# Structured Specification of Paraconsistent Transition Systems

Juliana Cunha[1], Alexandre Madeira[1](✉), and Luís Soares Barbosa[2]

[1] CIDMA, Department of Mathematics, Aveiro University, Aveiro, Portugal
{juliana.cunha,madeira}@ua.pt

[2] INESC TEC and Department of Informatics, Minho University, Braga, Portugal
lsb@di.uminho.pt

**Abstract.** This paper sets the basis for a compositional and structured approach to the specification of paraconsistent transitions systems, framed as an institution. The latter and theirs logics were previously introduced in [CMB22] to deal with scenarios of inconsistency in which several requirements are on stake, either reinforcing or contradicting each other.

## 1 Introduction

In Software Engineering it is often a challenge to cope with modelling contexts in which the classical bivalent logic distinction is not enough. Several modal logics have been proposed [BEGR09] to address such a challenge, namely to capture vagueness or uncertainty. Typically, their semantics is based on residuated lattices, i.e. complete lattices equipped with a commutative monoidal structure such that the monoid composition has a right adjoint, the residue. The lattice carrier stands for the set of truth values, a typical example being the real $[0, 1]$ interval.

Often, however, there is also a need to go further and equip the underlying Kripke structure with both *positive* and *negative* accessibility relations, one weighting the possibility of a transition to be present, the other weighting the possibility of being absent. Moreover, in a number of real situations, such weights are not complementary, and thus both relations should be formally taken into consideration. For this purpose, in a previous work [CMB22] we introduced *paraconsistent transition systems*, abbreviated to PLTS, and the corresponding modal logic, which generalises Belnap-Dunn four-valued logic [RJJ15] in a very generic way. Actually, all the relevant constructions are parametric in a class of residuated lattices, thus admitting different instances according to the structure of the truth values domain that better suits each modelling problem at hands.

The present study was developed in the scope of the Project "Agenda ILLIANCE" [C644919832-00000035 | Project no. 46], financed by PRR – Plano de Recuperação e Resiliência under the Next Generation EU from the European Union. FCT, the Portuguese funding agency for Science and Technology suports with the projects UIDB/04106/2020 and PTDC/CCI-COM/4280/2021.

H. Hojjat and E. Ábrahám (Eds.): FSEN 2023, LNCS 14155, pp. 1–17, 2023.
https://doi.org/10.1007/978-3-031-42441-0_1

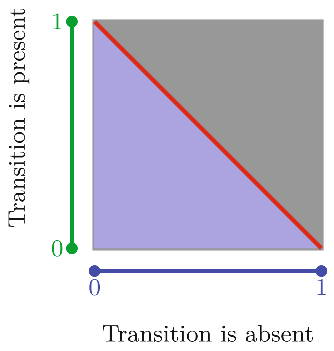


**Fig. 1.** The vagueness-inconsistency square [CMB22].

To exemplify suppose, for example, that weights for both transitions come from a residuated lattice over the real $[0, 1]$ interval.
Then, the two accessibility relations jointly express a scenario of

– *inconsistency*, when the positive and negative weights are contradictory, i.e. they sum to some value greater than 1 (cf, the upper triangle in Fig. 1 filled in grey). Exploring this area of the square
– *vagueness*, when the sum is less than 1 (cf, the lower, periwinkle triangle in Fig. 1);
– *strict consistency*, when the sum is exactly 1, which means that the measures of the factors enforcing or preventing a transition are complementary, corresponding to the red line in the figure.

Exploring the upper triangle calls for paraconsistent logics [Jas69, CCM07], in which inconsistent information is considered as potentially informative. Introduced more than half a century ago, through the pioneering work of F. Asenjo and Newton da Costa, such logics are becoming increasingly popular (see, for example, reference [Aka16], a recent book on engineering applications). This paper goes a step ahead. First the modal logic associated to PLTS is extended to the multi-modal case. Then it is prepared to act as a *structured specification logic* [ST12] equipped with specific versions of the standard structured specification operators *à la CASL* [MHST03]. This offers to the working software engineer the (formal) tools to specify, in a compositional way, paraconsistent transition systems. The approach builds on previous work documented in reference [JGMB21] where a similar agenda is proposed for the specification of fuzzy transition systems. Technically, the price to be paid to support this move consists of framing the logic as an institution [GB92].

The rest of the paper is divided in two sections. Section 2 characterizes an institution for paraconsistent transition systems $\text{L}(\mathcal{A})$. The formalism is parametric to the truth space $\mathcal{A}$, formalised as a metric twisted structure. Then, in Sect. 3, the usual structured specification operators [ST12] are re-built on top of this institution. These are the basic (technical) results for supporting a specification framework for this sort of systems, within the well-established tradition of algebraic specification. Going a step further into the specification methodology and engineering practices will be discussed in a twin publication.

## 2 An Institution for Paraconsistent Transitions Systems

We start by recalling the notion of an institution, followed, in Sect. 2.2, by a characterization of metric twisted algebras which continue the semantic domain upon which the logic is parametrised, as mentioned in the introduction. Such structures amount to a particular class of residuated lattices in which the lattice meet and the monoidal composition coincide, equipped with a metric which entails a concrete meaning to the vagueness-inconsistency square informally described in the introduction. Finally, in sub-sect. 2.3, the relevant institution(s) for $\mathrm{L}(\mathcal{A})$ is built in a step by step way and suitably illustrated.

### 2.1 Institutions

An institution abstractly defines a logic system by describing the kind of signatures in the system, the kind of models and a satisfaction relation between models and sentences.

**Definition 1** **([GB92]).** *An institution $I$ is a tuple*

$$I = (\mathsf{Sign}_I, \mathsf{Sen}_I, \mathsf{Mod}_I, \models_I)$$

*consisting of*

- *a category $\mathsf{Sign}_I$ of signatures*
- *a functor $\mathsf{Sen}_I : \mathsf{Sign}_I \to \mathbb{S}et$ giving a set of $\Sigma-$sentences for each signature $\Sigma \in |\mathsf{Sign}_I|$. For each signature morphism $\sigma : \Sigma \to \Sigma'$ the function*

$$\mathsf{Sen}_I(\sigma) : \mathsf{Sen}_I(\Sigma) \to \mathsf{Sen}_I(\Sigma')$$

  *translates $\Sigma-$sentences to $\Sigma'-$sentences*
- *a functor $\mathsf{Mod}_I : \mathsf{Sign}_I^{op} \to Cat$ assigns to each signature $\Sigma$ the category of $\Sigma-$models. For each signature morphism $\sigma : \Sigma \to \Sigma'$ the functor*

$$\mathsf{Mod}_I(\sigma) : \mathsf{Mod}_I(\Sigma') \to \mathsf{Mod}_I(\Sigma)$$

  *translates $\Sigma'-$models to $\Sigma-$models*
- *a satisfaction relation $\models_I^{\Sigma} \subseteq |\mathsf{Mod}_I(\Sigma)| \times \mathsf{Sen}_I(\Sigma)$ determines the satisfaction of $\Sigma-$sentences by $\Sigma-$models for each signature $\Sigma \in |\mathsf{Sign}_I|$.*

