



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Caracterización y Definibilidad en Fragmentos Modales de Primer Orden

Characterization and Definability in Modal First-Order Fragments

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Facundo Matías Carreiro

Director: Santiago Daniel Figueira

Codirector: Carlos Eduardo Areces

Buenos Aires, 5 de Marzo de 2010

CONTENTS

Abstract	iii
Agradecimientos	v
Abstract extendido	vii
1. Introduction	1
1.1 A bit of history	1
1.2 Basic modal logic	2
1.3 Model equivalence	3
1.4 Saturation	8
1.5 What this thesis is about	10
2. Known results for BML	13
2.1 Characterization	13
2.2 Definability	17
3. The generalized framework	21
3.1 Basic definitions	22
3.2 General model equivalence	25
3.3 Saturation	26
4. Main Results	29
4.1 Adequate pair	29
4.2 Characterization	30
4.3 Definability	34
5. Applications	39
5.1 Memory Logics	39
5.2 Hybrid Logics	45
6. Conclusions and further work	49
Appendix A: Auxiliary results	51
Appendix B: Filters and ultraproducts	53
Bibliography	55

ABSTRACT

Model theoretic results such as Characterization and Definability give important information about different logics. It is well known that the proofs of those results for several modal logics have, somehow, the same ‘taste’. A general proof for most modal logics below first order is still too ambitious. In this thesis we plan to isolate sufficient conditions for the characterization and definability theorems to hold in a wide range of logics. Along with these conditions we will prove that, whichever logic that meets them, satisfies both theorems. Therefore, one could give an unifying proof for logics with already known results. Moreover, one will be able to prove characterization and definability results for logics that have not yet been investigated. In both cases, it is only needed to check that a logic meets the requirements to automatically derive the desired results.

Keywords: logic, modal, characterization, definability, saturation.

AGRADECIMIENTOS

Una gran cantidad de personas invirtió tiempo, esfuerzo y buena voluntad para que pudiera llegar a este momento. La siguiente lista de agradecimientos no planea ser extensiva ni mucho menos. Probablemente olvide nombres pero todo el que sienta que haya hecho un aporte puede sentirse incluido.

Primero que nada quiero agradecer a Santiago Figueira, por confiar en mi desde el principio y abrirme un mundo de oportunidades y personas nuevas. Gracias por incentivar-me a la investigación y poner todo tu empeño para crear un ambiente de trabajo genial. También por ser una persona abierta, dispuesta a compartir mucho más que algunas reuniones por semana.

Le debo mucho también, a Carlos Areces, por orientarme en mi tesis y fuera de ella. Por darme la oportunidad de conocer lugares nuevos y personas muy importantes. Por encargarse de todos los arreglos necesarios, siempre! Quiero agradecer a ambos, Carlos y Luciana Benotti, por darme el mejor desayuno de mi vida, recibirme en su casa como si fuera la mía, ayudarme y compartir un montón de cosas. Son personas realmente bondadosas y les estoy muy agradecido.

A todo el Grupo de Lógica y Computabilidad, por haberme recibido con los brazos abiertos sin pedir nada a cambio. Gracias a Sergio Mera y Daniel Gorín por haberme enseñado innumerables cosas sobre lógica y sobre cómo ser un buen docente, con teoría y dando el ejemplo. A Ricardo Rodriguez, por mostrar la mejor disposición al recibirme en sus proyectos y por actuar como jurado de mi tesis. Estoy en deuda con todos ellos por haber revisado detalladamente mi trabajo y haber aportado comentarios (o discutido) al respecto. También quisiera agradecer a Verónica Becher por mostrar un contagioso gran entusiasmo por su trabajo y responder mis preguntas sobre ‘que hace un investigador’.

Un muy merecido agradecimiento a la ‘Gente de la Facu’, por ser mis compañeros dentro y fuera de las aulas en los últimos seis años: Matías Blanco, Fernando Bugni, Luis Brassara, Bruno Cuervo Parrino, Diego Freijo, Maxi Giusto, Pablo Laciana, Sergio Medina, Ariel Neisen, Santiago Palladino, Leandro Radusky, Leo Rodriguez, Nati Rodriguez, Viviana Siless, Javier Silveira, Leo Spett, Andrés Taraciuk, Martín Verzilli, Pablo Zaidenvoren, Eddy Zoppi. El apoyo del grupo fue muy importante para poder concretar varios proyectos.

Al Departamento de Computación de la Universidad de Buenos Aires, por superar todas mis expectativas. Encontré allí, además de un excelente nivel académico, una calidad y calidez humana que será difícil de superar.

A mi familia, mención especial para mis padres por intentar siempre resolverme la mayor cantidad de problemas para que yo pueda continuar tranquilo con mis cosas. También quiero destacar a José Cuba y María Inés Cuba por demostrar real interés en mi inentendible trabajo.

A mis amigos Gabriel, Murray, Nico, Martina, Gon y el resto del grupo por ayudarme a despejar mi cabeza y acompañarme en el mundo real.

Finalmente, agradezco profundamente a Lucila Schmidt, por su apoyo constante de mil maneras distintas y soportar mis mejores y peores momentos. Por ayudarme a ser una mejor persona.

ABSTRACT EXTENDIDO

Para una amplia variedad de aplicaciones que usan la lógica como herramienta, la lógica de primer orden es suficiente para resolver sus problemas de manera *teórica*. Sin embargo, cuando se considera el comportamiento *práctico* de la lógica de primer orden uno se encuentra con varias complicaciones. Primero que nada, la lógica de primer orden es *indecidable*. Esto quiere decir que no existe un algoritmo general para decidir si una fórmula arbitraria es satisfacible. Segundo, en general la mayoría de las aplicaciones que usan esta lógica no aprovechan al máximo su poder. Por lo tanto, incluso cuando se trabaja con fragmentos decidibles de primer orden, se puede estar pagando un precio excesivo por cualidades que no serán utilizadas.

Las lógicas modales proposicionales ofrecen una alternativa a los lenguajes tradicionales. Pueden ser pensadas como un conjunto de herramientas que permiten diseñar lógicas específicamente construidas para una tarea en particular, posibilitando un control fino en su expresividad. Más aún, las lógicas modales resultaron tener un buen comportamiento computacional que probó ser bastante robusto frente a extensiones. Estas características, entre otras, ubicaron a las lógicas modales como una alternativa atractiva con respecto a los lenguajes clásicos como por ejemplo la lógica de primer orden.

En esta tesis trabajaremos con lógicas modales que son *a lo sumo* tan expresivas como la lógica de primer orden. Informalmente, esto quiere decir que si uno puede expresar una propiedad con una fórmula de dicha lógica modal entonces existe una manera de expresar *la misma* propiedad en primer orden. En otras palabras, uno puede decir que si una fórmula modal φ denota una propiedad dada entonces existe algún tipo de *traducción* cuyo resultado es una fórmula de primer orden φ^t que denota la misma propiedad.

En ciencias de la computación, una bisimulación es, a grandes rasgos, una relación binaria entre modelos que asocia aquellos que se comportan de la misma manera. Así, dos modelos son bisimilares cuando no pueden ser distinguidos mutuamente por un observador. La noción de bisimulación es ampliamente empleada en varias áreas como la lógica modal, la teoría de concurrencia, la teoría de conjuntos, la verificación formal, etc.

La noción de bisimulación fue descubierta de manera independiente y relativamente simultánea por van Benthem, en el contexto de teoría de correspondencia modal; Milner y Park, en teoría de la concurrencia; y Forti y Honsell en teoría de conjuntos sin axioma de buena fundación. Estos últimos utilizan bisimulaciones para mostrar la equivalencia de objetos con estructura infinita no-inductiva y garantizar así extensionalidad de los modelos de su teoría [FH83]. Van Benthem [vB76] obtiene la idea de bisimulación como una generalización del concepto de p -morfismo entre modelos; con ella caracteriza a la lógica modal básica como el fragmento de primer orden invariante bajo bisimulaciones (lo que se conoce como *Teorema de Caracterización de van Benthem*). Milner y Park fueron los que acuñaron el término *bisimulación*, técnica que utilizaron como herramienta para probar la equivalencia de procesos concurrentes [Mil80, Par81]. En [San09] se da un interesante panorama histórico del área.

La *bisimulación* es una herramienta crucial en el proceso de estudiar estructuras relacionales y abre el camino para poder analizar formalmente caracterizaciones de la expresividad de los lenguajes modales. Intuitivamente, fijada una lógica \mathcal{L} , la noción de

bisimulación define cuándo dos modelos son indistinguibles para \mathcal{L} (es decir, no existe una fórmula de \mathcal{L} que sea verdadera en un modelo y falsa en otro). Existe una gran variedad de áreas en donde la bisimulación juega ahora un rol central: lógica modal [vB76], teoría de concurrencia [Par81], teoría de conjuntos [FH83], verificación formal [COP00], generación de lenguaje natural [AKS08], etc.

El teorema de caracterización de van Benthem para la lógica modal básica caracteriza el fragmento de primer orden invariante bajo la definición de bisimulación. Informalmente, puede ser enunciado de la siguiente manera.

Teorema. Una fórmula de primer orden α es equivalente a la traducción de una fórmula de la lógica modal básica si y solo si α es invariante bajo bisimulaciones.

Ahora bien, desde un punto de vista lógico, no existe una única noción de *bisimulación*. A cada lenguaje modal le corresponde una noción de bisimulación distinta (o, en el caso de lógicas sub-booleanas, una *simulación* [KdR97, KdR99]).

En general, cada combinación de lógica y bisimulación tiene su demostración de un teorema equivalente a la caracterización de van Benthem. Un problema esencial es que no parece haber una demostración general y cada caso necesita una nueva prueba usando herramientas ad-hoc.

El nacimiento del concepto de bisimulación y la teoría de correspondencia ayudó a responder nuevas preguntas desde una perspectiva puramente de teoría de modelos. Un ejemplo es la caracterización de *definibilidad* en lógica modal. Informalmente decimos que una clase de modelos es *definible* por un conjunto de fórmulas Γ si está compuesta por exactamente todos los modelos donde Γ es válida. Una clase se dice definible por una fórmula modal si es definible por un conjunto singleton.

Sería interesante saber qué propiedades debería cumplir una clase de modelos para ser definible ya sea por un conjunto de fórmulas o por única una fórmula modal. Esta pregunta ya se ha enunciado y respondido para la lógica de primer orden. Para ese caso, la respuesta está formulada en términos de *isomorfismos potenciales*. En cambio, en el caso de las lógicas modales la noción de bisimulación juega un rol esencial. Para dar un ejemplo citamos el siguiente resultado para la lógica modal básica [BdRV01].

Teorema. Una clase de modelos K es definible por una fórmula modal si y solo si K y \bar{K} están cerrados por bisimulaciones y ultraproductos.

Por el momento, no es necesario preocuparse por la definición formal de ‘ultraproductos’. Sólo es necesario saber que los ultraproductos son una construcción de modelos (con orígenes algebraicos) muy útiles. Inicialmente, dicha construcción llamó la atención a los lógicos porque podía ser usada para dar una demostración *puramente algebraica* del Teorema de Compacidad para primer orden. Para un desarrollo detallado sobre ultraproductos recomendamos la lectura de [Kei08].

Como con el teorema de caracterización, resultados de definibilidad similares al aquí presentado valen para una amplia variedad de lógicas modales. De la misma manera, cada lógica tiene su propia demostración especialmente diseñada para ese caso en particular.

Claramente, resultados como los de Caracterización y Definibilidad nos sirven para entender mejor una lógica. Es más, estos resultados también tienen un gran impacto en aplicaciones prácticas de Ciencias de la Computación.

Consideremos el siguiente problema: Supongamos que estamos interesados en realizar *model checking*, esto quiere decir, dado el modelo de un sistema, verificar automáticamente si el modelo cumple una cierta especificación. Supongamos también que la especificación puede ser escrita como una fórmula de primer orden φ .

Siempre podemos usar herramientas de primer orden para verificar si el modelo satisface φ pero eso puede resultar, como ya hemos mencionado, en un alto costo en cuanto a complejidad computacional. Uno podría tratar de encontrar lógicas más ‘baratas’ que puedan resolver el problema. Si la misma propiedad pudiera ser expresada en una lógica modal entonces probablemente podríamos mejorar la performance del proceso drásticamente.

Discutamos un ejemplo concreto: Supongamos que los puntos del dominio de nuestro modelo son diferentes estados en la ejecución de un programa. De esta manera, hay una transición desde un punto a otro si es posible ejecutar una transformación del programa que lo lleve del estado a al estado b . Pensando en el modelo de esta manera se puede ver que los estados sin sucesores representan estados donde el programa ha finalizado.

Una propiedad deseable del modelo podría ser que “en cada estado del programa se debe poder ‘escapar’ del flujo de ejecución”. Esto quiere decir que todo punto debe poder ver directamente a un estado sin sucesores. Esta propiedad puede ser verificada probando que la fórmula de primer orden

$$\varphi(x) = \exists y.R(x, y) \rightarrow (\exists z.R(x, z) \wedge \forall w.\neg R(z, w))$$

sea válida en el modelo pero también puede ser verificada probando que la fórmula de la lógica modal básica $\psi = \diamond\top \rightarrow \diamond\Box\perp$ sea válida en el modelo. Como estas dos fórmulas representan la misma propiedad (son equivalentes) podemos usar *model checkers* que acepten fórmulas de la lógica modal básica como entrada para poder resolver nuestro problema.

Aparte de ser mucho más ‘amigable’, la simple existencia de la fórmula ψ nos dice que la propiedad en cuestión es invariante bajo bisimulaciones. Esta información nos brinda un beneficio extra. Supongamos que el modelo es automáticamente generado a partir de una porción de código. Si, por ejemplo, alimentamos al generador con el código de un sistema operativo entero, el modelo resultante será *muy* grande.

No es el objetivo de esta tesis meterse en estos temas pero existen algoritmos eficientes para minimizar el modelo automáticamente. Estos algoritmos preservan la verdad de las fórmulas invariantes por bisimulación [Hop71, Gri73]. Por lo tanto, al tener una fórmula modal que representa nuestra propiedad, uno podría primero minimizar el modelo y hacer *model checking* sobre el modelo resultante que será, muy probablemente, mucho más chico que el original.

Por otra parte, supongamos que se quiere verificar si el modelo es ‘irreflexivo’. Esto quiere decir, que ningún elemento está relacionado consigo mismo. Si interpretamos esta propiedad en el escenario descrito anteriormente, la propiedad diría que ningún estado debe poder quedarse ‘colgado’ en si mismo.

Para este caso, incluso cuando la propiedad puede ser puesta a prueba verificando la validez de la fórmula de primer orden $\neg R(x, x)$, no existe ninguna fórmula de la lógica modal básica que sea equivalente. Esto puede ser demostrado fácilmente ya que la ‘irreflexividad’ no es invariante bajo bisimulaciones. Es más, la lógica modal básica tiene una propiedad llamada la ‘propiedad de modelo de árbol’ o *tree model property*. Esto quiere decir que cualquier fórmula satisfacible es también satisfacible en un modelo que

tiene forma de árbol. Como corolario surge que no existe ninguna fórmula de la lógica modal básica que caracterice irreflexividad, antisimetría ni intransitividad.

Es este el fin de la lógica modal? Estamos condenados a usar la lógica de primer orden en este caso? Afortunadamente, la respuesta es *no*. Aunque la lógica modal básica no pueda expresar esas propiedades, existen lógicas modales más ricas (que aún se mantienen por debajo del poder expresivo de primer orden) que pueden hacerlo. Como ejemplo, pueden usarse las lógicas híbridas [BdRV01] y las menos conocidas Memory Logics [AFFM08] que serán oportunamente introducidas.

En síntesis, resultados de teoría de modelos como los de Caracterización y Definibilidad dan información importante sobre las distintas lógicas. Se sabe que las demostraciones de estos resultados para las diferentes lógicas modales tienen, de algún modo, el mismo ‘sabor’. Una prueba general que cubra todas las lógicas modales por debajo de primer orden es, por el momento, un plan demasiado ambicioso.

En esta tesis damos condiciones muy generales pero suficientes para que las propiedades de Caracterización y Definibilidad valgan en una amplia gama de lógicas modales: cualquier lógica modal que cumpla nuestras condiciones verificará las propiedades de Caracterización y Definibilidad. El resultado se puede aplicar tanto a lógicas para las que se saben ciertas las propiedades en cuestión, como a lógicas para las que se desconocía si estas propiedades valían o no. En el primer caso, obtenemos demostraciones nuevas de resultados ya conocidos (en particular, aplicamos nuestro esquema a las lógicas híbridas con el operador @ y nominales). En el segundo caso, obtenemos resultados novedosos, aplicando nuestras herramientas a las memory logics, una familia de lógicas modales con comportamiento dinámico introducidas recientemente en el área.

1. INTRODUCTION

1.1 A bit of history

The first traces of modal logic go back to 1918 with the work of C. Lewis [Lew18]. In this publication he enriched the propositional calculus with two operators to try to solve some problems with material implication. In a modern notation these operators would be \Box and \Diamond . Given a formula φ , then, $\Box\varphi$ was meant to be interpreted as “it is necessary that φ ” and $\Diamond\varphi$ as “it is possible that φ ”. At this point (called the ‘syntactic era’) all the work on Modal Logic was strictly *syntactical*, there was no model theory for it.

Later, during the end of 1950s and early 1960s (sometimes called the ‘classical era’) the first ideas on modal logic semantics were born. The seminal work of Prior [Pri57] (with *tense logic*) and Jónson and Tarski with boolean algebras with operators [JT51, JT52] later gave birth to Kripke semantics for modal logics. Kripke’s work [Kri63a, Kri63b] proposes a relational semantic for modal logic, that is, a suitable model to evaluate a modal formula is just a set of *worlds* (or points) and *relations* among them.

With these semantics, many difficult problems (such as knowing whether two axiomatic systems are equivalent) had now turned a lot easier. The emergence of *cannonical models* and *completeness* results were predominant in this period which helped link the ancient ‘syntactic era’ with the new semantics. Although the research made in the ‘classical era’ was not syntactical, it was anyways *syntactically driven*. That is, relational semantics, were used as a tool to analyze logics and prove syntactical results. Model theory, was not playing a big role by itself.

The so-called ‘modern era’ goes from the 1970s to the present days. In this period, modal logic started to be used to *describe* relational structures and not just as a mere tool. The germ of modal logic also started to spread to other fields, as an example, computer scientists started to use modal logics to reason about *programs* represented as relational models. The first steps in this line of work were taken by Pratt [Pra79] with his work on *propositional dynamic logic* (PDL). Computer scientists added new problems to the already growing pool of questions. Complexity of the satisfiability problem for modal logics started to be studied with the work of Ladner [Lad77] for normal logics and Ladner, Fischer and Pratt [FL79, Pra79] for PDL.

The discovery of *frame incompleteness* results showed that there are classes of models for which there is no possible axiomatization (Thomason [Tho72, Tho74] and Fine [Fin74]). This shows that modal logics can’t be analyzed from a purely syntactical perspective.

Modal logic is not isolated from the rest of the world. During this period, the *expressive* power of modal logics was put into question. Which logic is the best to describe certain relational structures? Now that we know that different logics have different computational complexities, which is the ‘cheapest’ logic that solves my problem? The power of these logics could be compared between each other and also with respect to classical logics such as first and second order logic.

The results brought to light by this period helped shift the view of modal logics as ‘intensional’ formalisms that were only able to talk about ‘modes of truth’ to a much broader panorama, which constitutes the current way of looking at modal logics.

1.2 Basic modal logic

It is now time to formally meet the modal logics we are going to work with and its relational semantics. We start by defining the *basic modal language* BML. Because we are interested in working with many modalities at the same time, the diamond (\diamond) and box (\square) operators are going to turn into the operators $\langle r \rangle$ and $[r]$, where r indicates the modality we are working with. When we are in a case where there is a single modality, we are going to use \diamond and \square again.

Definition 1.2.1 (Syntax). Suppose we have a set of propositional symbols $\text{PROP} = \{p_1, p_2, \dots\}$ and a set of modality symbols $\text{REL} = \{r_1, r_2, \dots\}$. We assume that both sets are pairwise disjoint and countable infinite. A specific choice of PROP and REL is called the *signature* of the language. We define the set of formulas of the *basic modal language* over the signature $\langle \text{PROP}, \text{REL} \rangle$ as:

$$\varphi ::= \top \mid \perp \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle r \rangle\varphi \mid [r]\varphi$$

where $p \in \text{PROP}$, $r \in \text{REL}$ and φ, ψ are formulas.

Of course this is not a minimal definition. One can fix an adequate set of primitive boolean connectors (like \neg and \wedge) and define all the other boolean connectors in terms of that primitive set. Also, as it will follow from the satisfaction definition we are going to present below, diamond and box are dual operators, and therefore for all $r \in \text{REL}$, $\langle r \rangle\varphi$ can be defined as $\neg[r]\neg\varphi$, and conversely, $[r]\varphi$ is equivalent to $\neg\langle r \rangle\neg\varphi$. We are not going to bother yet to pick a set of primitive operators, since it is not really important at this point. When we do that, we will only have to worry about choosing a convenient set that allows us to generate the whole language.

Now we formally define the models for the basic modal language. As we mention before, Kripke semantics define models as graphs, and in fact, as *directed* graphs with *decorations*.

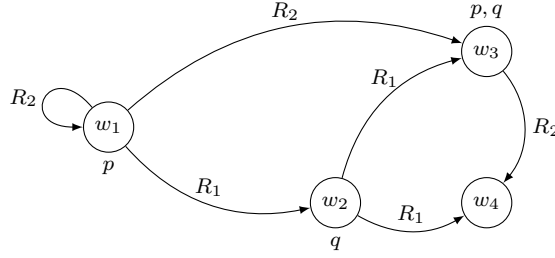
Definition 1.2.2 (Kripke models). Let $\mathcal{S} = \langle \text{PROP}, \text{REL} \rangle$ be a signature. A *Kripke model* \mathcal{M} for \mathcal{S} is a tuple $\langle W, (R_r)_{r \in \text{REL}}, V \rangle$ satisfying the following conditions:

- (i) W , the *domain*, is a nonempty set whose elements are called points, but also, depending on the context, states, worlds, times, etc.
- (ii) Each R_r , an *accessibility relation*, is a binary relation on W .
- (iii) $V : \text{PROP} \rightarrow \mathcal{P}(W)$, the *valuation*, is a labeling function that assigns to each propositional symbol $p \in \text{PROP}$ a subset of W . We can think of $V(p)$ as the set of points in \mathcal{M} where p holds.

Given a model \mathcal{M} and $w \in |\mathcal{M}|$, we call $\langle \mathcal{M}, w \rangle$ a *pointed model*.

Before moving on, let us see an example of a Kripke model, in order to clarify the concept. In the following model we will give a graphical representation of the domain and the relations of the model. A node represents an element in the domain and an edge from w to w' labeled as R means that wRw' .

Example 1.2.3. Consider the following model $\mathcal{M} = \langle W, (R_r)_{r \in \text{REL}}, V \rangle$:



This model has a domain of four points, $W = \{w_1, w_2, w_3, w_4\}$. The signature in which it is based on is $\langle \text{PROP} = \{p, q\}, \text{REL} = \{r_1, r_2\} \rangle$, that is, it has two modalities, r_1 and r_2 , and two propositional symbols, p and q . We explicitly indicate in the picture the places where the propositional symbols hold. Translated to the valuation function V , that means that $V(p) = \{w_1, w_3\}$ and $V(q) = \{w_2, w_3\}$. Observe that at w_4 no propositional symbol holds.

Now we are ready to define the semantics for the basic modal language, since we already have both the syntax and the structures the language is going to talk about. Recall that modal logics describe Kripke structures *from an internal perspective*. This means that, in contrast with first order logic in which formulas see models from some kind of omniscient lookout point, modal formulas are evaluated *at some particular point* of the model.

