\!]^{\mathcal{D}}_{\rho[d/y]}/x_1^{s_1},\ldots,[\![t_n]\!]^{\mathcal{D}}_{\rho[d/y]}/x_n^{s_n}]} \text{ (por hipótesis de inducción)} \\
&= [\![\varphi']\!]^{\mathcal{D}}_{\rho[[\![t_1]\!]^{\mathcal{D}}_{\rho}/x_1^{s_1},\ldots,[\![t_n]\!]^{\mathcal{D}}_{\rho}/x_n^{s_n}][d/y]} \\
&= [\![\varphi']\!]^{\mathcal{D}}_{\rho'[d/y]}
\end{aligned}$$

Por tanto $[\![\varphi\tau]\!]^{\mathcal{D}}_{\rho} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'}$, por la semántica. ■

4.5 Tableaux básicos para LDT

A continuación presentamos un método de tableaux básico para *LDT*, correcto y completo. Como siempre, suponemos que Σ ha sido extendida a una signatura $\overline{\Sigma}$, con un conjunto numerable de nuevas constantes. Las reglas α y β se definen como en los tableaux básicos de primer orden. Las reglas γ y δ las definimos esquemáticamente como sigue:

$$\gamma)\quad \frac{\begin{array}{c}\neg\exists x^s\varphi \\ t \in s\end{array}}{\neg\varphi[t/x^s]} \qquad\qquad \delta)\quad \frac{\exists x^s\varphi}{\begin{array}{c}\varphi[c/x^s] \\ c \in s\end{array}}$$

En γ, t es un término básico; en δ, c es una nueva constante que no aparece en la rama. Obsérvese cómo manejamos dinámicamente la información de los géneros, usando la declaración $(t \in s)$ o introduciendo $(c \in s)$ en la rama.

Definición 4.5.1 *Una rama B de un tableau está cerrada si una contradicción atómica φ y $\neg\varphi$ (φ es un átomo) aparece en B. Un tableau está cerrado si todas sus ramas están cerradas.*

4.5.1 Corrección y completitud

La corrección del método se basa en que la existencia de un modelo para un tableau se preserva durante el proceso de expansión.

Lema 4.5.2 *Si $\mathcal{T}'$ es un tableau obtenido a partir de $\mathcal{T}$ mediante la aplicación de una regla de expansión y $\mathcal{T}$ tiene un modelo, entonces $\mathcal{T}'$ tiene un modelo.*

Demostración. La prueba de los casos α y β es trivial. Para los otros, sea $\mathcal{D}$ la estructura que es modelo de $\mathcal{T}$ y B la rama de $\mathcal{T}$ tal que $\mathcal{D} \models B$. Analicemos los casos:

γ) $\mathcal{D} \models \neg\exists x^s\varphi, t \in s$. Por la semántica, $[\![t]\!]^{\mathcal{D}} \in D^s$, por lo que $\underline{t} = [\![\neg\varphi]\!]^{\mathcal{D}}_{[[\![t]\!]^{\mathcal{D}}/x^s]}$ (por la semántica) $= [\![\neg\varphi[t/x^s]]\!]^{\mathcal{D}}$ (por el Lema 4.4.3).

δ) $\mathcal{D} \models \exists x^s\varphi$, por lo que existe $d \in D^s$ tal que $\underline{t} = [\![\varphi]\!]^{\mathcal{D}}_{[d/x^s]}$ (por la semántica) $=$ $[\![\varphi[c/x^s]]\!]^{\mathcal{D}'}$ (por el Lema 4.4.3), donde c es una nueva constante y $\mathcal{D}'$ es como $\mathcal{D}$, pero definiendo $c^{\mathcal{D}'} = d$. En estas condiciones tenemos $\mathcal{D}' \models \varphi[c/x] \wedge c \in s$. ■

Teorema 4.5.3 (Corrección) *Dado un conjunto de Σ-sentencias Φ, si Φ tiene un tableau cerrado entonces Φ es insatisfactible.*

La completitud del método significa que podemos construir un tableau cerrado para cualquier conjunto de sentencias que sea insatisfactible. Como en *LPGD*, la demostración es *à la Hintikka*.

Definición 4.5.4 (Conjunto de Hintikka) *Un conjunto de $\overline{\Sigma}$-sentencias H es un conjunto de Hintikka si satisface las siguientes condiciones:*

1. *Si $\varphi_1 \wedge \varphi_2 \in H$ entonces $\varphi_1, \varphi_2 \in H$.*
2. *Si $\neg(\varphi_1 \wedge \varphi_2) \in H$ entonces $\neg\varphi_1 \in H$ o $\neg\varphi_2 \in H$.*
3. *Si $\neg\neg\varphi \in H$ entonces $\varphi \in H$.*
4. *Si $\exists x^s\varphi \in H$ entonces existe una constante c tal que $\varphi[c/x^s], (c \in s) \in H$.*
5. *Si $\neg\exists x^s\varphi, (t \in s) \in H$ entonces $\neg\varphi[t/x^s] \in H$.*
6. *H es coherente, es decir ninguna fórmula y su negación, siendo ambas literales, pertenecen a H.*

Teorema 4.5.5 *Todo conjunto de Hintikka tiene un modelo.*

Demostración. Sea $\mathcal{D}$ la $\overline{\Sigma}$-estructura (de *Herbrand*) compuesta por:

- $D = T(\overline{\Sigma})$
- $D^s = \{t \in D \,|\, (t \in s) \in H\}$
- $f^{\mathcal{D}} : D^n \to D, f^{\mathcal{D}}(t_1, \ldots, t_n) =_{def} f(t_1, \ldots, t_n)$. En particular, $c^{\mathcal{D}} =_{def} c$
- $P^{\mathcal{D}} : D^n \to \{\underline{t}, \underline{f}\}$, $P^{\mathcal{D}}(t_1, \ldots, t_n) =_{def} \begin{cases} \underline{t} & \text{si } P(t_1, \ldots, t_n) \in H \\ \underline{f} & \text{en otro caso} \end{cases}$

Resulta obvio que $[\![t]\!]^{\mathcal{D}} = t$, para todo $t \in D$. Demostramos por inducción sobre la estructura de φ que $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$, para toda $\varphi \in H$. Veamos los casos relevantes:

- $\varphi = t \in s$. $[\![t]\!]^{\mathcal{D}} = t \in D^s$, por definición de $\mathcal{D}$, por lo que $[\![t \in s]\!]^{\mathcal{D}} = \underline{t}$.
- $\varphi = \neg(t \in s)$. Debido a la condición (6) de la definición anterior, no es posible que $(t \in s) \in H$ y, por tanto, $[\![t]\!]^{\mathcal{D}} = t \notin D^s$ y $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$.
- $\varphi = \exists x^s\psi$. Debido a la condición (4) de la definición anterior, existe c tal que $\psi[c/x], (c \in s) \in H$ por lo que, aplicando hipótesis de inducción, tenemos $[\![c]\!]^{\mathcal{D}} = c \in D^s$ y $[\![\psi[c/x]]\!]^{\mathcal{D}} = \underline{t}$. Usando el Lema 4.4.3 deducimos que $[\![\psi]\!]^{\mathcal{D}}_{[c/x]} = [\![\psi[c/x]]\!]^{\mathcal{D}}$ y, por tanto, $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$.

$$5 : \forall x^s(x^s \in s')$$
$$\mid \quad \gamma \text{ a } 5,\ 1$$
$$6 : a \in s'$$
$$\mid \quad \gamma \text{ a } 3,\ 1$$
$$7 : a \in s' \to \forall y^s(P(a, y^s))$$
$$\beta \text{ a } 7$$
$$8 : \neg a \in s' \qquad\qquad 9 : \forall y^s(P(a, y^s))$$
$$\mid \quad \gamma \text{ a } 4,\ 1$$
$$10 : f(a) \in s$$
$$\mid \quad \gamma \text{ a } 9,\ 10$$
$$11 : P(a, f(a))$$

Figura 4.1: El tableau básico cerrado $\mathcal{T}$.

- $\varphi = \neg\exists x^s\psi$. Sea $t \in D^s$, entonces $(t \in s) \in H$ y, por tanto, $\neg\psi[t/x] \in H$, por la condición (5) de la definición anterior. Aplicando hipótesis de inducción y el Lema 4.4.3 tenemos $\underline{t} = [\![\neg\psi[t/x]]\!]^{\mathcal{D}} = [\![\neg\psi]\!]^{\mathcal{D}}_{[t/x]}$. Concluimos $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$, por la semántica de los cuantificadores. ■

Teorema 4.5.6 (Completitud) *Dado un conjunto de Σ-sentencias Φ, si Φ es insatisfactible entonces Φ tiene un tableau básico cerrado.*

Demostración. Supongamos que Φ no tiene un tableau básico cerrado, entonces podemos construir sistemáticamente un tableau $\mathcal{T}$, de forma que contenga una rama abierta con las propiedades de saturación de un conjunto de Hintikka. Entonces, por el lema previo, esta rama tiene un modelo que sirve para asegurar que Φ es satisfactible. ■

Ejemplo 4.5.7 *El tableau básico $\mathcal{T}$ de la Figura 4.1 prueba el razonamiento presentado en el Ejemplo 4.3.2. Hemos etiquetado con* 5 *la negación de la conclusión. Obsérvese que el tableau $\mathcal{T}$ está cerrado por 6 y 8 en la rama de la izquierda y 2 y 11 en la de la derecha.*

Si formuláramos el ejemplo anterior en *LPO* mediante fórmulas del tipo $\forall x(S(x) \to (S'(x) \to \forall y(S(y) \to P(x, y))))$ (para la 3, por ejemplo), la eficacia del método de tableaux básicos disminuiría debido a que el número de ramas que hay que cerrar aumenta considerablemente y además la información del género en las γ-aplicaciones se pierde (por ejemplo, x en la fórmula previa podría instanciarse por el término $f(f(f(a)))$ lo cual es inviable en *LDT*). En este sentido, las deducciones permitidas en *LDT* son más eficaces que si representáramos los géneros como predicados unarios en *LPO*.

4.6 Tableaux con variables libres

Ahora asumimos que la signatura extendida $\overline{\Sigma}$ también contiene un conjunto numerable de nuevos símbolos de función. El método de tableaux con variables libres consta de las siguientes dos nuevas reglas para los cuantificadores:

$$\gamma') \quad \frac{\neg\exists x^s \varphi}{\neg\varphi[y^s/x^s]} \qquad\qquad \delta') \quad \frac{\exists x^s \varphi}{\begin{array}{c}\varphi[f(x_1^{s_1},\ldots,x_n^{s_n})/x^s] \\ f(x_1^{s_1},\ldots,x_n^{s_n}) \in s\end{array}}$$

En γ', y^s es una variable libre nueva en el tableau; en δ', f es un nuevo símbolo de función aplicado a las variables libres que aparecen en la rama.

Obviamente, las variables libres de un tableau pueden ser sustituidas. Como las variables están tipadas, la aplicación de una sustitución es correcta si está bien tipada con respecto a todo el tableau.

Definición 4.6.1 (Sustituciones bien tipadas con respecto a un tableau) *Una sustitución τ está bien tipada con respecto a un tableau $\mathcal{T}$ con ramas $B_1, \ldots, B_n$, si $\tau|_{var(B_i)}$ está bien tipada con respecto a la teoría incluida en B_i, para todo $1 \leq i \leq n$.*

Las sustituciones bien tipadas pueden ser aplicadas de forma segura en los tableaux con variables libres. Denotamos con $\mathcal{S}1$ el sistema de tableaux compuesto por α, β, γ', δ' y la regla de sustitutividad *sub* definida a continuación.

Definición 4.6.2 (sub) *Si $\mathcal{T}$ es un tableau con variables libres y τ es una sustitución idempotente bien tipada con respecto a $\mathcal{T}$ entonces $\mathcal{T}\tau$ es un tableau con variables libres.*

Los conceptos de rama cerrada y tableau cerrado se definen como para los tableaux básicos (Definición 4.5.1).

El sentido en el que el sistema $\mathcal{S}1$ preserva la corrección debe precisarse ya que los dominios pueden estar vacíos. De hecho, cuando manejamos estructuras con géneros vacíos, la regla γ' puede conducirnos a conclusiones erróneas (fórmulas sin modelo) a partir de hipótesis ciertas (fórmulas con modelo). Por ejemplo, la sentencia $\varphi = \forall x^s(\neg P(a) \wedge P(a))$ es satisfactible en estructuras con $D^s = \emptyset$, pero después de una γ'-expansión obtenemos el conjunto de sentencias $\{\varphi, \neg P(a) \wedge P(a)\}$ que obviamente es insatisfactible. Para evitar estos casos, probamos la correción del método en el siguiente sentido.

Definición 4.6.3 *Una estructura $\mathcal{D}$ es modelo de un tableau con variables libres $\mathcal{T}$ si para toda valoración ρ para $\mathcal{D}$, que verifique $var(\mathcal{T}) \subseteq dom(\rho)$, existe una rama B tal que $\langle\mathcal{D}, \rho\rangle \models B$.*

Lema 4.6.4 *Sea $\mathcal{D}$ una estructura sin géneros vacíos que es modelo de un tableau con variables libres $\mathcal{T}$. Si $\mathcal{T}'$ es un tableau construido a partir de $\mathcal{T}$ mediante la aplicación de una de las $\mathcal{S}1$-reglas, entonces $\mathcal{T}'$ tiene un modelo sin géneros vacíos.*

Demostración. Procedemos por casos. Para α y β la demostración es trivial. Para el resto tenemos:

γ' Veamos que $\mathcal{D}$ también es modelo de $\mathcal{T}'$. Supongamos que la fórmula elegida $\neg\exists x^s\varphi$ aparece en una rama B de $\mathcal{T}$. Sea ρ una valoración tal que $var(\mathcal{T}') \subseteq dom(\rho)$, entonces $var(\mathcal{T}) \subseteq dom(\rho)$, por construcción. Por hipótesis existe una rama $B' \in \mathcal{T}$ tal que $\langle\mathcal{D},\rho\rangle \models B'$. Si B' no es B hemos acabado; si no, $\langle\mathcal{D},\rho\rangle \models \neg\exists x^s\varphi$ y, por tanto, $[\![\neg\varphi]\!]^{\mathcal{D}}_{\rho[d/x^s]} = \underline{t}$, para todo $d \in D^s \neq \emptyset$. Probamos que $\langle\mathcal{D},\rho\rangle \models \neg\varphi[y^s/x^s]$ distinguiendo casos. Si $x^s \notin var(\varphi)$ entonces $\neg\varphi[y^s/x^s] = \neg\varphi$ y $[\![\neg\varphi]\!]^{\mathcal{D}}_{\rho} = [\![\neg\varphi]\!]^{\mathcal{D}}_{\rho[d/x^s]} = \underline{t}$, donde d es un elemento arbitrario de $D^s(\neq\emptyset)$. Si $x^s \in var(\varphi)$ entonces $y^s \in dom(\rho)$ y $\rho(y^s) \in D^s$. En estas condiciones $[\![\neg\varphi[y^s/x^s]]\!]^{\mathcal{D}}_{\rho} = [\![\neg\varphi]\!]^{\mathcal{D}}_{\rho[\rho(y^s)/x^s]} = \underline{t}$, por el Lema 4.4.3 aplicado a la teoría $\{y^s \in s\}$.

δ' Costruimos la nueva estructura extendiendo el modelo $\mathcal{D}$ de $\mathcal{T}$. Supongamos que la fórmula elegida $\exists x^s\varphi$, aparece en una rama B de $\mathcal{T}$. Sea $\mathcal{D}'$ la estructura que coincide con $\mathcal{D}$ salvo en la interpretación de f, que fijaremos más tarde. Sea ρ una valoración tal que $var(\mathcal{T}) = var(\mathcal{T}') \subseteq dom(\rho)$. Por hipótesis, existe una rama B' en $\mathcal{T}$ tal que $\langle\mathcal{D}',\rho\rangle \models B'$, ya que $\mathcal{D}'$ se comporta como $\mathcal{D}$ sobre $\mathcal{T}$. Si B' no es B hemos acabado; si no, $\langle\mathcal{D}',\rho\rangle \models \exists x^s\varphi$ y, por tanto, existe $d_\rho \in D^s$ tal que $[\![\varphi]\!]^{\mathcal{D}'}_{\rho[d_\rho/x^s]} = \underline{t}$. Ahora bien, como f es nueva, supongamos que hemos fijado $f^{\mathcal{D}'}(\rho(x_1^{s_1}),\ldots,\rho(x_n^{s_n})) = d_\rho$, entonces $[\![f(x_1^{s_1},\ldots,x_n^{s_n}) \in s]\!]^{\mathcal{D}'}_{\rho} = \underline{t}$ y $[\![\varphi]\!]^{\mathcal{D}'}_{\rho[d_\rho/x^s]} = [\![\varphi]\!]^{\mathcal{D}'}_{\rho[f^{\mathcal{D}'}(\rho(x_1^{s_1}),\ldots,\rho(x_n^{s_n}))/x^s]}$. Aplicando el Lema 4.4.3, usando la teoría $\{f(x_1^{s_1},\ldots,x_n^{s_n}) \in s\}$, obtenemos que $[\![\varphi[f(x_1^{s_1},\ldots,x_n^{s_n})/x^s]]\!]^{\mathcal{D}'}_{\rho} = [\![\varphi]\!]^{\mathcal{D}'}_{\rho[f^{\mathcal{D}'}(\rho(x_1^{s_1}),\ldots,\rho(x_n^{s_n}))/x^s]} = \underline{t}$ y, por tanto, $\langle\mathcal{D}',\rho\rangle$ es modelo de la rama extendida.

sub Sea τ una sustitución idempotente bien tipada con respecto a $\mathcal{T}$. Si $\mathcal{D}$ es el modelo de $\mathcal{T}$, probamos que también es modelo de $\mathcal{T}' = \mathcal{T}\tau$. Sea ρ una valoración tal que $var(\mathcal{T}\tau) \subseteq dom(\rho)$; extendemos ρ para hacerla definida sobre $\mathcal{T}$, lo cual es posible porque $\mathcal{D}$ no tiene géneros vacíos. Esta nueva valoración ρ' se comporta como ρ sobre $\mathcal{T}'$.

Por hipótesis, existe una rama B_1 en $\mathcal{T}$ tal que $\langle\mathcal{D},\rho'\rangle \models B_1$. Si definimos $\tau_1 = \tau|_{var(B_1)}$, entonces τ_1 está bien tipada con respecto a B_1, por lo que aplicando el Lema 4.4.3 deducimos que, para toda φ con $var(\varphi) \subseteq dom(\rho')$, tenemos $[\![\varphi\tau_1]\!]^{\mathcal{D}}_{\rho'} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'\tau_1}$ *.

Aplicando hipótesis de nuevo, existe B_2 en $\mathcal{T}$ tal que $\langle\mathcal{D},\rho'\tau_1\rangle \models B_2$. Si $B_1 = B_2$ hemos acabado porque $[\![B_2\tau]\!]^{\mathcal{D}}_{\rho'} = [\![B_2\tau_1]\!]^{\mathcal{D}}_{\rho'} = [\![B_2]\!]^{\mathcal{D}}_{\rho'\tau_1} = \underline{t}$. De lo contrario, sea $\tau_2 = \tau|_{var(B_2)-var(B_1)}$. Como τ_2 está bien tipada con respecto a B_2, usamos el Lema 4.4.3 para deducir que, para toda φ con $var(\varphi) \subseteq dom(\rho'\tau_1)$, tenemos $[\![\varphi\tau_2]\!]^{\mathcal{D}}_{\rho'\tau_1} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'\tau_1\tau_2}$.

Una vez más, existe una rama B_3 en $\mathcal{T}$ tal que $\langle\mathcal{D},\rho'\tau_1\tau_2\rangle \models B_3$. Si $B_3 \in \{B_1,B_2\}$ hemos acabado porque $[\![B_3\tau]\!]^{\mathcal{D}}_{\rho'} = [\![B_3\tau_2\tau_1]\!]^{\mathcal{D}}_{\rho'} = [\![B_3\tau_2]\!]^{\mathcal{D}}_{\rho'\tau_1} = [\![B_3]\!]^{\mathcal{D}}_{\rho'\tau_1\tau_2} = \underline{t}$ (obsérvese que usamos que τ es idempotente). De lo contrario, sea $\tau_3 = \tau|_{var(B_3)-(var(B_1)\cup var(B_2))}$.

* τ_1 es la valoración que evalúa cada variable de su dominio x como $[\![\tau_1(x)]\!]^{\mathcal{D}}_{\rho}$.

$$a \in s$$
$$b \in s$$
$$\neg Q(a)$$
$$\neg P(b)$$

$$Q(x^s) \qquad\qquad P(y^{s'})$$
$$\qquad\qquad\qquad x^s \in s'$$

Figura 4.2: El sketch de tableau $\mathcal{T}$.

Como τ_3 está bien tipada con respecto a B_3, usamos el Lema 4.4.3 para deducir que, para toda φ con $var(\varphi) \subseteq dom(\rho'\tau_1\tau_2)$, tenemos $[\![\varphi\tau_3]\!]^{\mathcal{D}}_{\rho'\tau_1\tau_2} = [\![\varphi]\!]^{\mathcal{D}}_{\rho'\tau_1\tau_2\tau_3}$.

Probamos que $\mathcal{D}$ es modelo de $\mathcal{T}\tau$ sin más que repetir este proceso hasta que consigamos una rama ya usada. Esto sucederá porque $\mathcal{T}$ tiene un número finito de ramas. ■

Teorema 4.6.5 (Corrección de $\mathcal{S}1$) *Dado un conjunto de Σ-sentencias Φ, si Φ tiene un tableau con variables libres cerrado entonces Φ es insatisfactible en estructuras con géneros no vacíos.*

Demostración. Supongamos que Φ es satisfactible en una estructura $\mathcal{D}$ con géneros no vacíos y que Φ tiene un tableau cerrado $\mathcal{T}$. Entonces, por el Lema 4.6.4, existe una estructura sin géneros vacíos $\mathcal{D}'$ que es modelo de $\mathcal{T}$. Por consiguiente, podemos construir una valoración ρ, definida sobre toda variable libre de $\mathcal{T}$, para la que existe una rama B que verifica $\langle \mathcal{D}', \rho \rangle \models B$. Por la semántica, llegamos a contradicción porque B está cerrada. ■

Si se observa la demostración del Lema 4.6.4, se deduce que la exigencia de idempotencia en la regla *sub* resulta crucial. De hecho, si dicha regla sólo pidiera la buena tipificación de τ, el sistema se volvería incorrecto como muestra el siguiente ejemplo.

Ejemplo 4.6.6 *Sea Σ una signatura con géneros s, s' y los símbolos de constante a, b y de predicado Q, P. Sea $\mathcal{D}$ una Σ-estructura tal que $a^{\mathcal{D}} \in D^s \cap D^{s'}$, $b^{\mathcal{D}} \in D^s - D^{s'}$, $P^{\mathcal{D}}$ es cierto en $D^{s'}$, pero falso en $D^s - D^{s'}$, y $Q^{\mathcal{D}}$ es falso en $D^{s'}$, pero cierto en $D^s - D^{s'}$. En estas condiciones, resulta fácil probar que $\mathcal{D}$ es modelo del conjunto de sentencias $\{a \in s, b \in s, \neg Q(a), \neg P(b), \forall x^s(Q(x^s) \vee (\forall y^{s'} P(y^{s'}) \wedge x^s \in s'))\}$. Sin embargo dicho conjunto admite un tableau cerrado cuando la regla sub no exige idempotencia. En efecto, aplicando α, β y γ' adecuadamente, podemos construir el sketch de tableau $\mathcal{T}$ de la Figura 4.2.*

Obsérvese que $\tau = [a/x^s, x^s/y^{s'}]$ está bien tipada con respecto a $\mathcal{T}$ y, si la aplicamos a $\mathcal{T}$, obtenemos el tableau $\mathcal{T}'$ de la Figura 4.3. La sustitución $\tau' = [b/x^s]$ está bien tipada con respecto a $\mathcal{T}'$, por lo que podemos construir el tableau cerrado $\mathcal{T}''$ de la Figura 4.4.

$$
\begin{array}{ccc}
 & a \in s & \\
 & b \in s & \\
 & \neg Q(a) & \\
 & \neg P(b) & \\
\swarrow & & \searrow \\
Q(a) & & P(x^s) \\
 & & a \in s'
\end{array}
$$

Figura 4.3: El tableau $\mathcal{T}'$.

$$
\begin{array}{ccc}
 & a \in s & \\
 & b \in s & \\
 & \neg Q(a) & \\
 & \neg P(b) & \\
\swarrow & & \searrow \\
Q(a) & & P(b) \\
 & & a \in s'
\end{array}
$$

Figura 4.4: El tableau cerrado $\mathcal{T}''$.

Hacemos observar también que la demostración de la corrección del sistema $S1$ difiere de la presentada para los tableaux con variables libres de $LPGD$. En concreto, en $LPGD$, las reglas de expansión preservaban lo que llamabamos *satisfactibilidad básica*, mientras que aquí se conserva la existencia de un modelo que sólo considera las variables libres del tableau. La razón de esta diferencia estriba en que en LDT no se verifica que la composición de sustituciones bien tipadas esté también bien tipada (esta era la principal idea usada en el Lema 3.7.4). De hecho, podemos acudir al ejemplo anterior para encontrar un contraejemplo: τ y τ' están bien tipadas con respecto a $\mathcal{T}$ y $\mathcal{T}'$, respectivamente, pero $\tau\tau' = [a/x^s, b/y^{s'}]$ no está bien tipada con respecto a $\mathcal{T}$. Una vez más, el problema reside en que τ no es idempotente.

A continuación demostramos la completitud del sistema $S1$.

Teorema 4.6.7 (Completitud de $S1$) *Dado un conjunto de Σ-sentencias Φ, si Φ es insatisfactible entonces Φ tiene un tableau con variables libres cerrado.*

Demostración. Como Φ es insatisfactible, existe un tableau básico cerrado $\mathcal{T}$ para Φ, por el Teorema 4.5.6. Ahora mostramos cómo construir sistemáticamente, en $S1$, un tableau con variables libres $\mathcal{T}'$ tal que $\mathcal{T} = \mathcal{T}'$.

- Cada vez que apliquemos α o β en $\mathcal{T}$, las aplicamos también en $\mathcal{T}'$.
- Cada vez que apliquemos γ en $\mathcal{T}$, aplicamos γ' y *sub* en $\mathcal{T}'$, consiguiendo $\mathcal{T}' = \mathcal{T}$ como sigue. Si usamos $(t \in s)$ y $\neg\exists x^s\varphi$ en $\mathcal{T}$, entonces introducimos $\neg\varphi[y^s/x^s]$ en $\mathcal{T}'$ y aplicamos la sustitución $[t/y^s]$. Esto es posible ya que $\mathcal{T}'$ coincide por ahora con $\mathcal{T}$, y por tanto $(t \in s)$ aparece en la rama y t es básico (la sustitución es trivialmente idempotente y está obviamente bien tipada con respecto a $\mathcal{T}'$).
- Cada vez que apliquemos δ en $\mathcal{T}$, aplicamos δ' en $\mathcal{T}'$, usando el mismo símbolo de constante. Obsérvese que esto es posible ya que tras cada uno de estos pasos, $\mathcal{T}'$ permanece básico, por lo que el símbolo de función introducido por δ' es una constante. ■

Como se puede observar, la corrección del sistema $S1$ establecida en el Teorema 4.6.5 no coincide con el inverso de este teorema, al contrario de lo que sucede en los tableaux básicos de la sección anterior. La razón de este hecho se debe a que la regla γ' permite expansiones que no preservan la existencia de un modelo cuando manejamos estructuras con géneros vacíos (véanse los comentarios anteriores a la Definition 4.6.3 para la fórmula $\forall x^s(\neg P(a) \wedge P(a)))$.

Podemos caracterizar las estructuras sin géneros vacíos de forma sintáctica mediante la fórmula $\exists x^s(x^s \in s)$ (o también, introduciendo una constante nueva c_s en la signatura y usando la fórmula $c_s \in s$). En esta condiciones, podemos probar fácilmente para $S1$ que todo conjunto de Σ-sentencias Φ satisface la siguiente propiedad:

Φ es insatisfactible en estructuras con géneros no vacíos si y sólo si $\Phi \cup \{\exists x^s(x^s \in s) \ / \ s \in S\}$ tiene un tableau cerrado.

$a \in s$	$a \in s$
$\neg P(a)$	$\neg P(a)$
$\forall x^s \ (x^s \in s')$	$\forall x^s \ (x^s \in s')$
$\forall u^{s'} \ P(u^{s'})$	$\forall u^{s'} \ P(u^{s'})$
$\vert$	$\vert$
$a \in s'$	$x^s \in s'$
$\vert$	$\vert$
$P(a)$	$P(u^{s'})$

Figura 4.5: Los sketches de tableau $\mathcal{T}$ y $\mathcal{T}'$.

4.7 Unificación rígida tipada

Como en los tableaux clásicos para *LPO*, mejorar los tableaux básicos implica restringir la aplicación de la regla *sub* y sólo usarla para cerrar ramas. Esta idea supone la integración de un cálculo de unificación que busca unificadores bien tipados para parejas de literales potencialmente complementarios que aparecen en las ramas del tableau. Sin embargo, con vistas a diseñar un sistema completo de deducción, los unificadores deben estructurarse en una forma particular como muestra el siguiente ejemplo.

Ejemplo 4.7.1 *Sea $\mathcal{T}$ el sketch de tableau básico cerrado que aparece a la izquierda de la Figura 4.5 y $\mathcal{T}'$ el tableau con variables libres construido como $\mathcal{T}$, que aparece en la parte derecha.*

$\mathcal{T}'$ debería cerrarse resolviendo el problema de unificación $u^{s'} \simeq a$ correspondiente a la única rama de $\mathcal{T}'$; sin embargo este problema no puede ser resuelto por ninguna sustitución bien tipada con respecto a la teoría presente en la rama $\{a \in s, x^s \in s'\}$. No obstante, hay una secuencia de sustituciones unitarias idempotentes $\sigma = [x^s/u^{s'}][a/x^s]$, que relaciona ambos tableaux y que está bien tipada gradualmente, en el sentido de que cada componente unitaria está bien tipada tras la aplicación de su prefijo en la secuencia. Por eso σ puede ser aplicada a $\mathcal{T}'$ usando la regla sub dos veces. Obsérvese cómo la secuencia σ señala el orden seguido por las aplicaciones de la regla sub en $\mathcal{T}'$, que corresponde al orden seguido por las γ-aplicaciones en $\mathcal{T}$.