*Satisfaction must be preserved under change of signature that is for any signature morphism $\sigma : \Sigma \to \Sigma'$, for any $\varphi \in \mathsf{Sen}_I(\Sigma)$ and $M' \in |\mathsf{Mod}_I(\Sigma')|$*

$$\Big(M' \models_I^{\Sigma'} \mathsf{Sen}_I(\sigma)(\varphi)\Big) \Leftrightarrow \big(\mathsf{Mod}_I(\sigma)(M') \models_I^{\Sigma} \varphi\big) \tag{1}$$

Actually, when formalising multi-valued logics as institutions, the equivalence on the satisfaction condition (1) can be replaced by an equality (c.f. [ACEGG91]):

$$\Big(M' \models_I^{\Sigma'} \mathsf{Sen}_I(\sigma)(\varphi)\Big) = \big(\mathsf{Mod}_I(\sigma)(M') \models_I^{\Sigma} \varphi\big) \tag{2}$$

The institution formalisation several logics, including Propositional, Equational, First-order, High-Order, etc., can be found in reference [ST12].

### 2.2 (Metric) Twisted Algebras

A residuated lattice $\langle A, \sqcap, \sqcup, 1, 0, \odot, \rightharpoonup, e\rangle$ over a set $A$ is a complete lattice $\langle A, \sqcap, \sqcup, 1, 0\rangle$, equipped with a monoid $\langle A, \odot, e\rangle$ such that $\odot$ has a right adjoint, $\rightharpoonup$, called the residuum. We will, however, focus on a particular class of residuated lattices in which the lattice meet ($\sqcap$) and monoidal composition ($\odot$) coincide. Thus the adjunction is stated as $a \sqcap b \leqslant c$ iff $b \leqslant a \rightharpoonup c$. Additionally, we will enforce a pre-linearity condition

$$(a \rightharpoonup b) \sqcup (b \rightharpoonup a) = 1 \tag{3}$$

A residuated lattice obeying prelinearity is known as a MTL-algebra [EG01]. With a slight abuse of nomenclature, the designation iMTL-algebra, from *integral MTL-algebra*, will be used in the sequel for the class of semantic structures considered, i.e. prelinear, residuated lattices such that $\sqcap$ and $\odot$ coincide. Examples of iMTL-algebras are:

– the Boolean algebra $\mathbf{2} = \langle\{0, 1\}, \wedge, \vee, 1, 0, \rightarrow\rangle$

– $\mathbf{3} = \langle\{\top, u, \bot\}, \wedge_3, \vee_3, \top, \bot, \rightarrow_3\rangle$, where

| $\wedge_3$ | $\bot$ | $u$ | $\top$ |
|---|---|---|---|
| $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $u$ | $\bot$ | $u$ | $u$ |
| $\top$ | $\bot$ | $u$ | $\top$ |

| $\vee_3$ | $\bot$ | $u$ | $\top$ |
|---|---|---|---|
| $\bot$ | $\bot$ | $u$ | $\top$ |
| $u$ | $u$ | $u$ | $\top$ |
| $\top$ | $\top$ | $\top$ | $\top$ |

| $\rightarrow_3$ | $\bot$ | $u$ | $\top$ |
|---|---|---|---|
| $\bot$ | $\top$ | $\top$ | $\top$ |
| $u$ | $\bot$ | $\top$ | $\top$ |
| $\top$ | $\bot$ | $u$ | $\top$ |

– $\ddot{\boldsymbol{G}} = \langle[0, 1], \min, \max, 0, 1, \rightarrow\rangle$, with implication defined as
$$a \rightarrow b = \begin{cases} 1 & if a \leqslant b \\ b & otherwise \end{cases}$$

We focus on *iMTL-algebras* $\boldsymbol{A}$ whose carrier $A$ supports a metric space $(A, d)$, with suitable choice of $d$. Where $d \colon A \times A \rightarrow \mathbb{R}^+$ such that $d(x, y) = 0$ iff $x = y$ and $d(x, y) \leqslant d(x, z) + d(z, y)$.

In order to operate with pairs of truth weights, it was introduced in [CMB22] the notion of **A**-*twisted algebra*. This algebraic structure will play a crucial role in the semantics of our institution, consists of an enrichment of a twist-structure [Kra98] with a metric. The latter is relevant to interpret the consistency operator of the logic:

**Definition 2** ([CMB22])**.** *Given a iMTL-algebra* $\boldsymbol{A}$ *enriched with a metric d, a* **A**-twisted algebra $\mathcal{A} = \langle A \times A, ⩎, ⩏, \Longrightarrow, /\!\!/, D\rangle$ *is defined as:*

– $(a, b) ⩎ (c, d) = (a \sqcap c, b \sqcup d)$
– $(a, b) ⩏ (c, d) = (a \sqcup c, b \sqcap d)$
– $(a, b) \Longrightarrow (c, d) = (a \rightharpoonup c, a \sqcap d)$
– $/\!\!/(a, b) = (b, a)$
– $D((a, b), (c, d)) = \sqrt{d(a, c)^2 + d(b, d)^2}$

*The order in* $\boldsymbol{A}$ *is lifted to* $\mathcal{A}$ *as* $(a, b) \preceq (c, d)$ *iff* $a \leqslant c$ *and* $b \geqslant d$.

### 2.3 Institutional Framing of L($\mathcal{A}$)

Let us fix a given twisted algebra $\mathcal{A}$. In the following subsections we will introduce the ingredients for an institution $\mathrm{L}(\mathcal{A}) = (\mathsf{Sign}, \mathsf{Sen}, \mathsf{Mod}, \models)$.

## Signatures

**Definition 3.** *A signature $\Sigma$ is a pair* $(\mathrm{Prop}, \mathrm{Act})$ *where* Prop *is a set of propositions and* Act *is a set of action symbols. A signature morphism $\sigma : \Sigma \to \Sigma'$ is a pair of functions $\sigma_{\mathrm{Prop}} : \mathrm{Prop} \to \mathrm{Prop}'$ and $\sigma_{\mathrm{Act}} : \mathrm{Act} \to \mathrm{Act}'$.*

The category of signatures and their morphisms will be called signature category and will be denoted by Sign.