Definition 1.2.4. Given the model $\mathcal{M} = \langle W, (R_r)_{r \in \text{REL}}, V \rangle$ and $w \in W$, we inductively define the notion of a formula φ being satisfied (or *true*) in \mathcal{M} at the point w as follows:

$\mathcal{M}, w \models \top$	always
$\mathcal{M}, w \models \perp$	never
$\mathcal{M}, w \models p$	iff $w \in V(p)$ $p \in \text{PROP}$
$\mathcal{M}, w \models \neg\varphi$	iff $\mathcal{M}, w \not\models \varphi$
$\mathcal{M}, w \models \varphi \wedge \psi$	iff $\mathcal{M}, w \models \varphi$ and $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \varphi \vee \psi$	iff $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \varphi \rightarrow \psi$	iff $\mathcal{M}, w \not\models \varphi$ or $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \varphi \leftrightarrow \psi$	iff $\mathcal{M}, w \models \varphi$ if and only if $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \langle r \rangle \varphi$	iff there is a w' such that $wR_r w'$ and $\mathcal{M}, w' \models \varphi$
$\mathcal{M}, w \models [r]\varphi$	iff for all w' such that $wR_r w'$, $\mathcal{M}, w' \models \varphi$

Given a model \mathcal{M} , we say that φ is *globally satisfied* (or *globally true*) on \mathcal{M} , and write $\mathcal{M} \models \varphi$, if for all points w in the domain of \mathcal{M} we have that $\mathcal{M}, w \models \varphi$. A formula φ is *universally valid* if it is globally satisfied in all models, and in that case we write $\models \varphi$. A formula φ is *satisfied in a model* \mathcal{M} when there is a point in \mathcal{M} where φ is true, and φ is *satisfiable* if there is some point in some model at which it is satisfied. When working with sets of formulas, these definitions are lifted in the expected way.

1.3 Model equivalence

Let \mathcal{M} and \mathcal{M}' be two models for a logic \mathcal{L} , and w and w' be two points in \mathcal{M} and \mathcal{M}' respectively. We say that w and w' are \mathcal{L} -equivalent (notation: $w \equiv_{\mathcal{L}} w'$) if they make

the same \mathcal{L} -formulas true.¹ This means that, although the models may be different, if we look at w and w' “through the glasses of the logic \mathcal{L} ” they are indistinguishable. Consider now the following two models.

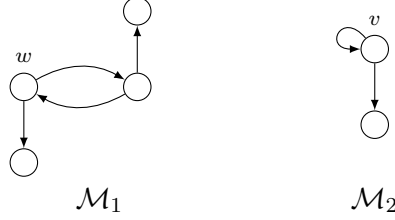


Fig. 1.1: The points w and v are BML equivalent.

Let us consider the basic modal language. Assuming that $V(p) = \emptyset$ in both models for all $p \in \text{PROP}$, is there a way to distinguish w from v in BML? That is, is there a basic modal formula that is true at w and false at v ? It doesn't seem to be easy to find one. On the other hand, if we can use first order logic this is quite straightforward: the formula $\neg R(x, x)$ is true if we assign w to x , and false in the case of v .

Equivalence as a structural notion

One could pick two pointed models $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ and ask whether they are \mathcal{L} -equivalent for a given logic \mathcal{L} without checking every possible formula. For example, in Figure 1.1, we would like to know if there is a *structural* relationship between the models that makes them equivalent for BML.

In classical first-order logic this relationship corresponds to *potential isomorphisms*, which is defined as follows in [CK90].²

Definition 1.3.1 (Potential isomorphism). Let \mathcal{M}^f and \mathcal{N}^f be first order models with domains M and N respectively. A *potential isomorphism* between \mathcal{M}^f and \mathcal{N}^f is a relation Z on the set of pairs of finite sequences $\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_n \rangle$ of elements of A and B of the same length such that:

- (i) $\emptyset Z \emptyset$.
- (ii) If $\langle a_1, \dots, a_n \rangle Z \langle b_1, \dots, b_n \rangle$ then $(\mathcal{M}^f, a_1, \dots, a_n)$ and $(\mathcal{N}^f, b_1, \dots, b_n)$ satisfy the same *atomic* formulas.
- (iii) If $\langle a_1, \dots, a_n \rangle Z \langle b_1, \dots, b_n \rangle$ then for all $c \in M$ there exists $d \in N$ such that $\langle a_1, \dots, a_n, c \rangle Z \langle b_1, \dots, b_n, d \rangle$ and vice versa.

We use $\mathcal{M}^f \cong \mathcal{N}^f$ to note that there exists a potential isomorphism between \mathcal{M}^f and \mathcal{N}^f . Observe that potential isomorphism relations are symmetrical, that is, $\mathcal{M}^f \cong \mathcal{N}^f$ if and only if $\mathcal{N}^f \cong \mathcal{M}^f$.

Given two first-order models $\mathcal{M}^f, \mathcal{N}^f$ if $\mathcal{M}^f \cong \mathcal{N}^f$ then the models are indistinguishable by first-order logic [CK90, Proposition 2.4.4] (they are also called *elementary equivalent*).

¹ When the logic is clear from context we don't add the subscript \mathcal{L} .

² In the literature, such as [CK90], potential isomorphism are sometimes called ‘partial isomorphism’ because they are formed of sequences of isomorphism with restricted domain.

In the modal domain, take the basic modal logic as an example, the notion of *bisimulation* between models gives a structural notion which implies that the models are equivalent when looking through the glasses of basic modal logic [BdRV01, Section 2.2]. For a detailed historical insight on bisimulation we recommend [San09].

Let's take a look at the definition of bisimulation for the basic modal logic. We will give the definition for the monomodal version of BML because its simplicity is suitable for this introduction but all the definitions and results of this chapter also hold for the multimodal case.

Definition 1.3.2 (Bisimulation for BML). A *bisimulation* between two BML models $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$ is a non-empty binary relation $Z \subseteq W \times W'$ between their domains such that whenever wZw' we have that:

Atomic harmony: w and w' satisfy the same propositional symbols.

Forth: if wRv , then there exists a point v' in \mathcal{M}' such that vZv' and $w'R'v'$.

Back: if $w'R'v'$, then there exists a point v in \mathcal{M} such that vZv' and wRv .

If there is a bisimulation between two models \mathcal{M} and \mathcal{M}' we say that \mathcal{M} and \mathcal{M}' are bisimilar and we write $\mathcal{M} \simeq \mathcal{M}'$. Moreover, we say that two points $w \in \mathcal{M}$ and $w' \in \mathcal{M}'$ are bisimilar if they are related by some bisimulation, and we write $\mathcal{M}, w \simeq \mathcal{M}', w'$. We write $w \simeq w'$ when the models are clear from context.

Returning to the models \mathcal{M}_1 and \mathcal{M}_2 we have just presented in Figure 1.1, it is easy to see that $\mathcal{M}_1, w \simeq \mathcal{M}_2, v$. The bisimulation would be as follows (the dotted line indicates the pairs in the bisimulation relationship):

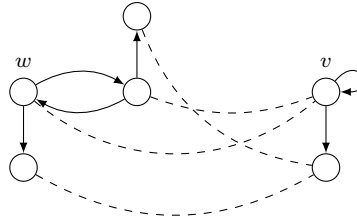


Fig. 1.2: Bisimilar models.

The definition of bisimulation we just gave is specifically designed for the basic modal logic, and thus the expected property is that satisfiability of formulas in the basic modal logic is invariant under bisimulations as proved in [BdRV01].

Theorem 1.3.3. Let \mathcal{M} and \mathcal{M}' be two Kripke models over the same signature. Then, for every $w \in \mathcal{M}$ and $w' \in \mathcal{M}'$, if $w \simeq w'$ then for every formula φ of BML, $\mathcal{M}, w \models \varphi$ if and only if $\mathcal{M}', w' \models \varphi$.

The aforementioned logics, namely BML and first order logic, have both negation and disjunction in their languages and both “model equivalence” notions (bisimulation and potential isomorphisms) are symmetrical. In many areas of computer science one finds logical formalisms that lack some of the standard Boolean connectives ‘and’, ‘or’ and ‘not’. In particular, negation-free logics are widely used in areas as diverse as semantics of programming and knowledge representation. In some applications, such as the generation of referring expressions [AKS08], Boolean negation may be unnatural.

Take now the basic sub-boolean modal logic, BML^- , which is defined as BML but *doesn't* have negation nor the \Box modality. As the language is weaker, the notion of model equivalence should change. Bisimulations, for example, are too strong for negation-free logics because they preserve negation. In [BdRV01, Definition 2.77] we can find the definition of the concept of *simulation* for negation-free logics. If there is a simulation from \mathcal{M}, w to \mathcal{N}, v then every formula true at \mathcal{M}, w is also true at \mathcal{N}, v . The formal definition is as follows.

Definition 1.3.4 (Simulation for BML^-). A *simulation* between two models $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$ is a non-empty binary relation $Z \subseteq W \times W'$ between their domains such that whenever wZw' we have that:

Atomic condition: If $w \in V(p)$ then $w' \in V(p)$ for all $p \in \text{PROP}$.

Forth: if wRv , then there exists a point v' in \mathcal{M}' such that vZv' and $w'R'v'$.

If there is a simulation between two models \mathcal{M} and \mathcal{M}' we write $\mathcal{M} \rightrightarrows \mathcal{M}'$. Moreover, we say that two points $w \in \mathcal{M}$ and $w' \in \mathcal{M}'$ are similar if they are related by some simulation, and we write $\mathcal{M}, w \rightrightarrows \mathcal{M}', w'$. We write $w \rightrightarrows w'$ when the models are clear from context.

Observe that, in this case, half of the “Atomic harmony” condition has been removed. Another point to be taken into account is that even though BML 's bisimulation is symmetrical, simulations need not to be. This notion is specially suited for BML^- and preserves every formula formed from \wedge, \vee and \diamond . The following theorem states this formally.

Theorem 1.3.5. Let \mathcal{M} and \mathcal{M}' be two Kripke models over the same signature. Then, for every $w \in \mathcal{M}$ and $w' \in \mathcal{M}'$, if $w \rightrightarrows w'$ then for every formula φ of BML^- ; $\mathcal{M}, w \models \varphi$ implies $\mathcal{M}', w' \models \varphi$.

As the notion of simulation is less restrictive than the notion of bisimulation it should be no surprise to find models which are similar but not bisimilar. Take, for example, the following two models.

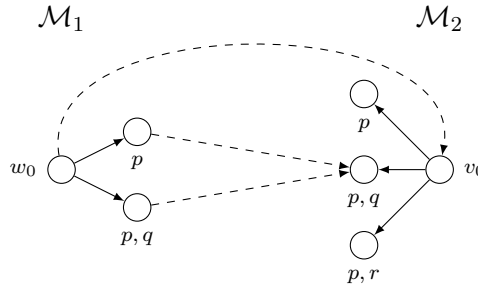


Fig. 1.3: Similar but not bisimilar.

Again, the dashed lines indicate the pairs in the simulation relation. We can see, that $\mathcal{M}_1, w_0 \rightrightarrows \mathcal{M}_2, v_0$, on the other hand, there is no bisimulation linking them. To show this, it is enough to exhibit a formula φ such that $\mathcal{M}_2, v_0 \models \varphi$ and $\mathcal{M}_1, w_0 \not\models \varphi$. In this case, a possible formula is $\diamond r$.

Equivalence as a game

These notions of model equivalence can also be presented using a more dynamic perspective, closer to a form of process equivalence. For example, the task of determining whether two models are bisimilar can be recast in the form of an Ehrenfeucht-Fraïssé game [EFT84].

Let $\langle \mathcal{M}_1, w_1 \rangle$ and $\langle \mathcal{M}_2, w_2 \rangle$ be two pointed models. An *Ehrenfeucht-Fraïssé game for the basic modal logic* is defined as follows. There are two players called *Spoiler* and *Duplicator*. Intuitively, Spoiler tries to devise a property true in one model and false in the other. Conversely, Duplicator tries to ‘copy’ the property from one model to the other by imitating Spoiler’s movements.

The two players compare successive pairs, starting from (\mathcal{M}_1, w_1) and (\mathcal{M}_2, w_2) . Duplicator immediately loses if w_1 and w_2 do not coincide in the propositional symbols. Otherwise, the game starts, with the players moving alternatively. Spoiler always makes the first move of the game. In a turn of the game, Spoiler starts by choosing in which model he will make a move. After that, he chooses a point which is a successor of the current w_1 or w_2 , and Duplicator responds with a matching successor in the other model. If the chosen points differ in the atomic propositions, Spoiler wins. If one player cannot move, the other wins. Duplicator wins on infinite runs.

Note that with this definition, exactly one of Spoiler or Duplicator wins each game. A *strategy for Duplicator* is a function that takes a valid state of the game (i.e. a pair $\langle a, b \rangle$ with $a \in |\mathcal{M}_1|$ and $b \in |\mathcal{M}_2|$) and returns a possible next move for Duplicator. A strategy for Spoiler is defined in the same way but note that the function should also return the model in which Spoiler should make the move. We say that a player is *following a strategy* σ when all his moves in a game comply with the answer of σ for every stage of the game. A strategy is *winning* if the player following it necessarily wins the game, no matter what his opponent plays. Given two pointed models $\langle \mathcal{M}_1, w_1 \rangle$ and $\langle \mathcal{M}_2, w_2 \rangle$ we will write $\langle \mathcal{M}_1, w_1 \rangle \equiv_b \langle \mathcal{M}_2, w_2 \rangle$ when Duplicator has a winning strategy for the game.

Intuitively, this game captures exactly the zigzag behavior of bisimulations, and the atomic harmony condition. The two notions are equivalent, but depending on the context, one can be more natural than the other.

Proposition 1.3.6. [GO05] Let $\langle \mathcal{M}_1, w_1 \rangle$ and $\langle \mathcal{M}_2, w_2 \rangle$ be two BML pointed models, then $\langle \mathcal{M}_1, w_1 \rangle \equiv_b \langle \mathcal{M}_2, w_2 \rangle$ if and only if $\langle \mathcal{M}_1, w_1 \rangle \Leftrightarrow \langle \mathcal{M}_2, w_2 \rangle$.

The perspective of model equivalence as a game is not restricted to the basic modal logic. With minor modifications to this notion of game we can create a notion that is suitable for BML $^-$, for instance. An *Ehrenfeucht-Fraïssé game for BML $^-$* is the same as the game for the basic modal logic *but* spoiler can’t choose the model where he is playing. That is, Spoiler starts playing in \mathcal{M}_1 and he won’t be able to change to \mathcal{M}_2 .

Suppose that Spoiler and Duplicator start a game standing in (\mathcal{M}_1, w_1) and (\mathcal{M}_2, w_2) respectively. If Duplicator has a winning strategy then $\mathcal{M}_1, w_1 \Rightarrow \mathcal{M}_2, w_2$. On the other hand, if Spoiler has a winning strategy then $\mathcal{M}_1, w_1 \not\Rightarrow \mathcal{M}_2, w_2$. We will write $\langle \mathcal{M}_1, w_1 \rangle \equiv_s \langle \mathcal{M}_2, w_2 \rangle$ when Duplicator has a winning strategy for the game. The following proposition states the equivalence between BML $^-$ simulation and the game definition.

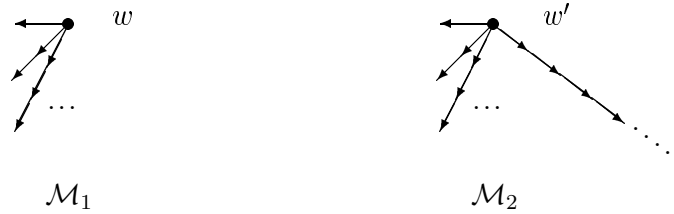
Proposition 1.3.7. Let $\langle \mathcal{M}_1, w_1 \rangle$ and $\langle \mathcal{M}_2, w_2 \rangle$ be two BML $^-$ pointed models, then $\langle \mathcal{M}_1, w_1 \rangle \equiv_s \langle \mathcal{M}_2, w_2 \rangle$ if and only if $\langle \mathcal{M}_1, w_1 \rangle \Rightarrow \langle \mathcal{M}_2, w_2 \rangle$.

Summing up, simulations and bisimulations are very powerful tools to measure the expressivity of a logic: they provide us with *structural* conditions on the models that char-

acterizes the appropriate structure preserving morphisms. Since simulations are directly linked to the expressivity of a given logic, there is not a *unique* notion of simulation. Here we have just presented the notion of bisimulation for BML and simulation for BML $\bar{}$, but for *every* logic we need to find a suitable definition, and this notion will reflect the logic's expressive power. In this sense, looking for the appropriate notion of model equivalence allows us to learn about the logic we are working with.

1.4 Saturation

We have discussed a lot about model equivalence notions, in particular about simulations and bisimulations, but we have been avoiding a fundamental question. Let's focus on bisimulations, we know that $\mathcal{M}, w \Leftrightarrow \mathcal{N}, v$ implies $\mathcal{M}, w \equiv \mathcal{N}, v$. Does, in general, the converse hold? That is, does $\mathcal{M}, w \equiv \mathcal{N}, v$ imply $\mathcal{M}, w \Leftrightarrow \mathcal{N}, v$? The answer is *no*. Consider the following two models:



It can be shown that, although w and w' satisfy the same BML formulas, there is no possible bisimulation between them. Recall that in Section 1.3 we presented an alternative interpretation of bisimulations as games. We will now use that notion to prove that these models are not bisimilar.

Set a game between Duplicator and Spoiler with them starting at (\mathcal{M}_1, w) and (\mathcal{M}_2, w') respectively. The first turn is for Spoiler. He chooses to stay in \mathcal{M}_2 and move to the successor of w' that lays in the infinite branch of the model. Now it is Duplicator's turn, he must move to a matching world in \mathcal{M}_1 . As the *atomic harmony* condition is trivially satisfied by any two pairs of these models, the only problem could arise if Spoiler makes a move and Duplicator has no possible successors to move to. Duplicator has to choose a branch in \mathcal{M}_1 and move to it. Suppose, without loss of generality, that Duplicator chooses a branch with k nodes. You can easily see that, as there is no going back, if Spoiler carries on moving in his infinite branch there will be a moment (after k moves) when Duplicator hits the end of his branch in \mathcal{M}_1 . In that moment Spoiler wins the match.

The strategy we've described guarantees that Spoiler will win no matter what Duplicator chooses. Therefore, as Spoiler has a winning strategy the models are not bisimilar. In fact, as Spoiler starts the game in \mathcal{M}_2 and never changes the model this argument proves that $\mathcal{M}_2, w' \not\equiv \mathcal{M}_1, w$ which is a stronger result.

Do not panic. There *are* some classes of models where modal equivalence implies bisimilarity. A very useful one is the class of ω -saturated models. In order to present this class we need some previous definitions.

The following notions will be given in terms of first order models and not BML models but this shouldn't carry any problem. Later, in Section 4.2, we will see that there is a straightforward formalization that lets us think of a model as a BML or first order model interchangeably.

Notation 1.4.1. We will use $\varphi(x)$ to note first order formulas with at most one free variable x this notation extends to sets $\Gamma(x)$ as expected. The notation $g[x \mapsto w]$ denotes a valuation g' that is the same as g on every parameter except on x where $g'(x) = w$. Given a first order formula $\varphi(x)$ we will hereafter note $\mathcal{M}^f \models \varphi(x)[w]$ to mean $\mathcal{M}^f, g[x \mapsto w] \models \varphi(x)$. Observe that, as φ has only one free variable x the valuation will be irrelevant. Given a model \mathcal{M} we use $|\mathcal{M}|$ to denote the domain (or universe) of \mathcal{M} .

Definition 1.4.2. A set of first order formulas with (at most) one free variable is called a *type*. Given a model \mathcal{M}^f we say that a type $\Gamma(x)$ has a *witness* if there exists a state w such that for every formula $\varphi(x) \in \Gamma(x)$ we have $\mathcal{M}^f \models \varphi(x)[w]$. A type is *finitely realizable* if every finite subset has a witness.

Definition 1.4.3 (Expansion). Let \mathcal{M}^f be a first order model with domain W . For $A \subseteq W$, the *expansion* of \mathfrak{F} with A (noted $\mathfrak{F}[A]$) is obtained by extending \mathfrak{F} with new constants \underline{a} for every element $a \in A$. The model \mathcal{M}_A^f is the same as \mathcal{M}^f but interprets the constants as expected.

We are now ready to define ω -saturation. Informally, it resembles some kind of ‘intra-model’ compactness. That is, given a type $\Gamma(x)$ if every finite subset is satisfied in (possibly different) elements in \mathcal{M}^f then there is a *single element* in \mathcal{M}^f which satisfies the whole set. Formally speaking the definition is as follows.

Definition 1.4.4 (ω -saturation). A first order model \mathcal{M}^f is called *ω -saturated* if for every finite $A \subseteq |\mathcal{M}^f|$ the expansion \mathcal{M}_A^f has a witness for every type $\Gamma(x)$ that is finitely realizable in \mathcal{M}_A^f .

In the beginning of this section we presented two models. One of them had branches of increasing length, the other one was an exact copy of the first but with an extra *infinite* branch. We have already seen that, in some sense, the first model was ‘lacking’ something that the second one had. The saturation that ω -saturated models have make them complete in this sense. The following theorem is a very important result which gives strength to the class of ω -saturated models.

Theorem 1.4.5 ([BdRV01]). Let $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ be two ω -saturated models,

$$\text{If } \mathcal{M}, w \equiv \mathcal{N}, v \text{ then } \mathcal{M}, w \Leftrightarrow \mathcal{N}, v.$$

Therefore, in the ω -saturated class, bisimulation and BML equivalence coincide. This proof strongly uses the structural definition of bisimulations and thus we will not get into this kind of detail until we need it in Chapter 5.

A particularly interesting fragment of the ω -saturated class is the *finitely branching* fragment, that is, every world has only finitely many successors. Another different (and more restrictive) example of ω -saturated class is the class of finite models.

To finish this section we want to say some final words about ω -saturated models. The use of ω -saturated models will be crucial to prove the results in this thesis. Because of their special properties one could think that these models are rather scarce but, fortunately, they abound. Moreover, there is a standard way of, given an \mathfrak{F} -model \mathcal{M}^f , construct an ω -saturated model \mathcal{M}_*^f such that $\mathcal{M}^f \equiv_{\mathfrak{F}} \mathcal{M}_*^f$. This theorem is stated as Theorem B.7 and proved in the Appendix.

1.5 What this thesis is about

For a wide spectrum of applications, which use logic as a tool, first order logic is enough to *theoretically* solve their problems. However, complications arise when we consider the behavior of first order logic in *practice*. First of all, first order logic is *undecidable*, that is, there is no algorithm to decide whether an arbitrary formula is satisfiable. Second, in general, most applications do not use the entire expressive power that first order gives. Therefore, even when working in decidable fragments of first order logic, they may be paying an excessive payload for things they will not be using.

Modal logics are very good at molding themselves to fit a particular purpose. If you know what you need it is most likely that you can end up with a modal logic which has exactly the required expressive power but with better properties than first order logic in terms of complexity and decidability. For example, BML is decidable and has a PSPACE-complete satisfiability problem.

Along this thesis we will work with logics that are less (or equally) expressive than first order logic. Informally, this means that if one can express a property with a modal formula then there is always some way to express the same property in first order. In other words, one can say that if a modal formula φ denotes some property then there exists some kind of *translation* to a first order formula φ^t which denotes the same property.

Johan van Benthem studied the connection between modal and first order logic [vB84]. One of his best known results in this area is the ‘Characterization Theorem’ which identifies BML as the bisimulation-invariant fragment of first order logic. Informally, one can state the theorem as follows.

Theorem. A first order formula α is equivalent to the translation of a BML formula if and only if α is invariant under bisimulations.

Note that in this case the notion of bisimulation is that of BML. As we have said before, every modal logic should have a potentially different notion of bisimulation. For example, we have already seen the notion of simulation for BML. Using this notion, Kurtonina and de Rijke proved that BML is the simulation-invariant fragment of first order logic.

BML is just the tip of the iceberg, there exist plenty of extensions of BML to suit particular needs. Many modal logics admit a translation to first order logic and a characterization of this kind has been given for some of them. One essential problem is that there seems to be no general proof and every case needs a new ad-hoc proof.

The birth of the concept of bisimulation and correspondence theory helped answer new questions from a purely model-theoretic perspective. One example is the characterization of modal *definability*. Informally, we say that a class of models is definable by a set of formulas Γ if it is composed of exactly all the models where Γ is valid. A class is definable by a single formula if it is definable by a singleton set.

It would be interesting to know which properties should a class of models satisfy in order to be defined by a modal formula or by a set of modal formulas. This question had previously been stated and answered for classical first order logic [CK90]. Whereas the answer for first order logic is presented in terms of *potential isomorphisms*, in the case of modal logics, the notion of bisimulation plays an essential role. To uncover the panorama we cite the following result for BML which can be found in [BdRV01].

Theorem. A class of models K is definable by means of a single BML formula if and only if both K and \bar{K} are closed under bisimulations and ultraproducts.

