Logic and **L**ogical **P**hilosophy
Online First Articles (2025)
DOI: 10.12775/LLP.2025.018

Julian M. Valdes-Toro

Contradictions in Multiverse: Translation of Paraconsistent Logic *daC* into Many-Sorted Logic

Abstract. A translation of Priest's paraconsistent logic *daC* into many-sorted logic is presented. Besides, following the project of (Manzano, 1996), the representation theorem, the main theorem and the calculi equivalence are proved. So, it is demonstrated that the formulated translation preserves the set of valid formulas, the consequence relation, and the derivation relation of *daC*. Furthermore, the compactness and Löwenheim-Skolem theorems are proved for this logic. Alternative proofs for the soundness and completeness theorems for *daC* based on the translation are also presented.

Keywords: translation; paraconsistent logic; logic *daC*; many-sorted logic

Introduction

Informally, a translation could be understood as a function that projects the formulas of a source logic into formulas of a frame logic. This projection varies depending on what each author understands by translation; so, for example, there are even structural elements of the semantics that are projected in some translations.

Following (Ohlbach et al., 2001; Shen et al., 2010) and generalizing a bit the visions of (Manzano, 1996; Manzano et al., 2022), it is possible to affirm that translations have, among others, the following advantages: (1) they allow a better understanding of the expressive power of translated logics, (2) some translations can export procedures, tools and properties from the target logic to the source logic, and (3) some translations allow to combine and compare two logics if both are translated into the framework of a third. The advantage most easily found in

Received January 4, 2023. *Accepted* August 11, 2025. *Published online* September 12, 2025

the literature devoted to this subject is the export of logical properties. Thus, for example, a conservative translation preserves, among others, the Deduction Theorem under certain conditions (Carnielli et al., 2009).

Although it will not be the emphasis of this paper, it is interesting to note that, from a philosophical (rather than a technical) point of view, the possibility of translation between logics is very suggestive. This is so because of, at least, two reasons: first, the discussion about the meaning of the formulas in a logic becomes more complex; and second, the translations enlighten the alleged rivalry between some logical systems and the relationships between them. It is not worthless that the intuitionist logician Kolmogorov (1925) translates a classical propositional logical system into intuitionist logic to explain why the use of the Excluded Middle principle in transfinite arguments had not led to contradictions. Other specific translations, all on the relationship between classical logic, modal logics, and intuitionist logic were developed by Glivenko, Lewis, Langford, Gödel, Gentzen and van Benthem(see, e.g., Carnielli et al., 2009; Manzano, 2004; Manzano et al., 2022; van Benthem, 1983, 2001). After these, translations in the world of logic began to be widely developed and studied.

Thus, the aim of this text, framed in the previous field of study, is to define a translation function between Priest's paraconsistent logic daC and many-sorted logic MSL following the Manzano's translation project presented in (Manzano, 1996). The logic daC was introduced in (Priest, 2009, 2010) and developed in (Ferguson, 2013, 2018; Osorio Galindo et al., 2016). On the other side, MSL is a powerful logical system introduced by Hao Wang in 1952 and Solomon Feferman in 1967 (see Manzano et al., 2022) widely used in technological environment and proposed by Manzano (1996) as a frame to translate multiple logical systems. Beside translation itself, the other aim of this paper is to prove the three levels of the translation proposed by Manzano (1996). In the first level, the Representation Theorem is proved; i.e., A is a valid formula or tautology of daC iff its translation is a MSL valid formula or tautology modulo Δ, where Δ is a set of MSL formulas that serves as a set of axioms to adequately represent daC as a theory inside MSL language. Afterward, it will be proved the second level, the Main Theorem; i.e., A is a consequence in daC of a set Γ of premises iff the translation of A is a consequence in MSL of the union of the set of the translation of the original premises and the set Δ from Representation Theorem. And in the last level, the Calculi Equivalence is demonstrated: A is derived from a set Γ of premises in *daC* calculus iff the translation of A is derived from the union of the set of the translation of the original premises in *MSL* calculus and the set Δ from Representation Theorem. At this last level, the Completeness and Soundness theorems for *daC* can be exported from *MSL* due to the proposed translation.

To achieve all this, a presentation of both logical systems is made first, with special emphasis on their semantics, then a translation function t, a structure conversion function $Conv_1$ and a function H (used to restrict the set of structures of MSL) are defined by recursion. At the first level, an equivalence is established between validity in *daC* and *MSL*, restricted to the particular class of *MSL* structures ($Conv_1(Str(daC))$). Starting from this result, the Representation Theorem is proved using the set Δ. Afterward, in the second level of the translation, a function $Conv_2 = {Conv_1}^{-1} \circ H$ is defined for proving the Main Theorem with the help of the set Δ. Finally, in the last level, it is shown that all derivations in *daC* calculus are translated into arguments in *MSL* modulo Δ which conclusion is derived from the premises; and that all derivations in *MSL* modulo Δ are the result of the translation function applied to derivations in *daC*. To aim this, the Canonical Model of *daC* is defined. Hence, it would have been proved that *daC* calculus and *MSL* modulo Δ calculus are equivalent.

This process follows the steps of the particular translation project proposed by María Manzano in (1996). In this project, translations are understood as functions that represent the meaning (the semantics) of the formulas of a specific logic into formulas of the target logic. At a technical level, the importance of this type of translation lies in the fact that at the end of different levels, it is possible to export various metalogical properties of *MSL* to the translated logic (*daC* in this case): after proving the Main Theorem, the Compactness and Löwenheim-Skolem theorems are exported into *daC*; at the end of the last level of the translation, the Soundness and Completeness theorems are exported, too.

1. Paraconsistent Logic *daC*

The da Costa logic, also known as Priest-da Costa logic (or simply *daC*), is a paraconsistent logical system created by the Australian logician and philosopher Priest (2009) that should not be confused with the developments of da Costa's systems of inconsistency C_n ($1 \leqslant n \leqslant \omega$). Priest developed *daC* as a dualization of the intuitionist negation. Indeed, the definitions in *daC* semantics are almost equal to the correspondent intuitionist definitions, except for the negation; which, while in the latter system is expressed by the condition '$\mathfrak{A}, \boldsymbol{w} \models {\sim}A$ iff for every $\boldsymbol{u}$ if $(\boldsymbol{w}, \boldsymbol{u}) \in R$ then $\boldsymbol{u} \not\models A$', it becomes '$\mathfrak{A}, \boldsymbol{w} \models {\sim}A$ iff there is a $\boldsymbol{u}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R$ and $\boldsymbol{u} \not\models A$' in the semantics of *daC*. Specifically, this system presents the following family of structures and the following intensional logic connectors.

1.1. Language of *daC*

DEFINITION 1.1. The alphabet of *daC* is made up of the elements of the next sets only: $Atom := \{p_1, p_2, \ldots\}$, $Con := \{\wedge, \vee, \rightarrowtail, {\sim}\}$.[1]

DEFINITION 1.2. The set of *daC* formulas $Form(daC)$ consists of finite strings of elements of *Atom* and *Con* that meet the following rules:

F1. $Atom \subseteq Form(daC)$.
F2. If $A, B \in Form(daC)$ then ${\sim}A, A \wedge B, A \vee B, A \rightarrowtail B \in Form(daC)$.
F3. No other string of elements of *Atom* and *Con* is in $Form(daC)$.

1.2. Semantics for *daC*

DEFINITION 1.3. A Kripke model $\mathfrak{A} = \langle \boldsymbol{W}, R, \langle P_n^{\mathfrak{A}} \rangle_{n \in N, N \subseteq \mathbb{N}} \rangle$ is a structure for *daC* such that (1) $\boldsymbol{W} \neq \emptyset$, (2) $R \subseteq \boldsymbol{W}^2$ is a reflexive and transitive accessibility relation, (3) $P_n^{\mathfrak{A}} \subseteq \boldsymbol{W}$ for every atom p_n (the set of worlds where p_n holds), and (4) the Heredity Constraint is satisfied; i.e, for every $\boldsymbol{w}, \boldsymbol{u} \in \boldsymbol{W}$, if $(\boldsymbol{w}, \boldsymbol{u}) \in R$ and $\boldsymbol{w} \in P_n^{\mathfrak{A}}$ then $\boldsymbol{u} \in P_n^{\mathfrak{A}}$. The class of structures in *daC* is called $Str(daC)$.

DEFINITION 1.4. Let $\mathfrak{A}$ be a structure for *daC* and $A, B \in Form(daC)$. A formula holds in a world $\boldsymbol{w} \in \boldsymbol{W}$ in $\mathfrak{A}$ if it satisfies the following conditions:

1. $\mathfrak{A}, \boldsymbol{w} \models p_n$ iff $\boldsymbol{w} \in P_n^{\mathfrak{A}}$.
2. $\mathfrak{A}, \boldsymbol{w} \models A \wedge B$ iff $\mathfrak{A}, \boldsymbol{w} \models A$ and $\mathfrak{A}, \boldsymbol{w} \models B$.
3. $\mathfrak{A}, \boldsymbol{w} \models A \vee B$ iff $\mathfrak{A}, \boldsymbol{w} \models A$ or $\mathfrak{A}, \boldsymbol{w} \models B$.
4. $\mathfrak{A}, \boldsymbol{w} \models A \rightarrowtail B$ iff for every $\boldsymbol{u} \in \boldsymbol{W}$, $(\boldsymbol{w}, \boldsymbol{u}) \in R$ implies that, if $\mathfrak{A}, \boldsymbol{u} \models A$, then $\mathfrak{A}, \boldsymbol{u} \models B$.

[1] Original presentation in (Priest, 2009) uses the symbols $\neg$ and $\rightarrow$ for negation and implication. Nevertheless, the notation has been modified here in order to keep these two symbols for classical negation and classical implication.

5. $\mathfrak{A}, \boldsymbol{w} \models \sim A$ iff there is a $\boldsymbol{u} \in \boldsymbol{W}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R$ and $\mathfrak{A}, \boldsymbol{u} \not\models A$.

When it is clear which structure $\mathfrak{A}$ is referred to, $\boldsymbol{w} \models A$ is used instead of $\mathfrak{A}, \boldsymbol{w} \models A$.

From the last definition, it is importat to remark that while in (4), the world $\boldsymbol{w}$ access to the world $\boldsymbol{u}$, in (5) the world $\boldsymbol{u}$ does access to the world $\boldsymbol{w}$; i.e, it is possible to interpret that, in (4), the relation R is used, but in (5) the inverse relation $\overline{R}$ is. Due to this, the negation in *daC* is not standard ($\mathfrak{A}, \boldsymbol{w} \models \sim A$ is not equivalent to $\mathfrak{A}, \boldsymbol{w} \not\models A$). Moreover, the implication in *daC* ($A \rightarrowtail B$) is different from the classical one, too. The former behaves like the strict implication; which means, for example, that $A \rightarrowtail B \not\equiv \sim A \vee B$. This is true even if a classical negation is added to *daC* and used in this pretended equivalence.

DEFINITION 1.5. Let $\mathfrak{A} \in Str(daC)$ and $A \in Form(daC)$. $A^{\mathfrak{A}}$ is the set of worlds in $\boldsymbol{W}$ where A holds; i.e, $A^{\mathfrak{A}} = \{\boldsymbol{w} \in W \mid \boldsymbol{w} \models A\}$.

It is obvious that

PROPOSITION 1.1. *Let* $\overline{R} = \{(\boldsymbol{u}, \boldsymbol{w}) \mid (\boldsymbol{w}, \boldsymbol{u}) \in R\}$.
$(\sim A)^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W} \mid$ *there is a* $\boldsymbol{u} \in \boldsymbol{W}$ *s. t.* $(\boldsymbol{u}, \boldsymbol{w}) \in R$ *and* $\boldsymbol{u} \notin A^{\mathfrak{A}}\} = Dom(\overline{R} \cap (\boldsymbol{W} \times (\boldsymbol{W} - A^{\mathfrak{A}}))$
$(A \rightarrowtail B)^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W} \mid$ *for every* $\boldsymbol{u} \in \boldsymbol{W}$ *s. t.* $(\boldsymbol{w}, \boldsymbol{u}) \in R$, *if* $\boldsymbol{u} \in A^{\mathfrak{A}}$ *then* $\boldsymbol{u} \in B^{\mathfrak{A}}\} = \boldsymbol{W} - Dom(R \cap (\boldsymbol{W} \times (A^{\mathfrak{A}} - B^{\mathfrak{A}}))$
$(A \wedge B)^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W} \mid \boldsymbol{w} \in A^{\mathfrak{A}}$ *and* $\boldsymbol{w} \in B^{\mathfrak{A}}\} = A^{\mathfrak{A}} \cap B^{\mathfrak{A}}$
$(A \vee B)^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W} \mid \boldsymbol{w} \in A^{\mathfrak{A}}$ *or* $\boldsymbol{w} \in B^{\mathfrak{A}}\} = A^{\mathfrak{A}} \cup B^{\mathfrak{A}}$

1.3. On negation in *daC*

It is interesting to notice that, unlike intuitionist and classical logics, to say that a formula A does not hold is not the same as saying that the negation of the formula holds.

PROPOSITION 1.2. *There are models for daC* $\mathfrak{A}$ *such that, for some* $A \in Form(daC)$ *and some* $\boldsymbol{w} \in \boldsymbol{W}$, $\boldsymbol{w} \models A$ *and* $\boldsymbol{w} \models \sim A$, *but* $\boldsymbol{w} \not\models B$ *for some* $B \in Form(daC)$; *i.e., the explosion principle does not hold in daC.*

PROOF. Take as an example the structure $\mathfrak{A}$ where $\boldsymbol{W} = \{\boldsymbol{a}, \boldsymbol{b}\}$, $R = \{(\boldsymbol{a}, \boldsymbol{a}), (\boldsymbol{b}, \boldsymbol{b}), (\boldsymbol{b}, \boldsymbol{a})\}$, $\boldsymbol{a} \in A^{\mathfrak{A}}$, $\boldsymbol{a} \notin B^{\mathfrak{A}}$ and $\boldsymbol{b} \notin A^{\mathfrak{A}}$. Due to 1.4, in this case, $\boldsymbol{a} \models A, \boldsymbol{a} \models \sim A$ and $\boldsymbol{a} \not\models B$. ⊣

COROLLARY 1.3. *For any daC structure* $\mathfrak{A}$ *and any* $A \in Form(daC)$, $\boldsymbol{w} \models {\sim}A$ *does not imply* $\boldsymbol{w} \not\models A$.

Despite the above, the following is true:

PROPOSITION 1.4. *For any* $\mathfrak{A}, \boldsymbol{w} \in \boldsymbol{W}$ *and* $A \in Form(daC)$, *if* $\boldsymbol{w} \not\models A$ *then* $\boldsymbol{w} \models {\sim}A$.

PROOF. Take a random world $\boldsymbol{w}$ such that $\boldsymbol{w} \not\models A$. Since R is reflexive, for Definition 1.4, $\boldsymbol{w} \models {\sim}A$. ⊣

As usual in paraconsistent logics, *daC* distinguish 'A does not holds' from '${\sim}A$ holds'. As such, a problematic inconsistency or strong contradiction is not $\boldsymbol{a} \models A$ and $\boldsymbol{a} \models {\sim}A$, but $\boldsymbol{a} \models A$ and $\boldsymbol{a} \not\models A$ since it would be required for the world $\boldsymbol{a}$ to belong and not belong to the set $A^{\mathfrak{A}}$, which is impossible in the classical set theory behind the semantics of *daC*. Thus, the worlds $\boldsymbol{w} \in \boldsymbol{W}$ are not standard regarding *daC* negation; i.e., it is possible, for a not trivial model, to have a world $\boldsymbol{w}$ such that $\boldsymbol{w} \models A$ and $\boldsymbol{w} \models {\sim}A$. It is obvious, then, that *daC* meets the next definition:

DEFINITION 1.6. A logic $\mathbb{L}$ is paraconsistent iff the explosion principle does not hold; i.e., for some $A, B \in Form(\mathbb{L})$, $A \wedge {\sim}A \not\models B$.

1.4. Validity and consequence in *daC*

Starting from Definition 1.4, the following definitions for validity and consequence are presented. These are the usual definitions for any Kripke semantics.

Let $A, B \in Form(daC)$.

