



HAL
open science

On the Universality of Atomic and Molecular Logics via Protologics

Guillaume Aucher

► **To cite this version:**

Guillaume Aucher. On the Universality of Atomic and Molecular Logics via Protologics. Logica Universalis, 2022, 16 (1-2), pp.285-322. 10.1007/s11787-022-00298-5 . hal-03911802

HAL Id: hal-03911802

<https://hal.science/hal-03911802>

Submitted on 23 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Universality of Atomic and Molecular Logics via Protologics

Guillaume Aucher

Abstract. After observing that the truth conditions of connectives of non-classical logics are generally defined in terms of formulas of first-order logic, we introduce ‘protologics’, a class of logics whose connectives are defined by arbitrary first-order formulas. Then, we introduce atomic and molecular logics, which are two subclasses of protologics that generalize our gaggle logics and which behave particularly well from a theoretical point of view. We also study and introduce a notion of equi-expressivity between two logics based on different classes of models. We prove that, according to that notion, every pure predicate logic with $k \geq 0$ free variables and constants is as expressive as a predicate atomic logic, some sort of atomic logic. Then, we prove that the class of protologics is equally expressive as the class of molecular logics. That formally supports our claim that atomic and molecular logics are somehow ‘universal’. Finally, we identify a subclass of molecular logics that we call predicate molecular logics and which constitutes its representative core: every molecular logic is as expressive as a predicate molecular logic.

Mathematics Subject Classification (2010). 03, 03B, 03C.

Keywords. Universal logic, expressivity, first-order logics, non-classical logics.

1. Introduction

A wide variety of non-classical logics have been introduced over the past decades, such as relevant logics, linear logics and Lambek calculi, to name just a few. On the one hand, this diversity is an asset since each logic is relevant to a specific purpose, and one can select, and resort to, some of them when reasoning about a given applicative issue [32]. On the other hand, and from a theoretical point of view, this plurality can be felt as problematic because it threatens the unity and the unifying power of logic. Even if all logics

already have in common the same terminology and notions, non-classical logics are still disorganized and scattered and lack a common formal ground. In response to that situation, a number of attempts have been made by various logicians to introduce a genuine unity to logic as witnessed, for example, by the development of abstract model theory and “institutions” [7, 30, 20], the “labelled deductive systems” of Gabbay [18] or the “basic logic” of Sambin & al. [40] (see [8, 17] for details and more examples). This led to the rise of a research thread sometimes referred to (nowadays) as “Universal Logic” [8]. According to Béziau, “Universal logic is a general theory of logical systems, studying the relations between them, comparing them and combining them” [9].

We share the ideal and the objectives of “Universal Logic”. In this article, we propose a novel and simple approach to address that grand problem. Our starting point will be the observation that the truth conditions of non-classical connectives are often defined in terms of first-order formulas (without functions). This will lead us to introduce protologics, a class of logics whose connectives are defined by arbitrary first-order formulas. Even if protologics are quite general and capture a wide range of non-classical logics, they do not lend themselves naturally to a systematic exploration of their theoretical properties. The truth conditions of their connectives are arbitrary formulas of first-order logic and, as such, do not yield us much information about their associated theoretical properties. To overcome this difficulty we will introduce atomic and molecular logics, a subclass of protologics whose connectives are of a specific form and which are particularly well-behaved from a theoretical point of view. They are a generalization of polyadic modal logics that take into account some sort of monotonicity in the truth conditions of their connectives. They generalize our gaggle logics [4, 5]. We will show that atomic and molecular logics are ‘paradigmatic’ logics or ‘normal form’ logics in the sense that they can capture a very wide range of non-classical logics. We will prove in this article that first-order logic with $k \geq 0$ free variables and constants is as expressive as a specific atomic logic and that every protologic is as expressive as a molecular logic. That result supports our claim that atomic and molecular logics are (somehow) ‘universal’.

Structure of the article. We start in Section 2 by recalling first-order logics and some of the most well-known non-classical logics: that is, modal logic, the Lambek calculus, modal intuitionistic logic, temporal logic and many-valued logics. In Section 3, we also study and introduce a notion of equi-expressivity between logics based on different classes of models. Then, in Section 4, we introduce protologics: that is, logics whose connectives are defined by arbitrary first-order formulas, as well as atomic and molecular logics. We show that modal logic, many-valued logics and the Lambek calculus are atomic logics whereas modal intuitionistic logic and temporal logics are molecular logics. In Section 5, we prove that first-order logic is as expressive as an atomic logic and that the class of molecular logics is as expressive as the class of protologics. In Section 6, we introduce predicate molecular logics and we show

that the class of molecular logics is as expressive as the class of predicate molecular logics. We conclude in Section 7.

2. Classical and Non-Classical Logics

In this section, we recall first-order logic (FOL) and some of the most well-known non-classical logics. Logics will always be semantically presented by following a tri-partite representation: language, class of models, satisfaction relation.

2.1. Classical Logics

We deal with FOL without function symbols. The set $\mathcal{P} \triangleq \{R_1, \dots, R_n, \dots\}$ is a set of *predicate symbols* of arity k_1, \dots, k_n, \dots respectively (one of them can be the identity predicate = of arity 2), $\mathcal{V} \triangleq \{v_1, \dots, v_n, \dots\}$ is a set of *variables* and $\mathcal{C} \triangleq \{c_1, \dots, c_n, \dots\}$ is a set of *constants*. Each of these sets can be finite or infinite. v_1, v_2, v_3, \dots are the names of the variables and we use the expressions $x, x_1, x_2, \dots, y, y_1, y_2, \dots, z, z_1, z_2, \dots$ to refer to arbitrary variables or constants, which can be for example $v_{42}, v_5, c_{101}, c_{21}, \dots$.

The *first-order language* $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ is defined inductively by the following grammars in BNF:

$$\begin{aligned} \mathcal{L}_{\text{FOL}}^{\mathcal{V}\mathcal{C}} : t ::= x \mid c \\ \mathcal{L}_{\text{FOL}}^{\mathcal{P}} : \varphi ::= R t \dots t \mid \perp \mid (\varphi \rightarrow \varphi) \mid \forall x \varphi \end{aligned}$$

where $x \in \mathcal{V}$, $c \in \mathcal{C}$, $t \in \mathcal{L}_{\text{FOL}}^{\mathcal{V}\mathcal{C}}$ and $R \in \mathcal{P}$. Elements of $\mathcal{L}_{\text{FOL}}^{\mathcal{V}\mathcal{C}}$ are called *terms* and elements of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ are called *first-order formulas*. Formulas of the form $R t_1 \dots t_k$ are called *atomic formulas*. If $\varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$, the *Boolean negation* of φ , denoted $\neg\varphi$, is defined by the abbreviation $\neg\varphi \triangleq (\varphi \rightarrow \perp)$. We also use the abbreviations $\top \triangleq \neg\perp$, $(\varphi \vee \psi) \triangleq (\neg\varphi \rightarrow \psi)$, $(\varphi \wedge \psi) \triangleq \neg(\neg\varphi \vee \neg\psi)$ and $(\varphi \leftrightarrow \psi) \triangleq (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ as well as the abbreviations $\exists x \varphi \triangleq \neg\forall x \neg\varphi$, $\forall x_1 \dots x_n \varphi \triangleq \forall x_1 \dots \forall x_n \varphi$, $\exists x_1 \dots x_n \varphi \triangleq \exists x_1 \dots \exists x_n \varphi$ and $\forall \bar{x} \varphi \triangleq \forall x_1 \dots x_n \varphi$ if $\bar{x} = (x_1, \dots, x_n)$ is a tuple of variables.

Let $\varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$. An occurrence of a variable x in φ is *free* (in φ) if, and only if, x is not within the scope of a quantifier of φ . We say that a formula of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ is a *sentence* (or is *closed*) when it contains no free variable. We denote by $\varphi(x_1, \dots, x_k)$ a formula of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ whose free variables or constants coincide *exactly* with x_1, \dots, x_k . We assume that these variables and constants are all distinct. In doing so, we depart from the literature in which this notation means that the free variables of φ are *included* in $\{x_1, \dots, x_k\}$. Free variables may be used to bind elements of two different subformulas. For example, the formula $Ryx \vee R'xz$ with free variables x, y, z will be evaluated in a structure in such a way that x will be assigned the same element of the domain in the two subformulas Ryx and $R'xz$.

We denote by $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x})$ the fragment of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ whose formulas all contain at least one free variable or constant and by $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\emptyset)$ the set of sentences of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$. For all $k \in \mathbb{N}^*$ and $\bar{x} = (x_1, \dots, x_k) \in \mathcal{V}^k$, we denote by $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)$

the fragment of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ whose formulas all contain *exactly* k free variables or constants and these variables or constants are \bar{x} . Note that $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x})$ can be partitionned into sublanguages of the form $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)$.

A *structure* is a tuple $M \triangleq (W, \{R_1, \dots, R_n, \dots, c_1, \dots, c_n, \dots\})$ where:

- W is a non-empty set called the *domain*;
- R_1, \dots, R_n, \dots are relations over W with the same arity as R_1, \dots, R_n, \dots respectively;
- $c_1, \dots, c_n, \dots \in W$ are elements of the domain called *distinguished elements*.

An *assignment* over M is a mapping $s : \mathcal{V} \cup \mathcal{C} \rightarrow W$ such that for all $c_i \in \mathcal{C}$, $s(c_i) = c_i$. If s is an assignment, $s[x := w]$ is the same assignment as s except that the value of the variable $x \in \mathcal{V}$ is assigned to w . A pair of structure and assignment (M, s) is called a *pointed structure*. The class of all pointed structures (M, s) is denoted \mathcal{M}_{FOL} .

The *satisfaction relation* $\models_{\text{FOL}} \subseteq \mathcal{M}_{\text{FOL}} \times \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ is defined inductively as follows. Below, we write $(M, s) \models \varphi$ for $((M, s), \varphi) \in \models_{\text{FOL}}$.

$$\begin{aligned} (M, s) \models \perp & & \text{never;} \\ (M, s) \models R_i t_1 \dots t_k & \text{ iff } & (s(t_1), \dots, s(t_k)) \in R_i; \\ (M, s) \models (\varphi \rightarrow \psi) & \text{ iff } & \text{if } (M, s) \models \varphi \text{ then } (M, s) \models \psi; \\ (M, s) \models \forall x \varphi & \text{ iff } & (M, s[x := w]) \models \varphi \text{ for all } w \in W. \end{aligned}$$

In the literature [11], $(M, s) \models \varphi(x_1, \dots, x_k)$ is sometimes denoted $M \models \varphi(x_1, \dots, x_k)[w_1, \dots, w_k]$, $M \models \varphi[w_1/x_1, \dots, w_k/x_k]$ or simply $M \models \varphi[w_1, \dots, w_k]$, with $w_1 = s(x_1), \dots, w_k = s(x_k)$. Some other times [16], it is denoted $M \models \varphi(x_1, \dots, x_n)[s]$, $M, s \models \varphi(x_1, \dots, x_n)$ or simply $M \models \varphi[s]$. In that case, we say that (M, s) makes φ *true*. We say that the formula $\varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ is *realized in* M when there is an assignment s such that $(M, s) \models \varphi$.

We depart from the literature by treating constants on a par with variables: the denotation of constants is usually not dealt with by means of assignments. In doing so, we can avoid to introduce distinguished elements in the very definition of a structure. Two (pointed) structures are *elementarily equivalent* when they make true the same sentences, and for all $k \in \mathbb{N}^*$ and all $\bar{x} = (x_1, \dots, x_k) \in \mathcal{V}^k$ we write $(M, s) \equiv_{\bar{x}} (M', s')$ when for all $\varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)$ it holds that $(M, s) \models \varphi$ iff $(M', s') \models \varphi$.

If \mathcal{E}_{FOL} is a class of pointed structures, the triple $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}, \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ is called *pure predicate logic (associated to \mathcal{E}_{FOL})*, the triple $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}), \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ is called *pure predicate logic with free variables and constants (associated to \mathcal{E}_{FOL})* and the triple $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k), \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ is called *pure predicate logic with k free variables and constants (associated to \mathcal{E}_{FOL})*. When \mathcal{E}_{FOL} is \mathcal{M}_{FOL} , they are simply called respectively *pure predicate logic* and *pure predicate logic with (k) free variables and constants*.

2.2. Non-Classical Logics

The examples of non-classical logics that follow are among the most well-known and most studied non-classical logics.

In this section, \mathbb{A} is a set of *propositional letters* which can be finite or infinite.

2.2.1. Modal Logic. Let \mathbb{I} be a set of indices which can be finite or infinite. The *multi-modal language* \mathcal{L}_{ML} is defined inductively by the following grammar in BNF:

$$\mathcal{L}_{\text{ML}} : \varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \diamond_j \varphi \mid \square_j \varphi$$

where $p \in \mathbb{A}$ and $j \in \mathbb{I}$.

We present the so-called *possible world semantics* of modal logic. A *Kripke model* M is a tuple $M \triangleq (W, \{R_1, \dots, R_m, \dots, P_1, \dots, P_n, \dots\})$ where

- W is a non-empty set whose elements are called *possible worlds*;
- $R_1, \dots, R_m, \dots \subseteq W \times W$ are binary relations over W called *accessibility relations*;
- $P_1, \dots, P_n, \dots \subseteq W$ are unary relations interpreting the propositional letters of \mathbb{P} .

We write $w \in M$ for $w \in W$ by abuse and the pair (M, w) is called a *pointed Kripke model*. The class of all pointed Kripke models is denoted \mathcal{E}_{ML} .

We define the *satisfaction relation* $\models_{\text{ML}} \subseteq \mathcal{E}_{\text{ML}} \times \mathcal{L}_{\text{ML}}$ inductively by the following *truth conditions*. Below, we write $(M, w) \models \varphi$ for $((M, w), \varphi) \in \models_{\text{ML}}$. For all $(M, w) \in \mathcal{E}_{\text{ML}}$, all $\varphi, \psi \in \mathcal{L}_{\text{ML}}$, all $p_i \in \mathbb{P}$ and all $j \in \mathbb{I}$,

$$\begin{aligned} (M, w) \models p_i & \quad \text{iff } P_i(w) \text{ holds;} \\ (M, w) \models \neg p_i & \quad \text{iff } P_i(w) \text{ does not hold;} \\ (M, w) \models (\varphi \wedge \psi) & \quad \text{iff } (M, w) \models \varphi \text{ and } (M, w) \models \psi; \\ (M, w) \models (\varphi \vee \psi) & \quad \text{iff } (M, w) \models \varphi \text{ or } (M, w) \models \psi; \\ (M, w) \models \diamond_j \varphi & \quad \text{iff there exists } v \in W \text{ such that } R_j wv \text{ and } (M, v) \models \varphi; \\ (M, w) \models \square_j \varphi & \quad \text{iff for all } v \in W \text{ such that } R_j wv, (M, v) \models \varphi. \end{aligned}$$

The triple $(\mathcal{L}_{\text{ML}}, \mathcal{E}_{\text{ML}}, \models_{\text{ML}})$ forms a logic, that we call *modal logic*.

2.2.2. Lambek Calculus. The *Lambek language* \mathcal{L}_{LC} is the set of formulas defined inductively by the following grammar in BNF:

$$\mathcal{L}_{\text{LC}} : \varphi ::= p \mid (\varphi \circ \varphi) \mid (\varphi / \varphi) \mid (\varphi \backslash \varphi)$$

where $p \in \mathbb{P}$. A *Lambek model* is a tuple $M = (W, \{R, P_1, \dots, P_n, \dots\})$ where:

- W is a non-empty set;
- $R \subseteq W \times W \times W$ is a ternary relation over W ;
- $P_1, \dots, P_n, \dots \subseteq W$ are unary relations over W .

We write $w \in M$ for $w \in W$ by abuse and (M, w) is called a *pointed Lambek model*. The class of all pointed Lambek models is denoted \mathcal{E}_{LC} . We define the *satisfaction relation* $\models_{\text{Int}} \subseteq \mathcal{E}_{\text{LC}} \times \mathcal{L}_{\text{LC}}$ by the following *truth conditions*.

Below, we write $(M, w) \models \varphi$ for $((M, w), \varphi) \in \models_{\text{LC}}$. For all Lambek models $M = (W, \{R, P_1, \dots, P_n, \dots\})$, all $w \in M$, all $\varphi, \psi \in \mathcal{L}_{\text{LC}}$ and all $p_i \in \mathbb{P}$,

$$\begin{aligned}
(M, w) \models p_i & \quad \text{iff} \quad P_i(w) \text{ holds;} \\
(M, w) \models (\varphi \circ \psi) & \quad \text{iff} \quad \text{there are } v, u \in W \text{ such that } Rvwu, \\
& \quad (M, v) \models \varphi \text{ and } (M, u) \models \psi; \\
(M, w) \models (\varphi \setminus \psi) & \quad \text{iff} \quad \text{for all } v, u \in W \text{ such that } Rvwu, \\
& \quad \text{if } (M, v) \models \varphi \text{ then } (M, u) \models \psi; \\
(M, w) \models (\psi / \varphi) & \quad \text{iff} \quad \text{for all } v, u \in W \text{ such that } Rvwu, \\
& \quad \text{if } (M, v) \models \varphi \text{ then } (M, u) \models \psi.
\end{aligned}$$

The triple $(\mathcal{L}_{\text{LC}}, \mathcal{E}_{\text{LC}}, \models_{\text{LC}})$ forms a logic, that we call the *Lambek calculus*.

2.2.3. Modal Intuitionistic Logic. The *modal intuitionistic language* \mathcal{L}_{Int} is defined inductively by the following grammar in BNF:

$$\mathcal{L}_{\text{Int}} : \varphi ::= \top \mid \perp \mid p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \Rightarrow \varphi) \mid \diamond \varphi \mid \square \varphi$$

where $p \in \mathbb{A}$. A *modal intuitionistic model* is a tuple $M = (W, \{R, R_\diamond, P_1, \dots, P_n, \dots\})$ where:

- W is a non-empty set;
- $R \subseteq W \times W$ is a binary relation over W which is reflexive and transitive (R is *reflexive* if for all $w \in W$ Rww and *transitive* if for all $u, v, w \in W$, Ruv and Rvw imply Ruw);
- $R_\diamond \subseteq W \times W$ is a binary relation over W ;
- $P_1, \dots, P_n, \dots \subseteq W$ are unary relations over W such that for all $v, w \in W$, if Rvw and $P_n(v)$ then $P_n(w)$.

