Towards a specification theory for fuzzy modal logic

Manisha Jain
CIDMA, Dep Mathematics & INL
University of Aveiro
Aveiro, Portugal
manu1328@outlook.com

Leandro Gomes
INESC TEC
University of Minho
Braga, Portugal
leandro.r.gomes@inesctec.pt

Alexandre Madeira
CIDMA, Dep Mathematics
University of Aveiro
Aveiro, Portugal
madeira@ua.pt

Luis S. Barbosa
INESC TEC
University of Minho
Braga, Portugal
lsb@di.uminho.pt

2021 International Symposium on Theoretical Aspects of Software Engineering (TASE) | 978-1-6654-4163-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/TASE52547.2021.00031

***Abstract*—Fuzziness, as a way to express imprecision, or uncertainty, in computation is an important feature in a number of current application scenarios: from hybrid systems interfacing with sensor networks with error boundaries, to knowledge bases collecting data from often non-coincident human experts. Their abstraction in e.g. fuzzy transition systems led to a number of mathematical structures to model this sort of systems and reason about them. This paper adds two more elements to this family: two modal logics, framed as institutions, to reason about fuzzy transition systems and the corresponding processes. This paves the way to the development, in the second part of the paper, of an associated theory of structured specification for fuzzy computational systems.**

***Index Terms*—Dynamic Logic, Fuzzy Logic, Specification.**

I. Introduction

The control of systems dealing with some form of imprecision or uncertainty are suitably modelled by fuzzy transition systems. For example, the requirement *"if the water flow is too high, slightly close the valve"* has its qualifiers represented by fuzzy sets in which membership is relative, i.e. established up to a certain degree. Thus, for example, a water flow of $0.6\,m^3/s$ can be simultaneously considered *too high* with, say, a membership degree of 0.7, and *leaning to high* with a degree of 0.3. Examples of applications in which this sort of behaviour is present include clinical decision support for medical diagnosis [VMA10], and the control of a robot in a labyrinth [CAF13]. The pervasiveness of this sort of behaviour entails the need for not only a suitable logic, but also formal specification methodology along the lines of e.g. the work of D. Sannella and A. Tarlecki [ST12]. This paper is a step in that direction.

Since originally proposed by L. Zadeh [Zad65], fuzzy logic emerged as na expressive setting for both fundamental and applied research in fuzzy systems — see [Ata20] and [DETRK15] for recent accounts, and [BDK17] for a historical overview. Based on the observation that people make decisions on imprecise and non-numerical information, fuzzy logic allows to express precisely the vagueness of properties like *"how close two cities are from each other"*, or the *water flow* requirement mentioned above.

Fuzzy transition systems and corresponding (fuzzy) logics are addressed in different flavours in e.g. [DK05], [WD16] and [WC14]. The first distinguishes between different classes of fuzzy automata, with *fuzziness* itself introduced at different levels. The second introduces suitable notions of bisimulation and their logical characterisation, framed coalgebraically in the last reference.

This paper revisits *fuzzy transition systems* and *fuzzy processes*, the latter identifying an initial state and bound by a reachability constraint. Two fuzzy modal logics are then introduced for systems and processes, designated by $\mathcal{FML}$ and $\mathcal{PFML}$, respectively. $\mathcal{PFML}$ generalises our previous work [MBHM16], which combines modalities with regular expressions, typical of dynamic logic, and binders in state variables to explicitly refer to states in formulæ, as in hybrid logic [Bra10]. A fuzzy hybrid modal logic was originally introduced in [Lia01]. However, it used crisp nominals (i.e. constants on the states) rather than crisp state variables, as proposed here.

Both $\mathcal{FML}$ and $\mathcal{PFML}$ are used as *structured specification logics* to build systems with fuzzy behaviour in a compositional way. Hence both logics are framed as a particular sort of institutions [GB92], known as *many-valued institutions* [Dia13], in which the satisfaction condition is generalised from the standard, Boolean setting to a weighted one. Some steps towards a theory of structured specifications of fuzzy systems are undertaken through the introduction of well known CASL-like operators [ST12], and a discussion of horizontal and vertical composition. Behavioural and abstract implementations of specifications of fuzzy transition systems and processes are also developed, along the path introduced in [HMW18].

Outline. The paper is organized as follows. Section II introduces logics $\mathcal{FML}$ and $\mathcal{PFML}$ and proves they form many-valued institutions. Bisimilarity and a quotient construction in the corresponding model categories are discussed in Section III. Section IV develops the basis of a corresponding

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER- 029946 - PTDC/CCI-COM/29946/2017.

978-1-6654-4163-6/21/$31.00 ©2021 IEEE
DOI 10.1109/TASE52547.2021.00031

structured specification framework. Finally, Section V concludes with some lines for future work.

II. Two fuzzy modal logics

A. $\mathcal{FML}$, a logic for fuzzy transition systems

$\mathcal{FML}$ is the basic language of fuzzy transition systems, i.e. labelled transition systems whose transitions are weighted in the real interval $[0,1]$.

A fuzzy transition system is syntactically supported by two disjoint sets Prop and Act of proposition and action symbols, respectively. Jointly they define the system signature $(\mathrm{Prop},\mathrm{Act})$. Any pair of functions $\sigma_{\mathrm{Prop}} : \mathrm{Prop} \to \mathrm{Prop}'$ and $\sigma_{\mathrm{Act}} : \mathrm{Act} \to \mathrm{Act}'$ define a signature morphism $\sigma : (\mathrm{Prop},\mathrm{Act}) \to (\mathrm{Prop}',\mathrm{Act}')$, through which the language of a system can be mapped into the language of another. Clearly, signatures and signature morphisms define a category, denoted by Sign, whose structure is inherited from Set, the familiar category of sets and set-theoretic functions.

Definition 1. *Let* $(\mathrm{Prop},\mathrm{Act})$ *be a signature. A* $(\mathrm{Prop},\mathrm{Act})$-fuzzy transition system *is a tuple* $M = (W,R,V)$ *such that,*

- W *is a non-empty set of states,*
- $R = (R_a\colon W \times W \to [0,1])_{a\in\mathrm{Act}}$ *is an* Act*-indexed family of weighted transition functions.*
- $V : W \times \mathrm{Prop} \to [0,1]$ *is a valuation function, assigning a weight in* $[0,1]$ *to a proposition in a given state.*

A morphism connecting two $(\mathrm{Prop},\mathrm{Act})$-fuzzy transition systems (W,R,V) and (W',R',V') is a function $h : W \to W'$ compatible with the source valuation and transition functions, i.e.

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

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 $h' \circ h = id_W$ and $h \circ h' = id_{W'}$.

