

UNIVERSIDAD NACIONAL DE CÓRDOBA  
FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y  
COMPUTACIÓN



**Cálculo de tableaux para fórmulas  
elementales en lógicas de  
separación**

Andrés Román Saravia  
Director: Raul Fervari

Licenciatura en Ciencias de la Computación  
2020



Esta obra está bajo una licencia Creative Commons "Reconocimiento-CompartirIgual 4.0  
Internacional".



*A todos los que me preceden y a todos los que me sucedan.*



# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Lógicas modales dinámicas . . . . .	7
1.2. Modal Separation Logics y fórmulas elementales . . . . .	8
1.3. Cálculos de tableaux . . . . .	8
1.4. Objetivos . . . . .	9
1.5. Organización . . . . .	9
<b>2. MSL(*, <math>\diamond</math>) y fórmulas elementales</b>	<b>11</b>
2.1. Introducción . . . . .	11
2.2. Sintaxis y semántica de MSL(*, $\diamond$ ) . . . . .	11
2.3. Sintaxis y semántica de las fórmulas elementales . . . . .	13
2.4. Equivalencia . . . . .	14
<b>3. Cálculo de Tableaux para fórmulas elementales</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Reglas . . . . .	17
3.3. Ramas cerradas, abiertas y saturadas . . . . .	19
3.4. Completitud . . . . .	19
3.5. Corrección . . . . .	23
<b>4. Conclusiones y trabajos futuros</b>	<b>25</b>
4.1. Recapitulación . . . . .	25
4.2. Trabajos futuros . . . . .	25
4.3. Pensamientos personales y agradecimientos . . . . .	26



# Capítulo 1

## Introducción

En ciencias de la computación, muchas situaciones pueden ser modeladas por medio de estructuras relacionales o requieren el uso de estas: un grafo, una base de datos, el flujo de ejecución de un programa, el manejo de memoria dinámica o simplemente las relaciones entre varios elementos. Por este motivo, comprender y razonar con lenguajes que describan este tipo de estructuras resulta ser muy importante, ya que por medio de estos podemos extraer nueva información al ser utilizados en situaciones como las mencionadas anteriormente.

Uno de los formalismos más utilizados para razonar sobre estas estructuras relacionales son las *lógicas modales* [1,2]. Las lógicas modales son una familia de lenguajes que extienden la lógica proposicional con nuevos operadores que describen diferentes modos de verdad, usualmente llamados *modalidades*. Los mismos incluyen temporalidad [3], creencia [4], conocimientos [5] y obligaciones [6], solo por mencionar algunas. Una de las ventajas de las lógicas modales es que en general proveen un buen balance entre la expresividad y el comportamiento computacional. Esto ha dado lugar a que los lenguajes modales sean muy populares en la práctica y muy utilizados en su aplicación para realizar tareas de razonamiento, es decir, aquellas tareas que nos permiten inferir propiedades a partir de fórmulas expresadas en un lenguaje determinado. Un ejemplo de esto es el caso de la lógica temporal lineal (LTL), que ha resultado ser importante en el proceso de model checking y verificación de software ya que permite expresar fácilmente las propiedades que deben cumplir los sistemas reactivos [7,8].

Por otro lado, las lógicas modales tienen otra ventaja. Nos brindan la posibilidad de definir o elegir el operador más adecuado a nuestras necesidades y estudiarlo utilizando las herramientas clásicas de la lógica modal. Es por este motivo que elegimos la “*perspectiva modal*” en este trabajo: estudiaremos desde una perspectiva modal, los procedimientos para una variante de lógica de separación [9].

### 1.1. Lógicas modales dinámicas

Las lógicas modales han sido utilizadas para representar diferentes conceptos. En particular, un concepto que nos resulta interesante estudiar en ciencias de la computación, es el de cambio o evolución de nuestros objetos de estudio. Ejemplos de esto son los cambios de estado de un programa, actualizaciones de valores en una base de datos o la evolución de la memoria. Si utilizamos lógicas modales para razonar sobre este tipo de situaciones, una alternativa es modelar el escenario de todos los posibles cambios de nuestros objetos y sistemas, y luego utilizar operadores modales que exploren estas situaciones. Sin embargo, esto resulta inconveniente en la práctica ya que el espacio de posibilidades puede resultar demasiado grande. Es por eso que recientemente se ha estudiado en profundidad una nueva familia de *lógicas modales dinámicas*, que permiten modificar la estructura subyacente a medida que se evalúa una fórmula. Ejemplos de estos son los *frames reactivos* [10], las *lógicas modales de sabotaje* [11–15] y las *lógicas de anuncios públicos* [16]. Además, otros tipos de operaciones que actualizan la relación de accesibilidad del modelo han sido investigadas desde un punto de vista abstracto en [17–23], mientras que conexiones con lógicas epistémicas han sido establecidas en [24–27]. De la misma manera, es posible también estudiar otras lógicas dinámicas desde un punto de vista modal.

Un caso particular de lógicas dinámicas particularmente interesantes son las *lógicas de separación* (*SL*, por sus siglas en inglés), un conjunto de lenguajes dinámicos que surgieron como una extensión de la lógica de Hoare para razonar sobre programas con estructuras de datos mutables [9]. Si bien las expresiones de *SL* se interpretan sobre una abstracción del heap denominada estado abstracto, que esencialmente modela las celdas de memoria y sus contenidos, dichas abstracciones también pueden ser representadas como modelos relacionales de lógica modal, donde la relación es finita y funcional. Esta reinterpretación permitió expresar tanto propiedades modales como de separación y dio origen a una nueva familia de lenguajes denominada Modal Separation Logics (*MSL*) [28, 29].

En [29] se investigan diferentes combinaciones de operadores modales y de separación de manera incremental, y se realiza un análisis minucioso del impacto de la inclusión de cada una de ellos en la complejidad computacional de sus tareas de inferencia. Para ello se introduce primero el lenguaje básico, que contiene el operador modal  $\diamond$  junto a la constante **emp** y el operador de conjunción de separación  $*$  ( $\text{MSL}(*, \diamond)$ ), provenientes de las *SL*. Se demostró que el problema de model checking para este lenguaje es polinomial, y que el problema de satisfacibilidad es NP-completo. Análogamente, se demuestra lo mismo para el fragmento donde se incluye el operador de diferencia  $\langle \neq \rangle$  [30] en vez de  $\diamond$  ( $\text{MSL}(*, \langle \neq \rangle)$ ). Sin embargo, combinando ambos fragmentos o incluyendo los operadores restantes como  $\neg$ , la complejidad se incrementa o, más aún, termina siendo indecidible: para el fragmento  $\text{MSL}(*, \diamond, \langle \neq \rangle)$  el problema de satisfacibilidad es TOWER-completo, mientras que para el lenguaje completo incluyendo  $\neg$ , dicho problema es indecidible.

## 1.2. Modal Separation Logics y fórmulas elementales

Como se mencionó anteriormente, las lógicas de separación comenzaron recientemente a estudiarse desde el punto de vista modal. En [29] se estudia la complejidad computacional de diferentes combinaciones de operadores modales y operadores de separación, interpretados sobre modelos relacionales. Esto nos da una perspectiva acerca de cuáles fragmentos son tratables computacionalmente a la hora de diseñar sistemas de prueba para estas lógicas. En [31] se comienza con el estudio de sistemas de prueba para dos fragmentos de *MSL*. El primero,  $\text{MSL}(*, \diamond)$ , combina los operadores  $*$  y **emp** de las lógicas de separación con el  $\diamond$  clásico de la lógica modal. El segundo,  $\text{MSL}(*, \langle \neq \rangle)$ , considera el operador de diferencia  $\langle \neq \rangle$  en vez de  $\diamond$ . Al considerar el operador dinámico  $*$ , diseñar sistemas de prueba se vuelve particularmente complejo. Es por eso que se utiliza una técnica similar a los axiomas de reducción utilizados en lógicas dinámicas epistémicas [32–34]. Básicamente, la técnica consiste en transformar toda fórmula del lenguaje dinámico en una fórmula sin el operador dinámico, y luego axiomatizar el fragmento sin dicho operador. De esta manera obtenemos una axiomatización para el lenguaje completo. Para ello en el caso de las *MSL*, se definen las *fórmulas elementales*.

Las fórmulas elementales son un conjunto de fórmulas que describen propiedades básicas sobre la estructura de los modelos de  $\text{MSL}(*, \diamond)$  y  $\text{MSL}(*, \langle \neq \rangle)$ , con el objetivo de crear una axiomatización de estos fragmentos. Si bien en dicho trabajo se establece que las fórmulas elementales son un fragmento de *MSL*, se demuestra que toda fórmula en  $\text{MSL}(*, \diamond)$  es lógicamente equivalente a una disyunción de fórmulas elementales de una determinada forma. Este resultado no solo nos permite hablar de fórmulas elementales y fórmulas de  $\text{MSL}(*, \diamond)$  indistintivamente sino también de analizar la satisfacibilidad de una fórmula en  $\text{MSL}(*, \diamond)$  con solo ver su equivalente en fórmulas elementales. En [31] se presenta un sistema a la Hilbert para razonar sobre  $\text{MSL}(*, \diamond)$  mediante fórmulas elementales. Sin embargo, este tipo de sistemas no suelen ser usables en la práctica, y menos aún, apropiados para ser implementados. Es por eso que en este trabajo nos enfocaremos en el estudio de sistemas de tableaux [35].

## 1.3. Cálculos de tableaux

En términos generales, un *algoritmo de tableaux* es un procedimiento en el que se decide la satisfacibilidad de una fórmula dada utilizando la satisfacibilidad de sus subfórmulas, por medio de reglas de inferencia previamente establecidas. Dicho procedimiento toma una fórmula como entrada, decide si es satisfacible o no y, en caso afirmativo, devuelve un modelo el cual

satisface dicha fórmula.

Esencialmente, el algoritmo utiliza una estructura de datos con forma de árbol (a la cual llamamos tableaux), el cual tiene como raíz la fórmula dada en la entrada y como ramas las subfórmulas de la misma a medida que el algoritmo va aplicando las reglas de inferencia. Dependiendo del tipo de lógica con la que se esté tratando, el tableaux puede tener características adicionales. Por ejemplo, para la lógica modal usualmente se usan etiquetas, las cuales agregan la posibilidad de modelar más fácilmente el razonamiento local sobre modelos relacionales, propio de la lógica modal.

## 1.4. Objetivos

Teniendo en cuenta lo mencionado anteriormente, el objetivo central de este trabajo consistirá en definir un cálculo de tableaux para lenguajes modales dinámicos, en particular para  $MSL(*, \diamond)$ . Para ello, introduciremos brevemente los conceptos básicos de  $MSL(*, \diamond)$  y las *fórmulas elementales* y, basándonos en el resultado demostrado en [31], desarrollaremos un *cálculo de tableaux etiquetado* basado exclusivamente en *fórmulas elementales*. Esto nos permitirá lidiar con un fragmento más manejable del lenguaje original. Una vez definido este cálculo, demostraremos que es completo y correcto.