We write $w \in M$ for $w \in W$ by abuse and the pair (M, w) is called a *pointed modal intuitionistic model*. The class of all pointed modal intuitionistic models is denoted \mathcal{E}_{Int} . We define the *satisfaction relation* $\models_{\text{Int}} \subseteq \mathcal{E}_{\text{Int}} \times \mathcal{L}_{\text{Int}}$ by the following *truth conditions*. Below, we write $(M, w) \models \varphi$ for $((M, w), \varphi) \in \models_{\text{Int}}$. For all modal intuitionistic models $M = (W, \{R, R_\diamond, P_1, \dots, P_n, \dots\})$, all $w \in M$, all $\varphi, \psi \in \mathcal{L}_{\text{Int}}$ and all $p_i \in \mathbb{P}$,

$$\begin{aligned}
(M, w) \models \top & \quad \text{always;} \\
(M, w) \models \perp & \quad \text{never;} \\
(M, w) \models p_i & \quad \text{iff} \quad P_i(w) \text{ holds;} \\
(M, w) \models (\varphi \wedge \psi) & \quad \text{iff} \quad (M, w) \models \varphi \text{ and } (M, w) \models \psi; \\
(M, w) \models (\varphi \vee \psi) & \quad \text{iff} \quad (M, w) \models \varphi \text{ or } (M, w) \models \psi; \\
(M, w) \models (\varphi \Rightarrow \psi) & \quad \text{iff} \quad \text{for all } v \in W \text{ such that } Rvw, \text{ if } (M, v) \models \varphi \\
& \quad \text{then } (M, v) \models \psi; \\
(M, w) \models \square \varphi & \quad \text{iff} \quad \text{for all } v \in W \text{ such that } Rvw, \\
& \quad \text{for all } u \in W \text{ such that } R_\diamond vu, (M, u) \models \varphi; \\
(M, w) \models \diamond \varphi & \quad \text{iff} \quad \text{for all } v \in W \text{ such that } Rvw, \\
& \quad \text{there is } u \in W \text{ such that } R_\diamond vu \text{ and } (M, u) \models \varphi.
\end{aligned}$$

The triple $(\mathcal{L}_{\text{Int}}, \mathcal{E}_{\text{Int}}, \models_{\text{Int}})$ forms a logic, that we call *modal intuitionistic logic*.

2.2.4. Temporal Logic. The *temporal language* \mathcal{L}_{TL} is defined inductively by the following grammar in BNF:

$$\mathcal{L}_{\text{TL}}: \varphi ::= \top \mid \perp \mid p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid U(\varphi, \varphi) \mid S(\varphi, \varphi)$$

where $p \in \mathbb{A}$. A *temporal model* is a tuple $M = (W, \{<, P_1, \dots, P_n, \dots\})$ where:

- W is a non-empty set;
- $< \subseteq W \times W$ is a binary relation over W ;
- $P_1, \dots, P_n, \dots \subseteq W$ are unary relations over W .

We write $w \in M$ for $w \in W$ by abuse and the pair (M, w) is called a *pointed temporal model*. The class of all pointed temporal models is denoted \mathcal{E}_{TL} . We define the *satisfaction relation* $\models_{\text{TL}} \subseteq \mathcal{E}_{\text{TL}} \times \mathcal{L}_{\text{TL}}$ by the following *truth conditions*. Below, we write $(M, w) \models \varphi$ for $((M, w), \varphi) \in \models_{\text{TL}}$. For all temporal models $M = (W, \{<, P_1, \dots, P_n, \dots\})$, all $w \in M$, all $\varphi, \psi \in \mathcal{L}_{\text{TL}}$ and all $p_i \in \mathbb{P}$,

$(M, w) \models \top$	always;
$(M, w) \models \perp$	never;
$(M, w) \models p_i$	iff $P_i(w)$ holds;
$(M, w) \models \neg p_i$	iff $P_i(w)$ does not hold;
$(M, w) \models (\varphi \wedge \psi)$	iff $(M, w) \models \varphi$ and $(M, w) \models \psi$;
$(M, w) \models (\varphi \vee \psi)$	iff $(M, w) \models \varphi$ or $(M, w) \models \psi$;
$(M, w) \models U(\varphi, \psi)$	iff there is $v \in W$ such that $w < v$ and $(M, v) \models \varphi$ and for all $u \in W$ such that $w < u < v$, $(M, u) \models \psi$;
$(M, w) \models S(\varphi, \psi)$	iff there is $v \in W$ such that $v < w$ and $(M, v) \models \varphi$ and for all $u \in W$ such that $v < u < w$, $(M, u) \models \psi$.

The triple $(\mathcal{L}_{\text{TL}}, \mathcal{E}_{\text{TL}}, \models_{\text{TL}})$ forms a logic, that we call *temporal logic*.

2.2.5. Many-valued Logics. Our presentation of many-valued logic is inspired by Priest [36] but is slightly different from the usual presentation. V is a set called the *truth values*. Let $D \subseteq V$ be a subset of *designated values* and let $\mathbf{C} = \{\star_1, \dots, \star_m, \dots\}$ be a countable set of connectives of arity k_1, \dots, k_m, \dots . The *many-valued language* $\mathcal{L}_{\text{MV}}^{\mathbf{C}}$ associated to \mathbf{C} is defined inductively by the following grammar in BNF:

$$\mathcal{L}_{\text{MV}}^{\mathbf{C}}: \varphi ::= p \mid \star(\varphi, \dots, \varphi)$$

where $p \in \mathbb{P}$ and $\star \in \mathbf{C}$. A *many-valued model* is a tuple $M = (V, \{R_{\star_1}, \dots, R_{\star_m}, \dots, P_1, \dots, P_n, \dots\})$ where:

- V is the set of truth values;
- $R_{\star_1}, \dots, R_{\star_m}, \dots$ are relations over V of arity $k_1 + 1, \dots, k_m + 1, \dots$;
- $P_1, \dots, P_n, \dots \subseteq V$ are unary relations over V .

The relations R_{\star} are obtained from the usual truth functions f of many-valued logics by the connection $R_{\star}w_1 \dots w_k w$ iff $f(w_1, \dots, w_k) = w$. We write $w \in M$ for $w \in V$ and the pair (M, w) is called a *pointed many-valued model*. The class of all pointed many-valued models is denoted \mathcal{E}_{MV} . We also define a *designated many-valued model* as a pair (M, D) where M is a many-valued model (and D is the set of designated values). We define the *satisfaction*

relation $\models_{\text{MV}} \subseteq \mathcal{E}_{\text{MV}} \times \mathcal{L}_{\text{MV}}^{\text{C}}$ by the following *truth conditions*. Below, we write $(M, w) \models \varphi$ for $((M, w), \varphi) \in \models_{\text{MV}}$. For all many-valued models $M = (V, \{R_{*1}, \dots, R_{*m}, \dots, P_1, \dots, P_n, \dots\})$, all $w \in M$, all $\varphi_1, \dots, \varphi_k \in \mathcal{L}_{\text{MV}}^{\text{C}}$ and all $p_i \in \mathbb{P}$,

$$\begin{aligned} (M, w) \models p_i & \quad \text{iff } P_i(w) \text{ holds;} \\ (M, w) \models *(\varphi_1, \dots, \varphi_k) & \quad \text{iff there are } w_1, \dots, w_k \in V \text{ such that } R_{*}w_1 \dots w_k w \\ & \quad \text{and } (M, w_1) \models \varphi_1 \text{ and } \dots \text{ and } (M, w_k) \models \varphi_k \end{aligned}$$

We extend the satisfaction relation \models_{ML} to the set of designated many-valued models (M, D) as follows: we set $(M, D) \models \varphi$ iff there is $w \in D$ such that $(M, w) \models \varphi$.

Typically, a many-valued logic is based on a class of designated many-valued models whose truth functions associated to the same connective are the same in every model of the class. A class of designated many-valued models satisfying this condition is called a *many-valued class of designated models*. So, a *many-valued logic* associated to a set of connectives C and designated values D is a triple $(\mathcal{L}_{\text{MV}}^{\text{C}}, \mathcal{E}_{\text{MV}}^D, \models_{\text{MV}})$ where $\mathcal{E}_{\text{MV}}^D$ is a many-valued class of designated models. Fuzzy logic, the 3-valued logics of Kleene and Łukasiewicz are examples of many-valued logics in which the unary predicates P_i are singletons. Our general approach also allows us to capture the logic of first-degree entailment (FDE) since in that case the P_i s are not necessarily singletons (see for instance [36] for more details on many-valued logics).

2.3. Common Logical Notions

In the present section, we define a number of notions which are common to all logics and in particular to the logics introduced beforehand. The way we define logics is different from many proposals considered in universal logic [8] such as pairs of Suzsko’s abstract logics, Tarski’s consequence operators or logical structures. Often a logic is viewed as a pair of a language together with a consequence relation on this language. Our approach to defining logics is somehow more ‘semantic’ in that respect than the usual proposals. It corresponds in fact to the “abstract logics” of García-Matos & Väänänen [19] and to the “rooms” of Mossakowski et al. [33].

We will say that a *logic* is a triple $\mathbb{L} \triangleq (\mathcal{L}, \mathcal{E}, \models)$ where

- \mathcal{L} is a *logical language* defined as a set of well-formed expressions built from a set of *connectives* C and a set of *propositional letters* \mathbb{A} ;
- \mathcal{E} is a *class of pointed models*;
- \models is a *satisfaction relation* which relates in a compositional manner elements of \mathcal{L} to models of \mathcal{E} by means of so-called *truth conditions*.

Let $\mathbb{L} = (\mathcal{L}, \mathcal{E}, \models)$ be a logic and let $\Gamma \subseteq \mathcal{L}$, $\varphi \in \mathcal{L}$ and $M \in \mathcal{E}$. We write $M \models \Gamma$ when for all $\psi \in \Gamma$, we have $M \models \psi$. Then, we say that

- φ is *true (satisfied)* at M or M is a *model* of φ when $M \models \varphi$;
- φ is a *logical consequence* of Γ , written $\Gamma \models_{\mathbb{L}} \varphi$, when for all $M \in \mathcal{E}$, if $M \models \Gamma$ then $M \models \varphi$;
- φ is *valid*, written $\models_{\mathbb{L}} \varphi$, when for all models $M \in \mathcal{E}$, we have $M \models \varphi$;

- φ is *satisfiable* when $\neg\varphi$ is not valid in \mathcal{E} , *i.e.* when there is a model $M \in \mathcal{E}$ such that $M \models \varphi$.

If Γ is a singleton $\Gamma = \{\psi\}$, we also write by abuse $\psi \models \varphi$ for $\{\psi\} \models \varphi$.

A set of formulas of \mathcal{L} is called a *theory*. A set Δ of formulas of \mathcal{L} is said to be a *set of axioms* for a theory Γ iff Γ and Δ have the same logical consequences. A theory is called *finitely axiomatizable* iff it has a finite set of axioms. A logic L is *axiomatizable* if its set of validities is finitely axiomatizable.

3. On the Relative Expressivity of Logics

When two logics $L_1 = (\mathcal{L}_1, \mathcal{E}, \models_1)$ and $L_2 = (\mathcal{L}_2, \mathcal{E}, \models_2)$ are interpreted on the same class of models \mathcal{E} , there is a standard way to compare their relative expressiveness. We say that L_1 is *at least as expressive as* L_2 , denoted $L_2 \leq L_1$, when there is a mapping $T : \mathcal{L}_2 \rightarrow \mathcal{L}_1$ such that for all $\varphi_2 \in \mathcal{L}_2$ and all $M \in \mathcal{E}$, $M \models \varphi_2$ iff $M \models T(\varphi_2)$. These mappings naturally induce conservative translation morphisms between logics viewed as pairs of language and consequence relation in the sense of Arndt & Al. [2] and yield a category of logics, which are all based on the same class of models \mathcal{E} .

When $L_1 \geq L_2$ and $L_2 \geq L_1$, we say that L_1 is *as expressive as* L_2 and denote it $L_1 \equiv L_2$. In that case, the definition rewrites as follows:

- there is a mapping $T_1 : \mathcal{L}_1 \rightarrow \mathcal{L}_2$ such that for all $\varphi_1 \in \mathcal{L}_1$ and all $M \in \mathcal{E}$, $M \models \varphi_1$ iff $M \models T_1(\varphi_1)$;
- there is a mapping $T_2 : \mathcal{L}_2 \rightarrow \mathcal{L}_1$ such that for all $\varphi_2 \in \mathcal{L}_2$ and all $M \in \mathcal{E}$, $M \models \varphi_2$ iff $M \models T_2(\varphi_2)$.

Now, given a logic $(\mathcal{L}, \mathcal{E}, \models)$, for all $\varphi, \psi \in \mathcal{L}$, we write $\varphi \equiv \psi$ when for all $M \in \mathcal{E}$, it holds that $M \models \varphi$ iff $M \models \psi$ and for all $M, N \in \mathcal{E}$, we write $M \equiv N$ when for all $\varphi \in \mathcal{L}$, it holds $M \models \varphi$ iff $N \models \varphi$. If $\mathcal{M}, \mathcal{N} \subseteq \mathcal{E}$, we write $\mathcal{M} \equiv \mathcal{N}$ when for all $M \in \mathcal{M}$ there is $N \in \mathcal{N}$ such that $M \equiv N$, and vice versa. With these notations, that definition of equi-expressivity entails in particular the following two facts: for all $\varphi_1 \in \mathcal{L}_1$ and all $\varphi_2 \in \mathcal{L}_2$,

$$\{M \in \mathcal{E} \mid M \models \varphi_1\} = \{M \in \mathcal{E} \mid M \models T_1(\varphi_1)\} \quad \varphi_1 \equiv T_2(T_1(\varphi_1)) \quad (1)$$

$$\{M \in \mathcal{E} \mid M \models \varphi_2\} = \{M \in \mathcal{E} \mid M \models T_2(\varphi_2)\} \quad \varphi_2 \equiv T_1(T_2(\varphi_2)) \quad (2)$$

However, when two logics $L_1 = (\mathcal{L}_1, \mathcal{E}_1, \models_1)$ and $L_2 = (\mathcal{L}_2, \mathcal{E}_2, \models_2)$ are interpreted over different classes of models \mathcal{E}_1 and \mathcal{E}_2 , there is no widely accepted way to compare their relative expressiveness, even if some proposals have already been made [19, 33] (we will come back to them at the end of this section). We are now going to propose some notions to deal with that issue. Our proposal is the following.

Definition 1 (Equal expressivity). A logic $L_1 = (\mathcal{L}_1, \mathcal{E}_1, \models_1)$ is *as expressive as* a logic $L_2 = (\mathcal{L}_2, \mathcal{E}_2, \models_2)$, written $L_1 \equiv L_2$, when the following conditions hold:

1. there is a mapping $T_1^\varphi : \mathcal{L}_1 \rightarrow \mathcal{L}_2$ and a mapping $T_1^M : \mathcal{E}_1 \rightarrow \mathcal{E}_2$ such that for all $\varphi_1 \in \mathcal{L}_1$ and all $M_1 \in \mathcal{E}_1$, it holds that $M_1 \models \varphi_1$ iff $T_1^M(M_1) \models T_1^\varphi(\varphi_1)$;
2. there is a mapping $T_2^\varphi : \mathcal{L}_2 \rightarrow \mathcal{L}_1$ and a mapping $T_2^M : \mathcal{E}_2 \rightarrow \mathcal{E}_1$ such that for all $\varphi_2 \in \mathcal{L}_2$ and all $M_2 \in \mathcal{E}_2$, it holds that $M_2 \models \varphi_2$ iff $T_2^M(M_2) \models T_2^\varphi(\varphi_2)$;
3. for all $M_1 \in \mathcal{E}_1$ and all $M_2 \in \mathcal{E}_2$, it holds that $T_2^M(T_1^M(M_1)) \equiv M_1$ and $T_1^M(T_2^M(M_2)) \equiv M_2$. \dashv

Our third condition states that T_2^M and T_1^M are inverse bijections of each other (modulo some natural congruence \equiv). Hence, our definition is set in such a way that we compare the relative expressivity of each logic by comparing them over their whole class of models, taking into account the specificities of *all* the models of each logic in the comparison. This stems from the surjectivity of T_2^M and T_1^M . Moreover, if two logics over the same class of models are as expressive in the previous sense, they are also equi-expressive: it suffices to take the identity mappings for T_1^M and T_2^M . Our definition of equi-expressivity entails also the following two facts, which generalize the two previous expressions (1) and (2): for all $\varphi_1 \in \mathcal{L}_1$ and all $\varphi_2 \in \mathcal{L}_2$,

$$T_1^M(\{M_1 \in \mathcal{E}_1 \mid M_1 \models \varphi_1\}) \equiv \{M_2 \in \mathcal{E}_2 \mid M_2 \models T_1^\varphi(\varphi_1)\} \quad \varphi_1 \equiv T_2^\varphi(T_1^\varphi(\varphi_1))$$

$$T_2^M(\{M_2 \in \mathcal{E}_2 \mid M_2 \models \varphi_2\}) \equiv \{M_1 \in \mathcal{E}_1 \mid M_1 \models T_2^\varphi(\varphi_2)\} \quad \varphi_2 \equiv T_1^\varphi(T_2^\varphi(\varphi_2))$$

We canonically extend our definitions of equi-expressivity to classes of logics.

Definition 2 (Equal expressivity of classes of logics). Let \mathbb{L}_1 and \mathbb{L}_2 be two classes of logics. We say that \mathbb{L}_1 is *as expressive as* \mathbb{L}_2 , written $\mathbb{L}_1 \equiv \mathbb{L}_2$, when for all $L_1 \in \mathbb{L}_1$ there is $L_2 \in \mathbb{L}_2$ such that $L_1 \equiv L_2$, and vice versa. \dashv

In a sense, our definition of equal expressivity can be viewed as a partial solution to the so-called “identity problem” of universal logic [8] for logics which are defined semantically by triples like in the previous section. Our proposal is different from the one of García-Matos & Väänänen [19], although they deal with a more general notion of embedding between logics based on different classes of models, that is an embedding in only one direction, of one logic into another. If two logics L_1 and L_2 are equally expressive in our sense then there exist two “model-expansive corridors” from L_1 to L_2 and from L_2 to L_1 in the sense of Mossakowski et al. [33] (with the proviso that the surjection holds modulo the congruence \equiv). So, our notion of equi-expressivity is more demanding and stronger than their notion. On the other hand, our “logics” can in fact be seen as institutions [33]. In that case, one can prove that two logics are “equally expressive” in our sense if, and only if, they are “equivalent” in the sense of institutions [34, Definition 3.5, p. 118].¹

¹I thank Peter Arndt for checking and proving that result.

4. Protologics, Atomic and Molecular Logics

Non-classical logics have common features: their syntax is defined by means of connectives, like $\wedge, \vee, \neg, \rightarrow, \Box, \Diamond, \circ, \backslash, /, \Rightarrow, \dots$; there is not any explicit variable quantification and no variables appear in formulas; they can be given a Kripke-style relational semantics by means of specific structures; the semantics of their connectives are defined by means of truth conditions.

Our overall approach is based on the observation that we can view truth conditions as formulas of first-order logic and that the models considered are very often specific kinds of structures. We revisit below the definitions of non-classical logics of Section 2 and provide the first-order formula $\star(x)$ or $c(x)$ with one free variable x corresponding to the respective truth condition of the non-classical connective. These first-order formulas are written in a specific form that will be explained and become clear later in the article.