$(\mathrm{Prop},\mathrm{Act})$-fuzzy transition systems and the corresponding morphisms form a category denoted by $\mathsf{Mod}^{\mathcal{FML}}(\mathrm{Prop},\mathrm{Act})$ (or simply $\mathsf{Mod}(\mathrm{Prop},\mathrm{Act})$ when clear in the context), which acts as the model category for $\mathcal{FML}$. Any signature morphism σ defines a *model reduct*, i.e. a canonical way to see a system (a model) through the lens provided by σ applied to another one. Formally,

Definition 2. *Let* $\sigma : (\mathrm{Prop},\mathrm{Act}) \to (\mathrm{Prop}',\mathrm{Act}')$ *be a signature morphism and* $M' = (W',R',V')$ *a* $(\mathrm{Prop}',\mathrm{Act}')$-*fuzzy transition system. The* σ-reduct of M' *is the* $(\mathrm{Prop},\mathrm{Act})$-*fuzzy transition system* $\mathsf{Mod}(\sigma)(M') = (W,R,V)$ *where*

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

Reducts preserve morphisms in the sense that, for each morphism $h : M'_1 \to M'_2$, there is a morphism $h' : \mathsf{Mod}(\sigma)(M'_1) \to \mathsf{Mod}(\sigma)(M'_2)$, which is the restriction of h to the states of $\mathsf{Mod}(\sigma)(M'_1)$. Hence, each signature morphism $\sigma : (\mathrm{Prop},\mathrm{Act}) \to (\mathrm{Prop}',\mathrm{Act}')$ defines a functor $\mathsf{Mod}(\sigma) : \mathsf{Mod}(\mathrm{Prop}',\mathrm{Act}') \to \mathsf{Mod}(\mathrm{Prop},\mathrm{Act})$ mapping systems and morphisms to the corresponding reducts. More generally, as one would expect, this lifts to a contravariant 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.

Once characterised models for $\mathcal{FML}$, let us define its syntax and the satisfaction relation.

Definition 3. *Given a signature* $(\mathrm{Prop},\mathrm{Act})$ *the set* $\mathsf{Sen}^{\mathcal{FML}}(\mathrm{Prop},\mathrm{Act})$ *of sentences is given by the grammar*

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \to \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a\rangle\varphi \mid [a]\varphi$$

with $a \in \mathrm{Act}, p \in \mathrm{Prop}$. *As usual,* $\perp$ *abbreviates* $\neg\top$.

Each signature morphism $\sigma : (\mathrm{Prop},\mathrm{Act}) \to (\mathrm{Prop}',\mathrm{Act}')$ induces a translation scheme

$$\mathsf{Sen}^{\mathcal{FML}}(\sigma) : \mathsf{Sen}^{\mathcal{FML}}(\mathrm{Prop},\mathrm{Act}) \to \mathsf{Sen}^{\mathcal{FML}}(\mathrm{Prop}',\mathrm{Act}')$$

recursively defined as follows:

- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(p) = \sigma_{\mathrm{Prop}}(p)$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\top) = \top$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\neg\varphi) = \neg\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi\to\varphi') = \mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)\to\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi')$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi\vee\varphi') = \mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)\vee\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi')$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi\wedge\varphi') = \mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)\wedge\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi')$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\langle a\rangle\varphi) = \langle\sigma_{\mathrm{Act}}(a)\rangle\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)$
- $\mathsf{Sen}^{\mathcal{FML}}(\sigma)([a]\varphi) = [\sigma_{\mathrm{Act}}(a)]\mathsf{Sen}^{\mathcal{FML}}(\sigma)(\varphi)$

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.

Definition 4. *Given a signature* $(\mathrm{Prop},\mathrm{Act})$, *and a* $(\mathrm{Prop},\mathrm{Act})$-*fuzzy transition system* $M = (W,R,V)$, *the weighted satisfaction relation*

$$\models^{\mathcal{FML}}_{(\mathrm{Prop},\mathrm{Act})} : \mathsf{Mod}(\mathrm{Prop},\mathrm{Act}) \times \mathsf{Sen}^{\mathcal{FML}}(\mathrm{Prop},\mathrm{Act}) \to [0,1]$$

is defined by $(M \models^{\mathcal{FML}}_{(\mathrm{Prop},\mathrm{Act})} \varphi) = \mathsf{MIN}_{w\in W}(M,w \models \varphi)$ *where* $\models$ *is recursively defined as follows,*

- $(M,w \models p) = V(w,p)$, *for* $p \in \mathrm{Prop}$
- $(M,w \models \top) = 1$
- $(M,w \models \neg\varphi) = \mathsf{N}(M,w \models \varphi)$
- $(M,w \models \varphi \to \varphi') = \mathsf{I}((M,w \models \varphi),(M,w \models \varphi'))$
- $(M,w \models \varphi \vee \varphi') = \mathsf{max}((M,w \models \varphi),(M,w \models \varphi'))$
- $(M,w \models \varphi \wedge \varphi') = \mathsf{min}((M,w \models \varphi),(M,w \models \varphi'))$
- $(M,w \models \langle a\rangle\varphi) = \mathsf{MAX}_{w'\in W}\big\{\mathsf{min}((R_a(w,w'),(M,w' \models \varphi))\big\}$
- $(M,w \models [a]\varphi) = \mathsf{MIN}_{w'\in W}\big\{\mathsf{I}(R_a(w,w'),(M,w' \models \varphi))\big\}$

where auxiliary functions N,I *over* $[0,1]$ *are given by* $\mathsf{I}(x,y) = \begin{cases}1 & x \le y\\ y & \textit{otherwise}\end{cases}$ *and* $\mathsf{N}(x) = \begin{cases}1 & x = 0\\ 0 & \textit{otherwise}\end{cases}$ *and* MAX *and* MIN *are the monoidal reductions of the binary functions* max *and* min.

We have, therefore, framed the fuzzy modal logic $\mathcal{FML}$ as an institution. Actually,

Theorem 1. *Let* $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to (\mathrm{Prop}', \mathrm{Act}')$ *be a signature morphism,* M' *a* $(\mathrm{Prop}', \mathrm{Act}')$*-fuzzy transition structure, and* $\varphi \in \mathsf{Sen}(\mathrm{Prop}, \mathrm{Act})$ *a formula. Then,*

$$\big(\mathsf{Mod}(\sigma)(M') \models^{\mathcal{FML}}_{(\mathrm{Prop},\mathrm{Act})} \varphi\big) \;=\; \big(M' \models^{\mathcal{FML}}_{(\mathrm{Prop}',\mathrm{Act}')} \mathsf{Sen}(\sigma)(\varphi)\big) \tag{1}$$

Proof. According to the definition of $\models^{\mathcal{FML}}$ it is enough to prove that for any $w \in W$, $\big(\mathsf{Mod}(\sigma)(M'), w \models \varphi\big) = \big(M', w \models \mathsf{Sen}(\sigma)(\varphi)\big)$. The proof is by induction over the structure of sentences. The case of $\top$ is trivial, and for propositions one observes that $\big(M', w \models \mathsf{Sen}(\sigma)(p)\big) = \big(M', w \models \sigma(p)\big) = V'(w, \sigma(p))$. Then, by definition of reduct, this is equal to $V(w,p)$, i.e., $\mathsf{Mod}(\sigma)(M'), w \models p$. The other cases are proven by application inductively. For instance: By definition of Sen, $M', w \models \mathsf{Sen}(\sigma)(\langle a\rangle\varphi) = M', w \models \langle a\rangle \mathsf{Sen}(\sigma)(\varphi)$, i.e. $\mathsf{MAX}_{w'\in W}\big\{\mathsf{min}(R'_{\sigma(a)}(w,w'), (M', w' \models \mathsf{Sen}(\varphi))\big\}$. By definition of reduct and the induction hypothesis, yields $\mathsf{MAX}_{w'\in W}\big\{\mathsf{min}(R_a(w,w'), (\mathsf{Mod}(M'), w' \models \varphi)\big\}$ i.e. $\mathsf{Mod}(\sigma)(M'), w \models \langle a\rangle\varphi$ □