Por tanto definiremos un cálculo de unificación que eleve cualquier tableau básico cerrado a otro con variables libres, también cerrado, sin más que derivar una *secuencia* de sustituciones unitarias bien tipadas. Previamente definimos dos conceptos, el de triangularidad, que captura el orden seguido por las γ-aplicaciones efectuadas en los tableaux básicos, y el de secuencia bien tipada, que extiende la noción de buena tipificación a las secuencias.

Definición 4.7.2 *[Kogel 95] Una secuencia de sustituciones unitarias $[t_1/{x_1}^{s_1}] \ldots [t_n/{x_n}^{s_n}]$ es triangular si satisface:*

1. $var(t_i) \cap \{x_1{}^{s_1}, \ldots, x_i{}^{s_i}\} = \emptyset$*, para todo* $1 \leq i \leq n$
2. $x_i \neq x_j$*, para todos* $1 \leq i < j \leq n$.

Definición 4.7.3 *Sean* $\sigma = \sigma_1 \ldots \sigma_n$, $\mathcal{L}$ *y* $\mathcal{T}$ *una secuencia triangular de sustituciones unitarias, una teoría y un tableau con variables libres, respectivamente. Decimos que* σ *está bien tipada con respecto a* $\mathcal{L}$ *(resp.* $\mathcal{T}$*), si* σ_i *está bien tipada con respecto a* $\mathcal{L}\sigma_1 \ldots \sigma_{i-1}$ *(resp.* $\mathcal{T}\sigma_1 \ldots \sigma_{i-1}$*), para todo* $1 \leq i \leq n$.

Obsérvese que las secuencias bien tipadas con respecto a los tableaux pueden ser aplicadas de forma segura usando la regla *sub*, ya que aplicamos gradualmente cada una de sus componentes unitarias. Así, en el Ejemplo 4.7.1, $[x^s/u^{s'}][a/x^s]$ está bien tipada con respecto a $\mathcal{T}'$ y puede ser usada para cerrarlo. Consecuentemente debemos diseñar un cálculo que obtenga secuencias bien tipadas en lugar de una única sustitución bien tipada.

4.7.1 El cálculo tipado $\mathcal{C}$

En esta sección presentamos cómo resolver los problemas de unificación que surgen cuando cerramos una única rama. Concretamente, un problema de *Unificación Rígida Tipada*, en adelante *URT*-problema, tiene la siguiente estructura:

Dada una teoría finita $\mathcal{L}$ *y un conjunto finito de ecuaciones* Γ*, ¿existe una secuencia de sustituciones unitarias bien tipada con respecto a* $\mathcal{L}$ *que unifique* Γ*?*

Para resolver *URT*-problemas, definimos el cálculo de unificación $\mathcal{C}$. Las reglas de $\mathcal{C}$ que no son de fallo tienen la forma

$$\frac{\Gamma \quad \sigma_1 \ldots \sigma_n}{\Gamma' \quad \sigma_1 \ldots \sigma_n \sigma'}$$

donde Γ, Γ' son conjuntos de ecuaciones orientadas, es decir distiguimos el lado derecho del izquierdo, y $\sigma_1 \ldots \sigma_n$, $\sigma_1 \ldots \sigma_n \sigma'$ son sucesiones de sustituciones unitarias. $\mathcal{C}$ está compuesto por diez reglas: las seis reglas estándar de unificación sintáctica más cuatro para tener en cuenta las declaraciones de términos.

Las reglas estándar de unificación sintáctica

(Tautología)
$$\frac{x^s \simeq x^s,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{\Gamma \quad \sigma_1 \ldots \sigma_n}$$

(Orientación)
$$\frac{t \simeq x^s,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{x^s \simeq t,\ \Gamma \quad \sigma_1 \ldots \sigma_n}$$

si t no es una variable

(Descomposición)
$$\frac{f(t_1, \ldots, t_q) \simeq f(t'_1, \ldots, t'_q),\ \Gamma \quad \sigma_1 \ldots \sigma_n}{t_1 \simeq t'_1, \ldots, t_q \simeq t'_q,\ \Gamma \quad \sigma_1 \ldots \sigma_n}$$

(Aplicación)
$$\frac{x^s \simeq t,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{x^s \simeq t,\ \Gamma[t/x^s] \quad \sigma_1 \ldots \sigma_n}$$

si $x^s \notin var(t)$ y $x^s \in var(\Gamma)$

(Conflicto) $$\frac{g(t_1, ..., t_p) \simeq f(t'_1, ..., t'_q),\ \Gamma \quad \sigma_1 \ldots \sigma_n}{Fallo}$$

si $f \neq g$

(Ciclo) $$\frac{x^s \simeq t,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{Fallo}$$

si $x^s \in var(t)$

Las reglas tipadas de $\mathcal{C}$

(LW) Instanciación por la izquierda $$\frac{x^s \simeq t',\ \Gamma \quad \sigma_1 \ldots \sigma_n}{t \simeq t',\ \Gamma \quad \sigma_1 \ldots \sigma_n[t/x^s]}$$

si $(t \in s) \in \mathcal{L}\sigma_1 ... \sigma_n$ y $x^s \notin var(t)$

(RW) Instanciación por la derecha $$\frac{y^{s'} \simeq x^s,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{y^{s'} \simeq t,\ \Gamma[t/x^s] \quad \sigma_1 \ldots \sigma_n[t/x^s]}$$

si $(t \in s) \in \mathcal{L}\sigma_1 ... \sigma_n$ y $x^s \notin var(t)$

(FWF) Fallo de instanciación funcional $$\frac{x^s \simeq f(t_1, ..., t_q),\ \Gamma \quad \sigma_1 \ldots \sigma_n}{Fallo}$$

si no existe ninguna declaración $t \in s$ en $\mathcal{L}\sigma_1 ... \sigma_n$ tal que $x^s \notin var(t)$

(VWF) Fallo de instanciación de variables $$\frac{y^{s'} \simeq x^s,\ \Gamma \quad \sigma_1 \ldots \sigma_n}{Fallo}$$

si no existe ninguna declaración $t \in s$ en $\mathcal{L}\sigma_1 ... \sigma_n$ tal que $x^s \notin var(t)$, ni $t \in s'$ tal que $y^{s'} \notin var(t)$

Cuando resolvemos *URT*-problemas, la aplicación de las seis reglas estándar siempre tiene preferencia. Además, suponemos que disponemos de un algoritmo terminante $\mathcal{A}$, para unificación sintáctica, que transforma un conjunto de ecuaciones Γ en *Fallo* o en un conjunto de ecuaciones resuelto, mediante la aplicación no determinista de las seis reglas estándar. En este sentido, el algoritmo $\mathcal{A}$ funciona como una caja negra, por lo que nos olvidamos del indeterminismo generado por la aplicación de sus reglas. La incorporación de cálculos auxiliares para resolver problemas conocidos es una técnica ya usada en otras áreas [DV 98], [NR 95].

Definición 4.7.4 *Sea Γ un conjunto de ecuaciones y $\sigma = \sigma_1 \ldots \sigma_n$ una secuencia de sustituciones unitarias. Un $\mathcal{C}$-paso estándar consiste en la aplicación del algoritmo $\mathcal{A}$ sobre el par $\langle \Gamma, \sigma \rangle$ hasta obtener Fallo o alcanzar un conjunto de ecuaciones resuelto Γ'. Un $\mathcal{C}$-paso tipado consiste en la aplicación de una regla tipada sobre el par $\langle \Gamma, \sigma \rangle$ usando una teoría. Un $\mathcal{C}$-paso es un $\mathcal{C}$-paso estándar o un $\mathcal{C}$-paso tipado. Escribimos $\langle \Gamma, \sigma_1 \ldots \sigma_n \rangle \vdash_{\mathcal{C}} \langle \Gamma', \sigma_1 \ldots \sigma_n \sigma_{n'} \rangle$ ($n' \in \{n, n+1\}$) (resp. $\langle \Gamma, \sigma_1 \ldots \sigma_n \rangle \vdash_{\mathcal{C}}$ Fallo) para expresar un $\mathcal{C}$-paso sin fallo (resp. con fallo).*

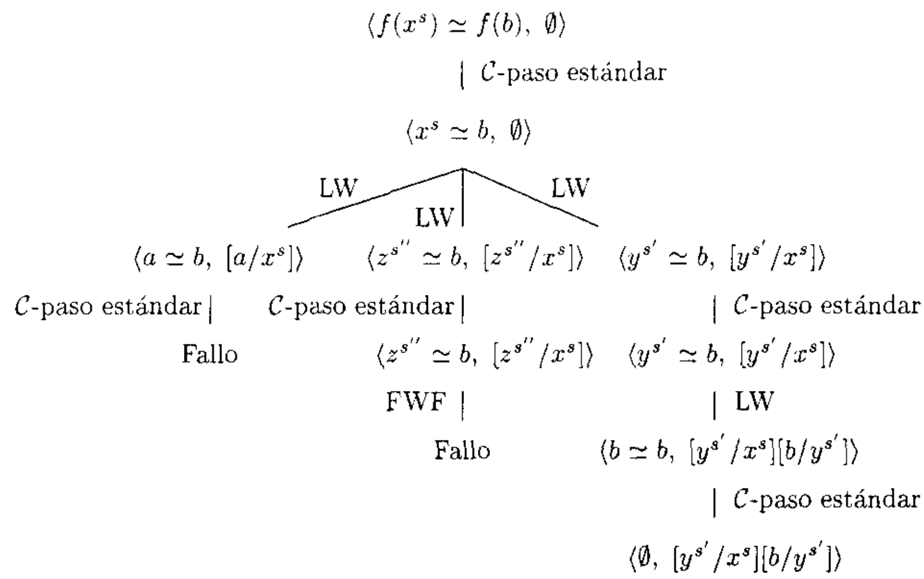

Figura 4.6: Cómputo del $\mathcal{C}$-unificador $[y^{s'}/x^s][b/y^{s'}]$ de Γ con respecto a $\mathcal{L}$.

Decimos que el cálculo $\mathcal{C}$ unifica un conjunto de ecuaciones Γ con respecto a una teoría $\mathcal{L}$ mediante la secuencia de sustituciones unitarias $\sigma_1 \ldots \sigma_n$, o también que $\sigma_1 \ldots \sigma_n$ es un $\mathcal{C}$-unificador de Γ con respecto a $\mathcal{L}$, si existe una cadena de $\mathcal{C}$-pasos, alternando $\mathcal{C}$-pasos estándar y $\mathcal{C}$-pasos tipados, que comienza con $\langle\Gamma, \emptyset\rangle$ y finaliza con $\langle\emptyset, \sigma_1 \ldots \sigma_n\rangle$.

Obsérvese que los $\mathcal{C}$-pasos estándar no añaden nuevos elementos a la secuencia de sustituciones unitarias. Además, estos pasos pueden ser innecesarios si el conjunto de ecuaciones ya está en forma resuelta tras la aplicación de un $\mathcal{C}$-paso tipado. Obsérvese también que los $\mathcal{C}$-pasos tipados se aplican siempre sobre conjuntos de ecuaciones en forma resuelta.

Podemos entender la computación de una solución para un *URT*-problema como la búsqueda de $\mathcal{C}$-unificadores en un $\mathcal{C}$-árbol de derivación: los nodos son o bien pares $\langle\Gamma, \sigma\rangle$ o bien hojas de *Fallo*, y las ramas alternan $\mathcal{C}$-pasos estándar y $\mathcal{C}$-pasos tipados. Las ramificaciones en dicho árbol se deben al indeterminismo explícito generado por los $\mathcal{C}$-pasos tipados; el indeterminismo derivado de la unificación sintáctica se oculta de forma implícita en el algoritmo $\mathcal{A}$. Las hojas son o bien pares con éxito de la forma $\langle\emptyset, \sigma\rangle$ o bien hojas de *Fallo*. Como veremos, si un nodo produce *Fallo* tras la aplicación de un $\mathcal{C}$-paso estándar podemos podar la búsqueda en el subárbol que cuelga desde dicho nodo, mientras que si el *Fallo* se produce tras la aplicación de un $\mathcal{C}$-paso tipado, podemos podar la búsqueda en el subárbol que cuelga desde su padre.

Ejemplo 4.7.5 *Sea $\mathcal{L} = \{a \in s, y^{s'} \in s, z^{s''} \in s, b \in s'\}$ y $\Gamma = \{f(x^s) \simeq f(b)\}$. El $\mathcal{C}$-árbol de derivación para este URT-problema aparece en la Figura 4.6.*

La primera rama acaba en una hoja de Fallo después de un $\mathcal{C}$-paso estándar, y la segunda tras un $\mathcal{C}$-paso tipado. Con la tercera rama obtenemos el único $\mathcal{C}$-unificador

$[y^{s'}/x^s][b/y^{s'}]$.

4.7.2 Propiedades del cálculo $\mathcal{C}$

Primero demostramos que el cálculo de unificación $\mathcal{C}$ siempre termina, independientemente del *URT*-problema.

Teorema 4.7.6 (Terminación) *El $\mathcal{C}$-árbol de derivación de todo URT-problema es finito.*

Demostración. El indeterminismo debido a la aplicación de las $\mathcal{C}$-reglas tipadas es finito ya que, en cada nodo sin *Fallo*, el conjunto de ecuaciones y la teoría en curso son finitas. Por esta razón, el $\mathcal{C}$-árbol de derivación está finitamente ramificado.

Cada $\mathcal{C}$-paso tipado produce una sustitución unitaria $[t/x^s]$ que elimina la variable x^s del conjunto (finito) de ecuaciones y de la teoría en curso; podemos introducir nuevas variables en las ecuaciones a través de t, pero estas variables ya aparecen en la teoría en curso (finita) y son distintas de x^s ($x^s \notin var(t)$). Por tanto, la profundidad de cada rama es finita, ya que el número de variables del conjunto inicial de ecuaciones y de la teoría era finito. ■

El cálculo $\mathcal{C}$ es correcto en el sentido de que, dado un conjunto de ecuaciones Γ y una teoría $\mathcal{L}$, cada $\mathcal{C}$-unificador es una solución del correspondiente *URT*-problema.

Teorema 4.7.7 (Corrección) *Sean $\Gamma, \mathcal{L}$ y σ un conjunto de ecuaciones, una teoría y una secuencia de sustituciones unitarias, respectivamente. Si σ es un $\mathcal{C}$-unificador de Γ con respecto a $\mathcal{L}$ entonces:*

1. *σ está bién tipado con respecto a $\mathcal{L}$.*
2. *σ unifica Γ.*

Demostración. (1) Sea $\sigma = \sigma_1 \dots \sigma_n$. Después de añadir $\sigma_i (= [t_i/x_i])$ a la secuencia, x_i desaparece de Γ_{i+1}. Esto es inmediato para la regla *RW*; para la regla *LW* esto se deduce ya que Γ_i está en forma resuelta y por tanto x_i sólo aparece una vez en Γ_i. También es obvio que x_i desaparece de $\mathcal{L}\sigma_1 \dots \sigma_i$. Por esto las variables sustituidas de σ son diferentes dos a dos. Además, por la definición de las reglas *LW* y *RW*, $x_i \notin var(t_i)$. En consecuencia, σ es triangular. Visto esto, la buena tipificación de σ se obtiene de forma inmediata a partir de la definición de las reglas *LW* y *RW*.

(2) Es suficiente probar que después de cada $\mathcal{C}$-paso $\langle \Gamma, \sigma_1 \dots \sigma_n \rangle \vdash_{\mathcal{C}} \langle \Gamma', \sigma_1 \dots \sigma_n \sigma_{n'} \rangle$, si θ unifica Γ' entonces $\sigma_{n'}\theta$ unifica Γ. Si el paso es estándar, esto es trivial ($\sigma_{n'}$ no existe como tal); si se trata de un paso tipado, digamos que se debe a la aplicación de la regla *LW*, entonces Γ está en forma resuelta, $\sigma_{n'} = [t/x^s]$, y para la única ecuación de $\Gamma \backslash \Gamma'$ tenemos:

$$x^s \sigma_{n'} \theta = t\theta = t'\theta = t'\sigma_{n'}\theta \text{ (pues } x^s \notin var(t') \text{ ya que } \Gamma \text{ está en forma resuelta).}$$

Análogamente si el $\mathcal{C}$-paso se ha dado aplicando la regla *RW*. ■

La completitud de $\mathcal{C}$ debería interpretarse como sigue: si existe una secuencia de sustituciones unitarias σ bien tipada con respecto a $\mathcal{L}$ que unifica Γ, entonces $\mathcal{C}$ unifica Γ con respecto a $\mathcal{L}$ mediante una secuencia τ que es más general que σ[†]. Sin embargo, sólo estamos interesados en elevar una clase particular de secuencias de sustituciones unitarias, aquellas sucesiones σ que se deducen de un tableau básico cerrado $\mathcal{T}$ de la forma siguiente. Supongamos que $\mathcal{T}'$ es un tableau con variables libres construido como $\mathcal{T}$, entonces σ se obtiene añadiendo a la secuencia, las sustituciones unitarias que corresponden a las γ-aplicaciones usadas en $\mathcal{T}$; es decir, si $\forall x^s \varphi$ y $t \in s$ se usan en $\mathcal{T}$ entonces añadimos $[t'/x^s]$ al comienzo de la secuencia σ actual, donde la declaración $t' \in s$ es la que corresponde a $t \in s$ en $\mathcal{T}'$ y x^s es la variable libre introducida. En el Ejemplo 4.7.1 obtendríamos $[x^s/u^{s'}][a/x^s]$. Como hemos dicho, estas secuencias son básicas y pueden caracterizarse mediante el concepto de *tipificación superbuena*. Sólo las secuencias superbién tipadas serán consideradas en la completitud de $\mathcal{C}$.

Definición 4.7.8 *Una secuencia triangular de sustituciones unitarias* $[t_1/x_1{}^{s_1}]...[t_n/x_n{}^{s_n}]$ *está superbién tipada con respecto a una teoría* $\mathcal{L}$*, si* $(t_i \in s_i) \in \mathcal{L}$*, para todo* $1 \leq i \leq n$.

En una secuencia superbién tipada, el orden entre las sustituciones unitarias es irrelevante, puesto que las declaraciones de los términos reemplazados aparecen explícitamente en la teoría. Resulta inmediato que cada secuencia superbién tipada está también bien tipada; sin embargo, el resultado inverso no es cierto, por ejemplo $[a/x^s][a/u^{s'}]$ está bien tipada, pero no superbién tipada con respecto a la teoría $\{a \in s, x^s \in s'\}$.

Para demostrar la completitud, examinamos la parte estándar y la tipada del proceso de unificación. Para la primera, supondremos que el algoritmo $\mathcal{A}$ de unificación sintáctica es completo, es decir, falla siempre que el conjunto de ecuaciones dado no es sintácticamente unificable, o calcula un conjunto de ecuaciones resuelto, en caso contrario. Para la parte tipada, probaremos los Lemas 4.7.10 y 4.7.11. Antes presentamos el siguiente lema técnico que establece que la extracción y el movimiento de una componente unitaria a través de una secuencia, desde su lugar de origen hacia la izquierda, preserva la superbuena tipificación y el comportamiento de la secuencia resultante.

Lema 4.7.9 *Sea* $\sigma_1 \ldots \sigma_n$ *una secuencia tal que* $\sigma_i = [t_i/x_i^{s_i}]$*, para todo* $1 \leq i \leq n$*. Para un índice fijo* $m \in \{1, \ldots, n\}$ *definimos* $\sigma'_i = [t_i[t_m/x_m^{s_m}]/x_i^{s_i}]$*, para todo* $1 \leq i \leq m-1$*. Si* $\sigma_1 \ldots \sigma_n$ *está superbién tipada con respecto a una teoría* $\mathcal{L}$ *entonces:*

1. $\sigma'_1 \ldots \sigma'_{m-1}\sigma_{m+1} \ldots \sigma_n$ *está superbién tipada con respecto a* $\mathcal{L}\sigma_m$.
2. $\sigma_m\sigma'_1 \ldots \sigma'_{m-1}\sigma_{m+1} \ldots \sigma_n = \sigma_1 \ldots \sigma_n$.

Demostración. (1) Resulta inmediato que las variables son diferentes dos a dos. Como $var(t_i[t_m/x_m^{s_m}]) \subseteq var(t_i) \cup var(t_m)$, entonces $t_i[t_m/x_m^{s_m}]$ no contiene ninguna variable previamente instanciada en la secuencia, para todo $1 \leq i \leq m-1$; para $m+1 \leq i \leq n$, también

[†] Las secuencias de sustituciones unitarias se comparan a través de las respectivas sustituciones resultantes de la composición de sus componentes unitarias.

es cierto de manera trivial. En estas condiciones, la secuencia $\sigma'_1 \ldots \sigma'_{m-1}\sigma_{m+1} \ldots \sigma_n$ es triangular.

Por la superbuena tipificación de $\sigma_1 \ldots \sigma_n$ tenemos $(t_i \in s_i) \in \mathcal{L}$, para todo $1 \leq i \leq m-1$, por lo que $(t_i[t_m/x_m{}^{s_m}] \in s_i) \in \mathcal{L}\sigma_m$. Para $m+1 \leq i \leq n$, la superbuena tipificación se consigue por triangularidad.

(2) Inmediato por la definición de $\sigma'_1 \ldots \sigma'_{m-1}$. ∎

Los dos lemas siguientes demuestran la completitud del caso tipado. Si un conjunto de ecuaciones es unificable vía una secuencia superbién tipada entonces podemos dar un $\mathcal{C}$-paso tipado sin fallo, mediante la extracción de una componente unitaria de la secuencia tal como se explica en el lema anterior. Obsérvese que este paso puede darse porque, en una secuencia superbién tipada, la declaración de cada término reemplazado aparece explícitamente en la teoría, independientemente del lugar donde aparezca en la secuencia. Por el contrario, si podemos aplicar un $\mathcal{C}$-paso tipado con fallo, entonces el conjunto de ecuaciones no es unificable mediante ninguna secuencia superbién tipada.

Lema 4.7.10 (Completitud tipada) *Sean Γ y $\mathcal{L}$ un conjunto de ecuaciones no vacío en forma resuelta y una teoría, respectivamente. Sea $\tau = \tau_1 \ldots \tau_n$ una secuencia superbién tipada con respecto a $\mathcal{L}$ que unifica Γ. Entonces existe un conjunto de ecuaciones Γ' y una sustitución unitaria σ tal que $\langle \Gamma, \emptyset \rangle \vdash_{\mathcal{C}} \langle \Gamma', \sigma \rangle$. Además existe otra secuencia $\theta_1 \ldots \theta_k$ que está superbién tipada con respecto a $\mathcal{L}\sigma$ y que unifica Γ'.*

Demostración. Sea $x^s \simeq t' \in \Gamma$ y distingamos casos dependiendo de la forma de t':

(i) $t' = f(\ldots)$. Puesto que τ está superbién tipada con respecto a $\mathcal{L}$ y unifica Γ, existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$, $(t \in s) \in \mathcal{L}$ y $x^s \notin var(t)$. Por tanto podemos aplicar *LW* obteniendo $\Gamma' = (\Gamma \backslash \{x^s \simeq t'\}) \cup \{t \simeq t'\}$ y $\sigma = [t/x^s]$. Construimos la nueva secuencia $\theta_1 \ldots \theta_k$ extrayendo τ_i de τ, como se indica en el Lema 4.7.9. Este lema asegura que la secuencia así obtenida está superbién tipada con respecto a $\mathcal{L}[t/x^s]$ y verifica que $[t/x^s]\theta_1 \ldots \theta_k$ y τ se comportan igual, por lo que $\theta_1 \ldots \theta_k$ unifica Γ'.

(ii) $t' = y^{s'}$. Puesto que τ unifica Γ, si existe $1 \leq i \leq n$ tal que $\tau_i = [t/y^{s'}]$, entonces $(t \in s') \in \mathcal{L}$ e $y^{s'} \notin var(t)$, por la superbuena tipificación de τ; por tanto podemos aplicar *RW* obteniendo $\Gamma' = (\Gamma[t/y^{s'}] \backslash \{x^s \simeq y^{s'}\}) \cup \{x^s \simeq t\}$ y $\sigma = [t/y^{s'}]$. Construimos la nueva secuencia $\theta_1 \ldots \theta_k$ extrayendo τ_i de τ, como en el Lemma 4.7.9, y razonamos como en el caso anterior. Si existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$, se razona como en el caso anterior, aplicando *LW*. ∎

Lema 4.7.11 (Fallo Tipado) *Sean Γ y $\mathcal{L}$ un conjunto de ecuaciones resuelto y una teoría, respectivamente. Si $\langle \Gamma, \emptyset \rangle \vdash_{\mathcal{C}}$ Fallo después de un $\mathcal{C}$-paso tipado entonces no existe ninguna secuencia bien tipada, y por tanto ninguna superbién tipada, con respecto a $\mathcal{L}$ que unifique Γ.*

Demostración. Supongamos que $\tau = \tau_1 \ldots \tau_n$ es una secuencia bien tipada con respecto a $\mathcal{L}$ que unifica Γ. Distinguimos los dos casos posibles:

(i) Hemos aplicado la regla *FWF* a $x^s \simeq f(t_1 \ldots t_n) \in \Gamma$. Puesto que τ unifica Γ, existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$. Como la secuencia está bien tipada, existe t' tal que $t'\tau_1 \ldots \tau_{i-1} = t$ y $(t' \in s) \in \mathcal{L}$. Por triangularidad de τ, $x^s \notin var(t)$ y entonces, $x^s \notin var(t')$. En estas condiciones obtenemos contradicción ya que *LW* puede ser aplicada.

(ii) Hemos aplicado la regla *VWF* a $x^s \simeq y^{s'} \in \Gamma$. Puesto que τ unifica Γ, si existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$, entonces podemos comprobar que *LW* puede aplicarse, como en el caso anterior; si no es así, podemos aplicar *RW* de forma similar. ∎

Teorema 4.7.12 (Completitud) *Sean Γ y $\mathcal{L}$ un conjunto de ecuaciones y una teoría, respectivamente. Sea $\sigma_1 \ldots \sigma_n$ una secuencia superbién tipada con respecto a $\mathcal{L}$ que unifica Γ. Entonces existe un $\mathcal{C}$-unificador de Γ con respecto a $\mathcal{L}$.*

Demostración. Supongamos que cada hoja del correspondiente $\mathcal{C}$-árbol de derivación es un nodo de *Fallo*, entonces $\Gamma \neq \emptyset$. Sea h una hoja cualquiera. Iterando el Lema 4.7.10, existe un nodo interno $\langle \Gamma', \tau_1 \ldots \tau_m \rangle$ y una secuencia superbién tipada $\theta_1 \ldots \theta_l$ con respecto a $\mathcal{L}\tau_1 \ldots \tau_m$, que unifica Γ'. Además este nodo es padre de h por lo que se cumple uno de los siguientes casos:

(i) Un $\mathcal{C}$-paso estándar produce *Fallo* en $\langle \Gamma', \tau_1 \ldots \tau_m \rangle$. Esto es imposible porque $\theta_1 \ldots \theta_l$ unifica Γ'.

(ii) Un $\mathcal{C}$-paso tipado produce *Fallo* en $\langle \Gamma', \tau_1 \ldots \tau_m \rangle$. Por el Lema 4.7.11, no existe una secuencia bien tipada con respecto a $\mathcal{L}\tau_1 \ldots \tau_m$ que unifique Γ', lo que se contradice con el hecho de que $\theta_1 \ldots \theta_l$ está superbién tipada (y, por tanto, bien tipada). ∎

Usando los resultados de corrección y completitud, podemos resolver cualquier *URT*-problema dado, sin más que examinar su $\mathcal{C}$-árbol de derivación asociado.

Corolario 4.7.13 *El URT-problema es decidible.*

Demostración. Dado un *URT*-problema y su $\mathcal{C}$-árbol de derivación asociado:

(i) respondemos *sí* siempre que exista una hoja con éxito. Esta respuesta es correcta por el Teorema 4.7.7,

(ii) respondemos *no* siempre que toda rama acabe en un nodo de *Fallo*. En este caso no existe una secuencia superbién tipada con respecto a la teoría, por el Teorema 4.7.12. Aunque las nociones de superbuena y buena tipificación no son iguales, esta respuesta es correcta puesto que su mutua existencia es equivalente, como pasamos a demostrar en la siguiente subsección. ∎

4.7.3 Buena tipificación frente a superbuena tipificación con respecto a teorías

Vamos a mostrar cómo obtener secuencias superbién tipadas con respecto a una teoría a partir de otra bien tipada con respecto a la misma teoría. Presentamos las principales ideas sobre el siguiente ejemplo.

Consideremos la teoría $\mathcal{L} = \{y^{s'} \in s'', u^{s''} \in s, a \in s'\}$. La secuencia $\tau = [a/y^{s'}][u^{s''}/x^s]$ $[a/u^{s''}]$ no está superbién tipada con respecto a $\mathcal{L}$, puesto que $(a \in s'') \notin \mathcal{L}$, pero sí está bien tipada, ya que dicha declaración pertenece a $\mathcal{L}[a/y^{s'}][u^{s''}/x^s]$.

Para recuperar la superbuena tipificación[‡] cambiamos $[a/u^{s''}]$ por $[y^{s'}/u^{s''}]$ y, para conservar la triangularidad, reorganizamos el conjunto de sustituciones unitarias en una nueva secuencia; por ejemplo, $[u^{s''}/x^s][y^{s'}/u^{s''}][a/y^{s'}]$ está ahora superbién tipada con respecto a $\mathcal{L}$.

Brevemente, la idea de restaurar la superbuena tipificación, incluyendo triangularidad, puede explicarse como sigue. Dada una secuencia $\tau = \tau_1 \dots \tau_n$, donde $\tau_i = [t_i/x_i^{s_i}]$, para todo $1 \leq i \leq n$:

1. Buscamos el primer τ_i que destruya la superbuena tipificación de la secuencia.

2. Buscamos la declaración $(t'_i \in s_i) \in \mathcal{L}$ tal que $t'_i\tau_1 \dots \tau_{i-1} = t_i$. Esta declaración debe existir por la buena tipificación de τ con respecto a $\mathcal{L}$; además $x_i \notin var(t'_i)$.

3. Cambiamos τ_i por $\tau'_i = [t'_i/x_i^{s_i}]$. La nueva secuencia puede ya no ser triangular, por lo que reordenamos sus componentes como sigue:

   - Movemos τ'_i desde su lugar hacia el comienzo de la secuencia, hasta que no exista $j < i$ tal que $x_j \in var(t'_i)$; entonces t'_i no contendrá ninguna variable previamente instanciada por la secuencia. En el ejemplo, pondremos $[y^{s'}/u^{s''}]$ a la izquierda de $[a/y^{s'}]$.
   - Después del anterior movimiento, las sustituciones unitarias τ_j, $j < i$, tales que $x_i \in var(t_j)$, pueden aparecer ahora a la derecha de τ'_i. Entonces, movemos estas τ_j desde su lugar hacia el comienzo de la secuencia, hasta que $x_i \notin var(t_j)$; de esta forma x_i no aparecerá en ningún término posteriormente sustituido en la secuencia. En el ejemplo pondríamos $[u^{s''}/x^s]$ a la izquierda de $[y^{s'}/u^{s''}]$.