## The Models

**Definition 4.** *Let* $(\mathrm{Prop}, \mathrm{Act})$ *be a signature. A* $(\mathrm{Prop}, \mathrm{Act})$-$\mathrm{L}(\mathcal{A})$ *paraconsistent labelled transition system, is a tuple $M = (W, R, V)$ such that,*

- *$W$ is a non-empty set of states,*
- *$R = (R_a \colon W \times W \to A \times A)_{a \in \mathrm{Act}}$ is an* Act*-indexed family of partial functions, given any pair of states $(w_1, w_2) \in W \times W$ and an action $a \in \mathrm{Act}$, relation $R$ assigns a pair $(tt, ff) \in A \times A$ such that $tt$ represents the evidence degree of the transition from $w_1$ to $w_2$ occurring through action $a$ and $ff$ represents the evidence degree of the transition being prevented from occurring.*
- *$V : W \times \mathrm{Prop} \to A \times A$ is a valuation function, that assigns to a proposition $p \in \mathrm{Prop}$ at a given state $w$ a pair $(tt, ff) \in A \times A$ such that $tt$ is the evidence degree of $p$ holding in $w$ and $ff$ the evidence degree of not holding*

*The images of a state through an action $a$ is the set of states for which the transition is defined, i.e. the set $R_a[w] = \{w' \in W \mid R_a(w, w') = (tt, ff)$ for some $\in tt, ff \in A\}$. For any pair $(tt, ff) \in A \times A$, $(tt, ff)^+$ denotes $tt$ and $(tt, ff)^-$ denotes $ff$.*

**Definition 5.** *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two* $(\mathrm{Prop}, \mathrm{Act})$-*PLTS. A morphism between $M$ and $M'$ is a function $h : W \to W'$ compatible with the source valuation and transition functions, i.e.*

- *for each $a \in \mathrm{Act}$, $R_a(w_1, w_2) \preccurlyeq R'_a(h(w_1), h(w_2))$, and*
- *for any $p \in \mathrm{Prop}$, $w \in W$, $V(w, p) \preccurlyeq V'(h(w), p)$.*

We say that $M$ and $M'$ are isomorphic, in symbols $M \cong M'$, whenever there are morphisms $h : M \to M'$ and $h^{-1} : M' \to M$ such that $h' \circ h = id_{W'}$ $h \circ h' = id_W$.

$(\mathrm{Prop}, \mathrm{Act})$-PLTSs and the corresponding morphisms form a category denoted by Mod, which acts as the model category for our $\mathrm{L}(\mathcal{A})$ logic.

**Definition 6.** *Let $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to (\mathrm{Prop}', \mathrm{Act}')$ be a signature morphism and $M' = (W', R', V')$ a* $(\mathrm{Prop}', \mathrm{Act}')$-*PLTS. The $\sigma$-*reduct *of $M'$ is the* $(\mathrm{Prop}, \mathrm{Act})$-*PLTS $M|_\sigma = (W, R, V)$ where*

- *$W = W'$,*
- *for $p \in \mathrm{Prop}$, $w \in W$, $V(w, p) = V'(w, \sigma(p))$, and*
- *for $w, v \in W$ and $a \in \mathrm{Act}$, $R_a(w, v) = R'_{\sigma(a)}(w, v)$.*

Reducts preserve morphism. Hence, each signature morphism $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to (\mathrm{Prop}', \mathrm{Act}')$ defines a functor $\mathsf{Mod}(\sigma) : \mathsf{Mod}(\mathrm{Prop}', \mathrm{Act}') \to \mathsf{Mod}(\mathrm{Prop}, \mathrm{Act})$ that maps systems and morphisms to the corresponding reducts. This lifts to a functor, $\mathsf{Mod} : (\mathsf{Sign})^{op} \to \mathsf{CAT}$, mapping each signature to the category of its models, and each signature morphism to its reduct functor.

**The Sentences.** Once characterised models for $\mathrm{L}(\mathcal{A})$. Let us define its syntax and the satisfaction relation.

**Definition 7.** *Given a signature* $(\mathrm{Prop}, \mathrm{Act})$ *the set* $\mathsf{Sen}(\mathrm{Prop}, \mathrm{Act})$ *of sentences is given by the following grammar*

$$\varphi ::= p \mid \bot \mid \neg\varphi \mid \varphi \to \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid [a]\varphi \mid \langle a\rangle\varphi \mid \cancel{[a]}\varphi \mid \cancel{\langle a\rangle}\varphi \mid \circ\,\varphi$$

*with* $p \in \mathrm{Prop}$ *and* $a \in \mathrm{Act}$. *Note that* $\top = \neg\bot$ *and* $\varphi_1 \leftrightarrow \varphi_2 = (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1)$.

Each signature morphism $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to (\mathrm{Prop}', \mathrm{Act}')$ induces a sentence translation scheme $\mathsf{Sen}(\sigma) : \mathsf{Sen}(\mathrm{Prop}, \mathrm{Act}) \to \mathsf{Sen}(\mathrm{Prop}', \mathrm{Act}')$ recursively defined as follows:

- $\mathsf{Sen}(\sigma)(p) = \sigma_{\mathrm{Prop}}(p)$
- $\mathsf{Sen}(\sigma)(\bot) = \bot$
- $\mathsf{Sen}(\sigma)(\neg\varphi) = \neg\mathsf{Sen}(\sigma)(\varphi)$
- $\mathsf{Sen}(\sigma)(\varphi \odot \varphi') = \mathsf{Sen}(\sigma)(\varphi) \odot \mathsf{Sen}(\sigma)(\varphi')$, $\odot \in \{\vee, \wedge, \to\}$
- $\mathsf{Sen}(\sigma)([a]\,\varphi) = [\sigma_{Act}(a)]\,\mathsf{Sen}(\sigma)(\varphi)$
- $\mathsf{Sen}(\sigma)(\langle a\rangle\varphi) = \langle\sigma_{Act}(a)\rangle\,\mathsf{Sen}(\sigma)(\varphi)$
- $\mathsf{Sen}(\sigma)(\cancel{[a]}\,\varphi) = \cancel{[\sigma_{Act}(a)]}\,\mathsf{Sen}(\sigma)(\varphi)$
- $\mathsf{Sen}(\sigma)(\cancel{\langle a\rangle}\varphi) = \cancel{\langle\sigma_{Act}(a)\rangle}\,\mathsf{Sen}(\sigma)(\varphi)$
- $\mathsf{Sen}(\sigma)(\circ\varphi) = \circ\,\mathsf{Sen}(\sigma)(\varphi)$

which entails a functor $\mathsf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ mapping each signature to the set of its sentences, and each signature morphism to the corresponding translation of sentences.

**The Satisfaction Relation**

**Definition 8.** *Given a signature* $(\mathrm{Prop}, \mathrm{Act})$*, and a* $(\mathrm{Prop}, \mathrm{Act})$*-PLTS* $M = (W, R, V)$*, the satisfaction relation*

$$\models\, : \mathsf{Mod}(\mathrm{Prop}, \mathrm{Act}) \times \mathsf{Sen}(\mathrm{Prop}, \mathrm{Act}) \to A \times A$$