B. $\mathcal{PFML}$, a logic for fuzzy processes

A process is a 'system in action', which means it comes equipped with an initial state from where its behaviour unfolds and every other state is reachable. Therefore, a logic for fuzzy processes restricts models to reachable fuzzy transition systems, and introduces crisp state variables and state binders. Let us start by formalising *reachability* in fuzzy transition systems. We say that a state w is reachable in a $(\mathrm{Prop}, \mathrm{Act})$-fuzzy transition system $M = (W, R, V)$ if there are $n \geq 0$, $a_1, \ldots, a_n \in \mathrm{Act}$, and $w_1, \ldots w_n \in W$ such that, for any $i \in \{0, \ldots, n-1\}$, $R_{a_{i+1}}(w_i, w_{i+1}) > 0$ and $w = w_n$. The w*-restriction of* M is the fuzzy transition system $M[w] = (W[w], R[w], V[w])$, where $W[w] \subseteq W$ is the set of the w-reachable states of M, for any $a \in \mathrm{Act}$, $(R[w])_a = R_a \cap (W[w] \times W[w])$ and, for any $w \in W[w]$, $V[w](w,p) = V(w,p)$.

Definition 5. *A* $(\mathrm{Prop}, \mathrm{Act})$-fuzzy process *is a tuple* $P = (W, R, V, w_0)$*, where* (W, R, V) *is a* $(\mathrm{Prop}, \mathrm{Act})$*-fuzzy transition system,* $w_0 \in W$ *and* W *is* w_0*-reachable.*

Morphisms relating fuzzy processes are just morphisms between the underlying fuzzy transition systems that preserve initial states. However, reducts as proposed in Definition 2 do not preserve reachability. The following definition makes the necessary adjustment.

Definition 6. *Let* $\sigma : (\mathrm{Prop}, \mathrm{Act}) \longrightarrow (\mathrm{Prop}', \mathrm{Act}')$ *be a signature morphism and* $P' = (W', R', V', w'_0)$ *a* $(\mathrm{Prop}', \mathrm{Act}')$*-process. The* σ-reduct of P' *is the* $(\mathrm{Prop}, \mathrm{Act})$*-fuzzy process* $\mathsf{Mod}(\sigma)(P) = (W, R, V, w_0)$ *such that* $w_0 = w'_0$ *and* (W, R, V) *is the* w_0*-restriction of the* σ*-reduct of* (W', R', V').

We may now define the language for $\mathcal{PFML}$ and the corresponding satisfaction relation. The former extends that of $\mathcal{FML}$ with a set X of state variables and binders. Thus $\mathcal{PFML}$ formulæ are generated by

$$\varphi ::= x \mid {\downarrow} x.\varphi \mid p \mid \top \mid \neg\varphi \mid \varphi \to \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle a\rangle\varphi \mid [a]\varphi$$

with $a \in \mathrm{Act}, p \in \mathrm{Prop}$. As usual, formulæ without free variables are called sentences and collected in $\mathsf{Sen}^{\mathcal{PFML}}(\mathrm{Prop}, \mathrm{Act})$, for a given signature $(\mathrm{Prop}, \mathrm{Act})$.

For any signature morphism σ, the sentence translation $\mathsf{Sen}^{\mathcal{PFML}}(\sigma)$ is defined by $\mathsf{Sen}^{\mathcal{PFML}}(x) = x$ and by $\mathsf{Sen}^{\mathcal{PFML}}({\downarrow} x.\varphi) = {\downarrow} x.\mathsf{Sen}^{\mathcal{PFML}}(\varphi)$. The mapping of the other sentences is defined as for $\mathsf{Sen}^{\mathcal{FML}}(\sigma)$.

Similarly, the satisfaction relation extends Definition 4 with two cases corresponding precisely to state variables and binders. Thus, a proper valuation of variables in states $g : X \longrightarrow W$ is required.

Definition 7. *Given a signature* $(\mathrm{Prop}, \mathrm{Act})$*, the satisfaction relation is given by*

$$(P \models^{\mathcal{PFML}}_{(\mathrm{Prop},\mathrm{Act})} \varphi) \;=\; \mathsf{MIN}_{g \in W^X}(P, g, w_0 \models \varphi)$$

where $\models$ *extends the corresponding relation used in the definition of* $\models^{\mathcal{FML}}_{(\mathrm{Prop},\mathrm{Act})}$ *by introducing variable valuations as a parameter, and the following new cases*

$$(P, g, w \models x) \;=\; \begin{cases} 1 & \textit{if } g(x) = w \\ 0 & \textit{otherwise} \end{cases}$$

$$(P, g, w \models {\downarrow} x.\varphi) \;=\; (P, g[x \mapsto w], w \models \varphi)$$

where $g[x \mapsto w](x) = w$ *and* $g[x \mapsto w](y) = g(y)$ *for any other* $y \neq x \in X$.

As expected, $\mathcal{PFML}$ also forms an institution.

Theorem 2. *Let* $\sigma : (\mathrm{Prop}, \mathrm{Act}) \to (\mathrm{Prop}', \mathrm{Act}')$ *be a signature morphism,* $P' \in \mathsf{Mod}^{\mathcal{PFML}}(\mathrm{Prop}', \mathrm{Act}')$ *a fuzzy process and* $\varphi \in \mathsf{Sen}^{\mathcal{PFML}}(\mathrm{Prop}, \mathrm{Act})$*. Then*

$$\big(\mathsf{Mod}(\sigma)(P') \models^{\mathcal{PFML}}_{(\mathrm{Prop},\mathrm{Act})} \varphi\big) = \big(P' \models^{\mathcal{PFML}}_{(\mathrm{Prop}',\mathrm{Act}')} \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(\varphi)\big) \tag{2}$$

Proof. By definition of $\models^{\mathcal{PFML}}$ it is enough to prove that, for any $w \in W$, and $g : X \to W'$,

$$\big(\mathsf{Mod}(\sigma)(M'), g, w_0 \models \varphi\big) \;=\; \big(M', g, w_0 \models \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(\varphi)\big) \tag{3}$$

The proof is by induction over the structure of sentences. For the case of state variables, we know that $M', w \models \mathsf{Sen}(\sigma)(x)$ is either 1 or 0. Thus,

$$\begin{array}{ll} & \big(M', g, w \models \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(x)\big) = 1 \\ \Leftrightarrow & \quad \{\text{ defn of Sen}\} \\ & \big(M', g, w \models x\big) = 1 \\ \Leftrightarrow & \quad \{\text{ defn.} \models\} \\ & g(x) = w \\ \Leftrightarrow & \quad \{\text{ defn} \models\} \\ & \big(\mathsf{Mod}(M'), g, w \models x\big) = 1 \end{array}$$

and analogously, $\big(M', g, w \models \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(x)\big) = 0 \Leftrightarrow \big(\mathsf{Mod}(M'), g, w \models x\big) = 0$. Hence, $(M', g, w \models x) = (\mathsf{Mod}(M'), g, w \models x)$. The case for binder is as follows.

$$
\begin{aligned}
& M', g, w \models \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(\downarrow x.\varphi) \\
= \quad & \{ \text{ defn of } \mathsf{Sen} \} \\
& M', g, w \models \downarrow x.\mathsf{Sen}^{\mathcal{PFML}}(\sigma)(\varphi) \\
= \quad & \{ \models \text{defn.} \} \\
& M', g[x \to w], w \models \mathsf{Sen}^{\mathcal{PFML}}(\sigma)(\varphi) \\
= \quad & \{ \text{ I.H.} \} \\
& \mathsf{Mod}(\sigma)(M'), g[x \to w], w \models \varphi \\
= \quad & \{ \text{ defn. } \models \} \\
& \mathsf{Mod}(\sigma)(M'), w \models \downarrow x.\varphi
\end{aligned}
$$

The remaining cases are proved similarly to the satisfaction condition for $\mathcal{FML}$. □

III. Bisimulation and Quotient

The study of behavioural equivalences is crucial to support reuse, refinement and minimization of transition systems. This section characterises what it means for a relation between two states to be a bisimulation, and discusses the relationship with modal equivalence and model quotients. Let us start with the basic definition, extending to the multi-modal case the characterization introduced in [JMM20].