DEFINITION 1.7 (Validity in a Structure $\mathfrak{A} \in Str(daC)$). A holds in a structure $\mathfrak{A} \in Str(daC)$ (i.e., $\mathfrak{A} \models A$) iff $\mathfrak{A}, \boldsymbol{w} \models A$ for every $\boldsymbol{w}$ in $\mathfrak{A}$.

DEFINITION 1.8 (Valid Formulas in *daC*). A is a valid formula in *daC* (i.e., $\models_{daC} A$) iff $\mathfrak{A} \models A$ for every $\mathfrak{A}$.

DEFINITION 1.9 (Validity of Sets in *daC*). A set $\Gamma \subseteq Form(daC)$ holds in a structure $\mathfrak{A} \in Str(daC)$ (i.e., $\mathfrak{A} \models \Gamma$) iff $\mathfrak{A} \models A$ for every $A \in \Gamma$.

DEFINITION 1.10 (Consequence in *daC* structure). A is a consequence of a set $\Gamma \subseteq Form(daC)$ in a structure $\mathfrak{A} \in Str(daC)$ (i.e., $\Gamma \models_{\mathfrak{A}} A$) iff $\mathfrak{A} \models \Gamma$ implies $\mathfrak{A} \models A$.

DEFINITION 1.11 (Consequence in daC). A is a consequence of a set $\Gamma \subseteq Form(daC)$ (i.e., $\Gamma \models_{daC} A$) iff for every structure $\mathfrak{A} \in Str(daC)$, if $\mathfrak{A} \models \Gamma$ then $\mathfrak{A} \models A$.

1.5. *daC* calculus (natural deduction)

Priest (2009) also presented a tableaux system and a natural deduction calculus for daC. Priest demonstrated Soundness and Completeness theorems for both of them. The following is the natural deduction calculus $ND(daC)$. As this is the only calculus for daC in this paper, it can also be referred to as $Cal(daC)$.

DEFINITION 1.12 (Natural Deduction Rules for daC). The $ND(daC)$ rules are the following:

Let $A, B, C \in Form(daC)$

$$\{A \wedge B\} \vdash_{daC} A \tag{1}$$

$$\{A \wedge B\} \vdash_{daC} B \tag{2}$$

$$\{A, B\} \vdash_{daC} A \wedge B \tag{3}$$

$$\text{If } A \vdash_{daC} C \text{ and } B \vdash_{daC} C \text{ then } \{A \vee B\} \vdash_{daC} C \tag{4}$$

$$\{A\} \vdash_{daC} A \vee B \tag{5}$$

$$\{B\} \vdash_{daC} A \vee B \tag{6}$$

$$\{A \rightarrowtail B, A\} \vdash_{daC} B \tag{7}$$

$$\text{If } A \vdash_{daC} B \text{ then } \vdash_{daC} A \rightarrowtail B \tag{8}$$

$$\vdash_{daC} A \vee {\sim}A \tag{9}$$

$$\text{If } \vdash_{daC} A \vee B \text{ then } \{{\sim}A\} \vdash_{daC} B \tag{10}$$

As usual, $\vdash_{daC}$ is the derivation symbol for daC, and $\Gamma \vdash_{daC} A$ means that $A \in Form(daC)$ is derivable from $\Gamma \subseteq Form(daC)$. When $\Gamma = \emptyset$, $\vdash_{daC} A$ might be used instead of $\emptyset \vdash_{daC} A$. Hence, $\vdash_{daC} A$ means that A is derivable in $ND(daC)$ without any assumption. Notice that the rule (10) is applicable only when $\vdash_{daC} A \vee B$; i.e., when the premise $A \vee B$ is derivable in $ND(daC)$ without assumptions.

DEFINITION 1.13 (Derivation in $ND(daC)$). $A \in Form(daC)$ is derived from a set $\Gamma \subseteq Form(daC)$ (i.e., $\Gamma \vdash_{daC} A$) iff there is a finite string of formulas result of the applications of rules from Definition 1.12 over finite subsets of the union of Γ and the set of formulas of the string, such that A is the last line of the string.

Since this is the only calculus for daC presented here, we will also use $Cal(daC)$ as a synonym for $ND(daC)$.

The next propositions about $\vdash_{daC}$ are demonstrated as they are useful for some future proofs.

PROPOSITION 1.5. *In* $ND(daC)$, *if* $\Gamma \vdash_{daC} A$ *and* $A \vdash_{daC} B$, *then* $\Gamma \vdash_{daC} B$, *for any* $\Gamma \cup \{A, B\} \subseteq Form(daC)$.

PROOF. Assume that $\Gamma \vdash_{daC} A$ and $A \vdash_{daC} B$. By Definition 1.13, the fact that A is the final formula of the string in the former derivation, and $A \vdash_{daC} B$, then there is a new finite string of formulas result of the applications of rules from Definition 1.12 over finite subsets of the union of Γ and the set of formulas of the string (including A), such that B is the last line of this new string. Thus, from Definition 1.13 again, $\Gamma \vdash_{daC} B$. $\dashv$

PROPOSITION 1.6. *Let* $A \in Form(daC)$. *In* $ND(daC)$, $A \vdash_{daC} A$.

PROOF. Suppose $\Gamma = \{A\}$, as $\{A\} = \{A, A\}$, by rule (3) in $ND(daC)$, $A \vdash_{daC} A \wedge A$; and by rule (1) and Proposition 1.5, $A \vdash_{daC} A$. $\dashv$

PROPOSITION 1.7. *Let* $\Gamma \cup \{A\} \subseteq Form(daC)$. *In* $ND(daC)$, $\Gamma \vdash_{daC} A$ *iff* $\Gamma \vdash_{daC} A \vee A$.

PROOF. ($\Rightarrow$) Suppose $\Gamma \vdash_{daC} A$. By rule (5) from Definition 1.12 and Proposition 1.5, $\Gamma \vdash_{daC} A \vee A$. ($\Leftarrow$) Suppose $\Gamma \vdash_{daC} A \vee A$. Since, from Proposition 1.6, $A \vdash_{daC} A$, then by rule (4) from Definition 1.12, $\Gamma \vdash_{daC} A$ is derived. $\dashv$

PROPOSITION 1.8. *Let* $A, B \in Form(daC)$ *and* $\Gamma \subseteq Form(daC)$. *In* $ND(daC)$, *if* $\Gamma \nvdash_{daC} A \vee B$, *then* $\Gamma \nvdash_{daC} A$ *and* $\Gamma \nvdash_{daC} B$.

PROOF. Suppose $\Gamma \vdash_{daC} A$ or $\Gamma \vdash_{daC} B$. In the former option, $\Gamma \vdash_{daC} A \vee B$ by rule (5) in Definition 1.12. In the later option, the same is derived by rule (6) in Definition 1.12. Therefore, $\Gamma \vdash_{daC} A \vee B$. $\dashv$

PROPOSITION 1.9. *Let* $A \in Form(daC)$ *and* $\Gamma \subseteq Form(daC)$. *For* $ND(daC)$, *if* $A \in \Gamma$ *then* $\Gamma \vdash_{daC} A$.

PROOF. Take $A \in \Gamma$ as hypothesis. By rule (3) in Definition 1.12, $\Gamma \vdash_{daC} A \wedge A$; and by rule (1) and Proposition 1.5, $\Gamma \vdash_{daC} A$. $\dashv$

1.6. Canonical model for *daC*

In this section, the canonical model for *daC* is presented based on (Priest, 2009). Unlike usual canonical models for other logics, the set of formulas used as worlds in the universe of the structure are not *maximal consistent*, but *prime* and *deductively closed*. This change is mandatory since *daC* is a paraconsistent logic; which implies that a maximal consistent set of formulas is no longer the most detailed possible description of a structure because now we have structures satisfying contradictions $A \wedge {\sim}A$. Nevertheless, the usual propositions and theorems are demonstrated to prove that $\Gamma \vdash_{daC} A$ iff $A \in \Gamma$ for every Γ in the universe of the canonical model (Theorem 1.12). This theorem will be used in the last level of the translation in order to prove the calculi equivalence. Every definition in this section is based on the correspondent ones developed by Priest himself in (Priest, 2009, 2010).

Definition 1.14 (Prime Set). Let $\Gamma \cup \{A, B\} \subseteq \mathit{Form}(daC)$. Γ is prime iff, if $A \vee B \in \Gamma$ then $A \in \Gamma$ or $B \in \Gamma$.

Definition 1.15 (Deductively Closed Set). Let $A, B \in \mathit{Form}(daC)$ and $\Gamma \subseteq \mathit{Form}(daC)$. Γ is deductively closed iff, if $\Gamma \vdash_{daC} A$ then $A \in \Gamma$.

Definition 1.16. Let $\Gamma, \Pi \subseteq \mathit{Form}(daC)$. $\Gamma \vdash_{daC} \Pi$ iff there is a finite set $\{A_1, \ldots, A_n\} \subseteq \Pi$ such that $\Gamma \vdash_{daC} A_1 \vee \cdots \vee A_n$.

Definition 1.17 (Canonical Model for *daC*). Let $\Gamma, \Sigma \in \boldsymbol{W}_{CAN}$. The Canonical Model for *daC* is $\mathfrak{A}_{CAN} = \langle \boldsymbol{W}_{CAN}, R_{CAN}, \langle P_n^{\mathfrak{A}} \rangle_{n \in N, N \subseteq \mathbb{N}} \rangle$, s.t.:
$\boldsymbol{W}_{CAN} = \{\Gamma \subseteq \mathit{Form}(daC) \mid \Gamma \text{ is prime and deductively closed}\}$,
$R_{CAN} \subseteq (\boldsymbol{W}_{CAN})^2$ such that $(\Gamma, \Sigma) \in R_{CAN}$ iff $\Gamma \subseteq \Sigma$,
$P_n^{\mathfrak{A}} = \{\Gamma \in \boldsymbol{W}_{CAN} \mid p_n \in \Gamma\}$ for any $p_n \in \mathit{Atom}$ and any $n \in N, N \subseteq \mathbb{N}$.

Proposition 1.10. $\mathfrak{A}_{CAN} \in \mathit{Str}(daC)$.

Proof. $\boldsymbol{W}_{CAN}$ and $\langle P_n^{\mathfrak{A}} \rangle_{n \in N, N \subseteq \mathbb{N}}$ clearly meet the corresponding characteristics in Definition 1.3. For R_{CAN}, since it is defined by usual set theory relation $\subseteq$, then it is reflexive and transitive. Also, the Heredity Constraint holds in R_{CAN} by usual characteristics of $\subseteq$ and Definition 1.17. ⊣

The proof of the following facts can be found in (Priest, 2009, pp. 173 and 174).

Lemma 1.11 (Fundamental Lemma). *Let* $\Gamma, \Pi, \Sigma \subseteq Form(daC)$. *If* $\Gamma \nvdash_{daC} \Pi$, *then there is a prime and deductively closed set* Σ *such that* $\Gamma \subseteq \Sigma$ and $\Sigma \nvdash_{daC} \Pi$.

Theorem 1.12. *For* $\Gamma \in \boldsymbol{W}_{CAN}$, $A \in Form(daC)$: $\Gamma \models_{daC} A$ *iff* $A \in \Gamma$

Proposition 1.13. *Let* $\Gamma \subseteq Form(daC)$ *and* $A \in Form(daC)$. *If* $A \in \Gamma$ *for every prime and deductively closed set* Γ, *then* $\vdash_{daC} A$.

Proof. Assume $\nvdash_{daC} A$. From Proposition 1.7 and Definition 1.16, $\nvdash_{daC} \{A\}$. Due to Lemma 1.11, there is a $\Gamma \subseteq Form(daC)$ such that Γ is *prime and deductively closed*, $\emptyset \subseteq \Gamma$ and $\Gamma \nvdash_{daC} \{A\}$. By Definition 1.16 and Proposition 1.8, $\Gamma \nvdash_{daC} A$. Finally, from Proposition 1.9 $A \notin \Gamma$. ⊣

2. Many-Sorted Logic *MSL*

Leaving aside daC for a moment, the logic that will serve as a framework in this translation is presented: Many-Sorted logic or *MSL*. Its main characteristic is the possibility offered by its language and semantics to work with different universes. This is achieved by introducing the concept of *sort*. At a semantic level, this is expressed in the structures presenting more than one universe; in fact, they present as many universes as sorts the researcher may have decided to introduce. This shows, among other things, a very particular characteristic of this logical system: through the concept of signature it is possible to restrict the quantity and characteristics of each sort that one wishes to use; thus, there is no exact number of sorts and universes in Many-Sorted logic, but this is left to free choice and corresponds to the intentions of the researcher who makes use of this logic. However, a general presentation of *MSL* is first made below, and the presentation of *MSL* with a particular signature to be used in the translation is left for the next section.

2.1. *MSL* signatures

Informally, a signature Σ for *MSL* is simply the way to inform which sorts and which operators (beside $\neg, \vee, \approx$, and other ones definable in terms of the former) are used by the researcher; it also informs the domain, codomain, and arity of the operators.

Definition 2.1. A signature Σ is a couple $\Sigma = \langle Sorts, Func \rangle$, where

1. $Sorts = Sorts(\Sigma)$ is a set of indexes such that $0 \in Sorts$.

2. *Func* = *Func*(Σ) is a function with domain *Sym.oper* and image in $(S(Sorts)) \cup (\mathbb{N} - \{0\})$. Here, $\mathbb{N} - \{0\}$ is the set of arities for some elements in *Sym.oper*, $S(Sorts)$ is the set of countable sequences of elements from the set *Sorts*, and *Sym.oper* is an arbitrary countable set of symbols or symbolic operators such that, at least, $\neg, \vee, \approx$ are in *Sym.oper*.
3. $Func(\neg) = \langle 0, 0\rangle$, $Func(\vee) = \langle 0, 0, 0\rangle$ and $Func(\approx) = 2$.
4. In *Sym.oper* only $\neg$ and $\vee$ are f such that $Func(f) = \langle 0, i_1, \ldots, i_n\rangle$ and $0 \in \{i_1, \ldots, i_n\}$.

Intuitively, the indexes in *Sorts* are names that allow grouping all the elements of the same kind. *Func* is a function such that on one hand, if $Func(f) = \langle i_1, \ldots, i_n\rangle$ (i.e., if $Func(f) \in S(Sorts)$), then it informs the operator f domain and codomain; but, on the other hand, if $Func(f) = n$, $n \in \mathbb{N} - \{0\}$, then it assigns the arity of the symbolic operators whose argument has no restrictions about the class of elements to which they apply. Thus, the case $Func(f) = \langle 0, 1, 1\rangle$ means that the operator f is a function that takes two elements, both of sort 1, and returns a sort 0 element. When $Func(f) = \langle i_1, \ldots, i_n\rangle$, the arity of f is $n-1$. However, if $Func(f) = 2$, this means that f has no sort restriction, and its arity is 2.

2.2. *MSL* structures

The sets of universes and functions in an *MSL* structure are also variable; but, they are up to the restrictions made by the signature. So, there should be as many universes as sorts are; and the set of functions should have a counterpart in the structure, too.

DEFINITION 2.2. A structure $\mathfrak{A}$ for *MSL* with a specific signature $\Sigma = \langle Sorts, Func\rangle$ is a couple $\mathfrak{A}^{\Sigma} = \langle \langle \boldsymbol{W}_i\rangle_{i \in Sorts}, \langle f^{\mathfrak{A}^{\Sigma}}\rangle_{f \in Sym.oper}\rangle$ such that

1. $\langle \boldsymbol{W}_i\rangle_{i \in Sorts}$ is a family of non-empty sets such that every $\boldsymbol{W}_i$ is the sort i universe and $\boldsymbol{W}_0 = \{T, F\}$.
2. $\langle f^{\mathfrak{A}^{\Sigma}}\rangle_{f \in Sym.oper}$ is a family of functions such that, at least, $\neg^{\mathfrak{A}^{\Sigma}}$, $\vee^{\mathfrak{A}^{\Sigma}}$, and $\approx^{\mathfrak{A}^{\Sigma}}$ belong to it. Besides, for any $f^{\mathfrak{A}^{\Sigma}}$, different from $\neg^{\mathfrak{A}^{\Sigma}}$ and $\vee^{\mathfrak{A}^{\Sigma}}$, such that $f \in Sym.oper$, there are three options: (1) $Func(f) = \langle 0, i_1, \ldots, i_n\rangle$, (2) $Func(f) = \langle i_0, i_1, \ldots, i_n\rangle$, $i_j \neq 0$ for any j such that $0 \leqslant j \leqslant n$, or (3) $Func(f) = n$, $n \in \mathbb{N} - \{0\}$. In case (1) $f^{\mathfrak{A}^{\Sigma}} \colon \boldsymbol{W}_{i_1} \times \cdots \times \boldsymbol{W}_{i_n} \to \boldsymbol{W}_0$. In case (2) $f^{\mathfrak{A}^{\Sigma}} \colon \boldsymbol{W}_{i_1} \times \cdots \times \boldsymbol{W}_{i_n} \to \boldsymbol{W}_{i_0}$, where $i_0 \neq 0$. Finally, in case (3), $f^{\mathfrak{A}^{\Sigma}} \colon (\bigcup_{i \in Sorts - \{0\}} \boldsymbol{W}_i)^n \to \boldsymbol{W}_0$.