## 1.5. Organización

El trabajo constará de dos partes. En la primera (sección 2) nos ocuparemos de introducir los conceptos principales referidos a la lógica  $MSL(*, \diamond)$  y las fórmulas elementales, tanto su definición sintáctica y como semántica. Además de ello, explicaremos de qué manera ambas lógicas tienen el mismo poder expresivo.

En la segunda (sección 3) utilizaremos los resultados de la sección anterior para definir el procedimiento de tableaux para un cierto tipo de fórmulas elementales. En esta incluiremos las reglas de derivación, los conceptos de rama saturada, abierta y cerrada y las demostraciones de completitud y corrección. Finalmente, en la sección 4 discutiremos el trabajo realizado y las posibles líneas de trabajo futuro.



# Capítulo 2

## MSL( $*$ , $\diamond$ ) y fórmulas elementales

### 2.1. Introducción

En esta sección describiremos tanto sintáctica como semánticamente las dos componentes fundamentales de este trabajo: la primera son las lógicas modales de separación (MSL por sus siglas en inglés), que surgen de la combinación de dos familias de lógicas: (1) las lógicas modales [1, 2] que son una extensión de la lógica proposicional; y (2) las lógicas de separación que son una extensión de la lógica de Hoare que permiten razonar sobre programas de bajo nivel que utilizan estructuras de datos mutables [9]. Específicamente trabajaremos sobre la lógica modal de separación que denotaremos como  $\text{MSL}(*, \diamond)$ , el cual tiene los operadores  $\diamond$  y  $*$ , y la constante  $\text{emp}$ .

Por otro lado, la segunda parte importante son las fórmulas elementales (o como se define en [31], *core formulae*). Estas fórmulas describen las propiedades esenciales que tienen los modelos de  $\text{MSL}(*, \diamond)$ . Si bien estas últimas están definidas como una extensión de  $\text{MSL}(*, \diamond)$ , por lo demostrado en [31], sabemos que las mismas son definibles en  $\text{MSL}(*, \diamond)$ . Más aún, toda fórmula en  $\text{MSL}(*, \diamond)$  es equivalente a una combinación Booleana de fórmulas elementales. Esto último nos conducirá a la parte central del trabajo en la próxima sección.

### 2.2. Sintaxis y semántica de $\text{MSL}(*, \diamond)$

Sea  $\text{PROP} = \{p, q, \dots\}$  un conjunto infinito enumerable de símbolos proposicionales. Las fórmulas de  $\text{MSL}(*, \diamond)$  están definidas por la siguiente gramática:

$$\phi ::= \top \mid p \mid \text{emp} \mid \neg\phi \mid \phi \wedge \phi \mid \diamond\phi \mid \phi * \phi,$$

donde  $p \in \text{PROP}$ . Además tenemos que:

- $\perp \stackrel{\text{def}}{=} \neg\top$ ;
- $\phi_1 \vee \phi_2 \stackrel{\text{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2)$ ;
- $\Box\phi \stackrel{\text{def}}{=} \neg\diamond\neg\phi$ ;
- $\phi_1 \Rightarrow \phi_2 \stackrel{\text{def}}{=} \neg\phi_1 \vee \phi_2$ .

Un *modelo* es una tupla  $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$  tal que:

- el conjunto de los números naturales  $\mathbb{N}$  es el conjunto de *mundos*,
- $\mathfrak{R} \subseteq \mathbb{N} \times \mathbb{N}$  es finita y determinística ( $(l, l') \in \mathfrak{R}$  y  $(l, l'') \in \mathfrak{R}$  implica  $l' = l''$ ),
- $\mathfrak{V} : \text{PROP} \rightarrow \mathcal{P}(\mathbb{N})$  es una valuación.

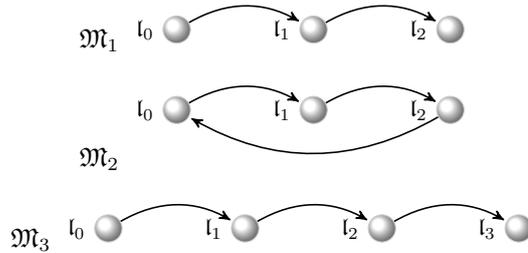
Sea  $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$  y  $l \in \mathbb{N}$ , la relación de satisfacción  $\models$  está definida como:

$\mathfrak{M}, l \models \top$	$\stackrel{\text{def}}{\iff}$	siempre
$\mathfrak{M}, l \models p$	$\stackrel{\text{def}}{\iff}$	$l \in \mathfrak{V}(p)$
$\mathfrak{M}, l \models \text{emp}$	$\stackrel{\text{def}}{\iff}$	$\mathfrak{R} = \emptyset$
$\mathfrak{M}, l \models \neg\phi$	$\stackrel{\text{def}}{\iff}$	$\mathfrak{M}, l \not\models \phi$
$\mathfrak{M}, l \models \phi_1 \wedge \phi_2$	$\stackrel{\text{def}}{\iff}$	$\mathfrak{M}, l \models \phi_1$ y $\mathfrak{M}, l \models \phi_2$
$\mathfrak{M}, l \models \diamond\phi$	$\stackrel{\text{def}}{\iff}$	$\mathfrak{M}, l' \models \phi$ , para algún $l' \in \mathbb{N}$ tal que $(l, l') \in \mathfrak{R}$
$\mathfrak{M}, l \models \phi_1 * \phi_2$	$\stackrel{\text{def}}{\iff}$	$\langle \mathbb{N}, \mathfrak{R}_1, \mathfrak{V} \rangle, l \models \phi_1$ y $\langle \mathbb{N}, \mathfrak{R}_2, \mathfrak{V} \rangle, l \models \phi_2$ , para alguna partición $\{\mathfrak{R}_1, \mathfrak{R}_2\}$ de $\mathfrak{R}$ .

Sea  $\phi$  una fórmula en  $\text{MSL}(*, \diamond)$ , diremos que  $\phi$  es satisfacible si existe un modelo  $\mathfrak{M}$  y un mundo  $l \in \mathbb{N}$  tal que  $\mathfrak{M}, l \models \phi$ . Una fórmula  $\phi$  es insatisfacible si no es satisfacible. Además, dada una fórmula  $\phi$  y un modelo  $\mathfrak{M}$ , si para todo mundo  $l$  se tiene que  $\mathfrak{M}, l \models \phi$ , escribiremos  $\mathfrak{M} \models \phi$ . Diremos que  $\phi$  es válida si para todo modelo  $\mathfrak{M}$  tenemos que  $\mathfrak{M} \models \phi$ . Finalmente, definimos  $\mathfrak{R}(l) \stackrel{\text{def}}{=} \{l' \in \mathbb{N} \mid (l, l') \in \mathfrak{R}\}$ . A continuación introduciremos una serie de ejemplos que servirán para ilustrar el poder expresivo de  $\text{MSL}(*, \diamond)$ .

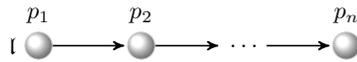
Debido a la definición de los modelos, la fórmula  $\diamond\top * \diamond\top$  es insatisfacible en  $\text{MSL}(*, \diamond)$ . Esto se debe a que la relación  $\mathfrak{R}$  es, por definición, una relación funcional, por lo tanto no es posible separar el modelo en dos partes disjuntas tal que en cada una de ellas, el mundo actual tenga un sucesor (recordar que el operador  $*$  mantiene el mismo punto de evaluación en cada una de las partes).

Consideremos ahora la fórmula  $\diamond\diamond\neg\diamond\top * \top$ . La misma es satisfacible en los siguientes modelos, en el mundo  $l_0$  correspondiente:



Esto se debe a que podemos particionar el modelo en dos partes, una que contenga dos aristas, que cumplirán con la parte  $\diamond\diamond\neg\diamond\top$  y otra que cumplirá trivialmente  $\top$ .

Consideremos ahora la fórmula  $(p_1 \wedge \diamond(p_2 \wedge \diamond(p_3 \wedge \dots \diamond(p_n \wedge \square \perp) \dots)))$  a la cual denotaremos como  $\text{truncated}_n$ . Dicha fórmula es satisfacible por modelos de la siguiente forma desde el mundo  $l$ :



La misma establece que existe un camino de tamaño  $(n - 1)$ . Notar que el determinismo en la relación de accesibilidad, fuerza a que el camino sea entre  $n$  mundos distintos.

La expresividad de  $\text{MSL}(*, \diamond)$  incluso puede definir la fórmula  $\text{loop}_2$ , que interpretada en el mundo  $l$ , establece un modelo que contiene exactamente un ciclo de longitud dos que incluye a  $l$ :

$$(\neg\text{emp} * \neg\text{emp}) \wedge \neg(\neg\text{emp} * \neg\text{emp} * \neg\text{emp}) \wedge \diamond\diamond\top \wedge \neg(\neg\text{emp} * \diamond\diamond\top) \wedge \neg\diamond(\neg\text{emp} * \diamond\diamond\top)$$



La primera parte de la expresión,  $(\neg\text{emp} * \neg\text{emp}) \wedge \neg(\neg\text{emp} * \neg\text{emp} * \neg\text{emp})$ , describe que el modelo tiene al menos dos aristas y no más de tres, lo cual nos dice que hay exactamente dos aristas. La segunda parte,  $\diamond\diamond\top$ , describe que desde el mundo  $l$  podemos movernos a través de la relación al menos tres veces. Esto establece que dicho mundo tiene un sucesor a distancia

3, lo cual significa que existe un ciclo, aunque éste aún puede ser de tamaño 1. La tercera parte,  $\neg(\neg\text{emp} * \diamond\diamond\top)$ , establece que si sacamos una o más aristas, cualquiera sean, la propiedad de la segunda parte no se cumple. Por lo tanto, la posibilidad de que  $\mathfrak{l}$  esté en un ciclo de longitud uno queda descartada pues si existiera dicho ciclo, entonces no importaría sacar la arista restante. Esto nos ayuda a deducir que, a través de la relación, nos podemos mover del mundo  $\mathfrak{l}$  a otro  $\mathfrak{l}'$  distinto. Sin embargo,  $\mathfrak{l}'$  puede tener una arista hacia sí mismo o hacia otro mundo  $\mathfrak{l}''$  y no hacia  $\mathfrak{l}$ . Pero la posibilidad de un tercer mundo  $\mathfrak{l}''$  no sería factible ya que solo hay dos aristas y según la segunda parte, tenemos que desde  $\mathfrak{l}$  podemos movernos a través de la relación al menos tres veces. Ahora bien, para evitar que  $\mathfrak{l}''$  tenga un ciclo en sí mismo, tenemos la cuarta parte de la expresión,  $\neg\diamond(\neg\text{emp} * \diamond\diamond\top)$ , que establece que si nos movemos al mundo  $\mathfrak{l}'$  y quitamos al menos una de las dos aristas, cualquiera sea, no podremos movernos a través de la relación al menos unas tres veces. Lo cual nos permite decir que la arista que va desde  $\mathfrak{l}'$  justamente va a  $\mathfrak{l}$ , puesto que si suponemos lo contrario entonces podríamos descartar una arista (en este caso,  $\mathfrak{R}\mathfrak{l}'$ ).