Definition 8. *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two $(\mathrm{Prop}, \mathrm{Act})$-fuzzy transition systems. A relation $E \subseteq W \times W'$ is a* bisimulation between M and M', *whenever $w\ E\ w'$,*

(Atom) $V(w, p) = V'(w', p)$, *for any $p \in \mathrm{Prop}$*

(Fzig) *for any $a \in \mathrm{Act}$, $u \in W$,*

$$R_a(w, u) \le \mathsf{MAX}\{R'_a(w', u') \mid \text{ for any } u' \text{ st } u\ E\ u'\}$$

(Fzag) *for any $a \in \mathrm{Act}$, $u' \in W'$,*

$$R'_a(w', u') \le \mathsf{MAX}\{R_a(w, u) \mid \text{ for any } u \text{ st } u\ E\ u'\}$$

Definition 9. *Two fuzzy transition systems $M = (W, R, V)$ and $M' = (W', R', V')$ are* behaviourally equivalent, *in symbols $M \equiv M'$, if there is a bisimulation E between M and M'.* Two fuzzy processes $P = (W, R, V, w_0)$ and $P' = (W', R', V', w'_0)$ are behavioural equivalent, *in symbols $P \equiv P'$, if there exists a bisimulation E between (W, R, V) and (W', R', V') such that $w_0\ E\ w'_0$.*

Note that $\equiv$ is an equivalence relation between fuzzy transition systems/fuzzy processes. Moreover, as it is well known, behavioural equivalence over (the same) M witnessed by the greatest bisimulation between M and itself boils down to an equivalence relation over its state space. This relation, denoted by $\sim_M$, is called *bisimilarity* (on M). In the sequel, $\sim_M$ will be used to define quotients on fuzzy transition systems and fuzzy processes.

Theorem 3 ([JMM20]). *Let $M = (W, R, V)$, $M' = (W', R', V')$ be two $(\mathrm{Prop}, \mathrm{Act})$-fuzzy transition systems,*

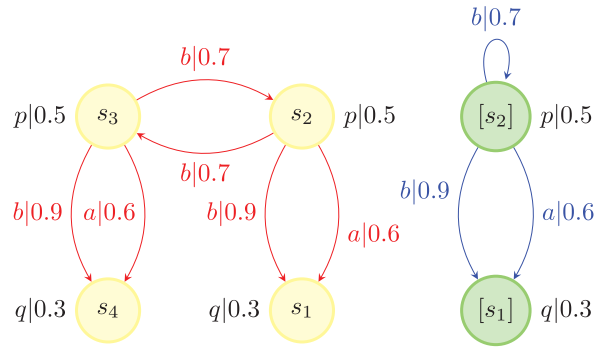
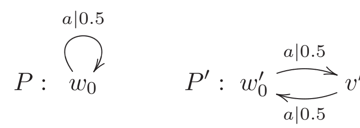


Fig. 1: Construction of a quotient model.

and $E \subseteq W \times W'$ a bisimulation. Then, for any formula $\phi \in \mathsf{Sen}^{\mathcal{FML}}(\mathrm{Prop}, \mathrm{Act})$ and for any two states $w \in W$, $w' \in W'$, such that $w\ E\ w'$, $(M, w \models \phi) = (M', w' \models \phi)$.

This result of *modal invariance* holds for $\mathcal{FML}$ models, but it fails for $\mathcal{PFML}$, as shown below.

Example 1. *Consider the following two fuzzy processes*



Clearly, $P \equiv P'$, because $E = \{(w_0, w'_0), (w_0, v')\}$ is a bisimulation. However, $(P \models \downarrow x.\langle a \rangle x) \neq (P' \models \downarrow x.\langle a \rangle x)$.

Resorting to bisimilarity, on the other hand, one may define quotient fuzzy transition systems or fuzzy processes:

Definition 10. *Let $M = (W, R, V)$ a $(\mathrm{Prop}, \mathrm{Act})$-fuzzy transition system. The* quotient *of M (w.r.t. $\sim_M$) is defined as $M/{\sim_M} = (W/{\sim_M},\ R/{\sim_M},\ V/{\sim_M})$ where*

- $W/{\sim_M} = \{[w]_{\sim_M} : w \in W\}$
- *For any $a \in \mathrm{Act}$, $(R/{\sim_M})_a \colon W/{\sim_M} \times W/{\sim_M} \longrightarrow [0, 1]$ is defined by $(R/{\sim_M})_a\left([u]_{\sim_M}, [v]_{\sim_M}\right) = \; = \mathsf{MAX}\{R_a(u_1, v_1) \mid u_1 \in [u]_{\sim_M}, v_1 \in [v]_{\sim_M}\}$*
- $V/{\sim_M} \colon W/{\sim_M} \times \mathrm{Prop} \longrightarrow [0, 1]$ *is given by* $V/{\sim}([w]_{\sim_M}, p) = V(w, p)$

Additionally, for a given $(\mathrm{Prop}, \mathrm{Act})$-fuzzy process $P = (W, R, V, w_0)$, the quotient of P (w.r.t. $\sim_P$) is the process $P/{\sim_P} = \big(W/{\sim_P},\ R/{\sim_P},\ V/{\sim_P}, [w_0]_{\sim_P}\big)$.

The following example computes the quotient w.r.t. bisimilarity of a fuzzy transition system, thus reducing the cardinality of its state space.

Example 2. *Consider the fuzzy transition system $M = (W, R, V)$ depicted in Fig 1. The quotient w.r.t. $\sim_M$ is represented in the right side of Figure 1, with green-coloured states and blue-coloured transitions. It is easy to see that $s_1 \sim_M s_4$ and $s_3 \sim_M s_2$.*

Theorem 4. *Let $M = (W, R, V)$ be a $(\mathrm{Prop}, \mathrm{Act})$-model. Then, $M \equiv M/{\sim_M}$; and similarly for processes.*

Proof. Consider the set $E = \{(w, [w]) | w \in W\}$. We omit, in this proof, subscript $\sim_M$ to identify bisimilarity equivalence classes. To prove that E is a bisimulation relating M and $M/\sim_M$, we first consider the (Fzig) condition. For every action $a \in \text{Act}$, $R_a(w, u) \leq \mathsf{MAX}\{R_a(w_1, u_1) \mid w_1 \in [w], u_1 \in [u]\} = (R_a)/\sim_M ([w], [u]) = \mathsf{MAX}\{(R_a)/\sim_M ([w], [u']) \mid [u'] \in E[\{u\}]\}$. Similarly, consider the (Fzag) condition. For every $[w] \in W/\sim_M$, choose $z \in [w]$ such that for every $u_1 \in [u]$ and $a \in \text{Act}$, $(R_a)/\sim_M ([w], [u]) = \mathsf{MAX}\{R_a(z, u_1) \mid u_1 \in [u]\}$. For every $a \in \text{Act}$, and $[u] \in W/\sim_M$, $(R_a)/\sim_M([z], [u]) = \mathsf{MAX}\{R_a(z, u) \mid u \in [u]\} \leq \mathsf{MAX}\{R_a(z, u') \mid u' \in E^{-1}([u])\}$. Finally, for every $w \in W$ and $p \in Prop$, $V([w], p) = V(w, p)$.