4. Volvemos a 1.

En el ejemplo, después de hacer este proceso, obtenemos la secuencia $[u^{s''}/x^s][y^{s'}/u^{s''}]$ $[a/y^{s'}]$. Como vemos, la condición de superbuena tipificación (Paso 2) se alcanza fácilmente, mientras que la de triangularidad puede perderse y debe ser recuperada reorganizando la secuencia (Paso 3). En esta subsección vamos a demostrar que esta reorganización siempre es posible.

Dada una secuencia de sustituciones unitarias $\tau = \tau_1 \dots \tau_n$, donde $\tau_i = [t_i/x_i^{s_i}]$, para todo $1 \leq i \leq n$, consideremos las dos propiedades siguientes:

[‡] Las secuencias superbién tipadas son triangulares, por definición. Sin embargo, en esta subsección, a veces estudiamos separadamente la condición de superbuena tipificación y la de triangularidad.

(i) $x_i \notin var(t_i)$, para cada $1 \leq i \leq n$

(ii) $x_i \neq x_j$, para cada $1 \leq i < j \leq n$

(i) y (ii) son condiciones requeridas por el concepto de triangularidad. Como muestra el ejemplo anterior, cuando tratamos con secuencias bien tipadas (y por tanto triangulares), estas dos propiedades (i) y (ii) son preservadas a lo largo del anterior proceso. Por ello, durante la comprobación de triangularidad, no las mencionaremos.

Definición 4.7.14 *Dada la secuencia $\tau = \tau_1 \ldots \tau_n$, donde $\tau_i = [t_i/x_i^{s_i}]$, para todo $1 \leq i \leq n$, el grafo dirigido G_τ asociado a τ está compuesto por:*

1. *El conjunto de vértices $V = \{1, \ldots, n\}$*[§]
2. *El conjunto de aristas $E = \{(i,j)/x_j \in var(t_i)\}$.*

En el ejemplo, para $\tau = [a/y^{s'}][u^{s''}/x^s][a/u^{s''}]$ tenemos $G_\tau = (\{1,2,3\}, \{(2,3)\})$.

El conjunto de aristas E del grafo G_τ representa la relación de precedencia entre las variables sustituidas por τ. En particular, si τ satisface la propiedad (i) entonces G_τ no tiene auto-ciclos.

Lema 4.7.15 *Si τ es una secuencia triangular entonces el grafo G_τ es acíclico.*

Demostración. Supongamos que $(i_1, i_2) \ldots (i_{n-1}, i_n)(i_n, i_1)$ es un ciclo de G_τ. Entonces:

$$(i_k, i_{k'}) \in E \Rightarrow x_{i_{k'}} \in var(t_{i_k}) \Rightarrow i_k < i_{k'} \text{ (por triangularidad de } \tau)$$

por lo que tendríamos $i_1 < i_2 < \ldots < i_n < i_1$ (contradicción). ∎

El movimiento de una sustitución unitaria a través de una secuencia, necesario para restaurar la triangularidad (anterior Paso 3), puede verse como una reordenación de las componentes unitarias, que corresponde a un camino específico en el grafo asociado a la secuencia.

Lema 4.7.16 *Dada una secuencia que satisface las propiedades (i) y (ii), si su grafo asociado es acíclico entonces cada recorrido topológico de sus vértices denota una secuencia triangular. Además, todas estas secuencias se comportan igual (como una única sustitución).*

Demostración. Sea $(i_1 \ldots i_n)$ un recorrido topológico de G_τ y $\tau' = \tau_{i_1} \ldots \tau_{i_n}$, la secuencia que representa. Es inmediato que τ' satisface las propiedades (i) y (ii) así que la triangularidad de τ' sólo depende de que $x_{i_k} \notin var(t_{i_{k'}})$, si $k < k'$. Veamos que esto es cierto: dados $1 \leq k < k' \leq n$, si $x_{i_k} \in var(t_{i_{k'}})$ entonces $(i_{k'}, i_k) \in E$ por tanto $k' < k$, puesto que el recorrido es topológico (contradicción).

Sea $(j_1 \ldots j_n)$ algún otro recorrido topológico de G_τ y $\tau'' = \tau_{j_1} \ldots \tau_{j_n}$, la secuencia que representa. Entonces comprobamos que $\tau'(x_i) = \tau''(x_i)$, para todo $1 \leq i \leq n$, comenzando

[§] Para todo $1 \leq i \leq n$, el i-ésimo vértice representa la sustitución unitaria τ_i.

por x_{j_n}. Supongamos que $i_k = j_n$, entonces $\tau'(x_{j_n}) = (\tau_{i_k} \dots \tau_{i_n})(x_{j_n})$ (por triangularidad) $= \tau_{i_k}(x_{j_n})$, puesto que $var(t_{i_k}) \cap \{x_{i_{k+1}}, \dots, x_{i_n}\} = \emptyset$. ya que el recorrido $(j_1 \dots j_n)$ es topológico. Por tanto $\tau'(x_{j_n}) = \tau_{i_k}(x_{j_n}) = \tau_{j_n}(x_{j_n}) = \tau''(x_{j_n})$. Ahora podemos repetir el mismo razonamiento con las secuencias $\tau_{j_1} \dots \tau_{j_{n-1}}$ y $\tau_{i_1} \dots \tau_{i_{k-1}} \tau_{i_{k+1}} \dots \tau_{i_n}$, puesto que también son triangulares y corresponden a recorridos topológicos sobre el grafo que resulta de eliminar el vértice j_n de G_τ. ■

Corolario 4.7.17 *Si τ es una secuencia triangular entonces cada recorrido topológico de G_τ representa una secuencia igual a τ.*

Demostración. Por el Lema 4.7.15, G_τ es acíclico. Por el Lema 4.7.16, cada recorrido topológico, en particular τ, representa la misma sustitución. ■

Aplicando estos resultados a $\tau = [a/y^{s'}][u^{s''}/x^s][a/u^{s''}]$, tenemos que las secuencias $[u^{s''}/x^s][a/y^{s'}][a/u^{s''}]$ y $[u^{s''}/x^s][a/u^{s''}][a/y^{s'}]$, correspondientes a los recorridos topológicos (213) y (231) de G_τ, respectivamente, son triangulares e iguales a τ.

Teorema 4.7.18 *Sea τ una secuencia bien tipada con respecto a $\mathcal{L}$, entonces existe una secuencia τ'' superbién tipada con respecto a $\mathcal{L}$ tal que τ'' se comporta como τ.*

Demostración. Supongamos que $\tau = \tau_1 \dots \tau_n$ y usemos inducción sobre n:

$n = 1$. Entonces τ está también superbién tipada con respecto a $\mathcal{L}$.

$n \to n+1$. Sea $\tau = \tau_1 \dots \tau_n[t/x^s]$. Entonces $\tau_1 \dots \tau_n$ está bien tipada con respecto a $\mathcal{L}$, por lo que, usando hipótesis de inducción, existe una secuencia $\tau'_1 \dots \tau'_n$ superbién tipada con respecto a $\mathcal{L}$ que se comporta como $\tau_1 \dots \tau_n$.

Puesto que τ está bien tipada, existe un término t' tal que $(t' \in s) \in \mathcal{L}$ y $t = t'\tau_1 \dots \tau_n = t'\tau'_1 \dots \tau'_n$ (por hipótesis de inducción). Consideremos la secuencia $\tau' = \tau'_1 \dots \tau'_n[t'/x^s]$. Obsérvese que esta secuencia puede no ser triangular, pero su grafo $G_{\tau'}$ es acíclico. En efecto, el grafo G asociado a $\tau'_1 \dots \tau'_n$ es acíclico, por el Lema 4.7.15, por lo que si $G_{\tau'}$ no es acíclico, entonces el vértice v correspondiente a la sustitución unitaria $[t'/x^s]$ produce un ciclo en $G_{\tau'}$. Sin pérdida de generalidad, representemos el ciclo como $(v, i_1) \dots (i_{k-1}, i_k)(i_k, v)$, $i_j \in \{1, \dots, n\}$, $1 \le j \le k$. Entonces, por definición del conjunto de aristas, tenemos $x^s \in var(t'\tau'_{i_1} \dots \tau'_{i_k})$, y entonces $x^s \in var(t'\tau'_1 \dots \tau'_n)$, puesto que $x^s \notin \{x_1, \dots, x_n\}$, por lo que $x^s \in var(t)$, en contra de la triangularidad de τ. En consecuencia $G_{\tau'}$ es acíclico.

Finalizamos usando el Lema 4.7.16. Sea $\tau'' = \tau'_{i_1} \dots \tau'_{i_{n+1}}$ un recorrido topológico cualquiera de $G_{\tau'}$, entonces se cumple:

(a) τ'' satisface las propiedades (i) y (ii):

(i) $x_i \notin var(t'_i)$, para todo $1 \le i \le n$, por triangularidad de $\tau'_1 \dots \tau'_n$; $x^s \notin var(t')$, si no $x^s \in var(t)$, en contra de la triangularidad de τ

(ii) $x_1, \dots, x_n, x^s$ son diferentes dos a dos, por la triangulardidad de τ

(b) τ'' es triangular, por el Lema 4.7.16

(c) τ'' está superbién tipada con respecto a $\mathcal{L}$, por la definición de t' y la superbuena tipificación de $\tau_1' \dots \tau_n'$ con respecto a $\mathcal{L}$

(d) $\tau'' = \tau$. En efecto, sea $y \in \{x_1, \dots, x_n, x^s\}$ entonces:

$$
\begin{aligned}
&(\tau'_{i_1} \dots \tau'_{i_{n+1}})(y)\\
&= (\tau'_{i_1} \dots \tau'_{i_{j-1}}[t'/x^s]\tau'_{i_{j+1}} \dots \tau'_{i_{n+1}})(y)\\
&= ((\tau'_{i_1} \dots \tau'_{i_{j-1}}\tau'_{i_{j+1}} \dots \tau'_{i_{n+1}})(y))[t'\tau'_{i_1} \dots \tau'_{i_{j-1}}\tau'_{i_{j+1}} \dots \tau'_{i_{n+1}}/x^s]\\
&\qquad \text{(por definición y triangularidad de } \tau'')\\
&= ((\tau_1' \dots \tau_n')(y))[t'\tau_1' \dots \tau_n'/x^s]\\
&\qquad \text{(por el Corolario 4.7.17 aplicado a } \tau_1' \dots \tau_n'\text{, utilizando que esta secuencia es}\\
&\qquad \text{ triangular y } \tau'_{i_1} \dots \tau'_{i_{j-1}}\tau'_{i_{j+1}} \dots \tau'_{i_{n+1}} \text{ es un recorrido topológico de } G)\\
&= ((\tau_1 \dots \tau_n)(y))[t/x^s]\\
&\qquad \text{(por hipótesis de inducción y definición de } t')\\
&= \tau(y) \quad \blacksquare
\end{aligned}
$$

Teorema 4.7.19 *Existe una secuencia superbién tipada con respecto a $\mathcal{L}$ que unifica Γ si y sólo si existe una secuencia bien tipada con respecto a $\mathcal{L}$ que unifica Γ.* ■

4.8 Unificación rígida tipada simultánea

Mientras no se diga lo contrario, $\mathcal{T}$ es un tableau finito con variables libres y ramas $B_1, \dots, B_m$.

La unificación rígida tipada puede ser introducida en un sistema de tableaux de dos formas diferentes. En una primera aproximación, podemos usar el cálculo $\mathcal{C}$ para cerrar una única rama; esta aproximación, que se sigue en [MGL 98], pero usando una variante del cálculo $\mathcal{C}$ no terminante, presenta una clara desventaja. El problema consiste en que la buena tipificación con respecto a una rama no equivale a la buena tipificación con respecto a todo el tableau, ya que las variables libres pueden aparecer distribuidas por diferentes ramas. De hecho, no todos los unificadores locales bien tipados (con respecto a la teoría incluida en la rama que cerramos) están bien tipados con respecto al tableau entero, por lo que necesitamos un *test adicional* para comprobar que el $\mathcal{C}$-unificador local obtenido es aplicable en (está bien tipado con respecto a) $\mathcal{T}$. Obsérvese que este *test* tiene éxito o falla una vez que el $\mathcal{C}$-unificador local ha sido totalmente construido.

En una segunda aproximación, podemos tratar de cerrar todo el tableau en un único paso, buscando un unificador simultáneo bien tipado. En estas condiciones, tratamos de unificar un conjunto de ecuaciones Γ compuesto por pares de literales potencialmente complementarios que aparecen en las ramas de $\mathcal{T}$. Un cálculo simultáneo evita la desventaja del cálculo local puesto que considera todas la ramas a la vez, por lo que incorpora implícitamente el anterior *test adicional* cada vez que extendemos la secuencia.

En este sentido, un cálculo simultáneo poda el espacio de búsqueda más que un cálculo local, ya que no prolonga secuencias erróneas que no van a llegar a estar bien tipadas con respecto a todo el tableau.

Dicho esto parece lógico estudiar los *problemas de Unificación Rígida Tipada Simultánea* (abreviadamente *URTS*):

Dado un tableau con variables libres $\mathcal{T}$ y un conjunto finito de ecuaciones Γ, ¿existe una secuencia bien tipada con respecto a $\mathcal{T}$ que unifique Γ?

Para resolver los *URTS* -problemas definimos el cálculo $\mathcal{D}$. Se trata de una extensión natural de $\mathcal{C}$, que se ocupa de todas las ramas de $\mathcal{T}$ cada vez que añadimos una nueva sustitución unitaria a la secuencia. Como para $\mathcal{C}$, además de las seis reglas estándar para unificación sintáctica, $\mathcal{D}$ cuenta con otras cuatro.

Las reglas tipadas de $\mathcal{D}$

(LW) Instanciación por la izquierda
$$\frac{x^s \simeq t',\ \Gamma \qquad \sigma_1 \ldots \sigma_n}{t \simeq t',\ \Gamma \qquad \sigma_1 \ldots \sigma_n[t/x^s]}$$
si $x^s \notin var(t)$ y $(x^s \in var(B_j\sigma_1 ... \sigma_n) \Rightarrow (t \in s) \in B_j\sigma_1 ... \sigma_n)$, para todo $1 \leq j \leq m$.

(RW) Instanciación por la derecha
$$\frac{y^{s'} \simeq x^s,\ \Gamma \qquad \sigma_1 \ldots \sigma_n}{y^{s'} \simeq t,\ \Gamma[t/x^s] \qquad \sigma_1 \ldots \sigma_n[t/x^s]}$$
si $x^s \notin var(t)$ y $(x^s \in var(B_j\sigma_1 ... \sigma_n) \Rightarrow (t \in s) \in B_j\sigma_1 ... \sigma_n)$, para todo $1 \leq j \leq m$.

(FWF) Fallo de instanciación funcional
$$\frac{x^s \simeq f(t_1, ..., t_n),\ \Gamma \quad \sigma_1 \ldots \sigma_n}{\textit{Fallo}}$$
si no existe un término t tal que $x^s \notin var(t)$ y $(x^s \in var(B_j\sigma_1 ... \sigma_n) \Rightarrow (t \in s) \in B_j\sigma_1 ... \sigma_n)$, para todo $1 \leq j \leq m$.

(VWF) Fallo de instanciación de variables
$$\frac{y^{s'} \simeq x^s,\ \Gamma \qquad \sigma_1 \ldots \sigma_n}{\textit{Fallo}}$$
si no existe un término t que satisfaga alguna de las siguientes condiciones:

- $x^s \notin var(t)$ y $(x^s \in var(B_j\sigma_1 ... \sigma_n) \Rightarrow (t \in s) \in B_j\sigma_1 ... \sigma_n)$, para todo $1 \leq j \leq m$.
- $y^{s'} \notin var(t)$ y $(y^{s'} \in var(B_j\sigma_1 ... \sigma_n) \Rightarrow (t \in s') \in B_j\sigma_1 ... \sigma_n)$, para todo $1 \leq j \leq m$.

Usamos el cálculo $\mathcal{D}$ de igual modo que $\mathcal{C}$, es decir, alternando pasos estándar y pasos tipados hasta que el conjunto de ecuaciones a unificar esté vacío.

Definición 4.8.1 *Sea Γ un conjunto de ecuaciones, $\sigma = \sigma_1\sigma_2 \ldots \sigma_n$ una secuencia de sustituciones unitarias y $\mathcal{T}$ un tableau con variables libres. Un $\mathcal{D}$-paso estándar consiste en la aplicación del algoritmo $\mathcal{A}$ sobre el par $\langle \Gamma, \sigma \rangle$ hasta obtener Fallo o alcanzar un conjunto de ecuaciones resuelto Γ'. Un $\mathcal{D}$-paso tipado consiste en la aplicación de una regla tipada sobre el par $\langle \Gamma, \sigma \rangle$, utilizando $\mathcal{T}$. Un $\mathcal{D}$-paso es un $\mathcal{D}$-paso estándar o un*

$\mathcal{D}$-paso tipado. Escribimos $\langle \Gamma, \sigma_1 \dots \sigma_n \rangle \vdash_\mathcal{D} \langle \Gamma', \sigma_1 \dots \sigma_n \sigma_{n'} \rangle$ ($n' \in \{n, n+1\}$) (resp. $\langle \Gamma, \sigma_1 \dots \sigma_n \rangle \vdash_\mathcal{D}$ Fallo) para expresar un $\mathcal{D}$-paso sin fallo (resp. con fallo).

Decimos que el cálculo $\mathcal{D}$ unifica un conjunto de ecuaciones Γ con respecto a un tableau $\mathcal{T}$ mediante la secuencia de sustituciones unitarias $\sigma_1 \dots \sigma_n$, o también que $\sigma_1 \dots \sigma_n$ es un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$, si existe una cadena de $\mathcal{D}$-pasos, alternando $\mathcal{D}$-pasos estándar y $\mathcal{D}$-pasos tipados, que comienza con $\langle \Gamma, \emptyset \rangle$ y finaliza con $\langle \emptyset, \sigma_1 \dots \sigma_n \rangle$.

4.8.1 Propiedades del cálculo $\mathcal{D}$

La noción de árbol de derivación para el cálculo $\mathcal{D}$ se define del mismo modo que para $\mathcal{C}$. Además, podemos comprobar que $\mathcal{D}$ también es terminante, por lo que podemos construir un algoritmo que busque el conjunto finito de todos los $\mathcal{D}$-unificadores.

Teorema 4.8.2 (Terminación) *El $\mathcal{D}$-árbol de derivación de todo URTS-problema es finito.*

Demostración. Similar al Teorema 4.7.6, pero utilizando en cada $\mathcal{D}$-paso la finitud del tableau en lugar de la finitud de la teoría. ■

El cálculo $\mathcal{D}$ sólo construye secuencias bien tipadas con respecto al tableau $\mathcal{T}$, que unifican el conjunto inicial de ecuaciones Γ. Por tanto, siempre que exista un $\mathcal{D}$-unificador, podemos responder que *sí* al correspondiente *URTS*-problema.

Teorema 4.8.3 (Corrección) *Sean $\Gamma, \mathcal{T}$ y σ un conjunto de ecuaciones, un tableau finito con variables libres y una secuencia de sustituciones unitarias, respectivamente. Si σ es un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$ entonces:*

1. *σ está bien tipada con respecto a $\mathcal{T}$.*
2. *σ unifica Γ.*

Demostración. Similar al Teorema 4.7.7, utilizando el tableau $\mathcal{T}$ en lugar de la teoría. ■

Como en la sección previa, sólo estamos interesados en elevar las secuencias que puedan ser inferidas a partir de un tableau básico cerrado. Con este fin, extendemos la noción de tipificación superbuena al caso de los tableaux.

Definición 4.8.4 *Una secuencia triangular de sustituciones $[t_1/x_1^{s_1}] \dots [t_n/x_n^{s_n}]$ está superbién tipada con respecto a un tableau con variables libres $\mathcal{T}$, si $x_i^{s_i} \in var(B) \Longrightarrow (t_i \in s_i) \in B$, para todo $1 \leq i \leq n$ y toda rama B.*

Para comprobar la completitud de $\mathcal{D}$ necesitamos resultados similares a los Lemas 4.7.9, 4.7.10 y 4.7.11, pero para tableaux.

Lema 4.8.5 *Sea $\sigma_1 \dots \sigma_n$ una secuencia tal que $\sigma_i = [t_i/x_i^{s_i}]$, para todo $1 \leq i \leq n$. Para un índice fijo $m \in \{1, \dots, n\}$ definimos $\sigma_i' = [t_i[t_m/x_m^{s_m}]/x_i^{s_i}]$, para todo $1 \leq i \leq m-1$. Si $\sigma_1 \dots \sigma_n$ está superbién tipada con respecto a un tableau con variables libres $\mathcal{T}$ entonces:*

1. $\sigma'_1 \ldots \sigma'_{m-1}\sigma_{m+1} \ldots \sigma_n$ *está superbién tipada con respecto a* $\mathcal{T}\sigma_m$.

2. $\sigma_m\sigma'_1 \ldots \sigma'_{m-1}\sigma_{m+1} \ldots \sigma_n = \sigma_1 \ldots \sigma_n$.

Demostración. (1) La triangularidad se prueba como en el Lema 4.7.9. Para la superbuena tipificación, sea $B\sigma_m$ una rama de $\mathcal{T}\sigma_m$ y $x_i^{s_i} \in var(B\sigma_m)$. Distinguimos casos:

- Si $i \leq m-1$ entonces $x_i^{s_i} \in var(B)$ pues, de lo contrario, $x_i^{s_i}$ habría sido introducida por t_m, lo cual es imposible por triangularidad. En estas condiciones, $(t_i \in s_i) \in B$ por superbuena tipificación, por lo que $(t_i\sigma_m \in s_i) \in B\sigma_m$.

- Si $i \geq m+1$ entonces $x_i^{s_i} \in var(B)$ pues, de lo contrario, $x_i^{s_i}$ habría sido introducida por t_m, pero entonces $x_m^{s_m} \in var(B)$ y $var(t_m) \not\subseteq var(B)$, lo cual es imposible por la hipótesis de superbuena tipificación. Por tanto, $(t_i \in s_i) \in B$ y, entonces, $(t_i\sigma_m \in s_i) \in B\sigma_m$, que es como decir $(t_i \in s_i) \in B\sigma_m$, por triangularidad.

(2) Inmediato por la definición de $\sigma'_1 \ldots \sigma'_{m-1}$. ■

Lema 4.8.6 (Completitud tipada) *Sean* Γ *y* $\mathcal{T}$ *un conjunto de ecuaciones no vacío en forma resuelta y un tableau con variables libres, respectivamente. Sea* $\tau = \tau_1 \ldots \tau_n$ *una secuencia superbién tipada con respecto a* $\mathcal{T}$ *que unifica* Γ. *Entonces existe un conjunto de ecuaciones* Γ' *y una sustitución unitaria* σ *tal que* $\langle \Gamma, \emptyset \rangle \vdash_{\mathcal{D}} \langle \Gamma', \sigma \rangle$. *Además existe otra secuencia* $\theta_1 \ldots \theta_k$ *superbién tipada con respecto a* $\mathcal{T}\sigma$ *que unifica* Γ'.

Demostración. Como en el Lema 4.7.10, dada $x^s \simeq t' \in \Gamma$ distinguimos dos casos:

(i) $t' = f(\ldots)$. Puesto que τ está superbién tipada con respecto a $\mathcal{T}$ y unifica Γ, existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$, $x^s \notin var(t)$ y $(t \in s) \in B$, para cada rama B donde x^s aparece libre. Aplicando *LW* obtenemos $\Gamma' = (\Gamma \backslash \{x^s \simeq t'\}) \cup \{t \simeq t'\}$ y $\sigma = [t/x^s]$. Extrayendo τ_i de τ, como en el Lema 4.8.5, obtenemos una secuencia $\theta_1 \ldots \theta_k$ superbién tipada con respecto a $\mathcal{T}[t/x^s]$ que verifica que $[t/x^s]\theta_1 \ldots \theta_k$ y τ se comportan igual, por lo que $\theta_1 \ldots \theta_k$ unifica Γ'.

(ii) $t' = y^{s'}$. Puesto que τ unifica Γ, si existe $1 \leq i \leq n$ tal que $\tau_i = [t/y^{s'}]$ entonces $y^{s'} \notin var(t)$ y $(t \in s') \in B$, para cada rama B donde $y^{s'}$ aparece libre. Aplicando *RW* obtenemos $\Gamma' = (\Gamma[t/y^{s'}] \backslash \{x^s \simeq y^{s'}\}) \cup \{x^s \simeq t\}$ y $\sigma = [t/y^{s'}]$. Extrayendo τ_i de τ, como en el Lema 4.8.5, procedemos como en el caso anterior. Si existiera $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$, aplicaríamos *LW* como en (i). ■

Lema 4.8.7 (Fallo tipado) *Sean* Γ *y* $\mathcal{T}$ *un conjunto de ecuaciones resuelto y un tableau con variables libres, respectivamente. Si* $\langle \Gamma, \emptyset \rangle \vdash_{\mathcal{D}}$ *Fallo después de un* $\mathcal{D}$*-paso tipado entonces no existe una secuencia superbién tipada con respecto a* $\mathcal{T}$ *que unifique* Γ.

Demostración. Sea τ una secuencia $\tau_1 \ldots \tau_n$ superbién tipada con respecto a $\mathcal{T}$ que unifica Γ. Distinguimos casos de acuerdo con la regla que ha producido el fallo:

(i) Hemos aplicado la regla *FWF* a $x^s \simeq f(t_1 \ldots t_n) \in \Gamma$. Puesto que τ unifica Γ, existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$. Por la superbuena tipificación de τ, $(t \in s) \in B$, para cada rama B donde x^s aparece libre, y $x^s \notin var(t)$, por triangularidad; entonces obtenemos contradicción ya que podemos aplicar *LW*.

(ii) Hemos aplicado la regla *VWF* a $x^s \simeq y^{s'} \in \Gamma$. Puesto que τ unifica Γ, si existe $1 \leq i \leq n$ tal que $\tau_i = [t/x^s]$ entonces puede aplicarse *LW*, razonando como en el caso anterior; si no fuera así, aplicaríamos *RW*. ■

A continuación establecemos el teorema de completitud para el cálculo $\mathcal{D}$.

Teorema 4.8.8 (Completitud) *Sean Γ y $\mathcal{T}$ un conjunto de ecuaciones y un tableau con variables libres, respectivamente. Sea $\sigma_1 \ldots \sigma_n$ una secuencia superbién tipada con respecto a $\mathcal{T}$ que unifica Γ. Entonces existe un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$.*

Demostración. Si cada hoja del correspondiente $\mathcal{D}$-árbol de derivación es un nodo de *Fallo* entonces $\Gamma \neq \emptyset$. Sea h una hoja cualquiera. Iterando el Lema 4.8.6, existe un nodo interno $\langle \Gamma', \tau_1 \ldots \tau_m \rangle$ y una secuencia superbién tipada $\theta_1 \ldots \theta_l$ con respecto a $\mathcal{T}\tau_1 \ldots \tau_m$ que unifica Γ'. Además este nodo es padre de h, por lo tanto se cumple alguno de los siguientes casos:

(i) Un $\mathcal{D}$-paso estándar produce *Fallo*: imposible puesto que $\theta_1 \ldots \theta_l$ unifica Γ'

(ii) Un $\mathcal{D}$-paso tipado produce *Fallo*: por el Lema 4.8.7, no existe ninguna secuencia superbién tipada con respecto a $\mathcal{T}\tau_1 \ldots \tau_m$ que unifique Γ', en contradicción con la existencia de $\theta_1 \ldots \theta_l$. ■

Obsérvese que no podemos resolver un *URTS*-problema examinando su $\mathcal{D}$-árbol de derivación asociado (en consonancia con el Corolario 4.7.13), ya que no se verifica un resultado similar al Teorema 4.7.19 para el caso simultáneo. El siguiente ejemplo muestra este hecho.

Ejemplo 4.8.9 *Sea $\mathcal{T}$ el sketch de tableau con variables libres que aparece en la Figura 4.7.*

La secuencia $\tau = [a/z^{s'}][a/x^s]$ está bien tipada con respecto a $\mathcal{T}$ y unifica $\Gamma = \{a \simeq x^s\}$, por lo que $[a/z^{s'}][a/x^s]$ es una solución del URTS-problema relacionado y $\mathcal{T}$ puede cerrarse. Sin embargo, no corresponde a ningún tableau básico cerrado; de hecho, no existe ninguna secuencia superbién tipada con respecto a $\mathcal{T}$ que unifique Γ, ni un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$. Esto se debe a que x^s debe instanciarse por la constante a en la primera rama, mientras que tiene que hacerlo por $z^{s'}$ en la segunda.

El anterior ejemplo tiene dos consecuencias. Por una parte, el cálculo $\mathcal{D}$ no resuelve completamente el *URTS*-problema, aunque la completitud del sistema de tableaux no se ve afectada. Por otra parte, la decidibilidad del *URTS*-problema queda como problema abierto.

$$
\begin{array}{c}
\neg P(a) \\
\forall x^s(\ (a \in s \wedge P(x^s))\ \vee\ (\forall z^{s'}(z^{s'} \in s) \wedge a \in s' \wedge P(x^s))\ )
\end{array}
$$

$$
\begin{array}{ccc}
a \in s & \qquad & a \in s' \\
| & & | \\
P(x^s) & & P(x^s) \\
 & & | \\
 & & \forall z^{s'} z^{s'} \in s \\
 & & | \\
 & & z^{s'} \in s
\end{array}
$$

Figura 4.7: El sketch de tableau $\mathcal{T}$.

4.9 Tableaux con variables libres con unificación rígida tipada simultánea

A continuación integramos el cálculo $\mathcal{D}$ dentro de los tableaux con variables libres. Sea $\mathcal{S}2$ el sistema de tableaux compuesto por las reglas $\alpha, \beta, \gamma', \delta'$ y la nueva regla de cierre:

Definición 4.9.1 (Regla de cierre *URTS***)** *Un tableau con variables libres $\mathcal{T}$ y ramas $B_1, \ldots, B_m$ está cerrado si existe un conjunto de ecuaciones $\Gamma = \{L_1 \simeq L'_1, \ldots, L_m \simeq L'_m\}$, donde $L_i \simeq L'_i$ se obtiene a partir de un par de literales potencialmente complementarios que aparecen en B_i, y un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$.*

Usamos el sistema $\mathcal{S}2$ para construir tableaux cerrados como sigue:

1. Se expande de forma indeterminista el tableau, usando las reglas $\alpha, \beta, \gamma', \delta'$.

2. Se define un multiconjunto de ecuaciones Γ, seleccionando un par de literales potencialmente complementarios de cada rama del tableau actual. Se construye el $\mathcal{D}$-árbol finito de derivación para Γ con respecto al tableau actual. Si existe un $\mathcal{D}$-unificador entonces el tableau está cerrado, usando la regla de cierre *URTS*; si no, se intenta con otro multiconjunto de ecuaciones, o se vuelve a 1.