3. For $\neg^{\mathfrak{A}^{\Sigma}}$, $\vee^{\mathfrak{A}^{\Sigma}}$ and $\approx^{\mathfrak{A}^{\Sigma}}$ specific cases, the following holds:
   - $\neg^{\mathfrak{A}^{\Sigma}} : \boldsymbol{W}_0 \to \boldsymbol{W}_0$ such that $\neg^{\mathfrak{A}^{\Sigma}}(T) = F$ and $\neg^{\mathfrak{A}^{\Sigma}}(F) = T$.
   - $\vee^{\mathfrak{A}^{\Sigma}} : \boldsymbol{W}_0 \times \boldsymbol{W}_0 \to \boldsymbol{W}_0$ such that $\vee^{\mathfrak{A}^{\Sigma}}(i^0, j^0) = F$ iff $i^0 = j^0 = F$, with $i^0, j^0 \in \boldsymbol{W}_0$.
   - $\approx^{\mathfrak{A}^{\Sigma}} : (\bigcup_{i \in Sorts-\{0\}} \boldsymbol{W}_i)^2 \to \boldsymbol{W}_0$ such that if $i^i \in \boldsymbol{W}_i$ and $i^j \in \boldsymbol{W}_j$, then $\approx^{\mathfrak{A}^{\Sigma}} (i^i, i^j) = T$ iff $i^i = i^j$.

The class of structures for MSL^{Σ} (i.e. for *MSL* with signature Σ) is denoted $Str(MSL^{\Sigma})$.

It is important to insist that, when $Func(f) = \langle i \rangle$, f is a constant of sort i and $f^{\mathfrak{A}^{\Sigma}} \in \boldsymbol{W}_i$. Equally, when $Func(f) = \langle 0, i_1, \dots, i_n \rangle$ and every $i \neq 0$, f is a relation symbol such that $f^{\mathfrak{A}^{\Sigma}} \subseteq \boldsymbol{W}_{i_1} \times \cdots \times \boldsymbol{W}_{i_n}$.

Remark 2.1. In *MSL*, characteristic functions $f^{\mathfrak{A}^{\Sigma}} : \boldsymbol{W}_{i_1} \times \cdots \times \boldsymbol{W}_{i_n} \to \boldsymbol{W}_0$ are also relations $f^{\mathfrak{A}^{\Sigma}} \subseteq \boldsymbol{W}_{i_1} \times \cdots \times \boldsymbol{W}_{i_n}$.

2.3. *MSL* language

The most interesting characteristic of *MSL* language is the use of different sets of variables: there are as many of these sets as sorts had been defined by the signature. So, a language L for *MSL* with a signature Σ, L^{Σ} is defined as follows:

Definition 2.3. The alphabet for L^{Σ} include:

- Countable and disjoint sets Var^i of variables $x^i_1, x^i_2 \dots$ for every $i \in Sorts - \{0\}$.
- Every symbol $f \in Sym.oper$.
- Existential quantifier $\exists$.

Definition 2.4. The expressions of L^{Σ} are defined recursively:

1. Any variable x^i is an expression of sort i for every $i \in Sorts - \{0\}$.
2. If $f \in Sym.oper$, $Func(f) = \langle i_0, i_1, \dots, i_n \rangle$ and $\varepsilon_1 \dots, \varepsilon_n$ are expressions of sort $i_1, \dots, i_n$, then $f\varepsilon_1, \dots, \varepsilon_n$ is an expression of sort i_0
3. If ε is an expression of sort 0 and x^i is an expression of sort $i \in Sorts - \{0\}$, then $\exists x^i \varepsilon$ is an expression of sort 0.
4. No other combination of elements from the alphabet of L^{Σ} is an expression of L^{Σ}.

The set of terms in L^{Σ}, $Term(MSL^{\Sigma})$ is made up of all expression of sort $i \in Sorts - \{0\}$. The set of formulas of L^{Σ}, $Form(MSL^{\Sigma})$ is made up of every expression of sort 0. These are represented by meta-variables $A, B, \ldots$. Free variables and sentences $Sent(MSL^{\Sigma}) \subseteq Form(MSL^{\Sigma})$ are defined in the usual way with obvious modifications.

Remark 2.2. Regarding the use of parentheses and other punctuation marks, these are used in the traditional way, especially to clear up possible ambiguities. In addition to the above, in the language of *MSL* it is simplified $\exists x_1^{i_1} \ldots \exists x_n^{i_n} A$ as $\exists x_1^{i_1}, \ldots, x_n^{i_n} A$ where $A \in Form(MSL^{\Sigma})$. Furthermore, the following abbreviations are presented: Let $A, B \in Form(MSL^{\Sigma})$. $A \wedge B \stackrel{\text{def}}{=} \neg(\neg A \vee \neg B)$, $A \to B \stackrel{\text{def}}{=} \neg A \vee B$, $A \leftrightarrow B \stackrel{\text{def}}{=} (A \to B) \wedge (B \to A)$, and $\forall x^i A \stackrel{\text{def}}{=} \neg \exists x^i \neg A$.

2.4. Interpretation, validity and consequence in *MSL*

The interpretation in *MSL* is defined in the usual way.

DEFINITION 2.5. Let L^{Σ} be a language for *MSL* and $\mathfrak{A}^{\Sigma}$ be a structure with the same signature Σ. Then an assignment is a function $M\colon \bigcup_{i \in Sorts - \{0\}} Var_i \to \bigcup_{i \in Sorts - \{0\}} \boldsymbol{W}_i$ such that each Var_i is a set of variables x^i of sort i and each $\boldsymbol{W}_i$ is a universe of sort i in $\mathfrak{A}^{\Sigma}$ made up of elements $\boldsymbol{x}^i$ of the same sort. In addition, $M(Var_i) \subseteq \boldsymbol{W}_i$; i.e., the assignment values are in a universe of the same sort as the sort of the variables in the argument of the function.

Note that x^i is a variable symbol, but $\boldsymbol{x}^i$ is an element from $\boldsymbol{W}_i$. The bold type is used here for elements in the structure only.

DEFINITION 2.6. An interpretation $\mathfrak{I}$ of expressions in L^{Σ} over a structure $\mathfrak{A}^{\Sigma}$ is a couple $\mathfrak{I} = \langle \mathfrak{A}^{\Sigma}, M \rangle$ where M is an assignment from L^{Σ} into $\mathfrak{A}^{\Sigma}$ and the following holds:

$$\mathfrak{I}(x^i) = M(x^i)$$

$$\mathfrak{I}(a_i) = a_i^{\mathfrak{A}^{\Sigma}} \text{ such that } a_i^{\mathfrak{A}^{\Sigma}} \in \boldsymbol{W}_i$$

$$\mathfrak{I}(f\tau_1 \ldots \tau_n) = f^{\mathfrak{A}^{\Sigma}}(\mathfrak{I}(\tau_1), \ldots, \mathfrak{I}(\tau_n))$$

$$\mathfrak{I}(R\tau_1 \ldots \tau_n) = R^{\mathfrak{A}^{\Sigma}}(\mathfrak{I}(\tau_1), \ldots, \mathfrak{I}(\tau_n))$$

$$\mathfrak{I}(\neg A) = \neg^{\mathfrak{A}^{\Sigma}} \mathfrak{I}(A)$$

$$\mathfrak{I}(A \vee B) = \vee^{\mathfrak{A}^{\Sigma}}(\mathfrak{I}(A), \mathfrak{I}(B))$$

$$\mathfrak{I}(\exists x^i A) = T \text{ iff } \{\boldsymbol{x}^i \in \boldsymbol{W}_i \mid \mathfrak{I}_{x^i}^{\boldsymbol{x}^i}(A) = T\} \neq \emptyset$$

Note that $\mathfrak{I}^{\boldsymbol{x}^i}_{x^i}(A) = \langle \mathfrak{A}^{\Sigma}, M^{\boldsymbol{x}^i}_{x^i} \rangle (A)$ is an interpretation of A in $\mathfrak{A}^{\Sigma}$ whose function $M^{\boldsymbol{x}^i}_{x^i}$ is an x^i-variant of assignment M which assigns the element $\boldsymbol{x}^i \in \boldsymbol{W}_i$ from $\mathfrak{A}^{\Sigma}$ to the variable x^i of sort i. This can be generalized to $\mathfrak{I}^{\boldsymbol{x}^{i_1}_1 \ldots \boldsymbol{x}^{i_n}_m}_{x^{i_1}_1 \ldots x^{i_n}_m}(A) = \langle \mathfrak{A}^{\Sigma}, M^{\boldsymbol{x}^{i_1}_1 \ldots \boldsymbol{x}^{i_n}_m}_{x^{i_1}_1 \ldots x^{i_n}_m} \rangle (A)$.

Satisfaction, validity and consequence are defined in the usual way. Let $A, B \in Form(MSL^{\Sigma})$.

Definition 2.7 (Satisfaction in MSL). $A \in Form(MSL^{\Sigma})$ is satisfied in $\mathfrak{A}^{\Sigma}$ with an assignment M ($\langle \mathfrak{A}^{\Sigma}, M \rangle \models A$) iff $\langle \mathfrak{A}^{\Sigma}, M \rangle (A) = T$. The set $\Gamma \subseteq Form(MSL^{\Sigma})$ is satisfied in $\mathfrak{A}^{\Sigma}$ with an assignment M (i.e., $\langle \mathfrak{A}^{\Sigma}, M \rangle \models \Gamma$) iff for every formula $A \in \Gamma$, $\langle \mathfrak{A}^{\Sigma}, M \rangle (A) = T$. If A is a sentence, the notation will be simplified to $\mathfrak{A}^{\Sigma} \models A$ and $\mathfrak{A}^{\Sigma} \models \Gamma$. In this case, $\mathfrak{A}^{\Sigma}$ is a model of A (or Γ).

Definition 2.8 (Satisfiable formulas in MSL). A formula A is satisfiable iff $\langle \mathfrak{A}^{\Sigma}, M \rangle (A) = T$ for some $\mathfrak{A}^{\Sigma}$ and M.

Definition 2.9 (Validity in MSL). A formula A is valid in MSL^{Σ} (i.e., $\models_{MSL^{\Sigma}} A$) iff $\langle \mathfrak{A}^{\Sigma}, M \rangle \models A$ for each structure $\mathfrak{A}^{\Sigma}$ and assignment M.

Definition 2.10 (Consequence in MSL). A formula A is consequence of a set $\Gamma \subseteq Form(MSL^{\Sigma})$ (i.e., $\Gamma \models_{MSL^{\Sigma}} A$) iff for every structure $\mathfrak{A}^{\Sigma}$ and assignment M, if $\langle \mathfrak{A}^{\Sigma}, M \rangle \models \Gamma$ then $\langle \mathfrak{A}^{\Sigma}, M \rangle \models A$.

2.5. Sequent calculus for *MSL*

The following is the calculus presented in (Manzano, 1996, pp. 241–242) for *MSL* based on Ebbinghaus' Sequent Calculus with some modifications. Furthermore, some extra rules are presented as derivations of the former ones. Those extra rules are used in order to simplify demonstrations. This calculus will be denoted by $Cal(MSL^{\Sigma})$

Definition 2.11. Let $\{A, B, C\} \cup \Gamma \subseteq Form(MSL^{\Sigma})$. A deduction is a finite and not empty sequence of applications of sequent calculus rules for MSL^{Σ}. In every step, a sequence $A_1, \ldots, A_n \hookrightarrow B$ called *sequent* is obtained, where $A_1, \ldots, A_n$ is the *antecedent* and B is the *consequent*. The rules are the following:

(HI) Hypothesis introduction: $\frac{\emptyset}{\Gamma \hookrightarrow A}$ if $A \in \Gamma$

(M) Monotony: $\frac{\Gamma \hookrightarrow A}{\Gamma^* \hookrightarrow A}$ if $\Gamma \subseteq \Gamma^*$

(PC) Proof by cases: $\frac{\Gamma, B \hookrightarrow A \qquad \Gamma, \neg B \hookrightarrow A}{\Gamma \hookrightarrow A}$

(NC) No contradiction: $\frac{\Gamma, \neg A \hookrightarrow B \qquad \Gamma, \neg A \hookrightarrow \neg B}{\Gamma \hookrightarrow A}$

(IDA) Introducing disjunction in the antecedent: $\frac{\Gamma, A \hookrightarrow C \qquad \Gamma, B \hookrightarrow C}{\Gamma, A \vee B \hookrightarrow C}$

(IDC) Introducing disjunction in the consequent: $a)\frac{\Gamma \hookrightarrow A}{\Gamma \hookrightarrow A \vee B}$, $b)\frac{\Gamma \hookrightarrow A}{\Gamma \hookrightarrow B \vee A}$

(IPA) Intro. individual particularization in the antecedent: $\frac{\Gamma, A(y^i|x^i) \hookrightarrow B}{\Gamma, \exists x^i A \hookrightarrow B}$ if y^i is not a free variable in $\Gamma \cup \{\exists x^i A, B\}$. Note that the sorts of y^i and x^i are the same one.

(IPC) Intro. individual particularization in the consequent: $\frac{\Gamma \hookrightarrow A(\tau|x^i)}{\Gamma \hookrightarrow \exists x^i A}$ if the sorts of τ and x^i are the same one.

(RE) Reflexivity of equality for individuals: $\frac{\emptyset}{\tau \approx \tau}$

(ES) Equals substitution for individuals: $\frac{\Gamma \hookrightarrow A(\tau_1|x^i)}{\Gamma, \tau_1 \approx \tau_2 \hookrightarrow A(\tau_2|x^i)}$ if the sorts of x^i and τ_1 are the same one iff so are the sorts of x^i and τ_2.

DEFINITION 2.12. If a sequent $A_1, \ldots, A_n \hookrightarrow B$ is obtained by the single application of the above rules, then $A_1, \ldots, A_n \hookrightarrow B$ is derivable in the calculus, and it is written $\vdash A_1, \ldots A_n \hookrightarrow B$.

DEFINITION 2.13. Let $\{A\} \cup \Gamma \subseteq Form(MSL^{\Sigma})$. $\Gamma \vdash_{MSL^{\Sigma}} A$ iff there is a finite subset $\Pi \subseteq \Gamma$, such that $\Pi \hookrightarrow A$ is derivable.

PROPOSITION 2.1. *Let $\{A, B, C\} \cup \Gamma \subseteq Form(MSL^{\Sigma})$. The following are rules derived from Definition 2.11:*

(DN) *Double negation*: $\frac{\Gamma \hookrightarrow A}{\Gamma \hookrightarrow \neg\neg A}$

(ED) *Eliminating disjunction*: $\frac{\Gamma \hookrightarrow A \vee B \qquad \Gamma \hookrightarrow \neg A}{\Gamma \hookrightarrow B}$

(IC) *Introducing conjunction*: $\frac{\Gamma \hookrightarrow A \qquad \Gamma \hookrightarrow B}{\Gamma \hookrightarrow A \wedge B}$

(EC) *Eliminating conjunction*: $\frac{\Gamma \hookrightarrow A \wedge B}{\Gamma \hookrightarrow A}$

(MP) *Modus Ponens*: $\frac{\Gamma \hookrightarrow A \to B \qquad \Gamma \hookrightarrow A}{\Gamma \hookrightarrow B}$

(DT) *Deduction Theorem*: $\frac{\Gamma, A \hookrightarrow B}{\Gamma \hookrightarrow A \to B}$

(DR) *Deduction rule*: $\frac{\Gamma \hookrightarrow A \to (B \to C)}{\Gamma \hookrightarrow (A \wedge B) \to C}$

(deM) *De Morgan*[2]: $\frac{\Gamma \hookrightarrow \neg(A \vee B)}{\Gamma \hookrightarrow \neg A \wedge \neg B}$

[2] Other forms of *de Morgan rule* are derivable; nevertheless, only this one is required here besides the one already presented in Remark 2.2.