Do not worry about what ‘ultraproducts’ means right now. They will be introduced when needed. Just bear in mind that they are useful model construction tools (with algebraic roots) which first caught the attention of logicians because they could be used to give a purely algebraic proof of the Compactness Theorem for first order logic. For a detailed survey on ultraproducts we recommend [Kei08].

As with the characterization theorem, definability results similar to the one presented here also hold for a vast number of modal logics. Similarly, every logic has a proof that is specially crafted for that case.

Clearly, characterization and definability results help us to better understand a logic. Interestingly, these results also have a great impact in practical computer science.

Consider the following problem: Suppose you are into model checking, that is, given a model of a system, test automatically whether this model meets a given specification. Suppose that the specification can be written as a first-order formula φ .

You could always use first-order tools to check if the model satisfies φ but that can result in a high complexity cost as we have already mentioned. One could try to see if there are ‘cheaper’ logics that can be used to solve the problem. If we can express the same property in some modal logic we may be able to drastically optimize the process.

Let’s discuss a concrete example. Suppose that the points in our domain model the different states in the execution of a program and there is a transition from one point to another if there is a possible transformation that brings state a into b . Thinking of the model in this way would imply that states without successors (also called endpoints) are states where the program has halted.

One possible property to be checked could be “every point should be able to immediately ‘escape’ from the flow of execution”, meaning that, every point should be able to directly see an endpoint. This property can be verified by checking that the first order formula

$$\varphi(x) = \exists y.R(x, y) \rightarrow (\exists z.R(x, z) \wedge \forall w.\neg R(z, w))$$

is valid in the model but it can also be checked by verifying that the BML formula $\psi = \diamond\top \rightarrow \diamond\Box\perp$ is valid in the model. As we have an equivalent BML formula, we can now use model checkers that accept BML formulas to solve our problem.

Apart from being more ‘user friendly’, the sole existence of the formula ψ tells us that the property is invariant under bisimulations and this information bears an extra benefit. Suppose that the model is automatically generated from a piece of code. If, for example, we feed the generator with the code of an entire operating system, the resulting model will be *very* large.

It is not the purpose of this thesis to get into this topic but there are (efficient) algorithms to automatically minimize the model which preserve the truth of formulas invariant under bisimulations [Hop71, Gri73]. Therefore one could first minimize the model and then model check over the resulting model which will most likely be small with respect to the original one.

On the other hand, suppose now that we want to check whether the model is ‘irreflexive’, that is, no element is related with itself. If we interpret this property in the setting described above, it would mean that no state has the possibility to ‘hang’ in itself.

For this case, although the property can be verified checking the validity of the first order formula $\neg R(x, x)$ in the model, there is no BML formula which does the job. This can be shown easily because ‘irreflexivity’ is *not* invariant under bisimulations. Moreover,

BML has the so-called *tree model property* which means that every formula satisfiable in a model is also satisfiable in a model which is a tree. As a corollary we get that there is no BML-formula characterizing irreflexivity, antisymmetry nor intransitivity.

Is this the end of modal logic? Are we condemned to use first-order logic in this case? Fortunately, the answer is *no*. Although BML can't express those properties, there are richer logics (which still lay below first order) which can do the job. For example, Hybrid Logics [BdRV01] and the less known Memory Logics [AFFM08] which will be introduced later in this thesis.

To summarize, model theoretic results such as Characterization and Definability give important information about different logics. It is well known that the proofs of those results for several modal logics have, somehow, the same 'taste'. A general proof for most modal logics below first order is still too ambitious. In this thesis we plan to isolate sufficient conditions for the characterization and definability theorems to hold in a wide range of logics. Along with these conditions we will prove that, whichever logic that meets them, satisfies both theorems. Therefore, one could give an unifying proof for logics with already known results. Moreover, one will be able to prove characterization and definability results for logics that have not yet been investigated. In both cases, it is only needed to check that a logic meets the requirements to automatically derive the desired results.

2. KNOWN RESULTS FOR BML

If we want to generalize a result we'd better understand how it works in specific cases. This chapter is devoted to sketching the proof of some theorems for BML. This will be helpful to identify the main ideas in their proofs and, with them in mind, get ready to undertake a generalization.

2.1 Characterization

We have talked about van Benthem's characterization theorem. We know that BML is strictly less expressive than first order logic, therefore, there are some 'statements' that you can make in first order logic which can't be made in BML. Informally, the Characterization Theorem identifies which first order formulas have an equivalent formula in the language of BML. More formally it is stated as follows.

Theorem (van Benthem). A first order formula $\alpha(x)$ with at most one free variable is equivalent to the translation of a BML formula if and only if $\alpha(x)$ is invariant under bisimulations.

Some work is needed for this wording to be precise. First of all, we are comparing modal formulas with first order formulas. Also, implicitly, when we talk about two formulas being 'equivalent', we are evaluating them in some model. That's a problem because BML formulas are evaluated in Kripke models and first order formulas aren't.

For us to be able to do such comparison between BML and first order logic we need to define a formula translation and a way to interpret every BML model as a first order model and vice-versa.

For this chapter we will set the signature for BML to be $\mathcal{S} = \langle \text{PROP}, \text{REL} \rangle$ with $\text{PROP} = \{p_1, p_2, \dots\}$ and $\text{REL} = \{R\}$ therefore we will use a single diamond. This restriction to the unimodal case is only to make this introduction simpler. All these results also hold for the multimodal case.

Definition 2.1.1 (Standard Translation). The *Standard Translation* function ST_x takes a BML formula and returns a first order formula with at most one free variable x . It is defined as follows.

$$\begin{aligned}
 ST_x(p_i) &= P_i(x) && \text{where } p_i \in \text{PROP} \\
 ST_x(\neg\varphi) &= \neg ST_x(\varphi) \\
 ST_x(\varphi \wedge \psi) &= ST_x(\varphi) \wedge ST_x(\psi) \\
 ST_x(\diamond\varphi) &= \exists y(xRy \wedge ST_y(\varphi)) && \text{where } y \text{ is a fresh variable}
 \end{aligned}$$

We only define the translation for a basic (and adequate) connective set. It extends to the full set of connectives as expected.

In this definition, we can already see that first order formulas include relations P_i and R , this may give us a hint to define the first order signature. A first order signature is a tuple $\langle \text{FREL}, \text{FCONST}, \text{FFUNC} \rangle$ where FREL is the set of relation symbols, FCONST is the set of constant symbols and FFUNC is the set of function symbols. In our case we define the first-order signature to be $\mathcal{F} = \langle \{R, P_1, P_2, \dots\}, \emptyset, \emptyset \rangle$.

Definition 2.1.2 (First order model). A *first order model* over the signature \mathcal{F} is a tuple

$$\mathcal{M}^f = \langle A, (R^I)_{R \in \text{FREL}}, (f^I)_{f \in \text{FFUNC}}, (c^I)_{c \in \text{FCONST}} \rangle$$

where A is the (non empty) domain, each R^I is the interpretation of the relation symbol R , each f^I is the interpretation for the function symbol f and each c^I is the interpretation for the constant symbol c .

In general we add a superscript or subscript f to first order models so it is easier to distinguish them from modal models at first sight. We use e, w, v, \dots to refer to elements of the domain of some model and g, h, \dots to refer to first order valuations.

The crucial point now is to see that there is a bijection between BML models over the signature \mathcal{S} and first order models over the signature \mathcal{F} . Given a BML model $\mathcal{M} = \langle W, R, V \rangle$ we can think of it as a first order model defined as

$$\mathcal{M}^f = \langle W, \{R, P_1, P_2, \dots\}, \emptyset, \emptyset \rangle$$

where $P_i = V(p_i)$. With this definition one can easily see that, given a BML model \mathcal{M} and $w \in W$; $\mathcal{M}, w \models p_i$ if and only if $\mathcal{M}^f \models P_i(x)[w]$.

On the other hand, observe that *any* first order model in this signature should be of the form $\mathcal{M}^f = \langle W, \{R, P_1, P_2, \dots\}, \emptyset, \emptyset \rangle$ and one can therefore build a BML model analogously.

As it is usual in the literature, we will use, for this chapter only, the same model and think of it as a BML or first order model as convenient. Now we can state the theorem that links BML with first order.

Theorem 2.1.3 (Truth preservation). Let \mathcal{M} be a BML model, $w \in |\mathcal{M}|$ and φ be a BML formula,

$$\mathcal{M}, w \models \varphi \text{ if and only if } \mathcal{M} \models ST_x(\varphi)[w].$$

This theorem states that for every BML formula there is a first order formula which is true in exactly the same worlds, thus, they are equivalent. Now that we have this theorem at hand it becomes clearer that we can compare formulas and models between BML and first order logic.

The Characterization Theorem is stated in terms of ‘bisimulations’ and uses notions we haven’t yet defined. To begin with, we copy the definition of BML bisimulation given in Section 1.2.

Definition 2.1.4 (Bisimulation). A *bisimulation* between two models $\mathcal{M} = \langle W, R, V \rangle$ and $\mathcal{M}' = \langle W', R', V' \rangle$ is a non-empty binary relation $Z \subseteq W \times W'$ between their domains such that whenever wZw' we have that:

Atomic harmony: w and w' satisfy the same propositional symbols.

Forth: if wRv , then there exists a point v' in \mathcal{M}' such that vZv' and $w'R'v'$.

Back: if $w'R'v'$, then there exists a point v in \mathcal{M} such that vZv' and wRv .

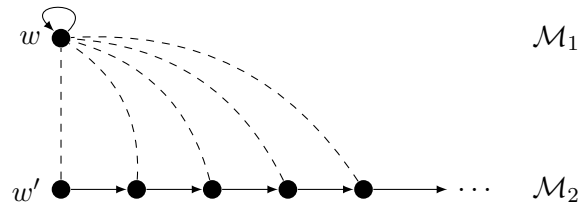
In particular, the theorem talks about first order formulas being *bisimulation-invariant*. Van Benthem defines this concept as follows:

Definition 2.1.5 (Bisimulation invariance). A first order formula $\alpha(x)$ is *invariant for bisimulations* if for all BML models \mathcal{M}, \mathcal{N} and $w \in |\mathcal{M}|, v \in |\mathcal{N}|$ such that $\mathcal{M}, w \Leftrightarrow \mathcal{N}, v$ the following holds:

$$\mathcal{M} \models \alpha(x)[w] \text{ iff } \mathcal{M} \models \alpha(x)[v].$$

Observe that, so far, we only know that BML formulas *are* invariant for bisimulations and we don't have a result regarding first order formulas. The result for BML formulas was stated in Theorem 1.3.3. On the other hand, when talking about first order formulas, some may be invariant for bisimulations and some others not. The set of formulas that are invariant for bisimulations is exactly the one identified by the characterization theorem.

As an example, take the following two BML-bisimilar models. The first model is a single reflexive point and the second one is isomorphic to $\langle \mathbb{N}, < \rangle$.¹ The dashed lines represent the pairs in the bisimulation relation.



Now take the first order formula $\varphi(x) = R(x, x)$. This formula holds at an element of the domain if and only if it is reflexive. It is clear that $\mathcal{M}_1 \models \varphi(x)[w]$ and $\mathcal{M}_2 \not\models \varphi(x)[w']$. As there is a bisimulation $w \Leftrightarrow w'$, this two models serve as a proof that reflexivity is *not* invariant under bisimulations.

This also means that there is no possible BML formula equivalent to $\varphi(x)$. Suppose that there exists a BML formula ψ whose translation is equivalent to φ . By Theorem 2.1.3 we have that $\mathcal{M}_1, w \models \psi$ iff $\mathcal{M}_1 \models \varphi(x)[w]$ and $\mathcal{M}_2, w' \models \psi$ iff $\mathcal{M}_2 \models \varphi(x)[w']$. We can conclude that $\mathcal{M}_1, w \models \psi$ and $\mathcal{M}_2, w' \not\models \psi$. This contradicts Theorem 4.2 because, as it is a BML formula, ψ should be invariant under BML bisimulations.

We have proved that as $\varphi(x)$ is not invariant under bisimulations it is not equivalent to the translation of any BML formula. What we have done for one particular case, the characterization theorem proves for an arbitrary first order formula. Moreover, it also proves the converse. Now that we understand what we are trying to prove we are ready to begin with the proof itself.

The proof of the Characterization Theorem

In this section we will skim through the proof of van Benthem's Characterization Theorem. It is not the goal of this section to give a detailed proof but to review the main ideas that support it. For a detailed proof refer to [BdRV01, Section 2.6].

Theorem. A first order formula $\alpha(x)$ with at most one free variable is equivalent to the translation of a BML formula if and only if $\alpha(x)$ is invariant under bisimulations.

¹ Therefore BML can't distinguish between a single reflexive point and the naturals. It is surprising that, as weak as it is, BML is still useful in practice.

Left to right. This direction is easy, we argue by contradiction. Suppose that $\alpha(x)$ is equivalent to the translation of a BML formula φ and it is not invariant under bisimulations. That is, there exist $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ such that $\mathcal{M}, w \Leftrightarrow \mathcal{N}, v$ but $\mathcal{M} \models \alpha(x)[w]$ and $\mathcal{N} \not\models \alpha(x)[v]$.

Using Theorem 2.1.3 we get that $\mathcal{M}, w \models \varphi$ and $\mathcal{N}, v \not\models \varphi$. As we have a bisimulation linking those points and φ is a BML formula this drives us to a contradiction to Theorem 1.3.3. Absurd. \square

Right to left. All the magic is in the proof of this direction. Suppose that $\alpha(x)$ is invariant under bisimulations. Define the ‘modal consequences of α ’ as follows.

$$MOC(\alpha) = \{ST_x(\varphi) : \varphi \text{ is a BML formula and } \alpha(x) \models ST_x(\varphi)\}$$

It is trivial (by definition) that $\alpha(x) \models MOC(\alpha)$. As $MOC(\alpha)$ is formed by the translation of BML formulas, if we can show that $MOC(\alpha) \models \alpha(x)$ then we are done. We first sketch the proof for this statement and then carry on.

Suppose that $MOC(\alpha) \models \alpha(x)$, by compactness of first order logic there exists a finite subset $\Delta \subseteq MOC(\alpha)$ such that $\Delta \models \alpha(x)$. We therefore have $\models \bigwedge \Delta \leftrightarrow \alpha(x)$. As every formula in Δ is the translation of a BML formula and $ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi)$ we can conclude that $\bigwedge \Delta$ is also the translation of some BML formula. Therefore we have proved that $\alpha(x)$ is equivalent to the translation of a BML formula.

Hence, it all boils down to proving that $MOC(\alpha) \models \alpha(x)$. Assume that an arbitrary model satisfies $\mathcal{M} \models MOC(\alpha)[w]$; we need to show that $\mathcal{M} \models \alpha(x)[w]$. The proof goes as follows (we now focus on the ideas and then provide more detailed steps):

1. We first ‘create’ a new model $\langle \mathcal{N}, v \rangle$ such that $\mathcal{M}, w \equiv \mathcal{N}, v$ and $\mathcal{N}, v \models \alpha(x)$. We would like to transfer the validity of $\alpha(x)$ in \mathcal{N}, v to \mathcal{M}, w .
2. Using standard model theoretic tools (that will be explained later) we construct, for \mathcal{M}, w and \mathcal{N}, v , ω -saturated extensions \mathcal{M}^*, w^* and \mathcal{N}^*, v^* which are elementary equivalent to their originators. That is, they have the same first-order theory and they are ω -saturated. Observe that this implies $\mathcal{M}^*, w^* \equiv \mathcal{N}^*, v^*$ and $\mathcal{N}^* \models \alpha(x)[v^*]$.
3. Using Theorem 1.4.5 seen in Chapter 1, as $\mathcal{M}^*, w^* \equiv \mathcal{N}^*, v^*$ (and they are saturated) we have that $\mathcal{M}^*, w^* \Leftrightarrow \mathcal{N}^*, v^*$.
4. Finally, as $\mathcal{N}^* \models \alpha(x)[v^*]$ and $\alpha(x)$ is invariant under bisimulations we get that $\mathcal{M}^* \models \alpha(x)[w^*]$. As \mathcal{M}^* has the same first order theory that its originator we conclude that $\mathcal{M} \models \alpha(x)[w]$. \square

That’s it! Those are the main points to bear in mind. The whole idea is to make a ‘detour’ through the class of first order ω -saturated models where bisimulation and equivalence do coincide. We can now proceed with the dissection of each point.

For the first point we do the following. Define the set Γ as the translation of the BML theory of \mathcal{M}, w , formally speaking

$$\Gamma = \{ST_x(\varphi) : \varphi \text{ is a BML formula and } \mathcal{M} \models ST_x(\varphi)[w]\}.$$

We claim that $\Gamma \cup \{\alpha(x)\}$ is satisfiable. Suppose not, by first order compactness there is an unsatisfiable finite subset $\Gamma_0 \subseteq \Gamma \cup \{\alpha(x)\}$. Observe that $\Gamma_0 = \{\alpha(x), \gamma_1, \dots, \gamma_n\}$

should include $\alpha(x)$. If it is unsatisfiable it means that $\models \alpha(x) \rightarrow \neg \bigwedge \gamma_i$. Hence, $\neg \bigwedge \gamma_i \in \text{MOC}(\alpha)$ because it is a modal consequence of $\alpha(x)$. Remember that one of our hypothesis was $\mathcal{M} \models \text{MOC}(\alpha)[w]$ therefore $\mathcal{M} \models \neg \bigwedge \gamma_i$ but this is impossible since every formula in Γ was true at \mathcal{M}, w by definition. Absurd.

As $\Gamma \cup \{\alpha(x)\}$ is satisfiable we can say that there exists a model \mathcal{N} and an element $v \in |\mathcal{N}|$ such that $\mathcal{N} \models \Gamma[v]$ and $\mathcal{N} \models \alpha(x)[v]$.

For the next three points the explanation given above should suffice. For further details we give the references where the theorems that we used are proved. For the second point we use Theorem B.7 of the Appendix on \mathcal{M} and \mathcal{N} and conclude exactly what we need. The theorems needed for the third point are already mentioned in the enumeration so there is nothing to add. The last point is the *grande finale* where the validity of $\alpha(x)$ is transferred over the bisimulation to end up in \mathcal{M}, w .

Observe that this proof works for BML which is a logic that has negation and disjunction. For negation-free logics the proof needs to be changed a little and for logics lacking disjunction the proof really changes a lot. In [KdR97] you can find proofs for these languages.

In Chapter 3 we propose a framework to generalize this proof. Although the proof developed in this chapter looks pretty simple, in every step it makes use of a lot of suppositions that we may not be aware of. In a general scenario we will be working with an arbitrary translation, an arbitrary signature, an almost unknown model structure, etc. Because of the amount of uncertainty that we will have, we will need to do more complex detours to take the flux of the proof to some better known landscape. Be sure to remember this proof when reading Chapter 3 and 4. Going back and forth may be useful to understand the motivation for some definitions.

2.2 Definability

In Section 1.5 we presented a piece of one of the Definability results for BML. In this section we start by defining the concepts needed to state the full result.

Notation 2.2.1. Let K be a class of models we write \bar{K} to denote the complement of K with respect to the class of all models. This notation will be used for both modal and first order models.

Definition 2.2.2. Let K be a class of pointed BML models.

- (i) K is *definable by a set of formulas* Γ if and only if for every $\langle \mathcal{M}, w \rangle$ it holds that $\mathcal{M}, w \models \Gamma$ if and only if $\langle \mathcal{M}, w \rangle \in K$.
- (ii) K is *definable by a single formula* if it is definable by a singleton set.

Theorem 2.2.3. Let K be a class of BML models.

- (i) K is definable by means of a *set* of BML formulas if and only if K is closed under ultraproducts and bisimulations and \bar{K} is closed under ultrapowers.
- (ii) K is definable by means of a *single* BML formula if and only if both K and \bar{K} are closed under bisimulations and ultraproducts.

As before, we have to define what the closure condition mean for this theorem to make sense. We start by giving an informal introduction to ultraproducts. For this section it is enough to think of ultraproducts as follows: Given a family of first order models $(\mathcal{M}_i, w_i)_{i \in I}$ we can combine them and get a resulting model which is called the *ultraproduct*.² When every model in the family is the same we call the resulting model an *ultrapower*.

This new model satisfies some nice properties that will be useful for us. We take one of them from Appendix B to illustrate the idea.

Theorem. Let \mathcal{M}^f, w be the ultraproduct of $(\mathcal{M}_i, w_i)_{i \in I}$ and let Γ be a set of first order formulas.

- If every $\mathcal{M}_i, w_i \models \Gamma$ then $\mathcal{M}, w \models \Gamma$.
- In the particular case of an ultrapower this implies that $\mathcal{M}_i, w_i \models \varphi$ if and only if $\mathcal{M}, w \models \varphi$.

We are ready to define the closure under ultraproducts and ultrapowers. These definitions should *only* be used for the special case of BML. In the next sections we will need to redefine these notions to have a broader reach.

Definition 2.2.4. A class K of pointed BML models is *closed under ultraproducts* if and only if, for every family of BML models $(\mathcal{M}_i)_{i \in I}$ with $\mathcal{M}_i \in K$ the ultraproduct of those models also belongs to K . The closure under ultrapowers is defined as expected.

With respect to the closure under bisimulations, it is a lot easier to imagine what it means. We define the notion of closure under bisimulations for the special case of BML bisimulations.

Definition 2.2.5. Let K be a class of BML models, it is *closed under bisimulations* if and only if the following holds: For every $\langle \mathcal{M}, w \rangle \in K$, if $\langle \mathcal{N}, v \rangle$ is such that $\mathcal{M}, w \leftrightarrow \mathcal{N}, v$ then $\langle \mathcal{N}, v \rangle \in K$.

The proof of the Definability Theorem

In this section we give a sketch of the proofs for the *right to left* directions of the theorem. A detailed version can be found in [BdRV01, Section 2.6].

Right to left of (i). Suppose that K is closed under ultraproducts and bisimulations and \bar{K} is closed under ultrapowers. The main ideas to prove this theorem are the following:

1. Propose a set $\Gamma =$ ‘theory of K ’ as a candidate set of formulas defining K . Every model of K trivially makes Γ true. For Γ to define K we still need to prove the other half, that is: If $\mathcal{M}, w \models \Gamma$ then $\langle \mathcal{M}, w \rangle \in K$.
2. Take any $\mathcal{M}, w \models \Gamma$, we will get to a contradiction by assuming that $\langle \mathcal{M}, w \rangle \in \bar{K}$. We start by showing that there is a model $\langle \mathcal{N}, v \rangle \in K$ such that $\mathcal{M}, w \equiv \mathcal{N}, v$. Here we will use that K is closed under ultraproducts.

² Strictly speaking, there is also another ingredient which is called ‘ultrafilter’. Consult Appendix B for further information on ultrafilters and ultraproducts. We recommend its lecture.

3. As we did in the proof of the characterization theorem, we construct, for \mathcal{M}, w and \mathcal{N}, v , ω -saturated extensions \mathcal{M}^*, w^* and \mathcal{N}^*, v^* which are elementary equivalent to their originators. That is, they have the same first-order theory and they are ω -saturated. Observe, again, that this implies $\mathcal{M}^*, w^* \equiv \mathcal{N}^*, v^*$ and $\mathcal{N}^* \models \alpha(x)[v^*]$.
One *important* difference with respect to the proof of the characterization proof is that here we use that \mathbf{K} and $\overline{\mathbf{K}}$ are closed under ultrapowers and conclude that $\langle \mathcal{M}^*, w^* \rangle \in \mathbf{K}$ and $\langle \mathcal{N}^*, v^* \rangle \in \overline{\mathbf{K}}$.
4. Using Theorem 1.4.5 seen in Chapter 1, as $\mathcal{M}^*, w^* \equiv \mathcal{N}^*, v^*$ (and they are saturated) we have that $\mathcal{M}^*, w^* \leftrightarrow \mathcal{N}^*, v^*$.
5. Finally, as $\mathcal{M}^*, w^* \leftrightarrow \mathcal{N}^*, v^*$ and \mathbf{K} is closed under bisimulations then $\langle \mathcal{N}^*, v^* \rangle \in \mathbf{K}$. Absurd, in point 3 we had said that $\langle \mathcal{N}^*, v^* \rangle \in \overline{\mathbf{K}}$.