Obsérvese que el único paso que tiene en cuenta los géneros (paso 2) siempre finaliza -puede verse como un procedimiento de decisión. Por esta razón, hemos separado la complejidad de los géneros de la indecidibilidad de la lógica de primer orden.

Teorema 4.9.2 (Corrección de $\mathcal{S}2$) *Dado un conjunto de Σ-sentencias Φ, si Φ tiene un tableau con variables libres cerrado entonces Φ es insatisfactible en estructuras con géneros no vacíos.*

$$
\begin{array}{ccc}
 & 5: \forall x^s (x^s \in s') & \\
 & \mid \quad \gamma \text{ a } 5 & \\
 & 6: x^s \in s' & \\
 & \mid \quad \gamma \text{ a } 3 & \\
 & 7: z^s \in s' \to \forall y^s (P(z^s, y^s)) & \\
 & \swarrow \quad \searrow \quad \beta \text{ a } 7 & \\
8: \neg z^s \in s' & & 9: \forall y^s (P(z^s, y^s)) \\
 & & \mid \quad \gamma \text{ a } 4 \\
 & & 10: f(y^s) \in s \\
 & & \mid \quad \gamma \text{ a } 9 \\
 & & 11: P(z^s, u^s)
\end{array}
$$

Figura 4.8: El tableau con variables libres $\mathcal{T}$.

Demostración. $\mathcal{S}2$ es correcto porque es un caso particular de $\mathcal{S}1$. En efecto, por el Teorema 4.8.3, siempre que apliquemos la regla de cierre *URTS* en un tableau $\mathcal{T}$, estamos usando una secuencia bien tipada con respecto a $\mathcal{T}$ y, por tanto, un caso particular de la regla *sub* de $\mathcal{S}1$. ∎

Teorema 4.9.3 (Completitud de $\mathcal{S}2$) *Dado un conjunto de Σ-sentencias Φ, si Φ es insatisfactible entonces Φ tiene un tableau con variables libres cerrado.*

Demostración. Puesto que Φ no es satisfactible, existe un tableau básico cerrado $\mathcal{T}_b$ para Φ. Podemos construir sistemáticamente un tableau con variables libres $\mathcal{T}$ y una secuencia $\sigma_1 \ldots \sigma_n$ superbién tipada con respecto a $\mathcal{T}$ tal que $\mathcal{T}\sigma_1 \ldots \sigma_n = \mathcal{T}_b$, como sigue: cada vez que aplicamos una γ-expansión en $\mathcal{T}_b$, añadimos la correspondiente instancia como una sustitución unitaria al comienzo de la secuencia actual; esta secuencia estará superbién tipada, por construcción. Además, existe un multiconjunto de ecuaciones Γ, compuesto por pares de literales potencialmente complementarios que aparecen en las ramas de $\mathcal{T}$, tal que $\sigma_1 \ldots \sigma_n$ unifica Γ. Por el Teorema 4.8.8, existe un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$ y, por tanto, podemos cerrar $\mathcal{T}$ aplicando la regla de cierre *URTS*. ∎

Ejemplo 4.9.4 *Usamos el sistema $\mathcal{S}2$ para resolver el problema del Ejemplo 4.3.2. Primero aplicamos las reglas γ y β para construir el tableau con variables libres $\mathcal{T}$ de la Figura 4.8.*

En segundo lugar usamos el cálculo $\mathcal{D}$ para unificar el conjunto de ecuaciones $\Gamma = \{z^s \simeq x^s, z^s \simeq a, u^s \simeq f(a)\}$ con respecto a $\mathcal{T}$. Obsérvese que debe existir un $\mathcal{D}$-unificador, puesto que Γ es unificado por la secuencia superbién tipada $[f(y^s)/u^s][a/y^s][a/z^s][a/x^s]$ (esta es la secuencia que relaciona $\mathcal{T}$ con el tableau básico del Ejemplo 4.5.7 tal como se

$$
\begin{array}{lll}
 & \{z^s \simeq x^s, z^s \simeq a, u^s \simeq f(a)\} & \\
\hline
\mathrm{LW} & \{x^s \simeq a, z^s \simeq a, u^s \simeq f(a)\} & [a/x^s] \\
\cline{2-2}
 & \{a \simeq a, z^s \simeq a, u^s \simeq f(a)\} & \\
\hline
\mathrm{LW} & \{z^s \simeq a, u^s \simeq f(a)\} & [a/x^s][a/z^s] \\
\cline{2-2}
 & \{a \simeq a, u^s \simeq f(a)\} & \\
\hline
\mathrm{LW} & \{u^s \simeq f(a)\} & [a/x^s][a/z^s][f(y^s)/u^s] \\
\cline{2-2}
 & \{f(y^s) \simeq f(a)\} & \\
\hline
\mathrm{LW} & \{y^s \simeq a\} & [a/x^s][a/z^s][f(y^s)/u^s][a/y^s] \\
\cline{2-2}
 & \{a \simeq a\} & \\
\hline
 & \emptyset &
\end{array}
$$

Figura 4.9: Cómputo de un $\mathcal{D}$-unificador.

explica en la demostración del Teorema 4.9.3). En la Figura 4.9 presentamos el cómputo de un $\mathcal{D}$-unificador de Γ con respecto a $\mathcal{T}$.

Comparemos el cálculo simultáneo $\mathcal{D}$ con respecto a una aproximación local (consúltense los comentarios hechos al comienzo de la sección anterior) que consiste en a) aplicar el cálculo local $\mathcal{C}$ a cada rama, independientemente, y b) realizar un test para comprobar si el $\mathcal{C}$-unificador obtenido está bién tipado con respecto a todo el tableau $\mathcal{T}$. Esto nos lleva a resolver dos problemas:

1. *$\{z^s \simeq x^s\}$ con respecto a la teoría $\{a \in s, x^s \in s'\}$.*
2. *$\{z^s \simeq a, u^s \simeq f(a)\}$ con respecto a la teoría $\{a \in s, x^s \in s', f(y^s) \in s\}$.*

En el segundo problema, podemos aplicar la $\mathcal{C}$-regla LW, utilizando la declaración $f(y^s) \in s$, y obtener la sustitución unitaria $\sigma = [f(y^s)/z^s]$. Sin embargo, cualquier secuencia que prolongue σ no estará bién tipada con respecto a $\mathcal{T}$ (¡el test fallará, pero sólo cuando el $\mathcal{C}$-unificador local haya sido construido totalmente!) y, por eso, la exploración del $\mathcal{C}$-subárbol de derivación que extiende este prefijo resulta inútil. En este sentido, el cálculo $\mathcal{D}$ es más eficiente, ya que impide la prolongación de secuencias erróneas que no van a estar bién tipadas con respecto a todo el tableau.

4.10 Conclusiones y trabajos relacionados

Hemos presentado la lógica con declaraciones de términos LDT. Esta es una lógica heterogénea que extiende la lógica de primer orden mediante la introducción de un nuevo constructor de fórmulas $t \in s$, que permite la declaración dinámica del término t como elemento de género s. Las lógicas con declaraciones de términos ya aparecen en [Fris 91], [Weid 91] y [Weid 95]. Allí, las variables pueden restringirse a otros géneros no unitarios; por ejemplo, $x^{s \cap s'}$ denota un individuo del género correspondiente a la intersección $s \cap s'$. En LDT, esta forma de tipificación de variables puede expresarse incluyendo la declaración $x^s \in s'$ donde se requiera.

Aparte de los artículos [GLMN 97], [MGL 98], [MG 99], [MGL 00], sólo [Weid 95] usa métodos de deducción basados en tableaux. [Fris 91] y [Weid 91] presentan métodos basados en resolución, el primero en un marco más general. En estos últimos, las variables tipadas se comportan de forma universal en el proceso de unificación usado, en contraste con la aproximación rígida necesaria para los tableaux.

A la hora de diseñar sistemas de tableaux con variables libres para *LDT*, la primera cuestión a resolver es cómo definir las sustituciones bien tipadas. Este concepto es la clave para permitir una integración adecuada de cualquier cálculo de unificación tipada en un sistema de tableaux. En [MGL 00] demostramos que otros posibles intentos de definir la regla de substitutividad (por ejemplo, el que aparece en [Weid 95]) son erróneos. En este sentido, los resultados de decidibilidad sobre unificación rígida tipada presentados en [Weid 95] son inservibles para los tableaux, ya que su cálculo es correcto y completo con respecto a una definición *no correcta* de buena tipificación; es decir, la aplicación de los unificadores que su cálculo obtiene puede producir deducciones incorrectas. En estas condiciones, es ahora cuando conocemos resultados válidos sobre la decidibilidad de la unificación tipada para tableaux con declaraciones de términos.

Los resultados que aquí aparecen son el fruto de una línea de investigación sobre la construcción de métodos de tableaux para *LDT* que se ha desarrollado como sigue:

1. En [GLMN 97] presentamos un método de tableaux básico para una extensión de *LDT* que consideraba preórdenes (anti)monótonos. Además, las declaraciones de (anti)monotonía se expresaban dinámicamente mediante un constructor de fórmulas nuevo.

2. En [MGL 98] presentamos un cálculo de unificación local que requería un test adicional para comprobar la buena tipificación con respecto a todo el tableau. Además dicho cálulo no era terminante, por lo que no podía ser integrado satisfactoriamente en el sistema de tableaux. Obsérvese que cualquier cálculo de unificación no terminante resulta poco útil dentro del marco de los métodos de tableaux, ya que puede no acabar nunca cuando intenta resolver un problema no unificable. El cálculo $\mathcal{D}$ también mejora el presentado en [MGL 98] en otros aspectos, ya menores. Tiene menos reglas y las condiciones de aplicabilidad son más simples. Debido a la terminación, la técnica usada para demostrar la completitud de $\mathcal{D}$ es diferente y simplifica considerablemente la demostración extremadamente técnica del cálculo presentado en [MGL 98] -allí usábamos órdenes bien fundamentados para probar que el proceso de elevación terminaba. Ahora establecemos fácilmente la completitud, demostrando que la existencia de soluciones superbién tipadas puede preservarse durante el proceso de $\mathcal{D}$-unificación.

3. [MGL 00] es una extensión de [MGL 98] en la que se presentan los contraejemplos correspondientes a las otras posibles formas de definir el concepto de sustitución bien tipada. Además allí estudiamos el impacto que la ramas ya cerradas en un tableau tenían durante el cierre de una rama abierta. Así, demostramos que el test adicional sólo necesita considerar las ramas que están abiertas todavía. En

este sentido, definimos sistemas de tableaux más eficientes que los presentados en [MGL 98].

4. En [MG 99] definimos el cálculo de unificación simultáneo $\mathcal{D}$ que aquí aparece. Como ya dijimos, $\mathcal{D}$ incorpora implícitamente el test adicional cada vez que prolongamos una secuencia. En este sentido, hemos mostrado que el cálculo $\mathcal{D}$ poda de forma más eficiente el espacio de búsqueda que una aproximación local como la de [MGL 98]. Además $\mathcal{D}$ es terminante, por lo que hemos separado la complejidad de los géneros de la indecidibilidad de la lógica de primer orden. En este sentido, $\mathcal{D}$ puede usarse como procedimiento de decisión, ya que es suficiente explorar un espacio de búsqueda finito para resolverlo.

Capítulo 5

Preórdenes monótonos

En este capítulo, nos olvidamos de los géneros y estudiamos técnicas especializadas en el manejo de preórdenes monótonos. Para ello, integramos cálculos de unificación rígida en los métodos de tableaux con variables libres. El diseño de tales cálculos de unificación procede por etapas. Primero no consideramos monotonía y definimos un cálculo correcto, completo y terminante, que luego mejoramos utilizando técnicas de reescritura. Después presentamos un cálculo correcto y completo, pero no terminante, que resuelve problemas de unificación rígida en preórdenes monótonos. Este último cálculo representa una base sobre la que aplicar técnicas de reescritura y llegar a alcanzar la terminación.

5.1 Introducción

El uso de la igualdad parece ser una técnica esencial a la hora de diseñar demostradores automáticos que razonen sobre dominios extraidos del mundo real. Es de sobra conocido que la incorporación de la igualdad incrementa considerablemente el poder expresivo de la lógica. No obstante, también sabemos que su integración implica un crecimiento sustancial del espacio de búsqueda, por lo que el estudio de métodos eficientes para el manejo de la igualdad se considera una importante línea de investigación en diferentes áreas de las ciencias de la computación.

Recientemente algunas investigaciones han señalado la necesidad de extender el estudio de la igualdad al de otras relaciones transitivas (no simétricas). Este es el caso, por ejemplo, de CLP [JM 94], donde la resolución de restricciones se añade a la programación lógica. Las teorías preordenadas también se utilizan en el área de las álgebras de procesos [Inv 94] como una herramienta útil a la hora de establecer relaciones entre las distintas descripciones del sistema, y en concreto se han usado para especificaciones parciales [Walk 90] y divergencias entre modelos [AH 92].

En el campo de la demostración automática, esta situación ya ha propiciado el desarrollo de demostradores con reglas específicas para la relación que se estudia. Esta línea se sigue por ejemplo en [BKS 85] y [Hines 92] y, más recientemente, en [LA 93], [Levy 94], [BG 94], [BG 95] y [LA 96] aplicando técnicas de reescritura que evitan el cálculo del cierre transitivo de la teoría. Estas y otras aproximaciones están basadas en la resolución como método de inferencia. Si integramos relaciones transitivas en los tableaux por medio de un cálculo de unificación (simultáneo), en lugar de introducir nuevas reglas de inferencia, los sistemas de tableaux parecen ser una mejor herramienta que la resolución por, al menos, dos razones: por una parte, podemos construir procedimientos de decisión que separen la dificultad de las relaciones transitivas, de la lógica de primer orden y, por otra parte, las variables libres funcionan rígidamente, por lo que el encadenamiento en posiciones de variable no es un problema crítico.

Este capítulo está estructurado en dos partes. En la primera, incorporamos técnicas de reescritura a preórdenes (relaciones transitivas y reflexivas) y definimos cálculos de unificación para los tableaux con variables libres. La Sección 4 presenta un cálculo de unificación para preórdenes, que extendemos a una versión simultánea, e integramos en los tableaux, en la Sección 5. La completitud de ambos sistemas se basa en un método de tableaux básico que presentamos en la Sección 3. Aunque estos cálculos ya son terminantes, los mejoramos usando técnicas de reescritura en la Sección 6.

En la segunda parte, extendemos los preórdenes a preórdenes monótonos. Todos los problemas que se plantean cuando extendemos las relaciones de equivalencia al caso de las equivalencias congruentes, también aparecen para los preórdenes monótonos. No obstante, surge un nuevo problema cuando permitimos el uso de variables no lineales porque, en este caso, debemos adivinar ciertos contextos. Por ejemplo, $f(g(b), x, x) \leq f(y, y, g(a))$ es unificable con respecto a la teoría $\{a \leq b\}$, si utilizamos la sustitución $\sigma = [g(a)/x, g(b)/y]$; sin embargo no podemos utilizar unificación sintáctica, ni encadenamientos en posiciones de subtérmino, para resolver el problema. En efecto, dejando aparte cualquier orden de reducción, las únicas posibilidades (instanciando x a a o y a b) no construyen el contexto apropiado. En una primera aproximación, podemos usar los axiomas de reflexividad funcional para solucionar el problema de forma ingenua, como hicimos en el Capítulo 3 mediante la regla *Reff*; esto es, podemos extender una rama con la fórmula $f(x_1, \ldots, x_n) \sqsubseteq f(x_1, \ldots, x_n)$, donde f es un símbolo de función de aridad n y las variables x_i son nuevas en el tableau. Sin embargo, esta regla es muy ineficiente debido a que el símbolo de función f debe adivinarse, por lo que necesitamos diseñar reglas especiales para el tratamiento de la monotonía. La Sección 7 presenta un cálculo de unificación completo y correcto que evita el uso de los axiomas de reflexividad funcional. Este cálculo se extiende a una versión simultánea y se integra en los tableaux, en la Sección 8.

5.2 Presentación de la lógica LPM

La Lógica con Preórdenes Monótonos (*LPM* en adelante) extiende la lógica de primer orden mediante el constructor de fórmulas $\sqsubseteq$. Una *inecuación* es una fórmula atómica $s \sqsubseteq t$, donde $s, t \in T(\Sigma, X)$. El conjunto $F(\Sigma, X)$ de Σ-fórmulas de *LPM* se construye a partir de inecuaciones, utilizando $\neg, \wedge$ y $\exists$ como en *LPO*. Llamamos desinecuación a la negación de una inecuación y la representamos por $s \not\sqsubseteq t$. Para centrarnos en el estudio de los preórdenes (monótonos), $\sqsubseteq$ es el único símbolo de predicado disponible en las signaturas de *LPM*. En lógica de primer orden con igualdad $\simeq$, los símbolos de predicado P pueden representarse introduciendo un género *bool*, una constante *true* de género *bool*, y reemplazando $P(\bar{t})$ por $true \simeq P(\bar{t})$ (para más detalles, véase por ejemplo [BGLS 95]). Obsérvese que podemos usar la misma idea en *LPM*, reemplazando $P(\bar{t})$ por $true \sqsubseteq P(\bar{t}) \wedge P(\bar{t}) \sqsubseteq true$.

Definición 5.2.1 *El conjunto $T(\Sigma, X)$ de Σ-términos se define mediante las siguientes reglas de formación:*

$$t ::= x(\in X) \mid c(\in \mathcal{C}) \mid f(t_1, \ldots, t_n) \ (f \in \mathcal{F} \text{ de aridad } n;\ t_i \in T(\Sigma, X))$$

El conjunto $F(\Sigma, X)$ de Σ-fórmulas se define mediante las siguientes reglas de formación:

$$\varphi ::= s \sqsubseteq t \ (s, t \in T(\Sigma, X)) \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \exists x \varphi$$

Una inecuación (resp. desinecuación) es una fórmula del tipo $s \sqsubseteq t$ (resp. $s \not\sqsubseteq t$), donde $s, t \in T(\Sigma, X)$.

Una $\sqsubseteq$-*teoría*, o simplemente una *teoría*, consiste en un conjunto finito de inecuaciones. Las representaremos mediante las letras $\mathcal{I}$, $\mathcal{I}'$. Obsérvese de nuevo que nuestro concepto de teoría difiere del habitualmente usado.

Las estructuras para *LPM* son dominios preordenados, según la Definición 3.2.1, en los que interpretamos los símbolos de función.

Definición 5.2.2 *Una Σ-estructura $\mathcal{D}$ en LPM está compuesta por el preorden $\langle D, \sqsubseteq^{\mathcal{D}} \rangle$ e interpretaciones $f^{\mathcal{D}}$ y $c^{\mathcal{D}}$, para cada símbolo de constante c y de función f de Σ. Una Σ-estructura es monótona si las interpretaciones de los símbolos de función son monótonas en todos sus argumentos, es decir, si $f^{\mathcal{D}}(\ldots d \ldots) \sqsubseteq^{\mathcal{D}} f^{\mathcal{D}}(\ldots d' \ldots)$, siempre que $d \sqsubseteq^{\mathcal{D}} d'$.*

Una $\mathcal{D}$-valoración es una aplicación $\rho : X \to D$. Una Σ-interpretación (respectivamente, monótona) es un par $\langle \mathcal{D}, \rho \rangle$, donde $\mathcal{D}$ es una Σ-estructura (respectivamente, monótona) y ρ es una $\mathcal{D}$-valoración.

Los términos y fórmulas de *LPM* se interpretan en una Σ-interpretación (monótona) como en *LPO*, pero utilizando $\sqsubseteq^{\mathcal{D}}$ como semántica asociada a $\sqsubseteq$. Por ejemplo

$$[\![s \sqsubseteq t]\!]^{\mathcal{D}}_{\rho} = \begin{cases} \underline{t} & \text{si } [\![s]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}} [\![t]\!]^{\mathcal{D}}_{\rho} \\ \underline{f} & \text{en otro caso.} \end{cases}$$

Las nociones de satisfactibilidad y modelo coinciden con las usadas en *LPO*. En cuanto a la consecuencia lógica, debemos tener en cuenta que las variables libres funcionan rígidamente en los tableaux, lo que significa que una variable de un tableau sólo puede ser sustituida una vez y, por tanto, representa un único elemento. En este sentido, las deducciones efectuadas a partir de una teoría inecuacional, que son las involucradas en el cierre de una rama, no son deducciones universales, sino rígidas. Semánticamente la rigidez se puede expresar como sigue.

Definición 5.2.3 *Dada una teoría $\mathcal{I} = \{s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n\}$ y una inecuación $s \sqsubseteq t$, con $\mathcal{I} \models_p s \sqsubseteq t$ (respectivamente, $\mathcal{I} \models_m s \sqsubseteq t$) expresamos que la fórmula $(s_1 \sqsubseteq t_1 \wedge \ldots \wedge s_n \sqsubseteq t_n) \to s \sqsubseteq t$ es cierta en toda Σ-interpretación (respectivamente, monótona).*

Cuando introduzcamos las técnicas de reescritura, usaremos órdenes de reducción $\succ$, esto es, órdenes parciales definidos sobre $T(\Sigma, X)$, que están bien fundamentados y que son monótonos y estables bajo sustituciones (es decir, $s \succ t$ implica $s\theta \succ t\theta$, para todas las sustituciones θ). Suponemos, de ahora en adelante, que estos órdenes de reducción son totales sobre los términos básicos.

Para presentar los cálculos de unificación, introducimos la notación de *restricción* [NR 95], [DV 98], que captura la unificación sintáctica y el orden entre términos, en un único marco.

Definición 5.2.4 *Una restricción C es un conjunto de restricciones de igualdad $s \simeq t$ y restricciones de orden $s \succ t$. Una sustitución θ es solución de una restricción de igualdad $s \simeq t$ (respectivamente, de una restricción de orden $s \succ t$) si θ es básica para $var(s) \cup var(t)$ y $s\theta$ coincide con $t\theta$ (respectivamente, si $s\theta \succ t\theta$). Una sustitución θ es solución de, o resuelve, una restricción C si θ es solución de cada restricción, de igualdad u orden, de C. Una restricción C es satisfactible si tiene una solución.*

En este capítulo, asumimos la existencia de procedimientos efectivos para comprobar la satisfactibilidad de una restricción, lo cual depende del orden escogido. Por ejemplo, en [NR 95] y [Nieu 93] se pueden encontrar métodos efectivos para órdenes lexicográficos de caminos.

5.3 Tableaux básicos para LPM

En esta sección presentamos dos restricciones sucesivas del método de los tableaux básicos de *LPGD*, primero considerando preórdenes y luego considerando preórdenes monótonos. Procedemos por etapas en cuanto al manejo de la monotonía; es decir, estos sistemas de tableaux básicos serán correctos y completos con respecto a la noción de satisfactibilidad en Σ-estructuras y en Σ-estructuras monótonas, respectivamente.

Como es habitual, suponemos una signatura fija Σ que extendemos a otra $\overline{\Sigma}$ mediante un conjunto numerable de constantes nuevas AC.

Cuando no consideramos monotonía, además de las habituales reglas de expansión α, β, γ y δ, contamos con las dos reglas siguientes:

$$\textit{Ref)}\quad \frac{}{t \sqsubseteq t} \qquad\qquad \textit{Tran)}\quad \frac{\begin{array}{c} t_1 \sqsubseteq t_2 \\ t_2 \sqsubseteq t_3 \end{array}}{t_1 \sqsubseteq t_3}$$

Como en *LPGD*, las ramas cerradas son las que contienen una contradicción atómica. Esto significa ahora, que existen en ella dos literales de la forma $s \sqsubseteq t$ y $s \not\sqsubseteq t$.

La corrección del método de tableaux expresa que la insatisfactibilidad de un conjunto de fórmulas Φ es consecuencia de la existencia de un tableau cerrado para él. Como en *LPGD*, la prueba de este resultado se basa en el hecho de que las reglas de expansión preservan la existencia de modelo.

Lema 5.3.1 *Las reglas de expansión preservan la satisfactibilidad de un tableau.*

Demostración. Para las reglas α, β, γ y δ la demostración es inmediata. Para las reglas *Ref* y *Tran*, también es trivial porque las estructuras que consideramos están preordenadas. ■

Teorema 5.3.2 (Corrección) *Dado un conjunto de Σ-sentencias Φ, si Φ tiene un tableau básico cerrado entonces Φ es insatisfactible.*

La completitud del método se establece *à la Hintikka*. Las condiciones que caracterizan ahora un conjunto de Hintikka son un subconjunto de las que exigíamos en la Definición 3.6.5.

Definición 5.3.3 (Conjunto de Hintikka) *Un conjunto de $\overline{\Sigma}$-sentencias H es un conjunto de Hintikka si satisface las siguientes condiciones:*

1. *Si $\varphi_1 \wedge \varphi_2 \in H$ entonces $\varphi_1, \varphi_2 \in H$.*
2. *Si $\neg(\varphi_1 \wedge \varphi_2) \in H$ entonces $\neg\varphi_1 \in H$ o $\neg\varphi_2 \in H$.*
3. *Si $\neg\neg\varphi \in H$ entonces $\varphi \in H$.*
4. *Si $\exists x\varphi \in H$ entonces existe una constante $c \in AC$ tal que $\varphi[c/x] \in H$.*
5. *Si $\neg\exists x\varphi \in H$ entonces $\neg\varphi[t/x] \in H$, para todo término básico t.*
6. *Para todo término básico t, $t \sqsubseteq t \in H$.*
7. *Sean $t_1, t_2, t_3 \in T(\overline{\Sigma})$, si $t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_3 \in H$ entonces $t_1 \sqsubseteq t_3 \in H$.*
8. *H es coherente, es decir ninguna inecuación y su negación pertenecen a H.*

Teorema 5.3.4 *Todo conjunto de Hintikka H tiene un modelo.*

Demostración. Sea $\mathcal{D}$ la $\overline{\Sigma}$-estructura (de *Herbrand*) compuesta por:

- $D = T(\overline{\Sigma})$
- Si $t_1, t_2 \in D$ entonces $t_1 \sqsubseteq^{\mathcal{D}} t_2 \Leftrightarrow_{def} t_1 \sqsubseteq t_2 \in H$
- $f^{\mathcal{D}} : D^n \to D, f^{\mathcal{D}}(t_1, \ldots, t_n) =_{def} f(t_1, \ldots, t_n)$. En particular, $c^{\mathcal{D}} =_{def} c$.

Obsérvese que $\mathcal{D}$ es una estructura porque las condiciones (6) y (7) de la definición de conjunto de Hintikka nos aseguran que $\sqsubseteq^{\mathcal{D}}$ es un preorden. Además es fácil probar que $[\![t]\!]^{\mathcal{D}} = t$, para todo término básico $t \in T(\overline{\Sigma})$. En estas condiciones probamos el teorema haciendo inducción sobre $\varphi \in H$. Veamos los casos básicos:

- $\varphi = t_1 \sqsubseteq t_2$. Resulta trivial que $[\![\varphi]\!]^{\mathcal{D}} = \underline{t}$, por la definición de $\sqsubseteq^{\mathcal{D}}$.

- $\varphi = t_1 \not\sqsubseteq t_2$. Por la coherencia de H, $(t_1 \sqsubseteq t_2) \notin H$, por lo que $[\![t_1 \sqsubseteq t_2]\!]^{\mathcal{D}} = \underline{f}$, por la definición de $\sqsubseteq^{\mathcal{D}}$. ∎

Teorema 5.3.5 (Completitud) *Dado un conjunto finito de Σ-sentencias Φ, si Φ es insatisfactible entonces Φ tiene un tableau básico cerrado.*

Demostración. Supongamos que Φ no tiene un tableau cerrado. Entonces podemos construir un tableau con una rama B que es un conjunto de Hintikka. Obtenemos contradicción porque B, y por tanto Φ, tiene un modelo. ∎

Si consideramos monotonía, debemos añadir reglas que capturen los encadenamientos en posiciones internas. Veamos que, además de las reglas de expansión $\alpha, \beta, \gamma, \delta$, *Ref* y *Tran*, basta con introducir la siguiente regla:

$$Mon) \quad \frac{t_1 \sqsubseteq t_2}{u[t_1]_p \sqsubseteq u[t_2]_p}$$

donde $u \in T(\overline{\Sigma}, X)$ es un contexto cualquiera y p es una posición de u.

Aunque la corrección y la completitud de este nuevo sistema de tableaux básicos tiene que ver con la satisfactibilidad en estructuras monótonas, los resultados se demuestran como para el sistema que no consideraba la monotonía. La corrección se deriva de que las reglas de expansión preservan la existencia de un modelo monótono y la completitud se prueba *à la Hintikka*.

Lema 5.3.6 *Las reglas de expansión preservan la satisfactibilidad de un tableau en estructuras monótonas.*

Demostración. Inmediato para las reglas α, β, γ y δ. Para las reglas *Ref* y *Tran* también es trivial porque las estructuras que manejamos son preórdenes (monótonos). Por último usamos el Lema 3.5.5(3) para probar que *Mon* también preserva modelos monótonos. ∎

Teorema 5.3.7 (Corrección) *Dado un conjunto de Σ-sentencias Φ, si Φ tiene un tableau básico cerrado entonces Φ es insatisfactible en estructuras monótonas.*

Con la aparición de la monotonía, debemos modificar las condiciones que caracterizan un conjunto de Hintikka. Veamos que basta con añadir una única condición.

Definición 5.3.8 (Conjunto de Hintikka) *Un conjunto de Hintikka H es monótono si, además de las condiciones de la Definición 5.3.3, satisface la siguiente condición:*

9. Sean $t_1, t_2, t \in T(\overline{\Sigma})$ y p una posición de t, si $t_1 \sqsubseteq t_2 \in H$ entonces $t[t_1]_p \sqsubseteq t[t_2]_p \in H$.