## 2.3. Sintaxis y semántica de las fórmulas elementales

Las fórmulas elementales son un tipo de fórmulas en  $\text{MSL}(*, \diamond)$  introducidas en [31] que si bien son simples, describen propiedades esenciales de los modelos. Más aún, capturan exactamente todo aquello que es expresable en  $\text{MSL}(*, \diamond)$ . Dichas fórmulas elementales están divididas en dos familias: una denominada *fórmulas size* que expresan el tamaño del modelo (es decir el número de ejes por la relación de accesibilidad entre mundos) y otra llamada *fórmulas grafo* que describen la forma que tiene el modelo desde el mundo actual. Como se establecerá más abajo, estas fórmulas son lo suficientemente expresivas para capturar  $\text{MSL}(*, \diamond)$ .

Para las *fórmulas size* se introducen las expresiones de la forma  $\text{size} \geq \beta$  que son verdaderas sii  $\mathfrak{R}$  tiene al menos  $\beta$  pares ( $\beta \in \mathbb{N}_0$ ). Un *literal size* es una fórmula de la forma  $\text{size} \geq \beta$  o  $\neg\text{size} \geq \beta$  (o equivalentemente  $\text{size} < \beta$ ). Cada combinación Booleana de literales size es una *fórmula size*. La expresión  $\text{size} = \beta$  es una abreviación de  $\text{size} \geq \beta \wedge \neg\text{size} \geq \beta + 1$ . Por otro lado, cada *fórmula grafo* es una expresión derivada del elemento  $\mathcal{G}$  no terminal de la siguiente gramática:

$$\ell := \top \mid \perp \mid p \mid \neg p \quad Q := \ell \mid Q \wedge Q \quad \mathcal{G} := \mid Q_1, \dots, Q_n \rangle \mid \mid Q_1, \dots, Q_n \mid \mid \mid Q_1, \dots, \overleftarrow{Q_i, \dots, Q_n} \mid,$$

donde  $p \in \text{PROP}$ , y  $\mathcal{G}$  debe contener al menos una conjunción  $Q$ . Abusando de la notación, diremos que  $Q$  es una conjunción de literales (a pesar de poder tener símbolos  $\perp$  y  $\top$ ). Dado que estaremos trabajando con relaciones finitas y débilmente funcionales (o determinísticas), las fórmulas grafo representan caminos que satisfacen una conjunción de literales  $Q$  en cada posición. Una fórmula de la forma  $\mid Q_1, \dots, Q_n \rangle$  expresa que hay un camino de longitud  $n$ , en el cual todos los mundos son distintos, y que luego del  $(n + 1)$ -ésimo mundo, el camino puede continuar o no. A este tipo de fórmulas las llamaremos *fórmulas continuadas*. La fórmula  $\mid Q_1, \dots, Q_n \mid$  establece que hay un camino de longitud  $n - 1$ , en el cual todos los mundos son distintos y que el último de estos no tiene sucesor. A este tipo de fórmulas las llamaremos *fórmulas truncadas*. Finalmente, la fórmula de la forma  $\mid Q_1, \dots, \overleftarrow{Q_i, \dots, Q_n} \mid$  expresa que hay un camino de longitud  $n - 1$  con mundos todos distintos, y que hay un ciclo que incluye desde el  $i$ -ésimo mundo hasta el  $n$ -ésimo. A este tipo de fórmulas las llamaremos *fórmulas cíclicas*. En la Figura 2.1 ilustramos las clases de modelos especificadas por cada tipo de fórmula grafo.

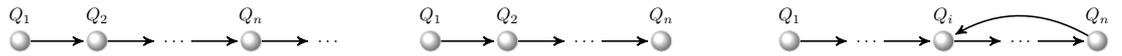


Figura 2.1: Modelos para las fórmulas grafo  $\mid Q_1, \dots, Q_n \rangle$ ,  $\mid Q_1, \dots, Q_n \mid$  y  $\mid Q_1, \dots, \overleftarrow{Q_i, \dots, Q_n} \mid$ , respectivamente.

Escribiremos  $\mid Q_1, \dots, Q_n ? \rangle$  para referirnos a cualquier tipo de fórmula grafo y  $\#(\mid Q_1, \dots, Q_n ? \rangle)$

para denotar el *tamaño del grafo* de  $|Q_1, \dots, Q_n\rangle$  definido de la siguiente forma:

$$\#(|Q_1, \dots, Q_n\rangle) \stackrel{\text{def}}{=} n \quad \#(|Q_1, \dots, Q_n\rangle) \stackrel{\text{def}}{=} n-1 \quad \#(|Q_1, \dots, \overline{Q_i, \dots, Q_n}\rangle) \stackrel{\text{def}}{=} n.$$

Teniendo en cuenta lo anterior, definiremos la semántica de las fórmulas elementales. Dado un modelo  $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{W} \rangle$  con las mismas características definidas en la subsección anterior,  $\mathfrak{l} \in \mathbb{N}$ , extenderemos la relación  $\models$  a las fórmulas elementales de la siguiente forma:

$$\begin{aligned} \mathfrak{M}, \mathfrak{l} \models \text{size} \geq \beta & \stackrel{\text{def}}{\iff} \text{card}(\mathfrak{R}) \geq \beta \\ \mathfrak{M}, \mathfrak{l} \models |Q_1, \dots, Q_n\rangle & \stackrel{\text{def}}{\iff} \text{existen } \mathfrak{l}_1, \dots, \mathfrak{l}_{n+1} \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_1 \mathfrak{R} \mathfrak{l}_2 \mathfrak{R} \dots \mathfrak{R} \mathfrak{l}_{n+1}, \\ & \text{y para todo } j \in [1, n], \mathfrak{M}, \mathfrak{l}_j \models Q_j \\ \mathfrak{M}, \mathfrak{l} \models |Q_1, \dots, Q_n] & \stackrel{\text{def}}{\iff} \text{existen } \mathfrak{l}_1, \dots, \mathfrak{l}_n \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_1 \mathfrak{R} \mathfrak{l}_2 \mathfrak{R} \dots \mathfrak{R} \mathfrak{l}_n, \\ & \mathfrak{R}(\mathfrak{l}_n) = \emptyset \text{ y para todo } j \in [1, n], \mathfrak{M}, \mathfrak{l}_j \models Q_j \\ \mathfrak{M}, \mathfrak{l} \models |Q_1, \dots, \overline{Q_i, \dots, Q_n}] & \stackrel{\text{def}}{\iff} \text{existen } \mathfrak{l}_1, \dots, \mathfrak{l}_n \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_1 \mathfrak{R} \mathfrak{l}_2 \mathfrak{R} \dots \mathfrak{R} \mathfrak{l}_n \mathfrak{R} \mathfrak{l}_i \\ & \text{y para todo } j \in [1, n], \mathfrak{M}, \mathfrak{l}_j \models Q_j \end{aligned}$$

Dado que las fórmulas **size** son independientes del mundo en el que se encuentren en el modelo, podemos decir que  $\mathfrak{M}, \mathfrak{l} \models \text{size} \geq \beta$  sii  $\mathfrak{M} \models \text{size} \geq \beta$ . Además, introducimos una notación que utilizaremos a lo largo de este trabajo,  $|Q_1, \dots, Q_n \uparrow \mathfrak{l}_k\rangle$  (con  $\mathfrak{l}_k \in \mathbb{N}$ ) definido de la siguiente forma:

$$\mathfrak{M}, \mathfrak{l} \models |Q_1, \dots, Q_n \uparrow \mathfrak{l}_k\rangle \stackrel{\text{def}}{\iff} \begin{aligned} & \text{existen } \mathfrak{l}_1, \dots, \mathfrak{l}_n \text{ distintos entre sí y con respecto a } \mathfrak{l}_k \\ & \text{tales que } \mathfrak{l} = \mathfrak{l}_1 \mathfrak{R} \mathfrak{l}_2 \mathfrak{R} \dots \mathfrak{R} \mathfrak{l}_n \mathfrak{R} \mathfrak{l}_k \text{ y para todo } j \in [1, n], \mathfrak{M}, \mathfrak{l}_j \models Q_j \end{aligned}$$

A este tipo de fórmulas las llamaremos *fórmulas enlace*. Una observación interesante de esta notación es que expresiones como  $|Q \uparrow \mathfrak{l}_k\rangle$  no satisfacen la relación  $\mathfrak{M}, \mathfrak{l}_k \models |Q \uparrow \mathfrak{l}_k\rangle$  puesto que el mundo actual y el mundo fijo (en este caso  $\mathfrak{l}_k$ ) son los mismos. Por lo tanto, no es posible decir que exista un mundo  $\mathfrak{l}_1$  distinto a  $\mathfrak{l}_k$  tal que  $\mathfrak{l}_1 \mathfrak{R} \mathfrak{l}_k$  y  $\mathfrak{M}, \mathfrak{l}_1 \models Q$ . Para facilitar la lectura de las fórmulas grafo en algunas partes del trabajo, escribiremos  $|Q_1, \alpha, Q_n\rangle$  cada vez que querramos decir  $|Q_1, \dots, Q_n\rangle$  estableciendo que  $\alpha$  denota 0 o más conjunciones  $Q$ 's. Lo mismo haremos con las demás fórmulas grafo. Vale aclarar que ambas notaciones son intercambiables.