(EGC) *Elim. individual generalization in the consequent*: $\frac{\Gamma \hookrightarrow \forall x^i A}{\Gamma \hookrightarrow A(\tau|x^i)}$ *if the sorts of τ and x^i are the same one.*

(IGA) *Intro. individual generalization in the antecedent*: $\frac{\Gamma, A(\tau|x^i) \hookrightarrow B}{\Gamma, \forall x^i A \hookrightarrow B}$ *if the sorts of τ and x^i are the same one.*

(IGC) *Intro. individual generalization in the consequent*: $\frac{\Gamma \hookrightarrow A(y^i|x^i)}{\Gamma \hookrightarrow \forall x^i A}$ *if y^i is not a free variable in $\Gamma \cup \{\forall x^i A\}$. Note that the sorts of x^i and y^i are the same one.*

(EPC) *Elim. individual particularization in the consequent*: $\frac{\Gamma \hookrightarrow \exists x^i A}{\Gamma \hookrightarrow A(y^i|x^i)}$ *if y^i is not a free variable in $\Gamma \cup \{\exists x^i A\}$ Note that the sorts of x^i and y^i are the same one.*

(EGA) *Elim. individual generalization in the antecedent*: $\frac{\Gamma, \forall x^i A \hookrightarrow B}{\Gamma, A(y^i|x^i) \hookrightarrow B}$ *if y^i is not a free variable in $\Gamma \cup \{B\}$. Note that the sorts of x^i and y^i are the same one.*

(NP) *Negation of individual particularization*: $\frac{\Gamma \hookrightarrow \neg \exists x^i A}{\Gamma \hookrightarrow \forall x^i \neg A}$

Proof. This is a simple but very long exercise. The proofs for some of the rules are very similar to the demonstrations for a Classical First Order sequent calculus in (Manzano, 1989, pp. 110–114). ⊣

2.6. Some important theorems for *MSL*

Manzano (1996) proves Compactness, Löwenheim-Skolem, Soundness and Completeness theorems for *MSL*, which will be exported into *daC* due to the translation developed in this paper. The proof of the following theorems can be found in (Manzano, 1996, pp. 244–256, 256–257).

Theorem 2.2 (Compactness). *Let $\Gamma \cup \{A\} \subseteq Form(MSL^{\Sigma})$. Then if $\Gamma \models_{MSL^{\Sigma}} A$, there is a finite set $\Pi \subseteq \Gamma$ such that $\Pi \models_{MSL^{\Sigma}} A$.*

Theorem 2.3 (Löwenheim-Skolem). *Let $\Gamma \subseteq Form(MSL^{\Sigma})$. If Γ has a model, then it has a countable model.*

Theorem 2.4 (Soundness). *Let $\Gamma \cup \{A\} \subseteq Form(MSL^{\Sigma})$. If $\Gamma \vdash_{MSL^{\Sigma}} A$ then $\Gamma \models_{MSL^{\Sigma}} A$.*

Theorem 2.5 (Completeness). *Let $\Gamma \cup \{A\} \subseteq Form(MSL^{\Sigma})$. If $\Gamma \models_{MSL^{\Sigma}} A$, then $\Gamma \vdash_{MSL^{\Sigma}} A$.*

2.7. *MSL* with a particular signature for the translation of *daC*

As expected, *MSL* with a particular signature Σ^*, MSL^*, is presented in the following. The signature Σ^* is designed for the particular translation developed in this paper.

Let P_i be a constant symbol of signature $\langle 0, 1\rangle$ for $i \in I, I \subseteq \mathbb{N}$, and let R, ε and $\approx$ be binary relation symbols.

Definition 2.14 (Signature Σ^*). The signature $\Sigma^* = \langle Sorts, Func^*\rangle$ such that $Sorts = \langle 0, i, j\rangle$, $Sym.oper^* = \{\langle P_n\rangle_{n\in N, N\subseteq\mathbb{N}}, R, \approx, \varepsilon, \vee, \neg\}$ and $Func^* \colon Sym.oper^* \to S(Sorts) \cup \mathbb{N} - \{0\}$ where

$$\begin{aligned}
&Func^*(P_n) = \langle j\rangle \text{ for } n \in N, N \subseteq \mathbb{N}\\
&Func^*(R) = \langle 0, i, i\rangle\\
&Func^*(\neg) = \langle 0, 0\rangle\\
&Func^*(\vee) = \langle 0, 0, 0\rangle\\
&Func^*(\varepsilon) = \langle 0, i, j\rangle\\
&Func^*(\approx) = 2
\end{aligned}$$

This means that the signature has a set *Sorts* composed of three indices: index 0 for the class of truth values, and indices i and j. Moreover, the function $Func^*$ indicates that each P_n is a constant symbol of *sort* j, ε is a relation symbol that represents a relation between an element of *sort* i and another of *sort* j, which, when applied between them, receives a truth value, that is, it is of *sort* 0. Furthermore, R is a relation symbol between two individual symbols of *sort* i which, when applied to them, receives a truth value; that is, it is of *sort* 0.

Definition 2.15 (Language L^*). The alphabet of L^* contains the symbols in *Oper.sim* in addition to the existential quantifier $\exists$ and two countable, disjoint sets of variables $Var_{\boldsymbol{W}_i} = \{w, v, u, \dots, w_1, w_2, \dots\}$ as variables of *sort* i and $Var_{\boldsymbol{W}_j} = \{x, y, z, \dots, x_1, x_2, \dots\}$ as variables of *sort* j. As in the initial presentation of *MSL*, the logical connectives $\wedge, \to$, and $\leftrightarrow$, as well as the quantifier $\forall$, are taken as abbreviations with their usual definitions. Expressions, terms, and formulas are defined as before, with the obvious modifications. Likewise, the class of formulas of this language will be denoted $Form(MSL^*)$.

Definition 2.16 (Structure $\mathfrak{A}^*$). The structure $\mathfrak{A}^*$ for MSL^* is

$$\mathfrak{A}^* = \langle \boldsymbol{W}_0, \boldsymbol{W}_i, \boldsymbol{W}_j, \langle P_n^{\mathfrak{A}^*}\rangle_{n\in N, N\subseteq\mathbb{N}}, R^{\mathfrak{A}^*}, \approx^{\mathfrak{A}^*}, \varepsilon^{\mathfrak{A}^*}, \vee^{\mathfrak{A}^*}, \neg^{\mathfrak{A}^*}\rangle$$

such that

1. $\boldsymbol{W}_i$ and $\boldsymbol{W}_j$ are two distinct and disjoint universes.
2. $P_n^{\mathfrak{A}^*} \in \boldsymbol{W}_j$ for each $n \in N$.
3. $R^{\mathfrak{A}^*} : \boldsymbol{W}_i^2 \to \boldsymbol{W}_0$ is a binary relation between elements of $\boldsymbol{W}_i$.
4. $\varepsilon^{\mathfrak{A}^*} : \boldsymbol{W}_i \times \boldsymbol{W}_j \to \boldsymbol{W}_0$ is a relation between an element of *sort* i and another of *sort* j.
5. $\boldsymbol{W}_0, \approx^{\mathfrak{A}^*}, \vee^{\mathfrak{A}^*}$, and $\neg^{\mathfrak{A}^*}$ satisfy the properties of the definition of *MSL* previously presented.

The class of structures for MSL^* is denoted by $Str(MSL^*)$.

3. Translation from *daC* into *MSL*

Now that both logics involved have been displayed, it is time to present the translation itself and demonstrate that it meets the three levels outlined in the introduction. Hence, following the strategy from (Manzano, 1996), the translation function from *daC* formulas into MSL^* formulas is presented, beside a function to convert *daC* structures into MSL^* structures modifying as little as possible.

Afterward, in the first level of the translation, the Representation Theorem is proved ($\models_{daC} A$ iff $\Delta \models_{MSL^*} \forall w(t(A)[w])$); i.e, it is proved that the translation preserves the set of valid formulas of *daC* when a set $\Delta \subset Form(MSL^*)$ is assumed. Intuitively, Δ works as a set of axioms used to accurately represent *daC* into MSL^*. To achieve this, first, some theorems and propositions must be demonstrated in order to grant that the functions defined do not modify the meaning of *daC* formulas.

In the second level of the translation, it is time for the Main Theorem ($\Gamma \models_{daC} A$ iff $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$); i.e., it is demonstrated that the translation preserves the set of valid arguments modulo Δ. Finally, in the last level, the calculi equivalence is proved ($\Gamma \vdash_{daC} A$ iff $t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w]$); i.e., it is proved that the translation preserves the set of derivations from $Cal(daC)$ in $Cal(MSL^*)$, modulo Δ. In order to achieve this, on one hand, it is shown that the set of rules of $Cal(daC)$ are translated into correct derivations in $Cal(MSL^*)$, if Δ is assumed. On the other hand, it is proved that the set of derivations of $Cal(MSL^*)$ modulo Δ correctly represents the set of derivations of $Cal(daC)$. To demonstrate this, the *daC* Canonical Model presented above is used.

3.1. Preliminaries: translation and structure conversion functions

Definition 3.1. The function $t\colon Form(daC) \to Form(MSL^*)$ assigns to every formula $A \in Form(daC)$ a single formula $t(A)[w] \in Form(MSL^*)$ with free variable w in the language of MSL^* as follows:

$$t(p_n)[w] = \varepsilon w P_n$$
$$t(\sim A)[w] = \exists u(Ruw \land \neg t(A)[u])$$
$$t(A \rightarrowtail B)[w] = \forall u(Rwu \to (t(A)[u] \to t(B)[u]))$$
$$t(A \land B)[w] = t(A)[w] \land t(B)[w]$$
$$t(A \lor B)[w] = t(A)[w] \lor t(B)[w]$$

Remark 3.1. It is possible to apply the above function to set of formulas from daC: let $\Gamma \subseteq Form(daC)$. $t(\Gamma)[w] = \{t(A)[w] \mid A \in \Gamma\}$.

Definition 3.2. Let $\mathfrak{A} = \langle \boldsymbol{W}, R, \langle P_n^{\mathfrak{A}} \rangle_{n \in N, N \subseteq \mathbb{N}} \rangle \in Str(daC)$ and $\mathfrak{A}^* \in Str(MSL^*)$. $Conv_1(\mathfrak{A}) = \mathfrak{A}^*$, such that:

$$\mathfrak{A}^* = \langle \boldsymbol{W}_0, \boldsymbol{W}_i, \boldsymbol{W}_j, \langle P_n^{\mathfrak{A}^*} \rangle_{n \in N, N \subseteq \mathbb{N}}, R^{\mathfrak{A}^*}, \approx^{\mathfrak{A}^*}, \varepsilon^{\mathfrak{A}^*}, \vee^{\mathfrak{A}^*}, \neg^{\mathfrak{A}^*} \rangle,$$

where

1. $\boldsymbol{W}_0 = \{T, F\}$.
2. $\boldsymbol{W}_i = \boldsymbol{W}$ from $\mathfrak{A}$.
3. $P_n^{\mathfrak{A}^*} = P_n^{\mathfrak{A}}$ for each $n \in N \subseteq \mathbb{N}$.
4. $R^{\mathfrak{A}^*} = R$.
5. $\boldsymbol{W}_j$ is the smallest set satisfying the following conditions:
   Let $\overline{R^{\mathfrak{A}^*}} = \{(\boldsymbol{w}, \boldsymbol{u}) \in \boldsymbol{W}_i^2 \mid (\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{A}^*}\}$.
   (i) $\emptyset, \boldsymbol{W}_i \in \boldsymbol{W}_j$.
   (ii) $P_n^{\mathfrak{A}^*} \in \boldsymbol{W}_j$ for each atom p_n of the daC language.
   (iii) If $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{W}_j$, then $\boldsymbol{x} \cup \boldsymbol{y}, \boldsymbol{x} \cap \boldsymbol{y} \in \boldsymbol{W}_j$.
   (iv) If $\boldsymbol{x} \in \boldsymbol{W}_j$, then $Dom(\overline{R^{\mathfrak{A}^*}} \cap (\boldsymbol{W}_i \times (\boldsymbol{W}_i - \boldsymbol{x}))) \in \boldsymbol{W}_j$.
   (v) If $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{W}_j$, then $\boldsymbol{W}_i - Dom(R^{\mathfrak{A}^*} \cap (\boldsymbol{W}_i \times (\boldsymbol{x} - \boldsymbol{y}))) \in \boldsymbol{W}_j$.
6. $\varepsilon^{\mathfrak{A}^*} = \{(\boldsymbol{w}, \boldsymbol{x}) \in \boldsymbol{W}_i \times \boldsymbol{W}_j \mid \boldsymbol{w} \in \boldsymbol{x}\}$.
7. The functions $\approx^{\mathfrak{A}^*}$, $\vee^{\mathfrak{A}^*}$, and $\neg^{\mathfrak{A}^*}$ are the same as those presented above.

The class of those $\mathfrak{A}^* \in Str(MSL^*)$ such that $\mathfrak{A}^* = Conv_1(\mathfrak{A})$ for some $\mathfrak{A} \in Str(daC)$ is denoted $Conv_1(Str(daC))$.

Claim 3.1. *In the above definition, since $R^{\mathfrak{A}^*}$ is defined by R in $\mathfrak{A}$, then $(\boldsymbol{w}, \boldsymbol{u}) \in R^{\mathfrak{A}^*}$ iff $(\boldsymbol{w}, \boldsymbol{u}) \in R$. Hence, $R^{\mathfrak{A}^*}$ is also reflexive and transitive, and Heredity Constraint holds.*

We now proceed to prove some important propositions for the first level of translation proposed by Manzano. These theorems are intended to show that the function $Conv_1$, together with the translation function t, preserves the set of formulas satisfied by the original daC-structures.

Proposition 3.2. *Let* $\mathfrak{A}^* = Conv_1(\mathfrak{A})$ *for a* daC *structure* $\mathfrak{A}$ *and let* $A^{\mathfrak{A}}$ *be the set of worlds in* $\mathfrak{A}$ *which satisfy* $A \in Form(daC)$. *For any* A*:*

$$A^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W}_1 \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(A)[w]) = T\}.$$

Proof. Let $\overline{R^{\mathfrak{A}^*}} = \{(\boldsymbol{w}, \boldsymbol{u}) \in \boldsymbol{W}_i^2 \mid (\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{A}^*}\}$. The proof is over the construction of formulas in daC.

Base case, $P^{\mathfrak{A}}$: $\boldsymbol{w} \in P^{\mathfrak{A}}$ iff $\boldsymbol{w} \in P^{\mathfrak{A}^*}$ iff $(\boldsymbol{w}, P^{\mathfrak{A}^*}) \in \varepsilon^{\mathfrak{A}^*}$ (by Definition 3.2) iff $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(\varepsilon w P) = T$ (by Definition 2.6) iff $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(p)[w])$ $= T$ iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(p)[w]) = T\}$ (by Definition 3.1).

Let $A, B \in Form(daC)$ and $\boldsymbol{w} \in \boldsymbol{W}_i$. Take as induction hypothesis that $A^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(A)[w]) = T\}$ for any $A \in Form(daC)$.

Case $(\sim A)^{\mathfrak{A}}$: By Definitions 1.5 and 3.2,

- $\boldsymbol{w} \in (\sim A)^{\mathfrak{A}}$ iff $\boldsymbol{w} \in Dom(\overline{R^{\mathfrak{A}^*}} \cap (\boldsymbol{W}_i \times (\boldsymbol{W}_i - A^{\mathfrak{A}})))$.

From basic set theory definitions, induction hypothesis, and Definition 2.6, this is so iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{wu}}_{wu}(\exists u(Ruw \wedge \neg t(A)[u])) = T\}$. Finally, from Definition 3.1, this happens iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(\sim A)[w]) = T\}$.