- $(M, w) \models \Box_i p$ iff for all $v \in M$, if $R_i wv$ then $(M, v) \models p$
 $\star_1(x) \triangleq \forall y (Py \vee \neg R_i xy)$
- $(M, w) \models p \circ q$ iff there are $v, u \in M$ such that $Rvwu$, $(M, v) \models p$ and $(M, u) \models q$
 $\star_2(x) \triangleq \exists yz (Py \wedge Qz \wedge Ryzx)$
- $(M, w) \models p \backslash q$ iff for all $v, u \in M$ such that $Rwvu$, if $(M, v) \models p$ then $(M, u) \models q$
 $\star_2(x) \triangleq \forall yz (\neg Py \vee Qz \vee \neg Rxyz)$
- $(M, w) \models q / p$ iff for all $v, u \in M$ such that $Rvwu$ if $(M, v) \models p$ then $(M, u) \models q$
 $\star_2(x) \triangleq \forall yz (\neg Py \vee Qz \vee \neg Ryzx)$
- $(M, w) \models \star(\varphi_1, \dots, \varphi_n)$ iff there are $w_1, \dots, w_n \in W$ such that $R_\star w_1 \dots w_n w$ and $(M, w_1) \models \varphi_1$ and \dots and $(M, w_n) \models \varphi_n$
 $\star_n(x) \triangleq \exists x_1 \dots x_n (P_1 x_1 \wedge \dots \wedge P_n x_n \wedge R_{x_1 \dots x_n} x)$
- $(M, w) \models \Box \varphi$ iff for all $v \in W$ such that Rwv , for all $u \in W$ such that $R_\Diamond vu$, $(M, u) \models \varphi$
 $c_1(x) \triangleq \forall y (\forall z (Pz \vee \neg R_\Diamond yz) \vee \neg Rxy)$
- $(M, w) \models U(\varphi, \psi)$ iff there is $v \in W$ such that $w < v$ and $(M, v) \models \varphi$ and for all $u \in W$ such that $w < u < v$, $(M, u) \models \psi$
 $c_2(x) \triangleq \exists z (Pz \wedge \forall y (Qy \vee \neg(x < y < z))) \wedge x < z)$

4.1. Protologics

In this section, \mathcal{P} is a set of predicates symbols and the set of constants of first-order logics is empty, $\mathcal{C} = \emptyset$.

Definition 3 (Abstract connectives). The *abstract propositional letters* \mathbb{P}^a are a subset $\mathcal{Q} \subseteq \mathcal{P}$ of the predicate symbols \mathcal{P} and the *abstract connectives* \mathbb{C}^a are the formulas of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x})$ together with a non-empty sequence of distinct predicate symbols that occur in each formula. That is,

$$\mathbb{P}^a \triangleq \mathcal{Q}$$

$$\mathbb{C}^a \triangleq \{(\chi(\bar{x}), (\mathbb{Q}_1, \dots, \mathbb{Q}_n)) \mid \chi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}) \text{ and} \\ \mathbb{Q}_1, \dots, \mathbb{Q}_n \in \mathcal{Q} \text{ all distinct appear in } \chi\}$$

Let p be an abstract propositional letter whose arity as a predicate symbol is k . Then, the *arity* of p is 0 and its *type* is k . Let $\star = (\chi(x_1, \dots, x_k), (\mathbb{Q}_1, \dots, \mathbb{Q}_n)) \in \mathbb{C}^a$ where χ is a formula with k free variables and such that the predicates $\mathbb{Q}_1, \dots, \mathbb{Q}_n$ are of arity k_1, \dots, k_n respectively. Then, the *arity* of \star is n , its *type signature* is (k, k_1, \dots, k_n) and its *output type* is k , denoted $k(\star)$. The predicate symbols of $\chi(\bar{x})$ which do not belong to $\{\mathbb{Q}_1, \dots, \mathbb{Q}_n\}$ are called the *parameter predicates*. \dashv

One needs to explicitly specify the predicates $\mathbb{Q}_1, \dots, \mathbb{Q}_n$ for each abstract connective because a predicate symbol \mathbb{Q} could in principle be used both as an abstract propositional letter and as a parameter predicate.

Example 1. Let us consider the formula $\chi(x) \triangleq \forall y(\forall z(\mathbb{Q}z \vee \neg \mathbb{R}_{\diamond}yz) \vee \neg \mathbb{R}xy)$ or equivalently $\chi(x) \triangleq \forall y(\mathbb{R}xy \rightarrow \forall z(\mathbb{R}_{\diamond}yz \rightarrow \mathbb{Q}(z)))$ corresponding to the truth condition of the box operator of modal intuitionistic logic. The connective $(\chi(x), (\mathbb{Q}))$ of \mathbb{C}^a of arity 1, of type signature $(1, 1)$ and of output type 1 corresponds to the connective of modal intuitionistic logic. Its parameter predicates are \mathbb{R} and \mathbb{R}_{\diamond} . We could define other connectives based on $\chi(x)$ such as $(\chi(x), (\mathbb{R}_{\diamond}, \mathbb{Q}))$ and $(\chi(x), (\mathbb{R}, \mathbb{R}_{\diamond}, \mathbb{Q}))$ of arities 2 and 3 and of type signatures $(1, 1, 1)$ and $(1, 1, 1, 1)$ respectively, possibly with the predicates ordered differently in the tuples. The parameter predicate in $(\chi(x), (\mathbb{R}_{\diamond}, \mathbb{Q}))$ is \mathbb{R} and there is none in $(\chi(x), (\mathbb{R}, \mathbb{R}_{\diamond}, \mathbb{Q}))$. \dashv

Definition 4 (Protolanguage). The *protolanguage* \mathcal{L} is the smallest set that contains the abstract propositional letters and that is closed under the other abstract connectives, while respecting the type constraints:

- $\mathbb{P}^a \subseteq \mathcal{L}$;
- for all $\star \in \mathbb{C}^a$ of type signature (k, k_1, \dots, k_n) and for all $\varphi_1, \dots, \varphi_n \in \mathcal{L}$ of respective types k_1, \dots, k_n , we have $\star(\varphi_1, \dots, \varphi_n) \in \mathcal{L}$. The *type* of $\star(\varphi_1, \dots, \varphi_n)$ is k .

If $\mathbb{C}^a \subseteq \mathbb{C}^a$ is such that $\mathbb{C}^a \cap \mathbb{P}^a \neq \emptyset$, then an element of $\mathcal{L}_{\mathbb{C}^a}$ is an element of \mathcal{L} that contains only connectives of \mathbb{C}^a . *In the sequel, we always assume that all $\mathbb{C}^a \subseteq \mathbb{C}^a$ are such that $\mathbb{C}^a \cap \mathbb{P}^a \neq \emptyset$.* Elements of \mathcal{L} are called *protoformulas* and are generally denoted φ, ψ, α . \dashv

Example 2. If we want to recover the language of modal intuitionistic logic with only the box modality then we consider the set of connectives $\mathbb{C} = \{p, (\chi(x), (\mathbb{Q})) \mid p \in \mathcal{Q} \text{ is of arity } 1\}$ where $\chi(x)$ is the formula of the previous example. \dashv

Definition 5 (\mathbb{C}^a -model). Let $\mathbb{C}^a \subseteq \mathbb{C}^a$ be a set of connectives. A \mathbb{C}^a -*model* is a structure $M = (W, \mathcal{R})$ for $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x})$ where W is a non-empty set and \mathcal{R} is a set of relations over W such that each predicate $\mathbb{Q} \in \mathcal{Q}$ is associated to a relation Q of the same arity as \mathbb{Q} and such that the parameter predicates $\{\mathbb{R}_1, \dots, \mathbb{R}_m\}$ of each connective $\star = (\chi(\bar{x}), (\mathbb{Q}_1, \dots, \mathbb{Q}_n)) \in \mathbb{C}^a$ can be associated to a subset

of the relations $\mathcal{R}_\star = \{R_1, \dots, R_m\} \subseteq \mathcal{R}$ interpreting the predicates R_1, \dots, R_m (possibly with some overlap for different connectives).

A \mathbb{C}^a -assignment for M is a tuple of W^k , generally denoted \bar{w} , where $k = k(\star)$ for some $\star \in \mathbb{C}^a$. The set of all \mathbb{C}^a -assignments for M is denoted $\bar{w}(M, \mathbb{C}^a)$. A *pointed \mathbb{C}^a -model* (M, \bar{w}) is a \mathbb{C}^a -model M together with a \mathbb{C}^a -assignment \bar{w} for M . The class of all pointed \mathbb{C}^a -models is denoted \mathcal{M}^a . \dashv

Example 3. If we resume our previous example, a modal intuitionistic model $M = (W, \{R, R_\diamond, P_1, \dots, P_n, \dots\})$ is a \mathbb{C} -model. The parameter predicates $\{R, R_\diamond\}$ of $(\chi(x), (Q))$ are associated to the relations $\{R, R_\diamond\}$ and the predicates $p_n \in \mathcal{Q}$ are associated to the relations P_n . \dashv

Definition 6 (Model $M[Q_i := W_i]$). Let $\mathbb{C}^a \subseteq \mathbb{C}^a$ be a set of connectives, let M be a \mathbb{C}^a -model containing relations Q_1, \dots, Q_n of arity k_1, \dots, k_n respectively and let $W_1 \in \mathcal{P}(W^{k_1}), \dots, W_k \in \mathcal{P}(W^{k_n})$. We define the \mathbb{C}^a -model $M[Q_i := W_i]$ as the \mathbb{C}^a -model M where (the interpretation of the predicates Q_1, \dots, Q_n by) the relations Q_1, \dots, Q_n are replaced by the relations W_1, \dots, W_n (viewed as sets) respectively, all the rest being the same. \dashv

Definition 7 (Truth function associated to a connective of \mathbb{C}^a). Let $\star = (\chi(\bar{x}), (Q_1, \dots, Q_n)) \in \mathbb{C}^a$ be a connective of type signature (k, k_1, \dots, k_n) and let $M = (W, \mathcal{R})$ be a \mathbb{C}^a -model such that $(\chi(\bar{x}), (Q_1, \dots, Q_n)) \in \mathbb{C}^a$. The (k, k_1, \dots, k_n) -ary truth function $f_\star : \mathcal{P}(W^{k_1}) \times \dots \times \mathcal{P}(W^{k_n}) \rightarrow \mathcal{P}(W^k)$ associated to \star on M is defined as follows:

- if $n = 0$ and $\star = Q$, $f_\star \triangleq Q$;
- if $n > 0$, then for all $W_1 \in \mathcal{P}(W^{k_1}), \dots$, all $W_n \in \mathcal{P}(W^{k_n})$ we define

$$f_\star(W_1, \dots, W_n) \triangleq \{(w_1, \dots, w_k) \in W^k \mid M[Q_i := W_i] \models \chi[x_1/w_1, \dots, x_k/w_k]\} \dashv$$

In the above definition, M is really taken as a structure of first-order logic. The choice of the assignment s in the evaluation does not play a role in the determination of f_\star since the only value of variables that matter for that, (x_1, \dots, x_k) , are given by the definition.

Example 4. One can easily check that the truth functions associated to the connectives $(\star_1(x), (Q)), (\star_2(x), (Q_1, Q_2)), (\star_3(x), (Q_1, Q_2))$ and $(\star_4(x), (Q_1, Q_2))$ at the beginning of Section 4 correspond to the truth conditions of the connectives defined above each of them respectively. For example, for the case of modal intuitionistic logic, if $\star = (\chi(x), (Q))$ then $f_\star(W_1) = \{w \in W \mid M[P := W_1] \models \chi(x)\} = \{w \in W \mid M[P := W_1] \models \forall y(\forall z(Qz \vee \neg R_\diamond yz) \vee \neg Rxy)[x/w]\} = \{w \in W \mid \text{for all } v \in W \text{ such that } R w v, \text{ for all } u \in W \text{ such that } R_\diamond v u, u \in W_1\}$. \dashv

Definition 8 (Protologic). Let $\mathbb{C}^a \subseteq \mathbb{C}^a$ and let $M = (W, \mathcal{R})$ be a \mathbb{C}^a -model. The *extension function of $\mathcal{L}_{\mathbb{C}^a}$ in M* , denoted $\llbracket \cdot \rrbracket^M : \mathcal{L}_{\mathbb{C}^a} \rightarrow \bar{w}(M, \mathbb{C}^a)$, is defined inductively as follows: for all $p \in \mathbb{C}^a \cap \mathbb{P}^a$ and all $\star \in \mathbb{C}^a$ of arity $n > 0$ and type signature (k, k_1, \dots, k_n) , for all $\varphi_1, \dots, \varphi_n \in \mathcal{L}_{\mathbb{C}^a}$ of respective types

$k_1, \dots, k_n,$

$$\begin{aligned} \llbracket p \rrbracket^M &\triangleq f_p \\ \llbracket \star(\varphi_1, \dots, \varphi_n) \rrbracket^M &\triangleq f_\star(\llbracket \varphi_1 \rrbracket^M, \dots, \llbracket \varphi_n \rrbracket^M) \end{aligned}$$

where f_\star is the truth function associated to the abstract connective \star .

If $\mathcal{E}_{\mathcal{C}^a}$ is a class of pointed \mathcal{C}^a -models, we define the *satisfaction relation* $\Vdash \subseteq \mathcal{E}_{\mathcal{C}^a} \times \mathcal{L}_{\mathcal{C}^a}$ as follows: for all $\varphi \in \mathcal{L}_{\mathcal{C}^a}$ and all $(M, \bar{w}) \in \mathcal{E}_{\mathcal{C}^a}$, we set $(M, \bar{w}) \Vdash \varphi$ iff $\bar{w} \in \llbracket \varphi \rrbracket^M$. The triple $(\mathcal{L}_{\mathcal{C}^a}, \mathcal{E}_{\mathcal{C}^a}, \Vdash)$ is called the *protologic associated to $\mathcal{E}_{\mathcal{C}^a}$ and \mathcal{C}^a* . \dashv

When the type of φ does not match the size of the assignment $\bar{w} \in \bar{w}(M, \mathcal{C}^a)$ of a pointed \mathcal{C}^a -model, then it is not the case that $((M, \bar{w}), \varphi) \in \Vdash$. That is, $(M, \bar{w}) \Vdash \varphi$ does not hold. In that case, we do not say that φ is “undefined” or “false” at the pointed \mathcal{C}^a -model (M, \bar{w}) ; what only holds is that the pair $((M, \bar{w}), \varphi)$ does not belong to the relation \Vdash .

One may argue that protologics do not really deserve their qualification of being somehow ‘primal’ since they seem at first sight to be able to encode only two-valued logics. This is not the case, as the following example shows.

Example 5 (Many-valued logics). Many-valued logics associated to a set of connectives \mathcal{C} are examples of protologics. The first-order formulas $\chi(x) \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x})$ on which they are based and that define the connectives are of the form $\exists x_1 \dots x_n (\mathbf{Q}_1 x_1 \wedge \dots \wedge \mathbf{Q}_n x_n \wedge \mathbf{R}_\star x_1 \dots x_n x)$. \dashv

4.2. Atomic Logics

Atomic logics are protologics whose connective skeletons are defined by first-order formulas of the form $\forall x_1 \dots x_n (\pm_1 \mathbf{Q}_1 x_1 \vee \dots \vee \pm_n \mathbf{Q}_n x_n \vee \pm \mathbf{R} x_1 \dots x_n x)$ or $\exists x_1 \dots x_n (\pm_1 \mathbf{Q}_1 x_1 \wedge \dots \wedge \pm_n \mathbf{Q}_n x_n \wedge \pm \mathbf{R} x_1 \dots x_n x)$ where the \pm_i s and \pm are either empty or \neg . Likewise, propositional letters are defined by first-order formulas of the form $\pm \mathbf{R} x$. We will represent the structure of these formulas by means of so-called *skeletons* whose various arguments capture the different features and patterns from which they can be redefined completely. Atomic logics are also generalizations of our gaggle logics [4, 5] with types associated to formulas.

We recall that \mathbb{N}^* denotes the set of natural numbers minus 0 and that for all $n \in \mathbb{N}^*$, \mathfrak{S}_n denotes the group of permutations over the set $\{1, \dots, n\}$. Permutations are generally denoted σ, τ , the identity permutation Id is sometimes denoted 1 as the neutral element of every permutation group and σ^- stands for the inverse permutation of the permutation σ . For example, the permutation $\sigma = (3, 1, 2)$ is the permutation that maps 1 to 3, 2 to 1 and 3 to 2 (see for instance [39] for more details).

Definition 9 (Atomic skeletons and connectives). The sets of *atomic skeletons* \mathbb{P} and \mathbb{C} are defined as follows:

$$\begin{aligned} \mathbb{P} &\triangleq \mathfrak{S}_1 \times \{+, -\} \times \{\forall, \exists\} \times \mathbb{N}^* \\ \mathbb{C} &\triangleq \mathbb{P} \cup \bigcup_{n \in \mathbb{N}^*} \left\{ \mathfrak{S}_{n+1} \times \{+, -\} \times \{\forall, \exists\} \times \mathbb{N}^{*n+1} \times \{+, -\}^n \right\}. \end{aligned}$$

\mathbb{P} is called the set of *propositional letter skeletons* and \mathbb{C} is called the set of *connective skeletons*. They can be represented by tuples $(\sigma, \pm, \mathbb{A}, \bar{k}, \bar{\pm}_j)$ or $(\sigma, \pm, \mathbb{A}, k)$ if it is a propositional letter skeleton, where $\mathbb{A} \in \{\forall, \exists\}$ is called the *quantification signature* of the skeleton, $\bar{k} = (k, k_1, \dots, k_n) \in \mathbb{N}^{*n+1}$ is called the *type signature* of the skeleton and $\bar{\pm}_j = (\pm_1, \dots, \pm_n) \in \{+, -\}^n$ is called the *tonicity signature* of the skeleton; $(\mathbb{A}, \bar{k}, \bar{\pm}_j)$ is called the *signature* of the skeleton. The *arity* of a propositional letter skeleton is 0 and its *type* is k . The *arity* of a skeleton $\star \in \mathbb{C} - \mathbb{P}$ is n , its *input types* are k_1, \dots, k_n and its *output type* is k .

An *atomic connective* is a symbol generally denoted \star to which is associated an atomic skeleton. A *propositional letter* is a symbol generally denoted p to which is associated a propositional letter skeleton. Their arity, signature, quantification signature, type signature, tonicity signature, input and output types are the same as their skeleton. By abuse, we sometimes identify a connective with its skeleton. If \mathbb{C} is a set of atomic connectives, its set of propositional letters is denoted $\mathbb{P}(\mathbb{C})$.

Propositional letters are denoted $p, p_1, p_2, \text{ etc.}$ and connectives $\star, \star_1, \star_2, \text{ etc.}$

We need to distinguish between connectives and skeletons because in general we need a countable number of propositional letters or connectives of the same skeleton, like in some modal logics, where we need multiple modalities of the same (similarity) type/skeleton.

Remark 1. The permutations σ mentioned in atomic skeletons do not really play a role in this article. Permutations play an important role in the proof theory of atomic logics, which is dealt with in [4, 5].

Definition 10 (Atomic language). Let \mathbb{C} be a set of atomic connectives. The (*typed*) *atomic language* $\mathcal{L}_{\mathbb{C}}$ associated to \mathbb{C} is the smallest set that contains the propositional letters and that is closed under the atomic connectives. That is,

- $\mathbb{P}(\mathbb{C}) \subseteq \mathcal{L}_{\mathbb{C}}$;
- for all $\star \in \mathbb{C}$ of arity $n > 0$ and of type signature (k, k_1, \dots, k_n) and for all $\varphi_1, \dots, \varphi_n \in \mathcal{L}_{\mathbb{C}}$ of types k_1, \dots, k_n respectively, we have that $\star(\varphi_1, \dots, \varphi_n) \in \mathcal{L}_{\mathbb{C}}$ and $\star(\varphi_1, \dots, \varphi_n)$ is of type k .