As seen before, one of the central tricks is the detour through ω -saturated models. The details are as follows: In the first point the set should be defined as

$$\Gamma = \{\varphi : \text{for every model } \langle \mathcal{A}, u \rangle \text{ in } \mathbf{K}; \mathcal{A}, u \models \varphi\}.$$

For the second point, let $\Sigma = \{\varphi : \mathcal{M}, w \models \varphi\}$ be the theory of $\langle \mathcal{M}, w \rangle$. If we find a model in \mathbf{K} that models Σ then it will be BML equivalent to $\langle \mathcal{M}, w \rangle$.

The proof in [BdRV01] hand-crafts an ultraproduct of models to make this step but we will take a route which keeps us away from the inner works of ultraproducts. Suppose that there is no such set in \mathbf{K} making true all of Σ . By Theorem A.1 there exists a finite subset $\Sigma_0 \subseteq \Sigma$ that is not satisfiable in \mathbf{K} . Then $\neg \bigwedge \Sigma_0$ would be true in \mathbf{K} . In particular, $\mathcal{M}, w \not\models \bigwedge \Sigma_0$. This is absurd because Σ_0 is a subset of w 's theory.

Therefore, there exists a model $\langle \mathcal{N}, v \rangle \in \mathbf{K}$ such that $\mathcal{N}, v \models \Sigma$ which implies that $\mathcal{M}, w \equiv \mathcal{N}, v$.

The third and fourth points are justified as in the characterization theorem and the fifth point is self-explanatory. \square

Right to left of (ii). Suppose that both \mathbf{K} and $\overline{\mathbf{K}}$ are closed under bisimulations and ultraproducts. Using the first part of this theorem we know that there exist two sets Γ_1, Γ_2 respectively defining \mathbf{K} and $\overline{\mathbf{K}}$. It is clear that their union should be unsatisfiable because no model can be in \mathbf{K} and $\overline{\mathbf{K}}$ at the same time.

Using the compactness theorem, as $\Gamma_1 \cup \Gamma_2$ is unsatisfiable there must be a finite subset $\Gamma_0 \subseteq \Gamma_1 \cup \Gamma_2$ which is unsatisfiable. Let $\Gamma_0 = \{\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m\}$ where $\alpha_i \in \Gamma_1$ and $\beta_j \in \Gamma_2$. As Γ_0 is unsatisfiable we can say that $\models \alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \neg(\beta_1 \wedge \dots \wedge \beta_m)$. We show that it is exactly $\varphi = \alpha_1 \wedge \dots \wedge \alpha_n$ that defines \mathbf{K} .

Trivially every model in \mathbf{K} satisfies φ . For the converse, take $\mathcal{M}, w \models \alpha_1 \wedge \dots \wedge \alpha_n$, then $\mathcal{M}, w \not\models \beta_1 \wedge \dots \wedge \beta_m$ therefore $\langle \mathcal{M}, w \rangle \notin \overline{\mathbf{K}}$ which means that $\langle \mathcal{M}, w \rangle \in \mathbf{K}$. \square

To close this section we want to draw attention to one of the hypothesis in (ii): the need for *both* classes to be closed under bisimulations. Observe that, as the bisimulation relation is symmetric we could've just asked for either \mathbf{K} or $\overline{\mathbf{K}}$ to be closed under bisimulations and that would've been enough. One can prove that \mathbf{K} is closed under bisimulations if and only if $\overline{\mathbf{K}}$ is.

On the other hand, in the proof of (ii) from right to left, we *strongly* use that both classes are closed under bisimulations to get two sets that define each of the classes. What

would happen now if we were talking about simulations? As simulations are not necessarily symmetrical we can't be sure that K is closed under simulations if and only if $\bar{\mathsf{K}}$ is. This fault brings problems if we want to follow this same proof scheme.

In [KdR97, KR97] there are alternative proofs for this result for model equivalence notions that aren't symmetrical. None of them are general enough to fit the framework that we will develop but both have proved of great inspiration for the results given in Section 4.3.

3. THE GENERALIZED FRAMEWORK

In this chapter we set up a proper framework which will aid us to prove generalized results for modal logics which lay (in terms of expressivity) below first order logic. We start by stating in which sense our results pursue a generalization. We will focus on the following two axes.

Arbitrary modal logic

We want to obtain characterization and definability results which hold for an arbitrary modal logic. Due to the broad spectrum of different logics we still have to stop somewhere. When we say ‘arbitrary’ we mean any modal logic with conjunction and disjunction (interpreted as usual) which is interpreted over extensions of Kripke models.

These logics may come with different model equivalent notions. We want to be able to derive results no matter what the simulation or bisimulation relation looks like. We will only put constraints on the ‘arity’ of the relation, that is, it should link an element from the domain of one model to an element of the domain of other model. It will later become clear that this last generalization comes with a great price to pay: we know nothing about the structural properties involved in this notion.

Relativization to a particular class of models

The results presented in Chapter 2 were stated with respect to the class of *all models*. That is, BML is the fragment of first order formulas which are bisimulation invariant in the class of *all first models*. Think of the following motivational example.

The ‘Basic Temporal Logic’ is a modal logic which is defined as follows: Its language has the full boolean connective set and two modalities F and P which are often called ‘future’ and ‘past’. The classical perspective on this logic interprets it over Kripke models defined as a tuple $\langle W, R, V \rangle$ and its satisfaction definition is the following.

$$\begin{aligned} \mathcal{M}, w \models F\varphi & \text{ iff } \text{there is a } v \text{ such that } wRv \text{ and } \mathcal{M}, v \models \varphi \\ \mathcal{M}, w \models P\varphi & \text{ iff } \text{there is a } v \text{ such that } vRw \text{ and } \mathcal{M}, v \models \varphi \end{aligned}$$

In the definition it is clear that the F modality looks forward in the relation R and the P modality looks back on it, hence the names ‘future’ and ‘past’. Observe that the F modality can be thought as a normal ‘diamond’ over the relation R but that is not possible with the P modality.

An alternative interpretation is as follows. Interpret the logic over Kripke models which are tuples $\langle W, R_1, R_2, V \rangle$ where $R_1 = R_2^{-1}$. With this restriction we can give a different satisfaction definition for the modalities.

$$\begin{aligned} \mathcal{M}, w \models F\varphi & \text{ iff } \text{there is a } v \text{ such that } wR_1v \text{ and } \mathcal{M}, v \models \varphi \\ \mathcal{M}, w \models P\varphi & \text{ iff } \text{there is a } v \text{ such that } wR_2v \text{ and } \mathcal{M}, v \models \varphi \end{aligned}$$

In this case, both modalities are simple ‘diamonds’ (which have been given fancy names F and P). Does a similar characterization theorem hold in this case? Which properties should the restricted model class have for the characterization to hold? These are the kind of questions that we will be addressing in the following chapters.

When talking about definability we can think of *relative definability* as follows: Is the class K definable with a BML formula *given* that we only consider models within class K' ? That is, is there a formula φ such that for every model in K' , $\mathcal{M}, w \models \varphi$ if and only if $\langle \mathcal{M}, w \rangle \in \mathsf{K}$?

The results stated in Chapter 2 are valid for the special case where K' is the class of all models. In practice, depending on the domain of application, it is common to work with restricted classes of models such as finite models, tree models, acyclic models, etc. We want to know whether these restrictions give us extra information and turn classes that were previously undefinable into definable classes. A relativized version of the definability theorem should aid us in this quest.

3.1 Basic definitions

Definition 3.1.1 (Languages and formulas). We note \mathfrak{L} and \mathfrak{F} as the source and target languages respectively. The source language is an extension of the language

$$\mathfrak{P} = \langle (p_i)_{i \in \mathbb{N}}, \wedge, \vee, \top, \perp \rangle$$

which has infinitely many propositional variables, conjunction and the *true* and *false* constants. The target language is a (countable) first-order language which may or may not contain equality.

$\text{FORM}(\mathfrak{L})$ is the set of formulas of the language \mathfrak{L} and $\text{FORM}(\mathfrak{F}^1)$ is the subset of $\text{FORM}(\mathfrak{F})$ with at most one free variable (and that variable is x).

During this thesis we will deal with source logics which are at most as expressive as first order logic. If \mathfrak{L} is less or equally expressive than \mathfrak{F} we should be able to express in \mathfrak{F} everything that is expressible in \mathfrak{L} . We have seen before that, for BML, there exists a standard translation ST_x from BML to first order logic. In general we define a formula translation as follows.

Definition 3.1.2 (Formula translation). A *formula translation* is a function

$$\text{Tf}_x : \text{FORM}(\mathfrak{L}) \rightarrow \text{FORM}(\mathfrak{F}^1)$$

that translates formulas from the source language \mathfrak{L} to the first-order language \mathfrak{F} . This function is required to preserve conjunctions and disjunctions, that is, formally speaking: Let $\varphi_1, \varphi_2 \in \text{FORM}(\mathfrak{L})$ and $\odot \in \{\wedge, \vee\}$ then for every first-order formula of the form $\text{Tf}_x(\varphi_1) \odot \text{Tf}_x(\varphi_2)$ there exists an \mathfrak{L} -formula ψ such that $\text{Tf}_x(\psi) \equiv_{\mathfrak{F}} \text{Tf}_x(\varphi_1) \odot \text{Tf}_x(\varphi_2)$.

As we saw before in the definition of BML's standard translation, in general, formula translations are defined homomorphically with respect to the boolean connectives.

$$\begin{aligned} \text{Tf}_x(\varphi_1 \wedge \varphi_2) &= \text{Tf}_x(\varphi_1) \wedge \text{Tf}_x(\varphi_2) \\ \text{Tf}_x(\varphi_1 \vee \varphi_2) &= \text{Tf}_x(\varphi_1) \vee \text{Tf}_x(\varphi_2) \end{aligned}$$

Definition 3.1.3 (Models). We will be working with source logics that are interpreted over variations of Kripke models.¹ We define $\text{MODS}(\mathfrak{L})$ to be the class of all models of the source logic and $\text{MODS}(\mathfrak{F})$ to be the class of all models of the target first-order logic.

¹ Even propositional logic can be thought of as a modal logic without modal operators and restricted to models with one single point.

Sometimes we will use the notion of *pointed models*. A *pointed model* in the source logic \mathcal{L} is a model-world pair. We can see a pointed model as an \mathcal{L} -model where the evaluation point has been fixed. We define the class of pointed models for the source logic as

$$\text{PMODS}(\mathcal{L}) = \{\langle \mathcal{M}, w \rangle : \mathcal{M} \in \text{MODS}(\mathcal{L}) \text{ and } w \in |\mathcal{M}|\}$$

Similarly, in the target logic \mathfrak{F} , we use $\mathcal{M}^f, g \models \varphi$ to note that a formula φ is true in the model \mathcal{M}^f under the valuation (or assignment) g .² A *pointed model* of the target logic \mathfrak{F} is a model-assignment pair. We can see a pointed model as an \mathfrak{F} -model where the assignment function has been fixed.

Definition 3.1.4 (*x*-assignment). Let \mathcal{M}^f be an \mathfrak{F} model. An *x*-assignment for \mathcal{M}^f is a function

$$g : \{x\} \rightarrow |\mathcal{M}^f|$$

which assigns an element for the variable x . It can be seen as a finite valuation specialized to the variable x .

We will use the concept of *x*-assignment to define the class of first order pointed models. This notion is a technical detail needed to make things work in Definition 3.1.6. The problem and solution will become clear after that definition. We define the class of pointed models for the target logic as

$$\text{PMODS}(\mathfrak{F}) = \{\langle \mathcal{M}^f, g \rangle : \mathcal{M}^f \in \text{MODS}(\mathfrak{F}) \text{ and } g \text{ is an } x\text{-assignment for } \mathcal{M}^f\}$$

Observe that the formulas obtained through the translation defined in Definition 3.1.2 have *at most* one free variable and that variable is x . Therefore, if we want to evaluate those formulas in a first order model \mathcal{M}^f , an *x*-assignment g is enough for $\mathcal{M}^f, g \models \varphi(x)$ to be well defined.

Notation 3.1.5. Let $\langle \mathcal{M}^f, g \rangle, \langle \mathcal{N}^f, h \rangle \in \text{PMODS}(\mathfrak{F})$. We write $\mathcal{M}^f, g \equiv_{\mathfrak{F}} \mathcal{N}^f, h$ to mean that for every first order formula $\alpha(x)$; $\mathcal{M}^f, g \models \alpha(x)$ if and only if $\mathcal{N}^f, h \models \alpha(x)$.

There's one more thing to be taken into account, formulas from \mathcal{L} and formulas from \mathfrak{F} are not evaluated in the same models. The former are evaluated in Kripke models and the later are evaluated in first-order models. This is the reason why we are not yet able to compare φ with $\text{Tf}_x(\varphi)$.

We can think of models as ‘information bearers’, they represent some information relative to the world in a way that is compatible with some logic. Therefore, the information is not in the model itself but somewhere else. We need to define some way to ‘look at’ this information from different perspectives, one compatible with the source logic \mathcal{L} and other compatible with the target logic \mathfrak{F} . Following the same line we define a model translation that ‘converts’ the information between the source and target logic.

Definition 3.1.6 (Model translation). Given a class of models $\mathsf{K} \subseteq \text{PMODS}(\mathfrak{F})$, a *model translation* is a bijective function

$$\text{T}_{\mathsf{K}} : \text{PMODS}(\mathcal{L}) \rightarrow \mathsf{K}$$

² Observe that in this case g is a valuation and not a point of the domain.

We write \mathbb{T} instead of \mathbb{T}_K when the class of models is clear from the context. As an abuse of notation we use $\mathbb{T}(\mathcal{M})$ when we are not interested in the associated assignment and $\mathbb{T}^{-1}(\mathcal{A}^f, g)$ for the preimage of $\langle \mathcal{A}^f, g \rangle$.

Returning to the need for x -assignments, note that if we allowed g to be a standard assignment (i.e. $g : \text{FVAR} \rightarrow |\mathcal{M}^f|$) in Definition 3.1.3 then for every pointed \mathcal{L} -model $\langle \mathcal{M}, w \rangle$ we would have *many* pointed \mathfrak{F} -model $\langle \mathcal{M}^f, g_i \rangle$ where $g_i(x) = w$ and the assignment for the rest of the variables changes arbitrarily. Therefore, this could carry problems at the moment of satisfying the *surjectivity* requirement.

As an exercise, suppose that the class of pointed models is defined with standard assignments and try to define a model translation for BML. You will observe that there is a cardinality problem.

When we proved the results for BML we did not use a model translation, at least not explicitly. On the other hand, this translation was implicitly present when we gave an informal way to ‘look at’ models from both a BML and a first order perspective. The model translation function will serve us in this task. We are now ready to set proper constraints on the translations.

Definition 3.1.7 (Truth preserving pair of translations). A pair of translations $(\text{Tf}_x, \mathbb{T})$ is said to be *truth-preserving* if for all $\varphi \in \text{FORM}(\mathcal{L})$ and all $\langle \mathcal{M}, w \rangle \in \text{PMODS}(\mathcal{L})$

$$\mathcal{M}, w \models \varphi \text{ iff } \mathbb{T}(\mathcal{M}, w) \models \text{Tf}_x(\varphi)$$

Let’s fix $(\text{Tf}_x, \mathbb{T}_K)$ as our pair of *truth-preserving* translations for the rest of the thesis. We will also want to translate formulas from \mathcal{L} to \mathfrak{F} and then go back to \mathcal{L} -formulas. As we are not requiring Tf_x to be injective this could lead to a problem. We make the following claim.

Proposition 3.1.8. For any α, β such that $\text{Tf}_x(\alpha) = \text{Tf}_x(\beta)$ we have $\models_{\mathcal{L}} \alpha \leftrightarrow \beta$.

Proof. Suppose that $\not\models_{\mathcal{L}} \alpha \leftrightarrow \beta$, then we have a model \mathcal{M} and a point w such that $\mathcal{M}, w \models \alpha$ and $\mathcal{M}, w \not\models \beta$. Then by definition of truth-preservation of the translations we get $\mathbb{T}(\mathcal{M}, w) \models \text{Tf}_x(\alpha)$ and $\mathbb{T}(\mathcal{M}, w) \not\models \text{Tf}_x(\beta)$. Absurd. \square

We will use this proposition to make a simplification. First define the equivalence relation $\varphi \sim \psi$ iff $\models_{\mathcal{L}} \varphi \leftrightarrow \psi$. Regarding \mathcal{L} -formulas, we can always take the equivalence classes defined by the quotient set of \mathcal{L} -formulas by \sim and for each class choose a representative to work with.

To simplify the proofs in this thesis, and without loss of generality, we will assume that we are working with the set of formulas defined above. All of our proofs should also work with the original set of formulas but they would require excessive detours and justifications. In this setting we will be working up to formula equivalence and we will assume that our formula translation Tf_x is injective.

Definition 3.1.9. Let $K \subseteq \text{MODS}(\mathcal{L})$, \mathcal{M} be an \mathcal{L} -model and $w \in |\mathcal{M}|$. We define the *theory* of a pointed model, model and class of models as follows

$$\begin{aligned} \text{Th}(\mathcal{M}, w) &= \{\varphi : \mathcal{M}, w \models \varphi\} \\ \text{Th}(\mathcal{M}) &= \{\varphi : \mathcal{M} \models \varphi\} \\ \text{Th}(K) &= \{\varphi : \forall \mathcal{M} \in K \text{ it holds that } \mathcal{M} \models \varphi\} \end{aligned}$$

Given two \mathcal{L} -pointed models $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ we say that the *pointed* models are modally equivalent (noted $\mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$) when $\text{Th}(\mathcal{M}, w) = \text{Th}(\mathcal{N}, v)$. We say that two models (*not* pointed) are modally equivalent (noted $\mathcal{M} \equiv_{\mathcal{L}} \mathcal{N}$) when $\text{Th}(\mathcal{M}) = \text{Th}(\mathcal{N})$. We write $\mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$ when $\text{Th}(\mathcal{M}, w) \subseteq \text{Th}(\mathcal{N}, v)$ and $\mathcal{M} \sqsubseteq_{\mathcal{L}} \mathcal{N}$ when $\text{Th}(\mathcal{M}) \subseteq \text{Th}(\mathcal{N})$. All these definitions can be similarly defined for the target logic \mathfrak{F} and we will assume them defined.

The framework defined in this section will allow us to transfer results between the source and target logics. As an example we prove compactness for \mathcal{L} under some special closure conditions (which will be addressed later).

Lemma 3.1.10 (\mathcal{L} is compact). If \mathcal{L} has a pair of truth-preserving translations $(\text{Tf}_x, \text{T}_K)$ and K is closed under ultraproducts then \mathcal{L} is compact.

Proof. Let Γ be a set of \mathcal{L} -formulas and suppose that any finite set of Γ is \mathcal{L} -satisfiable. We will show that Γ is \mathcal{L} -satisfiable.

Take any finite $\Delta_f \subseteq \text{Tf}_x(\Gamma)$, we want to see that it is satisfiable in K . As our formula translation is injective we have a set $\Delta \subseteq \Gamma$ such that $\Delta = \text{Tf}_x^{-1}(\Delta_f)$. Observe that Δ is finite because Tf_x is injective. By hypothesis there exists $\langle \mathcal{M}, w \rangle$ such that $\mathcal{M}, w \models \Delta$ because Δ is a finite subset of Γ . Hence, by truth-preservation, $\text{T}(\mathcal{M}, w) \models \Delta_f$ and $\text{T}(\mathcal{M}, w) \in K$.

By Theorem A.1 we conclude that there exists a model $\langle \mathcal{N}^f, g \rangle$ in the class K such that $\mathcal{N}^f, g \models \text{Tf}_x(\Gamma)$. As the translations are truth preserving we get $\text{T}^{-1}(\mathcal{N}^f, g) \models \Gamma$. \square

3.2 General model equivalence

In Chapter 1 of this thesis we introduced the general idea of model equivalence and, in particular the notions of simulation and bisimulation for some specific logics (namely BML $\bar{\Gamma}$ and BML). We want the framework we are developing to be able to handle several types of model equivalence relations.

Restricting ourselves to the definition of simulation and bisimulation we can see that, the latter can be seen as a special case of the first where there is a symmetrical atomic condition and a ‘back’ clause. Looking at their common properties we can say that they both agree in the following points:

- (i) They relate a point in one model with a point in the other model. Thus, given \mathcal{M} and \mathcal{N} , if Z is such relation then $Z \subseteq |\mathcal{M}| \times |\mathcal{N}|$.
- (ii) They imply some kind of modal theory transfer. In the case of simulations, if wZv then $\mathcal{M}, w \sqsubseteq \mathcal{N}, v$. On the other hand, bisimulations imply full modal equivalence: if wZv then $\mathcal{M}, w \equiv \mathcal{N}, v$.
- (iii) For every (bi)simulation between \mathcal{M} and \mathcal{N} , the models’ structure doesn’t change. That is why the notion of (bi)simulation always links points from \mathcal{M} to points of \mathcal{N} . We will have to make a small change here to be able to handle (bi)simulations in dynamic logics.³

Following this analysis, given a source logic \mathcal{L} , we will give a *meta-definition* for model equivalence notions. That is, we will not define a relation but give the conditions that

³ Logics where the modal operators may change the model.

a valid model equivalence notion should satisfy. Any simulation or bisimulation relation satisfying the next definition fits into our framework.

As an abuse of language, we call it \mathcal{L} -simulation. We use this name because it reminds us of the properties that simulation relation defined for BML^- satisfies but it is *not* the same. As it will be clear from the definition, we don't impose any structural constraint.

Definition 3.2.1 (\mathcal{L} -simulation). Given two \mathcal{L} models \mathcal{M} and \mathcal{N} we define an \mathcal{L} -simulation to be a *non-empty* relation $Z \subseteq \text{PMODS}(\mathcal{L}) \times \text{PMODS}(\mathcal{L})$ with the following constraint

$$\text{If } \langle \mathcal{M}, w \rangle Z \langle \mathcal{N}, v \rangle \text{ then } \mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$$

We write $\mathcal{M}, w \rightrightarrows_{\mathcal{L}} \mathcal{N}, v$ to indicate that there exists a simulation between w and v and $\mathcal{M} \rightrightarrows_{\mathcal{L}} \mathcal{N}$ to indicate that there exists a point $w \in |\mathcal{M}|$ such that $\mathcal{M}, w \rightrightarrows_{\mathcal{L}} \mathcal{N}, v$ for some $v \in |\mathcal{N}|$. We write $w \rightrightarrows_{\mathcal{L}} v$ when the models are clear from context.

Note that the simulation definition for BML^- satisfies the above definition with minor changes. The only difference is that we have to take into account the 'model' component of the simulation relation. It can be re-defined as follows.

Example 3.2.2. A BML^- simulation is a non-empty binary relation between pointed models such that whenever $\langle \mathcal{M}, w \rangle Z \langle \mathcal{N}, v \rangle$ we have that:

Atomic condition: If $w \in V^{\mathcal{M}}(p)$ then $v \in V^{\mathcal{N}}(p)$ for all $p \in \text{PROP}$.

Forth: if wR_1w' , then there exists a point v' in \mathcal{N} such that vR_2v' and $\langle \mathcal{M}, w' \rangle Z \langle \mathcal{N}, v' \rangle$.

When thinking about BML 's bisimulation, it may seem that this definition is missing something. We know that if \mathcal{M}, w and \mathcal{N}, v are related by a *bisimulation* then $\mathcal{M}, w \equiv \mathcal{N}, v$ but the above definition only guarantees $\mathcal{M}, w \sqsubseteq \mathcal{N}, v$. Don't worry about that now, it will become clear in the next section that this condition is enough for what we need. Observe, also, that a bisimulation is a special case of Definition 3.2.1 where the relation is symmetric.

We want to stress that this definition of \mathcal{L} -simulation does not cover all possible types of model equivalences and it isn't suitable for all types of modal languages. One example where this notion is not adequate is when the language doesn't have *disjunction* nor *negation*. Let \mathcal{M} and \mathcal{N} be two models, the right notion of simulation for this language links *sets* of points from \mathcal{M} to a point of \mathcal{N} . As we have defined our possible languages in Section 3.1 this will not be a problem because we always have disjunction in our source language. For more information about model theory on disjunction-free languages refer to [KdR97].

3.3 Saturation

We have seen that, in general, modal equivalence does not imply bisimilarity. It is also the case with BML 's simulation that, in general $\mathcal{M}, w \rightrightarrows \mathcal{N}, v$ does not imply $\mathcal{M}, w \sqsubseteq \mathcal{N}, v$. This problem recurs with most model equivalence notions found in the literature.

We have stated that in the class of ω -saturated models, BML equivalence implies bisimulation. For this framework we need to define a similar notion which fits the logics we will be working with. Let's formally define the general condition that we're pursuing so we can focus on it.