Case $(A \rightarrowtail B)^{\mathfrak{A}}$: By Definitions 1.5 and 3.2, $\boldsymbol{w} \in (A \rightarrowtail B)^{\mathfrak{A}}$ iff $\boldsymbol{w} \in \boldsymbol{W}_i - Dom(R^{\mathfrak{A}^*} \cap (\boldsymbol{W}_i \times (A^{\mathfrak{A}} - B^{\mathfrak{A}})))$. From basic set theory definitions, induction hypothesis, and Definition 2.6, this is so iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{wu}}_{wu}(\forall u(Rwu \to (t(A)[u] \to t(B)[u]))) = T\}$. Finally, from Definition 3.1, this is so iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(A \rightarrowtail B)[w]) = T\}$.

Case $(A \wedge B)^{\mathfrak{A}}$: By Definition 1.5, $\boldsymbol{w} \in (A \wedge B)^{\mathfrak{A}}$ iff $\boldsymbol{w} \in A^{\mathfrak{A}} \cap B^{\mathfrak{A}}$. From basic set theory definitions, induction hypothesis, and Definitions 3.2 and 2.6, this is so iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(A)[w] \wedge t(B)[w]) = T\}$. Finally, by Definition 3.1, this happens iff $\boldsymbol{w} \in \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_{w}(t(A \wedge B)[w]) = T\}$.

The case for $(A \vee B)^{\mathfrak{A}}$ is similar, with obvious modifications. ⊣

Lemma 3.3. *Let* $\mathfrak{A} \in Str(daC)$, $\mathfrak{A}^* = Conv_1(\mathfrak{A}) \in Str(MSL^*)$, *and* $A \in Form(daC)$. *Then*

(a) $\mathfrak{A}, \boldsymbol{w} \models A$ *iff* ${\mathfrak{A}^*}^{\boldsymbol{w}}_{w}(t(A)[w]) = T$.
(b) $\mathfrak{A} \models A$ *iff* $\mathfrak{A}^*$ *is a model of* $\forall w(t(A)[w])$ *in* MSL^*.

PROOF. (a) $\mathfrak{A}, \boldsymbol{w} \models A$ iff $\boldsymbol{w} \in A^{\mathfrak{A}}$ iff $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(A)[w]) = T$ (from Definition 1.5 and Proposition 3.2).

(b) $\mathfrak{A} \models A$ iff for every $\boldsymbol{w} \in \boldsymbol{W}$, $\boldsymbol{w} \models A$ (by Definition 1.4) iff for every $\boldsymbol{w} \in \boldsymbol{W}$, $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(A)[w]) = T$ (by literal a) iff for every $\boldsymbol{w} \in \boldsymbol{W}_i$, $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(A)[w]) = T$ (by Definition 3.2) iff $Conv_1(\mathfrak{A}) \models \forall w(t(A)[w])$ (by Definition 2.6 and Remark 2.2). ⊣

THEOREM 3.4. *Let $Conv_1(Str(daC))$ be the class of $\mathfrak{A}^* \in Str(MSL^*)$ such that $\mathfrak{A}^* = Conv_1(\mathfrak{A})$ for the structure $\mathfrak{A} \in Str(daC)$; and let $A \in Form(daC)$. Then*

$$\models_{daC} A \textit{ iff } \models_{Conv_1(Str(daC))} \forall w(t(A)[w]).$$

PROOF. By Lemma 3.3, $\mathfrak{A} \models A$ iff $\mathfrak{A}^* \models \forall w(t(A)[w])$. Now, due to $\mathfrak{A}^*$ is any structure in $Conv_1(Str(daC))$, $\mathfrak{A}$ is any structure in $Str(daC)$ and Definition 1.8, it follows that $\models_{daC} A$ iff $\models_{Conv_1(Str(daC))} \forall w(t(A)[w])$. ⊣

3.2. First level: Representation Theorem

A set $\Delta \subseteq Form(MSL^*)$ works in this translation as a kind of set of axioms within MSL^* used to represent daC as a theory inside the latter. Specifically, Δ characterizes the class of MSL^* structures used to represent daC (i.e. $Mod(\Delta) \subseteq Str(MSL^*)$). Note that, up to now, it is known that the function $Conv_1$ together with the translation function t preserves the set of valid formulas of daC as long as the semantics is restricted to the class of images of $Conv_1$ (i.e. $Conv_1(Str(daC)) \subseteq Str(MSL^*)$). Ideally, the class of MSL^* structures characterized by Δ should be equal to the class obtained by applying $Conv_1$ to the structures of daC; however, this does not happen here, so in this section we also define the function H which adjusts the structures in $Mod(\Delta)$ so that this class coincides with $Conv_1(Str(daC))$.

Returning to Δ, it will be used in most of the following proofs. The first formula of Δ states that the relation R is reflexive; the second, that R is transitive. The third formula asserts the Heredity Constraint. The fourth is the extensionality axiom for ε and $\approx$, ensuring that ε behaves like set–membership ($\in$). Finally, the last formula is actually the comprehension schema for translations of each $A \in Form(daC)$; it guarantees that for every formula of daC there is a corresponding set in the MSL^* structures. By the end of this section, the Representation

Theorem will show that the proposed translation preserves the set of valid *daC*-formulas, modulo Δ. Unlike earlier results, we will now have the axiomatization of the fragment of MSL^* representing *daC*.

Definition 3.3. Let Δ be the set consisting of the following formulas of MSL^*: $\forall w Rww$, $\forall wuv((Rwu \wedge Ruv) \to Rwv)$, $\forall x \forall wu((\varepsilon wx \wedge Rwu) \to \varepsilon ux)$, $\forall xy(\forall w(\varepsilon wx \leftrightarrow \varepsilon wy) \to x \approx y)$, $\exists x \forall w(\varepsilon wx \leftrightarrow t(A)[w])$ for each $A \in Form(daC)$.

Since it is not yet guaranteed that $Mod(\Delta) = Conv_1(Str(daC))$, before the proof of the Representation Theorem we must define the function H. This function adjusts the class of MSL^* structures that are models of Δ so that they lie in the image of $Conv_1$. In other words, if $\mathfrak{A}^* \in Mod(\Delta)$, then there exists $\mathfrak{A} \in Str(daC)$ such that $H(\mathfrak{A}^*) = Conv_1(\mathfrak{A})$. It is also crucial to show that $H(\mathfrak{A}^*)$ is isomorphic to the original $\mathfrak{A}^*$, ensuring they satisfy exactly the same MSL^* formulas.

Definition 3.4. We define the function

$$H\colon Mod(\Delta) \to Str(MSL^*), \quad \mathfrak{A}^* \mapsto \mathfrak{B}^*,$$

by setting for any

$$\mathfrak{A}^* = \langle \boldsymbol{W}_0, \boldsymbol{W}_i, \boldsymbol{W}_j, \langle P_n^{\mathfrak{A}^*} \rangle_{n \in N, N \subseteq \mathbb{N}}, R^{\mathfrak{A}^*}, \approx^{\mathfrak{A}^*}, \varepsilon^{\mathfrak{A}^*}, \vee^{\mathfrak{A}^*}, \neg^{\mathfrak{A}^*} \rangle,$$
$$\mathfrak{B}^* = \langle \boldsymbol{W}_0^{\mathfrak{B}^*}, \boldsymbol{W}_i^{\mathfrak{B}^*}, \boldsymbol{W}_j^{\mathfrak{B}^*}, \langle P_n^{\mathfrak{B}^*} \rangle_{n \in N, N \subseteq \mathbb{N}}, R^{\mathfrak{B}^*}, \approx^{\mathfrak{B}^*}, \varepsilon^{\mathfrak{B}^*}, \vee^{\mathfrak{B}^*}, \neg^{\mathfrak{B}^*} \rangle$$

such that:

1. $\boldsymbol{W}_0^{\mathfrak{B}^*} = \boldsymbol{W}_0$ and $\boldsymbol{W}_i^{\mathfrak{B}^*} = \boldsymbol{W}_i$.
2. $\boldsymbol{W}_j^{\mathfrak{B}^*} = \{\{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid (\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}\} \mid \boldsymbol{x} \in \boldsymbol{W}_j\}$.
3. $\varepsilon^{\mathfrak{B}^*} = \{(\boldsymbol{w}, \boldsymbol{x}) \in \boldsymbol{W}_i^{\mathfrak{B}^*} \times \boldsymbol{W}_j^{\mathfrak{B}^*} \mid \boldsymbol{w} \in \boldsymbol{x}\}$.
4. $P_n^{\mathfrak{B}^*} = P_n^{\mathfrak{A}^*}$ for each $n \in N$, $N \subseteq \mathbb{N}$.
5. $R^{\mathfrak{B}^*} = R^{\mathfrak{A}^*}$, $\vee^{\mathfrak{B}^*} = \vee^{\mathfrak{A}^*}$, $\neg^{\mathfrak{B}^*} = \neg^{\mathfrak{A}^*}$, and $\approx^{\mathfrak{B}^*} = \approx^{\mathfrak{A}^*}$.

Clearly, H is a function; that is, there is a unique element $\mathfrak{B}^* = H(\mathfrak{A}^*)$, with $\mathfrak{A}^* \in Mod(\Delta)$ since conditions 1–5 are univocal.

Proposition 3.5. *Let $\mathfrak{A}^* \in Mod(\Delta)$ be a many-sorted structure of signature Σ^*, and let $\mathfrak{B}^* = H(\mathfrak{A}^*)$. Then $\mathfrak{A}^*$ and $\mathfrak{B}^*$ are isomorphic; i.e. there exists an isomorphism f between them.*

Proof. We define $f\colon \bigcup_{k \in Sorts} \boldsymbol{W}_k \to \bigcup_{k \in Sorts} \boldsymbol{W}_k^{\mathfrak{B}^*}$, where:

1. $f \upharpoonright \boldsymbol{W}_0$ and $f \upharpoonright \boldsymbol{W}_i$ are the identity.
2. For each $\boldsymbol{x} \in \boldsymbol{W}_j$, $f(\boldsymbol{x}) = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid (\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}\}$.

The proof begins by demonstrating that f is surjective. From the definition of f, each element in $\boldsymbol{W}_i^{\mathfrak{B}^*}$ is identical to an element in $\boldsymbol{W}_i$, with $i \in Sorts - \{j\}$. Moreover, every element in $\boldsymbol{W}_j^{\mathfrak{B}^*}$ is of the form $f(\boldsymbol{x})$, with $\boldsymbol{x} \in \boldsymbol{W}_j$, by item 2 of the definition of f for all $\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$.

Next, it is demonstrated that f is injective. The case for $f \upharpoonright \boldsymbol{W}_0$ and $f \upharpoonright \boldsymbol{W}_i$ is immediate. In the remaining case, assume that $f(\boldsymbol{x}) = f(\boldsymbol{y})$ for two $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{W}_j$. By item 2 in the definition of f, $(\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}$ iff $(\boldsymbol{w}, \boldsymbol{y}) \in \varepsilon^{\mathfrak{A}^*}$. Then, by definitions in *MSL* 2.6 of interpretation, 2.2 of structures and Remark 2.2 on abbreviations in the language, $\langle \mathfrak{A}^*, M_{xyw}^{\boldsymbol{xyw}} \rangle \models \varepsilon wx \leftrightarrow \varepsilon wy$. Moreover, since $\mathfrak{A}^* \models \forall xy(\forall w(\varepsilon wx \leftrightarrow \varepsilon wy) \to x \approx y)$ by Definition 3.3 of Δ, it is possible to conclude that $\langle \mathfrak{A}^*, M_{xyw}^{\boldsymbol{xyw}} \rangle \models x \approx y$; now, by Definitions 2.6 and 2.2 of interpretation and structures in *MSL*, $\boldsymbol{x} = \boldsymbol{y}$. All this means that f is a bijection.

Likewise, f is an isomorphism; i.e., for each $\boldsymbol{i}_m \in \boldsymbol{W}_k$ (with $k \in Sorts$, $1 \leqslant m \leqslant h$) and any $g^{\mathfrak{A}^*} \in \{R^{\mathfrak{A}^*}, \approx^{\mathfrak{A}^*}, \varepsilon^{\mathfrak{A}^*}, \vee^{\mathfrak{A}^*}, \neg^{\mathfrak{A}^*}\}$, it is the case that $g^{\mathfrak{A}^*}(\boldsymbol{i}_1, \ldots, \boldsymbol{i}_h) = T$ iff $g^{\mathfrak{B}^*}(f(\boldsymbol{i}_1), \ldots, f(\boldsymbol{i}_h)) = T$. The cases for $\vee^{\mathfrak{A}^*}$, $\neg^{\mathfrak{A}^*}$ and $R^{\mathfrak{A}^*}$ are immediate. For $\approx^{\mathfrak{A}^*}$, let $\boldsymbol{i}^p \in \boldsymbol{W}_p$ and $\boldsymbol{i}^q \in \boldsymbol{W}_q$ for $p, q \in Sorts$. Now, $\approx^{\mathfrak{A}^*} (\boldsymbol{i}^p, \boldsymbol{i}^q) = T$ iff $\boldsymbol{i}^p = \boldsymbol{i}^q$ iff $f(\boldsymbol{i}^p) = f(\boldsymbol{i}^q)$ iff $\approx^{\mathfrak{A}^*} (f(\boldsymbol{i}^p), f(\boldsymbol{i}^q)) = T$ iff $\approx^{\mathfrak{B}^*} (f(\boldsymbol{i}^p), f(\boldsymbol{i}^q)) = T$ (by Definitions 2.16, 2.2 and 3.4). For the remaining case, recall Remark 2.1. Therefore, $(\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}$ iff $(f(\boldsymbol{w}), \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}$ iff $f(\boldsymbol{w}) \in f(\boldsymbol{x})$ iff $(f(\boldsymbol{w}), f(\boldsymbol{x})) \in \varepsilon^{\mathfrak{B}^*}$ (by the definitions of f and H).

In summary, since there exists an isomorphism f between $\mathfrak{A}^* \in Mod(\Delta)$ and $\mathfrak{B}^*$, then these structures are isomorphic. Furthermore, given that $\varepsilon^{\mathfrak{B}^*}$ is defined using $\in$, it is possible to suppose that $\varepsilon^{\mathfrak{B}^*}$ behaves exactly like the membership relation. This also implies the same for $\varepsilon^{\mathfrak{A}^*}$ when restricted to $Mod(\Delta)$. ⊣

PROPOSITION 3.6. *Let $\mathfrak{A}^* \in Mod(\Delta)$ and $H(\mathfrak{A}^*)$, the structure obtained by H. Moreover, let f be the isomorphism between them from Proposition 3.5, let M be a variable assignment in $\mathfrak{A}^*$, and let $A \in Form(MSL^*)$.*

$$\langle \mathfrak{A}^*, M \rangle \models A \textit{ iff } \langle H(\mathfrak{A}^*), f \circ M \rangle \models A.$$

PROOF. The proof is by induction on the construction of formulas in MSL^*. For this, four base cases are considered: (1) εwx, (2) εwP_n, (3) Rwu, and (4) $i^p \approx i^q$, with $p, q \in Sorts$. In all of these, the proof structure is identical: use Definition 2.7 of satisfaction in *MSL*, Definition 2.6 of interpretation in *MSL*, and Proposition 3.5 establishing the isomorphism between $\mathfrak{A}^*$ and $H(\mathfrak{A}^*)$. The idea here is to exploit that both structures are isomorphic to show that the base-case formulas are satisfied in structure $\mathfrak{A}^*$ if and only if they are satisfied in $H(\mathfrak{A}^*)$. As an example, we present base case (1); the others are similar with obvious modifications:

- $\langle\mathfrak{A}^*, M\rangle \models \varepsilon wx$ iff $\langle\mathfrak{A}^*, M\rangle(\varepsilon wx) = T$ (by Definition 2.7 of satisfaction in *MSL*) iff $\varepsilon^{\mathfrak{A}^*}(M(w), M(x)) = T$ (by Definition 2.6 of interpretation in *MSL*) iff $\varepsilon^{H(\mathfrak{A}^*)}(f \circ M(w), f \circ M(x)) = T$ (since both structures are isomorphic by Proposition 3.5) iff $\langle H(\mathfrak{A}^*), f \circ M\rangle(\varepsilon wx) = T$ (by Definition 2.6 of interpretation in *MSL*) iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \models \varepsilon wx$ (by Definition 2.7 of satisfaction in *MSL*).