Teorema 5.3.9 *Todo conjunto de Hintikka monótono H tiene un modelo monótono.*

Demostración. Sea $\mathcal{D}$ la misma $\overline{\Sigma}$-estructura de *Herbrand* usada en la demostración del Teorema 5.3.4. Primero probamos que $\mathcal{D}$ es una estructura monótona. Usando las condiciones 5.3.3(6) y 5.3.3(7), deducimos que $\sqsubseteq^{\mathcal{D}}$ es un preorden. Usando ahora la condición 5.3.8(9), obtenemos su monotonía. En efecto, sea $f \in \mathcal{F}^n$ y $t_1, \ldots, t_i, t'_i, \ldots t_n \in T(\overline{\Sigma})$ tales que $t_i \sqsubseteq^{\mathcal{D}} t'_i$. Entonces $t_i \sqsubseteq t'_i \in H$, por lo que $f(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq f(t_1, \ldots, t'_i, \ldots, t_n) \in H$, por la condición 5.3.8(9). En consecuencia, $[\![f(t_1, \ldots, t_i, \ldots, t_n) \sqsubseteq f(t_1, \ldots, t'_i, \ldots, t_n)]\!]^{\mathcal{D}} = \underline{t}$.

Para probar que $\mathcal{D}$ es modelo de H procedemos por inducción sobre la estructura de la fórmula, como en la demostración del Teorema 5.3.4. ∎

Teorema 5.3.10 (Completitud) *Dado un conjunto finito de Σ-sentencias Φ, si Φ es insatisfactible en estructuras monótonas entonces Φ tiene un tableau básico cerrado.*

5.4 Unificación preordenada rígida

En esta sección mostramos cómo resolver los problemas de unificación que surgen cuando cerramos una única rama de un tableau con variables libres. Construimos estos tableaux mediante las reglas α, β, δ' y γ' como en *LPO*, sin embargo para cerrar ramas procedemos como sigue. Primero formamos los conjuntos $\mathcal{I}$ y $\mathcal{DI}$ de inecuaciones y desinecuaciones, respectivamente, que aparecen en la rama y, después, cerramos B si encontramos una sustitución θ tal que $\mathcal{I}\theta \models_p s'\theta \sqsubseteq t'\theta$, para alguna desinecuación $s' \not\sqsubseteq t' \in \mathcal{DI}$. Mostraremos que este problema semántico puede resolverse unificando sintácticamente el conjunto $\mathcal{I} \cup \{s' \sqsubseteq t'\}$.

Definición 5.4.1 *Un problema de Unificación Rígida Preordenada (de forma abreviada URP) es una expresión de la forma $\mathcal{I} \vdash s \sqsubseteq t$, donde $\underline{\mathcal{I}}$, $s \sqsubseteq t$ son una teoría finita y una inecuación, respectivamente. Una sustitución θ es una solución de, o resuelve, $\mathcal{I} \vdash s \sqsubseteq t$ si θ es básica y $\mathcal{I}\theta \models_p (s \sqsubseteq t)\theta$.*

Para resolver un *URP*-problema Γ consideraremos restricciones C de las que extraeremos las soluciones de Γ. Con este objetivo, definimos un cálculo de unificación que opera sobre parejas $\Gamma \cdot C$, de forma que resolver Γ y C significa buscar una sustitución que resuelva ambos.

Definición 5.4.2 *Un URP-problema restringido es un par compuesto por un URP-problema Γ y una restricción C, y se escribe $\Gamma \cdot C$. Una sustitución θ es una solución de, o resuelve, $\Gamma \cdot C$ si θ soluciona Γ y C.*

El cálculo de unificación $\mathcal{P}$ transforma *URP*-problemas restringidos mediante las dos reglas siguientes:

(left↓) Encadenamiento en la raíz por la izquierda $\qquad \dfrac{\{l \sqsubseteq r\} \cup \mathcal{I} \vdash s \sqsubseteq t \cdot C}{\mathcal{I} \vdash r \sqsubseteq t \cdot C \cup \{l \simeq s\}}$

(end) Cierre Reflexivo $\dfrac{\mathcal{I} \vdash s \sqsubseteq t \cdot C}{\mathcal{I} \vdash s \sqsubseteq s \cdot C \cup \{s \simeq t\}}$

El uso de estas reglas está restringido por las siguientes condiciones de aplicabilidad adicionales:

a) la restricción de la conclusión tiene que ser satisfactible,

b) la inecuación $s \sqsubseteq t$ de la premisa no es *trivial* ($s \neq t$),

c) la inecuación $l \sqsubseteq r$ no es trivial en la regla *left*$\downarrow$.

La primera condición asegurará la corrección del cálculo, la segunda establece que las inecuaciones triviales representan el final de la búsqueda, mientras que la tercera se añade como forma de optimización.

Hacemos notar que sólo introducimos restricciones de igualdad; las restricciones de orden aparecerán en la Sección 6 cuando mejoremos el cálculo con técnicas de reescritura. Obsérvese también que las inecuaciones son eliminadas de la teoría una vez que han sido usadas; esto determinará la terminación del cálculo.

En adelante, representamos los $\mathcal{P}$-pasos del cálculo por $\Gamma \cdot C \leadsto_{\mathcal{P}} \Gamma' \cdot C'$, y usamos $\leadsto^*_{\mathcal{P}}$ para indicar su cierre reflexivo y transitivo.

Definición 5.4.3 *La sustitución θ es un $\mathcal{P}$-unificador del URP-problema $\mathcal{I} \vdash s \sqsubseteq t$ si existe una teoría $\mathcal{I}'$, un término $r \in T(\Sigma, X)$ y una restricción C tal que:*

- $\mathcal{I} \vdash s \sqsubseteq t \cdot \emptyset \leadsto^*_{\mathcal{P}} \mathcal{I}' \vdash r \sqsubseteq r \cdot C$
- *θ es una solución para C.*

Dado un *URP*-problema Γ, podemos ver el cálculo de sus $\mathcal{P}$-unificadores como una búsqueda en el $\mathcal{P}$-árbol de derivación que tiene $\Gamma \cdot \emptyset$ en la raíz. Obsérvese que una hoja con éxito contiene una teoría, una inecuación trivial y una restricción que posiblemente representará varios $\mathcal{P}$-unificadores.

5.4.1 Propiedades del cálculo $\mathcal{P}$

Teorema 5.4.4 (Terminación) *El $\mathcal{P}$-árbol de derivación para cada URP-problema es finito.*

Demostración. Es inmediato puesto que cada $\mathcal{P}$-paso consume una inecuación de la teoría o produce una inecuación trivial. ∎

El cálculo $\mathcal{P}$ también es correcto en el sentido de que cada $\mathcal{P}$-unificador es una solución del correspondiente *URP*-problema. La demostración es consecuencia directa del siguiente lema.

Lema 5.4.5 *Si $\Gamma \cdot C \leadsto_{\mathcal{P}} \Gamma' \cdot C'$ y θ es una sustitución que resuelve $\Gamma' \cdot C'$ entonces θ resuelve $\Gamma \cdot C$.*

Demostración. Sea $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$ y $\Gamma' = \mathcal{I}' \vdash s' \sqsubseteq t'$. Distinguimos casos de acuerdo con la regla aplicada en el $\mathcal{P}$-paso:

$(left\downarrow)$ $\mathcal{I} = \mathcal{I}' \cup \{l \sqsubseteq r\}$, $C' = C \cup \{l \simeq s\}$, $s' = r$ y $t' = t$. Obviamente θ resuelve C. Ahora dada una interpretación $\langle \mathcal{D}, \rho \rangle$ que satisfaga $\mathcal{I}\theta$, también satisface $\mathcal{I}'\theta$, y por tanto $(s' \sqsubseteq t')\theta$. En estas condiciones, $\langle \mathcal{D}, \rho \rangle \models (l \sqsubseteq r)\theta \wedge (r \sqsubseteq t)\theta$, por lo que también satisface $(l \sqsubseteq t)\theta$. Concluimos que $\langle \mathcal{D}, \rho \rangle \models (s \sqsubseteq t)\theta$, puesto que θ resuelve C', y entonces $l\theta = s\theta$.

(end) $\mathcal{I}' = \mathcal{I}$, $C' = C \cup \{s \simeq t\}$, $s' = s$ y $t' = s$. De nuevo es inmediato que θ resuelve C. Si $\langle \mathcal{D}, \rho \rangle \models \mathcal{I}\theta$ entonces $\langle \mathcal{D}, \rho \rangle \models (s \sqsubseteq t)\theta$, puesto que θ resuelve C' y $s \simeq t \in C'$. ■

Teorema 5.4.6 (Corrección) *Sea Γ un URP-problema y θ un $\mathcal{P}$-unificador de Γ, entonces θ es una solución de Γ.*

Demostración. Sea $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$, entonces existe una restricción C tal que θ es una solución de C y $\mathcal{I} \vdash s \sqsubseteq t \cdot \emptyset \leadsto^*_{\mathcal{P}} \mathcal{I}' \vdash r \sqsubseteq r \cdot C$, para alguna teoría $\mathcal{I}'$ y algún término $r \in T(\Sigma, X)$. Utilizando inducción sobre la longitud de la $\mathcal{P}$-derivación y el lema anterior, concluimos que θ es una solución de Γ. ■

Recíprocamente, $\mathcal{P}$ es completo porque cada solución de un *URP*-problema Γ dado, puede obtenerse como un $\mathcal{P}$-unificador, aplicando el cálculo a $\Gamma \cdot \emptyset$. Antes de demostrar la completitud, caracterizamos sintácticamente cuándo un *URP*-problema tiene solución. Lo haremos mediante el concepto de *Teoría Rígida Preordenada* y demostrando una variante inecuacional del Teorema de Birkhoff.

Definición 5.4.7 *Sea $\mathcal{I}$ una teoría, la Teoría Rígida Preordenada inducida por $\mathcal{I}$, escrito $Th_p(\mathcal{I})$, es el mínimo conjunto de inecuaciones que cumple:*

1. $\mathcal{I} \subseteq Th_p(\mathcal{I})$.
2. $t \sqsubseteq t \in Th_p(\mathcal{I})$, *para todo* $t \in T(\Sigma, X)$.
3. *Si* $t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_3 \in Th_p(\mathcal{I})$ *entonces* $t_1 \sqsubseteq t_3 \in Th_p(\mathcal{I})$, *para todo* $t_1, t_2, t_3 \in T(\Sigma, X)$.

Teorema 5.4.8 (Teorema de Birkhoff) $\mathcal{I} \models_p s \sqsubseteq t$ *si y sólo si* $s \sqsubseteq t \in Th_p(\mathcal{I})$.

Demostración. (*sólo si*) Consideramos la estructura $\mathcal{FM}(\mathcal{I})$ que extiende con variables la estructura de Herbrand del Teorema 5.3.4; esto es, $\mathcal{FM}(\mathcal{I})$ está compuesta por:

- $D = T(\Sigma, X)$
- $s \sqsubseteq^{\mathcal{FM}(\mathcal{I})} t \Leftrightarrow_{def} s \sqsubseteq t \in Th_p(\mathcal{I})$
- $f^{\mathcal{FM}(\mathcal{I})} = f$

Obsérvese que, por la definición de $Th_p(\mathcal{I})$, $\sqsubseteq^{\mathcal{FM}(\mathcal{I})}$ es un preorden. Sea la interpretación formada por la estructura $\mathcal{FM}(\mathcal{I})$ y la valoración id(entidad). Es fácil probar que $[\![t]\!]_{id}^{\mathcal{FM}(\mathcal{I})} = t$, para todo $t \in T(\Sigma, X)$. Entonces resulta obvio que $\langle \mathcal{FM}(\mathcal{I}), id\rangle \models \mathcal{I}$ por lo que $\langle \mathcal{FM}(\mathcal{I}), id\rangle \models s \sqsubseteq t$. En estas condiciones, $s = [\![s]\!]_{id}^{\mathcal{FM}(\mathcal{I})} \sqsubseteq^{\mathcal{FM}(\mathcal{I})} [\![t]\!]_{id}^{\mathcal{FM}(\mathcal{I})} = t$, por lo que $s \sqsubseteq t \in Th_p(\mathcal{I})$.

(si) Distinguimos casos de acuerdo con la definición de $Th_p(\mathcal{I})$; es trivial cuando $s \sqsubseteq t \in \mathcal{I}$ o $t \sqsubseteq t \in Th_p(\mathcal{I})$; si $t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_3 \in Th_p(\mathcal{I})$ entonces $\mathcal{I} \models_p t_1 \sqsubseteq t_2 \wedge t_2 \sqsubseteq t_3$, por hipótesis de inducción. Usando la transitividad de las estructuras concluimos que $\mathcal{I} \models_p t_1 \sqsubseteq t_3$. ■

La noción de Teoría Rígida Preordenada asegura que $Th_p(\mathcal{I})$ está sintácticamente saturado con respecto a la reflexividad y la transitividad. En este sentido, $Th_p(\mathcal{I})$ captura las condiciones (6) y (7) de conjunto de Hintikka de la Definición 5.3.3. De hecho, la dirección (*sólo si*) del anterior teorema puede entenderse como una extensión del Teorema 5.3.4 que tiene en cuenta variables libres.

Como para el caso ecuacional, usamos *pruebas* para caracterizar cuándo una inecuación pertenece a la Teoría Rígida Preordenada. Dentro de estas pruebas podemos evitar *redundancias* -encadenamientos triviales o ciclos- como muestra el siguiente lema.

Lema 5.4.9 *Sea $\mathcal{I}$ una teoría y $s \sqsubseteq t \in Th_p(\mathcal{I})$ tal que $s \neq t$. Entonces existe una secuencia finita de inecuaciones $l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n \in \mathcal{I}$ tal que:*

1. *$l_1 = s$, $r_n = t$.*
2. *$r_i = l_{i+1}$, para todo $1 \leq i < n$.*
3. *$(l_i \sqsubseteq r_i) \neq (l_j \sqsubseteq r_j)$, para todo $1 \leq i < j \leq n$ (no hay ciclos).*
4. *$l_i \neq r_i$, para todo $1 \leq i \leq n$ (no hay encadenamientos triviales).*

Demostración. Usamos inducción estructural sobre el hecho de que $s \sqsubseteq t \in Th_p(\mathcal{I})$. Si $s \sqsubseteq t \in \mathcal{I}$ entonces es trivial. En caso contrario, existe un término w tal que $s \sqsubseteq w, w \sqsubseteq t \in Th_p(\mathcal{I})$. Por hipótesis de inducción, existen dos secuencias $(s =)l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n(= w) \in \mathcal{I}$ y $(w =)u_1 \sqsubseteq v_1, \ldots, u_m \sqsubseteq v_m(= t) \in \mathcal{I}$ que satisfacen las condiciones del lema. Si todas las inecuaciones son disjuntas dos a dos, unimos las secuencias; si no, sea $i \in \{1 \ldots n\}$ el menor índice para el que existe $j \in \{1 \ldots m\}$ con $(l_i \sqsubseteq r_i) = (u_j \sqsubseteq v_j)$. Entonces la secuencia $l_1 \sqsubseteq r_1, \ldots, l_i \sqsubseteq r_i, u_{j+1} \sqsubseteq v_{j+1}, \ldots, u_m \sqsubseteq v_m$ satisface las condiciones requeridas. ■

Volviendo al cálculo $\mathcal{P}$, su completitud significa que debe existir un $\mathcal{P}$-unificador para un *URP*-problema Γ dado, siempre que exista una sustitución θ que lo resuelva. De hecho, $\mathcal{P}$ es capaz de construir el propio θ. En primer lugar mostramos que podemos dar $\mathcal{P}$-pasos cuando el *URP*-problema restringido tenga solución. La idea consiste en elevar la prueba relacionada con $(s \sqsubseteq t)\theta \in Th_p(\mathcal{I}\theta)$, cuya existencia está asegurada por el lema anterior, si $s\theta \neq t\theta$.

Lema 5.4.10 *Si θ es una solución de $\mathcal{I} \vdash s \sqsubseteq t$ ($s \neq t$) y C, entonces existen $\mathcal{I}', C', s', t'$ tales que $\mathcal{I} \vdash s \sqsubseteq t \cdot C \rightsquigarrow_{\mathcal{P}} \mathcal{I}' \vdash s' \sqsubseteq t' \cdot C'$ y θ es una solución de $\mathcal{I}' \vdash s' \sqsubseteq t'$ y C'.*

Demostración. Si θ es una solución de $\mathcal{I} \vdash s \sqsubseteq t$ entonces $\mathcal{I}\theta \models_p (s \sqsubseteq t)\theta$ por lo que $(s \sqsubseteq t)\theta \in Th_p(\mathcal{I}\theta)$, por el Teorema 5.4.8. Distingamos casos:

1. $s\theta = t\theta$. Entonces aplicamos la regla *end*: $\mathcal{I} \vdash s \sqsubseteq t \cdot C \rightsquigarrow_{\mathcal{P}} \mathcal{I} \vdash s \sqsubseteq s \cdot C \cup \{s \simeq t\}$. La aplicación es posible porque θ es solución de $C \cup \{s \simeq t\}$. Además resulta obvio que θ también resuelve $\mathcal{I} \vdash s \sqsubseteq s$.

2. $s\theta \neq t\theta$. Por el Lema 5.4.9 existe una secuencia $l'_1 \sqsubseteq r'_1, \ldots, l'_n \sqsubseteq r'_n \in \mathcal{I}\theta$, de inecuaciones no triviales disjuntas dos a dos, tal que $l'_1 = s\theta, r'_n = t\theta, r'_i = l'_{i+1}$, para todo $1 \leq i < n$. Sean l_i y r_i tales que $l_i\theta = l'_i, r_i\theta = r'_i$, para todo $1 \leq i \leq n$, entonces $l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n \in \mathcal{I}$ es también una secuencia de inecuaciones no triviales disjuntas dos a dos. En estas condiciones, aplicamos la regla *left*$\downarrow$: $\mathcal{I} \vdash s \sqsubseteq t \cdot C \rightsquigarrow_{\mathcal{P}} \mathcal{I} \backslash \{l_1 \sqsubseteq r_1\} \vdash r_1 \sqsubseteq t \cdot C \cup \{l_1 \simeq s\}$. Esta aplicación es posible porque las condiciones adicionales de aplicabilidad se cumplen. En efecto, $l_1\theta = l'_1 = s\theta$ por lo que θ es una solución de la nueva restricción, y $l_1 \neq r_1$, ya que la secuencia no contiene inecuaciones triviales. Entonces, dada una interpretación $\langle \mathcal{D}, \rho \rangle$ que satisfaga $(\mathcal{I} \backslash \{l_1 \sqsubseteq r_1\})\theta$, también satisface $(l_2 \sqsubseteq r_2)\theta, \ldots, (l_n \sqsubseteq r_n)\theta$, por lo que $\langle \mathcal{D}, \rho \rangle \models (r_1 \sqsubseteq t)\theta$. Por tanto, θ es una solución de $\mathcal{I} \backslash \{l_1 \sqsubseteq r_1\} \vdash r_1 \sqsubseteq t$. ∎

Teorema 5.4.11 (Completitud) *Sea Γ un URP-problema y θ una solución de Γ, entonces θ es un $\mathcal{P}$-unificador de Γ.*

Demostración. Sea $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$. Como θ es solución de $\emptyset$, podemos aplicar el Lema 5.4.10 para concluir que podemos dar una secuencia de $\mathcal{P}$-pasos. Puesto que $\mathcal{P}$ es terminante, finalizamos en un *URP*-problema restringido $\mathcal{I}' \vdash r \sqsubseteq r \cdot C'$ tal que θ es una solución para C'. Entonces θ es un $\mathcal{P}$-unificador de $\mathcal{I} \vdash s \sqsubseteq t$. ∎

Obsérvese que gracias a los Teoremas 5.4.4, 5.4.6 y 5.4.11, $\mathcal{P}$ puede usarse como un procedimiento de decisión para los *URP*-problemas.

Corolario 5.4.12 *El URP-problema es decidible.*

5.5 Tableaux con variables libres y unificación rígida preordenada simultánea

El cierre de un tableau $\mathcal{T}$ implica el cierre de cada una de sus ramas. En esta sección presentamos un sistema de tableaux que cierra simultáneamente todas las ramas del tableau. Por esta razón, extendemos los *URP*-problemas a problemas *simultáneos*.

Definición 5.5.1 *Un problema de Unificación Rígida Preordenada Simultánea (abreviadamente URPS) consiste en una secuencia finita $\Gamma_1, \ldots, \Gamma_n$ de URP-problemas.*

Presentamos el cálculo $\mathcal{SP}$ como una extensión simultánea del cálculo $\mathcal{P}$. $\mathcal{SP}$ transforma secuencias de *URP*-problemas mediante la aplicación local de las reglas *left*$\downarrow$ y *end* sobre alguno de sus componentes. Como las variables se comportan rígidamente, unimos las restricciones asociadas a cada *URP*-problema en un único conjunto. Con una única restricción global, evitamos la construcción independiente de restricciones locales satisfactibles que pueden no ser compatibles globalmente. Por esta razón, podamos el espacio de búsqueda de forma más eficiente, si cerramos un tableau simultáneamente y utilizamos una única restricción global, en lugar de cerrar cada rama por separado. Al final de la sección haremos algunas reflexiones sobre esta afirmación.

Las reglas del cálculo $\mathcal{SP}$ son:

(sleft$\downarrow$) Encadenamiento en la raíz por la izquierda simultáneo

$$\frac{\Gamma_1, \ldots, \{l \sqsubseteq r\} \cup \mathcal{I}_i \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \mathcal{I}_i \vdash r \sqsubseteq t_i, \ldots, \Gamma_n \cdot C \cup \{l \simeq s_i\}}$$

(send) Cierre reflexivo simultáneo

$$\frac{\Gamma_1, \ldots, \mathcal{I}_i \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \mathcal{I}_i \vdash s_i \sqsubseteq s_i, \ldots, \Gamma_n \cdot C \cup \{s_i \simeq t_i\}}$$

Las condiciones adicionales para la aplicabilidad de estas reglas coinciden con las requeridas por el cálculo $\mathcal{P}$. Las relaciones $\leadsto_{\mathcal{SP}}$ y $\leadsto^*_{\mathcal{SP}}$ y el concepto de $\mathcal{SP}$-unificador se derivan de las análogas para $\mathcal{P}$; obsérvese que, de cada $\mathcal{SP}$-derivación, podemos obtener $\mathcal{P}$-derivaciones globalmente satisfactibles y viceversa. Como para el cálculo $\mathcal{P}$, el cómputo de los $\mathcal{SP}$-unificadores consiste en una búsqueda sobre el $\mathcal{SP}$-árbol de derivación que tiene por raíz $\mathcal{I}_1 \vdash s_1 \sqsubseteq t_1, \ldots, \mathcal{I}_n \vdash s_n \sqsubseteq t_n \cdot \emptyset$.

El *URPS*-problema es también decidible, ya que el cálculo $\mathcal{SP}$ satisface las mismas propiedades que $\mathcal{P}$.

Teorema 5.5.2 (Terminación, Corrección y Completitud) *Sea $\Gamma_1, \ldots, \Gamma_n$ un URPS-problema. Entonces:*

1. *El $\mathcal{SP}$-árbol de derivación para $\Gamma_1, \ldots, \Gamma_n$ es finito.*

2. *Si θ es un $\mathcal{SP}$-unificador de $\Gamma_1, \ldots, \Gamma_n$ entonces θ es una solución de Γ_i, para todo $1 \leq i \leq n$.*

3. *Si θ es una solución de cada URP-problema Γ_i, para todo $1 \leq i \leq n$, entonces θ es un $\mathcal{SP}$-unificador de $\Gamma_1, \ldots, \Gamma_n$.*

Demostración. (1) Si el $\mathcal{SP}$-árbol de derivación es infinito entonces, por el Lema de König, contiene un rama infinita, que puede entenderse como una $\mathcal{P}$-derivación infinita, en contra del Teorema 5.4.4.

(2) Trivial por el Teorema 5.4.6.

(3) Por el Teorema 5.4.11, θ es un $\mathcal{P}$-unificador de Γ_i, para todo $1 \leq i \leq n$; entonces existen teorías, términos y restricciones $\mathcal{I}_i, r_i, C_i (1 \leq i \leq n)$, tales que $\Gamma_i \cdot \emptyset \leadsto^*_{\mathcal{P}} \mathcal{I}_i \vdash r_i \sqsubseteq$

$r_i \cdot C_i$ y θ resuelve cada $C_i, 1 \leq i \leq n$. Sea $C = \bigcup_{i=1}^n C_i$ entonces $\Gamma_1, \ldots, \Gamma_n \cdot \emptyset \leadsto^*_{\mathcal{SP}} \mathcal{I}_1 \vdash r_1 \sqsubseteq r_1, \ldots, \mathcal{I}_n \vdash r_n \sqsubseteq r_n \cdot C$ y θ satisface C. En consecuencia, θ es un $\mathcal{SP}$-unificador de $\Gamma_1, \ldots, \Gamma_n$. ■

Integramos la unificación rígida preordenada simultánea en los tableaux con variables libres sin más que usar el cálculo $\mathcal{SP}$ como regla de cierre. En concreto, sea $\mathcal{SPT}$ el sistema de tableaux compuesto por las reglas $\alpha, \beta, \gamma', \delta'$ y la siguiente regla de cierre:

Definición 5.5.3 (Regla de cierre *URPS***)** *Sea $\mathcal{T}$ un tableau con variables libres y ramas $B_1, \ldots, B_n$. Sea $\mathcal{I}_i$ la teoría que aparece en B_i y $s_i \not\sqsubseteq t_i$ una desinecuación que aparece en B_i, $1 \leq i \leq n$. Entonces $\mathcal{T}$ está cerrado si existe un $\mathcal{SP}$-unificador del URPS-problema $\mathcal{I}_1 \vdash s_1 \sqsubseteq t_1, \ldots, \mathcal{I}_n \vdash s_n \sqsubseteq t_n$.*

Usamos el sistema de tableaux $\mathcal{SPT}$ como sigue:

1. Expandimos el tableau aplicando las reglas α, β, γ' y δ' de forma indeterminista.
2. Seleccionamos una desinecuación de cada rama y buscamos un $\mathcal{SP}$-unificador del *URPS*-problema correspondiente. Si existe entonces hemos finalizado; en caso contrario, escogemos otra desinecuación en alguna rama y probamos de nuevo, o volvemos a 1 si hemos agotado todas las posibilidades.

Obsérvese que sólo consideramos los preórdenes en el paso 2 y que éste siempre acaba. Por tanto, hemos separado la complejidad de los preórdenes de la indecidibilidad de la lógica de primer orden.

La corrección del sistema de tableaux $\mathcal{SPT}$ se deduce del Teorema 5.5.2(2). Para comprobar la completitud, elevamos cada tableau básico cerrado, utilizando el Teorema 5.5.2(3).

Teorema 5.5.4 (Corrección y completitud) *Un conjunto de sentencias Φ es insatisfactible si y sólo si existe un $\mathcal{SPT}$-tableau cerrado para Φ.*

Ejemplo 5.5.5 *Sean a, b, c, d constantes. Entonces podemos concluir $b \sqsubseteq c \vee d \sqsubseteq b$ en cualquier preorden total que verifique $\{a \sqsubseteq c, d \sqsubseteq a\}$. Para demostrar esto, primero construimos el sketch de $\mathcal{SPT}$-tableau $\mathcal{T}$ de la Figura 5.1 y luego lo cerramos utilizando el $\mathcal{SP}$-cálculo tal como muestra la Figura 5.2 (obsérvese que hemos simplificado las restricciones correspondientes). En este caso el $\mathcal{SP}$-unificador obtenido es $[a/x, b/y]$.*

5.5.1 Cierre simultáneo frente a cierre local

Una vez que hemos integrado el cálculo simultáneo $\mathcal{SP}$ en los tableaux con variables libres, uno se pregunta por qué no hemos usado el cálculo local $\mathcal{P}$ para cerrar ramas independientes. Aunque ambas aproximaciones comparten el mismo espacio de búsqueda, preferimos la versión simultánea por dos razones principales: la eficiencia a la hora de podar el espacio de búsqueda, y la elegancia con la que se enuncian los teoremas de completitud.

Veamos algunas evidencias de esto. Primero, obsérvese que podemos usar el cálculo $\mathcal{P}$ de diferentes formas dependiendo de cómo manejemos los unificadores locales obtenidos:

$$
\begin{array}{c}
a \sqsubseteq c \\
d \sqsubseteq a \\
\forall x \forall y (x \sqsubseteq y \vee y \sqsubseteq x) \\
\neg (b \sqsubseteq c \vee d \sqsubseteq b) \\
| \\
b \not\sqsubseteq c \\
d \not\sqsubseteq b \\
\begin{array}{cc}
x \sqsubseteq y \qquad\qquad & \qquad\qquad y \sqsubseteq x
\end{array}
\end{array}
$$

Figura 5.1: El sketch de $\mathcal{SPT}$-tableau $\mathcal{T}$.

	$\{a \sqsubseteq c, d \sqsubseteq a, x \sqsubseteq y\} \vdash d \sqsubseteq b$	$\{a \sqsubseteq c, d \sqsubseteq a, y \sqsubseteq x\} \vdash b \sqsubseteq c$	$\cdot\emptyset$
sleft$\downarrow$	$\{a \sqsubseteq c, x \sqsubseteq y\} \vdash a \sqsubseteq b$	$\{a \sqsubseteq c, d \sqsubseteq a, y \sqsubseteq x\} \vdash b \sqsubseteq c$	$\cdot\{d \simeq d\}$
sleft$\downarrow$	$\{a \sqsubseteq c\} \vdash y \sqsubseteq b$	$\{a \sqsubseteq c, d \sqsubseteq a, y \sqsubseteq x\} \vdash b \sqsubseteq c$	$\cdot\{x \simeq a\}$
send	$\{a \sqsubseteq c\} \vdash y \sqsubseteq y$	$\{a \sqsubseteq c, d \sqsubseteq a, y \sqsubseteq x\} \vdash b \sqsubseteq c$	$\cdot\{x \simeq a, y \simeq b\}$
sleft$\downarrow$	$\{a \sqsubseteq c\} \vdash y \sqsubseteq y$	$\{a \sqsubseteq c, d \sqsubseteq a\} \vdash x \sqsubseteq c$	$\cdot\{x \simeq a, y \simeq b\}$
sleft$\downarrow$	$\{a \sqsubseteq c\} \vdash y \sqsubseteq y$	$\{d \sqsubseteq a\} \vdash c \sqsubseteq c$	$\cdot\{x \simeq a, y \simeq b\}$

Figura 5.2: $\mathcal{SP}$-derivación.

(a) Podemos no aplicar el $\mathcal{P}$-unificador obtenido al cerrar una única rama. Entonces estamos cerrando todas las ramas sucesivamente, manteniendo las restricciones correspondientes a los unificadores locales, de forma separada. En este caso, debemos preguntarnos si la unión de todas estas restricciones es o no satisfactible, antes de cerrar el tableau. Obsérvese que podría ser posible computar un conjunto de restricciones localmente satisfactibles que no fueran globalmente satisfactibles. Así, no podemos concluir si el cierre del tableau es aplicable hasta que no hemos computado todos los unificadores locales separademente. Por este motivo, una única restricción global prevé estas situaciones, lo que supone una poda eficaz del espacio de búsqueda.