**Lema 1.** (*Reescritura*) Sea  $n - i + 1 \geq 2$  y  $\alpha$  una secuencia de 0 ó más conjunciones:

$$\begin{aligned} \mathfrak{M}, \mathfrak{l} \models |Q_i, \alpha, Q_n\rangle & \iff \text{existen } \mathfrak{l}_i, \mathfrak{l}_{i+1} \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_i \mathfrak{R} \mathfrak{l}_{i+1}, \\ & \mathfrak{M}, \mathfrak{l}_i \models Q_i \text{ y } \mathfrak{M}, \mathfrak{l}_{i+1} \models |\alpha, Q_n\rangle \\ \mathfrak{M}, \mathfrak{l} \models |Q_i, \alpha, Q_n] & \iff \text{existen } \mathfrak{l}_i, \mathfrak{l}_{i+1} \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_i \mathfrak{R} \mathfrak{l}_{i+1}, \\ & \mathfrak{M}, \mathfrak{l}_i \models Q_i \text{ y } \mathfrak{M}, \mathfrak{l}_{i+1} \models |\alpha, Q_n] \\ \mathfrak{M}, \mathfrak{l} \models |Q_i, \alpha, \overline{Q_j, \dots, Q_n}], i \neq j & \iff \text{existen } \mathfrak{l}_i, \mathfrak{l}_{i+1} \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_i \mathfrak{R} \mathfrak{l}_{i+1}, \\ & \mathfrak{M}, \mathfrak{l}_i \models Q_i \text{ y } \mathfrak{M}, \mathfrak{l}_{i+1} \models |\alpha, \overline{Q_j, \dots, Q_n}] \\ \mathfrak{M}, \mathfrak{l} \models |\overline{Q_i, \alpha, Q_n}], & \iff \text{existen } \mathfrak{l}_i, \mathfrak{l}_{i+1} \text{ distintos tales que } \mathfrak{l} = \mathfrak{l}_i \mathfrak{R} \mathfrak{l}_{i+1}, \\ & \mathfrak{M}, \mathfrak{l}_i \models Q_i \text{ y } \mathfrak{M}, \mathfrak{l}_{i+1} \models |\alpha, Q_n \uparrow \mathfrak{l}_i \\ \mathfrak{M}, \mathfrak{l} \models |Q_i, \alpha, Q_n \uparrow \mathfrak{l}_j\rangle, & \iff \text{existen } \mathfrak{l}_i, \mathfrak{l}_{i+1} \text{ distintos entre sí y } \mathfrak{l}_i \neq \mathfrak{l}_j \\ & \text{tales que } \mathfrak{l} = \mathfrak{l}_i \mathfrak{R} \mathfrak{l}_{i+1}, \mathfrak{M}, \mathfrak{l}_i \models Q_i \text{ y } \mathfrak{M}, \mathfrak{l}_{i+1} \models |\alpha, Q_n \uparrow \mathfrak{l}_j \end{aligned}$$

*Demostración.* Notar que la demostración se basa en aplicar la definición de  $\models$  en cada caso.  $\square$

## 2.4. Equivalencia

Hasta ahora hemos presentado dos tipos de fórmulas que a priori, expresan propiedades diferentes en los modelos. Sin embargo, gracias a [31], sabemos que efectivamente existe una equivalencia entre estas. En pocas palabras, para cada fórmula elemental existe una contraparte lógicamente equivalente en  $\text{MSL}(*, \diamond)$  y, a su vez, que cada fórmula en  $\text{MSL}(*, \diamond)$  es equivalente a una combinación Booleana de fórmulas elementales.

Para el primer caso, se definió para cada fórmula elemental  $\psi$ , la *extensión*  $\text{ext}(\psi)$  en  $\text{MSL}(*, \diamond)$  tal que:

$$\blacksquare \text{ ext}(\text{size} \geq 0) \stackrel{\text{def}}{=} \top \text{ y } \text{ ext}(\text{size} \geq \beta) \stackrel{\text{def}}{=} \overbrace{\neg \text{emp} * \dots * \neg \text{emp}}^{\beta \text{ veces}},$$

- $\text{ext}(|Q|) \stackrel{\text{def}}{=} Q \wedge \neg \diamond \top$ . Para  $n \geq 2$ ,  $\text{ext}(|Q_1, Q_2, \dots, Q_n|) \stackrel{\text{def}}{=} Q_1 \wedge \diamond \text{ext}(|Q_2, \dots, Q_n|)$ ,
- $\text{ext}(|Q_1, \dots, Q_n|) \stackrel{\text{def}}{=} \text{ext}(|Q_1, \dots, Q_n, \top|) * \top$ ,
- $\text{ext}(|\overline{Q_1, \dots, Q_n}|)$  está definida como la fórmula

$$\top * (\text{ext}(\text{size} = n) \wedge \diamond^{n+1} \top \wedge (\text{ext}(|Q_1, \dots, Q_n|) * \top) \wedge \neg \diamond (\text{ext}(\text{size} = 1) * \diamond^n \top)),$$

donde  $\diamond^0 \phi \stackrel{\text{def}}{=} \phi$  y  $\diamond^{i+1} \phi \stackrel{\text{def}}{=} \diamond \diamond^i \phi$ . Para  $i > 1$ ,  $\text{ext}(|Q_1, \dots, \overline{Q_i, \dots, Q_n}|)$  es

$$\top * (\text{ext}(\text{size} = n) \wedge \diamond^{n+1} \top \wedge (\text{ext}(|Q_1, \dots, Q_n|) * \top) \wedge \diamond^{i-1} (\text{ext}(\text{size} = i-1) * \text{ext}(|\overline{\top, \dots, \top}|_{(n-i+1)}))).$$

Esta definición describe una traducción de fórmulas elementales a fórmulas en  $\text{MSL}(*, \diamond)$ , y se puede concluir la siguiente propiedad:

**Lema 2.** [31] *Toda fórmula elemental  $\phi$  es lógicamente equivalente a  $\text{ext}(\phi)$ .*

Para la dirección opuesta, se introdujeron “axiomas de reducción” (para más detalles ver [33]), que permitieron transformar cualquier fórmula en  $\text{MSL}(*, \diamond)$  en una combinación Booleana de fórmulas elementales. Gracias a esta transformación, la axiomatización para fórmulas elementales resulta un sistema completo para  $\text{MSL}(*, \diamond)$ .

**Lema 3.** [31] *Sea  $\phi$  una fórmula en  $\text{MSL}(*, \diamond)$ , existe una fórmula elemental  $\psi$  tal que:*

1.  $\psi = \bigvee_{i=1}^n \varphi_i$  con  $\varphi_i = (\mathcal{G}_i \wedge \text{size} \geq \beta_i)$  ó  $(\mathcal{G}_i \wedge \text{size} \geq \beta_i \wedge \neg \text{size} \geq \alpha_i)$ ,
2.  $\psi$  y  $\phi$  son equivalentes.

Teniendo en cuenta el Lema 3, dada una fórmula  $\phi$  en  $\text{MSL}(*, \diamond)$  y su fórmula elemental equivalente  $\psi$ , tendremos que una será satisfacible (resp. válida) si y solo si la otra es satisfacible (resp. válida). Esto nos garantiza que, para determinar la satisfacibilidad (resp. validez) de  $\phi$ , basta con ver la de su fórmula elemental  $\psi$  equivalente.

**Corolario 4.** [31] *Las fórmulas elementales tienen el mismo poder expresivo que  $\text{MSL}(*, \diamond)$*

Con respecto a la complejidad de la lógica en [29] se demostró:

**Proposición 5.** *El problema de satisfacibilidad para  $\text{MSL}(*, \diamond)$  es NP-completo.*

Vale destacar que la transformación de fórmulas en  $\text{MSL}(*, \diamond)$  a fórmulas elementales produce una explosión exponencial en el tamaño de la fórmula. Por lo tanto, diseñar algoritmos eficientes basados en fórmulas elementales no es una tarea trivial. En este trabajo nos ocuparemos de definir un cálculo eficiente para fórmulas elementales sin preocuparnos por la complejidad del mismo para decidir la satisfacibilidad de  $\text{MSL}(*, \diamond)$ .

En la siguiente sección, introduciremos un cálculo de tableaux para decidir si una fórmula elemental es satisfacible.



## Capítulo 3

# Cálculo de Tableaux para fórmulas elementales

### 3.1. Introducción

En esta sección, describiremos un procedimiento de tableaux para fórmulas de la forma  $\psi = \bigvee_{i=1}^n \varphi_i$  con  $\varphi_i = (\mathcal{G}_i \wedge \mathbf{size} \geq \beta_i)$  ó  $(\mathcal{G}_i \wedge \mathbf{size} \geq \beta_i \wedge \neg \mathbf{size} \geq \alpha_i)$  incluyendo las reglas de derivación y los conceptos de rama cerrada, abierta y saturada. Además, demostraremos que si este cálculo, para una determinada fórmula, termina teniendo una rama abierta y saturada, entonces dicha fórmula será satisficible. También demostraremos que la recíproca vale. En otras palabras, demostraremos que el cálculo es completo y correcto respectivamente.

### 3.2. Reglas

Para la definición de este cálculo, utilizaremos un etiquetado con los números naturales. Este etiquetado, cuando hablemos más adelante de un determinado modelo, representarán los mundos del mismo. Por convención y para futuras demostraciones, cada vez que querramos hacer un cálculo de tableaux de una fórmula  $\psi$ , empezaremos desde la etiqueta 1. Es decir que la raíz de nuestro árbol será el par  $(1 : \psi)$ .

Además, dado que este cálculo será para fórmulas de la forma  $\psi = \bigvee_{i=1}^n \varphi_i$  con  $\varphi_i = (\mathcal{G}_i \wedge \mathbf{size} \geq \beta_i)$  ó  $(\mathcal{G}_i \wedge \mathbf{size} \geq \beta_i \wedge \neg \mathbf{size} \geq \alpha_i)$ , se tendrá un cierto orden en la aplicación de las reglas de derivación. En dicho orden, se empezará por aplicar la regla para la disyunción  $\vee$ . Utilizando pattern matching, definimos la regla que deberá aplicarse:

$$\frac{i : \varphi_k \vee \bigvee_{j \neq k} \varphi_j}{i : \varphi_k \mid i : \bigvee_{j \neq k} \varphi_j} \vee \text{ para } \varphi_j\text{'s}$$

Como la fórmula es finita, esta regla podrá aplicarse una cantidad finita de veces y con ello, tendremos un número finito de ramas las cuales tendrán exactamente una única fórmula  $\varphi_k$  cada una. En el caso de  $\psi$ , que tiene  $n$  disyuntos, el total de ramas también será  $n$ . Para estas fórmulas  $\varphi_k$  aplicaremos otro tipo de reglas teniendo en cuenta que tienen necesariamente alguna de las siguientes estructuras:  $(\mathcal{G} \wedge \mathbf{size} \geq \beta)$  ó  $(\mathcal{G} \wedge \mathbf{size} \geq \beta \wedge \neg \mathbf{size} \geq \alpha)$ . Estas reglas serán:

$$\frac{i : \mathcal{G} \wedge \mathbf{size} \geq \beta}{i : \mathcal{G} \quad \mathbf{size} \geq \beta} \wedge(\text{I}) \quad \frac{i : \mathcal{G} \wedge \mathbf{size} \geq \beta \wedge \neg \mathbf{size} \geq \alpha}{i : \mathcal{G} \quad \mathbf{size} \geq \beta \quad \neg \mathbf{size} \geq \alpha} \wedge(\text{II})$$

Notar que escribimos los literales  $\mathbf{size}$  sin etiqueta puesto que hablan de propiedades globales que son independientes de la etiqueta en la que estén.