As induction hypothesis, assume $\langle\mathfrak{A}^*, M\rangle \models A$ iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \models A$ for $A \in \mathit{Form}(\mathit{MSL}^*)$. The next induction cases are (5) $\neg A$ and (6) $A \vee B$. As with the base cases, both proof structures are identical: use Definition 2.7 of satisfaction in *MSL*, Definition 2.6 of interpretation in *MSL*, Definition 2.2 of structure in *MSL*, and the induction hypothesis. We present case (5) as an example; the other is similar with obvious modifications:

- $\langle\mathfrak{A}^*, M\rangle \models \neg A$ iff $\langle\mathfrak{A}^*, M\rangle \not\models A$ by Definitions 2.7, 2.6, and 2.2 of satisfaction, interpretation, and structures in *MSL*. This is the case iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \not\models A$ (by the induction hypothesis) iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \models \neg A$ by the same preceding definitions.

The remaining case is (7) $\exists i^k A$ with $i^k \in \mathit{Var}(\boldsymbol{W}_k)$, $k \in \mathit{Sorts}$. This is similar to the previous cases, except that it does not use Definition 2.2:

- $\langle\mathfrak{A}^*, M\rangle \models \exists i^k A$ iff there exists at least one $\boldsymbol{i}^k \in \boldsymbol{W}_k$ such that $\langle\mathfrak{A}^*, M_{i^k}^{\boldsymbol{i}^k}\rangle \models A$ by Definitions 2.7 and 2.6 of satisfaction and interpretation in *MSL*. Now, this is the case iff there exists at least one $f(\boldsymbol{i}^k) \in \boldsymbol{W}_k^{H(\mathfrak{A}^*)}$ such that $\langle H(\mathfrak{A}^*), f \circ M_{i^k}^{f(\boldsymbol{i}^k)}\rangle \models A$ (by the induction hypothesis) iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \models \exists i^k A$ by the same preceding definitions. ⊣

Corollary 3.7. *Let $\mathfrak{A}^* \in \mathit{Mod}(\Delta)$ and H be the function from Definition 3.4. When $A \in \mathit{Sent}(\mathit{MSL}^*)$, $\mathfrak{A}^* \models A$ iff $H(\mathfrak{A}^*) \models A$.*

Proof. Assume $A \in \mathit{Sent}(\mathit{MSL}^*)$. $\mathfrak{A}^* \models A$ iff $\langle\mathfrak{A}^*, M\rangle \models A$ (by Definition 2.7 of satisfaction in *MSL*) iff $\langle H(\mathfrak{A}^*), f \circ M\rangle \models A$ (by Proposition 3.6) iff $H(\mathfrak{A}^*) \models A$ (again by Definition 2.7 of satisfaction in *MSL*). ⊣

This proposition will be usefull in the next proofs.

Proposition 3.8. *For the set* Δ *from Def. 3.3,* $Conv_1(Str(daC)) \subseteq Mod(\Delta)$.

Proof. Let $\mathfrak{A}^* \in Conv_1(\mathfrak{A})$ with $\mathfrak{A} \in Str(daC)$. By the definition of $R^{\mathfrak{A}^*}$ in 3.2, $\mathfrak{A}^*$ satisfies reflexivity, transitivity and heredity (the first three axioms of Δ). To prove that $\mathfrak{A}^*$ satisfies extensionality (the fourth axiom), let $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{W}_j$ and assume $\langle \mathfrak{A}^*, M^{\boldsymbol{xy}}_{xy} \rangle \models \forall w(\varepsilon wx \leftrightarrow \varepsilon wy)$. Then by Definition 2.6 of interpretation in MSL^*, $(\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}$ iff $(\boldsymbol{w}, \boldsymbol{y}) \in \varepsilon^{\mathfrak{A}^*}$, so by Definition 3.2 of $Conv_1$, $\boldsymbol{w} \in \boldsymbol{x}$ iff $\boldsymbol{w} \in \boldsymbol{y}$, hence $\boldsymbol{x} = \boldsymbol{y}$. By Definition 2.2 of structures in MSL^*, $\approx^{\mathfrak{A}^*} (\boldsymbol{x}, \boldsymbol{y}) = T$, and by Definition 2.6, $\langle \mathfrak{A}^*, M^{\boldsymbol{xy}}_{xy} \rangle \models x \approx y$. Thus extensionality holds.

Next, we show that $\mathfrak{A}^*$ also satisfies the comprehension scheme for each $A \in Form(daC)$. By Proposition 3.2, for any $A \in Form(daC)$, $A^{\mathfrak{A}} = \{\boldsymbol{w} \in \boldsymbol{W}_i \mid Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(A)[w]) = T\}$; this implies that $A^{\mathfrak{A}} \subseteq \boldsymbol{W}_i$ in $Conv_1(\mathfrak{A})$. By Proposition 1.1 and Definition 3.2 of $Conv_1$, $A^{\mathfrak{A}} \in \boldsymbol{W}_j$ for every $A \in Form(daC)$; that is, there is an $\boldsymbol{x} \in \boldsymbol{W}_j$ such that for every $A \in Form(daC)$, $\boldsymbol{w} \in \boldsymbol{x}$ iff $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w \models t(A)[w]$. Thus comprehension holds because $\varepsilon^{\mathfrak{A}^*}$ behaves like $\in$. ⊣

Proposition 3.9. *Let* Δ *be the set from Definition 3.3 and* H *the function from Definition 3.4. Then*

$$H(Mod(\Delta)) = Conv_1(Str(daC)).$$

Proof. We show that, for any $\mathfrak{B}^* \in Str(MSL^*)$, $\mathfrak{B}^* \in H(Mod(\Delta))$ iff $\mathfrak{B}^* \in Conv_1(Str(daC))$.

($\Rightarrow$) Assume $\mathfrak{B}^* \in H(Mod(\Delta))$. Then $\mathfrak{B}^*$ satisfies items (1) to (4), (6) and (7) of the definition of $Conv_1$ in 3.2. Next we show that the domain $\boldsymbol{W}^{\mathfrak{B}^*}_j$ defined by H also satisfies item (5) of Definition 3.2. As $\mathfrak{B}^* \in H(Mod(\Delta))$, by Corollary 3.7 the comprehension scheme, $\exists x \forall w(\varepsilon wx \leftrightarrow t(A)[w])$ for each $A \in Form(daC)$, holds in $\mathfrak{B}^*$. Therefore, $\{\boldsymbol{w} \in \boldsymbol{W}^{\mathfrak{B}^*}_i \mid \langle \mathfrak{B}^*, M^{\boldsymbol{w}}_w \rangle \models t(A)[w]\} \in \boldsymbol{W}^{\mathfrak{B}^*}_j$ for all $A \in Form(daC)$. That is, there is an $\boldsymbol{x} \in \boldsymbol{W}^{\mathfrak{B}^*}_j$ such that for all $\boldsymbol{w} \in \boldsymbol{W}^{\mathfrak{B}^*}_i$, $\boldsymbol{w} \in \boldsymbol{x}$ iff $\mathfrak{B}^* \models t(A)[w]$. We now verify each part of condition (5) of $Conv_1$ by instantiating this scheme with appropriate translations $t(A)[w]$ of formulas $A \in Form(daC)$.

The case $P^{\mathfrak{B}^*}_n$: By comprehension for the translation of $p_n \in Atom$, and Definitions 3.1 (translation), 2.6 (interpretation in MSL) and 3.4

(function H), there exists $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ such that $\langle \mathfrak{B}^*, M_x^{\boldsymbol{x}} \rangle \models \forall w(\varepsilon wx \leftrightarrow \varepsilon wP_n)$. As $\varepsilon^{\mathfrak{B}^*}$ is set membership, $\boldsymbol{x} = P_n^{\mathfrak{B}^*} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$, as desired.

The case $\boldsymbol{W}_i^{\mathfrak{B}^*}$: By the comprehension scheme for the translation of $p_n \vee {\sim} p_n$ and Definitions 3.1 and 2.6, there exists $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ such that $\langle \mathfrak{B}^*, M_x^{\boldsymbol{x}} \rangle \models \forall w(\varepsilon wx \leftrightarrow (\varepsilon wP_n \vee \exists u(Ruw \wedge \neg \varepsilon uP_n)))$. That is, there exists $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ such that, for all $\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$, $\boldsymbol{w} \in \boldsymbol{x}$ iff either $\boldsymbol{w} \in P_n^{\mathfrak{B}^*}$ or there is a $\boldsymbol{u} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{B}^*}$ and $\boldsymbol{u} \notin P_n^{\mathfrak{B}^*}$. Moreover, trivially, for every $\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$, either (1) $\boldsymbol{w} \in P_n^{\mathfrak{B}^*}$ or (2) $\boldsymbol{w} \notin P_n^{\mathfrak{B}^*}$. In case (2), since $R^{\mathfrak{B}^*}$ is reflexive, there is a $\boldsymbol{u} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{B}^*}$ and $\boldsymbol{u} \notin P_n^{\mathfrak{B}^*}$ (i.e. $\boldsymbol{w}$ itself). Consequently, $\boldsymbol{w} \in P_n^{\mathfrak{B}^*}$ or there is a $\boldsymbol{u} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{B}^*}$ and $\boldsymbol{u} \notin P_n^{\mathfrak{B}^*}$. Hence every $\boldsymbol{w}$ satisfies the membership condition for $\boldsymbol{x}$, and therefore $\boldsymbol{x} = \boldsymbol{W}_i^{\mathfrak{B}^*}$.

The case $\emptyset$: By the comprehension scheme for the translation of the formula ${\sim}\,(p_n \vee {\sim} p_n)$, and Definitions 3.1 and 2.6, there is an $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ such that $\langle \mathfrak{B}^*, M_x^{\boldsymbol{x}} \rangle \models \forall w(\varepsilon wx \leftrightarrow \exists u(Ruw \wedge \neg(t(p_n)[u] \vee \exists v(Rvu \wedge \neg t(p_n)[v]))))$. That is, there is an $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ such that for all $\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$, $\boldsymbol{w} \in \boldsymbol{x}$ iff there is a $\boldsymbol{u} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$ such that $(\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{B}^*}$, $\boldsymbol{u} \notin P_n^{\mathfrak{B}^*}$ and, for all $\boldsymbol{v} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$, if $(\boldsymbol{v}, \boldsymbol{u}) \in R^{\mathfrak{B}^*}$ then $\boldsymbol{v} \in P_n^{\mathfrak{B}^*}$. Now, fix $\boldsymbol{u}$ as a particular element; we know that $(\boldsymbol{u}, \boldsymbol{w}) \in R^{\mathfrak{B}^*}$, $\boldsymbol{u} \notin P_n^{\mathfrak{B}^*}$ and, if $(\boldsymbol{u}, \boldsymbol{u}) \in R^{\mathfrak{B}^*}$ then $\boldsymbol{u} \in P_n^{\mathfrak{B}^*}$; thus, since $R^{\mathfrak{B}^*}$ is reflexive, we also know that $\boldsymbol{u} \in P_n^{\mathfrak{B}^*}$. This is clearly a contradiction. Therefore, there is no $\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*}$ satisfying the conditions to belong to the particular set $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$; and so $\boldsymbol{x} = \emptyset$.

For the remaining cases, assume that $\boldsymbol{y} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$, $\boldsymbol{z} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$, and $A, B \in \mathit{Form}(daC)$, where $\boldsymbol{y} = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid \langle \mathfrak{B}^*, M_w^{\boldsymbol{w}} \rangle \models t(A)[w]\}$ and $\boldsymbol{z} = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid \langle \mathfrak{B}^*, M_w^{\boldsymbol{w}} \rangle \models t(B)[w]\}$.

The case $\boldsymbol{y} \cup \boldsymbol{z}$: By comprehension for the translation of $A \vee B$ and Definitions 3.1 and 2.6, there is $\boldsymbol{y} \cup \boldsymbol{z} = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid \langle \mathfrak{B}^*, M_w^{\boldsymbol{w}} \rangle \models t(A \vee B)[w]\} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$.

The case $\boldsymbol{y} \cap \boldsymbol{z}$: Similar, using comprehension for translation of $A \wedge B$.

The case $\mathit{Dom}(\overline{R^{\mathfrak{B}^*}} \cap (\boldsymbol{W}_i^{\mathfrak{B}^*} \times (\boldsymbol{W}_i^{\mathfrak{B}^*} - \boldsymbol{y})))$: By comprehension for translation of ${\sim} A$, Definitions 3.1, 2.6, and set theory basic definitions, there is $\mathit{Dom}(\overline{R^{\mathfrak{B}^*}} \cap (\boldsymbol{W}_i^{\mathfrak{B}^*} \times (\boldsymbol{W}_i^{\mathfrak{B}^*} - \boldsymbol{y}))) = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{B}^*} \mid \langle \mathfrak{B}^*, M_w^{\boldsymbol{w}} \rangle \models t({\sim} A)[w]\} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$.

The case $\boldsymbol{W}_i^{\mathfrak{B}^*} - \mathit{Dom}(R^{\mathfrak{B}^*} \cap (\boldsymbol{W}_i^{\mathfrak{B}^*} \times (\boldsymbol{y} - \boldsymbol{z})))$: Similar to the previous one, with obvious modifications. Use comprehension for translation of $A \rightarrowtail B$.

Hence $\mathfrak{B}^*$ satisfies Definition of Conv_1, so $\mathfrak{B}^* \in \mathit{Conv}_1(\mathit{Str}(daC))$.

($\Leftarrow$) Assume that $\mathfrak{A}^* \in Conv_1(Str(daC))$. This implies that $\mathfrak{A}^*$ satisfies items (1) and (3) to (5) of the definition of $H(Mod(\Delta))$ in 3.4. What follows shows that $\boldsymbol{W}_j^{\mathfrak{A}^*}$ also satisfies item (2) of the definition of H. This last condition requires that, for every $\boldsymbol{x}$ in that universe, $\boldsymbol{x} = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{A}^*} \mid (\boldsymbol{w}, \boldsymbol{x}^{\mathfrak{B}^*}) \in \varepsilon^{\mathfrak{B}^*}\}$ such that $\boldsymbol{x}^{\mathfrak{B}^*} \in \boldsymbol{W}_j^{\mathfrak{B}^*}$ for some $\mathfrak{B}^* \in Mod(\Delta)$. Since, by Proposition 3.8, $Conv_1(Str(daC)) \subseteq Mod(\Delta)$, $\mathfrak{A}^*$ itself can serve as the $\mathfrak{B}^*$ required by the condition. It is trivial that, for every $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{A}^*}$, $\boldsymbol{x} = \{\boldsymbol{w} \in \boldsymbol{W}_i^{\mathfrak{A}^*} \mid (\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}\}$ such that $\boldsymbol{x} \in \boldsymbol{W}_j^{\mathfrak{A}^*}$ because (by Definition 3.2 of $Conv_1$) $(\boldsymbol{w}, \boldsymbol{x}) \in \varepsilon^{\mathfrak{A}^*}$ iff $\boldsymbol{w} \in \boldsymbol{x}$. Therefore, $\mathfrak{A}^*$ satisfies all conditions of the definition of H, and thus $\mathfrak{A}^* \in H(Mod(\Delta))$. $\dashv$

THEOREM 3.10 (Representation Theorem). *Let $A \in Form(daC)$. $\Delta \subseteq Form(MSL^*)$ is a countable set such that $Conv_1(Str(daC)) \subseteq Mod(\Delta)$ and $\models_{daC} A$ iff $\Delta \models_{MSL^*} \forall w(t(A)[w])$.*

PROOF. Let Δ be the set from Definition 3.3. Then, by Proposition 3.8, $Conv_1(Str(daC)) \subseteq Mod(\Delta)$.

($\Rightarrow$) Suppose $\models_{daC} A$; that is, by Definition 1.4 of satisfaction in daC, for any $\mathfrak{A} \in Str(daC)$, $\mathfrak{A} \models A$. Now assume, for an arbitrary structure $\mathfrak{A}^* \in Str(MSL^*)$, that $\mathfrak{A}^* \models \Delta$. From Proposition 3.9, $H(\mathfrak{A}^*) \in Conv_1(Str(daC))$; that is, $H(\mathfrak{A}^*) = Conv_1(\mathfrak{A})$ for some $\mathfrak{A} \in Str(daC)$. Since, for any $\mathfrak{A} \in Str(daC)$, $\mathfrak{A} \models A$, then the particular daC structure mapped to $H(\mathfrak{A}^*)$ also satisfies A. Now, by Lemma 3.3, $H(\mathfrak{A}^*) \models \forall w(t(A)[w])$. By Corollary 3.7, $\mathfrak{A}^*$ and $H(\mathfrak{A}^*)$ satisfy the same sentences, so $\mathfrak{A}^* \models \forall w(t(A)[w])$. Since $\mathfrak{A}^*$ is an arbitrary structure in $Mod(\Delta)$, then $\Delta \models_{MSL^*} \forall w(t(A)[w])$ by Definition 2.10 of consequence in MSL.