*is defined by*

$$(M \models \varphi) = \bigsqcap_{w \in W} (M, w \models \varphi)$$

*where the relation $\models$ is recursively defined as follows:*

- $(M, w \models p) = V(w, p)$
- $(M, w \models \bot) = (0, 1)$
- $(M, w \models \neg\varphi) = /\!\!/(M, w \models \varphi)$
- $(M, w \models \varphi \to \varphi') = (M, w \models \varphi) \Longrightarrow (M, w \models \varphi')$
- $(M, w \models \varphi \vee \varphi') = (M, w \models \varphi) \Cup (M, w \models \varphi')$
- $(M, w \models \varphi \wedge \varphi') = (M, w \models \varphi) \Cap (M, w \models \varphi')$
- $(M, w \models [a]\varphi) = (\,[a^+](M, w, \varphi^+), \langle a^+\rangle(M, w, \varphi^-)\,)$
- $(M, w \models \langle a\rangle\varphi) = (\,\langle a^+\rangle(M, w, \varphi^+), [a^+](M, w, \varphi^-)\,)$
- $(M, w \models \cancel{[a]}\varphi) = (\,\langle a^-\rangle(M, w, \varphi^-), [a^-](M, w, \varphi^+)\,)$
- $(M, w \models \cancel{\langle a\rangle}\varphi) = (\,[a^-](M, w, \varphi^-), \langle a^-\rangle(M, w, \varphi^+)\,)$
- $(M, w \models \circ\varphi) = \begin{cases} (1, 0) & \textit{if } (M, w \models \varphi) \in \Delta_C \\ (0, 1) & \textit{otherwise} \end{cases}$

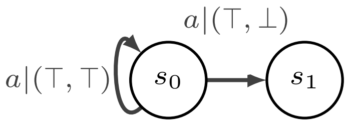
*where*

- $[a^+](M, w, \varphi^*) = \displaystyle\bigsqcap_{w' \in R_a[w]} (R_a^+(w, w') \rightharpoonup (M, w' \models \varphi)^*)$
- $[a^-](M, w, \varphi^*) = \displaystyle\bigsqcap_{w' \in R_a[w]} (R_a^-(w, w') \rightharpoonup (M, w' \models \varphi)^*)$
- $\langle a^+\rangle(M, w, \varphi^*) = \displaystyle\bigsqcup_{w' \in R_a[w]} (R_a^+(w, w') \sqcap (M, w' \models \varphi)^*)$
- $\langle a^-\rangle(M, w, \varphi^*) = \displaystyle\bigsqcup_{w' \in R_a[w]} (R_a^-(w, w') \sqcap (M, w' \models \varphi)^*)$
- $\Delta_C = \{(a, b) \mid D((a, b), (0, 0)) \leqslant D((a, b), (1, 1))\}$

*with $* \in \{^+, ^-\}$. Hence $\varphi$ is valid in $M$ if for any $w \in W$, $(M, w \models \varphi) = (1, 0)$.*

The following examples serve to illustrate the satisfaction relation in our logic.

*Example 1.* Consider **2** the underlying *iMTL-algebra*, a signature $(\{p, q\}, \{a\})$ and a PLTS $M = (\{s_0, s_1\}, R, V)$ depicted in the figure below:



where $V(s_0,p) = (\top,\bot)$, $V(s_0,q) = (\bot,\top)$, $V(s_1,p) = (\top,\top)$ and $V(s_1,q) = (\bot,\bot)$.

$$\begin{aligned}
&M, s_0 \models \cancel{\langle a\rangle} p \vee q \\
=&(M, s_0 \models \cancel{\langle a\rangle} p) \Cup (M, s_0 \models q) \\
=&([a^-](M, s_0, p^-), \langle a^-\rangle(M, s_0, p^+)) \Cup V(s_0, q) \\
=&\Big( (R_a^-(s_0,s_0) \rightharpoonup (M, s_0 \models p)^-) \sqcap (R_a^-(s_0,s_1) \rightharpoonup (M, s_1 \models p)^-), \\
&(R_a^-(s_0,s_0) \sqcap (M, s_0 \models p)^+) \sqcup (R_a^-(s_0,s_1) \sqcap (M, s_1 \models p)^+) \Big) \Cup (\bot,\top) \\
=&\Big( (\top \rightharpoonup \bot) \sqcap (\bot \rightharpoonup \top), (\top \sqcap \top) \sqcup (\bot \sqcap \top) \Big) \Cup (\bot,\top) \\
=&(\bot \sqcap \top, \top \sqcup \bot) \Cup (\bot,\top) = (\bot,\top) \Cup (\bot,\top) = (\bot \sqcup \bot, \top \sqcap \top) = (\bot,\top)
\end{aligned}$$

At state $s_0$ the sentence $\cancel{\langle a\rangle} p \vee q$ holds with evidence degree $\bot$ and doesn't hold with evidence degree $\top$ so we are in a case where the pair of weights are consistent!

*Example 2.* Let **3** be the underlying iMTL-algebra and $M = (\{s_0, s_1\}, R, V)$ be a $(\{p,q,r\},\{a,b\})$-PLTS depicted in the figure below. Where $V(s_0,p) = (\top,\top), V(s_0,q) = V(s_1,p) = (\bot,u)$, $V(s_0,r) = V(s_1,r) = (u,u)$, $V(s_1,q) = (\bot,\bot)$

$$\begin{tikzcd}
s_0 \arrow[loop left, "b|(\top,u)"] \arrow[r, bend left, "a|(\top,\bot)"] & s_1 \arrow[l, bend left, "b|(\top,\top)"] \arrow[loop right, "b|(u,\top)"]
\end{tikzcd}$$

$$\begin{aligned}
(M, s_0 \models r \to (p \vee q)) =&(M, s_0 \models r) \Rightarrow (M, s_0 \models (p \vee q)) \\
=&V(s_0,r) \Rightarrow ((M, s_0 \models p) \Cup (M, s_0 \models q)) \\
=&V(s_0,r) \Rightarrow (V(s_0,p) \Cup V(s_0,q)) \\
=&(u,u) \Rightarrow ((\top,\top) \Cup (\bot,u)) \\
=&(u,u) \Rightarrow (\top \sqcup \bot, \top \sqcap u) \\
=&(u,u) \Rightarrow (\top,u) \\
=&(u \rightharpoonup \top, u \sqcap u) = (\top,u)
\end{aligned}$$

At state $s_0$ the sentence $r \to (p \vee q)$ has an evidence degree $\top$ of holding and it's unknown, $u$, the evidence degree in which it doesn't hold.

Notice that,

$$\begin{aligned}\langle b^+\rangle(M,s_1,p^+) = &\bigsqcup_{s\in R_b[s_1]} (R_b^+(s_1,s)\sqcap(M,s\models p)^-)\\ =&(R_b^+(s_1,s_1)\sqcap(M,s_1\models p)^-)\sqcup(R_b^+(s_1,s_0)\sqcap(M,s_0\models p)^-)\\ =&\Big((u,\top)^+\sqcap(\top,u)^-\Big)\sqcup\Big((\top,\top)^+\sqcap(\top,\top)^-\Big)\\ =&(u\sqcap u)\sqcup(\top\sqcap\top)=\top\end{aligned}$$

Analogously, we can see that $([b^+](M,s_1,p^-))=\top$. Therefore, $(M,s_1\models\langle b\rangle p) = (\langle b^+\rangle(M,s_1,p^+)\,,\,[b^+](M,s_1,p^-))=(\top,\top)$. That is, in state $s_1$ the sentence $\langle b\rangle p$ has evidence degree $\top$ of holding and evidence degree $\top$ of not holding.