En este cálculo no hay reglas para aplicar en literales  $\mathbf{size}$ . Por lo tanto, después de este punto, solo tendremos reglas para el par  $(i : \mathcal{G})$ . Dichas reglas dependerán directamente de la

forma de la fórmula grafo  $\mathcal{G}$  y estarán definidas de manera recursiva con un caso base y uno o dos casos recursivos. Para el caso de las fórmulas continuadas, tendremos las siguientes reglas por pattern matching:

$$\frac{i : |Q_i\rangle}{i : Q_i} \text{ Continuada I} \quad \frac{i : |Q_i, \alpha, Q_n\rangle}{i : Q_i} \text{ Continuada II}$$

$$\frac{iR(i+1)}{\mathbf{size} \geq 1}$$

$$\frac{iR(i+1)}{\mathbf{size} \geq n - i + 1}$$

$$\frac{i+1 : |\alpha, Q_n\rangle}$$

En la primera regla contemplamos el caso en el que la fórmula es un camino de longitud uno y en la segunda para cuando el camino es de una longitud mayor o igual a dos. Notar que, si aplicamos la regla del caso recursivo una cantidad finita de veces, eventualmente llegaremos al caso base. Además, como estamos utilizando un cálculo con etiquetas, tendremos elementos del tipo  $iRj$ . Estos denotan que la etiqueta  $j$  es accesible directamente vía la etiqueta  $i$ . Una consecuencia negativa de estas definiciones es la población de literales  $\mathbf{size}$ , los cuales en su mayoría resultan innecesarios. Si por ejemplo,  $\mathcal{G} = |Q_1, Q_2, Q_3\rangle$ , tendríamos en la misma rama los elementos  $\mathbf{size} \geq 3$ ,  $\mathbf{size} \geq 2$  y  $\mathbf{size} \geq 1$  cuando el más representativo de los tres sería el primero. Esto es inevitable puesto que por la definición de nuestro cálculo, no se puede llevar una cuenta dinámica de cuál es el máximo límite inferior. Sin embargo, este defecto en nuestro cálculo no nos supondrá una dificultad a lo largo de este trabajo. De manera similar, para las fórmulas truncadas tendremos las siguientes reglas:

$$\frac{i : |Q_i|}{i : Q_i} \text{ Truncada I} \quad \frac{i : |Q_i, \alpha, Q_n]}{i : Q_i} \text{ Truncada II}$$

$$\frac{iR(i+1)}{\mathbf{size} \geq n - i}$$

$$\frac{i+1 : |\alpha, Q_n]}$$

Sin embargo, para las fórmulas cíclicas, tendremos una situación ligeramente diferente:

$$\frac{i : |\overline{Q_i}|}{i : Q_i} \text{ Cícl. I} \quad \frac{i : |Q_i, \alpha, \overline{Q_j, \dots, Q_n}|}{i : Q_i} \text{ Cícl. II}$$

$$\frac{i : |\overline{Q_i, \alpha, Q_n}|}{i : Q_i} \text{ Cícl. III}$$

$$\frac{iRi}{\mathbf{size} \geq 1}$$

$$\frac{iR(i+1)}{\mathbf{size} \geq n - i + 1}$$

$$\frac{i+1 : |\alpha, \overline{Q_j, \dots, Q_n}|}$$

$$\frac{iR(i+1)}{\mathbf{size} \geq n - i + 1}$$

$$\frac{i+1 : |\alpha, Q_n \uparrow_i}$$

En este caso tendremos no solo uno sino dos casos recursivos. El primero es común a los anteriores mientras que el segundo introduce la notación de las fórmulas enlace mencionada en la sección anterior. Esta introducción en la notación tiene sentido puesto que queremos que el cálculo ‘recuerde’ por medio del enlace  $\uparrow_i$ , que el último mundo que contenga la conjunción  $Q_n$  esté enlazado al mundo  $i$ . Dado que las reglas sobre las fórmulas grafo están definidas de una forma recursiva y que ninguna de las fórmulas continuadas o truncadas tienen un mecanismo de ‘memorización’, resulta imposible definir un ciclo sin necesitar crear una estructura auxiliar comprensible. Por otra parte, este segundo caso recursivo nos fuerza a definir un caso base y un caso recursivo para las fórmulas enlace a pesar de que estemos hablando de una simple notación auxiliar:

$$\frac{i : |Q_i \uparrow_j}{i : Q_i} \text{ Enlace I} \quad \frac{i : |Q_i, \alpha, Q_n \uparrow_j}{i : Q_i} \text{ Enlace II}$$

$$\frac{iRj}{i+1 : |\alpha, Q_n \uparrow_j}$$

En el caso de estas fórmulas, no es necesario poner en la definición recursiva que ( $\mathbf{size} \geq n - i + 1$ ) como en los anteriores puesto que, para que aparezcan fórmulas enlace, antes debe haber aparecido una fórmula cíclica. Al existir una fórmula cíclica previa, habrá un literal  $\mathbf{size}$  mucho más restrictivo, lo cual hará innecesario agregar más literales  $\mathbf{size}$  una vez utilizada dicha regla. Además, como consecuencia del segundo caso recursivo de las fórmulas cíclicas, siempre

se tendrá en todos los casos de las fórmulas enlace que  $j < i$ . Como sucede con las fórmulas continuadas, si aplicamos la regla del caso recursivo una cantidad finita de veces en cualquiera de estas fórmulas, eventualmente también llegaremos al caso base. Por lo tanto, ya no podremos aplicar reglas referidas a las fórmulas grafo y lo único que nos quedará serán las conjunciones de literales. En ese caso, usaremos una única regla:

$$\frac{i : \ell \wedge Q}{\begin{array}{c} i : \ell \\ i : Q \end{array}} \wedge \text{ para literales}$$

### 3.3. Ramas cerradas, abiertas y saturadas

Hasta ahora hemos introducido las reglas que posee este cálculo y la noción de que un tableau para una fórmula es un árbol que se va a ir actualizando conforme se vayan aplicando las reglas. Generalmente un tableau termina con múltiples ramas y estas pueden estar cerradas o abiertas. Para este cálculo una rama  $\mathcal{B}$  es cerrada si y solamente si sucede al menos una de las siguientes situaciones:

- $(i : \perp) \in \mathcal{B}$  para algún  $i \in \mathbb{N}$
- $(i : p), (i : \neg p) \in \mathcal{B}$  para algún  $i \in \mathbb{N}, p \in \text{PROP}$
- $\text{size} \geq \beta, \neg\text{size} \geq \alpha \in \mathcal{B}$  con  $\alpha \leq \beta$
- $\neg\text{size} \geq 0 \in \mathcal{B}$

Por otro lado, una rama es abierta si no es cerrada y una rama está saturada si no se le pueden aplicar más reglas.

### 3.4. Completitud

Cuando definimos un cálculo de prueba para una lógica, es natural que el mismo sea un procedimiento válido para la tarea de inferencia correspondiente. En este caso, hablamos de decidir *satisfacibilidad de una fórmula*. En particular, debemos demostrar que un tableau con raíz  $(1 : \psi)$  tiene una rama abierta y saturada, si y solo si la fórmula  $\psi$  es satisfacible. Cada dirección de este “sii” recibe el nombre de completitud y corrección, respectivamente. En esta sección nos ocuparemos de la primera, mientras que la segunda será el objeto de estudio de la siguiente.

Un cálculo de tableaux es completo si para cualquier fórmula, se cumple lo siguiente: si el tableau construido para esta fórmula tiene una rama abierta y saturada, entonces dicha fórmula será satisfacible. Intuitivamente, esto significa que el cálculo encuentra solo fórmulas satisfacibles. Para el lector con conocimientos en teoría de prueba, esta terminología puede sonar incorrecta, dado que en dicha área los conceptos de corrección y completitud pueden verse intercambiados. Sin embargo, en este trabajo nos apegaremos a la terminología utilizada más frecuentemente en tableaux.

En nuestro caso, el cálculo de tableaux está definido para fórmulas de la forma  $\bigvee_{i=1}^n (\mathcal{G}_i \wedge \mathcal{S}_i)$  con  $\mathcal{S}_i = \text{size} \geq \beta_i$  ó  $(\text{size} \geq \beta_i \wedge \neg\text{size} \geq \alpha_i)$ . Sin embargo, antes de probar completitud para las fórmulas elementales de este tipo, probaremos una serie de lemas que serán importantes.

**Lema 6.** *Sea  $Q$  una conjunción de literales,  $\mathcal{B}$  una rama tal que es abierta y saturada,  $\mathfrak{M}^{\mathcal{B}} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V}^{\mathcal{B}} \rangle$  un modelo tal que:*

1.  $\mathfrak{R}$  es una relación cualquiera
2. para todo  $p \in \text{PROP} : \mathfrak{V}^{\mathcal{B}}(p) = \{i \mid (i : p) \in \mathcal{B}\}$

*Si  $(i : Q) \in \mathcal{B}$ , entonces  $\mathfrak{M}^{\mathcal{B}}, i \models Q$ .*

*Demostración.* Por definición de  $\mathfrak{B}$ , se tiene que para cada  $p \in \text{PROP} : (i : p) \in \mathcal{B} \Leftrightarrow i \in \mathfrak{V}(p)$ . Un dato importante a tener en cuenta es que  $Q$  no contiene subfórmulas  $\perp$  ya que si contuviera alguna de ellas, la rama  $\mathcal{B}$  estaría cerrada. Debido a que  $Q$  es una conjunción de literales de la forma  $\top$ ,  $p$  y  $\neg p$ , para demostrar  $\mathfrak{M}^{\mathcal{B}}, i \models Q$  basta con probar los casos bases anteriores y el caso inductivo para el conector  $\wedge$ :

- $\mathfrak{M}^{\mathcal{B}}, i \models \top$  se cumple en todo momento.
- Sea  $p \in \text{PROP} : (i : p) \in \mathcal{B}$   
 $\Leftrightarrow i \in \mathfrak{V}(p)$   
 $\Leftrightarrow \mathfrak{M}^{\mathcal{B}}, i \models p$ .
- Sea  $p \in \text{PROP} : (i : \neg p) \in \mathcal{B}$ , como  $\mathcal{B}$  no es cerrada  
 $\Rightarrow (i : p) \notin \mathcal{B}$   
 $\Leftrightarrow i \notin \mathfrak{V}(p)$   
 $\Leftrightarrow \mathfrak{M}^{\mathcal{B}}, i \not\models p$   
 $\Leftrightarrow \mathfrak{M}^{\mathcal{B}}, i \models \neg p$ .
- Sean  $\ell, Q'$  tales que  $(i : \ell \wedge Q') \in \mathcal{B}$ . Como  $\mathcal{B}$  es saturada, ya se ha aplicado la regla  $\wedge$  para literales. Por lo tanto,  $(i : \ell), (i : Q') \in \mathcal{B}$ . Por hipótesis inductiva,  $\mathfrak{M}^{\mathcal{B}}, i \models \ell$  y  $\mathfrak{M}^{\mathcal{B}}, i \models Q'$ . Luego por definición,  $\mathfrak{M}^{\mathcal{B}}, i \models (\ell \wedge Q')$ .