Además, el teorema de completitud debería asegurar que *"$\mathcal{P}$ es capaz de computar un conjunto de restricciones locales que es globalmente satisfactible"* (como en [DV 98]). En este sentido, parece que una aproximación simultánea permite expresar dicho teorema de forma más elegante.

(b) Podemos aplicar el $\mathcal{P}$-unificador (el unificador de máxima generalidad correspondiente a la restricción con éxito) tras el cierre de una única rama, independientemente de cómo mezclemos el cierre de las ramas con el proceso de expansión. Entonces el teorema de completitud debería asegurar que *"todavía existe una solución para el nuevo tableau"* que parece ser peor que el correspondiente a una aproximación simultánea. Sin embargo, en este caso no habría diferencia con respecto a la poda del espacio de búsqueda.

5.6 Técnicas de reescritura para preórdenes

En esta sección introducimos estrategias de reescritura con el fin de reducir el $\mathcal{P}$-espacio de búsqueda; después aplicaremos las mismas ideas sobre el cálculo $\mathcal{SP}$.

Supongamos que queremos resolver un *URP*-problema básico $\mathcal{I} \vdash s \sqsubseteq t$, esto es, las inecuaciones de $\mathcal{I}$ y $s \sqsubseteq t$ son básicas. Entonces el cálculo $\mathcal{P}$ es capaz de computar el cierre transitivo de la teoría $\mathcal{I}$, fijando s como elemento mínimo, es decir, $\mathcal{P}$ puede encontrar cualquier término básico t' tal que $\mathcal{I} \models_p s \sqsubseteq t'$, para luego tratar de probar $\mathcal{I} \vdash t' \sqsubseteq t$. Por ejemplo, en el *URP*-problema $\Gamma = \{a \sqsubseteq b, a \sqsubseteq c, a \sqsubseteq d\} \vdash a \sqsubseteq b$ podemos aplicar la regla *left*$\downarrow$ de tres formas diferentes, usando que las constantes b, c y d son mayores o iguales que a. En estos casos, el espacio de búsqueda puede podarse drásticamente, si restringimos la aplicación de la regla *left*$\downarrow$ a las inecuaciones $l \sqsubseteq r \in \mathcal{I}$ que verifiquen $l \succ r$, para algún orden de reducción $\succ$. Por ello, si tenemos $c \succ d \succ a \succ b$ en el ejemplo anterior, las tres elecciones se reducen a una (aquella que usa $a \sqsubseteq b$).

Si introducimos órdenes de reducción, necesitamos nuevas reglas para asegurar la completitud del cálculo. Por ejemplo, si $b \succ a$ entonces no podemos resolver el *URP*-problema satisfactible $\{a \sqsubseteq b\} \vdash a \sqsubseteq b$, ya que $a \not\succ b$. En esta situación, debemos permitir reescribir b en a, en el lado derecho de la inecuación $a \sqsubseteq b$, por lo que necesitamos la regla *right*$_\succ\downarrow$ que aparece más abajo. Sin embargo, esta versión simétrica de la regla *left*$\downarrow$ restringida es aún insuficiente, puesto que podemos encontrarnos con *picos* con respecto al orden $\succ$ que no pueden resolverse: si $b \succ a$ y $b \succ c$ entonces no podemos resolver el *URP*-problema satisfactible $\{a \sqsubseteq b, b \sqsubseteq c\} \vdash a \sqsubseteq c$. En este caso, necesitamos reescribir en las inecuaciones de la teoría, por lo que recurrimos a la nueva regla $peaks_{\succ}$. De este modo definimos el cálculo $\mathcal{P}1$ como el compuesto por las cuatro reglas siguientes:

($left_{\succ} \downarrow$) Encadenamiento en la raíz por la izquierda

$$\frac{\{l \sqsubseteq r\} \cup \mathcal{I} \vdash s \sqsubseteq t \cdot C}{\mathcal{I} \vdash r \sqsubseteq t \cdot C \cup \{l \simeq s, l \succ r\}}$$

($right_{\succ} \downarrow$) Encadenamiento en la raíz por la derecha

$$\frac{\{l \sqsubseteq r\} \cup \mathcal{I} \vdash s \sqsubseteq t \cdot C}{\mathcal{I} \vdash s \sqsubseteq l \cdot C \cup \{t \simeq r, r \succ l\}}$$

($peaks_{\succ}$) Encadenamiento de picos

$$\frac{\{l_1 \sqsubseteq r_1, l_2 \sqsubseteq r_2\} \cup \mathcal{I} \vdash s \sqsubseteq t \cdot C}{\{l_1 \sqsubseteq r_2\} \cup \mathcal{I} \vdash s \sqsubseteq t \cdot C \cup \{r_1 \simeq l_2\ r_1 \succ l_1, l_2 \succ r_2\}}$$

(end) Cierre Reflexivo

$$\frac{\mathcal{I} \vdash s \sqsubseteq t \cdot C}{\mathcal{I} \vdash s \sqsubseteq s \cdot C \cup \{s \simeq t\}}$$

Las condiciones adicionales para la aplicabilidad de estas reglas son similares a las requeridas previamente para $\mathcal{P}$:

a) la restricción de la conclusión tiene que ser satisfactible,

b) la inecuación $s \sqsubseteq t$ de la premisa no es *trivial* ($s \neq t$),

c) las inecuaciones de la teoría usadas en cada regla ($l \sqsubseteq r$ en $left_{\succ} \downarrow$ y $right_{\succ} \downarrow$, $l_1 \sqsubseteq r_1, l_2 \sqsubseteq r_2$ en $peaks_{\succ}$) no son triviales.

Además, definimos las relaciones $\leadsto_{\mathcal{P}1}$, $\leadsto^*_{\mathcal{P}1}$ y el concepto de $\mathcal{P}1$-unificador como para el cálculo $\mathcal{P}$.

En cierto sentido, podemos entender $\mathcal{P}1$ como una restricción de $\mathcal{P}$ que considera restricciones de orden. En consecuencia, $\mathcal{P}1$ es también terminante y correcto.

Teorema 5.6.1 (Terminación) *El $\mathcal{P}1$-árbol de derivación para cada URP-problema es finito.*

Demostración. Las reglas $left_{\succ} \downarrow$, $right_{\succ} \downarrow$ y $peaks_{\succ}$ consumen inecuaciones de la teoría. ■

Teorema 5.6.2 (Corrección) *Sea Γ un URP-problema y θ un $\mathcal{P}1$-unificador para Γ, entonces θ es una solución para Γ.*

Demostración. Supongamos que $\Gamma \cdot C \leadsto_{\mathcal{P}1} \Gamma' \cdot C'$ y que θ resuelve $\Gamma' \cdot C'$, entonces probaremos que θ también resuelve $\Gamma \cdot C$. En estas condiciones, la corrección de $\mathcal{P}1$ se deduce usando inducción sobre la longitud de la $\mathcal{P}1$-derivación, como en el Teorema 5.4.6. Sea $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$ y $\Gamma' = \mathcal{I}' \vdash s' \sqsubseteq t'$. Distingamos casos de acuerdo con la regla aplicada en el $\mathcal{P}1$-paso. Los casos para *left*$_\succ \downarrow$, *right*$_\succ \downarrow$ y *end* son similares al cálculo $\mathcal{P}$; para *peaks*$_\succ$ tenemos que $\mathcal{I} = \mathcal{I}'' \cup \{l_1 \sqsubseteq r_1, l_2 \sqsubseteq r_2\}$, $\mathcal{I}' = \mathcal{I}'' \cup \{l_1 \sqsubseteq r_2\}$, $C' = C \cup \{r_1 \simeq l_2, r_1 \succ l_1, l_2 \succ r_2\}$, $s' = s$ y $t' = t$. Obviamente θ resuelve C. Por otra parte, dada una interpretación $\langle \mathcal{D}, \rho \rangle$ que satisfaga $\mathcal{I}\theta$, también verifica $\mathcal{I}''\theta$ y $(l_1 \sqsubseteq r_1)\theta, (l_2 \sqsubseteq r_2)\theta$, y por tanto verifica $(l_1 \sqsubseteq r_2)\theta$, puesto que $r_1\theta = l_2\theta$. De esta forma, $\langle \mathcal{D}, \rho \rangle \models \mathcal{I}'\theta$, por lo que $\langle \mathcal{D}, \rho \rangle \models (s \sqsubseteq t)\theta$. ■

Ahora podemos usar el Teorema 5.4.8 y el Lema 5.4.9 para probar que $\mathcal{P}1$ también es completo, siempre que partamos de soluciones que instancien básicamente las variables del *URP*-problema. Esto se debe a que nuestro orden de reducción $\succ$ sólo es total sobre términos básicos y, por tanto, no debemos permitir pruebas de $(s \sqsubseteq t)\theta \in Th_p(\mathcal{I}\theta)$ que contengan variables libres.

Teorema 5.6.3 (Completitud) *Sea θ una solución del URP-problema Γ que instancia de forma básica las variables de Γ, entonces θ es un $\mathcal{P}1$-unificador para Γ.*

Este teorema puede demostrarse como el Teorema 5.4.11; esto es, usando el siguiente lema técnico y aplicando inducción sobre la longitud de la $\mathcal{P}1$-derivación.

Lema 5.6.4 *Supongamos que θ resuelve el URP-problema $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$ $(s \neq t)$, y una restricción C. Supongamos que θ instancia de forma básica $var(\Gamma)$, entonces existen $\mathcal{I}', C', s', t'$ tales que $\mathcal{I} \vdash s \sqsubseteq t \cdot C \leadsto_{\mathcal{P}1} \mathcal{I}' \vdash s' \sqsubseteq t' \cdot C'$ y θ resuelve $\mathcal{I}' \vdash s' \sqsubseteq t'$ y C'.*

Demostración. Puesto que θ resuelve $\mathcal{I} \vdash s \sqsubseteq t$, tenemos $\mathcal{I}\theta \models_p (s \sqsubseteq t)\theta$. Por el Teorema 5.4.8, $(s \sqsubseteq t)\theta \in Th_p(\mathcal{I}\theta)$. Entonces distinguimos casos:

1. $s\theta = t\theta$. Podemos aplicar *end* y θ resuelve $C \cup \{s \simeq t\}$ y $\mathcal{I} \vdash s \sqsubseteq s$.

2. $s\theta \neq t\theta$. Por el Lema 5.4.9, existe una secuencia $l'_1 \sqsubseteq r'_1, \ldots, l'_n \sqsubseteq r'_n \in \mathcal{I}\theta$ de inecuaciones no triviales y disjuntas dos a dos, tal que $l'_1 = s\theta, r'_n = t\theta, r'_i = l'_{i+1}$, para todo $1 \leq i < n$. Entonces existe una secuencia de inecuaciones no triviales y disjuntas dos a dos $l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n \in \mathcal{I}$, tal que $l_i\theta = l'_i, r_i\theta = r'_i$, para todo $1 \leq i \leq n$, y $l_1\theta = s\theta, r_n\theta = t\theta$. Como θ instacia básicamente todas las variables de Γ, distingamos los siguientes casos excluyentes:

   (a) $l_1\theta \succ r_1\theta$. Entonces $\mathcal{I} \vdash s \sqsubseteq t \cdot C \leadsto_{\mathcal{P}} \mathcal{I}\backslash\{l_1 \sqsubseteq r_1\} \vdash r_1 \sqsubseteq t \cdot C \cup \{l_1 \simeq s, l_1 \succ r_1\}$, por la regla *left*$_\succ \downarrow$. El paso es posible porque θ resuelve la nueva restricción C'. Además, como en el Lema 5.4.10, podemos demostrar que θ resuelve $\mathcal{I}\backslash\{l_1 \sqsubseteq r_1\} \vdash r_1 \sqsubseteq t$.

   (b) $r_n\theta \succ l_n\theta$. De manera dual al caso anterior, podemos aplicar la regla *right*$_\succ \downarrow$.

(c) $r_1\theta \succ l_1\theta, l_n\theta \succ r_n\theta$. Tenemos que $l_i\theta = l'_i \neq r'_i = l'_{i+1} = l_{i+1}\theta$, para todo $1 \leq i < n$, y $l_n\theta = l'_n \neq r'_n = r_n\theta$, por lo que $\{l_1\theta, r_1\theta(= l_2\theta), \ldots, (r_{n-1}\theta =)l_n\theta, r_n\theta\}$ es un conjunto de términos básicos tal que $r_1\theta \succ l_1\theta$ y $l_n\theta \succ r_n\theta$; por tanto tiene que existir un $1 \leq i < n$ tal que $l_i\theta \prec r_i\theta = l_{i+1}\theta \succ r_{i+1}\theta$. Por ello, podemos aplicar la regla $peaks_\succ$, ya que θ resuelve la nueva restricción C'. Además, si una interpretación dada $\langle \mathcal{D}, \rho\rangle$ satisface $((\mathcal{I}\backslash\{l_i \sqsubseteq r_i, l_{i+1} \sqsubseteq r_{i+1}\}) \cup \{l_i \sqsubseteq r_{i+1}\})\theta$, entonces $\langle \mathcal{D}, \rho\rangle \models (l_1 \sqsubseteq r_n)\theta$. En consecuencia θ resuelve el nuevo *URP*-problema $\mathcal{I}' \vdash s' \sqsubseteq t'$. ■

Como en la sección anterior, extendemos $\mathcal{P}1$ a un cálculo simultáneo $\mathcal{SP}1$, con el fin de incorporarlo a los tableaux con variables libres. $\mathcal{SP}1$ transforma *URPS*-problemas mediante la aplicación local de las reglas $left_\succ \downarrow$, $right_\succ \downarrow$, $peaks_\succ$ y *end* sobre alguno de los *URP*-problemas que lo componen. El cálculo consta de las siguientes reglas:

($sleft_\succ \downarrow$) Encadenamiento en la raíz por la izquierda simultáneo

$$\frac{\Gamma_1, \ldots, \{l \sqsubseteq r\} \cup \mathcal{I}_i \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \mathcal{I}_i \vdash r \sqsubseteq t_i, \ldots, \Gamma_n \cdot C \cup \{l \simeq s_i, l \succ r\}}$$

($sright_\succ \downarrow$) Encadenamiento en la raíz por la derecha simultáneo

$$\frac{\Gamma_1, \ldots, \{l \sqsubseteq r\} \cup \mathcal{I}_i \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \mathcal{I}_i \vdash s_i \sqsubseteq l, \ldots, \Gamma_n \cdot C \cup \{t_i \simeq r, r \succ l\}}$$

($speaks_\succ$) Encadenamiento de picos simultáneo

$$\frac{\Gamma_1, \ldots, \{l_1 \sqsubseteq r_1, l_2 \sqsubseteq r_2\} \cup \mathcal{I} \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \{l_1 \sqsubseteq r_2\} \cup \mathcal{I} \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C \cup \{r_1 \simeq l_2\ r_1 \succ l_1, l_2 \succ r_2\}}$$

(send) Cierre reflexivo simultáneo

$$\frac{\Gamma_1, \ldots, \mathcal{I}_i \vdash s_i \sqsubseteq t_i, \ldots, \Gamma_n \cdot C}{\Gamma_1, \ldots, \mathcal{I}_i \vdash s_i \sqsubseteq s_i, \ldots, \Gamma_n \cdot C \cup \{s_i \simeq t_i\}}$$

Como el cálculo $\mathcal{SP}$, el cálculo $\mathcal{SP}1$ utiliza una única restricción. Además, las condiciones de aplicabilidad adicionales son las mismas que las de $\mathcal{P}1$. Aplicando los Teoremas 5.6.1, 5.6.2 y 5.6.3, podemos demostrar que $\mathcal{SP}1$ también es terminante, correcto y completo.

Teorema 5.6.5 (Terminación, Corrección y Completitud) *Sea $\Gamma_1, \ldots, \Gamma_n$ un URPS-problema. Entonces:*

1. *El $\mathcal{SP}1$-árbol de derivación para $\Gamma_1, \ldots, \Gamma_n$ es finito.*

2. *Si θ es un $\mathcal{SP}1$-unificador de $\Gamma_1, \ldots, \Gamma_n$ entonces θ es una solución de Γ_i, para todo $1 \leq i \leq n$.*

3. *Si θ es una solución de cada URP-problema Γ_i, para todo $1 \leq i \leq n$, entonces θ es un $\mathcal{SP}1$-unificador de $\Gamma_1, \ldots, \Gamma_n$.*

Sea $\mathcal{SPT}1$ el sistema de tableaux similar a $\mathcal{SPT}$, pero usando el cálculo de unificación simultáneo $\mathcal{SP}1$ en lugar de $\mathcal{SP}$ en su *regla de cierre URPS*, es decir:

Definición 5.6.6 (Regla de cierre *URPS* para $\mathcal{SPT}1$) *Sea $\mathcal{T}$ un tableau con variables libres y ramas $B_1, \ldots, B_n$. Sea $\mathcal{I}_i$ la teoría que aparece en B_i y $s_i \not\sqsubseteq t_i$ una desinecuación que aparece en B_i, $1 \leq i \leq n$. Entonces $\mathcal{T}$ está cerrado si existe un $\mathcal{SP}1$-unificador del URPS-problema $\mathcal{I}_1 \vdash s_1 \sqsubseteq t_1, \ldots, \mathcal{I}_n \vdash s_n \sqsubseteq t_n$.*

Usamos el sistema de tableaux $\mathcal{SPT}1$ como $\mathcal{SPT}$ y probamos su corrección y completitud de forma similar.

Teorema 5.6.7 (Corrección y completitud) *Un conjunto de sentencias Φ es insatisfactible si y sólo si existe un $\mathcal{SPT}1$-tableau cerrado para Φ.*

5.7 Unificación rígida preordenada y monótona

De ahora en adelante sólo consideramos preórdenes monótonos. Por tanto, los problemas de unificación que surgen cuando cerramos una única rama son también *URP*-problemas, pero sus soluciones deben incluir monotonía.

Definición 5.7.1 *Un problema de Unificación Rígida Preordenada y Monótona (de forma abreviada URPM) es una expresión de la forma $\mathcal{I} \vdash s \sqsubseteq t$, donde $\mathcal{I}$ y $s \sqsubseteq t$ son una teoría finita y una inecuación, respectivamente. Una sustitución θ es una solución de, o resuelve, $\mathcal{I} \vdash s \sqsubseteq t$ si θ es básica y $\mathcal{I}\theta \models_m (s \sqsubseteq t)\theta$.*

Con vistas a resolver *URPM*-problemas, uno se pregunta si la generalización natural de la regla *left*$\downarrow$ que permite reemplazamientos en posiciones internas, lo que llamamos *encadenamiento interno*, será suficiente para asegurar la completitud y la terminación del cálculo de unificación. Como vamos a ver en unas pequeñas reflexiones que hacemos antes de continuar, la monotonía introduce demasiados problemas para que ésto sea posible. Hagamos observar, de paso, que la mayoría de estos problemas también están presentes en la E-unificación rígida.

1. **Terminación frente a completitud:**
   Cuando excluimos la monotonía, el conjunto finito de $\mathcal{P}$-restricciones con éxito es suficiente para caracterizar todas las soluciones de un *URP*-problema (Teorema 5.4.11). Por el contrario, podemos necesitar un conjunto infinito, y por tanto un cálculo no terminante, si estamos interesados en todas las soluciones de un *URPM*-problema. Por ejemplo, necesitamos un número infinito de conjuntos de restricciones unitarias ($\{x \simeq f^n(a)\}$, con $n \geq 0$) para obtener el conjunto de todas las soluciones del *URPM*-problema $\{a \sqsubseteq f(a)\} \vdash a \sqsubseteq x$.

   Dos aproximaciones se han seguido para arreglar esta incompatibilidad entre terminación y completitud en el caso de la E-unificación rígida:

(a) diseñar cálculos de unificación terminantes que sean existencialmente completos (por ejemplo [GNPS 90]), esto es, cálculos que sean capaces de encontrar al menos un unificador cuando el problema tiene solución. Esta línea de investigación presenta dos problemas en el caso de los preórdenes monótonos. Por una parte, la propiedad de simetría -no disponible en los preórdenes- juega un papel importante en esta línea, ya que permite la comparación entre soluciones a través del concepto de E-generalidad ($\leq_E$). En efecto, el conjunto de todas las soluciones puede capturarse con un conjunto finito (llamado normalmente *completo*) de E-unificadores. Por ejemplo, en la versión con igualdad del problema anterior $\{a \simeq f(a)\} \vdash a \simeq x$, el conjunto unitario $\{[a/x]\}$ es un conjunto completo de unificadores que engloba todas las soluciones ($\{[f^n(a)/x]\}$ con $n \geq 0$). Por otra parte, se ha demostrado que la E-unificación rígida simultánea es indecidible [DV 96], por lo tanto esta aproximación no puede ser extendida de manera natural a tableaux con igualdad. Es más, la versión simultánea del *URPM*-problema también debe ser indecidible, puesto que la E-unificación rígida simultánea puede reducirse a éste, por lo que topamos con el mismo problema en los tableaux con variables libres

(b) diseñar cálculos incompletos, pero terminantes, que puedan ser integrados en los tableaux dando lugar a sistemas de deducción completos [DV 98]. La clave para esta aproximación consiste en restringir las soluciones básicas que debemos elevar.

Obsérvese que la razón por la que los cálculos $\mathcal{P}$ y $\mathcal{SP}$ eran terminantes no se tiene cuando consideramos la monotonía, debido a que las inecuaciones de la teoría no pueden ser eliminadas una vez utilizadas. Por ejemplo, si resolvemos el *URPM*-problema $\{a \sqsubseteq b\} \vdash f(a,a) \sqsubseteq f(b,b)$, debemos utilizar la inecuación $a \sqsubseteq b$ dos veces.

Como para el caso de la igualdad, las técnicas de reescritura y la restricción de las soluciones básicas que son elevadas parecen ser las herramientas adecuadas para lograr un cálculo de unificación terminante que esté integrado en un método de tableaux completo.

2. **Adivinación de contextos**:
Sea θ una sustitución básica que resuelve el *URPM*-problema $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$. Entonces es obvio que el *URPM*-problema básico asociado $\Gamma\theta = \mathcal{I}\theta \vdash (s \sqsubseteq t)\theta$ también es satisfactible. Sin embargo, los encadenamientos internos utilizados para resolver $\Gamma\theta$, que tienen lugar en las posiciones introducidas por la propia θ, no son aplicables sobre el problema Γ. Por ejemplo, si $\Gamma = \{a \sqsubseteq b\} \vdash x \sqsubseteq y$ y $\theta = [f(a)/x, f(b)/y]$, no podemos usar el encadenamiento interno $f(a) \sqsubseteq f(b)$ usado para resolver $\Gamma\theta$. No obstante, en este caso podemos resolver Γ siguiendo otros caminos: utilizando la regla *end* para obtener la restricción $\{x \simeq y\}$ o utilizando las reglas *left*$\downarrow$ y *end* para obtener $\{x \simeq a, y \simeq b\}$; pero θ no satisface ninguna de estas restricciones, puesto que no construyen el contexto apropiado.

Para el caso de la E-unificación rígida, la solución a este problema ha ido evolucionando gradualmente. Inicialmente los contextos se construían explícitamente mediante el uso de los *axiomas de reflexividad funcional* [Fitt 96]. Más tarde, se evitó tal reconstrucción restringiendo el estudio a las llamadas sustituciones *irreducibles* (cfr. por ejemplo [GNPS 90]), que son aquellas cuyos términos introducidos no admiten ningún encadenamiento interno. No obstante, la simetría es, una vez más, indispensable para definir tal concepto, por lo que esta idea resulta inútil sobre preórdenes monótonos si manejamos variables no lineales*. Por ejemplo, la sustitución $[g(a)/x, g(b)/y]$ resuelve el *URPM*-problema $\{a \sqsubseteq b\} \vdash f(g(b), x, x) \sqsubseteq f(y, y, g(a))$, pero, sorprendentemente, las reglas *left*$\downarrow$ y *end* no pueden resolverlo: *end* no es aplicable y los posibles encadenamientos internos –la instanciación de x a a o de y a b– no tienen éxito debido a que no construyen el contexto adecuado. Por lo tanto, *end* y el encadenamiento interno no son suficientes para asegurar la completitud (existencial). Obsérvese que el mismo problema no se presenta para el caso de la igualdad: podemos resolver fácilmentre el problema $\{a \simeq b\} \vdash f(g(b), x, x) \simeq f(y, y, g(a))$, utilizando un reemplazamiento interno (a en lugar de b -disponible debido a la simetría) y después *end* para obtener la solución $[g(a)/x, g(a)/y]$.

En esta sección presentamos el cálculo de unificación $\mathcal{MP}$ para resolver *URPM*-problemas. Este cálculo sólo resuelve los problemas específicos de los preórdenes monótonos relacionados con la adivinación de contextos, evitando el uso de los axiomas de reflexividad funcional. En este sentido, el cálculo $\mathcal{MP}$ es existencialmente completo, pero no terminante. Sin embargo, $\mathcal{MP}$ representa una base sobre la que integrar las técnicas de reescritura y alcanzar así la deseada terminación.

Para presentar el cálculo $\mathcal{MP}$, deducimos las reglas necesarias para asegurar la completitud. Previamente, caracterizamos sintácticamente cuándo un *URPM*-problema tiene solución y lo hacemos mediante el concepto de *Teoría Rígida Preordenada y Monótona* y probando una nueva variante inecuacional del Teorema de Birkhoff.

Definición 5.7.2 *Sea $\mathcal{I}$ una teoría, la Teoría Rígida Preordenada y Monótona inducida por $\mathcal{I}$, escrito $Th_m(\mathcal{I})$, es el conjunto mínimo de inecuaciones que cumple:*

1. $\mathcal{I} \subseteq Th_m(\mathcal{I})$.
2. $t \sqsubseteq t \in Th_m(\mathcal{I})$, *para todo* $t \in T(\Sigma, X)$.
3. *Si* $t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_3 \in Th_m(\mathcal{I})$ *entonces* $t_1 \sqsubseteq t_3 \in Th_m(\mathcal{I})$, *para todo* $t_1, t_2, t_3 \in T(\Sigma, X)$.
4. *Si* $s \sqsubseteq t \in Th_m(\mathcal{I})$ *entonces* $u[s]_p \sqsubseteq u[t]_p \in Th_m(\mathcal{I})$, *para todo* $u, s, t \in T(\Sigma, X)$ *y toda posición* p *de* u.

Teorema 5.7.3 (Teorema de Birkhoff) $\mathcal{I} \models_m s \sqsubseteq t$ *si y sólo si* $s \sqsubseteq t \in Th_m(\mathcal{I})$.

*Variables que aparecen más de una vez en un término.

Demostración. La parte (*sólo si*) se prueba como en el Teorema 5.4.8. Para la parte (*si*), distinguimos casos, pero sólo demostramos el último. Supongamos que $s \sqsubseteq t, u[s]_p \sqsubseteq u[t]_p \in Th_m(\mathcal{I})$, entonces $\mathcal{I} \models_m s \sqsubseteq t$, por hipótesis de inducción. Sea $\langle \mathcal{D}, \rho \rangle$ una Σ-interpretación monótona tal que $\langle \mathcal{D}, \rho \rangle \models \mathcal{I}$, entonces $[\![s]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}} [\![t]\!]^{\mathcal{D}}_{\rho}$, por lo que $[\![u[s]_p]\!]^{\mathcal{D}}_{\rho} \sqsubseteq^{\mathcal{D}} [\![u[t]_p]\!]^{\mathcal{D}}_{\rho}$, debido a que $\langle \mathcal{D}, \rho \rangle$ es una interpretación monótona. ■

Como en la Sección 4, utilizamos *pruebas* para caracterizar que una inecuación pertenece a una Teoría Rígida Preordenada y Monótona. En este caso, encadenamos inecuaciones pertenecientes a $\mathcal{I}$ después de que hayan sido incluidas en contextos arbitrarios. Recordemos que no se puede impedir el uso repetido de una inecuación en una prueba.

Lema 5.7.4 *Sea $\mathcal{I}$ una teoría y $s \sqsubseteq t \in Th_m(\mathcal{I})$ tal que $s \neq t$. Entonces existe una secuencia finita de inecuaciones $l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n \in \mathcal{I}$ y términos $u_i \in T(\Sigma, X)$ con posiciones p_i, para todo $1 \leq i \leq n$, tales que:*

1. $u_1[l_1]_{p_1} = s$, $u_n[r_n]_{p_n} = t$.
2. $u_i[r_i]_{p_i} = u_{i+1}[l_{i+1}]_{p_{i+1}}$, *para todo* $1 \leq i < n$.
3. $l_i \neq r_i$, *para todo* $1 \leq i < n$ *(no hay encadenamientos triviales).*

Demostración. Procedemos por inducción estructural sobre $s \sqsubseteq t \in Th_m(\mathcal{I})$. La demostración es similar a la del Lema 5.4.9, por tanto sólo consideramos el caso $u[s]_p \sqsubseteq u[t]_p \in Th_m(\mathcal{I})$ con $s \sqsubseteq t \in Th_m(\mathcal{I})$, para algún término u con posición p. Por hipótesis, $u[s]_p \neq u[t]_p$, por lo que $s \neq t$, y entonces, por hipótesis de inducción, existe una secuencia $l_1 \sqsubseteq r_1, \ldots, l_n \sqsubseteq r_n \in \mathcal{I}$ y términos u_i con posiciones p_i, para todo $1 \leq i \leq n$, tales que las condiciones del lema se satisfacen. Construimos la prueba usando esta misma secuencia, pero componiendo los contextos y las posiciones, es decir, consideramos los contextos $u[u_i]_p$ y las posiciones pp_i. ■

Supongamos que la sustitución θ resuelve el *URPM*-problema $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$, entonces $\mathcal{I}\theta \models_m (s \sqsubseteq t)\theta$. O bien $s\theta = t\theta$ o, por el Teorema 5.7.3 y el Lema 5.7.4, existe una *prueba* de la forma:

$$s\theta = u_1[l_1\theta]_{p_1} \sqsubseteq u_1[r_1\theta]_{p_1} = u_2[l_2\theta]_{p_2} \sqsubseteq \ldots \sqsubseteq u_n[r_n\theta]_{p_n} = t\theta \qquad (\S)$$

donde $(l_i \sqsubseteq r_i) \in \mathcal{I}$, $l_i\theta \neq r_i\theta$ y u_i y p_i $(1 \leq i \leq n)$ son términos y posiciones. Para que el cálculo $\mathcal{MP}$ sea completo debe ser capaz de elevar tal prueba cuando intente resolver Γ. Analizamos los encadenamientos dados en (§) para deducir las reglas que debería tener $\mathcal{MP}$:

1. Si $s\theta = t\theta$ entonces podemos aplicar un cierre reflexivo (regla *end*).
2. Si $s\theta \neq t\theta$ entonces $n > 0$. Si hay algún encadenamiento en (§) que use como posición de encadenamiento la raíz, sea i un índice tal que $p_i = \varepsilon$. En esta situación, podemos dividir la prueba en dos partes $s\theta = u_1[l_1\theta]_{p_1} \sqsubseteq \ldots \sqsubseteq l_i\theta$ y $r_i\theta = u_{i+1}[l_{i+1}\theta]_{p_{i+1}} \sqsubseteq \ldots \sqsubseteq t\theta$. Entonces necesitamos algún tipo de encadenamiento en la raíz para resolver el i-ésimo paso de (§) (regla *root* de más abajo).