Elements of $\mathcal{L}_{\mathbb{C}}$ are called *atomic formulas* and are denoted $\varphi, \psi, \alpha, \dots$. The *type of a formula* $\varphi \in \mathcal{L}_{\mathbb{C}}$ is denoted $k(\varphi)$.

The *skeleton syntactic tree* of a formula $\varphi \in \mathcal{L}_{\mathbb{C}}$ is the syntactic tree of the formula φ in which the nodes labeled with subformulas of φ are replaced by the skeleton of their outermost connective.

A set of atomic connectives \mathbb{C} is *plain* if for all $\star \in \mathbb{C}$ of skeleton $(\sigma, \pm, \mathbb{A}, (k, k_1, \dots, k_n), (\pm_1, \dots, \pm_n))$ there are atoms $p_1, \dots, p_n \in \mathbb{P}$ of types k_1, \dots, k_n respectively. *In the sequel, we assume that all sets of connectives \mathbb{C} are plain.*

Our assumption that all sets of connectives \mathbb{C} considered are plain makes sense. Indeed, we want all connectives of \mathbb{C} to appear in some formula of $\mathcal{L}_{\mathbb{C}}$.

If C was not plain then there would be a connective of C (with input type k) which would be necessarily composed with another connective of C (of output type k), if we want such a connective to appear in a formula of \mathcal{L}_{C} . Yet, in that case, we should instead view C as a set of *molecular* connectives (introduced in the next section).

Definition 11 (C–models). Let C be a set of atomic connectives. A *C–model* is a tuple $M = (W, \mathcal{R})$ where W is a non-empty set and \mathcal{R} is a set of relations over W such that each n -ary connective $\star \in \mathsf{C}$ of type signature (k, k_1, \dots, k_n) is associated to a $k_1 + \dots + k_n + k$ -ary relation $R_\star \in \mathcal{R}$.

An *assignment* is a tuple $(w_1, \dots, w_k) \in W^k$ for some $k \in \mathbb{N}^*$, generally denoted \bar{w} . A *pointed C–model* (M, \bar{w}) is a C–model M together with an assignment \bar{w} . In that case, we say that (M, \bar{w}) is of *type* k . The class of all pointed C–models is denoted \mathcal{M}_{C} . \dashv

Note that a C–model can be canonically seen as a structure, for some appropriate set of predicates \mathcal{P} associated to the relations of \mathcal{R} .

Definition 12 (Atomic logics). Let C be a set of atomic connectives and let $M = (W, \mathcal{R})$ be a C–model. We define the *interpretation function of \mathcal{L}_{C} in M* , denoted $\llbracket \cdot \rrbracket^M : \mathcal{L}_{\mathsf{C}} \rightarrow \bigcup_{k \in \mathbb{N}^*} W^k$, inductively as follows. For all propositional letters $p \in \mathsf{C}$, all connectives $\star \in \mathsf{C}$ of skeleton $(\sigma, \pm, \mathbb{A}, (k, k_1, \dots, k_n), (\pm_1, \dots, \pm_n))$ of arity $n > 0$, for all $\varphi_1, \dots, \varphi_n \in \mathcal{L}_{\mathsf{C}}$,

$$\begin{aligned} \llbracket p \rrbracket^M &\triangleq \pm R_p \\ \llbracket \star(\varphi_1, \dots, \varphi_n) \rrbracket^M &\triangleq f_\star(\llbracket \varphi_1 \rrbracket^M, \dots, \llbracket \varphi_n \rrbracket^M) \end{aligned}$$

where the function f_\star is defined as follows. For all $W_1 \in \mathcal{P}(W^{k_1}), \dots, W_n \in \mathcal{P}(W^{k_n})$, $f_\star(W_1, \dots, W_n) \triangleq \{\bar{w}_{n+1} \in W^k \mid \mathcal{C}^\star(W_1, \dots, W_n, \bar{w}_{n+1})\}$ where $\mathcal{C}^\star(W_1, \dots, W_n, \bar{w}_{n+1})$ is called the *truth condition* of \star and is defined as follows:

- if $\mathbb{A} = \forall$:
“ $\forall \bar{w}_1 \in W^{k_1} \dots \bar{w}_n \in W^{k_n} (\bar{w}_1 \text{ h}_1 W_1 \vee \dots \vee \bar{w}_n \text{ h}_n W_n \vee R_\star^{\pm\sigma} \bar{w}_1 \dots \bar{w}_n \bar{w}_{n+1})$ ”;
- if $\mathbb{A} = \exists$:
“ $\exists \bar{w}_1 \in W^{k_1} \dots \bar{w}_n \in W^{k_n} (\bar{w}_1 \text{ h}_1 W_1 \wedge \dots \wedge \bar{w}_n \text{ h}_n W_n \wedge R_\star^{\pm\sigma} \bar{w}_1 \dots \bar{w}_n \bar{w}_{n+1})$ ”;

where, for all $j \in \llbracket 1; n \rrbracket$, $\bar{w}_j \text{ h}_j W_j \triangleq \begin{cases} \bar{w}_j \in W_j & \text{if } \pm_j = + \\ \bar{w}_j \notin W_j & \text{if } \pm_j = - \end{cases}$ and

$R_\star^{\pm\sigma} \bar{w}_1 \dots \bar{w}_{n+1}$ holds iff $\pm R_\star \bar{w}_{\sigma^-(1)} \dots \bar{w}_{\sigma^-(n+1)}$ holds, with the notations $+R_\star \triangleq R_\star$ and $-R_\star \triangleq W^{k+k_1+\dots+k_n} - R_\star$. If \mathcal{E}_{C} is a class of pointed C–models, the *satisfaction relation* $\Vdash \subseteq \mathcal{E}_{\mathsf{C}} \times \mathcal{L}_{\mathsf{C}}$ is defined as follows: for all $\varphi \in \mathcal{L}_{\mathsf{C}}$ and all $(M, \bar{w}) \in \mathcal{E}_{\mathsf{C}}$, $((M, \bar{w}), \varphi) \in \Vdash$ iff $\bar{w} \in \llbracket \varphi \rrbracket^M$. We usually write $(M, \bar{w}) \Vdash \varphi$ instead of $((M, \bar{w}), \varphi) \in \Vdash$ and we say that φ is *true* in (M, \bar{w}) .

The logic $(\mathcal{L}_{\mathsf{C}}, \mathcal{E}_{\mathsf{C}}, \Vdash)$ is the *atomic logic associated to \mathcal{E}_{C} and C* . The logics of the form $(\mathcal{L}_{\mathsf{C}}, \mathcal{M}_{\mathsf{C}}, \Vdash)$ are called *basic atomic logics*. \dashv

We stress that the \pm sign in $R_\star^{\pm\sigma}$ is the \pm sign in $(\sigma, \pm, \mathbb{A}, (k, k_1, \dots, k_n), (\pm_1, \dots, \pm_n))$.

Permutations of \mathfrak{S}_2	unary signatures
$\tau_1 = (1, 2)$	$t_1 = (\exists, (1, 1), +)$
$\tau_2 = (2, 1)$	$t_2 = (\forall, (1, 1), +)$
	$t_3 = (\forall, (1, 1), -)$
	$t_4 = (\exists, (1, 1), -)$
Permutations of \mathfrak{S}_3	binary signatures
$\sigma_1 = (1, 2, 3)$	$s_1 = (\exists, (1, 1, 1), (+, +))$
$\sigma_2 = (3, 2, 1)$	$s_2 = (\forall, (1, 1, 1), (+, -))$
$\sigma_3 = (3, 1, 2)$	$s_3 = (\forall, (1, 1, 1), (-, +))$
$\sigma_4 = (2, 1, 3)$	$s_4 = (\forall, (1, 1, 1), (+, +))$
$\sigma_5 = (2, 3, 1)$	$s_5 = (\exists, (1, 1, 1), (+, -))$
$\sigma_6 = (1, 3, 2)$	$s_6 = (\exists, (1, 1, 1), (-, +))$
	$s_7 = (\exists, (1, 1, 1), (-, -))$
	$s_8 = (\forall, (1, 1, 1), (-, -))$

FIGURE 1. Permutations of \mathfrak{S}_2 and \mathfrak{S}_3 and ‘families’ of unary and binary signatures

Example 6 (Lambek calculus, modal logic, many-valued logics, multi-dimensional modal logics). The Lambek calculus, where $\mathbb{C} = \{p, \circ, \backslash, /\}$ is defined in Section 2.2, is an example of atomic logic. Here $\circ, \backslash, /$ are the connectives of skeletons $(\sigma_1, +, s_1)$, $(\sigma_5, -, s_3)$, $(\sigma_3, -, s_2)$ (permutations σ_i and signatures s_i are defined in Figure 1). Another example of atomic logic is modal logic where $\mathbb{C} = \{p, \top, \perp, \wedge, \vee, \diamond, \square\}$ is such that

- \top, \perp are connectives of skeletons $(1, +, \exists, 1)$ and $(1, -, \forall, 1)$ respectively;
- $\wedge, \vee, \diamond, \square$ are connectives of skeletons $(\sigma_1, +, s_1)$, $(\sigma_1, -, s_4)$, $(\tau_2, +, t_1)$ and $(\tau_2, -, t_2)$ respectively;
- the \mathbb{C} -models $M = (W, \mathcal{R}) \in \mathcal{E}_{\mathbb{C}}$ are such that $R_{\wedge} = R_{\vee} = \{(w, w, w) \mid w \in W\}$, $R_{\diamond} = R_{\square}$ and $R_{\top} = R_{\perp} = W$.

Indeed, one can easily show that, with these conditions on the \mathbb{C} -models of $\mathcal{E}_{\mathbb{C}}$, we have that for all $(M, w) \in \mathcal{E}_{\mathbb{C}}$, $(M, w) \Vdash \wedge(\varphi, \psi)$ iff $(M, w) \Vdash \varphi$ and $(M, w) \Vdash \psi$, and $(M, w) \Vdash \vee(\varphi, \psi)$ iff $(M, w) \Vdash \varphi$ or $(M, w) \Vdash \psi$. The Boolean conjunction and disjunction \wedge and \vee are defined using the connectives of \mathbb{C} by means of special relations R_{\wedge} and R_{\vee} . Many-valued logics are also examples of atomic logics (see our presentation in Section 2.2.5). Many more examples of atomic connectives are given in Figures 2 and 3. They are in fact just examples of gaggle connectives since all gaggle logics [4, 5] are also atomic logics. They are all of type signature $(1, 1, \dots, 1)$. All the possible truth conditions of unary and binary atomic connectives of this type signature are given in [4, 5]. Multi-dimensional modal logics [29], such as arrow logic or the (temporal) modal logic of intervals, are atomic logics. Their connectives are of type signature (k, k, \dots, k) , for some fixed $k \geq 1$ called the ‘dimension’, and of tonicity signature $(+, \dots, +)$. However, multi-dimensional

Atomic Connective	Truth condition	Non-classical connective in the literature
The existentially positive orbit		
$(\tau_1, +, t_1) \varphi$	$\exists v (v \in \llbracket \varphi \rrbracket \wedge Rvw)$	$\diamond^- \varphi$ [37] \diamond_{\downarrow} [12]
$(\tau_2, -, t_2) \varphi$	$\forall v (v \in \llbracket \varphi \rrbracket \vee -Rvw)$	$\Box \varphi$ [25]
The universally positive orbit		
$(\tau_1, +, t_2) \varphi$	$\forall v (v \in \llbracket \varphi \rrbracket \vee Rvw)$	$+_{\downarrow} \varphi$ [12] [14, p. 401]
$(\tau_2, -, t_1) \varphi$	$\exists v (v \in \llbracket \varphi \rrbracket \wedge -Rvw)$	[12]
The existentially negative orbit		
$(\tau_1, +, t_4) \varphi$	$\exists v (v \notin \llbracket \varphi \rrbracket \wedge Rvw)$	$? \varphi$ [12][14, p. 402] $\exists_1 \varphi$ [12][10, Def. 10.7.7]
$(\tau_2, +, t_4) \varphi$	$\exists v (v \notin \llbracket \varphi \rrbracket \wedge Rvw)$	$?_{\downarrow} \varphi$ [12][15] [14, p. 402] $\exists_2 \varphi$ [10, Def. 10.7.7]
The universally negative orbit		
$(\tau_1, +, t_3) \varphi$	$\forall v (v \notin \llbracket \varphi \rrbracket \vee Rvw)$	φ^{\perp} [12, 13] φ° [22] $\diamond_1 \varphi$ [10, Def. 10.7.2]
$(\tau_2, +, t_3) \varphi$	$\forall v (v \notin \llbracket \varphi \rrbracket \vee Rvw)$	$\sim \varphi$ [21] ${}^{\perp} \varphi$ [12, 13] ${}^{\circ} \varphi$ [22] $\diamond_2 \varphi$ [10, Def. 10.7.2]
The symmetrical existentially positive orbit		
$(\tau_1, -, t_1) \varphi$	$\exists v (v \in \llbracket \varphi \rrbracket \wedge -Rvw)$	[12]
$(\tau_2, +, t_2) \varphi$	$\forall v (v \in \llbracket \varphi \rrbracket \vee Rvw)$	$+ \varphi$ [12] [14, p. 402] φ^* [10, Def. 7.1.19]
The symmetrical universally positive orbit		
$(\tau_1, -, t_2) \varphi$	$\forall v (v \in \llbracket \varphi \rrbracket \vee -Rvw)$	$\Box^- \varphi$ [37] \Box_{\downarrow} [12]
$(\tau_2, +, t_1) \varphi$	$\exists v (v \in \llbracket \varphi \rrbracket \wedge Rvw)$	$\diamond \varphi$ [25]
The symmetrical existentially negative orbit		
$(\tau_1, -, t_4) \varphi$	$\exists v (v \notin \llbracket \varphi \rrbracket \wedge -Rvw)$	$? \varphi$ [12][10, Ex. 1.4.5] φ^{\perp} [22]
$(\tau_2, -, t_4) \varphi$	$\exists v (v \notin \llbracket \varphi \rrbracket \wedge -Rvw)$	$?_{\downarrow} \varphi$ [12] [10, Ex. 1.4.5] ${}^{\perp} \varphi$ [22]
The symmetrical universally negative orbit		
$(\tau_1, -, t_3) \varphi$	$\forall v (v \notin \llbracket \varphi \rrbracket \vee -Rvw)$	[12]
$(\tau_2, -, t_3) \varphi$	$\forall v (v \notin \llbracket \varphi \rrbracket \vee -Rvw)$	$\neg_h \varphi$ [26, 38] $\perp \varphi$ [15]

FIGURE 2. The unary connectives of atomic logics of type (1, 1)

modal logics do not allow connectives of type signatures (k, k_1, \dots, k_n) with different k, k_1, \dots, k_n . ←

Atomic connective	Truth condition	Non-classical con. in the literature
The conjunction orbit		
$\varphi (\sigma_1, +, s_1) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \circ \psi$ [28], $\varphi \otimes_3 \psi$ [3]
$\varphi (\sigma_2, -, s_2) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \notin \llbracket \psi \rrbracket \vee -Rvw)$	
$\varphi (\sigma_3, -, s_2) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \notin \llbracket \psi \rrbracket \vee -Rvw)$	/ [28], $\varphi \subset_2 \psi$ [3]
$\varphi (\sigma_4, +, s_1) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	
$= \psi (\sigma_1, +, s_1) \varphi$		
$\varphi (\sigma_5, -, s_3) \psi$	$\forall vu (v \notin \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	\ [28], $\varphi \supset_1 \psi$ [3]
$= \psi (\sigma_2, -, s_2) \varphi$		
$\varphi (\sigma_6, -, s_3) \psi$	$\forall vu (v \notin \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	
$= \psi (\sigma_3, -, s_2) \varphi$		
The not-but orbit		
$\varphi (\sigma_1, +, s_6) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \succ_3 \psi$ [3]
$\varphi (\sigma_2, +, s_6) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	
$\varphi (\sigma_3, -, s_4) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	$\varphi \oplus_2 \psi$ [3]
$\varphi (\sigma_4, +, s_5) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	
$= \psi (\sigma_1, +, s_6) \varphi$		
$\varphi (\sigma_5, +, s_5) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \prec_1 \psi$ [3]
$= \psi (\sigma_2, +, s_6) \varphi$		
$\varphi (\sigma_6, -, s_4) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	
$= \psi (\sigma_3, -, s_4) \varphi$		
The but-not orbit		
$\varphi (\sigma_1, +, s_5) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \prec_3 \psi$ [3]
$\varphi (\sigma_2, -, s_4) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	
$\varphi (\sigma_3, +, s_6) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \succ_2 \psi$ [3]
$\varphi (\sigma_4, +, s_6) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \in \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \odot \psi$ [23, 31]
$= \psi (\sigma_1, +, s_5) \varphi$		
$\varphi (\sigma_5, -, s_4) \psi$	$\forall vu (v \in \llbracket \varphi \rrbracket \vee u \in \llbracket \psi \rrbracket \vee -Rvw)$	$\varphi \oplus \psi$ [23, 31]
$= \psi (\sigma_2, -, s_4) \varphi$		$\varphi \oplus_1 \psi$ [3]
$\varphi (\sigma_6, +, s_5) \psi$	$\exists vu (v \in \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \oslash \psi$ [23, 31]
$= \psi (\sigma_3, +, s_6) \varphi$		
The stroke orbit		
$\varphi (\sigma_1, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \mid_3 \psi$ [1, 22]
$\varphi (\sigma_2, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	
$\varphi (\sigma_3, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	
$\varphi (\sigma_4, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	
$= \psi (\sigma_1, +, s_7) \varphi$		
$\varphi (\sigma_5, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \mid_1 \psi$ [1, 22]
$= \psi (\sigma_2, +, s_7) \varphi$		
$\varphi (\sigma_6, +, s_7) \psi$	$\exists vu (v \notin \llbracket \varphi \rrbracket \wedge u \notin \llbracket \psi \rrbracket \wedge Rvw)$	$\varphi \mid_2 \psi$ [1, 22]
$= \psi (\sigma_3, +, s_7) \varphi$		

FIGURE 3. Some binary connectives of atomic logics of type (1, 1, 1)

4.3. Molecular Logics

Molecular logics are basically logics whose primitive connectives are compositions of atomic connectives in which it is possible to repeat the same argument at different places in the connective. That is why we call them ‘molecular’, just as molecules are compositions of atoms in chemistry.

Definition 13 (Molecular skeleton and connective). The class \mathbb{C}^* of *molecular skeletons* is the smallest set such that:

- $\mathbb{P} \subseteq \mathbb{C}^*$ and \mathbb{C}^* contains as well, for each $k, l \in \mathbb{N}^*$, a symbol id_k^l of *type signature* (k, k) , *output type* k and *arity* 1;
- for all $\star \in \mathbb{C}$ of type signature (k, k_1^0, \dots, k_n^0) and all $c_1, \dots, c_n \in \mathbb{C}^*$ of output types or types (if they are propositional letters) k_1^0, \dots, k_n^0 respectively, $c \triangleq \star(c_1, \dots, c_n)$ is a molecular skeleton of \mathbb{C}^* of *output type* k .