**Proposition 1.** *Let* $\sigma:(\mathrm{Prop},\mathrm{Act})\to(\mathrm{Prop}',\mathrm{Act}')$ *be a signature morphism,* $M'$ *a* $(\mathrm{Prop}',\mathrm{Act}')$*-PLTS, and* $\varphi\in\mathsf{Sen}(\mathrm{Prop},\mathrm{Act})$ *a formula. Then, for any* $w\in W$*,*

$$\big(M'|_\sigma,w\models\varphi\big)\;=\;\big(M',w\models\mathsf{Sen}(\sigma)(\varphi)\big)\tag{4}$$

*Proof.* The proof, given by induction on the structure of sentences, is in the appendix.

**Theorem 1.** *For a given metric twisted structure* $\mathcal{A}$*,* $\mathrm{L}(\mathcal{A})$ *is an institution.*

Such abstraction is necessary to get away from the particular syntax of the logic and to focus on building larger specifications in a structured manner.

## 3 Structured Specification with L($\mathcal{A}$)

Usually one starts with flat specifications, that consist of a signature and a set of sentences in a logic, new specifications are then built through a composition of operators. These specification building operators are defined in an arbitrary but fixed institution which allows this theory to be applicable to a wide range of logics that can be framed as institutions.

**Definition 9.** *A specification is a pair*

$$SP=(Sig(SP),Mod(SP))$$

*where* $Sig(SP)$ *is a signature in* Sign *and the models of* $SP$ *is a function*

$$Mod(SP):\mathsf{Mod}(Sig(SP))\to A\times A.$$

*For some model* $M\in\mathsf{Mod}(Sig(SP))$ *we have that* $Mod(SP)(M)=(tt,ff)$*, with* $tt$ *representing the evidence degree of* $M$ *being a model of* $SP$ *and the value* $ff$ *representing the evidence degree of* $M$ *not being a model of* $SP$*.*

Specifications are built in a structured way as follows:

**Flat Specifications** If $\Sigma \in |\mathsf{Sign}|$ is a signature and $\Phi \subseteq \mathsf{Sen}(\Sigma)$ is a set of $\Sigma$-sentences, often called axioms, then $SP = \big(\Sigma, \Phi\big)$ is a flat specification consequently

- $Sig(SP) = \Sigma$
- $Mod(SP)(M) = \left( \bigsqcap_{\varphi \in \Phi} (M \models \varphi) \right) = \left( \bigsqcap_{\varphi \in \Phi} \bigsqcap_{w \in W} (M, w \models \varphi) \right)$

Flat specifications are a basic tool to build small specifications.

**Union** Let $SP$ and $SP'$ be two specifications over the same signature, $\Sigma$. Then $SP \cup SP'$ is

- $Sig(SP \cup SP') = \Sigma$
- $Mod(SP \cup SP')(M) = Mod(SP)(M) \sqcap Mod(SP')(M)$

If $SP_1 = \langle \Sigma, \Phi_1 \rangle$ and $SP_2 = \langle \Sigma, \Phi_2 \rangle$ are flat specifications then:

$$Mod(\langle \Sigma, \Phi_1 \rangle \cup \langle \Sigma, \Phi_2 \rangle)(M) = Mod(\langle \Sigma, \Phi_1 \cup \Phi_2 \rangle)(M)$$

**Translation** If $SP$ is a $\Sigma$-specification and $\sigma : \Sigma \to (\mathrm{Prop}', \mathrm{Act}')$ a signature morphism. Then,

- $Sig(SP \textbf{ with } \sigma) = (\mathrm{Prop}', \mathrm{Act}')$
- $Mod(SP \textbf{ with } \sigma)(M') = Mod(SP)(M'|_\sigma)$

Note that $M'$ is a $(\mathrm{Prop}', \mathrm{Act}') - model$.

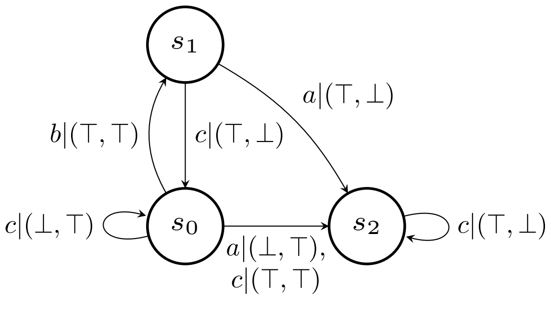
**Hiding** If $SP'$ is a $\Sigma'$-specification and $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to \Sigma'$ is a signature morphism then,

- $Sig(SP' \textbf{ hide via } \sigma) = (\mathrm{Prop}, \mathrm{Act})$
- $Mod(SP' \textbf{ hide via } \sigma)(M) = \Big( \bigsqcup_{N \in M^\sigma} Mod(SP')(N) \Big)$

  where $M^\sigma$ is the class of all $\sigma$-expansions of $M$, i.e. $M^\sigma = \{N \in Mod(SP') \mid N|_\sigma = M\}$.

The following examples illustrate some of the structured specifications operators defined above.

*Example 3.* Consider **2** the underlying iMTL-algebra. Given the signature $\Sigma = (\{p, q\}, \{b, c\})$, the specification $SP = \langle \Sigma, \Phi \rangle$ where $\Phi = \{[c]\cancel{\langle b \rangle}\top,\ \neg(p \vee q),\ q \to \langle c \rangle q\}$ and the inclusion morphism $\sigma : (\{p, q\}, \{b, c\}) \to (\{p, q\}, \{a, b, c\})$. Let $M' = (\{s_0, s_1, s_2\}, R', V')$ be a $(\{p, q\}, \{a, b, c\})$-transition model depicted below.



where $V'(s_0, p) = V'(s_0, q) = V'(s_1, q) = (\top, \top)$, $V'(s_1, p) = (\bot, \bot)$, $V'(s_2, p) = (\top, \bot)$ and $V'(s_2, q) = (\bot, \top)$.

The following $\Sigma$-model $M'|_\sigma = (W, R, V)$ is the $\sigma$-reduct of $M'$:

$$\begin{tikzcd}
s_1 \arrow[d, "c|(\top,\bot)"] \\
s_0 \arrow[u, bend left, "b|(\top,\top)"] \arrow[loop left, "c|(\bot,\top)"] \arrow[r, "c|(\top,\top)"'] & s_2 \arrow[loop right, "c|(\top,\bot)"]
\end{tikzcd}$$

By the definition of $\sigma$-reduct: $W = W'$ and $V(s,r) = V'(s,r)$ for any $s \in W$ and $r \in \{p,q\}$.
Then,

- $Sig(SP \textbf{ with } \sigma) = (\{p,q\},\{a,b,c\})$
- $Mod(SP \textbf{ with } \sigma)(M') = Mod(SP)(M'|_\sigma) = \left( \bigsqcap_{\substack{\varphi\in\Phi \\ s\in W}} (M'|_\sigma, s \models \varphi) \right)$

Notice that,

$$\begin{aligned}
&M'|_\sigma, s_0 \models \xcancel{\langle b \rangle}\top \\
&= ([b^-](M'|_\sigma, s_0, \top^-), \langle b^- \rangle(M'|_\sigma, s_0, \top^+)) \\
&= (R_b^-(s_0,s_1) \to (M'|_\sigma, s_1 \models \top)^-, R_b^-(s_0,s_1) \wedge (M'|_\sigma, \models \top)^+) \\
&= (\top \to \bot, \top \wedge \top) \;=\; (\bot, \top)
\end{aligned}$$