($\Leftarrow$) Take $\Delta \models_{MSL^*} \forall w(t(A)[w])$ as hypothesis; that is, by Definition 2.10 of consequence in MSL, for every $\mathfrak{A}^* \in Str(MSL^*)$, if $\mathfrak{A}^* \models \Delta$ then $\mathfrak{A}^* \models \forall w(t(A)[w])$. Fix an arbitrary structure $\mathfrak{A}^* \in Conv_1(Str(daC))$; for this one, obviously $\mathfrak{A}^* \in Mod(\Delta)$; that is, $\mathfrak{A}^* \models \Delta$. By the previous hypothesis, $\mathfrak{A}^* \models \forall w(t(A)[w])$. This means that all structures $\mathfrak{A}^* = Conv_1(\mathfrak{A})$, for some $\mathfrak{A} \in Str(daC)$, satisfy $\forall w(t(A)[w])$. From Lemma 3.3 and the fact that $Conv_1$ is a function, it follows that, for each structure $\mathfrak{A} \in Dom(Conv_1)$, $\mathfrak{A} \models A$. Finally, since $Dom(Conv_1) = Str(daC)$ then, for any $\mathfrak{A} \in Str(daC)$, $\mathfrak{A} \models A$; that is, $\models_{daC} A$. $\dashv$

At this point, although the Enumerability Theorem presumably holds for daC (due to the Soundness and Completeness theorems proved by Priest (2009)), it is possible to prove enumerability for daC as a consequence of the translation proposed. This follows from the Representation

Theorem and is therefore one of the first properties exported from *MSL* to *daC* via the proposed translation.

Theorem 3.11. *The set of valid formulas in daC is recursively enumerable.*

Proof. By Theorem 3.10, there exists a $\Delta \subseteq Form(MSL^*)$ such that $\models_{daC} A$ iff $\Delta \models_{MSL^*} \forall w(t(A)[w])$ for every $A \in Form(daC)$. Since Δ is recursively enumerable, it follows that the set $\{A \in Form(MSL^*) \mid \Delta \models_{MSL^*} A\}$ is also recursively enumerable. Finally, because the translation function t is recursive, its inverse t^{-1} is likewise recursive; hence $\{A \in Form(daC) \mid \models_{daC} A\}$ is also recursively enumerable. ⊣

3.3. Second level: Main Theorem

Having proved the Representation Theorem, the first level proposed in Manzano's version of the model-theoretic translation approach has been completed. Up to this point, it has been shown that the translation preserves the validity of *daC* formulas by transforming them into many-sorted formulas that are true in the models of Δ. The next step is to prove the Main Theorem. At this point, it will be shown that not only is the set of valid formulas preserved but also the set of valid arguments. To achieve this, the function $Conv_2$ must be created to convert MSL^*-structures that are models of Δ into *daC*-structures. If this function is properly designed, together with $Conv_1$, it will allow one to go back and forth between *daC* and MSL^* structures modulo Δ without altering the set of valid arguments (and obviously not the set of valid formulas); in this way, an equivalence between the two semantics will have been demonstrated.

Definition 3.5. Let Δ be the set from Definition 3.3. Then

$$Conv_2 \colon Str(MSL^*) \supseteq Mod(\Delta) \to Str(daC)$$

is the function $Conv_1^{-1} \circ H$ that assigns to each $\mathfrak{A}^*$ in $Mod(\Delta)$ the structure $Conv_1^{-1}(H(\mathfrak{A}^*))$.

Lemma 3.12. *There is a $\Delta \subseteq Form(MSL^*)$ with $Conv_1(Str(daC)) \subseteq Mod(\Delta)$ such that, for any $\mathfrak{A}^* \in Mod(\Delta)$, there is an $\mathfrak{A} \in Str(daC)$ where $\mathfrak{A} = Conv_2(\mathfrak{A}^*)$ and the following holds:*

(a) $Conv_2(\mathfrak{A}^*), \boldsymbol{w} \models A$ *iff* ${\mathfrak{A}^*}^{\boldsymbol{w}}_{w}(t(A)[w]) = T$.

(b) $Conv_2(\mathfrak{A}^*) \models A$ *iff* $\mathfrak{A}^* \models \forall w(t(A)[w])$, *for each* $A \in Form(daC)$ *and* $\boldsymbol{w} \in \boldsymbol{W}$.

PROOF. Let $\mathfrak{A}^* \in Mod(\Delta)$, $A \in Form(daC)$ and $\boldsymbol{w} \in \boldsymbol{W}_i$.

(a) $Conv_2(\mathfrak{A}^*), \boldsymbol{w} \models A$ iff $Conv_1(Conv_2(\mathfrak{A}^*))^{\boldsymbol{w}}_w(t(A)[w]) = T$, by Lemma 3.3. This is so iff $Conv_1(Conv_1^{-1} \circ H(\mathfrak{A}^*))^{\boldsymbol{w}}_w(t(A)[w]) = T$ (by Definition 3.5 of $Conv_2$) iff $H(\mathfrak{A}^*)^{\boldsymbol{w}}_w(t(A)[w]) = T$ iff $\mathfrak{A}^{*\boldsymbol{w}}_w(t(A)[w]) = T$ (by Proposition 3.6 which establishes that $\langle \mathfrak{A}^*, M\rangle$ and $\langle H(\mathfrak{A}^*), f \circ M\rangle$ satisfy the same formulas).

(b) $Conv_2(\mathfrak{A}^*) \models A$ iff $Conv_1(Conv_2(\mathfrak{A}^*)) \models \forall w(t(A)[w])$, by Lemma 3.3 which states that $Conv_1(\mathfrak{A})$ satisfies the translations of the formulas satisfied by $\mathfrak{A} \in Str(daC)$. This is so iff $Conv_1(Conv_1^{-1} \circ H(\mathfrak{A}^*)) \models \forall w(t(A)[w])$ (by Definition 3.5 of $Conv_2$) iff $H(\mathfrak{A}^*) \models \forall w(t(A)[w])$ iff $\mathfrak{A}^* \models \forall w(t(A)[w])$ (by Corollary 3.7 which establishes that $\mathfrak{A}^*$ and $H(\mathfrak{A}^*)$ satisfy the same sentences). ⊣

THEOREM 3.13 (Main Theorem). *Let $\Gamma \cup \{A\} \subseteq Form(daC)$. There exists a set $\Delta \subseteq Form(MSL^*)$ with $Conv_1(Str(daC)) \subseteq Mod(\Delta)$ such that*

$$\Gamma \models_{daC} A \text{ iff } t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w].$$

PROOF. Let Δ be the set from Definition 3.3. By Proposition 3.8 $Conv_1(Str(daC)) \subseteq Mod(\Delta)$.

($\Rightarrow$) Suppose $\Gamma \models_{daC} A$. Fix a structure $\mathfrak{A}^{*\boldsymbol{w}}_w \in Str(MSL^*)$ and suppose it is a model of $t(\Gamma)[w] \cup \Delta$. From the above, we know that $\mathfrak{A}^* \in Mod(\Delta)$. Therefore, by Lemma 3.12, Remark 3.1 (which specifies the application of translation to sets of formulas), and the fact that $\mathfrak{A}^* \models t(\Gamma)[w]$, it follows that there exists a structure $Conv_2(\mathfrak{A}^*) \in Str(daC)$ such that $Conv_2(\mathfrak{A}^*), \boldsymbol{w} \models B$ for every $B \in \Gamma$. Thus, since $\Gamma \models_{daC} A$, $Conv_2(\mathfrak{A}^*), \boldsymbol{w} \models A$. Finally, by Lemma 3.12 again, $\mathfrak{A}^{*\boldsymbol{w}}_w \models t(A)[w]$; that is, by Definition 2.10 of consequence in *MSL*, if $\Gamma \models_{daC} A$ then $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$.

($\Leftarrow$) Assume that $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$. Also, fix a structure $\mathfrak{A} \in Str(daC)$ and suppose it is a model of Γ. By Lemma 3.3 (which states that for $\mathfrak{A} \in Str(daC)$, $\mathfrak{A}, \boldsymbol{w} \models A$ iff $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(A)[w]) = T$) and Remark 3.1 (which specifies the application of translation to sets of formulas), there exists a structure $Conv_1(\mathfrak{A}) \in Str(MSL^*)$ such that $Conv_1(\mathfrak{A})^{\boldsymbol{w}}_w(t(B)[w]) = T$ for every $B \in \Gamma$. Moreover, by Propositions 3.6 (which establishes that $\langle \mathfrak{A}^*, M\rangle$ and $\langle H(\mathfrak{A}^*), f \circ M\rangle$ satisfy the same formulas) and 3.9 (which states that $H(Mod(\Delta)) = Conv_1(Str(daC))$), we know that $Conv_1(\mathfrak{A}) \in Mod(\Delta)$. Therefore, $Conv_1(\mathfrak{A}) \models t(\Gamma)[w] \cup \Delta$. Now, by the initial assumption, $Conv_1(\mathfrak{A}) \models t(A)[w]$. Finally, by Lemma

3.3 again, $\mathfrak{A} \models A$; that is, by Definition 1.11 of consequence in *daC*, if $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$ then $\Gamma \models A$. ⊣

At this point, it has just been shown that the semantics of *daC* and MSL^* are equivalent thanks to the proposed translation modulo Δ; in this way, not only does the translation preserve the set of valid formulas, but also the set of valid arguments. Thanks to this, it is possible to export the Compactness and Löwenheim–Skolem theorems from *MSL* to *daC*.

Theorem 3.14 (Compactness Theorem). *Let* $\Gamma \cup \{A\} \subseteq Form(daC)$*. If* $\Gamma \models_{daC} A$*, then there exists a finite set* $\Pi \subseteq \Gamma$ *such that* $\Pi \models_{daC} A$*.*

Proof. Assume $\Gamma \models_{daC} A$. By Theorem 3.13, there is a set $\Delta \subseteq Form(MSL^*)$ with $Conv_1(Str(daC)) \subseteq Mod(\Delta)$ such that $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$. Since Compactness Theorem holds in MSL^*, there is a finite set of MSL^* formulas $\{t(B_1)[w], \ldots, t(B_n)[w]\} \subseteq t(\Gamma)[w] \cup \Delta$ such that $\{t(B_1)[w], \ldots, t(B_n)[w]\} \models_{MSL^*} t(A)[w]$. Moreover, by monotonicity of *MSL*, $\{t(B_1)[w], \ldots, t(B_n)[w]\} \cup \Delta \models_{MSL^*} t(A)[w]$. Therefore, by the Main Theorem, there exists a finite set $\{B_1, \ldots, B_n\} \subseteq \Gamma$ such that $\{B_1, \ldots, B_n\} \models_{daC} A$. ⊣

Theorem 3.15 (Löwenheim–Skolem Theorem). *Let* $\Gamma \subseteq Form(daC)$*. If* Γ *has a model, then it has a countable model.*

Proof. Let Δ be the set from Definition 3.3. Suppose there is a model $\mathfrak{A}$ of Γ, i.e., $\mathfrak{A} \models \Gamma$. By Lemma 3.3 (which states that for $\mathfrak{A} \in Str(daC)$, $\mathfrak{A} \models A$ iff $Conv_1(\mathfrak{A}) \models \forall w(t(A)[w])$), $Conv_1(\mathfrak{A})$ is a model of $\{\forall w(t(A)[w]) \mid A \in \Gamma\}$. Now, since $Conv_1(\mathfrak{A}) \in Conv_1(Str(daC)) \subseteq Mod(\Delta)$, it follows that $Conv_1(\mathfrak{A}) \in Mod(\{\forall w(t(A)[w]) \mid A \in \Gamma\} \cup \Delta)$. As the Löwenheim–Skolem Theorem holds in *MSL* and there is a model for $\{\forall w(t(A)[w]) \mid A \in \Gamma\} \cup \Delta$, there exists a countable model $\mathfrak{B}^*$ of $\{\forall w(t(A)[w]) \mid A \in \Gamma\} \cup \Delta$. By Lemma 3.12, $Conv_2(\mathfrak{B}^*) \models \Gamma$; and since $Conv_2$ does not change the domain $\boldsymbol{W}_i = \boldsymbol{W}$, $Conv_2(\mathfrak{B}^*)$ is also a countable model. ⊣

3.4. Third level: Calculi equivalence

The third part of the translation proves that the set of derivations in $Cal(daC)$ is well represented in the set of derivations in $Cal(MSL^*)$ modulo Δ. To accomplish this, the first step is to show that each inference rule of *daC* is mapped as a correct derivation in $Cal(MSL^*)$ modulo Δ.

Next, the other direction will be shown; that is, that each derivation in *Cal*(MSL^*) modulo Δ corresponds to a derivation in *Cal*(*daC*).

THEOREM 3.16. *Let $\Gamma \subseteq Form(daC)$ and $A \in Form(daC)$. There exists a set $\Delta \subseteq Form(MSL^*)$ such that*

$$\textit{if } \Gamma \vdash_{daC} A \textit{ then } t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w].$$

PROOF. The proof is carried out over the inference rules of *Cal*(*daC*) given in Definition 1.12. Let Δ be the set from Definition 3.3, let $\Delta_0 \subseteq \Delta$ be a finite set, and let $A, B, C \in Form(daC)$. The rules used are those from the Sequent Calculus of *MSL* in Definition 2.11 and Proposition 2.1. For each rule $\Gamma \vdash_{daC} A$ of *Cal*(*daC*), we show that $t(\Gamma)[w], \Delta_0 \hookrightarrow t(A)[w]$. This means, in each case, that $t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w]$ by Definitions 2.12 and 2.13 that govern derivations in the *MSL* calculus.

1, $\{A \wedge B\} \vdash_{daC} A$: Use HI (with $t(A \wedge B)[w], \Delta_0 \hookrightarrow t(A \wedge B)[w]$), Definition 3.1 of translation, and EC.

2. $\{A \wedge B\} \vdash_{daC} B$: Similar to the previous case.

3. $\{A, B\} \vdash_{daC} A \wedge B$: Use HI (with $t(A)[w], t(B)[w], \Delta_0 \hookrightarrow t(A)[w]$ and $t(A)[w], t(B)[w], \Delta_0 \hookrightarrow t(B)[w]$), IC, and Definition 3.1 of translation.

4. If $A \vdash_{daC} C$ and $B \vdash_{daC} C$, then $\{A \vee B\} \vdash_{daC} C$: Assume $t(A)[w], \Delta_0 \hookrightarrow t(C)[w]$ and $t(B)[w], \Delta_0 \hookrightarrow t(C)[w]$. Use IDA and Definition 3.1 of translation.

5. $\{A\} \vdash_{daC} A \vee B$: Use HI (with $t(A)[w], \Delta_0 \hookrightarrow t(A)[w]$), IDC, and Definition 3.1 of translation.

6. $\{B\} \vdash_{daC} A \vee B$: Similar to the previous case.

7. $\{A \rightarrowtail B, A\} \vdash_{daC} B$: Let $\forall w Rww \in \Delta_0$. Use HI (first with $t(A \rightarrowtail B)[w], t(A)[w], \Delta_0 \hookrightarrow t(A \rightarrowtail B)[w]$; afterwards, use reflexivity of R, and finally use $t(A \rightarrowtail B)[w], t(A)[w], \Delta_0 \hookrightarrow t(A)[w]$), Definition 3.1 of translation, EGC, and MP.

8. If $A \vdash_{daC} B$ then $\vdash_{daC} A \rightarrowtail B$: Let $\exists x \forall w(\varepsilon wx \leftrightarrow t(B)[w]) \in \Delta_0$ and $\forall x \forall wu((\varepsilon wx \wedge Rwu) \to \varepsilon ux) \in \Delta_0$. Assume $t(A)[w], \Delta_0 \hookrightarrow t(B)[w]$. Use IGA, EGA, M (adding Rwu), HI, first with the comprehension scheme for the translation of $B \in Form(daC)$, then with the Heredity Constraint, and finally with the sequence $Rwu, t(A)[w], \Delta_0 \hookrightarrow Rwu$, EPC, EGC, MP, IC, DT, IGC, Remark 2.2 on abbreviations in the *MSL* language, and Definition 3.1 of translation.