□

As expected in any reasonable theory of systems, the last result identifies a particular role for quotients as a canonical representation of fuzzy transitions systems and fuzzy processes.

Theorem 5. *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two* $(\text{Prop}, \text{Act})$*-fuzzy transition systems. Hence, if $M \equiv M'$ then $M/\sim_M \;\cong\; M'/\sim_{M'}$; and similarly to fuzzy processes.*

Proof. Let $B \subseteq W \times W'$ be the largest bissimulation between M and M'. Then, $B^\circ \circ B \subseteq \sim_{M'}$, where B° is the converse of B (since bissimulations are reflexive and closed by composition). Now, let us consider the map $f_B : W/\sim_M \to W'/\sim_{M'}$ such that $f_B([w]_{\sim_M}) := \{[w']_{\sim_{M'}} | wBw'\}$. We will prove that f_B is a bijective morphism between $M/\sim_M$ and $M'/\sim_{M'}$. For any $[w']_{\sim_{M'}}, [v']_{\sim_{M'}} \in f_B([w]_{\sim_M})$, $(w, w') \in B$ and $(w, v') \in B$, thus $(v', w') \in B^\circ \circ B \subseteq \sim_{M'}$. Hence $[w']_{\sim_{M'}} = [v']_{\sim_{M'}}$, and therefore f_B is a function. Condition (Fzig) entails f_B is a fuzzy transition systems morphism. Analogously, we can see that $f_B^{-1} = f_{B^\circ}$ is a fuzzy transition systems morphism and that $f_{B^\circ} \circ f_B = id_{W/\sim_M}$ and $f_B \circ f_{B^\circ} = id_{W'/\sim_{M'}}$. Hence $M/\sim_M \;\cong\; M'/\sim_{M'}$.

□

An immediate corollary of this result with respect to fuzzy transition systems states that, for every $w \in W$ and $\varphi \in \mathsf{Sen}^{\mathcal{FML}}(\text{Prop}, \text{Act})$, $(M, w \models \varphi) = (M/\sim_M, [w]_{\sim_M} \models \varphi)$.

IV. On the stepwise development of fuzzy controllers and processes

A. Structured Specification

In formal development of software, specifications play a crucial role. Typically, one starts from a 'set of atomic, or flat specifications, consisting of a signature and a set of sentences in a given logic, and proceed to build new specifications form old through a "pallet" of composition operators. In bivalent logics, the semantics of a specification is given by the class of models that satisfies the set of sentences making up the specification. Conceptually, all of these models are potential implementations of the intended system. In a many-valued (fuzzy) setting, the entire development process has to be adapted and the components redefined to cater for fuzziness in system's descriptions. Thus,

Definition 11. *A fuzzy specification in $\mathcal{FML}$ is a pair $SP = (Sig(SP), Mod(SP))$ where $Sig(SP) \in |\mathsf{Sign}|$ and $Mod(SP)$ is a mapping* $Mod(SP) : \mathsf{Mod}(Sig(SP)) \to [0, 1]$

Specifications are built in a structured way as follows:

Flat Specifications $SP = \big((\text{Prop}, \text{Act}), \Phi\big)$ with $\Phi \subseteq \mathsf{Sen}(\text{Prop}, \text{Act})$. Thus,

- $Sig(SP) = (\text{Prop}, \text{Act})$
- $Mod(SP)(M) = \mathsf{MIN}_{\varphi \in \Phi}\big(M \models \varphi\big)$, i.e., in $\mathcal{FML}$, $Mod(SP)(M) = \mathsf{MIN}_{\varphi \in \Phi, w \in W}\big(M, w \models \varphi\big)$ and, in $\mathcal{PFML}$, $Mod(SP)(M) = \mathsf{MIN}_{\varphi \in \Phi}\big(M, w_0 \models \varphi\big)$.

Union $SP \cup SP'$, with SP and SP' specifications over the same signature. Thus,

- $Sig(SP \cup SP') = Sig(SP)$
- $Mod(SP \cup SP')(M) =$
  $= \min\big(Mod(SP)(M), Mod(SP')(M)\big)$

Translation SP **with** σ, where

$$\sigma : Sig(SP) \to (\text{Prop}', \text{Act}')$$

is a signature morphism. Thus,

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

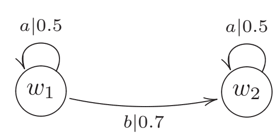
Hiding $Sig(SP$ **hide via** $\sigma)$, where

$$\sigma : Sig(SP) \to (\text{Prop}', \text{Act}')$$

is a signature morphism. Thus,

- $Sig(SP \textbf{ hide via } \sigma) = (\text{Prop}, \text{Act})$
- $Mod(SP \textbf{ hide via } \sigma)(M) =$
  $\mathsf{MAX}_{N \in M^\sigma} Mod(SP)(N)$,
  where M^σ stands for the class of all σ-expansions of M, i.e. $M^\sigma = \{N \in Mod(SP) | N|_\sigma = M\}$.

Example 3. *Consider the following $\mathcal{FML}$-specification $SP = SP_1 \cup SP_2$ where $SP_1 = (\Sigma, \{p \to [b]\bot\})$, $SP_2 = (\Sigma, \{q \to \langle a \rangle \top\})$, $\Sigma = (\{p, q\}, \{a\})$, and a model M depicted as*



with $V(w_1, q) = 1$, $V(w_2, p) = 0.5$ *and* $V(w_1, p) = V(w_2, q) = 0$. *Then,*

$$\begin{aligned}
&Mod(SP)(M) = \\
&= \min\{Mod(SP_1)(M), Mod(SP_2)(M)\} \\
&= \min\{\mathsf{MIN}_{w \in W}(M, w \models p \to [b]\bot), \\
&\qquad \mathsf{MIN}_{w \in W}(M, w \models q \to \langle a \rangle \top))\} \\
&= \min\{\mathsf{MIN}_{w \in W} I\big((M, w \models p), (M, w \models [b]\bot)\big), \\
&\qquad \mathsf{MIN}_{w \in W} I\big((M, w \models q), (M, w \models \langle a \rangle \top)\big)\} \\
&= \min\{ \\
&\min\{
\end{aligned}$$

$I\big(V(w_2,p),(M,w_2 \models [b]\bot)\big), I\big(V(w_1,p),(M,w_1 \models [b]\bot)\big)\},$
$\min\{$
$I\big(V(w_1,q),(M,w_1 \models \langle a\rangle\top)\big), I\big(V(w_2,q),(M,w_2 \models \langle a\rangle\top)\big)\}\}$
$= \min\{\min\{I\big(0.5,1\big), I\big(0,0\big)\}, \min\{I\big(1,0.5\big)\big), I\big(0,0.5\big)\}\}$
$= \min\{1, 0.5\} = 0.5$
$= 0.5$

B. Stepwise Implementation process

Finally, let us revisit the *implementation* processes, in sense of [ST12], of fuzzy systems and processes.