Similarly, we find that $(M'|_\sigma, s_1 \models \xcancel{\langle b \rangle}\top) = (\top,\bot) = (M'|_\sigma, s_2 \models \xcancel{\langle b \rangle}\top)$. Hence,

$$\begin{aligned}
&M'|_\sigma, s_0 \models [c]\xcancel{\langle b \rangle}\top \\
&= ([c^+](M'|_\sigma, s_0, (\xcancel{\langle b \rangle}\top)^+), \langle c^+ \rangle(M'|_\sigma, s_0, (\xcancel{\langle b \rangle}\top)^-)) \\
&= (R_c^+(s_0,s_2) \to (M'|_\sigma, s_2 \models \xcancel{\langle b \rangle}\top)^+, R_c^+(s_0,s_2) \wedge (M'|_\sigma, s_2 \models \xcancel{\langle b \rangle}\top)^-) \\
&= (\top \to \top, \top \wedge \bot) = (\top, \bot)
\end{aligned}$$

Similarly, we can check that $(M'|_\sigma, s_1 \models [c]\xcancel{\langle b \rangle}\top) = (\bot,\top)$ and that $(M'|_\sigma, s_2 \models [c]\xcancel{\langle b \rangle}\top) = (\top,\bot)$. Therefore, there is $\bot$ evidence of sentence $[c]\xcancel{\langle b \rangle}\top$ being true in model $M'|_\sigma$ and evidence $\top$ of being false:

$$(M'|_\sigma \models [c]\xcancel{\langle b \rangle}\top) = \bigsqcap_{s\in W}(M'|_\sigma, s \models [c]\xcancel{\langle b \rangle}\top) \;=\; (\top \wedge \bot \wedge \top, \bot \vee \top \vee \bot) = (\bot, \top)$$

For sentence $\neg(p \vee q)$:

$$\begin{aligned}
&(M'|_\sigma \models \neg(p \vee q)) \\
&= \Big( \Cap_{s \in W} (M'|_\sigma, s \models \neg(p \vee q)) \Big) \\
&= \Big( \Cap_{s \in W} \sslash (M'|_\sigma, s \models p \vee q) \Big) \\
&= \Big( \Cap_{s \in W} \sslash ((M'|_\sigma, s \models p) \Cup (M'|_\sigma, s \models q)) \Big) \\
&= (\sslash(\top \wedge \top, \top \vee \top)) \Cap (\sslash(\bot \wedge \top, \bot \vee \top)) \Cap (\sslash(\top \wedge \bot, \bot \vee \top)) \\
&= (\sslash(\top, \top)) \Cap (\sslash(\bot, \top)) \Cap (\sslash(\bot, \top)) \\
&= (\top, \top) \Cap (\top, \bot) \Cap (\top, \bot) = (\top \wedge \top \wedge \top, \top \vee \bot \vee \bot) = (\top, \top)
\end{aligned}$$

For sentence $q \to \langle c \rangle q$:

$$\begin{aligned}
&M'|_\sigma, s_0 \models q \to \langle c \rangle q \\
&= (M'|_\sigma, s_0 \models q) \Longrightarrow (M'|_\sigma, s_0 \models \langle c \rangle q) \\
&= (\top, \top) \Longrightarrow (\langle c^+ \rangle (M, s_0, q^+), [c^+](M, s_0, q^-)) \\
&= (\top, \top) \Longrightarrow \\
&\quad \left( \bigvee_{s \in W} (R_c^+(s_0, s) \wedge (M'|_\sigma, s \models q)^+), \bigwedge_{s \in W} (R_c^+(s_0, s) \to (M'|_\sigma, s \models q)^-) \right) \\
&= (\top, \top) \Longrightarrow ((\bot \wedge \top) \vee (\bot \wedge \top) \vee (\top \wedge \bot), (\bot \to \top) \wedge (\bot \to \top) \wedge (\top \to \top)) \\
&= (\top, \top) \Longrightarrow (\bot, \top) = (\top \to \bot, \top \wedge \top) = (\bot, \top)
\end{aligned}$$

Similarly, we have that $(M'|_\sigma, s_1 \models q \to \langle c \rangle q) = (\top, \top)$ and $(M'|_\sigma, s_2 \models q \to \langle c \rangle q) = (\top, \bot)$. Therefore, $(M'|_\sigma \models q \to \langle c \rangle q) = (\bot, \top) \Cap (\top, \top) \Cap (\top, \bot) = (\bot, \top)$. In conclusion,

$$\begin{aligned}
&Mod(SP)(M'|_\sigma) \\
&= (M'|_\sigma \models [c]\cancel{\langle b \rangle}\top) \Cap (M'|_\sigma \models \neg(p \vee q)) \Cap (M'|_\sigma \models q \to \langle c \rangle q) \\
&= (\top \wedge \bot \wedge \bot, \top \vee \top \vee \top) = (\bot, \top) \\
&= Mod(SP \textbf{ with } \sigma)(M')
\end{aligned}$$

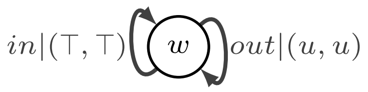
The degree of which there is evidence that model $M'$ is a model of $SP$ **with** $\sigma$, i.e. specification $SP$ translated via the morphism $\sigma$, is $\bot$ and the degree to which there is evidence of $M'$ not being a model of the specification is $\top$.
Notice that in this case we have *consistency*, we are completely certain that $M'$ is not a model of $SP$ **with** $\sigma$, that is, model $M'$ doesn't satisfy the requirements/axioms demanded by $SP$ **with** $\sigma$.

The following example is adapted from [MBHM18] to suit paraconsistent systems and specifications.

*Example 4.* Let **3** be the underlying iMTL-algebra and $\langle \text{Act}, \varnothing \rangle$ a signature where the set of propositions is empty and the set of actions is $\text{Act} = \{in, out\}$ with $\{in\}$ standing for the input of a text file and $\{out\}$ standing for the output of a zip-file.
This example considers a file compressing service working only with text files. Starting with a loose specification $SP_0$ whose requirements are that at any state:

0.1 $[in]\langle out \rangle \top$, whenever a text file is received for compression there has to exist an action where there is an output of a zip-file
0.2 $\langle a \rangle \top$, for some $a \in \{in, out\}$, that is, the system should never terminate

Let $M_0$ be the following model such that the information regarding the input action is inconsistent and the information regarding the output action is vague.