If $c \in \mathbb{C}^*$, we define its *decomposition tree* as follows. If $c = p \in \mathbb{P}$ or $c = id_k^l$, then its decomposition tree T_c is the tree consisting of a single node labeled with p or id_k^l respectively. If $c = \star(c_1, \dots, c_n) \in \mathbb{C}^*$ then its decomposition tree T_c is the tree defined inductively as follows: the root of T_c is c and it is labeled with \star and one sets edges between that root and the roots c_1, \dots, c_n of the decomposition trees T_{c_1}, \dots, T_{c_n} respectively.

If $c \triangleq \star(c_1, \dots, c_n)$ is a molecular skeleton with output type k and k_1, \dots, k_m are the ks of the different id_k^l s which appear in c_1, \dots, c_n (in an order which follows the first appearance of the id_k^l s in the inorder traversal of the decomposition trees of c_1, \dots, c_n), then the *type signature* of c is (k, k_1, \dots, k_m) and its *arity* is m . We also define the *quantification signature* $\mathbb{A}(c)$ of $c = \star(c_1, \dots, c_n)$ by $\mathbb{A}(c) \triangleq \mathbb{A}(\star)$.

A *molecular connective* is a symbol to which is associated a molecular skeleton. Its arity, type signature, output type, quantification signature and decomposition tree are the same as its skeleton.

The set of *atomic connectives associated to a set C of molecular connectives* is the set of labels different from id_k^l of the decomposition trees of the molecular connectives of \mathbb{C} . –1

Every atomic connective \star of type signature (k, k_1, \dots, k_n) can be seen as the (specific) molecular connectives $c \triangleq \star(id_{k_1}^1, \dots, id_{k_n}^n)$. Note that the same label (atomic connective) may appear several times in a decomposition tree. Note also that the vertices of a decomposition tree are molecular connectives. One needs to introduce the connective id_k^l to deal with molecular connectives whose skeletons are for example of the form $\star(p, id_k^l)$ where $p \in \mathbb{P}$ or molecular connectives in which the same argument(s) appear at different places, like for example in $\star(id_k^1, \dots, id_k^1)$ which is of arity 1.

Example 7 (Modal intuitionistic logic). Let us consider the skeletons $c, c', \star_1, \star_2, \star_3$ defined by the following first-order formulas.

$$\begin{aligned} c(x) &\triangleq \forall y (Rxy \rightarrow \forall z (R_{\diamond}yz \rightarrow Q(z))) \\ c'(x) &\triangleq \forall y (Rxy \rightarrow \exists z (R_{\diamond}yz \wedge Q(z))) \\ \star_1(x) &\triangleq \forall y (Rxy \rightarrow Q(y)) \\ \star_2(x) &\triangleq \forall z (R_{\diamond}xz \rightarrow Q(z)) \\ \star_3(x) &\triangleq \exists z (R_{\diamond}xz \wedge Q(z)) \end{aligned}$$

These first-order formulas can be naturally represented by molecular skeletons. Then, $\star_1, \star_2, \star_3$ are atomic skeletons and the connectives associated to c, c' are molecular connectives. Indeed, c is the composition of \star_1 and \star_2 , $c = \star_1(\star_2)$, and c' is the composition of \star_1 and \star_3 , $c' = \star_1(\star_3)$. Equivalently, c and c' will have the same semantics as $c = \star_1(\star_2(id_1^1))$ and $c' = \star_1(\star_3(id_1^1))$. The connective associated to c corresponds to the connective \square of modal intuitionistic logic and the connective associated to c' corresponds to the connective \diamond of modal intuitionistic logic [35] defined in Section 2.2.3. \dashv

Definition 14 (Molecular language). Let \mathbf{C} be a set of molecular connectives. The (*typed*) *molecular language* $\mathcal{L}_{\mathbf{C}}$ associated to \mathbf{C} is the smallest set that contains the propositional letters and that is closed under the molecular connectives while respecting the type constraints. That is,

- the propositional letters of \mathbf{C} belong to $\mathcal{L}_{\mathbf{C}}$;
- for all $\star \in \mathbf{C}$ of type signature (k, k_1, \dots, k_m) and for all $\varphi_1, \dots, \varphi_m \in \mathcal{L}_{\mathbf{C}}$ of types k_1, \dots, k_m respectively, we have that $\star(\varphi_1, \dots, \varphi_m) \in \mathcal{L}_{\mathbf{C}}$ and $\star(\varphi_1, \dots, \varphi_m)$ is of *type* k .

Elements of $\mathcal{L}_{\mathbf{C}}$ are called *molecular formulas* and are denoted $\varphi, \psi, \alpha, \dots$. The *type of a formula* $\varphi \in \mathcal{L}_{\mathbf{C}}$ is denoted $k(\varphi)$. We use the same abbreviations as for the atomic language. \dashv

Definition 15 (Molecular logic). If \mathbf{C} is a set of molecular connectives, then a \mathbf{C} -*model* M is a \mathbf{C}' -model M where \mathbf{C}' is the set of atomic connectives associated to \mathbf{C} . The truth conditions for molecular connectives are defined naturally from the truth conditions of atomic connectives. We define the *interpretation function of $\mathcal{L}_{\mathbf{C}}$ in M* , denoted $\llbracket \cdot \rrbracket^M : \mathcal{L}_{\mathbf{C}} \rightarrow \bigcup_{k \in \mathbb{N}^*} W^k$, inductively as follows: for all propositional letters $p \in \mathbf{C}$ of skeleton $(\sigma, \pm, \mathbb{A}, k)$, all molecular connectives $\star(c_1, \dots, c_n) \in \mathbf{C}$ of arity $m > 0$ and all $k, l \in \mathbb{N}^*$, for all $\varphi, \varphi_1, \dots, \varphi_m \in \mathcal{L}_{\mathbf{C}}$,

$$\begin{aligned} \llbracket p \rrbracket^M &\triangleq \pm R_p \\ \llbracket id_k^l(\varphi) \rrbracket^M &\triangleq \llbracket \varphi \rrbracket^M \\ \llbracket \star(c_1, \dots, c_n)(\varphi_1, \dots, \varphi_m) \rrbracket^M &\triangleq f_{\star}(\llbracket c_1(\varphi_1^1, \dots, \varphi_{i_1}^1) \rrbracket^M, \dots, \llbracket c_n(\varphi_1^n, \dots, \varphi_{i_n}^n) \rrbracket^M) \end{aligned}$$

where for all $j \in \{1, \dots, n\}$, the formulas $\varphi_1^j, \dots, \varphi_{i_j}^j$ are those $\varphi_1, \dots, \varphi_m$ for which there is a corresponding id_k^l in c_j (the φ_i^j s appear in the same order as their corresponding id_k^l s in c_j).

If \mathcal{E}_C is a class of pointed C-models, the triple $(\mathcal{L}_C, \mathcal{E}_C, \Vdash)$ is a logic called the *molecular logic associated to \mathcal{E}_C and C* . \dashv

As one can easily notice, every atomic logic can be canonically mapped to an equi-expressive molecular logic: each atomic connective \star of type signature (k, k_1, \dots, k_n) of the given atomic logic has to be transformed into the molecular connective of skeleton $\star(id_{k_1}^1, \dots, id_{k_n}^n)$.

Example 8 (Temporal logic). Temporal logic is an example of logic in which one needs to resort both to our types and to combine basic connectives in order to be able to express the truth conditions of its connectives. Let us consider the skeletons defined by the following first-order formulas:

$$\begin{aligned} \star_1(x) &\triangleq \exists yzz' (Py \wedge Qzz' \wedge Ryz z'x) \\ \star'_1(x) &\triangleq \exists yzz' (Py \wedge Qzz' \wedge Rxxz'y) \\ \star_2(x, x') &\triangleq \forall y (Py \vee \neg Syxx') \end{aligned}$$

\star_1 , \star'_1 and \star_2 are atomic skeletons (this is independent from the definitions of R and S). The connectives of skeletons $c = \star_1(id_1^1, \star_2(id_1^2))$ and $c' = \star'_1(id_1^1, \star_2(id_1^2))$ are molecular connectives. Together with some propositional letters, truth constants, Boolean conjunction and disjunction, they form a set of molecular connectives C . If we choose the class of C-models appropriately, the molecular connectives c and c' then correspond to the connectives ‘until’ U and ‘since’ S of temporal logic respectively. Let us be a bit more precise. Let $M = (W, \{<, P\})$ be a temporal model. We represent this temporal model by the C-model $M^{U,S} = (W, \{R, S, P\})$ such that for all $y, z, z', x \in W$,

$$Ryz z'x \text{ iff } x < y, x = z \text{ and } y = z' \quad (3)$$

$$Syxx' \text{ iff } x < y < x'. \quad (4)$$

One can show that for all $\varphi \in \mathcal{L}_{TL}$ and all $w \in W$, $(M, w) \models \varphi$ iff $(M^{U,S}, w) \Vdash T(\varphi)$ where $T: \mathcal{L}_{TL} \rightarrow \mathcal{L}_C$ is defined inductively on the formulas of \mathcal{L}_{TL} in such a way that $T(U(\varphi, \psi)) \triangleq c(T(\varphi), T(\psi))$ and $T(S(\varphi, \psi)) \triangleq c'(T(\varphi), T(\psi))$. Indeed, the standard translation of the until and since operators are:

$$\begin{aligned} ST_x(U(\varphi, \psi)) &= \exists y(x < y \wedge ST_y(\varphi) \wedge \forall z(x < z < y \rightarrow ST_z(\psi))) \\ ST_x(S(\varphi, \psi)) &= \exists y(y < x \wedge ST_y(\varphi) \wedge \forall z(y < z < x \rightarrow ST_z(\psi))) \end{aligned} \quad \dashv$$

4.4. Boolean Connectives

Note that atomic and molecular logics do not include Boolean connectives as primitive connectives. Boolean conjunction and disjunction can be defined in terms of specific atomic connectives as we showed in Example 6. We could in the same manner introduce specific atomic unary connective(s) to define Boolean negation(s) (for each type k). For example, the Boolean negation \neg of type 1 could be defined by one of the atomic skeletons $(\tau_1, +, t_4)$, $(\tau_2, +, t_4)$, $(\tau_1, -, t_3)$, $(\tau_2, -, t_3)$ of Figure 2 interpreted in any model over the identity binary relation $R_\neg = \{(w, w) \mid w \in W\}$.

It turns out that Boolean negation can also be simulated systematically at the level of atomic connectives by applying a transformation on them. The Boolean negation of a formula then boils down to taking the Boolean negation of the outermost connective of the formula. This transformation is defined as follows.

Definition 16 (Boolean negation). Let \star be a n -ary connective of skeleton $(\sigma, \pm, \mathbb{A}, \bar{k}, \pm_1, \dots, \pm_n)$. The *Boolean negation* of \star is the connective $-\star$ of skeleton $(\sigma, -\pm, -\mathbb{A}, \bar{k}, -\pm_1, \dots, -\pm_n)$ where $-\mathbb{A} \triangleq \exists$ if $\mathbb{A} = \forall$ and $-\mathbb{A} \triangleq \forall$ otherwise, which is associated in any \mathcal{C} -model to the same relation as \star . If $\varphi = \star(\varphi_1, \dots, \varphi_n)$ is an atomic formula, the *Boolean negation* of φ is the formula $-\varphi \triangleq -\star(\varphi_1, \dots, \varphi_n)$. \dashv

Proposition 1. Let \mathcal{C} be a set of atomic connectives such that $-\star \in \mathcal{C}$ for all $\star \in \mathcal{C}$. Let $\varphi \in \mathcal{L}_{\mathcal{C}}$ of type k and let $M = (W, \mathcal{R})$ be a \mathcal{C} -model. Then, for all $\bar{w} \in W^k$, $\bar{w} \in \llbracket -\varphi \rrbracket^M$ iff $\bar{w} \notin \llbracket \varphi \rrbracket^M$.

5. Relative Expressivity of Protologics, Atomic, Molecular and First-order Logics

In this section, we are going to investigate the relative expressivity of protologics, atomic, molecular and first-order logics: atomic logics versus FOL in Section 5.1 and protologics versus molecular logics in Section 5.2.

5.1. Atomic Logics versus FOL

Definition 17 (Predicate atomic connectives). Let \mathcal{P} be a set of predicate symbols. The set of *predicate atomic connectives associated to \mathcal{P}* is $\mathcal{C}^{\mathcal{P}} \triangleq \{\perp\} \cup \{\mathbb{R}_{f_l^k} \mid \mathbb{R} \in \mathcal{P} \text{ of arity } k \text{ and } f_l^k : \llbracket 1; k \rrbracket \rightarrow \llbracket 1; l \rrbracket \text{ is surjective with } l \leq k\} \cup \{\llbracket \sigma_k \rrbracket, \parallel_k, \triangleright_k, \forall_k \mid k \in \mathbb{N}^*, \sigma_k \in \mathfrak{S}_k\} \cup \{\forall_0\}$ where

- \perp has skeleton $(\text{Id}, -, \forall, 1)$;
- $\mathbb{R}_{f_l^k}$ has skeleton $(\text{Id}, +, \forall, l)$ for all $k, l \in \mathbb{N}^*$ (such that $l \leq k$);
- $\llbracket \sigma_k \rrbracket$ has skeleton $(\text{Id}, -, \forall, (k, k), +)$, for all $k \in \mathbb{N}^*$;
- \parallel_k has skeleton $(\text{Id}, -, \forall, (k+1, k, 1), (+, +))$;
- \triangleright_k has skeleton $(\text{Id}, -, \forall, (k, k, k), (-, +))$;
- \forall_k has skeleton $((2, 3, 1), -, \forall, (k, 1, k+1), (+, +))$, for all $k \in \mathbb{N}^*$;
- \forall_0 has skeleton $(\text{Id}, -, \forall, (1, 1), +)$. \dashv

Note that all predicate atomic connectives have the quantification signature \forall and that all tonicity signatures are positive $+$, except for the connective \triangleright_k which contains a negative tonicity $-$. This exception is crucial. It somehow encodes the whole Boolean negation. Existential quantification signatures may then indirectly reappear in the formula through a combination of this tonicity $-$ and a connective of universal quantification signature \forall . Moreover, we could have replaced our connective \parallel_k with a connective $\&_k$ whose skeleton would be $(\text{Id}, +, \exists, (k+1, k, 1), (+, +))$ and with the same associated relation, and likewise for other connectives. Doing so, we would obtain the same results.

Remark 2. Instead of the atomic connectives $\|_k$, we could also have taken the molecular connectives $\|_k (id_k^1, \perp)$ as primitive connectives. The former are in fact definable from the latter together with the other connectives $[\sigma_k], \supset_k, \perp$. Yet, in doing so, FOL would be as expressive as a molecular logic instead of a simpler atomic logic.

Definition 18 (Predicate atomic logic). Let \mathcal{P} be a set of predicate symbols. A *predicate $\mathcal{C}^{\mathcal{P}}$ -model* is a $\mathcal{C}^{\mathcal{P}}$ -model $M = (W, \mathcal{R})$ such that:

- for all $l \in \mathbb{N}^*$, the connectives $R_{f_l^k}$ are associated to l -ary relations $R_{f_l^k}$ over W ;
- the connective \perp is associated to the 1-ary relation $R_{\perp} \triangleq W$;
- for all $k \in \mathbb{N}^*$ and all $\sigma_k \in \mathfrak{S}_k$, the connectives $[\sigma_k]$ are associated to the $2k$ -ary relation R^{σ_k} such that $R^{\sigma_k} w_1 \dots w_k w'_1 \dots w'_k$ iff for all $i \in \llbracket 1; k \rrbracket$, $w'_i = w_{\sigma_k(i)}$;
- for all $k \in \mathbb{N}^*$, the connectives $\|_k$ are associated to the $2(k+1)$ -ary relation $R_{k,1} \subseteq W^{2(k+1)}$ such that for all $\bar{w}_1 \in W^k$, all $w_2 \in W$ and all $\bar{w}_3 \in W^{k+1}$, we have that $R_{k,1} \bar{w}_1 w_2 \bar{w}_3$ iff $\bar{w}_3 = (\bar{w}_1, w_2)$;
- for all $k \in \mathbb{N}^*$, the connectives \supset_k are associated to the $3k$ -ary relation $R_k \subseteq W^{3k}$ such that for all $\bar{w}_1 \in W^k$, all $\bar{w}_2 \in W^k$ and all $\bar{w}_3 \in W^k$, we have that $R_k \bar{w}_1 \bar{w}_2 \bar{w}_3$ iff $\bar{w}_1 = \bar{w}_2 = \bar{w}_3$;
- for all $k \in \mathbb{N}^*$, the connectives \forall_k are associated to the $2(k+1)$ -ary relation $R_{k,1}$ as defined for $\|_k$;
- \forall_0 is associated to the 2-ary relation $R_{\forall_0} \triangleq W^2$.

The class of all pointed predicate $\mathcal{C}^{\mathcal{P}}$ -models is denoted $\mathcal{M}_{\mathcal{C}^{\mathcal{P}}}$. The satisfaction relation $\Vdash \subseteq \mathcal{M}_{\mathcal{C}^{\mathcal{P}}} \times \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}$ is then defined following Definition 12. If $\mathcal{L} \subseteq \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}$ and $\mathcal{E} \subseteq \mathcal{M}_{\mathcal{C}^{\mathcal{P}}}$ then the triple $(\mathcal{L}, \mathcal{E}, \Vdash)$ is called the *predicate atomic logic associated to \mathcal{L} and \mathcal{E}* . We also define $\mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k) \triangleq \{\varphi \in \mathcal{L}_{\mathcal{C}^{\mathcal{P}}} \mid k(\varphi) = k\}$. If $\mathcal{L} = \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k)$ and all pointed $\mathcal{C}^{\mathcal{P}}$ -models are of type k then $(\mathcal{L}, \mathcal{E}, \Vdash)$ is a *predicate atomic logic of type k* .

Moreover, for all $(M, \bar{w}), (N, \bar{v}) \in \mathcal{M}_{\mathcal{C}^{\mathcal{P}}}$, we write $(M, \bar{w}) \equiv_k (N, \bar{v})$ when for all $\varphi \in \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k)$ it holds that $(M, \bar{w}) \Vdash \varphi$ iff $(N, \bar{v}) \Vdash \varphi$. \dashv

Definition 19 (Translation from FOL to predicate atomic logics). Let \mathcal{P} be a set of predicate symbols.

Syntax. For all $k \in \mathbb{N}$ and all $\bar{x} = (x_1, \dots, x_k) \in \mathcal{V}^k$, we define the mappings $T_{\bar{x}} : \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k) \rightarrow \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k)$ and $T_{\emptyset} : \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\emptyset) \rightarrow \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(1)$ inductively on the formula $\varphi(x_1, \dots, x_k) \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ (with or without free variables) as follows.

First, for all $\varphi \in \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}$ of type $k_1 \in \mathbb{N}^*$ and all $k_2 \in \mathbb{N}^*$, we define the formula $\|_{k_1, k_2}(\varphi, \perp)$ inductively as follows: $\|_{k_1, 1}(\varphi, \perp) \triangleq \|_{k_1}(\varphi, \perp)$ and $\|_{k_1, k_2+1}(\varphi, \perp) \triangleq \|_{k_1+k_2}(\|_{k_1, k_2}(\varphi, \perp), \perp)$.