Definition 12 (Implementation)**.** *Let SP and SP' be two specifications. We say that SP' implements SP, in symbols $SP \rightsquigarrow SP'$, if $Sig(SP) = Sig(SP')$ and, for any $M \in \mathsf{Mod}(Sig(SP))$, $Mod(SP)(M) \leq Mod(SP')(M)$. The value $Mod(SP)(M)$ is called the implementation degree of M w.r.t. SP. Note that this generalises the standard notion of simple implementation [ST12].*

Theorem 6 (Vertical Composition)**.** *Let SP_1, SP_2, SP_3 be three fuzzy specifications such that $SP_1 \rightsquigarrow SP_2$ and $SP_2 \rightsquigarrow SP_3$. Then $SP_1 \rightsquigarrow SP_3$.*

Proof. Straightforward from the transitivity of $\leq$. □

Theorem 7 (Horizontal Composition)**.** *Let SP_1, SP_2, SP_1', and SP_2' be fuzzy specifications. Then, if $SP_1 \rightsquigarrow SP_1'$ and $SP_2 \rightsquigarrow SP_2'$,*

1) $(SP_1 \cup SP_2) \rightsquigarrow (SP_1' \cup SP_2')$
2) $(SP_1$ ***with*** $\sigma) \rightsquigarrow (SP_1'$ ***with*** $\sigma)$
3) $(SP_1$ ***hide via*** $\sigma) \rightsquigarrow (SP_1'$ ***hide via*** $\sigma)$

Proof. To prove *1)*, observe that

$$\begin{array}{ll} & Mod(SP_1 \cup SP_2)(M) \\ = & \quad \{ \text{ defn } \cup \} \\ & \min\{Mod(SP_1)(M), Mod(SP_2)(M)\} \\ \leq & \quad \{ \ SP_1 \rightsquigarrow SP_1', SP_2 \rightsquigarrow SP_2' \text{ and min monotony}\} \\ & \min\{Mod(SP_1')(M), Mod(SP_2')(M)\} \\ = & \quad \{ \text{ defn } \cup \} \\ & Mod(SP_1' \cup SP_2')(M) \end{array}$$

For *2)*,

$$\begin{array}{ll} & Mod(SP_1 \textbf{ with } \sigma)(M') \\ = & \quad \{ \text{ defn } \textbf{with } \sigma \ \} \\ & Mod(SP_1)(M'|_\sigma) \\ \leq & \quad \{ \ \text{since } SP_1 \rightsquigarrow SP_1' \ \} \\ & Mod(SP_1')(M'|_\sigma) \\ = & \quad \{ \text{ defn } \textbf{with } \sigma \ \} \\ & Mod(SP_1' \textbf{ with } \sigma)(M') \end{array}$$

Finally, for *3)*, observe that

$$\begin{array}{ll} & Mod(SP_1 \textbf{ hide via } \sigma)(M) \\ = & \quad \{ \text{ defn } \textbf{hide via } \sigma \ \} \\ & \mathsf{MAX}_{N\in M^\sigma} Mod(SP_1)(N), \\ & \text{ where } M^\sigma = \{N \in Mod(SP) | N|_\sigma = M\} \\ \leq & \quad \{ \ \text{since } SP_1 \rightsquigarrow SP_1' \ \} \\ & \mathsf{MAX}_{N\in M^\sigma} Mod(SP_1')(N), \\ & \text{ where } M^\sigma = \{N \in Mod(SP) | N|_\sigma = M\} \\ = & \quad \{ \text{ defn } \textbf{with } \sigma \ \} \\ & Mod(SP_1' \textbf{ hide via } \sigma)(M) \end{array}$$

□

C. Abstractors and Behavioural implementations

Often to implement a specification it is enough to consider models where it is just 'behaviourally' satisfied. Thus, let us consider the following specification building operators based on the bisimilarity and behavioural equivalence relations:

The behaviour operator :

- $Sig(\textbf{behaviour } SP) = Sig(SP)$
- $Mod(\textbf{behaviour } SP)(M) = Mod(SP)(M/\sim_M)$

The abstractor operator :

- $Sig(\textbf{abstractor } SP \textbf{ w.r.t } \equiv) = Sig(SP)$
- $Mod(\textbf{abstractor } SP)(M) =$
  $= \mathsf{MAX}_{N\in[M]_\equiv} Mod(SP)(N)$

Example 4. *Consider again, from Example 1, fuzzy processes P and P'. For the $\mathcal{PFML}$-specification*

$$SP = ((\{a\},\{\}),\{\downarrow x.\langle a\rangle x\})$$

observe that $Mod(SP)(P') \neq Mod(SP)(P)$. On the other hand,

$$\begin{array}{rcl} Mod(\textbf{\textit{behaviour}} \ SP)(P') & = & Mod(SP)(P'/\sim_{P'}) \\ & = & Mod(SP)(P) \end{array}$$

which means that P' behaviourally implements SP with the same degree that P implements SP.

Example 5. *Let us consider the $\mathcal{PFML}$-specification*

$$SP = \big((\textbf{abstract } SP_0 \textbf{ w.r.t. } \equiv) \textbf{ with } \sigma\big)$$

for $SP_0 = \Big((\{a,b\},\{\}),\{\downarrow x.(\langle a\rangle x \wedge \langle b\rangle(\downarrow y.\langle a\rangle y))\}\Big)$ and the inclusion morphism $\sigma : (\{a,b\},\{\}) \to (\{a,b,c\},\{p,q\})$. The implementation degree of the process P in Figure 2 w.r.t SP is computed as

$$\begin{array}{l} Mod\big(\big((\textbf{abstract } SP_0 \textbf{ w.r.t. } \equiv) \textbf{ with } \sigma\big)\big)(P) \ = \\ = \ Mod(\textbf{abstract } SP_0 \textbf{ w.r.t. } \equiv)(P|_\sigma) \\ = \ \mathsf{MAX}_{P'\in[P|_\sigma]_\equiv} Mod(SP)(P|_\sigma) \end{array}$$

In particular, since $P|_\sigma \equiv Q$, we have

$$\begin{array}{l} Mod\big(\big((\textbf{abstract } SP_0 \textbf{ w.r.t. } \equiv) \textbf{ with } \sigma\big)\big)(P) \\ \quad \geq Mod(SP)(Q) \end{array}$$

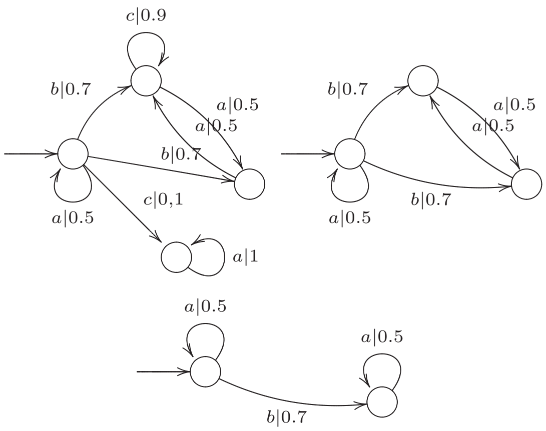


Fig. 2: Processes P, $P|_\sigma$ and Q.