It's possible to check that $(M_0, w \models \langle in \rangle \top) = (\top, \bot)$ and $(M_0, w \models [in]\langle out \rangle \top) = (u, \bot)$. Hence,

$$Mod(SP_0)(M_0) = (u, \bot) \sqcap\!\!\!\sqcap (\top, \bot) = (u \wedge \top, \bot \vee \bot) = (u, \bot)$$

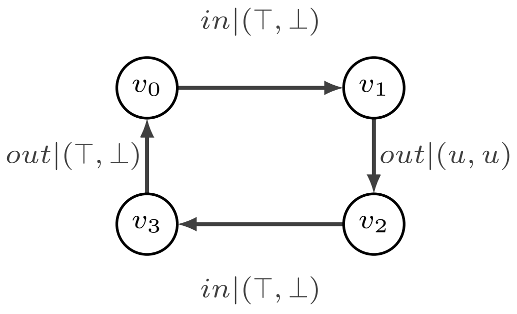
As stated, $SP_0$ is a very loose specification that doesn't demand, for example, that immediately after an output action must come an input action. Because of that we will now consider a new specification. Let $SP_1$ be a specification over $\Sigma$ whose requirement is that at any state:

1.1 $[out](\langle in \rangle \top \wedge [out]\bot)$, whenever there is an output action the system must go on with an input

Let $SP = SP_0 \cup SP_1$ be the union of both specifications. Then,

$$\begin{aligned} Mod(SP)(M_0) &= Mod(SP_0 \cup SP_1)(M_0) \\ &= Mod(SP_0)(M_0) \sqcap\!\!\!\sqcap Mod(SP_1)(M_0) \\ &= (u, \bot) \sqcap\!\!\!\sqcap (\bot, u) = (\bot, u) \end{aligned}$$

If we now consider the following PLTS, $M_1$:



For model $M_1$ we have that:

$$\begin{aligned} Mod(SP_0 \cup SP_1)(M_1) =& Mod(SP_0)(M_1) \Cap Mod(SP_1)(M_1) \\ =& (u,\bot) \Cap (\top,\bot) = (u,\bot) \end{aligned}$$

Since $SP$ results from the union of $SP_0$ and $SP_1$, both flat specifications. $SP_0$ axioms consists of the union of the axioms of $SP_0$ and $SP_1$, (0.1)+(0.2)+(1.1). Note that $Mod(SP)(M_0) \preccurlyeq Mod(SP)(M_1)$, thus there is a higher evidence degree that $M_1$ is a model of $SP$ and a lower evidence degree that $M_1$ isn't a model of $SP$, compared to $M_0$.

## 4 Conclusions

Paraconsistent transition systems [CMB22] were revisited in an institutional framework in order to develop a compositional, structured specification approach for engineering their composition.

Current work includes the study of horizontal and vertical refinement in this institution, as well as normalization structured specifications. Another important extension goes into the domain of observational abstraction: behavioural specifications resort to a notion of observational satisfaction for the axioms of a specification, whereas abstractor specifications define an abstraction from the standard semantics of a specification w.r.t. an observational equivalence relation between algebras.

Adding abstractor and behavioural operators [HMW18] and investigating a proper notion of observational equivalence for these systems is in order.

## Appendix

**Proposition** 1 Let $\sigma : (\text{Prop}, \text{Act}) \to (\text{Prop}', \text{Act}')$ be a signature morphism, $M'$ a $(\text{Prop}', \text{Act}')$-PLTS, and $\varphi \in \mathsf{Sen}(\text{Prop}, \text{Act})$ a formula. Then, for any $w \in W$,

$$\big(M'|_\sigma, w \models \varphi\big) = \big(M', w \models \mathsf{Sen}(\sigma)(\varphi)\big) \tag{5}$$

*Proof.* The proof is by induction over the structure of sentences. To simplify notation we will write $\sigma(p)$ instead of $\sigma_{\text{Prop}}(p)$ for any $p \in \text{Prop}$ and $\sigma(a)$ instead of $\sigma_{\text{Act}}(a)$ for any $a \in \text{Act}$. The case of $\bot$ is trivial, by the definition of $\models$ and $\mathsf{Sen}$ we have that $(M'|_\sigma, w \models \bot) = (0,1) = (M', w \models Sen(\sigma)(\bot))$. For sentences $p \in \text{Prop}$, one observes that by defn of $\mathsf{Sen}$, of $\models$ and of reducts, $(M', w \models \mathsf{Sen}(\sigma)(p)) = (M', w \models \sigma(p)) = V'(w, \sigma(p)) = V(w,p) = (M'|_\sigma, w \models p)$. For sentences $\neg\varphi$ we observe that, by definition of $\mathsf{Sen}$ and of $\models$, we have that $(M', w \models \mathsf{Sen}(\sigma)(\neg\varphi)) = M', w \models \neg\mathsf{Sen}(\sigma)(\varphi) = (/\!\!/(M', w \models \mathsf{Sen}(\sigma)(\varphi)))$. By induction hypothesis $(/\!\!/(M', w \models \mathsf{Sen}(\sigma)(\varphi))) = /\!\!/(M'|_\sigma, w \models \varphi)$ and, again, by definition of $\mathsf{Sen}$ and of $\models$, we have $/\!\!/(M'|_\sigma, w \models \varphi) = (M'|_\sigma, w \models \neg\varphi)$.

Let us consider now formulas composed by Boolean operators. Firstly, we can observe that, by definition of $\mathsf{Sen}$ and of $\models$, $(M', w \models \mathsf{Sen}(\sigma)(\varphi \wedge \varphi')) = (M', w \models \mathsf{Sen}(\sigma)(\varphi) \wedge \mathsf{Sen}(\sigma)(\varphi')) = (M', w \models \mathsf{Sen}(\sigma)(\varphi)) \sqcap\!\!\sqcap (M', w \models \mathsf{Sen}(\sigma)(\varphi'))$. By I.H. we have that $(M', w \models \mathsf{Sen}(\sigma)(\varphi')) = (M'|_\sigma, w \models \varphi) \sqcap\!\!\sqcap (M'|_\sigma, w \models \varphi')$ and by definition of $\models$, it is equal to $M'|_\sigma, w \models (\varphi \wedge \varphi')$. The proof for sentences $\varphi \vee \varphi'$ and $\varphi \to \varphi'$ is analogous.

$$
\begin{aligned}
& M', w \models \mathsf{Sen}(\sigma)([a]\,\varphi) \\
= \;& \{\text{defn of } \mathsf{Sen}\} \\
& M', w \models [\sigma(a)]\,\mathsf{Sen}(\sigma)(\varphi) \\
= \;& \{\text{defn of } \models\} \\
& ([\sigma(a)^+](M', w, \mathsf{Sen}(\sigma)(\varphi)^+)\,,\, \langle\sigma(a)^+\rangle(M', w, \mathsf{Sen}(\sigma)(\varphi)^-)) \\
= \;& \{\text{def. of } [a^+] \text{ and } \langle a^+\rangle\} \\
& \Bigg( \bigsqcap_{w' \in R'_{\sigma(a)}[w]} (R'^+_{\sigma(a)}(w,w') \rightharpoonup (M', w' \models \mathsf{Sen}(\sigma)(\varphi))^+), \\
& \quad \bigsqcup_{w' \in R'_{\sigma(a)}[w]} (R'^+_{\sigma(a)}(w,w') \sqcap (M', w' \models \mathsf{Sen}(\sigma)(\varphi))^-) \Bigg) \\
= \;& \{(\text{step } \star)\} \\
& \Bigg( \bigsqcap_{w' \in R_a[w]} (R^+_a(w,w') \rightharpoonup (M'|_\sigma, w \models \varphi)^+), \bigsqcup_{w' \in R_a[w]} (R^+_a(w,w') \sqcap (M'|_\sigma, w \models \varphi)^-) \Bigg) \\
= \;& \{\text{def. } [a^+] \text{ and } \langle a^+\rangle\} \\
& ([a^+](M'|_\sigma, w, \varphi^+)\,, \langle a^+\rangle(M'|_\sigma, w, \varphi^-)) \\
= \;& \{\text{defn of } \models\} \\
& M'|_\sigma, w \models [a]\,\varphi
\end{aligned}
$$