$\therefore \mathfrak{M}^{\mathcal{B}}, i \models Q$ . □

**Lema 7.** Sea  $\mathcal{B}$  una rama tal que es abierta y saturada,  $\mathfrak{M}^{\mathcal{B}} = \langle \mathbb{N}, \mathfrak{R}^{\mathcal{B}}, \mathfrak{V} \rangle$  un modelo tal que:

1.  $\mathfrak{R}^{\mathcal{B}}$  tal que  $\text{card}(\mathfrak{R}^{\mathcal{B}}) = \text{máx}(\{\omega \mid \text{size} \geq \omega \in \mathcal{B}\} \cup \{0\}) = \gamma$
2.  $\mathfrak{V}$  es una valuación cualquiera

Entonces:

- Si  $\text{size} \geq \beta \in \mathcal{B}$ , entonces  $\mathfrak{M}^{\mathcal{B}} \models \text{size} \geq \beta$ .
- Si  $\neg \text{size} \geq \alpha \in \mathcal{B}$ , entonces  $\mathfrak{M}^{\mathcal{B}} \models \neg \text{size} \geq \alpha$ .

*Demostración.* Como  $\mathcal{B}$  no es cerrada, tendremos que por cada ocurrencia de  $\text{size} \geq \beta$  y  $\neg \text{size} \geq \alpha$  en dicha rama, siempre se tendrá que  $\alpha > \beta$ . Incluso, si  $\neg \text{size} \geq \alpha$  ocurre en  $\mathcal{B}$ ,  $\alpha > 0$ . Luego:

- Si  $\text{size} \geq \beta \in \mathcal{B}$ ,  
 $\Rightarrow$  por definición,  $\text{card}(\mathfrak{R}^{\mathcal{B}}) = \gamma \geq \beta$   
 $\Leftrightarrow \mathfrak{M}^{\mathcal{B}} \models \text{size} \geq \beta$
- Si  $\neg \text{size} \geq \alpha \in \mathcal{B}$ , como  $\text{card}(\mathfrak{R}^{\mathcal{B}}) = \gamma$   
 $\Leftrightarrow$  por definición,  $\text{size} \geq \gamma \in \mathcal{B}$  ó  $\gamma = 0$   
 $\Rightarrow$  como  $\mathcal{B}$  no es cerrada,  $\alpha > \gamma$   
 $\Leftrightarrow \alpha > \text{card}(\mathfrak{R}^{\mathcal{B}})$   
 $\Leftrightarrow \mathfrak{M}^{\mathcal{B}} \models \neg \text{size} \geq \alpha$

□

Con esto procederemos a demostrar completitud:

**Teorema 8.** (*Completitud*) Sea  $\psi = \bigvee_{i=1}^n (\mathcal{G}_i \wedge \mathcal{S}_i)$  con  $\mathcal{S}_i = \text{size} \geq \beta_i$  ó  $(\text{size} \geq \beta_i \wedge \neg \text{size} \geq \alpha_i)$ . Si un tableaux para  $\psi$  tiene una rama abierta y saturada, entonces  $\psi$  es satisfacible.

*Demostración.* Como hemos dicho anteriormente, al ser  $\psi$  una fórmula con  $n$  disyuntos, por la regla  $\vee$  tendremos  $n$  ramas, cada una de estas con exactamente una única fórmula  $(\mathcal{G}_k \wedge \mathcal{S}_k)$ .

Sean:

- $\mathcal{B}$  la rama abierta y saturada de la hipótesis
- $(\mathcal{G}_{\mathcal{B}} \wedge \mathcal{S}_{\mathcal{B}})$  el único disyunto en  $\mathcal{B}$  tal que  $(1 : (\mathcal{G}_k \wedge \mathcal{S}_k)) \in \mathcal{B}$  (pues empezamos en  $(1 : \psi)$ )
- $\gamma = \text{máx}(\{\omega \mid \text{size} \geq \omega \in \mathcal{B}\} \cup \{0\})$
- $m = \text{máx}(\gamma - \#\mathcal{G}_{\mathcal{B}})$

Construiremos un modelo  $\mathfrak{M}^{\mathcal{B}} = \langle \mathbb{N}, \mathfrak{R}^{\mathcal{B}}, \mathfrak{V}^{\mathcal{B}} \rangle$  tal que:

1.  $\mathfrak{R}^{\mathcal{B}} = \text{Bas} \cup \text{Ext}$  tal que

- $\text{Bas} = \{(i, j) \mid iRj \in \mathcal{B}\}$
- $\text{Ext} = \{(a_1, a_2), \dots, (a_{m-1}, a_m), (a_m, a_1)\}$  con  $a_i \neq a_j$  cada vez que  $i \neq j$
- ningún  $a_i$  en  $\text{Ext}$  aparece en  $\text{Bas}$
- ningún  $a_i$  en  $\text{Ext}$  aparece como etiqueta en  $\mathcal{B}$   
(notar que  $\mathfrak{R}^{\mathcal{B}}$  es débilmente funcional)

2. para todo  $p \in \text{PROP}$ ,  $\mathfrak{V}^{\mathcal{B}}(p) = \{i \mid (i : p) \in \mathcal{B}\}$

Nuestro objetivo principal a partir de aquí será demostrar que  $\mathfrak{M}^{\mathcal{B}}, 1 \models \psi$ . Para esto será necesario probar que, para cada  $\phi$ , si  $(i : \phi) \in \mathcal{B}$  para algún  $i \in \mathbb{N}$ ,  $\mathfrak{M}^{\mathcal{B}}, i \models \phi$ . Sin embargo, dado que  $\psi = \bigvee_{i=1}^n (\mathcal{G}_i \wedge \mathcal{S}_i)$ , cualquier fórmula  $\phi$  tal que  $(i : \phi) \in \mathcal{B}$  para algún  $i \in \mathbb{N}$  será necesariamente de alguna de las siguientes formas: una disyunción que contenga a  $(\mathcal{G}_{\mathcal{B}} \wedge \mathcal{S}_{\mathcal{B}})$  ( $\phi = \bigvee_k \mathcal{G}_k \wedge \mathcal{S}_k$ ), la fórmula  $(\mathcal{G}_{\mathcal{B}} \wedge \mathcal{S}_{\mathcal{B}})$ ,  $\mathcal{G}_{\mathcal{B}}$  o una fórmula grafo derivada de  $\mathcal{G}_{\mathcal{B}}$ . Por lo tanto, demostrar que  $\mathfrak{M}^{\mathcal{B}}, 1 \models \psi$  se reduce a probar las siguientes dos proposiciones:

- si  $\mathcal{S}$  es un literal size tal que  $\mathcal{S} \in \mathcal{B}$ , entonces  $\mathfrak{M}^{\mathcal{B}} \models \mathcal{S}$
- si  $\mathcal{G}$  es una fórmula grafo tal que  $(i : \mathcal{G}) \in \mathcal{B}$  para algún  $i \in \mathbb{N}$ , entonces  $\mathfrak{M}^{\mathcal{B}}, i \models \mathcal{G}$

Para probar la primera, solo hará falta ver las definiciones establecidas para  $\gamma$ ,  $\text{Bas}$  y  $\text{Ext}$  y las reglas para las fórmulas grafo. En efecto, tendremos que  $\text{Bas} \cap \text{Ext} = \emptyset$ ,  $\text{card}(\text{Bas}) = \#\mathcal{G}_{\mathcal{B}}$  y  $\gamma \geq \#\mathcal{G}_{\mathcal{B}}$ . Luego:

$$\begin{aligned} \text{card}(\mathfrak{R}^{\mathcal{B}}) &= \text{card}(\text{Bas}) + \text{card}(\text{Ext}) \\ &= \#\mathcal{G}_{\mathcal{B}} + m \\ &= \#\mathcal{G}_{\mathcal{B}} + (\gamma - \#\mathcal{G}_{\mathcal{B}}) \\ &= \gamma \end{aligned}$$

Gracias al Lema 7 tenemos garantizado que para cualquier literal size  $\mathcal{S}$  tal que  $\mathcal{S} \in \mathcal{B}$ ,  $\mathfrak{M}^{\mathcal{B}} \models \mathcal{S}$ . La demostración de la segunda proposición es por inducción en el tamaño de la fórmula grafo  $\mathcal{G}$ . Comenzamos con los casos base:

- Si  $(i : |Q\rangle) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q) \in \mathcal{B}$ : por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q$
  2.  $iR(i+1) \in \mathcal{B}$ : por definición de  $\text{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
  3.  $\text{size} \geq 1 \in \mathcal{B}$ : por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \text{size} \geq 1$

Luego tenemos que existen distintas  $l_1 = i$ ,  $l_2 = i+1$  tales que  $l_1\mathfrak{R}^{\mathcal{B}}l_2$  y  $\mathfrak{M}^{\mathcal{B}}, l_1 \models Q$ . Por definición de  $\models$ ,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q\rangle$ .

- Si  $(i : |Q|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:

1.  $(i : Q) \in \mathcal{B}$ : por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q$

Dado que estamos en el caso  $|Q|$ , sabemos que no hay otra fórmula grafo de otro tipo en la rama y que  $i$  es una etiqueta, podemos decir que:

1.  $iRj \notin \mathcal{B}$  para todo  $j \in \mathbb{N}$

2.  $(i, j) \notin \mathcal{Bas}$  para todo  $j \in \mathbb{N}$
3.  $(i, k) \notin \mathcal{Ext}$  para todo  $k \in \mathbb{N}$

$\therefore \mathfrak{R}^{\mathcal{B}}(i) = \emptyset$ . Luego tenemos que existe  $l_1 = i$  tal que  $\mathfrak{R}^{\mathcal{B}}(l_1) = \emptyset$  y  $\mathfrak{M}^{\mathcal{B}}, l_1 \models Q$ . Por definición de  $\models$ ,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q|$ .

- Si  $(i : \overline{|Q|}) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q) \in \mathcal{B}$ : por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q$
  2.  $iRi \in \mathcal{B}$ : por definición de  $\mathcal{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}i$
  3.  $\mathbf{size} \geq 1 \in \mathcal{B}$ : por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \mathbf{size} \geq 1$

Luego tenemos que existe  $l_1 = i$  tal que  $l_1\mathfrak{R}^{\mathcal{B}}l_1$  y  $\mathfrak{M}^{\mathcal{B}}, l_1 \models Q$ . Por definición de  $\models$ ,  $\mathfrak{M}^{\mathcal{B}}, i \models \overline{|Q|}$ .

- Si  $(i : |Q\uparrow_j|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q) \in \mathcal{B}$ : por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q$
  2.  $iRj \in \mathcal{B}$ : por definición de  $\mathcal{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}j$

Dado que la ocurrencia de una fórmula enlace depende de que haya habido antes una fórmula cíclica y que esto implica que la etiqueta  $j$  debió haber sido introducida antes que la etiqueta  $i$ , tenemos que  $j < i$ . Luego existe  $l_1 = i$  tal que  $l_1\mathfrak{R}^{\mathcal{B}}j$ ,  $l_1 \neq j$  y  $\mathfrak{M}^{\mathcal{B}}, l_1 \models Q$ . Por definición de  $\models$ ,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q\uparrow_j|$ .