3. Si $s\theta \neq t\theta$ y no hay encadenamientos en (§) que usen como posición de encadenamiento la raíz, distinguimos los casos siguientes:

   (a) Si ni s, ni t son variables, entonces los dos tienen el mismo símbolo funcional en la raíz, es decir $s = f(s_1, \ldots, s_q)$ y $t = f(t_1, \ldots, t_q)$. Para este caso necesitamos descomponer la prueba (regla *dec*) y dividirla en q pruebas, una para cada $s_i\theta \sqsubseteq t_i\theta$, $1 \leq i \leq q$.

   (b) Si $s = x$ y $t = f(t_1, \ldots, t_q)$ entonces $x\theta = f(s_1, \ldots, s_q)$ y podemos dividir la prueba en q pruebas, una para cada $s_i \sqsubseteq t_i\theta$, $1 \leq i \leq n$. En este caso necesitamos algún tipo de imitación de la raíz (regla *imit1*), permitiendo reemplazar x por $f(y_1, \ldots, y_q)$, donde las variables y_i son nuevas. Para el caso simétrico $s = f(t_1, \ldots, t_q)$ y $t = x$, necesitamos también imitar la raíz (regla *imit2*).

   (c) Si s, t son variables y $s\theta = f(s_1, \ldots, s_q)$, $t\theta = f(t_1, \ldots, t_q)$, entonces podemos dividir la prueba en q pruebas, una para cada $s_i \sqsubseteq t_i$, $1 \leq i \leq n$. En este caso necesitamos imitar f como en el caso anterior.

Antes de definir el cálculo $\mathcal{MP}$, hagamos algunas reflexiones sobre el análisis anterior. Primero destaquemos que un único *URPM*-problema puede dividirse en varios; por ejemplo, en el caso 2, transformaríamos el único *URPM*-problema inicial $\mathcal{I} \vdash s \sqsubseteq t$ en dos nuevos *URPM*-problemas $\mathcal{I} \vdash s \sqsubseteq l$ y $\mathcal{I} \vdash r \sqsubseteq t$. Por lo tanto, debemos hecer frente a varios *URPM*-problemas simultáneamente, a pesar de considerar una sola rama del tableau. La notación $\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n$ representará n *URPM*-problemas de la forma $\mathcal{I} \vdash s_i \sqsubseteq t_i$, $1 \leq i \leq n$. Además, añadimos restricciones a estas expresiones para crear *URPM-problemas restringidos*, como en la Definición 5.4.2.

Segundo, obsérvese que las reglas de imitación de la raíz utilizadas en los casos 3(b) y 3(c) son duales, pero diferentes desde el punto de vista de la implementación, ya que el símbolo de función tiene que ser adivinado en el caso 3(c). De hecho, en este caso, la regla se comporta como los ineficientes axiomas de reflexividad funcional, puesto que construye contextos arbitrarios, a diferencia del caso 3(b), donde los contextos se destruyen. Por esta razón, el cálculo $\mathcal{MP}$ no contempla el caso 3(c) explícitamente, aunque debe construir estos contextos para asegurar la completitud. Esto se consigue aplicando el *umg* obtenido en los casos 1 y 3(b).

El proceso de unificación sigue, pues, la idea siguiente:

1. elevamos todos los encadenamientos correspondientes al resto de los casos, hasta que todas las inecuaciones que estén pendientes de ser probadas sean triviales o tengan la forma $x \sqsubseteq y$, y

2. aplicamos un cierre reflexivo *global* utilizando la regla *end*.

Hagamos observar que el último paso siempre es factible, porque las variables que aparecen en el *URPM*-problema en curso (por ejemplo x, y) no pertenecen al dominio del *umg* que resuelve su restricción asociada, siempre que apliquemos el *umg* obtenido en los casos 1 y 3(b).

Como consecuencia directa del análisis anterior, el cálculo de unificación $\mathcal{MP}$ transforma *URPM*-problemas restringidos mediante la aplicación de las cinco reglas siguientes:

(root) Encadenamiento en la raíz

$$\frac{\{l \sqsubseteq r\} \cup \mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_i \sqsubseteq t_i, \ldots, s_n \sqsubseteq t_n \cdot C}{\{l \sqsubseteq r\} \cup \mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_i \sqsubseteq l, r \sqsubseteq t_i, \ldots, s_n \sqsubseteq t_n \cdot C}$$

(dec) Descomposición

$$\frac{\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, f(u_1, \ldots, u_q) \sqsubseteq f(v_1, \ldots, v_q), \ldots, s_n \sqsubseteq t_n \cdot C}{\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, u_1 \sqsubseteq v_1, \ldots, u_q \sqsubseteq v_q, \ldots, s_n \sqsubseteq t_n \cdot C}$$

(imit1) Imitación de la raíz 1

$$\frac{\mathcal{I} \vdash \ldots, x \sqsubseteq f(v_1, \ldots, v_q), \ldots \cdot C}{\mathcal{I}\tau \vdash (\ldots, y_1 \sqsubseteq v_1, \ldots, y_q \sqsubseteq v_q, \ldots)\tau \cdot C \cup \{x \simeq f(y_1, \ldots, y_q)\}}$$

donde y_i son variables nuevas y $\tau = [f(y_1, \ldots, y_q)/x]$.

(imit2) Imitación de la raíz 2: similar a *imit1*, pero aplicada a $f(v_1, \ldots, v_q) \sqsubseteq x$.

(end) Cierre reflexivo

$$\frac{\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_i \sqsubseteq t_i, \ldots, s_n \sqsubseteq t_n \cdot C}{\mathcal{I}\tau \vdash (s_1 \sqsubseteq t_1, \ldots, s_i \sqsubseteq s_i, \ldots, s_n \sqsubseteq t_n)\tau \cdot C \cup \{s_i \simeq t_i\}}$$

donde τ es un umg de s_i y t_i.

Como para los cálculos anteriores, las condiciones adicionales para la aplicabilidad de las reglas son:

a) la restricción en la conclusión de la regla tiene que ser satisfactible,

b) la inecuación reemplazada en la premisa no es trivial ($s_i \sqsubseteq t_i$ en *root* y *end*, y $f(u_1, \ldots, u_q) \sqsubseteq f(v_1, \ldots, v_q)$ en *dec*),

c) la inecuación $l \sqsubseteq r$ no es trivial en la regla *root*.

Obsérvese que sólo introducimos restricciones de igualdad y que las restricciones sólo se usan para recordar la solución, ya que las sustituciones se aplican sobre el *URPM*-problema en curso, en las reglas *end* e *imit1/2*. Las relaciones $\rightsquigarrow_{\mathcal{MP}}$ y $\rightsquigarrow^*_{\mathcal{MP}}$ se definen como en los cálculos previos.

Definición 5.7.5 *La sustitución θ es un $\mathcal{MP}$-unificador del URPM-problema $\mathcal{I} \vdash s \sqsubseteq t$ si existe una teoría $\mathcal{I}'$, términos $r_i \in T(\Sigma, X)$, para todo $1 \leq i \leq n$, y una restricción C tal que:*

- $\mathcal{I} \vdash s \sqsubseteq t \cdot \emptyset \rightsquigarrow^*_{\mathcal{MP}} \mathcal{I}' \vdash r_1 \sqsubseteq r_1, \ldots, r_n \sqsubseteq r_n \cdot C$
- *θ es una solución de C.*

Dado un *URPM*-problema Γ, podemos ver la computación de sus $\mathcal{MP}$-unificadores como una búsqueda en el $\mathcal{MP}$-árbol de derivación con raíz $\Gamma \cdot \emptyset$. Obsérvese que las hojas con éxito contienen una secuencia de inecuaciones triviales y una restricción que posiblemente representa varios $\mathcal{MP}$-unificadores.

5.7.1 Propiedades del cálculo $\mathcal{MP}$

El cálculo $\mathcal{MP}$ es correcto y existencialmente completo. La corrección se expresa como para el cálculo $\mathcal{P}$ y se deduce fácilmente del siguiente resultado.

Lema 5.7.6 *Si* $\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n \cdot C \leadsto_{\mathcal{MP}} \mathcal{I}' \vdash s'_1 \sqsubseteq t'_1, \ldots, s'_p \sqsubseteq t'_p \cdot C'$ *y* θ *es una sustitución que resuelve* $\mathcal{I}' \vdash s'_i \sqsubseteq t'_i$ $(1 \leq i \leq p)$ *y* C'*, entonces* θ *resuelve* $\mathcal{I} \vdash s_i \sqsubseteq t_i$*, para todo* $1 \leq i \leq n$*, y* C.

Demostración Distinguimos casos de acuerdo con la regla aplicada en el $\mathcal{MP}$-paso. Obsérvese que $C \subseteq C'$ en todos los casos, por lo que θ siempre resuelve C, por hipótesis. Sea $\langle \mathcal{D}, \rho \rangle$ una interpretación monótona que satisface $\mathcal{I}\theta$, entonces tenemos:

(*root*) Como $\mathcal{I} = \mathcal{I}'$, la interpretación satisface:

$$(s_1 \sqsubseteq t_1)\theta, \ldots, (s_i \sqsubseteq l)\theta, (r \sqsubseteq t_i)\theta, \ldots, (s_n \sqsubseteq t_n)\theta$$

Como $(l \sqsubseteq r) \in \mathcal{I}$ entonces $(l \sqsubseteq r)\theta \in \mathcal{I}\theta$, y por tanto $\langle \mathcal{D}, \rho \rangle \models (l \sqsubseteq r)\theta$ y $\langle \mathcal{D}, \rho \rangle \models (s_j \sqsubseteq t_j)\theta$, para todo $1 \leq j \leq n$.

(*dec*) De nuevo $\mathcal{I} = \mathcal{I}'$, por lo que la interpretación satisface:

$$(s_1 \sqsubseteq t_1)\theta, \ldots, (u_1 \sqsubseteq v_1)\theta, \ldots, (u_q \sqsubseteq v_q)\theta, \ldots, (s_n \sqsubseteq t_n)\theta$$

Entonces $\langle \mathcal{D}, \rho \rangle \models (f(u_1, \ldots, u_q) \sqsubseteq f(v_1, \ldots, v_q))\theta$, porque la interpretación es monótona.

(*imit1*) θ resuelve $C' = C \cup \{x \simeq f(y_1, \ldots, y_q)\}$ por lo que $x\theta = f(y_1, \ldots, y_q)\theta$ y, en consecuencia, $\tau\theta = \theta$. Como asumimos que $\langle \mathcal{D}, \rho \rangle \models \mathcal{I}\theta (= \mathcal{I}\tau\theta)$, entonces la interpretación también satisface:

$$(s_1 \sqsubseteq t_1)\tau\theta, \ldots, (y_1 \sqsubseteq v_1)\tau\theta, \ldots, (y_q \sqsubseteq v_q)\tau\theta, \ldots, (s_n \sqsubseteq t_n)\tau\theta$$

Sea $s_j \sqsubseteq t_j$ cualquier inecuación de $\ldots, x \sqsubseteq f(v_1, \ldots, v_q), \ldots$ diferente de $x \sqsubseteq f(v_1, \ldots, v_q)$. Entonces $(s_j \sqsubseteq t_j)\theta$ es satisfecha por $\langle \mathcal{D}, \rho \rangle$ porque $\tau\theta = \theta$. Además tenemos $\langle \mathcal{D}, \rho \rangle \models (f(y_1, \ldots, y_q) \sqsubseteq f(v_1, \ldots, v_q))\theta$ por monotonía, y, como $x\theta = f(y_1, \ldots, y_q)\theta$, concluimos que $\langle \mathcal{D}, \rho \rangle \models (x \sqsubseteq f(v_1, \ldots, v_q))\theta$.

(*imit2*) La demostración es análoga a la de (*imit1*).

(*end*) θ resuelve $C' = C \cup \{s_i \simeq t_i\}$ por lo que $s_i\theta = t_i\theta$. De nuevo, como $\tau = umg(s_i, t_i)$, tenemos que $\tau\theta = \theta$ por generalidad. Como $\langle \mathcal{D}, \rho \rangle \models \mathcal{I}\theta (= \mathcal{I}\tau\theta)$, entonces la interpretación también satisface $(s_j \sqsubseteq t_j)\tau\theta = (s_j \sqsubseteq t_j)\theta$, $1 \leq j \leq n$, $j \neq i$, y de forma trivial $(s_i \sqsubseteq t_i)\theta$. ■

Teorema 5.7.7 (Corrección) *Si* θ *es un* $\mathcal{MP}$*-unificador del URPM-problema* Γ*, entonces* θ *resuelve* Γ.

Demostración. Sea $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$, entonces existe una restricción C tal que $\mathcal{I} \vdash s \sqsubseteq t \cdot \emptyset \rightsquigarrow^*_{\mathcal{MP}} \mathcal{I}' \vdash r_1 \sqsubseteq r_1, \ldots, r_n \sqsubseteq r_n \cdot C$, para cierta teoría $\mathcal{I}'$ y términos $r_i \in T(\Sigma, X)$, para todo $1 \leq i \leq n$, y θ resuelve C. Por inducción sobre la longitud de la $\mathcal{MP}$-derivación y usando el lema anterior, concluimos que θ es una solución de Γ. ■

La completitud existencial del cálculo significa que $\mathcal{MP}$ puede obtener un unificador siempre que el *URPM*-problema tenga solución. Supongamos que θ resuelve $\mathcal{I} \vdash s \sqsubseteq t$, es decir $\mathcal{I}\theta \models_m (s \sqsubseteq t)\theta$. Por el Teorema 5.7.3 resulta que $(s \sqsubseteq t)\theta \in Th_m(\mathcal{I}\theta)$. Entonces se pueden dar dos casos:

1. $s\theta \neq t\theta$. Por el Lema 5.7.4, existe una prueba Π:

$$s\theta = u_1[l_1]_{p_1} \sqsubseteq \ldots \sqsubseteq u_n[r_n]_{p_n} = t\theta$$

basada en una secuencia de inecuaciones no triviales $(l_i \sqsubseteq r_i) \in \mathcal{I}\theta$ $(1 \leq i \leq n)$.

Con objeto de comparar las pruebas que establecen la pertenencia a $Th_m(\mathcal{I}\theta)$, definimos la longitud de una prueba $\|\cdot\|$ como sigue:

$$\|u_1[l_1]_{p_1} \sqsubseteq \ldots \sqsubseteq u_n[r_n]_{p_n}\| =_{def} n$$

Es decir, $\|\Pi\|$ cuenta el número total de encadenamientos usados en Π. Obsérvese que esta medida se refiere a las pruebas y no a la inecuación inicial $s\theta \sqsubseteq t\theta$, debido a que esta inecuación puede tener diversas pruebas asociadas, con distintas longitudes.

2. $s\theta = t\theta$. El Lema 5.7.4 no es aplicable en este caso, por lo que puede existir una prueba como Π o no. Si existe, entonces la longitud de ésta consiste en el cómputo de los encadenamientos, como antes. Si no, extendemos la noción de prueba de forma natural para considerar también aquellas que no encadenan ninguna vez, esto es, admitimos que $s\theta = t\theta$ denota una *prueba trivial* que, por definición, tiene longitud

$$\|s\theta = t\theta\| =_{def} 0$$

.

Como en el cálculo $\mathcal{MP}$ manejamos varios *URPM*-problemas simultáneamente bajo la expresión $\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n$, para cada solución θ se verifica $(s_i \sqsubseteq t_i)\theta \in Th_m(\mathcal{I}\theta)$, para todo $1 \leq i \leq n$. Por este motivo necesitamos comparar secuencias de pruebas en lugar de pruebas individuales. Definamos, pues, la complejidad de una secuencia de pruebas.

Definición 5.7.8 *Sea $\langle \Gamma, \theta \rangle$ un par donde $\Gamma = \mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n$ es un URPM-problema y θ es una solución de Γ. Una secuencia $\mathcal{S} = \Pi_1, \ldots, \Pi_n$ es una secuencia de pruebas para $\langle \Gamma, \theta \rangle$ si Π_i es una prueba de $(s_i \sqsubseteq t_i)\theta \in Th_m(\mathcal{I}\theta)$, para todo $1 \leq i \leq n$. La complejidad de una secuencia de pruebas $\mathcal{S}$ para $\langle \Gamma, \theta \rangle$, escrito $\mu(\mathcal{S}, \Gamma, \theta)$, se define como:*

$$\mu(\Pi_1, \ldots, \Pi_n, , \Gamma, \theta) =_{def} \langle \sum_{i=1}^{n} \|\Pi_i\|, \{\{w(\Pi_i)/s_i \neq t_i, 1 \leq i \leq n\}\} \rangle$$

donde el peso $w(\Pi_i)$ *de la prueba* Π_i *se define como:*

$$w(\Pi_i) =_{def} \langle depth(s_i\theta), depth(t_i\theta)\rangle$$

donde $depth(t)$ *es la profundidad del término* t*. Cuando se deduzca del contexto escribiremos solamente* $\mu(\mathcal{S})$.

Obsérvese que la complejidad de una secuencia de pruebas depende del par $\langle\Gamma, \theta\rangle$ aunque, por razones de legibilidad, la notación $\mu(\Pi_1, \ldots, \Pi_n)$ no hará referencia a este hecho.

Siguiendo el análisis que precedió a la definición del cálculo $\mathcal{MP}$, es fácil comprobar que todas sus reglas transforman secuencias de pruebas para la premisa, en secuencias de pruebas para la conclusión, reduciendo estrictamente la complejidad de las mismas. En concreto, si $\Pi_1, \ldots, \Pi_n$ es la secuencia de pruebas para la premisa, la secuencia para la conclusión reduce el par $\mu(\Pi_1, \ldots, \Pi_n)$ dependiendo de la regla aplicada pues:

- disminuimos la primera componente cuando aplicamos un encadenamiento en la raíz (caso de la regla *root*), porque la suma es más pequeña
- disminuimos la segunda componente cuando pasamos de la prueba de un término funcional a la secuencia de pruebas correspondiente a sus argumentos (caso de las reglas *dec* e *imit1/2*), porque la profundidad de los términos es más pequeña
- la segunda componente del par también disminuye cuando obtenemos una inecuación trivial (caso de la regla *end*), porque descartamos el peso de la prueba del multiconjunto de pruebas correspondiente.

La noción de complejidad nos permite comparar secuencias de pruebas sin más que usar el orden lexicográfico $\succ$ inducido por los órdenes $>$ entre naturales y $\succ_{mult}$ entre multiconjuntos. En estas condiciones, el orden $\succ$ está bien fundamentado porque $>$ y $\succ_{mult}$ también lo están [DM 79].

Bajo ciertas condiciones probaremos que, dada una secuencia de pruebas S para $\langle\Gamma, \theta\rangle$, siempre podemos dar un $\mathcal{MP}$-paso sobre Γ para obtener Γ', de forma que exista otra sustitución θ' que resuelve Γ' y una secuencia de pruebas S' para $\langle\Gamma', \theta'\rangle$ que verifique $\mu(S) \succ \mu(S')$. Este hecho resultará crucial en la demostración de la completitud del cálculo.

Lema 5.7.9 *Sea* C *una restricción y* $\langle\Gamma, \theta\rangle$ *un par compuesto por el URPM-problema* $\Gamma = \mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n$ *y la sustitución* θ *que resuelve* Γ *y* C*. Sea* $\Pi_1, \ldots, \Pi_n$ *una secuencia de pruebas para* $\langle\Gamma, \theta\rangle$ *sin encadenamientos triviales. Supongamos que existe* $i \in \{1, \ldots, n\}$ *tal que la inecuación* $s_i \sqsubseteq t_i$ *no es trivial y ambos términos* s_i, t_i *no son variables a la vez. Entonces existe una sustitución* θ'*, un URPM-problema* Γ' *de la forma* $\mathcal{I}' \vdash s'_1 \sqsubseteq t'_1, \ldots, s'_{i-1} \sqsubseteq t'_{i-1}, u_1 \sqsubseteq v_1, \ldots, u_q \sqsubseteq v_q, s'_{i+1} \sqsubseteq t'_{i+1}, \ldots, s'_n \sqsubseteq t'_n$*, una restricción* C' *y una secuencia de pruebas* $\Pi'_1, \ldots, \Pi'_{i-1}, \Pi^1_i, \ldots, \Pi^q_i, \Pi'_{i+1}, \ldots, \Pi'_n$*, que verifican las siguientes condiciones:*

1. $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$.

2. θ' *resuelve* Γ' *y* C'.

3. $\Pi'_1, \ldots, \Pi'_{i-1}, \Pi^1_i, \ldots, \Pi^q_i, \Pi'_{i+1}, \ldots, \Pi'_n$ *es una secuencia de pruebas para* $\langle \Gamma', \theta' \rangle$ *sin encadenamientos triviales.*

4. $\mu(\Pi_1, \ldots, \Pi_n, \Gamma, \theta) \succ \mu(\Pi'_1, \ldots, \Pi'_{i-1}, \Pi^1_i, \ldots, \Pi^q_i, \Pi'_{i+1}, \ldots, \Pi'_n, \Gamma', \theta')$.

Demostración. Si $s_i\theta = t_i\theta$, consideremos $\tau = umg(s_i, t_i)$. Definimos $C' = C \cup \{s_i \simeq t_i\}$ y Γ' como:

$$\mathcal{I}\tau \vdash (s_1 \sqsubseteq t_1, \ldots, s_i \sqsubseteq s_i, \ldots, s_n \sqsubseteq t_n)\tau$$

entonces tenemos que $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$, mediante la regla *end*. Obsérvese que las condiciones para la aplicabilidad de la regla se cumplen porque θ resuelve C y $s_i\theta = t_i\theta$. Además, por la generalidad de τ, tenemos que $\tau\theta = \theta$, por lo que θ también resuelve Γ' y $\Pi_1, \ldots, \Pi_n$ es una secuencia de pruebas para $\langle \Gamma', \theta \rangle$. Por último, resulta obvio que la complejidad decrece porque, aunque el número de encadenamientos es el mismo, en Γ' aparece al menos una nueva inecuación trivial: $s_i\tau \sqsubseteq t_i\tau$.

Si $s_i\theta \neq t_i\theta$ distinguimos casos dependiendo de la forma de Π_i:

1. Existe un encadenamiento en la raíz en Π_i. Es decir, existe $(l \sqsubseteq r) \in \mathcal{I}$ tal que Π_i usa $(l \sqsubseteq r)\theta$ en la posición ε, y Π_i tiene la forma $s_i\theta \sqsubseteq \ldots \sqsubseteq l\theta \sqsubseteq r\theta \sqsubseteq \ldots \sqsubseteq t_i\theta$.

   Definimos $C' = C$ y Γ' como:

   $$\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_{i-1} \sqsubseteq t_{i-1}, s_i \sqsubseteq l, r \sqsubseteq t_i, s_{i+1} \sqsubseteq t_{i+1}, \ldots, s_n \sqsubseteq t_n$$

   entonces $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$, mediante la regla *root*. Obsérvese que $l \sqsubseteq r$ no es trivial y que θ satisface C'; por tanto las condiciones para la aplicabilidad de la regla se verifican. Además θ es solución de Γ' y $\Pi_1, \ldots, \Pi^1_i, \Pi^2_i, \ldots, \Pi_n$ es una secuencia de pruebas para $\langle \Gamma', \theta \rangle$, donde Π^1_i y Π^2_i son las partes de Π_i correspondientes a las pruebas de que $(s_i \sqsubseteq l)\theta, (r \sqsubseteq t_i)\theta \in Th_m(\mathcal{I}\theta)$. Por último, resulta obvio que la complejidad disminuye porque $\|\Pi_i\| = 1 + \|\Pi^1_i\| + \|\Pi^2_i\|$.

2. No hay encadenamientos en la raíz en Π_i. Distinguimos dos subcasos de acuerdo con la forma de s_i y t_i.

   (a) Los términos s_i y t_i son funcionales. Sean $s_i = f(u_1, \ldots, u_q)$ y $t_i = f(v_1, \ldots, v_q)$. Como $\|\Pi_i\| > 0$, existen pruebas Π^j_i, para todo $1 \leq j \leq q$, de que $u_j\theta \sqsubseteq v_j\theta \in Th_m(\mathcal{I}\theta)$. Consideramos $C' = C$ y Γ' definido como:

   $$\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, s_{i-1} \sqsubseteq t_{i-1}, u_1 \sqsubseteq v_1, \ldots, u_q \sqsubseteq v_q, s_{i+1} \sqsubseteq t_{i+1}, \ldots, s_n \sqsubseteq t_n$$

   entonces $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$, mediante la regla *dec*. Las condiciones de aplicabilidad de la regla se verifican trivialmente. Como en los casos anteriores, θ resuelve Γ' y C'. Además tenemos que $\Pi_1, \ldots, \Pi^1_i, \ldots, \Pi^q_i, \ldots, \Pi_n$ es una secuencia de pruebas para $\langle\Gamma', \theta\rangle$ sin encadenamientos triviales y con menor complejidad. En efecto, resulta que

$$\|\Pi_i\| = \sum_{j=1}^{q} \|\Pi_i^j\|$$

pero $w(\Pi_i) > w(\Pi_i^j)$, para todo $1 \le j \le q$, ya que hemos eliminado el símbolo funcional f de la raíz de los términos s_i y t_i. Por lo tanto, $\{\{\Pi_i\}\} \succ_{mult} \{\{\Pi_i^1, \ldots, \Pi_i^q\}\}$.

(b) Uno de los términos es una variable y el otro, un término funcional. Sin pérdida de generalidad, supongamos que $s_i = x$ y $t_i = f(v_1, \ldots, v_q)$. Como $\|\Pi_i\| > 0$, $x\theta = f(s'_1, \ldots, s'_q)$ y existen pruebas Π_i^j de que $s'_j \sqsubseteq v_j\theta \in Th_m(\mathcal{I}\theta)$, para todo $1 \le j \le q$.
Sean $C' = C \cup \{x \simeq f(y_1, \ldots, y_q)\}$, y_j variables nuevas, y Γ' definido como:

$$\mathcal{I}\tau \vdash (s_1 \sqsubseteq t_1, \ldots, s_{i-1} \sqsubseteq t_{i-1}, y_1 \sqsubseteq v_1, \ldots, y_q \sqsubseteq v_q, s_{i+1} \sqsubseteq t_{i+1}, \ldots, s_n \sqsubseteq t_n)\tau$$

donde $\tau = [f(y_1, \ldots, y_q)/x]$.
Sea $\theta' = \theta[s'_1/y_1, \ldots, s'_q/y_q]$, entonces θ' resuelve C'. En efecto, para toda ecuación $s \simeq t \in C$ tenemos que $s\theta' = s\theta = t\theta = t\theta'$, y para la nueva ecuación tenemos que $x\theta' = x\theta = f(s'_1, \ldots, s'_q) = f(y_1, \ldots, y_q)\theta'$. Entonces las condiciones de aplicabilidad se verifican y $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$, mediante la regla *imit1*. Además $\tau\theta'$ y θ se comportan idénticamente sobre $var(\Gamma)$ por lo que:

- si $j \in \{1, \ldots, n\}$, $j \ne i$, entonces $(s_j \sqsubseteq t_j)\theta \in Th_m(\mathcal{I}\theta)$, por lo que $(s_j \sqsubseteq t_j)\tau\theta' \in Th_m(\mathcal{I}\tau\theta')$
- $(y_j \sqsubseteq v_j)\tau\theta' \in Th_m(\mathcal{I}\tau\theta')$, para todo $1 \le j \le q$, porque $(s'_j \sqsubseteq v_j\tau\theta') \in Th_m(\mathcal{I}\tau\theta')$ e $y_j\tau\theta' = s'_j$.

En estas condiciones, θ' resuelve Γ'. Por el mismo motivo, $\Pi_1, \ldots, \Pi_i^1, \ldots, \Pi_i^q, \ldots, \Pi_n$ es una secuencia de pruebas para $\langle\Gamma', \theta'\rangle$ sin encadenamientos triviales. Como en el caso (a)

$$\|\Pi_i\| = \sum_{j=1}^{q} \|\Pi_i^j\|$$

pero $w(\Pi_i) > w(\Pi_i^j)$, para todo $1 \le j \le q$, por la funcionalidad de los términos $x\theta$ y $t_i\theta$. Por tanto, $\{\{\Pi_i\}\} \succ_{mult} \{\{\Pi_i^1, \ldots, \Pi_i^q\}\}$. ■

El uso repetido del lema anterior nos conduce a un *URPM*-problema $\Gamma \cdot C$ en el que todas las inecuaciones de Γ pendientes de prueba son triviales o de la forma $x \sqsubseteq y$. Entonces, la completitud del cálculo $\mathcal{MP}$ se deriva del siguiente resultado técnico, donde probamos que la unión de la restricción C y de estas inecuaciones pendientes admite un unificador sintáctico.

Lema 5.7.10 *Sea Γ un URPM-problema y C una restricción. Si existe un unificador de máxima generalidad θ de C que verifica $dom(\theta) \cap var(\Gamma) = \emptyset$, y $\Gamma \cdot C \rightsquigarrow_{\mathcal{MP}} \Gamma' \cdot C'$, entonces existe un unificador de máxima generalidad θ' de C' que verifica $dom(\theta') \cap var(\Gamma') = \emptyset$.*

Demostración. Distinguimos casos dependiendo de la regla aplicada:

($root$) Podemos definir $\theta' = \theta$ porque $C = C'$ y $var(\Gamma) = var(\Gamma')$. De manera similar se prueba el caso de la regla dec.