Definition 3.3.1 (Hennessy-Milner Property). Let \mathbf{K} be a class of \mathcal{L} -pointed models, we say that \mathbf{K} has the *Hennessy-Milner property* if for every two \mathcal{L} -models $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ in \mathbf{K} , whenever $\mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$ we have $\mathcal{M}, w \rightleftharpoons_{\mathcal{L}} \mathcal{N}, v$.

This definition should be interpreted as the converse of the \mathcal{L} -simulation (Definition 3.2.1) requirement and will be the definition of Hennessy-Milner class used in our framework.

Is this definition general enough to cover the cases we have been talking about? We know that if we fix \mathcal{L} as BML and the simulation relation as BML's bisimulation we have that $\mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$ implies $\mathcal{M}, w \rightleftharpoons_{\mathcal{L}} \mathcal{N}, v$ but the definition above seems to impose a stronger constraint. We only have $\mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$ as hypothesis and we should conclude the same thesis. We make the following statement that explains why there is no problem with this.

Proposition 3.3.2. If \mathcal{L} has negation then $\mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$ if and only if $\mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$.

As BML has negation, the seemingly weak hypothesis turns strong enough to prove the result in that particular case. The special case regarding saturation for BML is nicely covered in [BdRV01].

Regarding ω -saturation, the definition given in Definition 1.4.4 is in terms of first order models. In that moment, as we were looking at BML and first order models as if they were the same, that gave us no problems.

In this chapter we want to make an explicit differentiation between \mathcal{L} -models and \mathfrak{F} -models. To make our proofs simpler we choose the following definition for ω -saturated \mathcal{L} -models.

Definition 3.3.3. We say that an \mathcal{L} -model \mathcal{M} is *ω -saturated* if and only if $\mathsf{T}(\mathcal{M})$ is.

For details on ω -saturation, classical results can be found in [CK90]. Also, in Marco Hollenberg's thesis [Hol98], he extensively investigates Hennessy-Milner classes.

4. MAIN RESULTS

4.1 Adequate pair

In Chapter 3, while developing the framework, we explained the objectives we are pursuing when doing a generalization. In the following definition we will make explicit the requirements for the theorems in this chapter to hold for an arbitrary logic \mathcal{L} and with respect to a class of models \mathbb{K} .

Definition 4.1.1 (Adequate pair). A logic \mathcal{L} and a class of models $\mathbb{K} \subseteq \text{PMODS}(\mathfrak{F})$ is said to be an *adequate pair* if they fulfill the following requirements

1. \mathbb{K} is closed under ultraproducts (Definition 4.1.2).
2. There exist truth-preserving translations $\text{Tf}_x, \text{T}_\mathbb{K}$ (Definition 3.1.7).
3. There exists an \mathcal{L} -simulation notion (Definition 3.2.1).
4. The class of ω -saturated \mathcal{L} -models should have the Hennessy-Milner property with respect to \mathcal{L} -simulations (Definitions 1.4.4 and 3.3.1).

We need to formally define the closure under ultraproducts and ultrapowers used above, as the ones given in Chapter 2 were specifically crafted for BML.

Definition 4.1.2 (Closure under ultraproducts). A class $\mathbb{K} \subseteq \text{MODS}(\mathfrak{F})$ is said to be *closed under ultraproducts* if, let \mathcal{M}_i^f be a family of \mathfrak{F} -models in \mathbb{K} and let U be an ultrafilter, the ultraproduct $\prod_U \mathcal{M}_i^f$ is also in \mathbb{K} . A more sophisticated definition is needed for first-order pointed models.

A class $\mathbb{K} \subseteq \text{PMODS}(\mathfrak{F})$ is said to be *closed under ultraproducts* if, let $\langle \mathcal{M}_i^f, g_i \rangle$ be a family of \mathfrak{F} -pointed models and let U be an ultrafilter. Let the $\prod_U \mathcal{M}_i^f$ be the ultraproduct of the models then $\langle \prod_U \mathcal{M}_i^f, g_i^* \rangle \in \mathbb{K}$ for every g_i^* defined as $g_i^*(x) = \lambda z. g_i(x)$ for all x .¹

Definition 4.1.3 (Closure under ultrapowers). A class \mathbb{K} of \mathfrak{F} -models is said to be *closed under ultrapowers* if it is closed under ultraproducts where every \mathcal{M}_i^f is the same model. A similar definition can be given for pointed models.

Why are we requiring \mathbb{K} to be closed under ultraproducts? We could have asked for more, such as \mathbb{K} being definable by a first order formula, which implies closure under ultraproducts. We could've also tried to impose no restriction over \mathbb{K} .

We decided to require \mathbb{K} to be closed under ultraproducts because it is the weakest condition that lets us use the relativized version of the first order compactness theorem (stated and proved in the appendix as Theorem A.1). In particular, all first-order definable classes and the class of 'all models' will always fit in an adequate pair.

The second item in Definition 4.1.1 makes sure that \mathcal{L} is less or equally expressive than first order and that there is some way to compare between the formulas and models of both logics.

In the same way, the third item only asks for the definition of a simulation notion which is essential to develop the model theory of \mathcal{L} . All the results will be stated in terms of that \mathcal{L} -simulation notion.

¹ For a formal definition of the lambda notation refer to [Bar85].

With enough practice, points one to three can be easily checked by just ‘looking at’ \mathcal{L} . It is only when we get to the last item that we face the strongest requirement. This points says that the class of ω -saturated \mathcal{L} -models should have the Hennessy-Milner property.

In Chapter 2 we mentioned that the class of ω -saturated BML models had the Hennessy-Milner property with respect to BML bisimulations (although in that moment we didn’t call it ‘Hennessy-Milner’ yet). The proof of that result makes a link between the semantics of BML and the structure of the BML bisimulation. Therefore, it makes use of the *structural* definition of the BML bisimulation.

So far, given a logic \mathcal{L} , we are looking at \mathcal{L} -simulations as black boxes. All we know is that $w \Rightarrow_{\mathcal{L}} v$ implies $w \sqsubseteq_{\mathcal{L}} v$. We don’t know which structural properties it imposes. This is the reason why we still need this item to be proved for the results to work.

We think that there’s still much work to be done to weaken this last requirement and we will give our opinion on directions for further work in the conclusions.

4.2 Characterization

One of the central notions in the characterization theorem for BML was that of bisimulation invariance. Recall that bisimulations are defined between BML models but the notion of bisimulation invariance is defined for first order formulas.

Definition. A first order formula $\alpha(x)$ is *invariant for BML bimulations* if for all BML models \mathcal{M}, \mathcal{N} and $w \in |\mathcal{M}|, v \in |\mathcal{N}|$ such that $\mathcal{M}, w \Leftrightarrow \mathcal{N}, v$ the following holds:

$$\mathcal{M} \models \alpha(x)[w] \text{ iff } \mathcal{N} \models \alpha(x)[v].$$

When working with BML, this difference made no problem to us because we didn’t really distinguish between BML and first order models. It is time for us to give an invariance definition that fits our framework and there is an important decision to be made.

The property of ‘invariance’ is thought for first order formulas and the notion of \mathcal{L} -simulation is defined between \mathcal{L} models. We have two options: The first one is to call a first order formula $\alpha(x)$ ‘invariant for \mathcal{L} -simulations’ if, for every two \mathcal{L} models \mathcal{M}, w and \mathcal{N}, v such that $\mathcal{M}, w \Rightarrow_{\mathcal{L}} \mathcal{N}, v$ whenever $\alpha(x)$ holds in $\mathsf{T}(\mathcal{M}, w)$ it should hold in $\mathsf{T}(\mathcal{N}, v)$. In this case we are ‘mixing’ the models through the translation.

The other option is to ‘lift’ the \mathcal{L} -simulation notion to \mathfrak{F} models and define a simulation relation $\Rightarrow_{\mathfrak{F}}$. In this case we could just say that a first order formula $\alpha(x)$ is ‘invariant for \mathcal{L} -simulations’ if, for every two \mathfrak{F} models \mathcal{M}^f, g and \mathcal{N}^f, h such that $\mathcal{M}^f, g \Rightarrow_{\mathfrak{F}} \mathcal{N}^f, h$ whenever $\alpha(x)$ holds in \mathcal{M}^f, g it should hold in \mathcal{N}^f, h .

The advantage of the first option is that there’s no need for new definitions; in contrast, the second one would require a formalization for the ‘lifting’ to be defined along with the model translation. On the other hand, it would be nice to be able to check two \mathfrak{F} models for ‘model equivalence’ as we can do with \mathcal{L} .

In this thesis we choose the first option because it is the most direct one in this setting. Observe that, in particular, the first option can be seen as a special case of the second one when the following ‘canonical lifting’ is defined.

$$\mathcal{M}^f, g \Rightarrow_{\mathfrak{F}} \mathcal{N}^f, h \text{ iff } \mathsf{T}^{-1}(\mathcal{M}^f, g) \Rightarrow_{\mathcal{L}} \mathsf{T}^{-1}(\mathcal{N}^f, h)$$

Again, the problem with this definition is that it bears no structural information regarding model equivalence between first order models. It is just another detour.

Definition 4.2.1 (\mathcal{L} -simulation K -invariance). Let $\langle \mathcal{L}, K \rangle$ be an adequate pair. A formula $\alpha(x)$ of \mathfrak{F}^1 is K -invariant for \mathcal{L} -simulations if for all \mathcal{L} -models \mathcal{M}, \mathcal{N} and $w \in |\mathcal{M}|, v \in |\mathcal{N}|$: If $\mathcal{M}, w \rightleftharpoons_{\mathcal{L}} \mathcal{N}, v$ and $\mathbb{T}(\mathcal{M}, w) \models \alpha(x)$ then $\mathbb{T}(\mathcal{N}, v) \models \alpha(x)$.

Before stating the characterization theorem let's see the importance and role of the class of models K in these definitions. What happens to formula equivalence when we change the class of models? As a motivating example we will work with a first-order formula $\varphi = \forall x.R(x, x)$ which holds in a model if and only if R is a reflexive relation.

In the class of all models it is obvious that $\not\models \varphi$ (φ is not valid) because we can come up with some models where R is not reflexive. Given that “reflexivity” is not expressible in basic modal logic, we can conclude that φ is not equivalent to the translation of any basic modal formula.

Definition 4.2.2 (K -equivalence). Let $K \subseteq \text{MODS}(\mathfrak{F})$ and $\varphi, \psi \in \text{FORM}(\mathfrak{F})$. We say that φ and ψ are K -equivalent if and only if $\models_K \varphi \leftrightarrow \psi$.

Let's now restrict the class of models, let K be the class of *reflexive* models. Now $\models_K \varphi$ because it is valid in every model of the class. In this setting there *is* a basic modal formula whose translation is K -equivalent to φ . Take $\psi = \top$ we have $\models_K ST_x(\psi) \leftrightarrow \varphi$ because $\models_K \top \leftrightarrow \varphi$. What happened here is that, restricting our class of models the number of valid formulas has grown and with them the number of “formulas equivalent to a translation”.

Something similar occurs with \mathcal{L} -simulation invariance. Again, we have seen that in the class of all models we can have two bisimilar models where one has a reflexive relation and the other doesn't. Therefore “reflexivity” is not invariant under bisimulations.

If we change the class of models to the class of reflexive models we see that now the property becomes invariant over bisimulations. This happens because it is trivially invariant *all* over K . It is nice to observe that the concepts of invariance and equivalence are very closely related to each other when we change the class of models we are working with.

Theorem 4.2.3 (Characterization). Given an adequate pair $\langle \mathcal{L}, K \rangle$ then

A formula $\alpha(x)$ of \mathfrak{F}^1
is K -equivalent to the translation of an \mathcal{L} -formula iff
 $\alpha(x)$ is K -invariant for \mathcal{L} -simulations.

Left to right. Suppose $\alpha(x)$ is K -equivalent to the translation of an \mathcal{L} -formula φ . We want to see that it is invariant over \mathcal{L} -simulations. This is a consequence of the invariance of \mathcal{L} -formulas over \mathcal{L} -simulations. Suppose we have $\mathcal{M}, w \rightleftharpoons \mathcal{N}, v$ and $\mathbb{T}(\mathcal{M}, w) \models \alpha(x)$ but $\mathbb{T}(\mathcal{N}, v) \not\models \alpha(x)$. As $\models_K \alpha(x) \leftrightarrow \text{Tf}_x(\varphi)$ and the translations are truth-preserving it must hold that $\mathcal{M}, w \models \varphi$ and $\mathcal{N}, v \not\models \varphi$. But this is a contradiction because \mathcal{L} -formulas are invariant under \mathcal{L} -simulations and we have a simulation linking those points. \square

Right to left. Suppose $\alpha(x)$ is K -invariant for \mathcal{L} -simulations, we want to see that it is K -equivalent to the translation of an \mathcal{L} -formula. Consider the following set of consequences

$$\text{SLC}(\alpha) = \{\text{Tf}_x(\varphi) : \varphi \text{ is an } \mathcal{L}\text{-formula and } \alpha(x) \models_K \text{Tf}_x(\varphi)\}.$$

We will prove that if $\text{SLC}(\alpha) \models_K \alpha(x)$ we are done.

Proposition 4.2.4. If $\text{SLC}(\alpha) \models_{\mathbf{K}} \alpha(x)$ then $\alpha(x)$ is \mathbf{K} -equivalent to the translation of an \mathcal{L} -formula.

Proof. Suppose $\text{SLC}(\alpha) \models_{\mathbf{K}} \alpha(x)$, by relative compactness (Theorem A.1) there is a finite set $\Delta \subseteq \text{SLC}(\alpha)$ such that $\Delta \models_{\mathbf{K}} \alpha(x)$, therefore $\models_{\mathbf{K}} \bigwedge \Delta \rightarrow \alpha(x)$. Trivially (by definition) we have that $\models_{\mathbf{K}} \alpha(x) \rightarrow \bigwedge \Delta$ so we can conclude $\models_{\mathbf{K}} \alpha(x) \leftrightarrow \bigwedge \Delta$. As every $\beta \in \Delta$ is the translation of an \mathcal{L} -formula and the translation preserves conjunction then $\bigwedge \Delta$ is also the translation of some modal formula. \square

Lemma 4.2.5. $\text{SLC}(\alpha) \models_{\mathbf{K}} \alpha(x)$.

Proof. Suppose that $\mathsf{T}(\mathcal{M}, w) \models \text{SLC}(\alpha)$. We have to show that $\mathsf{T}(\mathcal{M}, w) \models \alpha(x)$. Define $\text{NTh}^w(x)$ as

$$\text{NTh}^w(x) = \{\neg \mathsf{Tf}_x(\varphi) : \varphi \text{ is an } \mathcal{L}\text{-formula and } \mathcal{M}, w \not\models \varphi\}$$

Observe that, if \mathcal{L} has negation then $\text{NTh}^w(x)$ will be the translation of w 's modal theory and every model of $\text{NTh}^w(x)$ will be modally equivalent to w . If \mathcal{L} doesn't have negation we will only preserve formulas that are not true in w . This definition fits for both cases. Now define the set

$$\Sigma(x) = \{\alpha(x)\} \cup \text{NTh}^w(x).$$

We will see that $\Sigma(x)$ has a model in \mathbf{K} .

Proposition 4.2.6. $\Sigma(x)$ has model in \mathbf{K} .

Proof. Let's suppose that there is no model in \mathbf{K} for $\Sigma(x)$ and use the contrapositive of Theorem A.1. We can conclude that there must be a finite subset $\{\alpha(x), \neg\delta_1, \dots, \neg\delta_n\} \subseteq \Sigma(x)$ with $\neg\delta_i \in \text{NTh}^w(x)$ which doesn't have model in \mathbf{K} . Note that this set should include $\alpha(x)$, otherwise it would have a had model, namely $\mathsf{T}(\mathcal{M}, w)$.

Observe that, for every model $\mathcal{A}^f \in \mathbf{K}$, as $\mathcal{A}^f \not\models \{\alpha(x), \neg\delta_1, \dots, \neg\delta_n\}$ then $\mathcal{A}^f \models \alpha(x) \rightarrow \neg(\neg\delta_1 \wedge \dots \wedge \neg\delta_n)$. This means that $\alpha(x) \rightarrow (\delta_1 \vee \dots \vee \delta_n)$ is valid in \mathbf{K} , therefore $\alpha(x) \models_{\mathbf{K}} \delta_1 \vee \dots \vee \delta_n$. If $\delta_1 \vee \dots \vee \delta_n$ is a \mathbf{K} -consequence of $\alpha(x)$ then, as the formula translation preserves disjunction, $\delta_1 \vee \dots \vee \delta_n \in \text{SLC}(\alpha)$. But, as $\mathsf{T}(\mathcal{M}, w) \models \text{SLC}(\alpha)$ then $\mathsf{T}(\mathcal{M}, w) \models \delta_1 \vee \dots \vee \delta_n$. This is absurd because $\mathsf{T}(\mathcal{M}, w) \not\models \delta_i$ for every i . \square

As $\Sigma(x)$ is satisfiable in \mathbf{K} we have a model \mathcal{N} and an element v such that $\mathsf{T}(\mathcal{N}, v) \models \Sigma(x)$. We make the following proposition.

Proposition 4.2.7. $\mathcal{N}, v \sqsubseteq_{\mathcal{L}} \mathcal{M}, w$.

Proof. Take the contrapositive. Suppose that $\mathcal{M}, w \not\models \varphi$ then $\neg \mathsf{Tf}_x(\varphi) \in \text{NTh}^w(x)$ and because $\text{NTh}^w(x) \subseteq \Sigma(x)$ we can state that $\mathsf{T}(\mathcal{N}, v) \models \neg \mathsf{Tf}_x(\varphi)$ which implies that $\mathsf{T}(\mathcal{N}, v) \not\models \mathsf{Tf}_x(\varphi)$. By truth-preservation of the translations we get $\mathcal{N}, v \not\models \varphi$. \square

We will need to link $\mathsf{T}(\mathcal{M}, w)$ and $\mathsf{T}(\mathcal{N}, v)$ in a way that lets us transfer the validity of $\alpha(x)$ from the second model to the first one. The next lemma will come handy.

Lemma 4.2.8 (Big Detour Lemma). Let $\alpha(x) \in \text{FORM}(\mathfrak{F}^1)$ be \mathcal{L} -bisimulation \mathbf{K} -invariant, if $\mathcal{N}, v \sqsubseteq_{\mathcal{L}} \mathcal{M}, w$ and $\mathsf{T}(\mathcal{N}, v) \models \alpha(x)$ then $\mathsf{T}(\mathcal{M}, w) \models \alpha(x)$.

Proof. We define some names to avoid cumbersome notation in this proof. We add a subscript f to the first-order translations of \mathcal{L} models, we add a superscript $+$ to first-order saturated models and a superscript $*$ to modal saturated models.

Applying Theorem A.3 to \mathcal{M}, w and \mathcal{N}, v (with $M_1 = M_2 = \text{MODS}(\mathcal{L})$) we build up new models. The theorem explicitly states the relationship among them, we will use this result to prove this lemma. Hereafter we will use the same notation as in Theorem A.3.

The following diagram helps to illustrate the actual situation along with the relationship between the various models. Think of it as a cube. The front face represents the models from the source language and the back face has the models from the first-order language.

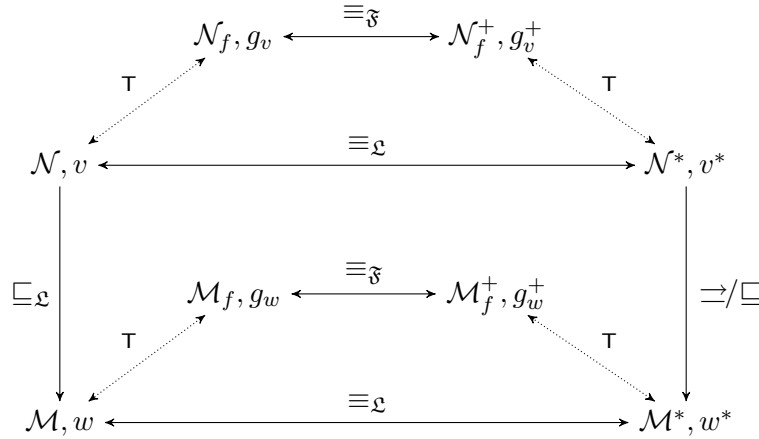


Fig. 4.1: Directions for the detour.

With this new notation the Big Detour Lemma can be restated as follows: Let $\alpha(x)$ be an \mathcal{L} -bisimulation \mathbf{K} -invariant \mathfrak{F} -formula, if $\mathcal{N}, v \sqsubseteq_{\mathcal{L}} \mathcal{M}, w$ and $\mathcal{N}_f, g_v \models \alpha(x)$ then $\mathcal{M}_f, g_w \models \alpha(x)$.

Using a simple diagram chase argument we can see that, as $\mathcal{N}_f, g_v \models \alpha(x)$ and \mathcal{N}_f^+, g_v^+ is elementary equivalent to \mathcal{N}_f, g_v , then $\mathcal{N}_f^+, g_v^+ \models \alpha(x)$. Because $\alpha(x)$ is invariant under \mathcal{L} -simulations and $\mathcal{N}^*, v^* \Rightarrow_{\mathcal{L}} \mathcal{M}^*, w^*$ we know that $\mathcal{M}_f^+, g_w^+ \models \alpha(x)$. Again by elementary equivalence we conclude that $\mathcal{M}_f, g_w \models \alpha(x)$ which is what we wanted to prove. \square

Applying this lemma to \mathcal{M}, w and \mathcal{N}, v and having transferred the validity of $\alpha(x)$ from $\mathbb{T}(\mathcal{N}, v)$ to $\mathbb{T}(\mathcal{M}, w)$ we can conclude that $\text{SLC}(\alpha) \models \alpha(x)$. With this final affirmation we have just proved the right-to-left direction of the characterization result.

4.3 Definability

The study of class definability is not new. There exist, for example, several results for first order logic regarding the definability of classes of models. In that case, a class of models that is definable by means of a set of first order formulas is called *elementary* and those that can be defined by means of a single formula are called *basic elementary* classes.

To develop the theory of this section we will use a relativized version of the concept of first order definability. It is defined as follows.

Definition 4.3.1 (C-elementary class). Let $C \subseteq \text{MODS}(\mathfrak{F})$.

1. A class $K \subseteq C$ is called *C-elementary* (noted $C\text{-}EC_{\Delta}$) if there exists a set of first order formulas Γ such that for all $\mathcal{M}^f \in C$ it occurs that $\mathcal{M}^f \models \Gamma$ iff $\mathcal{M}^f \in K$.²
2. A class $K \subseteq C$ is called *basic C-elementary* (noted $C\text{-}EC$) if there exists a first order formula φ such that for all $\mathcal{M}^f \in C$ it occurs that $\mathcal{M}^f \models \varphi$ iff $\mathcal{M}^f \in K$.

Definition 4.3.2 (Elementary class). Let $K \subseteq \text{MODS}(\mathfrak{F})$.

1. K is called *elementary* (noted EC_{Δ}) if it is C-elementary for $C = \text{MODS}(\mathfrak{F})$.
2. K is called *basic elementary* (noted EC) if it is basic C-elementary for $C = \text{MODS}(\mathfrak{F})$.

On the modal side, we will use *pointed models* for a smoother proof. We need some further definitions before stating the main theorem of this section. The concept of ‘definability’ in the source logic is given analogously to the one of the target logic.

Definition 4.3.3 (Definability). A class $M \subseteq \text{PMODS}(\mathfrak{L})$ is said to be *definable by a set of formulas* if there exists a set Γ of \mathfrak{L} -formulas such that $\langle \mathcal{M}, w \rangle \in M$ if and only if $\mathcal{M}, w \models \Gamma$.

Definition 4.3.4 (Closure under simulations). A class $M \subseteq \text{PMODS}(\mathfrak{L})$ is said to be *closed under simulations* if, whenever $\langle \mathcal{M}, w \rangle \in M$, and $\langle \mathcal{N}, v \rangle$ is an \mathfrak{L} -pointed model such that $\mathcal{M}, w \rightrightarrows_{\mathfrak{L}} \mathcal{N}, v$ then $\langle \mathcal{N}, v \rangle \in M$.

As in first order and BML, we distinguish between two types of classes. Those that can be defined by a set of formulas and those that can be defined by a single formula. Here we state the first theorem and then carry on with the second one.