Ahora solo nos quedan los casos inductivos.

- Si  $(i : |Q_i, \alpha, Q_n|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q_i) \in \mathcal{B}$ : por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q_i$
  2.  $iR(i+1) \in \mathcal{B}$ : por definición de  $\mathcal{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
  3.  $\mathbf{size} \geq n - i + 1 \in \mathcal{B}$ : por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \mathbf{size} \geq n - i + 1$
  4.  $(i+1 : |\alpha, Q_n|) \in \mathcal{B}$ : por hipótesis inductiva  $\mathfrak{M}^{\mathcal{B}}, (i+1) \models |\alpha, Q_n|$

Dado que  $i \neq i+1$ , usando el Lema 1,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q_i, \alpha, Q_n|$ .

- Si  $(i : |Q_i, \alpha, Q_n|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q_i) \in \mathcal{B}$  por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q_i$
  2.  $iR(i+1) \in \mathcal{B}$  por definición de  $\mathcal{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
  3.  $\mathbf{size} \geq n - i \in \mathcal{B}$  por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \mathbf{size} \geq n - i$
  4.  $(i+1 : |\alpha, Q_n|) \in \mathcal{B}$  por hipótesis inductiva  $\mathfrak{M}^{\mathcal{B}}, (i+1) \models |\alpha, Q_n|$

Dado que  $i \neq i+1$ , usando el Lema 1,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q_i, \alpha, Q_n|$ .

Para el caso de las fórmulas cíclicas, tenemos que realizar un análisis por casos dependiendo de la forma de la misma, tal como se han diferenciado en la definición de las reglas.

- Si  $(i : |Q_i, \alpha, \overline{Q_j, \dots, Q_n}|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:
  1.  $(i : Q_i) \in \mathcal{B}$  por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q_i$
  2.  $iR(i+1) \in \mathcal{B}$  por definición de  $\mathcal{Bas}$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
  3.  $\mathbf{size} \geq n - i + 1 \in \mathcal{B}$  por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \mathbf{size} \geq n - i + 1$
  4.  $(i+1 : |\alpha, \overline{Q_j, \dots, Q_n}|) \in \mathcal{B}$  por hipótesis inductiva  $\mathfrak{M}^{\mathcal{B}}, (i+1) \models |\alpha, \overline{Q_j, \dots, Q_n}|$

Dado que  $i \neq i+1$ , usando el Lema 1,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q_i, \alpha, \overline{Q_j, \dots, Q_n}|$ .

- Si  $(i : |\overline{Q_i, \alpha, Q_n}|) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:

1.  $(i : Q_i) \in \mathcal{B}$  por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q_i$
2.  $iR(i+1) \in \mathcal{B}$  por definición de  $\mathcal{B}as$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
3.  $\mathbf{size} \geq n - i + 1 \in \mathcal{B}$  por Lema 7,  $\mathfrak{M}^{\mathcal{B}} \models \mathbf{size} \geq n - i + 1$
4.  $(i+1 : |\alpha, Q_n \uparrow_i) \in \mathcal{B}$  por hipótesis inductiva  $\mathfrak{M}^{\mathcal{B}}, (i+1) \models |\alpha, Q_n \uparrow_i$

Dado que  $i \neq i+1$ , usando el Lema 1,  $\mathfrak{M}^{\mathcal{B}}, i \models |\overline{Q_i, \alpha, Q_n}$ .

■ Si  $(i : |Q_i, \alpha, Q_n \uparrow_j) \in \mathcal{B}$ , como  $\mathcal{B}$  está saturada:

1.  $(i : Q_i) \in \mathcal{B}$  por Lema 6,  $\mathfrak{M}^{\mathcal{B}}, i \models Q_i$
2.  $iR(i+1) \in \mathcal{B}$  por definición de  $\mathcal{B}as$ ,  $i\mathfrak{R}^{\mathcal{B}}(i+1)$
3.  $(i+1 : |\alpha, Q_n \uparrow_j) \in \mathcal{B}$  por hipótesis inductiva  $\mathfrak{M}^{\mathcal{B}}, (i+1) \models |\alpha, Q_n \uparrow_j$

Dado que la ocurrencia de una fórmula enlace depende de que haya habido antes una fórmula cíclica y que esto implica que la etiqueta  $j$  debió haber sido introducida antes que las etiquetas  $i$  e  $i+1$ , tenemos que  $j < i$  y  $j < i+1$ . Por lo tanto,  $i$  e  $i+1$  son distintos con respecto a  $j$ . Como también  $i \neq i+1$ , usando el Lema 1,  $\mathfrak{M}^{\mathcal{B}}, i \models |Q_i, \alpha, Q_n \uparrow_j$ .

- $\therefore \mathfrak{M}^{\mathcal{B}}, i \models \mathcal{G}$  para todo  $(i : \mathcal{G}) \in \mathcal{B}$ .
- $\therefore \mathfrak{M}^{\mathcal{B}}, i \models \phi$  para todo  $(i : \phi) \in \mathcal{B}$ .
- $\therefore \mathfrak{M}^{\mathcal{B}}, 1 \models \psi$  y  $\psi$  es satisfacible.
- $\therefore$  El cálculo es completo. □

### 3.5. Corrección

En la sección anterior demostramos que el cálculo definido es completo. Esto quiere decir, que si el cálculo no cierra para una determinada fórmula, entonces dicha fórmula es satisfacible. Sin embargo, también nos interesa la converso. Intuitivamente, esto significa que el cálculo no cierra cuando recibe como entrada una fórmula satisfacible. Esta propiedad es conocida como *corrección*, la cual será abordada en esta subsección.

**Teorema 9.** *Sea  $\psi$  una fórmula de la forma  $\bigvee_{i=1}^n (\mathcal{G}_i \wedge \mathcal{S}_i)$  con  $\mathcal{S}_i = \mathbf{size} \geq \beta_i$  ó  $(\mathbf{size} \geq \beta_i \wedge \neg \mathbf{size} \geq \alpha_i)$ , si  $\psi$  es satisfacible, entonces en el cálculo de tableaux tiene una rama abierta y saturada.*

*Demostración.* Sea  $\mathfrak{M} = \langle \mathbb{N}, \mathfrak{R}, \mathfrak{V} \rangle$  un modelo tal que  $\mathfrak{M}, 1 \models \psi$ . Dado que  $\psi$  es una disyunción, se tiene que para algún  $k$ ,  $\mathfrak{M}, 1 \models \mathcal{G}_k \wedge \mathcal{S}_k$ . Por lo tanto,  $\mathcal{G}_k \wedge \mathcal{S}_k$  es satisfacible. Si analizamos el tableaux saturado, la rama en donde aparece  $\mathcal{G}_k \wedge \mathcal{S}_k$  debería estar abierta. Supongamos que no lo está.

Sea  $\mathcal{K}$  la rama saturada y cerrada en la que se encuentra  $\mathcal{G}_k \wedge \mathcal{S}_k$ , debería cumplirse alguna de las siguientes condiciones:

1.  $(i : \perp) \in \mathcal{K}$  para algún  $i \in \mathbb{N}$
2.  $(i : p), (i : \neg p) \in \mathcal{K}$  para algún  $i \in \mathbb{N}, p \in \text{PROP}$
3.  $\mathbf{size} \geq \beta, \neg \mathbf{size} \geq \alpha \in \mathcal{K}$  con  $\alpha \leq \beta$
4.  $\neg \mathbf{size} \geq 0 \in \mathcal{K}$

Sin embargo, dependiendo de cada caso, llegaremos a una contradicción con la hipótesis.

1. Esto implica que la conjunción  $Q_i$  de donde proviene  $(i : \perp)$  es insatisfacible, lo que implicaría que  $\mathcal{G}_k \wedge \mathcal{S}_k$  también lo es.
2. Con el mismo razonamiento del punto anterior, llegamos a la misma contradicción.
3. Esto implica que  $\mathcal{S}_k = (\mathbf{size} \geq \beta_k \wedge \neg \mathbf{size} \geq \alpha_k)$  puesto que es la única forma de que aparezca un literal  $\mathbf{size}$  con una negación en dicha rama.

- Si  $\alpha_k \leq \beta_k$ ,  $\mathcal{S}_k$  es insatisfacible y por lo tanto  $\mathcal{G}_k \wedge \mathcal{S}_k$  también.
  - Si  $\alpha_k \leq \beta$  tal que  $\mathbf{size} \geq \beta$  deriva de la parte  $\mathcal{G}_k$ , tendremos que  $\beta$  es a lo sumo  $\sharp(\mathcal{G}_k)$ . Como consecuencia de que  $\mathcal{G}_k \wedge \mathcal{S}_k$  es satisfacible, el modelo  $\mathfrak{M}$  tendrá al menos  $\sharp(\mathcal{G}_k)$  aristas. Si  $\alpha_k \leq \beta$  con  $\beta \leq \sharp(\mathcal{G}_k)$ , tendremos que  $\mathcal{G}_k \wedge \mathcal{S}_k$  es insatisfacible.
4. Necesariamente  $\mathcal{S}_k = (\mathbf{size} \geq \beta_k \wedge \neg \mathbf{size} \geq 0)$ . Pero por definición de  $\models$ ,  $\mathcal{G}_k \wedge \mathcal{S}_k$  es insatisfacible.

Por lo tanto, si  $\psi$  es satisfacible, el tableaux para dicha fórmula tiene una rama abierta y saturada y el cálculo es correcto.  $\square$

Finalmente, vale la pena mencionar que el cálculo termina independientemente de la fórmula de entrada, y que lo hace en tiempo polinomial. Se omiten los detalles de este aspecto.

## Capítulo 4

# Conclusiones y trabajos futuros

### 4.1. Recapitulación

En este trabajo se ha propuesto un cálculo de tableaux etiquetado para una familia de fórmulas llamadas ‘fórmulas elementales’. Las mismas son un fragmento de la lógica de separación con operadores modales  $MSL(*, \diamond)$  introducida en [29]. Dichas fórmulas, introducidas en [31], realizan un tratamiento sintáctico de las propiedades semánticas estudiadas en [29], estableciendo qué fórmulas en  $MSL(*, \diamond)$  permiten solo expresar propiedades específicas sobre la forma y el tamaño del modelo. En particular, las fórmulas elementales son suficientes para capturar todo el poder expresivo de  $MSL(*, \diamond)$ . Luego de haber propuesto dicho cálculo, se demostró que es completo y correcto.

Una ventaja de este algoritmo por sobre el sistema de [29], es que facilita su tratamiento computacional, es decir, implementación, estudio de complejidad, etcétera. Si bien con este cálculo se puede conjeturar que el problema satisfacibilidad para este tipo de fórmulas está en P, esto no implica que el problema de satisfacibilidad  $MSL(*, \diamond)$  esté también en P. Esto sucede debido a que la traducción de  $MSL(*, \diamond)$  a fórmulas elementales contiene una explosión exponencial en el tamaño de la fórmula original. Es más, en [29] se demuestra que el problema de decidir si una fórmula de  $MSL(*, \diamond)$  es satisfacible es NP-completo. Esto se consigue definiendo un algoritmo de model checking en tiempo polinomial y demostrando una propiedad de modelo acotado.