- if $\varphi = \perp$ then we define $T_{\emptyset}(\perp) \triangleq \perp$;
- if $\varphi = R y_1 \dots y_l$ and $\{y_1, \dots, y_l\} = \{x_{i_1}, \dots, x_{i_k}\}$ is of cardinality $k \leq l$ (some variables can be the same) with x_{i_1}, \dots, x_{i_k} all distinct and in the same order of appearance as in y_1, \dots, y_l , then there is a unique surjective function $f_l^k : \llbracket 1; l \rrbracket \rightarrow \llbracket 1; k \rrbracket$ such that $(y_1, \dots, y_l) = (x_{i_{f_l^k(1)}}, \dots, x_{i_{f_l^k(l)}})$.

Then, we define the permutation σ_k on $\llbracket 1; k \rrbracket$ by the condition that for all $j \in \llbracket 1; k \rrbracket$, $i_{\sigma_k(j)} \triangleq j$. Finally, we define

$$T_{\bar{x}}(\mathbf{R}y_1 \dots y_l) \triangleq [\sigma_k] \mathbf{R}_{f_l^k};$$

- if $\varphi = (\varphi_1 \rightarrow \varphi_2)$ where φ_1 and φ_2 are both sentences, then we define

$$T_{\emptyset}(\varphi) \triangleq \supset_1 (T_{\emptyset}(\varphi_1), T_{\emptyset}(\varphi_2))$$

if $\varphi = (\varphi_1 \rightarrow \varphi_2(x))$ where φ_1 is a sentence, then we define

$$T_{(x)}(\varphi) \triangleq \supset_1 (T_{\emptyset}(\varphi_1), T_{(x)}(\varphi_2(x)))$$

if $\varphi = (\varphi_1(x) \rightarrow \varphi_2)$ where φ_2 is a sentence, then we define

$$T_{(x)}(\varphi) \triangleq \supset_1 (T_{(x)}(\varphi_1(x)), T_{\emptyset}(\varphi_2))$$

if $\varphi = (\varphi_1 \rightarrow \varphi_2(x_1, \dots, x_k))$ where φ_1 is a sentence and $k > 1$, then we define

$$T_{\bar{x}}(\varphi) \triangleq \supset_k (\|_{1, k-1} (T_{\emptyset}(\varphi_1), \perp), T_{(x_1, \dots, x_k)}(\varphi_2(x_1, \dots, x_k)))$$

if $\varphi = (\varphi_1(x_1, \dots, x_k) \rightarrow \varphi_2)$ where φ_2 is a sentence and $k > 1$, then we define

$$T_{\bar{x}}(\varphi) \triangleq \supset_k (T_{(x_1, \dots, x_k)}(\varphi_1(x_1, \dots, x_k)), \|_{1, k-1} (T_{\emptyset}(\varphi_2), \perp));$$

if $\varphi = (\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}) \rightarrow \varphi_2(y_{j_1}, \dots, y_{j_{k_2}}))$ then we define

$$T_{\bar{x}}(\varphi) \triangleq \supset_k (\text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})), \text{Exp}_{\bar{x}}(\varphi_2(x_{j_1}, \dots, x_{j_{k_2}})))$$

where

$$\text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})) \triangleq \begin{cases} T_{(x_1, \dots, x_k)}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})) \\ \text{if } k_1 = k \\ [\sigma_k] \|_{k_1, k-k_1} (T_{(x_{i_1}, \dots, x_{i_{k_1}})}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})), \perp) \\ \text{if } k_1 < k \end{cases}$$

where σ_k is a permutation defined as follows. First, we complete the indices i_1, \dots, i_{k_1} with indices i_{k_1+1}, \dots, i_k such that $\{x_{i_1}, \dots, x_{i_{k_1}}, x_{i_{k_1+1}}, \dots, x_{i_k}\} = \{x_1, \dots, x_k\}$. Second, we define the permutation σ_k such that for all $j \in \llbracket 1; k \rrbracket$, $i_{\sigma_k(j)} = j$.

The definition is similar for $\text{Exp}_{\bar{x}}(\varphi_2(x_{j_1}, \dots, x_{j_{k_2}}))$, one only needs to replace i by j and k_1 by k_2 .

- if $\varphi = \forall x \psi(x)$ where φ is a sentence, then we define

$$T_{\emptyset}(\varphi) \triangleq \forall_0 T_{(x)}(\psi(x));$$

if $\varphi = \forall x \psi(x_1, \dots, x_k, x)$ with $k \geq 1$, then we define

$$T_{\bar{x}}(\varphi) \triangleq \forall_k (\perp, T_{(x_1, \dots, x_k, x)}(\psi(x_1, \dots, x_k, x))).$$

Semantics. Let $M = (W, \{R_1, \dots, R_n, \dots, c_1, \dots, c_n, \dots\})$ be a structure without functions. We define the $\mathcal{C}^{\mathcal{P}}$ -model $T(M) = (W, \mathcal{R})$ as follows:

- the k -ary relation $R_{f_l^k}$ is defined from the l -ary relation R of M associated to each $R \in \mathcal{P}$ as follows: for all $w_1, \dots, w_k \in W$, we have that $R_{f_l^k} w_1 \dots w_k$ iff $R w_{f_l^k(1)} \dots w_{f_l^k(l)}$;
- the other relations of \mathcal{R} are defined like in Definition 18.

If $\bar{x} = (x_1, \dots, x_k)$ is a tuple of k variables then we define $T_{\bar{x}}(M, s) \triangleq (T(M), (s(x_1), \dots, s(x_k)))$ and $T_{\emptyset}(M, s) \triangleq (T(M), (s(x_0)))$ for an arbitrary $x_0 \in \mathcal{V}$. -4

In the syntactic part, the definitions may leave some freedom concerning the exact determination of the permutations σ_k for the implication case (more

precisely in the choice of indices for the remaining variables). This is not problematic and does not impact the results as the proof of the next lemma shows. Permutations are introduced so that the same variables which appear in different places in a formula be evaluated at the same points in the domain of the structure. Moreover, our definitions are set in such a way that for any sentence φ , we will have that $T_{\emptyset}(\varphi)$ will always be a formula of type 1.

Example 9. We provide two examples of translations.

- If P is a 8-ary predicate, then $T_{(x_1, x_2, x_3)}(P x_3 x_3 x_1 x_1 x_2 x_1 x_2 x_2) = [\sigma_3]P_f$ where $f(1) = 1, f(2) = 1, f(3) = 2, f(4) = 2, f(5) = 3, f(6) = 2, f(7) = 3, f(8) = 3$ and $\sigma_3 = (2, 3, 1)$ because here $(x_{i_1}, x_{i_2}, x_{i_3}) = (x_3, x_1, x_2)$.
- $T_{(x, y)}(\forall z (Pzx \rightarrow Ryzx) \rightarrow \forall z Qzz)$

$$\begin{aligned} &= \supset_2 (T_{(x, y)}(\forall z (Pzx \rightarrow Ryzx)), \|_1 (T_{\emptyset}(\forall z Qzz), \perp)) \\ &= \supset_2 (\forall_2 (\perp, T_{(x, y, z)}(Pzx \rightarrow Ryzx)), \|_1 (\forall_0 T_{(z)}(Qzz), \perp)) \\ &= \supset_2 (\forall_2 (\perp, \supset_3 ([[(1, 3, 2)] \|_2 (T_{(x, z)}(Pzx), \perp), T_{(x, y, z)}(Ryzx)])), \\ &\quad \|_1 (\forall_0 T_{(z)}(Qzz), \perp)) \\ &= \supset_2 (\forall_2 (\perp, \supset_3 ([[(1, 3, 2)] \|_2 (([(2, 1)]P, \perp), [(3, 1, 2)]R)], \\ &\quad \|_1 (\forall_0 [\text{Id}]Q_f, \perp))) \\ &\text{where } f(1) = 1 \text{ and } f(2) = 1. \end{aligned}$$

Lemma 1. Let $\varphi \in \mathcal{L}_{\text{FOL}}^P$, let \bar{x} be the tuple of free variables of φ (possibly empty) and let (M, s) be a pointed structure. Then, we have that

$$(M, s) \models \varphi \text{ iff } T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi).$$

Proof: By induction on φ . Let $\bar{x} = (x_1, \dots, x_k) \in \mathcal{V}^k$ and let (M, s) be a pointed structure of domain W .

- If $\varphi = \perp$ then the result trivially holds.
- If $\varphi = R y_1 \dots y_l$ and $\{y_1, \dots, y_l\} = \{x_{i_1}, \dots, x_{i_k}\}$ is of cardinality $k \leq l$ with x_{i_1}, \dots, x_{i_k} all distinct and in the same order of appearance as in y_1, \dots, y_l . Let $f_l^k : \llbracket 1; l \rrbracket \rightarrow \llbracket 1; k \rrbracket$ be the unique surjective function such that $(y_1, \dots, y_l) = (x_{i_{f_l^k(1)}}, \dots, x_{i_{f_l^k(l)}})$. Then,

$$\begin{aligned} &T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi) \\ &\text{iff } (T(M), (s(x_1), \dots, s(x_k))) \Vdash [\sigma_k]R_{f_l^k} x_{i_1} \dots x_{i_k} \\ &\text{iff } (T(M), (s(x_{i_1}), \dots, s(x_{i_k}))) \Vdash R_{f_l^k} x_{i_1} \dots x_{i_k} \\ &\text{iff } R_{f_l^k} s(x_{i_1}) \dots s(x_{i_k}) \text{ holds by definition} \\ &\text{iff } R s(x_{i_{f_l^k(1)}}) \dots s(x_{i_{f_l^k(l)}}) \text{ holds by definition} \\ &\text{iff } R s(y_1) \dots s(y_l) \text{ holds by definition of } f_l^k \\ &\text{iff } (M, s) \models R y_1 \dots y_l \\ &\text{iff } (M, s) \models \varphi. \end{aligned}$$

- If $\varphi = \varphi_1 \rightarrow \varphi_2(x)$ then
$$\begin{aligned} &T_{(x)}(M, s) \Vdash T_{(x)}(\varphi) \\ &\text{iff } T_{(x)}(M, s) \Vdash \supset_1 (T_{\emptyset}(\varphi_1), T_{(x)}(\varphi_2(x))) \text{ by definition} \\ &\text{iff } T_{(x)}(M, s) \Vdash T_{\emptyset}(\varphi_1) \text{ implies } T_{(x)}(M, s) \Vdash T_{(x)}(\varphi_2(x)) \\ &\text{by definition of } \supset_1 \end{aligned}$$

iff $T_{\emptyset}(M, s) \Vdash T_{\emptyset}(\varphi_1)$ implies $T_{(x)}(M, s) \Vdash T_{(x)}(\varphi_2(x))$
 because the truth value of $T_{\emptyset}(\varphi_1)$ is the same at all states
 iff $(M, s) \models \varphi_1$ implies $(M, s) \models \varphi_2$, by Induction Hypothesis
 iff $(M, s) \models \varphi$.

- If $\varphi = \varphi_1 \rightarrow \varphi_2(x_1, \dots, x_k)$ with $k > 1$ then
 - $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi)$
 - iff $T_{\bar{x}}(M, s) \Vdash \supset_k (\|_{1, k-1} (T_{\emptyset}(\varphi_1), \perp), T_{\bar{x}}(\varphi_2(\bar{x})))$ by definition
 - iff $T_{\bar{x}}(M, s) \models \|_{1, k-1} (T_{\emptyset}(\varphi_1), \perp)$ implies $T_{\bar{x}}(M, s) \models T_{\bar{x}}(\varphi_2(\bar{x}))$
 - by definition of \supset_k
 - iff $(T(M), (s(x_1), \dots, s(x_k))) \Vdash \|_{1, k-1} (T_{\emptyset}(\varphi_1), \perp)$ implies
 $(M, s) \models \varphi_2$, by definition of $T_{\bar{x}}$ and Induction Hypothesis
 - iff $((T(M), s(x_1)) \Vdash T_{\emptyset}(\varphi_1))$ implies $(M, s) \models \varphi_2$, by definition
of $\|_{1, k-1}$
 - iff $(T(M), s(x_1)) \Vdash T_{\emptyset}(\varphi_1)$ implies $(M, s) \models \varphi_2$
 - iff $T_{\emptyset}(M, s) \Vdash T_{\emptyset}(\varphi_1)$ implies $(M, s) \models \varphi_2$
 - iff $(M, s) \models \varphi_1$ implies $(M, s) \models \varphi_2$, by Induction Hypothesis
 - iff $(M, s) \models \varphi$.
- If $\varphi = \varphi_1(x_{i_1}, \dots, x_{i_{k_1}}) \rightarrow \varphi_2(y_{j_1}, \dots, y_{j_{k_2}})$ then
 - if $k_1 = k$ then $\{x_{i_1}, \dots, x_{i_{k_1}}\} = \{x_1, \dots, x_k\}$ and in that case
 - $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_k}))$
 - iff $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi_1)$
 - iff $(M, s) \models \varphi_1$ by Induction Hypothesis.
 - if $k_1 < k$ then
 - $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}))$
 - iff $(T(M), (s(x_1), \dots, s(x_k))) \Vdash$
 $[\sigma_k] \|_{k_1, k-k_1} (T_{(x_{i_1}, \dots, x_{i_{k_1}})}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})), \perp)$ by definition
 - iff $(T(M), (s(x_{i_1}), \dots, s(x_{i_{k_1}}), s(x_{\sigma^-(k_1+1)}), \dots, s(x_{\sigma^-(k)}))) \Vdash$
 $\|_{k_1, k-k_1} (T_{(x_{i_1}, \dots, x_{i_{k_1}})}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})), \perp)$, by definition of $[\sigma_k]$
 - because for all $j \in \llbracket 1; k \rrbracket$, since $i_{\sigma(j)} = j$, we have $i_j = \sigma^-(j)$
 - iff $(T(M), (s(x_{i_1}), \dots, s(x_{i_{k_1}}))) \Vdash T_{(x_{i_1}, \dots, x_{i_{k_1}})}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}))$
 - by definition of $\|_{k_1, k-k_1}$
 - iff $(T(M), (s(x_{i_1}), \dots, s(x_{i_{k_1}}))) \Vdash T_{(x_{i_1}, \dots, x_{i_{k_1}})}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}))$
 - iff $(M, s) \models \varphi_1$ by Induction Hypothesis

So, we have proved that

$(M, s) \models \varphi_1(x_{i_1}, \dots, x_{i_{k_1}})$ iff $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}))$ (*).

Likewise, we can prove that

$(M, s) \models \varphi_2(x_{j_1}, \dots, x_{j_{k_2}})$ iff $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_2(x_{j_1}, \dots, x_{j_{k_2}}))$ (**).

Then, we have that

$T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi)$
 iff $T_{\bar{x}}(M, s) \Vdash \supset_k (\text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}})), \text{Exp}_{\bar{x}}(\varphi_2(x_{j_1}, \dots, x_{j_{k_2}})))$
 iff $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_1(x_{i_1}, \dots, x_{i_{k_1}}))$ implies
 $T_{\bar{x}}(M, s) \Vdash \text{Exp}_{\bar{x}}(\varphi_2(x_{j_1}, \dots, x_{j_{k_2}}))$, by definition of \supset_k
 iff $(M, s) \models \varphi_1$ implies $(M, s) \models \varphi_2$ by (*) and (**)
 iff $(M, s) \models \varphi$.

- If $\varphi = \forall x\psi(x)$ then
 - $T_{\emptyset}(M, s) \Vdash T_{\emptyset}(\varphi)$
 - iff $(T(M), s(x_0)) \Vdash T_{\emptyset}(\varphi)$
 - iff $(T(M), s(x_0)) \Vdash \forall_0 T_{(x)}(\psi(x))$, by definition of T_{\emptyset} for an arbitrary $x_0 \in \mathcal{V}$
 - iff $(T(M), w) \Vdash T_{(x)}(\psi(x))$ for all $w \in M$, by definition of \forall_0
 - iff $T_{(x)}(M, s[x := w]) \Vdash T_{(x)}(\psi(x))$ for all $w \in M$, by definition of $T_{(x)}$
 - iff $(M, s[x := w]) \models \psi(x)$ for all $w \in M$, by Induction Hypothesis
 - iff $(M, s) \models \varphi$.
- If $\varphi = \forall x\psi(x_1, \dots, x_k, x)$ then
 - $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi)$
 - iff $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\forall x\psi(x_1, \dots, x_k, x))$
 - iff $T_{\bar{x}}(M, s) \Vdash \forall_k(\perp, T_{(x_1, \dots, x_k, x)}(\psi(x_1, \dots, x_k, x)))$ by definition
 - iff $(T(M), (s(x_1), \dots, s(x_k))) \Vdash \forall_k(\perp, T_{(x_1, \dots, x_k, x)}(\psi(x_1, \dots, x_k, x)))$ by definition of $T_{\bar{x}}$
 - iff $\forall w \in W \forall (w_1, \dots, w_{k+1}) \in W^{k+1} (w \in \llbracket \perp \rrbracket_{T(M)} \vee (w_1, \dots, w_{k+1}) \in \llbracket T_{(\bar{x}, x)}(\psi) \rrbracket_{T(M)} \vee -R_{k,1}^{\tau} w(w_1, \dots, w_{k+1})(s(x_1), \dots, s(x_k)))$ where τ is the permutation $(2, 3, 1)$
 - iff $\forall w \in W \forall (w_1, \dots, w_{k+1}) \in W^{k+1} ((w_1, \dots, w_{k+1}) \in \llbracket T_{(\bar{x}, x)}(\psi) \rrbracket_{T(M)} \vee -R_{k,1}(s(x_1), \dots, s(x_k))w(w_1, \dots, w_{k+1}))$
 - iff $\forall w \in W \forall (w_1, \dots, w_{k+1}) \in W^{k+1}$, if $(w_1, \dots, w_k) = (s(x_1), \dots, s(x_k))$ and $w_{k+1} = w$ then $(w_1, \dots, w_{k+1}) \in \llbracket T_{(\bar{x}, x)}(\psi) \rrbracket_{T(M)}$
 - iff $\forall w \in W, (s(x_1), \dots, s(x_k), w) \in \llbracket T_{(\bar{x}, x)}(\psi) \rrbracket_{T(M)}$
 - iff $(T(M), (s(x_1), \dots, s(x_k), w)) \Vdash T_{(\bar{x}, x)}(\psi)$ for all $w \in M$ by definition
 - iff $T_{(\bar{x}, x)}(M, s[x := w]) \Vdash T_{(\bar{x}, x)}(\psi)$ for all $w \in M$ by definition of $T_{(\bar{x}, x)}$
 - iff $(M, s[x := w]) \models \psi(x_1, \dots, x_k, x)$ for all $w \in M$, by Induction Hypothesis
 - iff $(M, s) \models \forall x\psi(x_1, \dots, x_k, x)$ by definition
 - iff $(M, s) \models \varphi$.

The cases $\varphi = \varphi_1(x) \rightarrow \varphi_2$ and $\varphi = \varphi_1(x_1, \dots, x_k) \rightarrow \varphi_2$ with $k > 1$ are dealt with like the cases $\varphi = \varphi_1 \rightarrow \varphi_2(x)$ and $\varphi = \varphi_1 \rightarrow \varphi_2(x_1, \dots, x_k)$. \square

Definition 20 (Translation from predicate atomic logics to FOL).