Theorem 4.3.5 (Definability by a set). Given an adequate pair $\langle \mathfrak{L}, K \rangle$ and a class of pointed models $M \subseteq \text{PMODS}(\mathfrak{L})$, the following are equivalent

- (i) M is definable by a set of \mathfrak{L} -formulas.
- (ii) M is closed under \mathfrak{L} -simulations, $\text{T}(M)$ is closed under ultraproducts and $\text{T}(\overline{M})$ is closed under ultrapowers.

From i to ii. Suppose that M is defined by the set Γ of \mathfrak{L} -formulas.

1. Suppose now that there is a model $\langle \mathcal{M}, w \rangle \in M$ such that $\mathcal{M}, w \rightrightarrows_{\mathfrak{L}} \mathcal{N}, v$ for some pointed model \mathcal{N}, v . As $\langle \mathcal{M}, w \rangle \in M$ it must occur that $\mathcal{M}, w \models \Gamma$. By simulation preservation we have $\mathcal{N}, v \models \Gamma$ therefore $\langle \mathcal{N}, v \rangle \in M$. Therefore M is closed under \mathfrak{L} -simulations.

² A $C\text{-}EC_{\Delta}$ class can be seen as the intersection of $C\text{-}EC$ classes. The Δ in the notation comes from the german word *Durchschnitt* which means ‘cross-section’ and makes reference to this fact.

2. To see that $\mathsf{T}(\mathsf{M})$ is closed under ultraproducts take a family of models $\langle \mathcal{M}_i^f, g_i \rangle \in \mathsf{T}(\mathsf{M})$. Because every \mathcal{M}_i^f, g_i is in $\mathsf{T}(\mathsf{M})$ we have that $\mathcal{M}_i^f, g_i \models \mathsf{Tf}_x(\Gamma)$ for all i . Let $\mathcal{M}^f = \prod_D \mathcal{M}_i^f$ be an ultraproduct of those models, by [CK90, Theorem 4.1.9] we have that $\mathcal{M}^f, g_i^* \models \mathsf{Tf}_x(\Gamma)$ for $g_i^*(x) = \lambda z.g_i(x)$. Therefore $\langle \mathcal{M}^f, g_i^* \rangle \in \mathsf{T}(\mathsf{M})$. Thus, the class is closed under ultraproducts.³
3. We still have to check that $\mathsf{T}(\overline{\mathsf{M}})$ is closed under ultrapowers. Take $\langle \mathcal{M}^f, g \rangle \in \mathsf{T}(\overline{\mathsf{M}})$, by definition $\mathcal{M}^f, g \not\models \mathsf{Tf}_x(\Gamma)$. Let $\mathcal{M}_*^f = \prod_D \mathcal{M}^f$ be an ultrapower of \mathcal{M}^f , by [CK90, Corollary 4.1.10] the ultrapower is elementary equivalent to the original model. Hence, let $h(x) = \lambda z.g(x)$ be the canonical mapping, $\mathcal{M}_*^f, h \not\models \mathsf{Tf}_x(\Gamma)$. This means that $\langle \mathcal{M}_*^f, h \rangle \in \mathsf{T}(\overline{\mathsf{M}})$ and therefore the class is closed under ultrapowers. \square

From *ii* to *i*. Suppose M is closed under \mathcal{L} -simulations, $\mathsf{T}(\mathsf{M})$ is closed under ultraproducts and $\mathsf{T}(\overline{\mathsf{M}})$ is closed under ultrapowers. Define the set $\Gamma = \mathsf{Th}(\mathsf{M})$. Trivially $\mathsf{M} \models \Gamma$, we still have to show that if $\mathcal{M}, w \models \Gamma$ then $\langle \mathcal{M}, w \rangle \in \mathsf{M}$. Define the following set

$$\mathsf{NTh}^w(x) = \{ \neg \mathsf{Tf}_x(\varphi) : \varphi \text{ is an } \mathcal{L}\text{-formula and } \mathcal{M}, w \not\models \varphi \}$$

Let's see that $\mathsf{NTh}^w(x)$ is finitely satisfiable in $\mathsf{T}(\mathsf{M})$. Suppose not, there is a finite subset $\Sigma_0 \subseteq \mathsf{NTh}^w(x)$ such that $\Sigma_0 = \{ \neg \sigma_1, \dots, \neg \sigma_n \}$ is not satisfiable in $\mathsf{T}(\mathsf{M})$. That means that the formula $\psi = \neg(\neg \sigma_1 \wedge \dots \wedge \neg \sigma_n)$ is valid in $\mathsf{T}(\mathsf{M})$. Observe that ψ is equivalent to $\psi' = \sigma_1 \vee \dots \vee \sigma_n$. As the formula translation preserves disjunction and truth there exists an \mathcal{L} -formula ψ^* such that $\psi' \equiv_{\mathfrak{F}} \mathsf{Tf}_x(\psi^*)$. Hence $\mathsf{Tf}_x(\psi^*)$ is valid in $\mathsf{T}(\mathsf{M})$ and therefore $\psi^* \in \Gamma$. This is absurd because it is obvious that $\mathcal{M}, w \not\models \psi^*$ and by hypothesis $\mathcal{M}, w \models \Gamma$.

Having proved that every subset of $\mathsf{NTh}^w(x)$ is satisfiable, by relative compactness, there is a model $\langle \mathcal{N}, v \rangle \in \mathsf{M}$ such that $\mathsf{T}(\mathcal{N}, v) \models \mathsf{NTh}^w(x)$. We have already proved (in Proposition 4.2.7) that these models satisfy $\mathcal{N}, v \sqsubseteq_{\mathcal{L}} \mathcal{M}, w$.

Suppose that $\langle \mathcal{M}, w \rangle \in \overline{\mathsf{M}}$, using Theorem A.3 (with $\mathsf{M}_1 = \mathsf{M}$ and $\mathsf{M}_2 = \overline{\mathsf{M}}$) we can conclude that there exist models $\langle \mathcal{N}^*, v^* \rangle \in \mathsf{M}$ and $\langle \mathcal{M}^*, w^* \rangle \in \overline{\mathsf{M}}$ such that $\mathcal{N}^*, v^* \sqsupseteq_{\mathcal{L}} \mathcal{M}^*, w^*$. As M is closed under simulations then $\langle \mathcal{M}, w \rangle \in \mathsf{M}$. Absurd, therefore $\langle \mathcal{M}, w \rangle$ must be in M . \square

Notation 4.3.6. Let $\langle \mathcal{M}^f, g \rangle, \langle \mathcal{N}^f, h \rangle \in \mathsf{PMODS}(\mathfrak{F})$ we write $\mathcal{M}^f, g \cong \mathcal{N}^f, h$ to mean that there exists a potential isomorphism I between \mathcal{M}^f and \mathcal{N}^f such that $\langle a \rangle I \langle b \rangle$ where $a = g(x)$ and $b = h(x)$. That is, there is a potential isomorphism that links the elements assigned by g and h .

Definition 4.3.7 (*C-closure under potential isomorphisms*). Let $\mathsf{C} \subseteq \mathsf{MODS}(\mathfrak{F})$. A class $\mathsf{K} \subseteq \mathsf{C}$ is *C-closed under potential isomorphisms* if for every $\mathcal{M}^f \in \mathsf{K}$ and $\mathcal{N}^f \in \mathsf{C}$ such that $\mathcal{M}^f \cong \mathcal{N}^f$ then $\mathcal{N}^f \in \mathsf{K}$.

The definition for pointed models is similar. Let $\mathsf{C} \subseteq \mathsf{PMODS}(\mathfrak{F})$. A class $\mathsf{K} \subseteq \mathsf{C}$ is *C-closed under potential isomorphisms* if for every $\langle \mathcal{M}^f, g \rangle \in \mathsf{K}$ and $\langle \mathcal{N}^f, h \rangle \in \mathsf{C}$ such that $\mathcal{M}^f, g \cong \mathcal{N}^f, h$ then $\langle \mathcal{N}^f, h \rangle \in \mathsf{K}$.

Lemma 4.3.8. Let $\mathsf{M} \subseteq \mathsf{PMODS}(\mathcal{L})$. If M is closed under \mathcal{L} -simulations and both $\mathsf{T}_{\mathsf{K}}(\mathsf{M})$ and $\mathsf{T}_{\mathsf{K}}(\overline{\mathsf{M}})$ are closed under ultrapowers then $\mathsf{T}_{\mathsf{K}}(\mathsf{M})$ and $\mathsf{T}_{\mathsf{K}}(\overline{\mathsf{M}})$ are K -closed under potential isomorphisms.

³ This application is a corollary of The Fundamental Theorem of Ultraproducts. This same application can be seen in the proof of Theorem's 4.1.12 in the same book.

Proof for $\mathsf{T}(\mathsf{M})$. Suppose that $\mathsf{T}(\mathsf{M})$ is not K -closed under potential isomorphisms. This means that there exist models $\langle \mathcal{M}^f, g \rangle \in \mathsf{T}(\mathsf{M})$ and $\langle \mathcal{N}^f, h \rangle \in \mathsf{T}(\overline{\mathsf{M}})$ such that $\mathcal{M}^f, g \cong \mathcal{N}^f, h$. Recall that $\mathsf{K} \setminus \mathsf{T}(\mathsf{M}) = \mathsf{T}(\overline{\mathsf{M}})$. For a smoother proof, call their modal counterparts \mathcal{M}, w and \mathcal{N}, v respectively. Therefore $\langle \mathcal{M}, w \rangle \in \mathsf{M}$ and $\langle \mathcal{N}, v \rangle \notin \mathsf{M}$.

As $\mathcal{M}^f, g \cong \mathcal{N}^f, h$ we know by [CK90, Proposition 2.4.4] that $\mathcal{M}^f, g \models \varphi(x)$ if and only if $\mathcal{N}^f, h \models \varphi(x)$. In particular they have the same modal theory, $\mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$. As this implies that $\mathcal{M}, w \sqsubseteq_{\mathcal{L}} \mathcal{N}, v$ we can use Theorem A.3 (instantiating with $\mathsf{K}_1 = \mathsf{T}(\mathsf{M})$, $\mathsf{K}_2 = \mathsf{T}(\overline{\mathsf{M}})$ and \mathcal{M}, \mathcal{N} interchanged) and get models $\langle \mathcal{M}^*, w^* \rangle \in \mathsf{M}$ and $\langle \mathcal{N}^*, v^* \rangle \in \overline{\mathsf{M}}$ such that $\mathcal{M}^*, w^* \sqsupseteq_{\mathcal{L}} \mathcal{N}^*, v^*$.

Knowing that $\mathcal{M}^*, w^* \sqsupseteq_{\mathcal{L}} \mathcal{N}^*, v^*$ and that M is closed under simulations we conclude that $\langle \mathcal{N}^*, v^* \rangle \in \mathsf{M}$. This is absurd because it contradicts $\langle \mathcal{N}^*, v^* \rangle \in \overline{\mathsf{M}}$. Hence $\mathsf{T}_{\mathsf{K}}(\mathsf{M})$ is K -closed under potential isomorphisms. \square

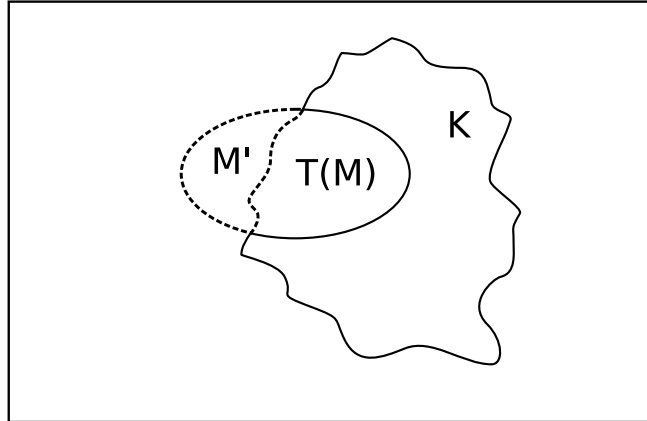
Proof for $\mathsf{T}(\overline{\mathsf{M}})$. To see that $\mathsf{T}(\overline{\mathsf{M}})$ is K -closed under potential isomorphisms we argue by contradiction. Suppose not, then there exist $\langle \mathcal{M}^f, g \rangle \in \mathsf{T}(\overline{\mathsf{M}})$ and $\langle \mathcal{N}^f, h \rangle \in \mathsf{K} \setminus \mathsf{T}(\overline{\mathsf{M}})$ such that $\mathcal{M}^f, g \cong \mathcal{N}^f, h$. As $\langle \mathcal{N}^f, h \rangle \in \mathsf{K} \setminus \mathsf{T}(\overline{\mathsf{M}})$ this means that $\langle \mathcal{N}^f, h \rangle \in \mathsf{T}(\mathsf{M})$. We have just proved that $\mathsf{T}(\mathsf{M})$ is K -closed under potential isomorphism then, as $\mathcal{M}^f, g \cong \mathcal{N}^f, h$, we conclude that $\langle \mathcal{M}^f, g \rangle \in \mathsf{T}(\mathsf{M})$ which contradicts our hypothesis.⁴ Absurd. \square

Theorem 4.3.9 (Definability by a single formula). Given an adequate pair $\langle \mathcal{L}, \mathsf{K} \rangle$, and a class of models $\mathsf{M} \subseteq \text{MODS}(\mathcal{L})$, the following are equivalent

- (i). M is definable by a single \mathcal{L} -formula.
- (ii). M is closed under \mathcal{L} -simulations and both $\mathsf{T}(\mathsf{M})$ and $\mathsf{T}(\overline{\mathsf{M}})$ are closed under ultraproducts.

From *i* to *ii*. Suppose M is definable by a single \mathcal{L} -formula φ .

1. Let's see that $\mathsf{T}(\mathsf{M})$ and $\mathsf{T}(\overline{\mathsf{M}})$ are closed under ultraproducts. Recall that M is definable by a single \mathcal{L} -formula φ . Take the class of first order models defined by $\text{Tf}_x(\varphi)$ and call it M^e . Observe that M^e can be expressed as the disjunct union $\mathsf{M}^e = \mathsf{T}(\mathsf{M}) \cup \mathsf{M}'$ between the translation of M and some other models that do not fall in K . Therefore $\mathsf{T}(\overline{\mathsf{M}}) = \overline{\mathsf{M}^e} \cap \mathsf{K} = \text{MODS}(\neg \text{Tf}_x(\varphi)) \cap \mathsf{K}$. The following diagram helps illustrate the different classes. The box represents the class of all \mathfrak{F} models, K is the class with an irregular border and M^e is the oval.



⁴ Here we use the symmetry of the potential isomorphism relation.

Using Theorem 4.3.5, as M is defined by the singleton set $T = \{\varphi\}$, $T(M)$ is closed under ultraproducts. To see that $T(\overline{M})$ is closed under ultraproducts proceed as follows.

- (a) As K is closed under ultraproducts then, any ultraproduct from K must reside in K . In particular, any ultraproduct from $T(\overline{M})$ must reside in K .
- (b) As \overline{M}^e is defined by $\neg T f_x(\varphi)$, it is closed under ultraproducts [CK90, Corollary 6.1.16]. This means that any ultraproduct from \overline{M}^e must reside in \overline{M}^e , in particular, any ultraproduct from $T(\overline{M})$ must reside in \overline{M}^e .

From (a) and (b) we can conclude that any ultraproduct from $T(\overline{M})$ must be in $K \cap \overline{M}^e = T(\overline{M})$.

2. Using Theorem 4.3.5, as M is defined by the singleton set $T = \{\varphi\}$ we can be sure that M is closed under \mathcal{L} -simulations. \square

From ii to i. Suppose that M is closed under \mathcal{L} -simulations and both $T(M)$ and $T(\overline{M})$ are closed under ultraproducts. Using Theorem 4.3.5 we have a set of formulas Γ defining M . By Lemma 4.3.8 $T(M)$ and $T(\overline{M})$ are K -closed under potential isomorphisms. Now we use the relativized version of first order's definability result, Theorem A.2, and conclude that there is a first order formula $\alpha(x)$ such that for every $\langle \mathcal{M}^f, g \rangle \in K$; $\mathcal{M}^f, g \models \alpha(x)$ if and only if $\langle \mathcal{M}^f, g \rangle \in T(M)$.

As M is closed under \mathcal{L} -simulations then α is K -invariant for \mathcal{L} -simulations. Using the Characterization Theorem (Theorem 4.2.3) we can conclude that $\alpha(x)$ is K -equivalent to the translation of a modal formula φ . Therefore there exists φ that defines M . \square

5. APPLICATIONS

In this chapter we will use the results that we have developed in the previous chapters and derive the characterization and definability theorems for particular cases of modal logics.

5.1 Memory Logics

Memory logics are a novel family of modal logics introduced in [AFFM08]. They allow to model dynamic behavior through explicit memory operators that *change* the evaluating structure. This proposal introduces a framework for studying the notion of state in a more general way, without bounding the analysis to any fixed domain (like knowledge change, time flow, linguistics contexts, etc.). Most of the work that has been done in this direction implicitly adds some specific native behavior in the “dynamic component”. The approach presented in this paper wants to study some of the dynamic capabilities of the above mentioned approaches from a more abstract point of view, and analyze the different aspects of this family in terms of logic properties.

This family of logics present several “memory operators” that can be considered modularly. We first present the syntax, signature and models for a broad set of operators and then analyze different possible combinations which form interesting logics.

It is important to note that there are no Characterization and Definability results known for this family of logics at this moment. Therefore, the results obtained through the use of the framework that we’ve developed will be original. We present the results for the unimodal case but it can be easily generalized for the multi-modal case.

Definition 5.1.1 (Signatures). Let $\text{PROP} = \{p_1, p_2, \dots\}$ (the propositional symbols) be a countable infinite set of symbols and $\text{REL} = \{r\}$ (the relational symbols) be disjoint. The source signature is defined to be $\mathcal{S} = \langle \text{PROP}, \text{REL} \rangle$.

Let $\text{FPROP} = \{P_1, P_2, \dots\}$ (the propositional predicates) and $\text{FREL} = \{R\}$ (the relational predicates) The target first-order signature is defined to be $\mathcal{T} = \langle \text{FPROP} \cup \text{FREL}, \emptyset, \emptyset \rangle$ with equality.

Definition 5.1.2 (Syntax). The syntax of the Memory Logics family over a given signature $\langle \text{PROP}, \text{REL} \rangle$ is defined as an extension of the propositional calculus with the following operators:

$$\varphi ::= \dots \mid \mathbb{k} \mid \mathbb{r}\varphi \mid \mathbb{e}\varphi \mid \mathbb{f}\varphi \mid \langle r \rangle \varphi \mid \langle\langle r \rangle\rangle \varphi$$

where $r \in \text{REL}$. We define the dual of $\langle\langle r \rangle\rangle$ in the usual way: for all $r \in \text{REL}$, $\llbracket r \rrbracket \varphi$ can be defined as $\neg \langle\langle r \rangle\rangle \neg \varphi$. We usually call these operators ‘known’, ‘remember’, ‘erase’, ‘forget’ and ‘double diamond’. Every logic of this family will be *required* to have at least the \mathbb{r} and \mathbb{k} operators and can have any combination of the other operators.

Observe that, defined this way, in the family of memory logics one has two types of diamonds: the ‘single’ diamond of BML and the ‘double diamond’. The semantic of the first is defined as usual and the later will be defined shortly.

The family of memory logics that we will be working with are all evaluated in an extension of Kripke models with a set that we call the ‘memory’ of the model. It is defined as follows, along with its first-order equivalent.

Definition 5.1.3 (Models). A *model* for the source language over the signature \mathcal{S} is a tuple $\mathcal{M} = \langle W, R, V, S \rangle$ satisfying

- (i) W is a nonempty set,
- (ii) $R \subseteq W \times W$ is a binary relation over W ,
- (iii) $V : \text{PROP} \rightarrow \mathcal{P}(W)$ is a valuation function, and
- (iv) $S \subseteq W$ is the memory of the model.

An \mathfrak{F} -*model* for the target language is a tuple $\mathcal{M}^t = \langle W^t, R^t, (P_i^t)_{i \in \text{FPROP}}, K \rangle$ where

- (i) W^t is a nonempty set,
- (ii) $K \subseteq W$,
- (iii) $R^t \subseteq W^t \times W^t$ is a binary relation over W^t , and
- (iv) $(P_i^t)_{i \in \text{FPROP}}$ are unary relations over W^t .

Notation 5.1.4. In the rest of this section the following notation will be useful. Let $\mathcal{M} = \langle W, R, V, S \rangle$ be a model, $w \in W$, and $S' \subseteq W$ then we define

$$\begin{aligned} \mathcal{M}[+w] &= \langle W, R, V, S \cup \{w\} \rangle \\ \mathcal{M}[-w] &= \langle W, R, V, S \setminus \{w\} \rangle \\ \mathcal{M}[+S'] &= \langle W, R, V, S \cup S' \rangle \\ \mathcal{M}[-S'] &= \langle W, R, V, S \setminus S' \rangle \\ \mathcal{M}[*] &= \langle W, R, V, \emptyset \rangle. \end{aligned}$$

We usually write $\mathcal{M}[w]$ instead of $\mathcal{M}[+w]$.

Definition 5.1.5 (Semantics). Given a model $\mathcal{M} = \langle W, R, V, S \rangle$ and $w \in W$, we extend the *propositional* part of the semantics presented in Definition 1.2.4 with the following rules:

$$\begin{aligned} \mathcal{M}, w \models \mathbb{K} &\text{ iff } w \in S \\ \mathcal{M}, w \models \mathbb{R}\varphi &\text{ iff } \mathcal{M}[w], w \models \varphi \\ \mathcal{M}, w \models \mathbb{F}\varphi &\text{ iff } \mathcal{M}[-w], w \models \varphi \\ \mathcal{M}, w \models \mathbb{E}\varphi &\text{ iff } \mathcal{M}[*], w \models \varphi \\ \mathcal{M}, w \models \langle r \rangle \varphi &\text{ iff } \exists w' \in W, wRw' \text{ and } \mathcal{M}, w' \models \varphi. \\ \mathcal{M}, w \models \langle\langle r \rangle\rangle \varphi &\text{ iff } \exists w' \in W, wRw' \text{ and } \mathcal{M}[w], w' \models \varphi. \end{aligned}$$

Observe that the double diamond acts as a normal diamond *but* it always remembers the current state before moving. Hence, it can be thought as if the formula were leaving a trace while being evaluated in the model.

Definition 5.1.6 (Formula translation). We will not give an explicit translation for these logics, as we will not use it explicitly. We know there exists a translation to \mathfrak{F}_{\perp}^1 because, in [AFFM08], there is an explicit translation from memory logics to \mathcal{HL} and there is a translation from \mathcal{HL} to \mathfrak{F}_{\perp}^1 given in [BvBW06]. This translation preserves both conjunction and disjunction as needed. Let's call it $\text{Tf}_x : \text{FORM}(\mathcal{S}) \rightarrow \text{FORM}(\mathcal{T})$.

Definition 5.1.7 (Model translation). Let \mathbf{K} be the class of all models for the signature \mathcal{T} . Let $\mathcal{M} = \langle W, R, V, S \rangle$ and $\mathcal{M}^t = \langle W^t, R^t, (P_i^t)_{i \in \text{FPROP}}, K \rangle$. Define the *model translation* $\text{T}_{\mathbf{K}}(\mathcal{M}, w) = \langle \mathcal{M}^t, g^t \rangle$ to be the function induced by the following equations

$$\begin{aligned}
W^t &= W \\
P_i^t &= V(p_i) \\
K &= S \\
R^t &= R \\
g^t(x) &= w
\end{aligned}$$

The simulation notion for a logic of this family allows a very modular definition. Let \sim be a binary relation between memory pointed models. So \sim relates tuples $\langle \mathcal{M}, m \rangle$ with $\langle \mathcal{N}, n \rangle$.

A simulation for a memory logic \mathfrak{L} can be defined imposing restrictions to \sim depending on the operators that \mathfrak{L} has. In the following table we summarize the restrictions associated with each operator. We write $S^{\mathcal{M}}$ to refer to the memory of the model \mathcal{M} . R_r^1 is used to denote a relation in \mathcal{M} and R_r^2 is used to denote a relation in \mathcal{N} .