($imit1$) Tenemos $C' = C \cup \{x \simeq f(y_1, \ldots, y_q)\}$ y $var(\Gamma') = (var(\Gamma) \backslash \{x\}) \cup \{y_1, \ldots, y_q\}$, donde $y_j \notin var(\Gamma) \cup var(C)$, para todo $1 \leq j \leq q$. Sea $\tau = [f(y_1, \ldots, y_q)/x]$ y definamos $\theta' = \theta\tau$. Entonces tenemos que:

1. θ' unifica C'. Para toda ecuación $(s \simeq t) \in C$, resulta que $s\theta = t\theta$ por lo tanto $(s\theta)\tau = (t\theta)\tau$; para la nueva ecuación $x \simeq f(y_1, \ldots, y_q)$ razonamos como sigue:

$$
\begin{aligned}
x\theta' &= x\tau \text{ (porque } x \in var(\Gamma), \text{ y entonces } x \notin dom(\theta)) \\
&= f(y_1, \ldots, y_q) \\
&= f(y_1, \ldots, y_q)\theta\tau \text{ (porque } y_j \text{ son variables nuevas)} \\
&= f(y_1, \ldots, y_q)\theta'
\end{aligned}
$$

2. θ' es un unificador de máxima generalidad de C'. En efecto, si η unifica C' entonces η unifica C y $\theta\eta = \eta$, por la generalidad de θ; además η unifica $x \simeq f(y_1, \ldots, y_q)$ y $\tau\eta = \eta$, por la generalidad de τ. En estas condiciones, $\theta'\eta = (\theta\tau)\eta = \theta(\tau\eta) = \theta\eta = \eta$.
3. $var(\Gamma') \cap dom(\theta') = \emptyset$. Esto resulta trivial porque $dom(\theta') = dom(\theta) \cup \{x\}$ y las variables y_j son nuevas.

(end) Tenemos $C' = C \cup \{s_i \simeq t_i\}$ y $var(\Gamma') = (var(\Gamma) \backslash dom(\tau)) \cup codom(\tau)$, donde $\tau = umg(s_i, t_i)$. Definiendo $\theta' = \theta\tau$ resulta:

1. θ' unifica C'. Sólo probamos el caso de la nueva ecuación $s_i \simeq t_i$ de C':

$$
\begin{aligned}
s_i\theta' &= (s_i\theta)\tau \\
&= s_i\tau \text{ (porque } var(s_i) \subseteq var(\Gamma) \text{ y } var(\Gamma) \cap dom(\theta) = \emptyset) \\
&= t_i\tau \\
&= (t_i\theta)\tau \\
&= t_i\theta'
\end{aligned}
$$

2. θ' es un unificador de máxima generalidad de C'. Se demuestra como en el caso de las reglas de imitación.

3. $var(\Gamma') \cap dom(\theta') = \emptyset$. En efecto, tenemos que $dom(\theta') = dom(\theta) \cup dom(\tau)$ por lo que si $z \in var(\Gamma')$ entonces $z \in var(\Gamma) - dom(\tau)$ o $z \in codom(\tau)$. En cualquiera de los casos resulta imposible que $z \in dom(\theta')$: en el primero esto es trivial y en el segundo se deduce del hecho de que τ es un unificador de máxima generalidad de s_i y t_i. ■

A continuación usamos los dos lemas anteriores para establecer la completitud del cálculo $\mathcal{MP}$.

Teorema 5.7.11 (Completitud existencial) *Sea θ una solución del URPM-problema $\Gamma = \mathcal{I} \vdash s \sqsubseteq t$, entonces existe un $\mathcal{MP}$-unificador de Γ.*

Demostración. Consideremos el *URPM*-problema restringido $\Gamma \cdot \emptyset$, entonces θ resuelve Γ y $\emptyset$. Aplicando repetidamente el Lema 5.7.9 concluimos que existe una sustitución θ' tal que:

1. $\Gamma \cdot C \leadsto^*_{\mathcal{MP}} \Gamma' \cdot C'$

2. θ' resuelve Γ' y C'

3. $\Gamma' = \mathcal{I}' \vdash s_1 \sqsubseteq t_1, \ldots, s_n \sqsubseteq t_n$, de forma que, para toda $1 \leq i \leq n$, $s_i = t_i$ o s_i, t_i son variables diferentes.

Además, por el Lema 5.7.10, existe un unificador de máxima generelidad η de C' tal que $var(\Gamma') \cap dom(\eta) = \emptyset$, por lo que el conjunto $C' \cup \{s_1 \simeq t_1, \ldots, s_n \simeq t_n\}$ es unificable. En estas condiciones, podemos aplicar –a lo sumo n veces– la regla *end* para obtener una restricción satisfactible C'' tal que $\Gamma' \cdot C' \leadsto^*_{\mathcal{MP}} \mathcal{I}' \vdash s'_1 \sqsubseteq s'_1, \ldots, s'_n \sqsubseteq s'_n \cdot C''$. Toda solución de C'' es un $\mathcal{MP}$-unificador de Γ. ■

El cálculo $\mathcal{MP}$ es eficiente debido a que no eleva las soluciones ingenuas correspondientes al caso 3(c) (cfr. comentarios que preceden a la definición de $\mathcal{MP}$). En este sentido, hemos restringido el conjunto de soluciones que nuestro cálculo puede construir. La idea de limitar el conjunto de soluciones *interesantes* ya aparece en [DV 98] para la E-unificación rígida y destaca como una de las más prometedoras opciones encaminadas a la consecución de la terminación (véanse los comentarios anteriores referentes a la subsección *terminación frente a completitud*). Por ejemplo, el cálculo $\mathcal{MP}$ no puede construir la solución $[f(a)/x, f(b)/y]$ del *URPM*-problema $\{a \sqsubseteq b\} \vdash x \sqsubseteq y$. No obstante, podemos resolverlo de otras formas menos ingenuas: utilizando *end* para obtener la restricción $\{x \simeq y\}$ o utilizando *root* y *end* para obtener $\{x \simeq a, y \simeq b\}$.

Obsérvese que podemos tratar el caso 3(c) de otra forma más ineficiente si permitimos la adivinación aleatoria de contextos. Así, podríamos resolver este caso mediante la regla de *adivinación* que definimos como sigue:

$$\frac{\mathcal{I} \vdash s_1 \sqsubseteq t_1, \ldots, x \sqsubseteq y, \ldots, s_n \sqsubseteq t_n \cdot C}{\mathcal{I} \vdash s_1 \sqsubseteq t_1, .., u_1 \sqsubseteq v_1, .., u_q \sqsubseteq v_q, .., s_n \sqsubseteq t_n \cdot C \cup \{x \simeq f(u_1, .., u_q), y \simeq f(v_1, .., v_q)\}}$$

donde f es un símbolo de función de aridad q y u_i, v_i son variables nuevas, para toda $1 \leq i \leq q$.

En esta regla el símbolo de función f debe adivinarse por lo que, en realidad, estamos simulando los ineficientes axiomas de reflexividad funcional. Por otra parte, si añadimos la regla de *adivinación* al cálculo $\mathcal{MP}$, no resulta necesaria la reconstrucción de los contextos mediante la aplicación efectiva de los unificadores de máxima generalidad correspondientes a las reglas *end* e *imit1/2*. En estas condiciones, el nuevo cálculo es correcto y totalmente completo, en el sentido de que es capaz de elevar cualquier solución básica.

Por otra parte, a pesar de haber eliminado los axiomas de reflexividad, el cálculo $\mathcal{MP}$ no es terminante debido a que podemos usar la regla *root* sin control. Por ejemplo, podemos construir la siguiente secuencia infinita de $\mathcal{MP}$-pasos:

$$\begin{aligned}\{b \sqsubseteq c\} \vdash a \sqsubseteq b \cdot \emptyset \leadsto_{\mathcal{MP}} {} & \{b \sqsubseteq c\} \vdash a \sqsubseteq b, c \sqsubseteq b \cdot \emptyset \leadsto_{\mathcal{MP}} \\ & \{b \sqsubseteq c\} \vdash a \sqsubseteq b, c \sqsubseteq b, c \sqsubseteq b \cdot \emptyset \leadsto_{\mathcal{MP}} \ldots\end{aligned}$$

Además también podemos usar las reglas *imit1/2* para desarrollar secuencias infinitas de $\mathcal{MP}$-pasos, debido a que aplicamos el *umg* correspondiente:

$$\begin{aligned}\emptyset \vdash x \sqsubseteq f(x) \cdot \emptyset \leadsto_{\mathcal{MP}} {} & \emptyset \vdash x_1 \sqsubseteq f(x_1) \cdot \{x \simeq f(x_1)\} \leadsto_{\mathcal{MP}} \\ & \emptyset \vdash x_2 \sqsubseteq f(x_2) \cdot \{x \simeq f(x_1), x_1 \simeq f(x_2)\} \leadsto_{\mathcal{MP}} \ldots\end{aligned}$$

5.8 Tableaux con variables libres y unificación rígida preordenada y monótona simultánea

A continuación extendemos la noción de *URPM*-problema para considerar simultáneamente varios problemas, y así cerrar todas las ramas de un tableau de una sola vez.

Definición 5.8.1 *Un problema de Unificación Rígida Preordenada y Monótona Simultáneo (abreviadamente URPMS-problema) consiste en una secuencia finita de URPM-problemas.*

Como para los cálculos $\mathcal{P}$ y $\mathcal{SP}$, presentamos el cálculo $\mathcal{SMP}$ como la extensión natural de $\mathcal{MP}$ que considera la simultaneidad. Esto es, $\mathcal{SMP}$ transforma secuencias de *URPM*-problemas mediante la aplicación local de *root, dec, imit1, imit2* o *end* sobre alguno de sus *URPM*-problemas individuales. Las variables se comportan rígidamente, por lo que unimos todas las restricciones en un único conjunto y aplicamos globalmente los *umg* correspondientes a las reglas *end* e *imit1/2*.

Las reglas de $\mathcal{SMP}$ son extensiones simultáneas de las reglas de $\mathcal{SP}$; por ejemplo, la versión simultánea de *imit1* es:

$$\frac{\Gamma_1, \ldots, \mathcal{I} \vdash s_1 \sqsubseteq t_1 \ldots, x \sqsubseteq f(v_1, \ldots, v_q), \ldots s_n \sqsubseteq t_n, \ldots, \Gamma_p \cdot C}{(\ldots, \mathcal{I} \vdash s_1 \sqsubseteq t_1 \ldots, y_1 \sqsubseteq v_1, \ldots, y_q \sqsubseteq v_q, \ldots s_n \sqsubseteq t_n, \ldots)\tau \cdot C \cup \{x \simeq f(y_1, \ldots, y_q)\}}$$

donde y_i son variables nuevas y $\tau = [f(y_1, \ldots, y_q)/x]$.

Las condiciones adicionales para la aplicabilidad de estas reglas coinciden con las requeridas anteriormente para $\mathcal{MP}$. La relación $\leadsto_{\mathcal{SMP}}$ y el concepto de $\mathcal{SMP}$-unificador se definen como para $\mathcal{MP}$, es decir buscamos en el $\mathcal{SMP}$-árbol de derivación con raíz $\Gamma_1, \dots, \Gamma_n \cdot \emptyset$.

El cálculo $\mathcal{SMP}$ satisface las mismas propiedades que $\mathcal{MP}$, esto es, es correcto y completo, pero no terminante.

Teorema 5.8.2 (Corrección, completitud existencial) *Sea $\Gamma_1, \dots, \Gamma_n$ un URPMS-problema. Entonces:*

1. *Si θ es un $\mathcal{SMP}$-unificador de $\Gamma_1, \dots, \Gamma_n$ entonces θ resuelve Γ_i, para todo $1 \leq i \leq n$.*

2. *Si θ resuelve cada URPM-problema Γ_i, para todo $1 \leq i \leq n$, entonces existe un $\mathcal{SMP}$-unificador de $\Gamma_1, \dots, \Gamma_n$.*

Demostración. La prueba no es similar a la del Teorema 5.5.2, porque la aplicación local de las reglas *end* e *imit1/2* afecta al resto de los *URPM*-problemas. Sin embargo, podemos extender de forma sencilla las demostraciones de los Teoremas 5.7.7 y 5.7.11. ■

Ahora integramos el cálculo $\mathcal{SMP}$ en los tableaux con variables libres mediante la siguiente regla de cierre.

Definición 5.8.3 (Regla de cierre *URPMS*) *Sea $\mathcal{T}$ un tableau con variables libres y ramas $B_1, \dots, B_n$. Sea $\mathcal{I}_i$ la teoría que aparece en B_i y $s_i \not\sqsubseteq t_i$ una desinecuación que aparece en B_i, para todo $1 \leq i \leq n$. Entonces $\mathcal{T}$ está cerrado si existe un $\mathcal{SMP}$-unificador del URPMS-problema $\mathcal{I}_1 \vdash s_1 \sqsubseteq t_1, \dots, \mathcal{I}_n \vdash s_n \sqsubseteq t_n$.*

Sea $\mathcal{SMPT}$ el sistema de tableaux compuesto por las reglas $\alpha, \beta, \gamma', \delta'$ y la anterior regla de cierre *URPMS*. Como cuando no considerábamos monotonía, usamos este sistema expandiendo el tableau hasta que la regla de cierre *URPMS* se pueda aplicar.

La corrección de $\mathcal{SMPT}$ se deriva del Teorema 5.8.2. Para la completitud usamos el sistema básico presentado en la Sección 3, compuesto por las reglas $\alpha, \beta, \gamma, \delta$, *Tr*, *Ref* y *Mon*. Entonces elevamos cualquier tableau básico cerrado y usamos el Teorema 5.8.2 para concluir que $\mathcal{SMPT}$ también es completo.

Teorema 5.8.4 (Correción y completitud) *Un conjunto de sentencias Φ es insatisfactible en estructuras monótonas si y sólo si existe un $\mathcal{SMPT}$-tableau cerrado para Φ.*

Obsérvese que el cálculo $\mathcal{SMP}$ invocado por la regla de cierre *URPMS* no es terminante, por lo que no puede usarse como un mecanismo de decisión. Por este motivo, dicho cálculo puede no acabar sobre determinados *URPMS*-problemas correspondientes a algunas expansiones. En este sentido, la terminación es necesaria si queremos integrar adecuadamente la unificación rígida preordenada y monótona en los métodos de tableaux.

5.9 Conclusiones y trabajo futuro

En este capítulo hemos integrado los preórdenes monótonos en los tableaux mediante el uso de un cálculo de unificación rígida simultáneo [MG 00]. Para el caso en que no consideremos la monotonía, hemos definido el cálculo de unificación $\mathcal{P}$ que no sólo es correcto y completo, sino también terminante, debido a que las inecuaciones pueden eliminarse una vez que han sido usadas. Además hemos introducido técnicas de reescritura para definir el cálculo $\mathcal{P}1$ y conseguir una poda eficiente del espacio de búsqueda. Dicho cálculo es completo porque incluye nuevas reglas que aseguran la conmutación entre las inecuaciones orientadas.

Después de explicar los problemas que introduce la monotonía, hemos presentado el cálculo $\mathcal{MP}$ que resuelve problemas de unificación rígida preordenada y monótona. Aunque este cálculo no es terminante, es eficiente porque elimina el uso de los, ya conocidos a lo largo de este trabajo, axiomas de reflexividad funcional; ello se consigue sin más que prohibir la construcción de ciertas soluciones ingenuas. En este sentido hemos establecido las bases requeridas para conseguir terminación, algo que resulta esencial para lograr una integración adecuada de cualquier cálculo de unificación en los métodos de tableaux.

Actualmente estamos estudiando cómo restringir las soluciones básicas que se necesitan elevar, con objeto de evitar la reconstrucción de cualquier contexto. La intención es introducir después técnicas de reescritura para resolver el problema monótono mediante un cálculo terminante.

Bibliografía

[AH 92] L. Aceto, M. Hennessy. *Termination, deadlock and divergence.* J. ACM, 1992.

[Baum 92] P. Baumgartner. *An ordered theory resolution calculus.* LPAR'92, LNCS 624, 119–130, 1992.

[Baum 98] P. Baumgartner. *Theory Reasoning in Connection Calculi.* LNCS, Springer, 1998.

[Beth 55] E. W. Beth. *Semantic entailment and formal derivability.* Mededelingen der Kon. Ned. Akad. v. Wet. 18, 13, 1955. También en *Semantic entailment and formal derivability.* Kommentierte Auswahl zur Geschichte der modernen Logik, ed. K. Berka, L. Kreiser, 262–266, Berlin, Akademic-Verlag, 1986.

[Beth 56] E. W. Beth. *Semantic construction of intuitionistic logic.* Mededelingen der Kon. Ned. Akad. v. Wet. 19, 11, 1956.

[Beth 59] E. W. Beth. *The Foundations of Mathematics.* North-Holland, Amsterdam, 1959.

[BFP 92] P. Baumgartner, U. Furbach, U. Petermann. *A unified approach to theory reasoning.* Forschungsbericht 15/92, University of Koblenz, 1992.

[BG 94] L. Bachmair, H. Ganzinger. *Rewrite techniques for transitive relations.* Procs. LICS'94, 384–393, 1994.

[BG 95] L. Bachmair, H. Ganzinger. *Ordered Chaining Calculi for First-Order Theories of Binary Relations.* MPI-I-95-2-009, 1995.

[BGHK 92] B. Beckert, S. Gerberding, R. Hähnle, W. Kerning. *The tableau-based theorem prover 3TAP for multiple valued logics.* CADE'11, LNAI 607, 758–760, Springer Verlag, 1992.

[BGLS 92] L. Bachmair, H. Ganzinger, C. Lynch, W. Snyder. *Basic paramodulation and superposition.* CADE'11, LNAI 607, 462–476, Springer Verlag, 1992.

[BGLS 95] L. Bachmair, H. Ganzinger, C. Lynch, W. Snyder. *Basic paramodulation.* Information and Computation 121, 172–192, 1995.

[BGV 97] L. Bachmair, H. Ganzinger, A. Voronkov. *Elimination of Equality via Transformation with Ordering Constraints.* MPI-I-97-2-012, 1997.

[BH 92] B. Beckert, R. Hähnle. *An Improved Method for Adding Equality to Free Variable Semantic Tableaux.* CADE'11, LNAI 607, 507–521, Springer Verlag, 1992.

[BHPSS 92] C. Beierle, U. Hedstück, U. Pletat, P. H. Schmitt, J. Siekmann. *An order-sorted logic for knowledge representation systems.* Artificial Intelligence 55, 149–191, 1992.

[BKAHS 87] W. Bibel, F. Kurfess, K. Aspetsberger, P. Hintenaus, J. Schumann. *Parallel inference machines.* En *Future Parallel Computers*, P. Treleaven, M. Vanneschi Eds., Springer Berlin, 185–226, 1987.

[BKS 85] W. W. Bledsoe, K. Kunen, R. Shostak. *Completeness Results for Inequality Provers.* Artificial Intelligence 27, 255-288, 1985.

[BP 94] B. Beckert, J. Posegga. *leanTAP: lean, tableau-based theorem proving.* CADE'12, LNAI 814, Springer Verlag, 1994.

[Bowen 80] K. A. Bowen. *Programming with first order logic.* Tech. Rep. 6-80, Syracuse University, Syracuse, NY, Nov. 1980.

[Brand 75] D. Brand. *Proving Theorems with the Modification Method.* SIAM J. Computation 4(4), 412–430, 1975.

[Broda 80] K. Broda. *The relation between semantic tableaux and resolution theorem provers.* Tech. Rep. DOC 80/20, Imperial College of Science and Technology, London, Oct. 1980.

[Carn 87] W. A. Carnielli. *Systematization of finite many-valued logics through the method of tableaux.* J. Symbolic Logic 52, 2, 473–493, 1987.

[Carn 91] W. A. Carnielli. *On sequents and tableaux for many-valued logics.* J. Non-Classical Logic 8, 1, 59–76, 1991.

[Cohn 87] A. G. Cohn. *A more expressive formulation of many sorted logic.* Journal of Automated Reasoning 3, 113–200, 1987.

[Cohn 92] A. G. Cohn. *A many sorted logic with possibly empty sorts.* CADE'11, LNAI 607, 633–647, Springer Verlag, 1992.

[CTW 74] J. Cohen, L. Trilling, P. Wegner. *A nucleous of a theorem-prover described in ALGOL-68.* International J. of Computer and Information Sciences 3, 1, 1–31, 1974.

[DGHP 99] M. D'Agostino, D. Gabbay, R. Hähnlen, J. Posegga. *Handbook of tableau methods.* Kluwer Academic Publishers, 1999.

[DM 79] N. Dershowitz, A. Manna. *Proving termination with multisets orderings.* Com. of the ACM 22(8), 465–476, 1979.

[DV 94] A. Degtyarev, A. Voronkov. *Equality Elimination for Semantic Tableaux.* UPMAIL TR 90, 1994.

[DV 96] A. Degtyarev, A. Voronkov. *The undecidability of simultaneous rigid E-unification.* Theoretical Computer Science 166(1-2), 291-300, 1996.

[DV 98] A. Degtyarev, A. Voronkov. *What you always wanted to know about rigid E-unification.* Journal of Automated Reasoning 20(1), 47–80, 1998.

[EFT 94] H. Ebbinghaus, J. Flum, W. Thomas. *Mathematical Logic.* Springer Verlag, 1994.

[Fitt 69] M. Fitting. *Intiutionistic Logic Model Theory and Forcing.* North-Holland Publishing Co., Amsterdam, 1969.

[Fitt 96] M. Fitting. *First-Order Logic and Automated Theorem Proving.* Segunda edición. Springer, 1996.

[Floyd 62] R. W. Floyd. *Algorithm 97: Shortest path.* Communications of the ACM 5(6), pag. 345, 1962.

[Fris 91] A. M. Frisch. *The substitutional framework for sorted deduction: fundamental results on hybrid reasoning.* Artificial Intelligence 49, 161–198, 1991.

[Furb 94] U. Furbach. *Theory reasoning in first order calculi.* K. Luck, H. Marburger Eds, Proceedings of the Third Workshop on Information System and Artificial Intelligence, LNCS 777, 139–156. Springer Verlag, 1994.

[Gent 35] G. Gentzen. *Untersuchungen über das logische Schiessen.* Mathematische Zeitschrift 39, 176–210, 405–431, 1935. Traducido al inglés en [Szab 69].

[GLMN 96] A. Gavilanes, J. Leach, P. J. Martín, S. Nieva. *Reasoning with preorders and dynamic sorts using free variable tableaux.* AISMC-3. LNCS 1138, 365–379, Springer Verlag, 1996.

[GLMN 97] A. Gavilanes, J. Leach, P. J. Martín, S. Nieva. *Semantic tableaux for a logic with preorders and dynamic declarations.* Proc. TABLEAUX'97 (Position papers). CRIN 97-R-030, 7–12, 1997.

[GLN 96] A. Gavilanes, J. Leach, S. Nieva. *Free Variable Tableaux for a Many Sorted Logic with Preorders.* AMAST'96, LNCS 1101, 102–116, Springer Verlag, 1996.

[GM 92] J. A. Goguen, J. Meseguer. *Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations.* Theoretical Computer Science 105, 217–273, 1992.

[GNPS 90] J. Gallier, P. Naredran, D. Plaisted, W. Snyder. *Rigid E-unification: NP-completeness and applications to equational matings.* Information and Computation, 87(1/2), 129-195, 1990.

[GRS 87] J. H. Gallier, S. Raatz, W. Snyder. *Theorem proving using rigid E-unification: Equational matings.* LICS'87, 338–346, IEEE Computer Society Press, 1987.

[Her+ 86] O. Herzog, C. Rollinger, P. Schmitt, P. Steffens, R. Studer, B. Wesche, B. Bartsch-Spörl, F. Guenthner, C. Habel, S. Kanngießer, C. Rohrer. *LILOG-Linguistic and Logic Methods for the Computational Understanding of German.* LILOG-Report 1b, IBM Germany, 1986.

[Hines 92] L. M. Hines. *The Central Variable Strategy of Str+ve.* CADE'11, LNAI 607, 35-49, Springer Verlag, 1992.

[Hint 55] K.J.J. Hintikka. *Form and Content in Quantification Theory.* Acta Philosophica Fenica, 8, 7–55, 1955.

[Hint 61] J. Hintikka. *Modality and quantification.* Theoria 27, 110–128, 1961.

[Hint 62] J. Hintikka. *Knowledge and Belief.* Cornell Univeresity Press, 1962.

[HS 94] R. Hähnle, P. H. Schmitt. *The liberalized δ-rule in free variable semantic tableaux.* Journal of Automated Reasoning 13, 211-221, 1994.

[Inv 94] P. Inverardi. *Rewriting for preorders relations.* CTRS-94, LNCS 968, 223–234, Springer Verlag, 1995.

[JM 94] J. Jaffar, M. J. Maher. *Constraint logic programming: A survey.* Journal of Logic Programming 19/20, 503–582, 1994.

[Kang 57] S. G. Kanger. *Provability in logic.* Acta Universitatis Stockholmiensis, Stockholm Studies in Philosophi, 1. Almqvist and Wiksell, Stockholm, 1957.

[Kang 63] S. G. Kanger. *A simplified proof method for elementary logic.* Siekmann, Wrightson Eds., Automation of Reasoning. Classical Papers on Computer Logic, vol. 1, 364–371. Springer Verlag, 1963.

[Kogel 95] E. Kogel. *Rigid E-unification simplified.* TABLEAUX'95, LNCS 918, 17–30, Springer Verlag, 1995.

[Krip 59] S. Kripke. *A completenes theorem in modal logic.* J. Symbolic Logic 24, 1–14, 1959.

[LA 93] J. Levy, J. Agustí. *Bi-rewriting, a term rewriting technique for monotonic order relations.* RTA'93, LNCS 690, 17–31, Springer Verlag, 1993.

[LA 96] J. Levy, J. Agustí. *Bi-rewrite systems.* J. of Symbolic Computation 22, 279–314, 1996.

[Levy 94] J. Levy. *The calculus of refinements, a formal specification model based on inclusions*. PhD thesis, Departament de Llenguatges i Sistemes Informàtics, Universidad Politécnica de Cataluña, 1994.

[LMM 86] J.L. Lassez, M.J. Maher, K. Marriot. *Unification reviseted*. LNCS 306, 67–113, Springer Verlag, 1986.

[Mart 96] P.J. Martín de la Calle. *Tableaux Semánticos para una Lógica con Preórdenes y Géneros Dinámicos*. Trabajo de Tercer Ciclo, DIA, 1996.

[Masl 67] S. Maslov. *An invertible sequential variant of constructive predicate calculus*. Zapiski Nauchnyh Seminarov LOMI, 4, 1967.

[Matu 63] V. A. Matulis. *On variants of predicate calculus with the unique deduction tree*. Soviet Mathematical Doklady 148, 768–770, 1963.

[MG 96] P.J. Martín, A. Gavilanes. *Semantic Tableaux for a Logic with Preorders and Dynamic Sorts*. Proc. Appia-Gulp-Prode, 139–151, 1996.

[MG 99] P.J. Martín, A. Gavilanes. *Simultaneous Sorted Unification for Free Variable Tableaux: An Elegant Calculus*. TABLEAUX'99, Position Paper, TR 99-1, Department of Computer Science, University at Albany, 1999.

[MG 00] P.J. Martín, A. Gavilanes. *Monotonic preorders for free variable tableaux*. Aparecerá en Proc. TABLEAUX'2000, 2000.

[MGL 98] P. J. Martín, A. Gavilanes, J. Leach. *Free Variable Tableaux for a Logic with Term Declarations*. TABLEAUX'98, LNAI 1397, 202-216, Springer Verlag, 1998.

[MGL 00] P. J. Martín, A. Gavilanes, J. Leach. *Tableau Methods for a Logic with Term Declarations*. J. of Symbolic Computation 29, 343–372, 2000.

[MW 90] Z. Manna, R. Waldinger. *The Logical Basis for Computer Programming*. Addison-Wesley, 1990. 2 vols.

[MW 93] Z. Manna, R. Waldinger. *The Deductive Foundations of Computer Programming*. Addison-Wesley, 1993.

[Nieu 93] R. Nieuwenhuis. *Simple LPO constrain solving methods*. Information Processing Letters 47, 65–69, 1993.

[NR 92] R. Nieuwenhuis, A. Rubio. *Basic superposition is complete*. ESOP'92, LNCS 582, 371-389, Springer Verlag, 1992.

[NR 95] R. Nieuwenhuis, A. Rubio. *Theorem Proving with Ordering and Equality Constrained Clauses*. J. Symbolic Computation 19, 321–351, 1995.

[OS 88] F. Oppacher, E. Suen. *HARP: A tableau-based theorem prover*. J. Automated Reasoning 4, 69–100, 1988.

[Rob 65] J.A. Robinson. *A machine-oriented logic based on the resolution principle.* Journal of the ACM 12, 13–41, 1965.

[Rusi 91] M. Rusinowitch. *Theorem proving with resolution and superposition: An extension to the Knuth and Bendix procedure as a complete set of inference rules.* J. of Symbolic Computation 11, 1991.

[Schm 87] P. H. Schmitt. *The THOT theorem prover.* Tech. Rep. TR-87.09.007, IBM Heidelberg Scientific Center, 1987.

[Smul 68] R. Smullyan. *First-Order Logic.* New York, Springer, 1968.

[SS 89] M. Schmidt-Schauss. *Computational Aspects of an Order Sorted Logic with Term Declarations.* LNAI 395, Springer, 1989.

[Such 74] W. Suchon. *La méthode de Smullyan de construire le calcul n-valent des propositions de Lukasiewicz avec implication et négation.* Reports on Mathematical Logic, Universities of Cracow and Katowice 2, 37–42, 1974.

[Surma 84] S. J. Surma. *An algorithm for axiomatizing every finite logic.* Computer Science and Multiple-Valued Logics, D. C. Rine Ed. North-Holland, Amsterdam, 143–149, 1977.

[SW 89] P. H. Schmitt, W. Wernecke. *Tableau calculus for order sorted logic.* Proc. Workshop on Sorts and Types in Artificial Intelligence, K. H. Bläsius, U. Hedstück, C. Rollinger Eds., LNAI 418, 49–60, Springer Verlag, 1989.

[Szab 69] M. E. Szabo, Ed. *The Collected Papers of Gerhard Gentzen.* North-Holland, Amsterdam, 1969.

[Walk 90] D. J. Walker. *Bisimulation and divergence.* Information and Computation 85, 202–242, 1990.

[Walt 87] C. Walther. *A Many-sorted Calculus based on Resolution and Paramodulation.* Research Notes in Artificial Intelligence. Pitman, 1987.

[Wang 60] H. Wang. *Toward mechanical mathematics.* IBM Journal for Research and Development 4, 2–22, 1960.

[Weid 91] C. Weidenbach. *A sorted logic using dynamic sorts.* MPI-I-91-218. 1991.

[Weid 93] C. Weidenbach. *Extending the resolution method with sorts.* Proc. of 13th International Joint Conference on artificial Intelligence, IJCAI'93, Morgan Kaufmann, 1993.

[Weid 95] C. Weidenbach. *First-order tableaux with sorts.* J. of the Interest Group in Pure and Applied Logics 3(6), 887–907, 1995.