Moreover,

$$
\begin{aligned}
Mod(SP)(Q) =& (Q \models^{\mathcal{PFML}} \downarrow x.(\langle a\rangle x \wedge \langle b\rangle(\downarrow y.\langle a\rangle y))\\
=& (Q, g[x \mapsto w_0], w_0 \models (\langle a\rangle x \wedge \langle b\rangle(\downarrow y.\langle a\rangle y)\\
=& \min\big\{(Q, g[x \mapsto w_0], w_0 \models (\langle a\rangle x),\\
& (Q, g[x \mapsto w_0], w_0 \models \langle b\rangle(\downarrow y.\langle a\rangle y)\big\}\\
=& \min\{\mathsf{MAX}_{w\in W} \min\{R_a(w_0, w),\\
& (Q, g[x \mapsto w_0], w \models x),\\
& \mathsf{MAX}_{w\in W} \min\big\{R_b(w_0, w),\\
& (Q, g[x \mapsto w_0], w \models \downarrow y.\langle a\rangle y)\}\big\}\\
=& \min\big\{\cdots, \mathsf{MAX}_{w'\in W} \min\{R_b(w_0, w),\\
& (Q, g[x \mapsto w_0, y \mapsto x], w \models \langle a\rangle y)\}\big\}\\
=& \min\Big\{0.5, \min\{0.7, 0.5\}\Big\}\\
=& 0.5
\end{aligned}
$$

Hence, we conclude that $Mod(SP)(P) \leq 0.5$.

These operators have no effect in flat specifications:

Lemma 1. *For a flat specification (Σ, Φ),*

$$
\begin{aligned}
& Mod(\textbf{behaviour }(\Sigma, \Phi))\\
&\quad = Mod(\Sigma, \Phi)\\
&\quad = Mod(\textbf{abstractor }(\Sigma, \Phi)\ \textbf{w.r.t.} \equiv)
\end{aligned}
$$

Proof. For any $M \in Mod(\Sigma, \Phi)$,

$$
\begin{aligned}
& Mod(\textbf{behaviour }(\Sigma, \Phi))(M)\\
= & \quad \{\text{ semantics of } \textbf{behaviour}\}\\
& Mod((\Sigma, \Phi))(M/\!\sim_M)\\
= & \quad \{\text{ semantics of } \textbf{flat}, \text{ Theorem 3}\}\\
& Mod((\Sigma, \Phi))(M)\\
= & \quad \{\text{ Theorem 3, monotonicity of } \max\}\\
& \mathsf{MAX}_{N\in[M]_\equiv} Mod((\Sigma, \Phi))(N)\\
= & \quad \{\text{ semantics of } \textbf{abstractor}\}\\
& Mod((\Sigma, \Phi))(M)
\end{aligned}
$$

□

However, this is not the case for specifications in $\mathcal{PFML}$, since model invariance does not hold for this logic. This observation stresses the relevance of the behavioural specification operators presented before. Indeed, differently from to $\mathcal{FML}$, in $\mathcal{PFML}$these operators do not collapse. As a consequence, it makes sense to consider more relaxed notions of implementation:

Definition 13 (Behavioural and abstractor Implementation)**.** *Let SP and SP two specifications. We say that:*

- *SP' behavioural implements SP, syntactically $SP \rightsquigarrow_{Bh} SP'$, if $SP \rightsquigarrow (\textbf{behaviour } SP')$.*
- *SP' is an abstractor implementation of SP, syntactically $SP \rightsquigarrow_{Abs} SP'$, if $(\textbf{abstractor } SP\ \textbf{w.r.t.} \equiv) \rightsquigarrow SP'$.*

Theorem 8. *Behavioural and abstractor implementations compose vertically, i.e.*

1) If $SP_0 \rightsquigarrow_{Abs} SP_1 \rightsquigarrow_{Abs} SP2$ then $SP_0 \rightsquigarrow_{Abs} SP2$
2) If $SP_0 \rightsquigarrow_{Bh} SP_1 \rightsquigarrow_{Bh} SP2$ then $SP_0 \rightsquigarrow_{Bh} SP2$.

Proof. Firstly, note that, because of Lemma 1, the result follows for specifications in $\mathcal{FML}$ directly from Theorem 6. Let us now prove the result for specifications in $\mathcal{PFML}$.

1) Suppose that $SP_0 \rightsquigarrow_{Abs} SP_1 \rightsquigarrow_{Abs} SP_2$, i.e. $(\textbf{abstractor } SP_0\ \textbf{w.r.t.} \equiv) \rightsquigarrow SP_1$ and $(\textbf{abstractor } SP_1\ \textbf{w.r.t.} \equiv) \rightsquigarrow SP_2$. In order to obtain $(\textbf{abstractor } SP_0\ \textbf{w.r.t.} \equiv) \rightsquigarrow SP_2$, by Theorem 6, it is sufficient to prove that $SP_1 \rightsquigarrow (\textbf{abstractor } SP_1\ \textbf{w.r.t.} \equiv)$, i.e. for any $P \in Mod(Sig(SP_2))$, $Mod(\textbf{abstractor } SP_1\ \textbf{w.r.t.} \equiv)(P) \leq Mod(SP_1)(P)$ and, by abstractor definition, that $\mathsf{MIN}_{Q\in[P]_\equiv} Mod(SP_1)(Q) \leq Mod(SP_1)(P)$. This is true since $P \in [P]_\equiv$.

2) Similarly, supposing $SP_0 \rightsquigarrow_{Bh} SP_1$ and $SP_1 \rightsquigarrow_{Bh} SP2$, i.e. that $SP_0 \rightsquigarrow (\textbf{behaviour } SP_1)$ and $SP_1 \rightsquigarrow (\textbf{behaviour } SP_2)$, we obtain $SP_0 \rightsquigarrow_{Bh} SP2$ by Theorem 6 by showing that $(\textbf{behaviour } SP_1) \rightsquigarrow SP_1$, i.e. that for any $P \in \mathsf{Mod}^{\mathcal{PFML}}(Sig(SP_1)$, we have $Mod(SP)(P) \leq Mod(\textbf{behaviour } SP_1)(P)$, which is equivalent to $Mod(SP)(P) \leq Mod(SP_1)(P/\!\sim_P)$ This proof can be done by induction on the structure of specifications. For flat specifications we have

$$
\begin{aligned}
& Mod\big((\mathrm{Act}, \mathrm{Prop}), \Phi\big)(P) \leq\\
& \qquad Mod\big(\textbf{behaviour } (\mathrm{Act}, \mathrm{Prop}), \Phi\big)(P)\\
\Leftrightarrow & \quad \{\text{ semantics of } \textbf{behaviour}\}\\
& Mod\big((\mathrm{Act}, \mathrm{Prop}), \Phi\big)(P) \leq Mod\big((\mathrm{Act}, \mathrm{Prop}), \Phi\big)(P/\!\sim_P)\\
\Leftrightarrow & \quad \{\text{ semantics of } ((\mathrm{Act}, \mathrm{Prop}), \Phi)\}\\
& \mathsf{MIN}_{\varphi\in\Phi}(P \models \varphi) \leq \mathsf{MIN}_{\varphi\in\Phi}(P/\!\sim_P \models \varphi)
\end{aligned}
$$

Then, proceed through induction on the structure of formulæ: for any $\varphi \in \mathsf{Sen}(\mathrm{Act}, \mathrm{Prop})$, $(P \models \varphi) \leq (P/\!\sim_P \models \varphi)$. For $\varphi = x$, with x a variable, we have, for any valuation $g : X \to W$, that for the case $(P/{\sim_P}, g/{\sim_P}, [w_0] \models x) = 1$, the inequality trivially holds. On the other hand, for $(P/{\sim_P}, g/{\sim_P}, [w_0] \models x) = 0$, we have $(P/{\sim_P}, g/{\sim_P}, [w_0] \models x) = 0 \Rightarrow g/{\sim_P}(x) \neq [w_0] \Rightarrow g/{\sim_P}(x) \neq w_0 \Rightarrow (P, g, w_0 \models x) = 0$. The remaining sentences are proved analogously.