9. $\vdash_{daC} A \vee {\sim}A$: Let $\forall w Rww \in \Delta_0$. Use HI (first with reflexivity of R, then with $\Delta_0, \neg t(A \vee {\sim}A) \hookrightarrow \neg t(A \vee {\sim}A)$), M (adding $\neg t(A \vee$

$\sim A$)), EGC, Definition 3.1 of translation, deM, EC, NP, Remark 2.2 on abbreviations in the *MSL* language, DN, ED, and NC.

10 If $\vdash_{daC} A \vee B$ then $\sim A \vdash_{daC} B$: Let $\exists x \forall w(\varepsilon wx \leftrightarrow t(B)[w]) \in \varDelta_0$ and $\forall x \forall wu((\varepsilon wx \wedge Rwu) \rightarrow \varepsilon ux) \in \varDelta_0$. Assume $\varDelta_0 \hookrightarrow t(A \vee B)[w]$. Use IGC, EGC, Definition 3.1 of translation, M (adding $t(\sim A)[w]$), HI (first with $t(\sim A)[w], \varDelta_0 \hookrightarrow t(\sim A)[w]$, then with the comprehension scheme for $B \in Form(daC)$, and finally with the Heredity Constraint), EPC, EC, ED, Remark 2.2 on abbreviations in the *MSL* language, MP, and IC. ⊣

Lemma 3.17. *Let $A \in Form(daC)$ and Δ be the set from Definition 3.3. If $\vdash_{daC} A$ then $\Delta \vdash_{MSL^*} \forall w(t(A)[w])$.*

Proof. Suppose that $\vdash_{daC} A$; therefore, by Theorem 3.16, $\Delta \vdash_{MSL^*} t(A)[w]$. From Definitions 2.12 and 2.13 governing derivations in the *MSL* calculus, there exists a finite set $\varDelta_0 \subseteq \Delta$ such that, in $Cal(MSL^*)$, the sequent $\varDelta_0 \hookrightarrow t(A)[w]$ is derived. Then, by the IGC rule in $Cal(MSL^*)$ from Proposition 2.1, and again by Definitions 2.12 and 2.13, it follows that $\Delta \vdash_{MSL^*} \forall w(t(A)[w])$. ⊣

Although Priest (2009) has already established the Soundness Theorem for daC, another proof of that theorem is presented here. Note that in this proof, the Soundness Theorem is a consequence of the translation developed here.

Theorem 3.18 (Soundness Theorem). *If $\varGamma \vdash_{daC} A$ then $\varGamma \models_{daC} A$.*

Proof. Assume that $\varGamma \vdash_{daC} A$. From Theorem 3.16, $t(\varGamma)[w] \cup \varDelta \vdash_{MSL^*} t(A)[w]$. By the Soundness Theorem of *MSL*, $t(\varGamma)[w] \cup \varDelta \models_{MSL^*} t(A)[w]$; and by the Main Theorem (3.13), $\varGamma \models_{daC} A$. ⊣

In the final part of the third level of the translation, it remains only to show that the set of derivations in $Cal(MSL^*)$ modulo Δ correctly represents the set of derivations in $Cal(daC)$; that is, that each derivation in $Cal(MSL^*)$ modulo $\varDelta$ corresponds to a derivation in $Cal(daC)$ (i.e., if $t(\varGamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w]$, then $\varGamma \vdash_{daC} A$). To achieve this, the Canonical Model of daC presented earlier is used.

Proposition 3.19. *For every $A \in Form(daC)$ and every $\varGamma \in \boldsymbol{W}_{CAN}$,*

$$Conv_1(\mathfrak{A}_{CAN})[\varGamma] \models t(A)[w] \textit{ iff } A \in \varGamma.$$

Proof. By Lemma 3.3(a), $Conv_1(\mathfrak{A}_{CAN})[\varGamma] \models t(A)[w]$ iff $\mathfrak{A}_{CAN}, \varGamma \models A$. Moreover, since by Theorem 1.12 we know that $\mathfrak{A}_{CAN}, \varGamma \models A$ iff $A \in \varGamma$, we conclude that $Conv_1(\mathfrak{A}_{CAN})[\varGamma] \models t(A)[w]$ iff $A \in \varGamma$. ⊣

PROPOSITION 3.20. *Let Δ be the set from Definition 3.3.*

$$Conv_1(\mathfrak{A}_{CAN}) \in Mod(\Delta).$$

PROOF. $Conv_1(\mathfrak{A}_{CAN}) \in Conv_1(Str(daC)) = H(Mod(\Delta)) \subseteq Mod(\Delta)$, by Propositions 1.10, 3.9 and 3.6. ⊣

LEMMA 3.21. *If $Conv_1(\mathfrak{A}_{CAN}) \models \forall w(t(A)[w])$ then $\vdash_{daC} A$, for any $A \in Form(daC)$.*

PROOF. Assume that $Conv_1(\mathfrak{A}_{CAN}) \models \forall w(t(A)[w])$. By Remark 2.2 on abbreviations in the *MSL* language, and Definitions 2.2 and 2.6 of structure and interpretation in *MSL*, $Conv_1(\mathfrak{A}_{CAN})[\Gamma] \models t(A)[w]$ for each Γ. Then, by Proposition 3.19, $A \in \Gamma$ for every Γ. Since Proposition 1.13 states that in the canonical model of daC, if $A \in \Gamma$ for every Γ then $\vdash_{daC} A$, we conclude $\vdash_{daC} A$. ⊣

LEMMA 3.22. *$\Delta \vdash_{MSL^*} \forall w(t(A)[w])$ iff $\vdash_{daC} A$.*

PROOF. (⇒) Suppose $\Delta \vdash_{MSL^*} \forall w(t(A)[w])$. By Definitions 2.12 and 2.13 governing *MSL* derivations, there is a finite $\Delta_0 \subseteq \Delta$ such that the sequent $\Delta_0 \hookrightarrow \forall w(t(A)[w])$ is derivable in $Cal(MSL^*)$. Hence, by the EGC rule from Proposition 2.1, $\Delta_0 \hookrightarrow t(A)[w]$. By the same definitions, $\Delta \vdash_{MSL^*} t(A)[w]$; and by the Soundness Theorem of *MSL*, $\Delta \models_{MSL^*} t(A)[w]$. Therefore, by the Main Theorem (3.13), $\models_{daC} A$. That is, by Definition 1.8 of validity in daC, for every $\mathfrak{A} \in Str(daC)$, $\mathfrak{A} \models A$; and by Proposition 1.10, $\mathfrak{A}_{CAN} \models A$. On the other hand, since by Proposition 3.20, $Conv_1(\mathfrak{A}_{CAN}) \in Mod(\Delta)$, then (by Lemma 3.12 stating that for $\mathfrak{A}^* \in Mod(\Delta)$, $Conv_2(\mathfrak{A}^*) \models A$ iff $\mathfrak{A}^* \models \forall w(t(A)[w])$) there is a structure $\mathfrak{A}_{CAN} = Conv_2(Conv_1(\mathfrak{A}_{CAN}))$ such that $\mathfrak{A}_{CAN} \models A$ iff $Conv_1(\mathfrak{A}_{CAN}) \models \forall w(t(A)[w])$. Since $\mathfrak{A}_{CAN} \models A$, $Conv_1(\mathfrak{A}_{CAN}) \models \forall w(t(A)[w])$. Finally, by Lemma 3.21, $\vdash_{daC} A$.

(⇐) Immediate from Lemma 3.17. ⊣

THEOREM 3.23 (Calculi Equivalence). *Let $\{A\} \cup \Gamma \subseteq Form(daC)$,*

$$\Gamma \vdash_{daC} A \text{ iff } t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w].$$

PROOF. (⇒) Immediate from Theorem 3.16.

(⇐) Assume $t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w]$. By Remark 3.1 (on translation of sets of formulas), and Definitions 2.12 and 2.13 (which govern derivation in *MSL*), there is a finite set $\{t(B_1)[w], \ldots, t(B_n)[w]\} \subseteq$

$t(\Gamma)[w]$ such that $B_i \in Form(daC)$, and another finite set $\Delta_0 \subseteq \Delta$ such that $\{t(B_1)[w], \ldots, t(B_n)[w]\} \cup \Delta_0 \hookrightarrow t(A)[w]$ is derivable in $Cal(MSL^*)$. By repeated applications of DT and DR from $Cal(MSL^*)$, we get the sequent $\Delta_0 \hookrightarrow (t(B_1)[w] \wedge \cdots \wedge t(B_n)[w]) \to t(A)[w]$. Therefore, by IGC and EGC, $\Delta_0 \hookrightarrow t(B_1 \wedge \cdots \wedge B_n)[u] \to t(A)[u]$. Now, by IDC, Remark 2.2 (on abbreviations in the *MSL* language) and IGC, $\Delta_0 \hookrightarrow \forall u(Rwu \to (t(B_1 \wedge \cdots \wedge B_n)[u] \to t(A)[u]))$. Moreover, by Definition 3.1 of translation and IGC, $\Delta_0 \hookrightarrow \forall w(t((B_1 \wedge \cdots \wedge B_n) \rightarrowtail A)[w])$. Thus, $\Delta \vdash_{MSL^*} \forall w(t((B_1 \wedge \cdots \wedge B_n) \rightarrowtail A)[w])$ (by Definitions 2.12 and 2.13), and hence $\vdash_{daC} (B_1 \wedge \cdots \wedge B_n) \rightarrowtail A$ (by Lemma 3.22). Now, suppose Γ is satisfied. By repeated application of rule 3 in $Cal(daC)$ (Definition 1.12) among the elements of Γ, we derive $B_1 \wedge \cdots \wedge B_n$, and then by rule 7, A. Therefore, by Definition 1.13, $\Gamma \vdash_{daC} A$. ⊣

Having established the Calculi Equivalence, we can likewise transfer the Completeness Theorem from *MSL* to *daC*, thus offering an alternative proof to that of Priest (2009).

Theorem 3.24 (Completeness Theorem). *Let* $\{A\} \cup \Gamma \subseteq Form(daC)$,

$$\text{If } \Gamma \models_{daC} A \text{ then } \Gamma \vdash_{daC} A.$$

Proof. Assume $\Gamma \models_{daC} A$. By Theorem 3.13, $t(\Gamma)[w] \cup \Delta \models_{MSL^*} t(A)[w]$. Therefore, by the Completeness Theorem of *MSL*, $t(\Gamma)[w] \cup \Delta \vdash_{MSL^*} t(A)[w]$; and by Calculi Equivalence (3.23), $\Gamma \vdash_{daC} A$. ⊣

Conclusion

A translation of *daC* into *MSL* has been presented here as a function that converts the formulas of the former into formulas of the latter, accompanied by a function $Conv_1$ that converts the structures of the semantics of *daC* into structures of the semantics of *MSL*. From the above, the three levels proposed by Manzano (1996) have been achieved; that is, the Representation and Main theorems have been proved, as well as the Calculi Equivalence. Thus, from the Representation Theorem it is possible to assert that the set of valid formulas of *daC* is well represented by the set of valid formulas of MSL^* if one assumes Δ as a set of axioms. Moreover, the Main Theorem implies that the set of valid arguments of *daC* is equal to the set of valid arguments of MSL^*, modulo Δ, using only formulas produced by the translation. Finally, from the Calculi

Equivalence it follows that the set of derivations in $Cal(daC)$ is equal to the set of derivations in $Cal(MSL^{*})$, modulo Δ, if the formulas used are images of the formulas translated from daC.

Thanks to this development, it was possible to prove the Compactness and Löwenheim-Skolem theorems for daC. In addition, an alternative way to prove the Soundness and Completeness theorems has been presented; although they had already been proved by Priest (2009), they can also be exported directly from *MSL* thanks to the translation developed.

It is important to stress that this translation does not imply any value judgment about daC. The translation developed here does not aim to show daC as reducible to *MSL* but as *emulable* within it. Note that, at the end of the day, this translation is a representation of daC as a theory inside *MSL*. While this does demonstrate the expressive capacity of the latter, daC remains a logic in its own right with a simple language that aids in exploring some of the particularities of the concept of contradiction. On the other hand, the translation developed here also sheds light on the concept of negation in daC, since this translation opens the door to possible studies in which new negation connectives are added to daC easily analyzable from the language of *MSL* (for example, defining $\mathfrak{A}, \boldsymbol{w} \models {\sim}^{*} A$ iff for all $\boldsymbol{u} \in \boldsymbol{W}$, if $(\boldsymbol{u}, \boldsymbol{w}) \in R$, then $\mathfrak{A}, \boldsymbol{u} \not\models A$, which could be translated as $t({\sim}^{*} A)[w] = \forall u(Rwu \rightarrow \neg t(A)[u])$. In this way, *MSL* could serve as a framework that simplifies such studies on daC.

Acknowledgments. I want to thank Mara Manzano, the members of the Episteme Research Group at Universidad del Valle in Cali, Colombia, and the anonymous reviewer for their invaluable contributions to this paper.

References

Carnielli, W., M. Coniglio, and I. D'Ottaviano, 2009, "New dimensions on translatios between logics", *Logica Universalis*, 3: 1–18. DOI: 10.1007/s11787-009-0002-5

Ferguson, T., 2013, "Extensions of Priest–da Costa Logic", *Studia Logica*, 102(1): 145–174.DOI: 10.1007/s11225-013-9469-4

Ferguson T., 2018, "Axiom $(cc)_0$ and verifiability in two extracanonical logics of formal inconsistency", *Principia*, 22(1): 113–138. DOI: 10.5007/1808-1711.2018v22n1p113

Kolmogorov, A., 1925, "On the principle of excluded middle", pages 414–437 in J. Heijenoort (ed.), *From Frege to Gödel: A Source Book in Mathematical Logic 1879–1931*, Harvard University Press, 1977.

Manzano, M., 1989, *Teoría de modelos*, Alianza.

Manzano, M., 1996, *Extensions of First Order Logic*, Cambridge University Press.

Manzano, M., 2004, "Divergencia y rivalidad entre lógicas", pages 277–312 in R. Orayen and A. Moretti (eds.), *Enciclopedia Iberoamericana de filosofía*, Vol. 27, Filosofía de la Lógica, Trotta.

Manzano, M., and V. Aranda, 2022, "Many-sorted logic", in E. N. Zalta and U. Nodelman (eds.), *The Stanford Encyclopedia of Philosophy* (Winter 2022 Edition). https://plato.stanford.edu/archives/win2022/entries/logic-many-sorted/

Ohlbach, H., A. Nonnengart, M. de Rijke, and D. M. Gabbay, 2001, "Encoding two-valued nonclassical logics in classical logic", pages 1403–1486 in A. Robinson and A. Voronkov (eds.), *Handbook of Automated Reasoning*, Elsevier. DOI: 10.1016/B978-044450813-3/50023-0

Osorio Galindo, M., V. Borja Macías, and J. Arrazola Ramírez, 2016, "Revisiting da Costa logic", *Journal of Applied Logic*, 16: 111–127. DOI: 10.1016/j.jal.2016.02.004

Priest, G., 2009. "Dualising intuitionistic negation", *Principia* (13), 2: 165–184. DOI: 10.5007/1808-1711.2009v13n2p165

Priest, G., 2010, "First order da Costa logic", *Studia Logica* 97: 183–198. DOI: 10.1007/s11225-010-9303-1

Shen, Y., Y. Ma, C. Cao, Y. Sui, and J. Wang, 2010, "Preservative translations between logical systems", pages 55–63 in Z. Shi, S. Vadera, A. Aamodt and D. Leake (eds.), *Intelligent Information Processing V. 6th IFIP TC 12 International Conference, IIP 2010*, Manchester, UK, October 13–16, 2010, Proceedings, Springer. DOI: 10.1007/978-3-642-16327-2

van Benthem, J., 1983, *Modal Logic and Classical Logic*, Bibliopolis. DOI: 10.2307/2274406

van Benthem, J., 2001, "Correspondence theory", in D. M. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, Second Edition, Vol. 3, Kluwer Academic Publishers. DOI: 10.1007/978-94-017-0454-0_4

Julian M. Valdes-Toro
Episteme Research Group
Universidad de Salamanca
Salamanca, Spain
Universidad del Valle
Cali, Colombia
jumaovat@hotmail.com id00792251@usal.es
https://orcid.org/0000-0002-8972-5061