Syntax. For all $k \in \mathbb{N}^*$ and all tuples $\bar{x} = (x_1, \dots, x_k)$ of variables and constants, we define the mappings $ST_{\bar{x}}^k : \mathcal{L}_{\mathcal{C}\mathcal{P}}^k \rightarrow \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$, where $\mathcal{L}_{\mathcal{C}\mathcal{P}}^k$ is the set of formulas of $\mathcal{L}_{\mathcal{C}\mathcal{P}}$ of type k , inductively as follows:

$$\begin{aligned}
ST_{\bar{x}}(R_{f_l^k}) &\triangleq R_{x_{f_l^k(1)} \dots x_{f_l^k(l)}} \\
ST_{\bar{x}}(\perp) &\triangleq \perp \\
ST_{\bar{x}}([\sigma_k]\varphi) &\triangleq ST_{(x_{\sigma_k^-(1)}, \dots, x_{\sigma_k^-(k)})}(\varphi) \\
ST_{\bar{x}}(\|_k(\varphi_1, \varphi_2)) &\triangleq (ST_{\bar{y}}(\varphi_1) \vee ST_{(x)}(\varphi_2)) \quad \text{if } \bar{x} = (\bar{y}, x) \\
ST_{\bar{x}}(\supset_k(\varphi_1, \varphi_2)) &\triangleq (ST_{\bar{x}}(\varphi_1) \rightarrow ST_{\bar{x}}(\varphi_2)) \\
ST_{\bar{x}}(\forall_k(\varphi_1, \varphi_2)) &\triangleq \forall x(ST_{(x)}(\varphi_1) \vee ST_{(\bar{x}, x)}(\varphi_2)) \\
ST_{(x)}(\forall_0\varphi) &\triangleq \forall xST_{(x)}(\varphi)
\end{aligned}$$

where t is an arbitrary term of $\mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ (we recall that \vee is here an abbreviation). *Semantics.* Let $(M, (w_1, \dots, w_k))$ be a pointed $\mathcal{C}^{\mathcal{P}}$ -model of type k and let $\bar{x} = (x_1, \dots, x_k)$ be a tuple of free variables or constants of size k . The (pointed) structure associated to (M, \bar{w}) , denoted $ST_{\bar{x}}(M, \bar{w}) \triangleq (ST(M), s_{\bar{x}}^{\bar{w}})$, is defined as follows. The assignment $s_{\bar{x}}^{\bar{w}}$ is such that for all $i \in \{1, \dots, k\}$, $s(x_i) = w_i$ and for all $x \in \mathcal{V} - \{x_1, \dots, x_k\}$, $s(x) = w_1$ and $ST(M)$ is the structure $ST(M) = (W, ST(\mathcal{R}))$ where $ST(\mathcal{R})$ is the set \mathcal{R} to which we remove the relations of the form R^{σ_k} , R_k , $R_{k,1}$ and R_{\forall_0} and replace the relations $R_{f_l^k}$ with the l -ary relations R associated to each $R \in \mathcal{P}$, which are defined as follows: for all $w_1, \dots, w_l \in W$, we have that $\{w_1, \dots, w_l\} = \{v_1, \dots, v_k\}$ for some $k \leq l$ with v_1, \dots, v_k all distinct and in the same order as w_1, \dots, w_l . There is a unique surjective function $f_l^k : \llbracket 1; l \rrbracket \rightarrow \llbracket 1; k \rrbracket$ such that $(w_1, \dots, w_l) = (v_{f_l^k(1)}, \dots, v_{f_l^k(l)})$. Then, we set $Rw_1 \dots w_l$ iff $R_{f_l^k}v_1 \dots v_k$. \dashv

Lemma 2. *Let (M, \bar{w}) be a pointed predicate $\mathcal{C}^{\mathcal{P}}$ -model of type k , let $\varphi \in \mathcal{L}_{\mathcal{C}^{\mathcal{P}}}$ of type k and let $\bar{x} \in \mathcal{V}^k$. Then,*

$$(M, \bar{w}) \Vdash \varphi \text{ iff } ST_{\bar{x}}(M, \bar{w}) \models ST_{\bar{x}}(\varphi).$$

Moreover, for all pointed structures (M, s) without distinguished elements, we have that $ST_{\bar{x}}(T_{\bar{x}}(M, s)) \equiv_{\bar{x}} (M, s)$ and for all pointed predicate $\mathcal{C}^{\mathcal{P}}$ -model (M, \bar{w}) , we have that $T_{\bar{x}}(ST_{\bar{x}}(M, \bar{w})) \equiv_k (M, \bar{w})$.

Proof: The first part of the proof is by induction on φ . The only difficult case is for the propositional letters, we only deal with it. Let $\bar{x} = (x_1, \dots, x_k) \in \mathcal{V}^k$. If $\bar{w} = (w_1, \dots, w_k)$ then

$$\begin{aligned}
(M, \bar{w}) \Vdash R_{f_l^k} & \\
\text{iff } R_{f_l^k}w_1 \dots w_k & \\
\text{iff } Rw_{f_l^k(1)} \dots w_{f_l^k(l)} & \\
\text{iff } (ST(M), s_{\bar{x}}^{\bar{w}}) \models R_{x_{f_l^k(1)} \dots x_{f_l^k(l)}} & \text{ because } (s(x_1), \dots, s(x_k)) = \\
(w_1, \dots, w_k) & \\
\text{iff } (ST(M), s_{\bar{x}}^{\bar{w}}) \models ST_{\bar{x}}(R_{f_l^k}) & \\
\text{iff } ST_{\bar{x}}(M, \bar{w}) \models ST_{\bar{x}}(R_{f_l^k}). &
\end{aligned}$$

Now, we prove the second part of the lemma. First, note that $ST(T(M)) = M$ (some relations are added and then removed). So, if $\bar{w} = (s(x_1), \dots, s(x_k))$ then $(M, s_{\bar{x}}^{\bar{w}}) = ST_{\bar{x}}(T_{\bar{x}}(M, s))$. Then, we have that $(M, s_{\bar{x}}^{\bar{w}}) \equiv_{\bar{x}} (M, s)$ because $\bar{w} = (s(x_1), \dots, s(x_k))$, that is, $ST_{\bar{x}}(T_{\bar{x}}(M, s)) \equiv_{\bar{x}} (M, s)$. Second, note that $T(ST(M)) = M$. Moreover, if $\bar{w} = (w_1, \dots, w_k)$ then $(s_{\bar{x}}^{\bar{w}}(x_1), \dots, s_{\bar{x}}^{\bar{w}}(x_k))$

$= (w_1, \dots, w_k)$. Therefore, $T_{\bar{x}}(ST_{\bar{x}}(M, \bar{w})) = (M, \bar{w})$, so the result. \square

Theorem 1. *Every pure predicate logic with $k \geq 0$ variables and constants is as expressive as a predicate atomic logic. The class of pure predicate logics with $k \geq 2$ free variables and constants is as expressive as the class of predicate atomic logics of type k .*

Proof: Let $\bar{x} \in \mathcal{V}^k$ be a tuple of variables of size $k \geq 0$. If $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k), \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ is a pure predicate logic with free variables and constants \bar{x} , then it is as expressive as $(T_{\bar{x}}(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)), T_{\bar{x}}(\mathcal{E}_{\text{FOL}}), \Vdash)$, where $T_{\bar{x}}(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)) \triangleq \{T_{\bar{x}}(\varphi) \mid \varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)\}$ and $T_{\bar{x}}(\mathcal{E}_{\text{FOL}}) \triangleq \{T_{\bar{x}}(M, s) \mid (M, s) \in \mathcal{E}_{\text{FOL}}\}$. The translations are, on the one hand, $T_{\bar{x}}$ (for formulas) and $T_{\bar{x}}$ (for models) and, on the other hand, $ST_{\bar{x}}$ (for formulas) and $ST_{\bar{x}}^t$ (for models) where $ST_{\bar{x}}^t$ is defined as follows: $ST_{\bar{x}}^t(T_{\bar{x}}(M, s))$ is a fixed and chosen pointed structure (M, s') such that $T_{\bar{x}}(M, s) = T_{\bar{x}}(M, s')$. Then, we have that $ST_{\bar{x}}^t(T_{\bar{x}}(M, s)) \equiv_{\bar{x}} (M, s)$ (*) and $T_{\bar{x}}(M, s) = T_{\bar{x}}(ST_{\bar{x}}^t(T_{\bar{x}}(M, s)))$ by Lemma 2 (so $T_{\bar{x}}(M, s) \equiv_k T_{\bar{x}}(ST_{\bar{x}}^t(T_{\bar{x}}(M, s)))$). Moreover, we also have that $(M, s) \models \varphi$ iff $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi)$ by Lemma 1. Likewise, $T_{\bar{x}}(M, s) \Vdash T_{\bar{x}}(\varphi)$ iff $ST_{\bar{x}}^t(T_{\bar{x}}(M, s)) \models ST_{\bar{x}}(T_{\bar{x}}(\varphi))$ by Lemma 2, iff $(M, s) \models ST_{\bar{x}}(T_{\bar{x}}(\varphi))$ again by Lemma 2, iff $ST_{\bar{x}}^t(T_{\bar{x}}(M, s)) \models ST_{\bar{x}}(T_{\bar{x}}(\varphi))$ by (*). So, we have proved that $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k), \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ is as expressive as $(T_{\bar{x}}(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k)), T_{\bar{x}}(\mathcal{E}_{\text{FOL}}), \Vdash)$.

For the second part of the theorem, the proof that every pure predicate logic is as expressive as a predicate atomic logic of type k is the same as in the first part. Conversely, if $(\mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k), \mathcal{E}_{\mathcal{C}^{\mathcal{P}}}(k), \Vdash)$ is a predicate atomic logic of type $k \geq 2$, then $(\mathcal{L}_{\mathcal{C}^{\mathcal{P}}}(k), \mathcal{E}_{\mathcal{C}^{\mathcal{P}}}(k), \Vdash)$ is as expressive as $(\mathcal{L}_{\text{FOL}}^{\mathcal{P}}(\bar{x}, k), \mathcal{E}_{\text{FOL}}, \models_{\text{FOL}})$ where this time $\mathcal{E}_{\text{FOL}} \triangleq \{ST_{\bar{x}}(M, \bar{w}) \mid (M, \bar{w}) \in \mathcal{E}_{\mathcal{C}^{\mathcal{P}}}(k)\}$ for some fixed and chosen tuple of variables $\bar{x} \in \mathcal{V}^k$. In that case, we use the plain translations $T_{\bar{x}}$ and $ST_{\bar{x}}$ and Lemmas 1 and 2 again. There is no ambiguity on the translations (whether we deal with sentences or formulas with a single free variable) because $k \geq 2$. \square

Since pure predicate logic can be split up into fragments with $k \geq 0$ free variables or constants, we could in principle embed pure predicate logic into a predicate atomic logic (with $\mathcal{C}^{\mathcal{P}}$ -models of different types). In fact, we can define a notion of equi-expressivity taking into account this possible splitting into sublogics that would allow us to prove formally that the class of pure predicate logics is as expressive as a specific class of predicate atomic logics. That notion would be the following: a logic $\mathsf{L}_1 = (\mathcal{L}_1, \mathcal{E}_1, \models_1)$ would be *modularly as expressive as* a logic $\mathsf{L}_2 = (\mathcal{L}_2, \mathcal{E}_2, \models_2)$ when it would be possible to split up $\mathcal{L}_1 \times \mathcal{E}_1$ and $\mathcal{L}_2 \times \mathcal{E}_2$ into partitions $\mathcal{L}_1 \times \mathcal{E}_1 = \bigsqcup_{i \in I} (\mathcal{L}_1^i \times \mathcal{E}_1^i)$ and $\mathcal{L}_2 \times \mathcal{E}_2 = \bigsqcup_{i \in J} (\mathcal{L}_2^j \times \mathcal{E}_2^j)$ such that for each $i \in I$ there is $j \in J$ such that $(\mathcal{L}_1^i, \mathcal{E}_1^i, \models_1)$ is as expressive as $(\mathcal{L}_2^j, \mathcal{E}_2^j, \models_2)$, and vice versa, and at least one element of each partition is infinite (so that the partitions are not trivial

partitions into an infinite number of singletons or finite sets of pairs of models and formulas, because in that case it would always be possible to define translation mappings). However, this notion of modular equi-expressivity is not transitive in general. This said, this modular equi-expressivity result supports our intuition that the class of pure predicate logics is as expressive as some class of predicate atomic logics.

Corollary 1. *Let $\varphi \in \mathcal{L}_{\text{FOL}}^{\mathcal{P}}$ and let \bar{x} be the tuple of $k \geq 0$ free variables of φ . Then, φ is valid (on the class of all pointed structures) if, and only if, $T_{\bar{x}}(\varphi)$ is valid on the class of all pointed predicate $\mathcal{C}^{\mathcal{P}}$ -models of type k (of type 1 if $k = 0$). Moreover, $\varphi \equiv ST_{\bar{x}}(T_{\bar{x}}(\varphi))$.*

Proof: It follows from the first part of Theorem 1 and the expressions that follow the definition of equal expressivity in Section 3. \square

Modalizing first-order logic. There have already been some attempts to ‘modalize’ first-order logics; we briefly mention two of them here. Marx & Venema obtain a result [29, Proposition 5.2.5] similar to our Corollary 1 for multi-dimensional modal logics, but with respect to some restricted first-order logics. Like here, they “wanted to identify assignments of the first-order semantics with tuple-states in the modal framework” [29, p. 154]. However, because multi-dimensional logics only allow atomic connectives of type signature (k, k, \dots, k) for a fixed $k > 0$ (called the ‘dimension’), their results hold with respect to fragments of first-order logics. They show for example that their k -dimensional cylindric (multi-dimensional) modal logic corresponds to a restricted first-order logic where the predicate symbols are all k -ary (and with k variables).

Kuhn [27] introduces a multi-typed S5 modal logic PREDBOX and obtains an equal expressivity theorem [27, p. 152] between PREDBOX and first-order logic whose formulation is very close to our Lemmas 1 and 2. The formulas of PREDBOX are also typed (sorted) but the tuples of states on which they are interpreted do not always correspond to the denotations of some free variables. Like [29], the translation in the modal framework of the first-order quantification is different from here, it is expressed by an S5 modality. Moreover, unlike the models of atomic logics, those of PREDBOX include *infinite* tuples of states.

5.2. Molecular Logics versus Protologics

In this section, \mathcal{Q} and \mathcal{P} are sets of predicates such that $\mathcal{Q} \subseteq \mathcal{P}$. We will also use the equality predicate $=$ so as to avoid formulas of the form $Qxyx$. That is, we want all variables in the scope of a predicate to be different. For example, $Q_i xyx$ will be translated into the logically equivalent formula $\forall z(x = z \rightarrow Q_i xyz)$. This is because we do not want to change the predicate symbols P into $P_{f_i^k}$ like in Definition 18. Note that this preprocessing with the equality predicate could also be applied in the translation from first-order formulas to predicate atomic logics in order to avoid the complication with the introduction of $P_{f_i^k}$ in Definition 18. These two approaches are in fact

equivalent in that previous case, but we refrained from doing so because we did not want to introduce the equality predicate in the predicate atomic logic.

Definition 21 (Translation from protologics to molecular logics). Let C^a be a set of abstract connectives.

Syntax. We define the mapping t from the abstract connectives of C^a to molecular connectives as follows:

- $t(Q) = Q$ for all predicate $Q \in \mathcal{Q}$;
- For abstract connectives of the form $(\chi(\bar{x}), (Q_1, \dots, Q_n))$, we proceed as follows. We first translate $\chi(\bar{x})$ into a logically equivalent formula of first-order logic such that each predicate $Q_i x_1 \dots x_k$ which occurs in χ has distinct variables x_1, \dots, x_k . This is possible using the equality predicate $=$. For example, $Q_i xyx$ is translated into the logically equivalent formula $\forall z(x = z \rightarrow Q_i xyz)$ (or $\exists z(x = z \wedge Q_i xyz)$). We obtain a logically equivalent formula denoted $\chi_=(\bar{x})$ in which the equality predicate may occur. We can define this first transformation so that it is injective because the equality predicate was not present in the initial formulas.

Then, we translate $\chi_=(\bar{x})$ into predicate atomic logic using the translation $T_{\bar{x}}$ of Definition 19 (the equality is simply viewed as a predicate $R_=($). We obtain a formula $\chi' = T_{\bar{x}}(\chi_=(\bar{x}))$ of predicate atomic logic. That second transformation is also injective.

That formula χ' is finally transformed into a molecular connective c . The skeleton decomposition tree of that molecular connective is the skeleton syntactic tree of χ' where the leafs labeled with the skeleton of a predicate Q_i are all replaced by the same skeleton symbol $id_{k_i}^l$ where k_i is the arity of each Q_i and l is a natural number which allows to distinguish $id_{k_i}^l$ from the other symbols of the form $id_{k_i}^{l'}$ appearing in the skeleton. The resulting molecular connective c of arity n and type signature (k, k_1, \dots, k_n) is denoted $t(\chi(\bar{x}), (Q_1, \dots, Q_n))$.

The resulting set of molecular connectives is denoted $t(C^a)$. Then, this translation t is extended to the whole language as follows: for all $\star \in C^a$ and all $\varphi_1, \dots, \varphi_n \in \mathcal{L}_{C^a}$ of appropriate types, $t(\star(\varphi_1, \dots, \varphi_n)) = t(\star)(t(\varphi_1), \dots, t(\varphi_n))$. By construction, this transformation t is injective.

Semantics. Let (M, \bar{w}) be a pointed C^a -model with $M = (W, \mathcal{R})$. The pointed $t(C^a)$ -model $t(M, \bar{w})$ is the $t(C^a)$ -model $((W, \mathcal{R}'), \bar{w})$ where \mathcal{R}' is \mathcal{R} together with the relations of Definition 18 (except those of the first item) as well as the equality predicate $R_=\triangleq \{(w, w) \mid w \in W\}$. \dashv

Lemma 3. Let C^a be a set of abstract connectives, let $\varphi \in \mathcal{L}_{C^a}$ and let (M, \bar{w}) be a pointed C^a -model where \bar{w} is of size the type of φ . Then, we have that

$$(M, \bar{w}) \Vdash \varphi \text{ iff } t(M, \bar{w}) \Vdash t(\varphi)$$

Proof: By induction on φ . The base case holds trivially. For the induction step, that is for abstract connectives of the form $(\chi(\bar{x}), (Q_1, \dots, Q_n))$, each transformation of the formula χ , into $\chi_=(\bar{x})$ and then χ' , preserves its truth

in any assignment \bar{w} by Lemma 1. \square

Definition 22 (Translation from molecular logics to protologics). Let C be a set of molecular connectives.