Finalmente, conjeturamos que es posible definir un algoritmo en EXPTIME para decidir la satisfacibilidad de fórmulas en  $MSL(*, \diamond)$ . Por un lado, dada una fórmula en  $MSL(*, \diamond)$  podemos construir la fórmula elemental equivalente (potencialmente con un tamaño exponencial con respecto a la original), según lo demostrado en [31]. Luego, es posible correr nuestro algoritmo polinomial sobre dicha fórmula.

### 4.2. Trabajos futuros

En paralelo con la realización de este trabajo, mediante una pasantía en el LSV, ENS Paris-Saclay bajo la supervisión de Stéphane Demri, se comenzó a estudiar la complejidad de algunos fragmentos de  $MSL(*, \diamond, \langle \neq \rangle)$ , la lógica  $MSL(*, \diamond)$  enriquecida con la modalidad  $\langle \neq \rangle$ , definida como:

$$\mathfrak{M}, l \models \langle \neq \rangle \phi \Leftrightarrow \mathfrak{M}, l' \models \phi, \text{ con } l \neq l'.$$

Dado que el problema de satisfacibilidad para  $MSL(*, \diamond, \langle \neq \rangle)$  tiene una complejidad TOWER-complete<sup>1</sup> [29], empezamos a analizar ciertos fragmentos de  $MSL(*, \diamond, \langle \neq \rangle)$  con menor expresividad conectados por medio de conectores Booleanos  $\vee$ ,  $\wedge$  y  $\neg$ .

El objetivo de este trabajo es demostrar que la complejidad del problema de satisfacibilidad para la combinación Booleana de fórmulas pertenecientes a los fragmentos  $MSL(*, \diamond)$ ,  $MSL(*, \langle \neq \rangle)$  y  $MSL(\diamond, \langle \neq \rangle)$  está en NP, una complejidad mucho menor que TOWER. Demostrar esto se reduce, gracias a los resultados en [29] que nos permiten trabajar sobre modelos

---

<sup>1</sup>Para detalles sobre la clase de problemas TOWER, ver [36]

pequeños, a demostrar la siguiente conjetura:

**Conjetura 10.** Sean  $\phi_1$ ,  $\phi_2$  y  $\phi_3$  tres fórmulas de las lógicas  $MSL(*, \diamond)$ ,  $MSL(*, \langle \neq \rangle)$  y  $MSL(\diamond, \langle \neq \rangle)$  respectivamente, y  $\mathfrak{M}_f = \langle \mathfrak{W}, \mathfrak{R}, \mathfrak{V} \rangle$  un modelo finito con  $\mathfrak{I} \in \mathfrak{W}$ . Entonces, existe un algoritmo de model checking tal que en tiempo polinomial puede determinar si  $\mathfrak{M}_f, \mathfrak{I} \models \phi_1 \wedge \phi_2 \wedge \phi_3$ .

Por otro lado, junto con Raul Fervari, estaremos trabajando en mi tesis doctoral acerca de Lógicas Epistémicas con Estrategias. En la misma se buscará desarrollar lenguajes provistos de agentes capaces de aprender estrategias, es decir, una cierta serie de acciones que puede hacer bajo determinadas condiciones para llegar a un objetivo. Si bien este trabajo no está directamente relacionado con lógicas de separación, seguiremos enfocados en el estudio de ciertas operaciones dinámicas en lógica modal.

### 4.3. Pensamientos personales y agradecimientos

Este trabajo me ha abierto un mundo de grandes posibilidades. No solo por tener la oportunidad de trabajar con Fervari y Demri, sino también por entender lo que significa trabajar en el ámbito científico en términos de cooperación y horizontalidad más allá de los defectos que pueda tener y efectivamente tiene. El futuro es realmente incierto y lleno de desafíos. Pero siempre existirá la oportunidad de aprender, crecer y “progresionar”.

En principio quiero agradecer a Ignacio Queralt y Luciana Benotti que me han dado el empujón inicial para empezar a buscar un tema para mi trabajo final de licenciatura. A Aleksey Boulchouk por haberme animado a presentarme a las Becas de Estímulo a las Vocaciones Científicas, una beca de investigación del Consejo Interuniversitario Nacional para estudiantes de grado. A mis compañeros de carrera, que son muchos para nombrarlos. Si bien nuestros caminos han divergido y convergido en diferentes tiempos, agradezco muchísimo el tiempo que he pasado con ustedes. A mi familia, en especial a mis padres, por el apoyo incondicional y la formación que me han dado para ser la persona que soy ahora. A Raul Fervari y a Stéphane Demri por haber podido tener la enriquecedora oportunidad de haber trabajado y seguir trabajando y aprendiendo con ellos.

Por supuesto, quiero agradecer Consejo Interuniversitario Nacional por haber becado este trabajo de investigación. Al centro de estudiantes de la facultad y a las demás agrupaciones estudiantiles, a la FaMAF, a la Universidad Nacional de Córdoba. Y finalmente a la educación pública, gratuita y de calidad que he podido tener a lo largo de estos cinco años.

# Bibliografía

- [1] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
- [2] Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*. North-Holland, 2007.
- [3] Arthur Prior. *Time and Modality*. Oxford University Press, 1957.
- [4] Leonard Linsky. On interpreting doxastic logic. *Journal of Philosophy*, 65(17):500–502, 1968.
- [5] Jaakko Hintikka. *Knowledge and Belief. An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca, NY, 1962.
- [6] George H. von Wright. Knowledge and Belief. An Introduction to the Logic of the Two Notions. *Mind*, 60:1–15, 1951.
- [7] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
- [8] Mordechai Ben-Ari. *Principles of the spin model checker*. Springer, 2008.
- [9] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, pages 55–74. IEEE Computer Society, 2002.
- [10] Dov M. Gabbay and Sérgio Marcelino. Modal logics of reactive frames. *Studia Logica*, 93(2-3):405–446, 2009.
- [11] Christof Löding and Philipp Rohde. Model checking and satisfiability for sabotage modal logic. In Paritosh K. Pandya and Jaikumar Radhakrishnan, editors, *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference, Mumbai, India, December 15-17, 2003, Proceedings*, volume 2914 of *Lecture Notes in Computer Science*, pages 302–313. Springer, 2003.
- [12] Johan van Benthem. An essay on sabotage and obstruction. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 268–276. Springer, 2005.
- [13] Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Tableaux for relation-changing modal logics. In Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt, editors, *Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013. Proceedings*, volume 8152 of *Lecture Notes in Computer Science*, pages 263–278. Springer, 2013.
- [14] Guillaume Aucher, Johan van Benthem, and Davide Grossi. Sabotage modal logic: Some model and proof theoretic aspects. In Wiebe van der Hoek, Wesley H. Holliday, and Wen-Fang Wang, editors, *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, volume 9394 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2015.

- [15] Guillaume Aucher, Johan van Benthem, and Davide Grossi. Modal logics of sabotage revisited. *J. Log. Comput.*, 28(2):269–303, 2018.
- [16] Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
- [17] Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Moving Arrows and Four Model Checking Results. In *Logic, Language, Information and Computation*, volume 7456 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2012.
- [18] Raul Fervari. *Relation-Changing Modal Logics*. PhD thesis, Universidad Nacional de Córdoba, Argentina, 2014.
- [19] Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Swap Logic. *Logic Journal of the IGPL*, 22(2):309–332, 2014.
- [20] Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Relation-Changing Modal Operators. *Logic Journal of the IGPL*, 23(4):601–627, 2015.
- [21] Carlos Areces, Raul Fervari, Guillaume Hoffmann, and Mauricio Martel. Relation-Changing Logics as Fragments of Hybrid Logics. In *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Italy*, pages 16–29, 2016.
- [22] Carlos Areces, Raul Fervari, Guillaume Hoffmann, and Mauricio Martel. Undecidability of Relation-Changing Modal Logics. In *Dynamic Logic. New Trends and Applications - First International Workshop, DALI 2017, Brasilia, Brazil, September 23-24, 2017, Proceedings*, pages 1–16, 2017.
- [23] Carlos Areces, Raul Fervari, Guillaume Hoffmann, and Mauricio Martel. Satisfiability for relation-changing logics. *J. Log. Comput.*, 28(7):1443–1470, 2018.
- [24] Carlos Areces, Hans van Ditmarsch, Raul Fervari, and François Schwarzentruber. Logics with Copy and Remove. In *Logic, Language, Information, and Computation*, volume 8652 of *Lecture Notes in Computer Science*, pages 51–65. Springer, 2014.
- [25] Carlos Areces, Hans van Ditmarsch, Raul Fervari, and François Schwarzentruber. The modal logic of copy and remove. *Inf. Comput.*, 255:243–261, 2017.
- [26] Raul Fervari and Fernando R. Velázquez-Quesada. Dynamic epistemic logics of introspection. In *Dynamic Logic. New Trends and Applications - First International Workshop, DALI 2017, Brasilia, Brazil, September 23-24, 2017, Proceedings*, pages 82–97, 2017.
- [27] Raul Fervari and Fernando R. Velázquez-Quesada. Introspection as an action in relational models. *J. Log. Algebr. Meth. Program.*, 108:1–23, 2019.
- [28] Stéphane Demri and Morgan Deters. Separation logics and modalities: a survey. *Journal of Applied Non-Classical Logics*, 25(1):50–99, 2015.
- [29] Stéphane Demri and Raul Fervari. On the complexity of modal separation logics. In Guram Bezhanishvili, Giovanna D’Agostino, George Metcalfe, and Thomas Studer, editors, *Advances in Modal Logic 12, proceedings of the 12th conference on “Advances in Modal Logic,” held in Bern, Switzerland, August 27-31, 2018*, pages 179–198. College Publications, 2018.
- [30] Krister Segerberg. A note on the logic of elsewhere. *Theoria*, 47:183–187, 1981.
- [31] Stéphane Demri, Raul Fervari, and Alessio Mansutti. Axiomatizing logics with separating conjunction and modalities. In Francesco Calimeri, Nicola Leone, and Marco Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 692–708. Springer, 2019.

- [32] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Kluwer, 2007.
- [33] Johan van Benthem. *Logical Dynamics of Information and Interaction*. Cambridge University Press, 2011.
- [34] Yanjing Wang and Qinxiang Cao. On axiomatizations of public announcement logic. *Synthese*, 190(Supplement-1):103–134, 2013.
- [35] Marcello D’Agostino, Dov M. Gabbay, Reiner Hähnle, and Joachim Posegga. *Handbook of tableau methods*. Kluwer Academic Publishers, 1999.
- [36] Sylvain Schmitz. Complexity hierarchies beyond Elementary. *ACM Transactions on Computation Theory*, 8(1):3:1–3:36, 2016.