□

V. Conclusions and future work

This paper introduced two fuzzy modal logics, $\mathcal{FML}$ and $\mathcal{PFML}$, to reason about fuzzy transitions systems and fuzzy processes, respectively. This extends previous results from the authors in [JMM20], extended here to the many-modal case, and [MBHM16], [MBHM18], also extended to the fuzzy case.

Both logics were framed as many-valued institutions in order to develop the fundamentals of a theory of (fuzzy) structured specifications, within the algebraic approach documented in D. Sannella and A. Tarlecki's landmark book [ST12].

The approach sketched in this paper can be extended in several directions. We are particularly interested in characterising behavioural specifications and their stepwise refinement through the development of appropriate observational equivalences and metrics, as initiated here, Two distinct paths are currently being explored, namely

- the study of other behavioural equivalences as abstractor relations, for example taking bisimulation as a fuzzy relation itself, as proposed in [Fan15], possibly in a coalgebraic setting [Jac16], understood as the "correct" mathematical way to frame state transition computations. References [BBB$^+$12], [WC14] provide interesting starting points;
- the development of the current specification formalism as a specific *behaviour-abstractor framework*, along the path taken in [HMW18].

The inclusion of binders in our fuzzy process logic makes the later close to a propositional version of a fuzzy descriptive logics (e.g. [Str15], [TM98], [Haj05]). This paves the way to explore the applicability of specification building operators proposed here for the structured definition of ontologies. Indeed the growing interest in fuzzy programming languages for concrete application domains, e.g. medicine [VMA10] and robotics [CAF13], calls for a suitable specification framework, as initiated in this paper.

On the other hand, the intersection of fuzzy and quantum computational approaches, as discussed in e.g. [Man06], [SNL09], will be worth to explore. Actually, while traditional quantum logic [BN36] is handled in classical terms, fuzzy reasoning may emerge as a possible complement to handle uncertainty in quantum measurements. We anticipate interesting challenges in the definition of semantics, specification and implementation of quantum systems with a fuzzy flavour.

References

[Ata20] K. T. Atanassov. *Interval-Valued Inttuitionistic Fuzzy Sets*. Springer, 2020.

[BBB$^+$12] F. Bonchi, M. M. Bonsangue, M. Boreale, J. J. M. M. Rutten, and A. Silva. A coalgebraic perspective on linear weighted automata. *Inf. Comp.*, 211:77–105, 2012.

[BDK17] R. Belohlávek, J. W. Dauben, and G. J. Klir. *Fuzzy Logic and Mathematics: A Historical Perspective*. Oxford U. Press, 2017.

[BN36] G Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[Bra10] T. Brauner. *Hybrid Logic and its Proof-Theory*. Applied Logic Series. Springer, 2010.

[CAF13] P. Cingolani and J. Alcalá-Fdez. jFuzzyLogic: a Java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *Int. J. Comp. Int. Sys.*, 6:61–75, 2013.

[DETRK15] D. E. Dan E. Tamir, N. D. Rishe, and A. Kandel. *Fifty Years of Fuzzy Logic and its Applications*, volume 326 of *Studies in Fuzziness and Soft Comp.* Springer, 2015.

[Dia13] R. Diaconescu. Institutional semantics for many-valued logics. *Fuzzy Sets Syst.*, 218:32–52, 2013.

[DK05] M. Doostfatemeh and S. C. Kremer. New directions in fuzzy automata. *Int. J. of Appr. Reas.*, 38:175–214, 2005.

[Fan15] T. Fan. Fuzzy bisimulation for gödel modal logic. *IEEE Trans. Fuzzy Syst.*, 23(6):2387–2396, 2015.

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

[Haj05] Petr Hajek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154(1):1–15, 2005.

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

[Jac16] B. Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Th. Comp. Sc.* Cambridge Univ. Press, 2016.

[JMM20] M. Jain, A. Madeira, and M. A. Martins. A fuzzy modal logic for fuzzy transition systems. *ENTCS*, 348:85 – 103, 2020.

[Lia01] C. Liau. Hybrid logic for possibilistic reasoning. In *Proc. Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf.*, volume 3, pages 1523–1528, 2001.

[Man06] M. Mannucci. Quantum fuzzy sets: Blending fuzzy set theory and quantum computation. *CoRR*, abs/cs/0604064, 2006.

[MBHM16] A. Madeira, L. S. Barbosa, R. Hennicker, and M. A. Martins. Dynamic logic with binders and its application to the development of reactive systems. In A. Sampaio and F. Wang, editors, *ICTAC 2016 - 13th Int. Colloquium, Taipei, Taiwan, ROC, Proc.*, volume 9965 of *LNCS*, pages 422–440. Springer, 2016.

[MBHM18] A. Madeira, L. S. Barbosa, R. Hennicker, and M. A. Martins. A logic for the stepwise development of reactive systems. *Th. Comp. Sc.*, 744:78–96, 2018.

[SNL09] I. Schmitt, A. Nürnberger, and S. Lehrack. *On the Relation between Fuzzy and Quantum Logic*, volume 243 of *Studies in Fuzziness and Soft Comp.*, pages 417–438. Springer, 2009.

[ST12] D. Sannella and A. Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs on TCS, an EATCS Series. Springer, 2012.

[Str15] Umberto Straccia. All about fuzzy description logics and applications. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2015.

[TM98] C.B. Tresp and R. Molitor. A description logic for vague knowledge. In *Proceedings of the 13th biennial European Conference on Artificial Intelligence (ECAI'98)*, pages 361–365, Brighton, UK, 1998. J. Wiley and Sons.

[VMA10] T. Vetterlein, H. Mandl, and K.-P. Adlassnig. Fuzzy Arden syntax: A fuzzy programming language for medicine. *Art. Int. in Medicine*, 49(1):1–10, 2010.

[WC14] H. Wu and Y Chen. Coalgebras for fuzzy transition systems. *Electronic Notes in Th. Comp. Sc.*, 301:91–101, 2014.

[WD16] H. Wu and Y. Deng. Logical characterizations of simulation and bisimulation for fuzzy transition systems. *Fuzzy Sets Syst.*, 301(C):19–36, 2016.

[Zad65] L. A. Zadeh. Fuzzy sets. *Inf. Control*, 8:338–353, 1965.