<i>always</i>	(nontriv)	\sim is not empty.
<i>always</i>	(agree)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then m and n make the same propositional variables true.
\mathbb{K}	(kagree)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then $m \in S^{\mathcal{M}}$ if and only if $n \in S^{\mathcal{N}}$.
\mathbb{R}	(remember)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then $\langle \mathcal{M}[m], m \rangle \sim \langle \mathcal{N}[n], n \rangle$.
\mathbb{F}	(forget)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then $\langle \mathcal{M}[-m], m \rangle \sim \langle \mathcal{N}[-n], n \rangle$.
\mathbb{E}	(erase)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then $\langle \mathcal{M}[*], m \rangle \sim \langle \mathcal{N}[*], n \rangle$.
$\langle r \rangle$	(forth)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$ and $R_r^1(m, m')$, then there exists $n' \in \mathcal{N} $ such that $R_r^2(n, n')$ and $\langle \mathcal{M}, m' \rangle \sim \langle \mathcal{N}, n' \rangle$.
	(back)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$ and $R_r^2(n, n')$, then there exists $m' \in \mathcal{M} $ such that $R_r^1(m, m')$ and $\langle \mathcal{M}, m' \rangle \sim \langle \mathcal{N}, n' \rangle$.
$\langle\langle r \rangle\rangle$	(mforth)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$ and $R_r^1(m, m')$, then there exists $n' \in \mathcal{N} $ such that $R_r^2(n, n')$ and $\langle \mathcal{M}[m], m' \rangle \sim \langle \mathcal{N}[n], n' \rangle$.
	(mback)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$ and $R_r^2(n, n')$, then there exists $m' \in \mathcal{M} $ such that $R_r^1(m, m')$ and $\langle \mathcal{M}[m], m' \rangle \sim \langle \mathcal{N}[n], n' \rangle$.

Fig. 5.1: Operator restrictions for a modular memory simulation definition.

Definition 5.1.8 (Memory simulation). From now on, given a memory logic \mathfrak{L} , we will refer as ‘the simulation for \mathfrak{L} ’ to the simulation defined by the sum of the necessary conditions of Figure 5.1 for the operators in \mathfrak{L} .

Observe that, as every memory logic has negation, the simulation notion for memory logics will be symmetrical. Therefore we will use the bisimulation symbol \Leftrightarrow to note memory simulations.

Example 5.1.9. To give some examples of possible memory logics we cite the following ones which are named in [Mer09].

$$\begin{aligned}
\mathcal{ML}(\langle r \rangle) &= \{ \mathbb{R}, \mathbb{K}, \langle r \rangle \} \\
\mathcal{ML}(\langle\langle r \rangle\rangle) &= \{ \mathbb{R}, \mathbb{K}, \langle\langle r \rangle\rangle \} \\
\mathcal{ML}(\langle r \rangle, \mathbb{F}) &= \{ \mathbb{R}, \mathbb{K}, \mathbb{F}, \langle r \rangle \} \\
\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{F}) &= \{ \mathbb{R}, \mathbb{K}, \mathbb{F}, \langle\langle r \rangle\rangle \} \\
\mathcal{ML}(\langle r \rangle, \mathbb{E}) &= \{ \mathbb{R}, \mathbb{K}, \mathbb{E}, \langle r \rangle \} \\
\mathcal{ML}(\langle\langle r \rangle\rangle, \mathbb{E}) &= \{ \mathbb{R}, \mathbb{K}, \mathbb{E}, \langle\langle r \rangle\rangle \} \\
\mathcal{ML}(\langle r \rangle, \mathbb{F}, \mathbb{E}) &= \{ \mathbb{R}, \mathbb{K}, \mathbb{F}, \mathbb{E}, \langle r \rangle \}
\end{aligned}$$

It is clear from the satisfaction definition of each operator that these logics have different capabilities. A detailed insight on the expressive power of these logics can be found in the aforementioned PhD. thesis.

Theorem 5.1.10. Let \mathcal{L} be a memory logic, let $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ be two memory models. If $\langle \mathcal{M}, w \rangle \Leftrightarrow_{\mathcal{L}} \langle \mathcal{N}, v \rangle$ then $\langle \mathcal{M}, w \rangle \equiv_{\mathcal{L}} \langle \mathcal{N}, v \rangle$.

Proof. Part of the proof can be found in [Mer09], it can be easily extended to the full set of operators. We will not present this proof here as it exceeds the focus of this thesis. \square

Before starting with the proof of the main theorem of this section we will prove some lemmas that will be useful. The model may change during the evaluation of a formula. For our special case, it will be enough to prove that adding a state to the memory preserves ω -saturation.

Lemma 5.1.11. If \mathcal{M} is ω -saturated then $\mathcal{M}[+A]$ is ω -saturated too for all finite $A \subseteq |\mathcal{M}|$.

Proof. The proof of this lemma can be found in [Mer09, Lemma 5.2.2]. \square

Lemma 5.1.12. Let $\mathcal{M} = \langle W, R, \dots \rangle$ be a ω -saturated Kripke model whose translation preserves the structure of the domain and the relations, that is, $\mathbb{T}(\mathcal{M}) = \langle W, R, \dots \rangle$ where $R \subseteq W \times W$.

Let Σ be a set of modal formulas and $w \in W$. If every finite subset $\Delta \subseteq \Sigma$ satisfies $\mathcal{M}, v_{\Delta} \models \Delta$ where v_{Δ} is an R -successor of w then there exists v , an R -successor of w , such that $\mathcal{M}, v \models \Sigma$.

Proof. Recall that the definition of ω -saturation lets us extend the first order language with a constant \underline{a} for each element $a \in W$. Define $\Sigma^* = \{R\underline{w}x\} \cup \mathbb{T}_x(\Sigma)$.

If we show that Σ^* is satisfiable in some pointed model $\mathbb{T}(\mathcal{M}, v)$ it is clear that Σ will be satisfiable in a successor of w . This is because the domain and relations of \mathcal{M} and $\mathbb{T}(\mathcal{M})$ are the same and if $\mathbb{T}(\mathcal{M}) \models R\underline{w}x[\underline{a}]$ this means that \underline{a} is a successor of w . The rest of the proof will focus on proving that Σ^* is satisfiable in the pointed model $\mathbb{T}(\mathcal{M}, v)$.

Take a finite subset $\Sigma_0 \subseteq \Sigma^*$. Observe that this set should satisfy the following inclusion $\Sigma_0 \subseteq \{R\underline{w}x, \sigma_1, \dots, \sigma_n\}$ with $\sigma_i \in \mathbb{T}_x(\Sigma)$. Therefore, if we show that this bigger set is satisfiable, it will also be the case with Σ_0 .

By hypothesis, every finite subset of Σ is satisfiable in a successor of w . Take the finite subset Δ such that $\mathbb{T}_x(\Delta) = \{\sigma_1, \dots, \sigma_n\}$. This Δ is satisfiable in some successor v_{Δ} which means that $R(w, v_{\Delta})$. We can conclude that $\mathbb{T}(\mathcal{M}, v_{\Delta}) \models R\underline{w}x$ and $\mathbb{T}(\mathcal{M}, v_{\Delta}) \models \mathbb{T}_x(\Delta)$.

We have taken an arbitrary finite subset of Σ^* and shown that it is satisfiable. By ω -saturation we can conclude that the set Σ^* is also satisfiable. \square

To be able to derive the characterization and definability results using the framework developed in the previous chapters we need to prove that, for every memory logic \mathcal{L} , the class of ω -saturated models has the Hennessy-Milner property with respect to \mathcal{L} -simulations. Each logic will have its own definition of simulation with the proper restrictions listed above.

As we want to consider all the possible logics from the family of memory logics we will need to prove that, given two models $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ such that $\langle \mathcal{M}, w \rangle \equiv_{\mathcal{L}} \langle \mathcal{N}, v \rangle$ we can construct an \mathcal{L} -simulation between them. We will do this by considering every possible operator and show that we can construct a simulation that satisfies the constraints associated for that operator.

Theorem 5.1.13. Let \mathcal{L} be a memory logic, the class of ω -saturated models has the Hennessy-Milner property with respect to \mathcal{L} -simulations.

Proof. Given two ω -saturated models \mathcal{M}, \mathcal{N} it suffices to give an \mathcal{L} -simulation between them. We propose the binary relation \sim defined as

$$\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle \text{ iff } \mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$$

Suppose that $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$. We first show that this relation satisfies the (*nontriv*) and (*agree*) restrictions which apply for every combination of operators, then we will undertake the proof for each special operator.

Basic restrictions. We can see that if we are given two equivalent worlds in two different models then, by definition, the relation will have at least one element and therefore (*nontriv*) will be satisfied. Also, the definition of the relation implies that w and v make true the same propositional variables and therefore (*agree*) is satisfied.

Restrictions for \mathbb{K} . We need to show that w is known in \mathcal{M} if and only if v is known in \mathcal{N} . The proof goes through easily using the satisfaction definition of the known operator

$$w \in S^{\mathcal{M}} \iff \mathcal{M}, w \models \mathbb{K} \iff \mathcal{N}, v \models \mathbb{K} \iff v \in S^{\mathcal{N}}.$$

The first and last implications are because of the semantics of \mathbb{K} , the implication in the middle is because of \mathcal{L} -equivalence between w and v . This proves that (*kagree*) is satisfied.

Restrictions for \mathbb{R} . As we suppose that $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$ then for every φ we have

$$\mathcal{M}, w \models \varphi \iff \mathcal{N}, v \models \varphi$$

so, given a formula ψ we can instantiate the equivalence and get

$$\mathcal{M}, w \models \mathbb{R}\psi \iff \mathcal{N}, v \models \mathbb{R}\psi$$

which by satisfaction definition holds precisely when

$$\mathcal{M}[+w], w \models \psi \iff \mathcal{N}[+v], v \models \psi$$

that means that those two states are equivalent and we can conclude (by def. of \sim) that

$$\langle \mathcal{M}[w], w \rangle \sim \langle \mathcal{N}[v], v \rangle$$

This proves that (*remember*) is satisfied.

Restrictions for \mathbb{F} . As in the last paragraph, for every φ we have

$$\mathcal{M}, w \models \varphi \iff \mathcal{N}, v \models \varphi$$

so, given a formula ψ we can instantiate the equivalence and get

$$\mathcal{M}, w \models \mathbb{F}\psi \iff \mathcal{N}, v \models \mathbb{F}\psi$$

which by satisfaction definition holds precisely when

$$\mathcal{M}[-w], w \models \psi \iff \mathcal{N}[-v], v \models \psi$$

that means that those two states are equivalent and we can conclude (by def. of \sim) that

$$\langle \mathcal{M}[-w], w \rangle \sim \langle \mathcal{N}[-v], v \rangle$$

This proves that (*forget*) is satisfied.

Restrictions for \textcircled{e} . We proceed as usual, for every φ we have

$$\mathcal{M}, w \models \varphi \iff \mathcal{N}, v \models \varphi$$

so, given a formula ψ we can instantiate the equivalence and get

$$\mathcal{M}, w \models \textcircled{e}\psi \iff \mathcal{N}, v \models \textcircled{e}\psi$$

which by satisfaction definition holds precisely when

$$\mathcal{M}[*], w \models \psi \iff \mathcal{N}[*], v \models \psi$$

that means that those two states are equivalent and we can conclude (by def. of \sim) that

$$\langle \mathcal{M}[*], w \rangle \sim \langle \mathcal{N}[*], v \rangle$$

This proves that (*erase*) is satisfied.

Restrictions for $\langle r \rangle$. As we have $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$ then $\mathcal{M}, w \equiv \mathcal{N}, v$. Suppose that w' is a successor of w . Let Σ be the set of all the formulas true at \mathcal{M}, w' . For every finite subset $\Delta \subseteq \Sigma$ we have $\mathcal{M}, w' \models \bigwedge \Delta$ and therefore $\mathcal{M}, w \models \diamond \bigwedge \Delta$. By \mathfrak{L} -equivalence we have $\mathcal{N}, v \models \diamond \bigwedge \Delta$ which means that for every Δ we have a v -successor which satisfies it. By Lemma 5.1.12 we can conclude that there exists v' a v -successor so that $\mathcal{N}, v' \models \Sigma$.

As \mathcal{M}, w' and \mathcal{N}, v' make the same formulas true then they are \mathfrak{L} -equivalent and by definition they will be related by the simulation. We conclude that $\langle \mathcal{M}, w' \rangle \sim \langle \mathcal{N}, v' \rangle$. This proves that (*forth*) is satisfied. The proof for (*back*) is similar but switching the models.

An alternative proof of this lemma, which uses a notion called m-saturation, can be found in [BdRV01].

Restrictions for $\langle\langle r \rangle\rangle$. As we have $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$ then for every φ we have

$$\mathcal{M}, w \models \varphi \iff \mathcal{N}, v \models \varphi$$

therefore if, given an arbitrary ψ we instantiate $\varphi = \textcircled{\text{r}}\psi$ we get¹

$$\mathcal{M}, m \models \textcircled{\text{r}}\psi \iff \mathcal{N}, n \models \textcircled{\text{r}}\psi$$

which, by satisfaction definition holds exactly when

$$\mathcal{M}[m], m \models \psi \iff \mathcal{N}[n], n \models \psi \tag{5.1}$$

Observe that equation 5.1 implies that $\mathcal{M}[m], m \equiv_{\mathfrak{L}} \mathcal{N}[n], n$. Using Lemma 5.1.11 we also know that $\langle \mathcal{M}[m], m \rangle$ and $\langle \mathcal{N}[n], n \rangle$ are both ω -saturated.

Suppose that w' is a successor of w . Let Σ be the set of all the formulas true at $\mathcal{M}[w], w'$. For every finite subset $\Delta \subseteq \Sigma$ we have $\mathcal{M}[w], w' \models \bigwedge \Delta$ and therefore $\mathcal{M}[w], w \models \langle\langle r \rangle\rangle \bigwedge \Delta$. By \mathfrak{L} -equivalence we have $\mathcal{N}[v], v \models \langle\langle r \rangle\rangle \bigwedge \Delta$ which means that for

¹ We can use ‘remember’ here because we required that every memory logic should have it. Without this requirement we would’ve been able to use only $\langle\langle r \rangle\rangle$.

every Δ we have a v -successor which satisfies it. By Lemma 5.1.12 we can conclude that there exists v' a v -successor so that $\mathcal{N}[v], v' \models \Sigma$.

As $\mathcal{M}[w], w'$ and $\mathcal{N}[v], v'$ make the same formulas true then they are \mathfrak{L} -equivalent and by definition they will be related by the simulation. This proves that (*mforth*) is satisfied because $\langle \mathcal{M}[w], w' \rangle \sim \langle \mathcal{N}[v], v' \rangle$. The proof for (*mback*) is similar but switching the models.

After analyzing all the possible operators we have shown that, for every case, the \mathfrak{L} -simulation relation will satisfy the required constraints. This proves that given two ω -saturated \mathfrak{L} -equivalent models we are able to construct an \mathfrak{L} -simulation between them. Therefore, the ω -saturated class of models has the Hennessy-Milner property with respect to \mathfrak{L} -simulations. \square

Corollary 5.1.14. The definitions given above, along with Theorem 5.1.13, prove that the pair $\langle \mathfrak{L}, \mathbf{K} \rangle$ is an adequate pair for every memory logic \mathfrak{L} . Therefore, the Characterization and Definability theorems (4.2.3, 4.3.5 and 4.3.9) hold for this family of logics. In particular, these theorems hold for the logics in Example 5.1.9.

5.2 Hybrid Logics

Hybrid Logics augment modal logics with machinery for describing and reasoning about identity, which is crucial in many settings. The notion of identity comes with the introduction of ‘nominals’ and operators to reason about them. In spirit, nominals work mostly as propositional variables but they have the particularity of being true in *at most* one point. We start by defining the signatures for these logics.

Definition 5.2.1 (Signatures). Let $\text{PROP} = \{p_1, p_2, \dots\}$ (the propositional symbols), $\text{NOM} = \{i_1, i_2, \dots\}$ (the nominal symbols) and $\text{REL} = \{r_1, r_2, \dots\}$ (the relational symbols) be disjoint, countable infinite sets of symbols. The source signature is defined to be $\mathcal{S} = \langle \text{PROP}, \text{REL}, \text{NOM} \rangle$.

Let $\text{FPROP} = \{P_1, P_2, \dots\}$ (the propositional predicates), $\text{FREL} = \{R_1, R_2, \dots\}$ (the relational predicates) and $\text{FCONST} = \{c_1, c_2, \dots\}$ (the constants). The target first-order signature is defined to be $\mathcal{T} = \langle \text{FPROP} \cup \text{FREL}, \text{FCONST}, \emptyset \rangle$ with equality.

In this thesis we will (re)prove the characterization and definability theorem for a small family of hybrid logics. We will only consider the cases which extend BML with nominals and possibly the $@$ operator. There exist other hybrid important logics such as, for example, the ones which include the downarrow binder \downarrow . Results for these logics are nicely developed in [BvBW06, Chapter 12].

Definition 5.2.2 (Syntax). The syntax of the Hybrid Logic family over a given signature $\langle \text{PROP}, \text{REL}, \text{NOM} \rangle$ is defined as an extension of the propositional calculus with the following operators:

$$\varphi ::= \dots \mid i \mid @_i \varphi \mid \langle r \rangle \varphi$$

where $i \in \text{NOM}$ and $r \in \text{REL}$. We define the dual of $\langle r \rangle$ in the usual way.

Definition 5.2.3 (Models). A *hybrid model* for the source language is a tuple

$$\mathcal{M} = \langle W, (R_r)_{r \in \text{REL}}, V, G \rangle$$

which satisfies

- (i) W is a nonempty set,
- (ii) $R_r \subseteq W \times W$ are binary relations over W ,
- (iii) $V : \text{PROP} \rightarrow \mathcal{P}(W)$ is a valuation function, and
- (iv) $G : \text{NOM} \rightarrow W$ is an assignment for the nominals.²

An \mathfrak{F} -model for the target language is a tuple

$$\mathcal{M}^t = \langle W^t, (R_r^t)_{r \in \text{FREL}}, (P_i^t)_{i \in \text{FPROP}}, (c_i^t)_{i \in \text{FCONST}} \rangle$$

which satisfies

- (i) W^t is a nonempty set,
- (ii) $R_r^t \subseteq W^t \times W^t$ are binary relations over W^t ,
- (iii) $(P_i^t)_{i \in \text{FPROP}}$ are unary relations over W^t , and
- (iv) c_i^t are constants.

Definition 5.2.4 (Semantics). Given a model $\mathcal{M} = \langle W, R, V, G \rangle$ and $w \in W$, we extend the semantics presented in Definition 1.2.4 (BML semantics) with the following rules:

$$\begin{aligned} \mathcal{M}, w \models i & \quad \text{iff} \quad w = i \\ \mathcal{M}, w \models @_i \varphi & \quad \text{iff} \quad \mathcal{M}, G(i) \models \varphi \end{aligned}$$

Observe that the satisfaction definition for the nominals acts as an identity checker and the @-operator lets us ‘jump’ to an identified world.

Definition 5.2.5 (Formula translation). A formula translation that meets our requirements is given in [BdRV01, BvBW06]. We will not give the explicit definition because we will not need to use it.

Definition 5.2.6 (Model translation). Let \mathbf{K} be the class of all first order models over the target signature \mathcal{T} . Let $\mathcal{M} = \langle W, (R_r)_{r \in \text{REL}}, V, G \rangle$ and

$$\mathcal{M}^t = \langle W^t, (R_r^t)_{r \in \text{FREL}}, (P_i^t)_{i \in \text{FPROP}}, (c_i^t)_{i \in \text{FCONST}} \rangle.$$

Define the model translation $\mathbb{T}_{\mathbf{K}}(\mathcal{M}, w) = \langle \mathcal{M}^t, g^t \rangle$ to be the function induced by the following equations

$$\begin{aligned} W^t &= W \\ P_i^t &= V(p_i) \text{ for } p_i \in \text{PROP} \\ c_i^t &= G(i) \text{ for } i \in \text{NOM} \\ R_i^t &= R_i \\ g^t(x) &= w \end{aligned}$$

As with memory logics, the small family of hybrid logics that we will analyze also allows a modular simulation definition. Let \sim be a binary relation between hybrid pointed models. A simulation for a hybrid logic \mathfrak{L} can be defined imposing restrictions to \sim depending on the operators that \mathfrak{L} has. It is important to stress that every hybrid logic *should* have nominals. In the following table we summarize the restrictions associated with each operator. We write G_1 for the nominal assignment of \mathcal{M} and G_2 for the nominal assignment of \mathcal{N} .

² In the literature one can find an equivalent definition where $V : \text{PROP} \cup \text{NOM} \rightarrow \mathcal{P}(W)$ and G doesn’t exist. In this case we should add $V(i) = \{i\}$ for all $i \in \text{NOM}$ as a restriction. It is easy to see that this two definitions are equivalent.

<i>nominals</i>	(nagree)	If $\langle \mathcal{M}, m \rangle \sim \langle \mathcal{N}, n \rangle$, then $G_1(i) = m$ if and only if $G_2(i) = n$ for all $i \in \text{NOM}$.
@	(nom)	If $G_1(i) = w$ and $G_2(i) = v$ for some $i \in \text{NOM}$ then $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$.

Fig. 5.2: Operator restrictions for a modular hybrid simulation definition.

Definition 5.2.7 (Hybrid simulation). From now on, given a hybrid logic \mathcal{L} , we will refer as ‘the simulation for \mathcal{L} ’ to the simulation defined by the sum of the necessary conditions of Figure 5.2 for the operators in \mathcal{L} and the (*nontriv*) and (*agree*).

Theorem 5.2.8. Let \mathcal{L} be a hybrid logic defined as in Definition 5.2.2, let $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$ be two hybrid models. If $\langle \mathcal{M}, w \rangle \simeq_{\mathcal{L}} \langle \mathcal{N}, v \rangle$ then $\langle \mathcal{M}, w \rangle \equiv_{\mathcal{L}} \langle \mathcal{N}, v \rangle$.

Proof. The proof of this theorem can be found in [BvBW06]. \square

In the following theorem we will prove that the ω -saturated class of models has the Hennessy-Milner property with respect to the simulations for the following hybrid logics.

$$\begin{aligned} \mathcal{HL} &= \{\text{nominals}, \langle r \rangle\} \\ \mathcal{HL}(@) &= \{\text{nominals}, \langle r \rangle, @\} \end{aligned}$$

We will achieve this by showing that, given two equivalent pointed hybrid models, we can construct a simulation between them.

Theorem 5.2.9. Let \mathcal{L} be a hybrid logic as in Definition 5.2.2, the class of ω -saturated models has the Hennessy-Milner property with respect to \mathcal{L} -simulations.

Proof. Given two ω -saturated models \mathcal{M}, \mathcal{N} it suffices to give an \mathcal{L} -simulation between them. We propose the binary relation \sim defined as

$$\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle \text{ iff } \mathcal{M}, w \equiv_{\mathcal{L}} \mathcal{N}, v$$

Suppose that $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$. The proof for the (*nontriv*), (*agree*), (*forth*) and (*back*) restrictions are the same as for memory logics (see Theorem 5.1.13). We prove the restrictions for nominals and the @ operator.

Restrictions for nominals. This proof goes through using the satisfaction definition for the nominals. Remember that the nominals can only be true in *one* world.

$$G^{\mathcal{M}}(i) = w \iff \mathcal{M}, w \models i \iff \mathcal{N}, v \models i \iff G^{\mathcal{N}}(i) = v$$

the first and last implications are because of the semantics of nominal satisfaction and the implication in the middle is because of the \mathcal{L} -equivalence between $\langle \mathcal{M}, w \rangle$ and $\langle \mathcal{N}, v \rangle$.

Restrictions for @. Suppose that $G_1(i) = w$ and $G_2(i) = v$. As the relation is non-empty we can always get two equivalent worlds $a \in |\mathcal{M}|$ and $b \in |\mathcal{N}|$. Then we have

$$\mathcal{M}, a \models \varphi \text{ iff } \mathcal{N}, b \models \varphi$$

for all φ . Given an arbitrary formula ψ we can instantiate $\varphi = @_i \psi$ thus obtaining

$$\mathcal{M}, a \models @_i \psi \text{ iff } \mathcal{N}, b \models @_i \psi$$

which by semantic definition means that

$$\mathcal{M}, G_1(i) \models \psi \text{ iff } \mathcal{N}, G_2(i) \models \psi.$$

By hypothesis we can replace $G_1(i)$ and $G_2(i)$ and get

$$\mathcal{M}, w \models \psi \text{ iff } \mathcal{N}, v \models \psi$$

therefore $\mathcal{M}, w \equiv_{\mathfrak{L}} \mathcal{N}, v$ and by definition $\langle \mathcal{M}, w \rangle \sim \langle \mathcal{N}, v \rangle$.