Syntax. We define the mapping st from molecular connectives to abstract connectives inductively as follows:

- $st(p) \triangleq p$ for all propositional letters p of C .
- For all molecular connectives of the form $c = \star(c_1, \dots, c_m)$ we proceed as follows. First, we replace all symbols id_k^l appearing in c by fresh and distinct propositional letters (p_1, \dots, p_n) . This yields a formula $\varphi \in \mathcal{L}_{\mathsf{C}}$ of some type k . Then, we pick a tuple of free variables \bar{x} of size k and we define the first-order formula $st_{\bar{x}}(\varphi)$ inductively as follows. If φ is a propositional letter p then $st_{\bar{x}}(p) \triangleq Q\bar{x}$, where Q is a predicate symbol of \mathcal{Q} . If φ is of the form $\star(\varphi_1, \dots, \varphi_m)$ then
 - if $\star = (\sigma, \pm, \forall, \bar{k}, (\pm_1, \dots, \pm_m))$ then

$$st_{\bar{x}}(\star(\varphi_1, \dots, \varphi_m)) \triangleq \forall \bar{y}_1 \dots \bar{y}_m (\pm_1 st_{\bar{y}_1}(\varphi_1) \vee \dots \vee \pm_m st_{\bar{y}_m}(\varphi_m) \vee \pm R^\sigma \bar{y}_1 \dots \bar{y}_m \bar{x});$$
 - if $\star = (\sigma, \pm, \exists, \bar{k}, (\pm_1, \dots, \pm_m))$ then

$$st_{\bar{x}}(\star(\varphi_1, \dots, \varphi_m)) \triangleq \exists \bar{y}_1 \dots \bar{y}_m (\pm_1 st_{\bar{y}_1}(\varphi_1) \wedge \dots \wedge \pm_m st_{\bar{y}_m}(\varphi_m) \wedge \pm R^\sigma \bar{y}_1 \dots \bar{y}_m \bar{x});$$

where $\bar{y}_1, \dots, \bar{y}_m$ are fresh tuples of free variables and R is a predicate symbol of $\mathcal{P} - \mathcal{Q}$. We recall that for all formulas ψ , $\pm_i \psi$ stands for ψ if $\pm_i = +$ and for $\neg \psi$ if $\pm_i = -$, and that $R^\sigma \bar{y}_1 \dots \bar{y}_m \bar{x} \triangleq R \bar{y}_{\sigma-(1)} \dots \bar{y}_{\sigma-(m)} \bar{y}_{\sigma-(m+1)}$.

Finally, we define the abstract connective $st(c) \triangleq (st_{\bar{x}}(\varphi), (Q_1, \dots, Q_n))$ where for all $i \in \llbracket 1; n \rrbracket$, $Q_i = st(p_i)$.

The resulting set of abstract connectives is denoted $st(\mathsf{C})$. Then, this translation st is extended to the whole language as follows: for all $\star \in \mathsf{C}$ and all $\varphi_1, \dots, \varphi_n \in \mathcal{L}_{\mathsf{C}}$ of appropriate types, $st(\star(\varphi_1, \dots, \varphi_n)) = st(\star)(st(\varphi_1), \dots, st(\varphi_n))$. By construction, this transformation st is injective. \dashv

Lemma 4. *Let C be a set of molecular connectives, let $\varphi \in \mathcal{L}_{\mathsf{C}}$ and let (M, \bar{w}) be a pointed C -model where \bar{w} is of size the type of φ . Then, we have that*

$$(M, \bar{w}) \Vdash \varphi \text{ iff } (M, \bar{w}) \Vdash st(\varphi)$$

Proof: By induction on φ . The base case holds trivially. The inductive case is just a reformulation in first-order logic of the truth conditions associated to each molecular connective. We do not need to introduce extra predicates R_f like in the previous case, the ones needed are all already present in the initial C -model. Therefore we keep the same C -model. \square

Theorem 2. *The class of protologics is as expressive as the class of molecular logics.*

Proof: We have to prove that every protologic is as expressive as a molecular logic, and vice versa. If $(\mathcal{L}_C, \mathcal{E}_C, \Vdash)$ is a molecular logic, it is as expressive as the protologic $(st(\mathcal{L}_C), \mathcal{E}_C, \Vdash)$, where $st(\mathcal{L}_C) \triangleq \{st(\varphi) \mid \varphi \in \mathcal{L}_C\}$. In that case, the mappings for models are the identity mappings, the mapping from \mathcal{L}_C to $st(\mathcal{L}_C)$ is st and the mapping from $st(\mathcal{L}_C)$ to \mathcal{L}_C is the inverse mapping of st . It exists because st is injective. We then obtain the result thanks to Lemma 4.

Conversely, if $(\mathcal{L}_{C^a}, \mathcal{E}_{C^a}, \Vdash)$ is a protologic, it is as expressive as the molecular logic $(t(\mathcal{L}_{C^a}), t(\mathcal{E}_{C^a}), \Vdash)$, where $t(\mathcal{L}_{C^a}) \triangleq \{t(\varphi) \mid \varphi \in \mathcal{L}_{C^a}\}$ and $t(\mathcal{E}_{C^a}) \triangleq \{t(M, \bar{w}) \mid (M, \bar{w}) \in \mathcal{E}_{C^a}\}$. We define the mapping t^- from $t(\mathcal{E}_{C^a})$ to \mathcal{E}_{C^a} by removing from any $t(C^a)$ -model all the relations which do not appear in the initial C^a -model before the translation t , we thus obtain the initial C^a -model. The mappings t and t^- for models are therefore inverse bijections of each other. The mappings for formulas are t and the inverse of t ; this inverse also exists in that case because t is injective. We then obtain the result thanks to Lemma 3. \square

Remark 3. The translation mappings between models are in fact inverse bijections of each other in this result (and this entails Condition 3 of the definition of equal expressivity). Hence, it holds also with the notion of equal expressivity of Mossakowski et al. [33] (*i.e.* “model-expansive corridors”). Moreover, the translation mappings for formulas are in fact ‘tighter’ than for Theorem 1 because they operate at the level of connectives.

Example 10. We resume our example of modal intuitionistic logic. We consider the abstract connective $(\chi(x), (Q))$ where $\chi(x) \triangleq \forall y(Rxy \rightarrow \forall z(R_\diamond yz \rightarrow Q(z)))$. We show how this abstract connective is transformed into a molecular connective. Applying $T_{(x)}$, we obtain

$$\begin{aligned} T_{(x)}(\chi(x)) &= \forall_1(\perp, T_{(x,y)}(Rxy \rightarrow \forall z(R_\diamond yz \rightarrow Qz))) \\ &= \forall_1(\perp, \supset_2(T_{(x,y)}(Rxy), [(2, 1)] \parallel_1 (T_{(y)}(\forall z(R_\diamond yz \rightarrow Q(z))), \perp))) \\ &= \forall_1(\perp, \supset_2([\text{Id}]R, [(2, 1)] \parallel_1 (\forall_1(\perp, T_{(y,z)}(R_\diamond yz \rightarrow Q(z))), \perp))) \\ &= \forall_1(\perp, \supset_2([\text{Id}]R, [(2, 1)] \parallel_1 (\forall_1(\perp, \supset_2(T_{(y,z)}(R_\diamond yz), [(2, 1)] \parallel_1 \\ &\quad (T_{(z)}(Qz), \perp)))))) \\ &= \forall_1(\perp, \supset_2([\text{Id}]R, [(2, 1)] \parallel_1 (\forall_1(\perp, \supset_2([\text{Id}]R_\diamond, [(2, 1)] \parallel_1 \\ &\quad ([\text{Id}]Q, \perp)))))) \end{aligned}$$

(The subscript mappings f_i^k are removed since in our algorithm they are the identity mapping.) Then we replace Q by id_1^1 and we obtain the following molecular connective:

$$\forall_1(\perp, \supset_2([\text{Id}]R, \parallel_1 (\forall_1(\perp, \supset_2([\text{Id}]R_\diamond, \parallel_1 (\perp, [\text{Id}]id_1^1))))))$$

We could remove the connectives $[\text{Id}]$. It turns out that the molecular connective that we obtain is quite different from the molecular connective that was introduced in Example 7. Yet, they both have the same effect, the former

on \mathcal{C} -models and the latter on $\mathcal{C}^{\mathcal{P}}$ -models when we consider only the part of the information present in the initial \mathcal{C} -model. \dashv

So, it is possible that a protologic corresponds naturally to a molecular logic but that its translation into a molecular logic using our algorithm of Definition 21 does not yield the expected natural outcome. However, the two logics, the one expected and the one obtained thanks to our translation, will be equally expressive. The pieces of information which are added in $\mathcal{C}^{\mathcal{P}}$ -models (corresponding to the connectives $[\sigma_k], \parallel_k, \supset_k, \forall_k$) actually do not increase the expressive power of the logic because the object language cannot refer to their associated relations. Yet, they allow us to reformulate it systematically under the form of an atomic or molecular logic.

6. Predicate Molecular Logics: the Core of Molecular Logics

The proofs of our previous theorems leads us to identify a subclass of molecular logics that we will call ‘predicate’ molecular logics. This subclass of molecular logics constitutes its ‘core’, in the sense that any molecular logic is as expressive as a predicate molecular logic.

Definition 23 (Predicate molecular logic). A *predicate molecular logic* is a molecular logic whose molecular connectives are built up from the predicate atomic connectives $\{\perp, \forall_0\} \cup \{[\sigma_k], \parallel_k, \supset_k, \forall_k \mid k \in \mathbb{N}^*, \sigma_k \in \mathfrak{S}_k\}$ of Definitions 17 and 18, together with some set of (atomic) propositional letters (possibly empty). \dashv

Theorem 3. *The class of predicate molecular logics is as expressive as the class of molecular logics and therefore also as expressive as the class of protologics.*

Proof: It follows easily from the proof of Theorem 2 and its lemmas. \square

7. Conclusion

This article contributes to the systematic exploration of non-classical logics. It shows in particular the central role played by atomic and molecular logics in that study. They behave as ‘paradigmatic logics’: every logic whose connectives are defined by first-order formulas – that is, every protologic – is as expressive as a molecular logic, as proved by Theorem 2. This result illustrates their (somehow) ‘universal’ feature.

The main advantage of our overall approach is that it provides a uniform and generic way to explore and study non-classical logics. Atomic and molecular logics can be (somehow) seen as normal forms for logics. It is better and easier to deal with atomic and molecular logics than with protologics directly, because of their specific shape and format. Indeed, one can define and compute automatically notions of bisimulations for any atomic and molecular logic. So, we can develop in a systematic way their model theory, and

therefore, also develop, indirectly, the model theory of any protologic and non-classical logic. In that respect, a number of results have already been obtained in the model theory of atomic and molecular logics such as a van Benthem type characterization theorem for molecular logics given by connectives which are ‘uniform’ [6].² Likewise, our approach based on atomic and molecular logics allows us to develop in a systematic and uniform way the proof theory of non-classical logics. Indeed, one can automatically compute a display calculus for any gaggle logics, which are specific basic atomic logics [4, 5]³.

Through close analysis of the proofs of our results, we realized that every protologic and every molecular logic is as expressive as a specific kind of molecular logic, which resorts to the connectives \perp , $[\sigma_k]$, $\|_k$, \supset_k , \forall_k , \forall_0 and propositional letters. This class of molecular logics, that we called predicate molecular logics, therefore constitutes the ‘core’ of the class of all logics since our Theorem 3 states that any protologic and any molecular logic is as expressive as one of them. More work and study regarding that specific class of molecular logics and its relationships with other logics is needed.

We have restricted our investigations in this article to non-classical logics which are fragments of first-order logic. Yet, the same methodology could be applied if we replaced in the truth conditions of connectives first-order logic by second-order logic and, in fact, by any higher-order logic. Indeed, we could consider logics whose connectives are defined by formulas of second-order logic instead of first-order logic, like the modal mu-calculus [24]. Following this research track, we may meet at some point the work on higher-order logics of abstract model theory [7]. That said, the move from first-order to second-order logic may be neither straightforward nor natural, and may lead to some complexifications since we would need, for example, to choose between a Henkin semantics and a standard semantics [41]. In any case, we believe that our overall approach and methodology based on that simple idea is the right track to follow if we want to explore and study non-classical logics in a systematic and comprehensive way; in particular, because it is grounded on a class and hierarchy of logics which are naturally well-defined and articulated and which have already been well studied.

References

- [1] Gerard Allwein and J. Michael Dunn. Kripke models for linear logic. *The Journal of Symbolic Logic*, 58(2):514–545, June 1993.

²The definition of molecular connectives and molecular logics of [6] is less general than in the present article, in the sense that it is not possible there to have the same argument at different places in the definition of a molecular connective. This said, it is nevertheless possible to define an appropriate notion of C-bisimulation for the more general definition of molecular logics of the present article.

³[5] differs very slightly from [4]. It essentially corrects minor mistakes and typos and proves that the rule of associativity is derivable in GGL_C.

- [2] Peter Arndt, Rodrigo de Alvarenga Freire, Odilon Otavio Luciano, and Hugo Luiz Mariano. A global glance on categories in logic. *Logica Universalis*, 1(1):3–39, 2007.
- [3] Guillaume Aucher. Displaying Updates in Logic. *Journal of Logic and Computation*, 26(6):1865–1912, March 2016.
- [4] Guillaume Aucher. Towards universal logic: gaggle logics. *Journal of Applied Logics – IfCoLog Journal of Logics and their Applications*, 7(6):875–945, 2020.
- [5] Guillaume Aucher. *Selected Topics from Contemporary Logics*, chapter Towards Universal Logic: Gaggle Logics, pages 5–73. Landscapes in Logic. College Publications, October 2021.
- [6] Guillaume Aucher. A van Benthem theorem for atomic and molecular logics. In Andrzej Indrzejczak and Michał Zawidzki, editors, Proceedings of the 10th International Conference on *Non-Classical Logics. Theory and Applications*, Łódź, Poland, 14-18 March 2022, volume 358 of *Electronic Proceedings in Theoretical Computer Science*, pages 84–101. Open Publishing Association, 2022.
- [7] Jon Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7(2):221 – 265, 1974.
- [8] Jean-Yves Béziau. *Logica Universalis*, chapter From Consequence Operator to Universal Logic: A Survey of General Abstract Logic. Birkhäuser Basel, 2007.
- [9] Jean-Yves Béziau. Editorial: Introduction to the universal logic corner. *J. Log. Comput.*, 19(6):1111, 2009.
- [10] Katalin Bimbó and J. Michael Dunn. *Generalized Galois Logics: Relational Semantics of Nonclassical Logical Calculi*. Number 188. Center for the Study of Language and Information, 2008.
- [11] C. C. Chang and H. J. Keisler. *Model Theory*. Studies in Logic and the Foundations of Mathematics. Elsevier, 1990.
- [12] J Michael Dunn. Gaggle theory: an abstraction of galois connections and residuation, with applications to negation, implication, and various logical operators. In *European Workshop on Logics in Artificial Intelligence*, pages 31–51. Springer Berlin Heidelberg, 1990.
- [13] J. Michael Dunn. *Philosophy of Language and Logic*, volume 7 of *Philosophical Perspectives*, chapter Perp and star: Two treatments of negation, pages 331–357. Ridgeview Publishing Company, Atascadero, California, USA, 1993.
- [14] J. Michael Dunn and Gary M. Hardegree. *Algebraic Methods in Philosophical Logic*. Number 41 in Oxford Logic Guides. Clarendon Press: Oxford, 2001.
- [15] J. Michael Dunn and Chunlai Zhou. Negation in the context of gaggle theory. *Studia Logica*, 80(2-3):235–264, 2005.
- [16] Herbert B. Enderton. *An Introduction to Mathematical Logic*. Academic Press, 2001.
- [17] Dov Gabbay, editor. *What is a Logical System ?* Studies in Logic and Computation. Oxford University Press, 1994.
- [18] Dov Gabbay. *Labelled Deductive Systems*, volume 1. Oxford University Press, 1996.
- [19] Marta García-Matos and Jouko Väänänen. Abstract model theory as a framework for universal logic. In Jean-Yves Béziau, editor, *Logica Universalis*, pages 19–33, Basel, 2007. Birkhäuser Basel.

- [20] Joseph A Goguen and Rod M Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM (JACM)*, 39(1):95–146, 1992.
- [21] Robert Goldblatt. Semantic analysis of orthologic. *Journal of Philosophical Logic*, 3:19–35, 1974.
- [22] Rajeev Goré. Substructural logics on display. *Logic Journal of IGPL*, 6(3):451–504, 1998.
- [23] Viktor Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A. I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334. Nauka, Moscow, 1983.
- [24] David Janin and Giacomo Lenzi. Relating levels of the mu-calculus hierarchy and levels of the monadic hierarchy. In *LICS*, pages 347–356, Boston, United States, 2001. IEEE Computer Society.
- [25] Saul A. Kripke. Semantical analysis of modal logic, i: Normal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 8:113–116, 1963.
- [26] Saul A. Kripke. *Formal Systems and Recursive Functions*, chapter Semantical Analysis of Intuitionistic Logic, I, pages 92–130. North Holland, Amsterdam, 1965.
- [27] Steven T Kuhn. Quantifiers as modal operators. *Studia Logica*, 39(2):145–158, 1980.
- [28] Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, 65:154–170, 1958.
- [29] Maarten Marx and Yde Venema. *Multi-dimensional modal logic*, volume 4 of *Applied logic series*. Kluwer, 1997.
- [30] José Meseguer. General logics. In H.-D. Ebbinghaus, J. Fernandez-Prida, M. Garrido, D. Lascar, and M. Rodríguez Artalejo, editors, *Logic Colloquium’87*, volume 129 of *Studies in Logic and the Foundations of Mathematics*, pages 275 – 329. Elsevier, 1989.
- [31] Michael Moortgat. Symmetries in natural language syntax and semantics: the lambek-grishin calculus. In *Logic, Language, Information and Computation*, pages 264–284. Springer, 2007.
- [32] Lawrence S Moss. Applied logic: A manifesto. In *Mathematical problems from applied logic I*, pages 317–343. Springer, 2006.
- [33] Till Mossakowski, Razvan Diaconescu, and Andrzej Tarlecki. What is a logic translation? *Logica Universalis*, 3(1):95–124, 2009.
- [34] Till Mossakowski, Joseph Goguen, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic? In Jean-Yves Beziau, editor, *Logica Universalis*, pages 111–133, Basel, 2007. Birkhäuser Basel.
- [35] Grigory K. Olkhovikov. On expressive power of basic modal intuitionistic logic as a fragment of classical FOL. *J. Appl. Log.*, 21:57–90, 2017.
- [36] Graham Priest. *An introduction to non-classical logic*. Cambridge University Press, 2011.
- [37] Arthur Prior. *Past, Present and Future*. Oxford: Clarendon Press, 1967.
- [38] Greg Restall. *An Introduction to Substructural Logics*. Routledge, 2000.

- [39] Joseph J. Rotman. *An Introduction to the Theory of Groups*, volume 148 of *Graduate texts in mathematics*. Springer, 1995.
- [40] Giovanni Sambin, Giulia Battilotti, and Claudia Faggian. Basic logic: reflection, symmetry, visibility. *The Journal of Symbolic Logic*, 65(03):979–1013, 2000.
- [41] Jouko Väänänen. Second-order and Higher-order Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition, 2021.

Guillaume Aucher
Univ Rennes, CNRS, IRISA, IRMAR
263, Avenue du Général Leclerc
35042 Rennes Cedex, France
e-mail: `guillaume.aucher@univ-rennes1.fr`