(step $\star$) We have by reduct that $R'_{\sigma(a)}[w] = R_a[w]$. Moreover, by I.H., it is true that $(M', w \models \mathsf{Sen}(\sigma)(\varphi)) = (M'|_\sigma, w \models \varphi)$, and hence

$$
\begin{aligned}
\Big((M', w \models \mathsf{Sen}(\sigma)(\varphi))^+, (M', w \models \mathsf{Sen}(\sigma)(\varphi))^-\Big) = \\
\Big((M'|_\sigma, w \models \varphi)^+, (M'|_\sigma, w \models \varphi)^-\Big)
\end{aligned}
\tag{6}
$$

Therefore, $M', w \models \mathsf{Sen}(\sigma)(\varphi))^+ = (M'|_\sigma, w \models \varphi)^+$ and $(M', w \models \mathsf{Sen}(\sigma)(\varphi))^- = (M'|_\sigma, w \models \varphi)^-$.

$$
\begin{aligned}
& M', w \models \mathsf{Sen}(\sigma)(\cancel{[a]}\,\varphi) \\
= {} & \{\text{defn of } \mathsf{Sen}\} \\
& M', w \models \cancel{[\sigma(a)]}\,\mathsf{Sen}(\sigma)(\varphi) \\
= {} & \{\text{defn of } \models\} \\
& (\langle \sigma(a)^- \rangle (M', w, \mathsf{Sen}(\sigma)(\varphi)^-), [\sigma(a)^-](M', w, \mathsf{Sen}(\sigma)(\varphi)^+)) \\
= {} & \{\text{def. of } [a^-] \text{ and } \langle a^- \rangle\} \\
& \Bigg( \bigsqcup_{w' \in R'_{\sigma(a)}[w]} (R'^{-}_{\sigma(a)}(w, w') \sqcap (M', w' \models \mathsf{Sen}(\sigma)(\varphi))^-), \\
& \bigsqcap_{w' \in R'_{\sigma(a)}[w]} (R'^{-}_{\sigma(a)}(w, w') \rightharpoonup (M', w' \models \mathsf{Sen}(\sigma)(\varphi))^+) \Bigg) \\
= {} & \{\text{analogous to (step } \star)\} \\
& (\langle a^- \rangle (M'|_\sigma, w, \varphi^-)\,,\ [a^-](M'|_\sigma, w, \varphi^+)) \\
= {} & \{\text{defn of } \models\} \\
& M'|_\sigma, w \models \cancel{[a]}\,\varphi
\end{aligned}
$$

The proofs for sentences $\langle a \rangle \varphi$ and $\cancel{\langle a \rangle}\varphi$ are analogous.
Finally, let us consider the proof for sentences $\circ\,\varphi$. By definition of $\mathsf{Sen}$, $M', w \models \mathsf{Sen}(\sigma)(\circ\,\varphi) = (M', w \models \circ\,\mathsf{Sen}(\sigma)(\varphi))$. By definition of $\models$, this evaluates to $(1, 0)$, if $(M', w \models \mathsf{Sen}(\sigma)(\varphi)) \in \Delta_C$ and to $(0, 1)$ otherwise. Hence, by I.H, it evaluates to $(1, 0)$ when $(M'|_\sigma, w \models \varphi) \in \Delta_C$ and to $(0, 1)$, i.e., we have $(M'|_\sigma, w \models \circ\,\varphi)$.

## References

[ACEGG91] Agustí-Cullell, J., Esteva, F., Garcia, P., Godo, L.: Formalizing multiple-valued logics as institutions. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) IPMU 1990. LNCS, vol. 521, pp. 269–278. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0028112

[Aka16] Akama, S. (ed.): Towards Paraconsistent Engineering. ISRL, vol. 110. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40418-9

[BEGR09] Bou, F., Esteva, F., Godo, L., Rodríguez, R.O.: On the minimum many-valued modal logic over a finite residuated lattice. J. Logic Comput. **21**(5), 739–790 (2009)

[CCM07] Carnielli, W., Coniglio, M.E., Marcos, J.: Logics of formal inconsistency. In: Handbook of Philosophical Logic, pp. 1–93 (2007)

[CMB22] Cruz, A., Madeira, A., Barbosa, L.S.: A logic for paraconsistent transition systems. In: Indrzejczak, A., Zawidzki, M. (eds.) 10th International Conference on Non-Classical Logics. Theory and Applications, vol. 358 of EPTCS, pp. 270–284 (2022)

[EG01] Esteva, F., Godo, L.: Monoidal t-norm based logic: towards a logic for left-continuous t-norms. Fuzzy Sets Syst. **124**, 271–288 (2001)

[GB92] Goguen, J.A., Burstall, R.M.: Institutions: abstract model theory for specification and programming. J. ACM **39**(1), 95–146 (1992)

[HMW18] Hennicker, R., Madeira, A., Wirsing, M.: Behavioural and abstractor specifications revisited. Theor. Comp. Sc. **741**, 32–43 (2018)

[Jas69] Jaskowski, S.: Propositional calculus for contradictory deductive systems (communicated at the meeting of march 19, 1948). Studia Logica: Int. J. Symb. Logic **24**, 143–160 (1969)

[JGMB21] Jain, M., Gomes, L., Madeira, A., Barbosa, L.S.: Towards a specification theory for fuzzy modal logic. In: International Symposium on Theoretical Aspects of Software Engineering, TASE 2021, pp. 175–182. IEEE (2021)

[Kra98] Kracht, M.: On extensions of intermediate logics by strong negation. J. Phil. Logic **27**(1), 49–73 (1998)

[MBHM18] Madeira, A., Barbosa, L.S., Hennicker, R., Martins, M.A.: A logic for the stepwise development of reactive systems. Theor. Comput. Sci. **744**, 78–96 (2018)

[MHST03] Mossakowski, T., Haxthausen, A., Sannella, D., Tarlecki, A.: CASL, the common algebraic specification language: semantics and proof theory. Comput. Inf. **22**, 285–321 (2003)

[RJJ15] Rivieccio, U., Jung, A., Jansana, R.: Four-valued modal logic: Kripke semantics and duality. J. Logic Comput. **27**(1), 155–199 (2015)

[ST12] Sannella, D., Tarlecki, A.: Foundations of Algebraic Specification and Formal Software Development. Monographs on TCS, an EATCS Series. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-17336-