After analyzing all the possible operators we have shown that, for every case, the \mathfrak{L} -simulation relation will satisfy the required constraints. This proves that given two ω -saturated \mathfrak{L} -equivalent models we are able to construct an \mathfrak{L} -simulation between them. Therefore, the ω -saturated class of models has the Hennessy-Milner property with respect to \mathfrak{L} -simulations. \square

Corollary 5.2.10. The definitions given above, along with Theorem 5.2.9, prove that the pair $\langle \mathfrak{L}, \mathbb{K} \rangle$ is an adequate pair for every hybrid logic \mathfrak{L} . Therefore, the Characterization and Definability theorems (4.2.3, 4.3.5 and 4.3.9) hold for this family of logics.

6. CONCLUSIONS AND FURTHER WORK

When developing a notion of simulation for a given logic \mathfrak{L} we need to be sure that we end up with the adequate notion. This means that it should *exactly* characterize model equivalence. If we prove that

$$\text{If } \langle \mathcal{M}, w \rangle \sim_{\mathfrak{L}} \langle \mathcal{N}, v \rangle \text{ then } \mathcal{M}, w \sqsubseteq_{\mathfrak{L}} \mathcal{N}, v \quad (6.1)$$

we have the half of the work done but the notion could still be wrong. Suppose that, for example, we say that the right notion of simulation for BML is the bisimulation notion of BML . It is obvious that we will be able to prove (6.1) but we are not working with the right notion of simulation: is is *too strong* for BML .

In the process of finding the right simulation notion, candidates are often checked ‘against’ finite or finitely branching models. In those cases, one expects to be able to prove the converse of (6.1). As we have seen, these classes of models are special cases of ω -saturated models. In the development of this thesis we arrive to the conclusion that if we can prove the converse of (6.1) for any ω -saturated model then we can, with little work, derive the Characterization and Definability theorems. This observation stresses the important relationship of ω -saturated models and the right simulation notion for a given logic.

When we defined the notion of adequate pair in Definition 4.1.1 we explained the strength of the Hennessy-Milner requirement. One would expect that a true generalization of the Characterization and Definability theorem doesn’t require the proof of a lemma. Instead, it should give a series of easily checkable properties that a logic should satisfy.

In sake of trying to give a result that could handle a broad spectrum of simulation notions we faced a big problem: we had no information regarding the structural properties of a simulation. In the applications chapter we saw that this information was *essential* to prove that ω -saturated models had the Hennessy-Milner property.

We think that an important line of work lies in the effort of trying to prove the Hennessy-Milner property without much information about the simulation notion. In future work we plan to integrate the results of this thesis with an approach similar to the one presented in [AG10], where coinductive model semantics are given.

Recall that in the beginning of Chapter 3 we presented two equivalent definitions of the ‘Basic Temporal Logic’. The classical one had custom made modalities F and P and the alternative view considered them as normal diamonds over a restricted class of models. The work done by Areces and Gorín in [AG10] generalizes this idea for (almost) any modality which can be defined with the pattern of the diamond operator ($\forall\exists$).

From our perspective, the most important point of their work is that, by restricting the class of models, we get a *unique* notion of model equivalence for every logic that fits in their framework. The right simulation notion turns to be the same as BML ’s bisimulation.

As far as we know, to the moment, there was no direct way to prove Characterization and Definability results using the framework developed in [AG10]. The problem laid in the restriction applied to the class of models, there is no classical proof which takes this kind of restrictions into account. With the results developed in this thesis it should be easy to prove a more general result using their framework.

Good as it is, the framework developed in [AG10] has its limits. Not every modality can be expressed with the pattern of a diamond. For example, there exists an extension of basic temporal logic which adds the Since and Until operators [BvBW06]. These operators don't respect the pattern of the definition of a normal diamond and therefore don't fit in the framework.

We think that one of the most exciting ways to continue this work is to try to expand the framework developed by Areces and Gorín to support more types of operators. This would allow us to give a 'canonical' simulation notion for a broader set of logics and therefore be able to prove the Hennessy-Milner property for them.

This line of work definitely looks as a promising path to give an automatic derivation of the Characterization and Definability theorems for a greater set of modal logics.

APPENDIX A: AUXILIARY RESULTS

Theorem A.1 (First order compactness relative to a class of models). Let \mathcal{C} be a class of first-order models which is closed under ultraproducts and let Σ be a set of first order formulas. If every finite set $\Delta \subseteq \Sigma$ has a model in \mathcal{C} then there is a model in \mathcal{C} for Σ .

Proof. Let \mathcal{M}_i^f be a model for each finite subset $\Delta_i \subseteq \Sigma$, algebraic proofs of the compactness theorem [Kei08, Theorem 4.3] show that the ultraproduct of the models $\mathcal{M} = \prod_U \mathcal{M}_i^f$ satisfies $\mathcal{M} \models \Sigma$.¹ As each \mathcal{M}_i^f is in \mathcal{C} and \mathcal{C} is closed under ultraproducts we conclude that $\mathcal{M} \in \mathcal{C}$. \square

Theorem A.2 (First order definability relative to a class of models). Let \mathcal{C} be a class of first-order models which is closed under ultraproducts and let $\mathcal{K} \subseteq \mathcal{C}$.

- (i) \mathcal{K} is a \mathcal{C} -elementary class (noted $\mathcal{C}\text{-}EC_\Delta$) if and only if \mathcal{K} is closed under ultraproducts, \mathcal{K} is \mathcal{C} -closed under potential isomorphisms and $\overline{\mathcal{K}} \cap \mathcal{C}$ is closed under ultrapowers.
- (ii) \mathcal{K} is a basic \mathcal{C} -elementary class (noted $\mathcal{C}\text{-}EC$) if and only if both \mathcal{K} and $\overline{\mathcal{K}} \cap \mathcal{C}$ are closed under ultraproducts and \mathcal{C} -closed under potential isomorphisms.

Proof. Left to right directions are left to the reader. Let's prove right-to-left directions.

- (i) Let \mathcal{C} be a class of first-order models which is closed under ultraproducts, let $\mathcal{K} \subseteq \mathcal{C}$ be such that \mathcal{K} is closed under ultraproducts, \mathcal{K} is \mathcal{C} -closed under potential isomorphisms and $\overline{\mathcal{K}} \cap \mathcal{C}$ is closed under ultrapowers.

Let $\Gamma = \{\varphi : \models_{\mathcal{K}} \varphi\}$, we show that Γ defines \mathcal{K} . For the easy part, take a model $\mathcal{M}^f \in \mathcal{K}$. By definition of Γ we have that $\mathcal{M}^f \models \Gamma$.

For the hard part, let $\mathcal{M}^f \in \mathcal{C}$ be such that $\mathcal{M}^f \models \Gamma$. Define the first order theory of the model \mathcal{M}^f as

$$\Sigma = \{\varphi : \varphi \text{ is a sentence and } \mathcal{M}^f \models \varphi\}.$$

Let's see that there is a model for Σ which lays in \mathcal{K} . Suppose not, by Theorem A.1 there is a finite subset $\Sigma_0 = \{\sigma_1, \dots, \sigma_n\}$ of Σ which is unsatisfiable in \mathcal{K} . Hence, $\models_{\mathcal{K}} \neg(\sigma_1 \wedge \dots \wedge \sigma_n)$ which means that $\neg(\sigma_1 \wedge \dots \wedge \sigma_n) \in \Gamma$. As $\mathcal{M}^f \models \Gamma$ we arrive to an absurd. We have proved that there exists $\mathcal{N}^f \in \mathcal{K}$ such that $\mathcal{N}^f \models \Sigma$.

By [CK90, Theorem 6.1.15], $\mathcal{M}^f \equiv_{\exists} \mathcal{N}^f$ if and only if there exist ultrapowers \mathcal{M}_*^f and \mathcal{N}_*^f such that $\mathcal{M}_*^f \cong \mathcal{N}_*^f$. Because \mathcal{K} is closed under ultraproducts, in particular it is closed under ultrapowers, therefore, $\mathcal{N}_*^f \in \mathcal{K}$. As both classes are closed under ultrapowers, \mathcal{M}^f and \mathcal{M}_*^f belong to the same class. Last but not least, as \mathcal{K} is \mathcal{C} -closed under potential isomorphisms and $\mathcal{M}_*^f \cong \mathcal{N}_*^f$ then $\mathcal{M}_*^f \in \mathcal{K}$. Finally we conclude that $\mathcal{M}^f \in \mathcal{K}$.

¹ With a suitable ultrafilter U .

- (ii) Let \mathbf{C} be a class of first-order models which is closed under ultraproducts, let $\mathbf{K} \subseteq \mathbf{C}$ be such that both \mathbf{K} and $\overline{\mathbf{K}} \cap \mathbf{C}$ are closed under ultraproducts and \mathbf{C} -closed under potential isomorphisms.

By (i) we know there exist two sets Γ, Γ_c respectively defining \mathbf{K} and $\overline{\mathbf{K}} \cap \mathbf{C}$. Observe that the union $\Gamma \cup \Gamma_c$ is not satisfiable in \mathbf{C} . By Theorem A.1 there exists a finite subset $\Sigma_0 \subseteq \Gamma \cup \Gamma_c$ which is unsatisfiable in \mathbf{C} . Call $\Sigma_0 = \{\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m\}$ with $\alpha_i \in \Gamma$ and $\beta_j \in \Gamma_c$. As Σ_0 is unsatisfiable in \mathbf{C} this means that $\models_{\mathbf{C}} \alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \neg(\beta_1 \wedge \dots \wedge \beta_m)$. Let's see that it is exactly $\varphi = \alpha_1 \wedge \dots \wedge \alpha_n$ that defines \mathbf{K} .

Let $\mathcal{M}^f \in \mathbf{C}$. If $\mathcal{M}^f \in \mathbf{K}$ then trivially $\mathcal{M}^f \models \varphi$. Suppose $\mathcal{M}^f \models \varphi$ then $\mathcal{M}^f \not\models \beta_1 \wedge \dots \wedge \beta_m$ therefore $\mathcal{M}^f \not\models \Gamma_c$ hence $\mathcal{M}^f \notin \overline{\mathbf{K}} \cap \mathbf{C}$. We conclude that $\mathcal{M}^f \in \mathbf{K}$. \square

Theorem A.3. Let $\langle \mathfrak{L}, \mathbf{K} \rangle$ be an adequate pair and let $M_1, M_2 \subseteq \text{MODS}(\mathfrak{L})$ be two classes such that $\mathsf{T}(M_1)$ and $\mathsf{T}(M_2)$ are closed under ultrapowers. Let $\mathcal{M} \in M_1$ and $\mathcal{N} \in M_2$ be two \mathfrak{L} -models such that for some $w \in |\mathcal{M}|$, $v \in |\mathcal{N}|$ they satisfy $\mathcal{N}, v \sqsubseteq_{\mathfrak{L}} \mathcal{M}, w$ then there exist models $\mathcal{M}^* \in M_1$ and $\mathcal{N}^* \in M_2$ and elements $w^* \in |\mathcal{M}^*|$, $v^* \in |\mathcal{N}^*|$ such that

1. $\mathsf{T}(\mathcal{M}, w) \equiv_{\mathfrak{F}} \mathsf{T}(\mathcal{M}^*, w^*)$ and $\mathsf{T}(\mathcal{N}, v) \equiv_{\mathfrak{F}} \mathsf{T}(\mathcal{N}^*, v^*)$
Their translations are pairwise elementary equivalent.
2. $\mathcal{M}, w \equiv_{\mathfrak{L}} \mathcal{M}^*, w^*$ and $\mathcal{N}, v \equiv_{\mathfrak{L}} \mathcal{N}^*, v^*$
They are pairwise equivalent.
3. $\mathcal{N}^*, v^* \rightrightarrows_{\mathfrak{L}} \mathcal{M}^*, w^*$
There is a simulation from \mathcal{N}^*, v^* to \mathcal{M}^*, w^* .

Proof. We define some names for the models which we will be working on before starting with the proof. Call $\mathcal{M}_f, g_w = \mathsf{T}(\mathcal{M}, w)$ and $\mathcal{N}_f, g_v = \mathsf{T}(\mathcal{N}, v)$. Take $\mathcal{M}_f^+, \mathcal{N}_f^+$ to be ω -saturated ultrapowers of \mathcal{M}_f and \mathcal{N}_f (their existence is proved in Theorem B.7). As the classes are closed under ultrapowers the saturated models are in the same class as their originators.

By [CK90, Corollary 4.1.13] we have an elementary embedding $d : |\mathcal{M}_f| \rightarrow |\mathcal{M}_f^+|$. Let g_w^+ be an assignment for \mathcal{M}_f^+ such that $g_w^+(x) = d(g_w(x))$. Take the modal preimage of \mathcal{M}_f^+, g_w^+ and call it $\mathcal{M}^*, w^* = \mathsf{T}^{-1}(\mathcal{M}_f^+, g_w^+)$. We make the same process and assign similar names to models and points deriving from \mathcal{N} .

1. As a corollary of [CK90, Corollary 4.1.13], as there is an elementary embedding, we have that $\mathcal{M}_f, g_w \equiv_{\mathfrak{F}} \mathcal{M}_f^+, g_w^+$. The same proof works with \mathcal{N}_f and \mathcal{N}_f^+ .
2. Following the last point, we can conclude, through the translations' truth-preservation, that $\mathcal{M}, w \equiv_{\mathfrak{L}} \mathcal{M}^*, w^*$. The same proof works with \mathcal{N}, v and \mathcal{N}^*, v^* . Corollary: $\mathcal{N}^*, v^* \sqsubseteq_{\mathfrak{L}} \mathcal{M}^*, w^*$.
3. As both \mathcal{M}_f^+ and \mathcal{N}_f^+ are ω -saturated, by definition of adequate pair, that implies that they have the Hennessy-Milner property. Therefore, because we've just proved that $\mathcal{N}^*, v^* \sqsubseteq_{\mathfrak{L}} \mathcal{M}^*, w^*$ we can conclude that $\mathcal{N}^*, v^* \rightrightarrows_{\mathfrak{L}} \mathcal{M}^*, w^*$. \square

APPENDIX B: FILTERS AND ULTRAPRODUCTS

The ultraproduct construction is a uniform method of building models of first order theories which has applications in many areas of mathematics. It is attractive because it is algebraic in nature, but preserves all properties expressible in first order logic. In this section we will make a brief summary of the tools presented in [Kei08] with some additional notes that may guide the reader. Unless explicitly stated the proofs can be checked in the publication that we've just mentioned.

Definition B.1 (Filter, proper filter, ultrafilter). Let I be a non-empty set.

A *filter* U over I is a set of subsets of I such that:

- (i) $I \in U$.
- (ii) U is closed under supersets; if $X \in U$ and $X \subseteq Y$ then $Y \in U$.
- (iii) U is closed under finite intersections; if $X \in U$ and $Y \in U$ then $X \cap Y \in U$.

A *proper filter* over I is a filter U over I such that:

- (iv) $\emptyset \notin U$.

An *ultrafilter* over I is a proper filter U over I such that:

- (v) For each $X \subseteq I$ exactly one of $X, I \setminus X$ belongs to U .

Theorem B.2 (Ultrafilter Theorem, Tarski). Every proper filter over the set I can be extended to an ultrafilter over I .

We first define the ultraproduct operation on *sets*. Let U be an ultrafilter over I , and for each $i \in I$ let A_i be a non-empty set. The ultraproduct $\prod_U A_i$ is obtained by first taking the cartesian product $C = \prod_{i \in I} A_i$. Observe that C is the set of all functions f such that for each $i \in I$, $f(i) \in A_i$. We continue by identifying elements which are equal for U -almost all $i \in I$. The formal definition is as follows.

Definition B.3 (U -equivalence). Let U be an ultrafilter over I . Two elements f, g of the cartesian product $\prod_{i \in I} A_i$ are said to be U -equivalent, noted $f =_U g$, if the set $\{i : f(i) = g(i)\}$ belongs to U . The U -equivalence class of f is the set $f_U = \{g : f =_U g\}$.

Definition B.4 (Ultraproduct of sets). Let U be an ultrafilter over I , the *ultraproduct of sets* $\prod_U A_i$ is defined as the set of U -equivalence classes

$$\prod_U A_i = \{f_U : f \in \prod_{i \in I} A_i\}.$$

An *ultrapower of sets* of A modulo U is defined as the ultraproduct $\prod_U A = \prod_U A_i$ where $A_i = A$ for each $i \in I$. The *natural* or *canonical embedding* is the mapping $d : A \rightarrow \prod_U A_i$ such that $d(a)$ is the U -equivalence class of the constant function with value a . That is $d(a) = (\lambda x. a)_U$.

We now introduce the ultraproduct operation on first order structures. For each $i \in I$, let \mathcal{M}_i^f be a first order model with universe set A_i . Briefly, the ultraproduct of models $\prod_U \mathcal{M}_i^f$ is the unique first order model with universe $\prod_U A_i$ such that each *basic* formula holds in the ultraproduct if and only if it holds in $\prod_U \mathcal{M}_i^f$ for U -almost all i . Here is the formal definition.

Definition B.5 (Ultraproduct of models). Let U be an ultrafilter over I , and let \mathcal{M}_i^f be a family of \mathcal{L} -structures with universe set A_i . The *ultraproduct of models* $\prod_U \mathcal{M}_i^f$ is the unique model \mathcal{M}^f such that:²

- (i) The universe of \mathcal{M}^f is $M = \prod_U A_i$.
- (ii) For each atomic formula $\varphi(x_1, \dots, x_k)$ which has at most one symbol from the vocabulary \mathcal{L} , and each $f_1, \dots, f_k \in \prod_{i \in I} A_i$,

$$\mathcal{M}^f \models \varphi(f_{1U}, \dots, f_{kU}) \text{ iff } \{i : \mathcal{M}_i^f \models \varphi(f_1(i), \dots, f_k(i))\} \in U.$$

Using the properties of ultrafilters one can check that such structure is unique and thus well-defined. Similarly, the *ultrapower of models* of the model \mathcal{M}^f modulo U is defined as the ultraproduct $\prod_U \mathcal{M}^f = \prod_U \mathcal{M}_i^f$ where $\mathcal{M}_i^f = \mathcal{M}^f$ for each $i \in I$.

Finally we present the theorem of Łoś which makes ultraproducts useful in model theory. It shows that each formula holds in the ultraproduct if and only if it holds in $\prod_U \mathcal{M}_i^f$ for U -almost all i . Observe that in this case there is no restriction to basic formulas as before in Definition B.5.

Theorem B.6 (Fundamental theorem of Ultraproducts, Łoś). Let U be an ultrafilter over I , and let \mathcal{M}_i^f be a family of \mathcal{L} -structures for each $i \in I$. For each formula $\varphi(x_1, \dots, x_k)$, and each $f_1, \dots, f_k \in \prod_{i \in I} A_i$, we have

$$\prod_U \mathcal{M}_i^f \models \varphi(f_{1U}, \dots, f_{kU}) \text{ iff } \{i : \mathcal{M}_i^f \models \varphi(f_1(i), \dots, f_k(i))\} \in U.$$

Corollary B.1. For each set of first-order sentences Γ , and family of models \mathcal{M}_i^f . If $\mathcal{M}_i^f \models \Gamma$ for all $i \in I$ then $\prod_U \mathcal{M}_i^f \models \Gamma$.

Corollary B.2 ([BdRV01, Corollary A.21]). Let $\prod_U \mathcal{M}^f$ be an ultrapower of \mathcal{M}^f , the diagonal mapping $d(a) = (\lambda x.a)_U$ is an elementary embedding. That is, for any first order formula $\varphi(x_1, \dots, x_k)$ and $a_1, \dots, a_k \in \mathcal{M}^f$

$$\mathcal{M}^f \models \varphi(a_1, \dots, a_k) \text{ if and only if } \prod_U \mathcal{M}^f \models \varphi(d(a_1), \dots, d(a_k)).$$

Using this results we can state an important theorem, the existence of elementary equivalent ω -saturated ultrapowers.

Theorem B.7. Let \aleph be countable and \mathcal{M}^f be an \aleph model then there exists an ω -saturated ultrapower $\prod_U \mathcal{M}^f$ such that $\mathcal{M}^f \equiv_{\aleph} \prod_U \mathcal{M}^f$.

Proof. Follows from a direct combination of [Kei08, Theorem 5.6] and Corollary B.2. \square

² The definition found in [Kei08] has a mistake in point (ii). The subscript of \mathcal{A} is missing in the set definition after the ‘iff’, it should be \mathcal{A}_i .

BIBLIOGRAPHY

- [AFFM08] C. Areces, D. Figueira, S. Figueira, and S. Mera. Expressive power and decidability for memory logics. In *Logic, Language, Information and Computation*, volume 5110 of *Lecture Notes in Computer Science*, pages 56–68, Edinburgh, Scotland, 2008. Springer Berlin / Heidelberg. Proceedings of WoLLIC 2008.
- [AG10] Carlos Areces and Daniel Gorín. Coinductive models and normal forms for modal logics. *Logic Journal of the IGPL (to appear)*, 2010.
- [AKS08] C. Areces, A. Koller, and K. Striegnitz. Referring expressions as formulas of description logic. In *Proceedings of the 5th International Natural Language Generation*, Salt Fork, OH, USA, 2008.
- [Bar85] Henk Barendregt. *The Lambda Calculus: its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North Holland, second edition, 1985.
- [BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [BvBW06] P. Blackburn, J. van Benthem, and F. Wolter. *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., New York, NY, USA, 2006.
- [CK90] C. C. Chang and H. J. Keisler. *Model Theory, Volume 73 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science B.V., Amsterdam, The Netherlands, 1990.
- [COP00] E. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 2000.
- [EFT84] H. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer-Verlag, 1984.
- [FH83] M. Forti and F. Honsell. Set theory with free construction principles. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, 10(3):493–522, 1983.
- [Fin74] K. Fine. An incomplete logic containing S4. *Theoria*, 40(1):23–29, 1974.
- [FL79] M.J. Fischer and R.E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.
- [GO05] V. Goranko and M. Otto. Model theory of modal logic. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of modal logic*, pages 249–329. Elsevier, 2005.
- [Gri73] D. Gries. Describing an algorithm by Hopcroft. *Acta Informatica*, 2:97–109, 1973.
- [Hol98] Marco Hollenberg. *Logic and Bisimulation*. PhD thesis, Philosophical Institute, Utrecht University, 1998.
- [Hop71] J. Hopcroft. An $n \log(n)$ algorithm for minimizing states in a finite automaton. In Z. Kohave, editor, *Theory of Machines and Computations*. Academic Press, 1971.
- [JT51] B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1951.
- [JT52] B. Jónsson and A. Tarski. Boolean algebras with operators, Part II. *American Journal of Mathematics*, 74:127–162, 1952.
- [KdR97] N. Kurtonina and M. de Rijke. Simulating without negation. *Journal of Logic and Computation*, 7:503–524, 1997.

- [KdR99] N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artif. Intell.*, 107(2):303–333, 1999.
- [Kei08] H. J. Keisler. The ultraproduct construction. In *Proceedings of the Ultramath Conference*, Pisa, Italy, 2008.
- [KR97] Natasha Kurtonina and Maarten De Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.
- [Kri63a] S.A. Kripke. Semantical analysis of modal logic I. Normal modal propositional calculi. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 9, 1963.
- [Kri63b] S.A. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.
- [Lad77] R.E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM journal on computing*, 6:467, 1977.
- [Lew18] C.I. Lewis. *A Survey of Symbolic Logic*. University of California Press, 1918.
- [Mer09] Sergio Mera. *Modal Memory Logics*. PhD thesis, Universidad de Buenos Aires & Université Henri Poincaré, Buenos Aires, Argentina, 2009.
- [Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.
- [Par81] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, volume 104 of *LNCS*, pages 167–183, London, UK, 1981. Springer-Verlag.
- [Pra79] V.R. Pratt. Models of program logics. In *20th Annual Symposium on Foundations of Computer Science*, pages 115–122, 1979.
- [Pri57] A.N. Prior. *Time and modality*. Oxford University Press, 1957.
- [San09] D. Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4):1–41, 2009.
- [Tho72] S.K. Thomason. Semantic analysis of tense logics. *Journal of Symbolic Logic*, pages 150–158, 1972.
- [Tho74] S.K. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40(1):30–34, 1974.
- [vB76] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Universiteit van Amsterdam, Instituut voor Logica en Grondslagenonderzoek van Exacte Wetenschappen, 1976.
- [vB84] J. van Benthem. Modal correspondence theory. *Handbook of Philosophical Logic*, 2:167–247, 1984.