Fitting's Style Many-Valued Interval Temporal Logic Tableau System: Theory and Implementation

Guillermo Badia
School of Historical and Philosophical Inquiry, University of Queensland, Brisbane, Australia

Carles Noguera
Department of Information Engineering and Mathematics, University of Siena, Italy

Alberto Paparella
Department of Mathematics and Computer Science, University of Ferrara, Italy

Guido Sciavicco
Department of Mathematics and Computer Science, University of Ferrara, Italy

Ionel Eduard Stan
Faculty of Engineering, Free University of Bozen-Bolzano, Italy

Abstract

Many-valued logics, often referred to as fuzzy logics, are a fundamental tool for reasoning about uncertainty, and are based on truth value algebras that generalize the Boolean one; the same logic can be interpreted on algebras from different varieties, for different purposes and pose different challenges. Although temporal many-valued logics, that is, the many-valued counterpart of popular temporal logics, have received little attention in the literature, the many-valued generalization of Halpern and Shoham's interval temporal logic has been recently introduced and studied, and a sound and complete tableau system for it has been presented for the case in which it is interpreted on some finite Heyting algebra. In this paper, we take a step further in this inquiry by exploring a tableau system for Halpern and Shoham's interval temporal logic interpreted on some finite $\mathrm{FL}_{\mathrm{ew}}$-algebra, therefore generalizing the Heyting case, and by providing its open-source implementation.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains

Keywords and phrases Interval temporal logic, many-valued logic, tableau system

Digital Object Identifier 10.4230/LIPIcs.TIME.2024.7

Funding This research has been partially funded by the Italian Ministry of University and Research through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 del 15/03/2022), Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by the European Union under the "NextGeneration EU programme". Moreover, this research has also been founded by the FIRD project *Modal Geometric Symbolic Learning* (University of Ferrara), and the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (INDAM-GNCS) project *Symbolic and Numerical Analysis of Cyberphysical Systems*, CUP code E53C22001930001. Guido Sciavicco and Ionel Eduard Stan are GNCS-INdAM members; Carles Noguera is a GNSAGA-INdAM member.

1 Introduction

Many-valued logics, also called *fuzzy logics*, extend beyond the binary truth values of classical logic by allowing formula truth to be graded. These logics are defined over algebraic systems, or *algebras*. Unlike classical logic, which is grounded in the Boolean two-valued algebra, many-valued logics involve a more complex algebraic framework, thereby supporting a richer set of truth values. Such generalizations have led to the development of a sophisticated algebraic taxonomy, accommodating different types of underlying domains and the properties of algebraic operators, which in turn influence the interpretation of logical connectives [10]. Noteworthy examples of varieties of algebras on which many-valued logics have been defined include *Gödel* algebras (G) [2], MV-algebras [8] on which *Łukasiewicz* logic is based [31], *product* algebras (Π) [22], and *Heyting* algebras (H) on which *intuitionistic* logic is based [15].

In practical applications, the supporting algebra is often assumed to be linearly ordered, that is, a *chain*. Prominent examples falling under this assumption are algebras defined on the interval $[0, 1]$ in $\mathbb{R}$; in this case, an algebra is termed *standard*. When an algebra only has a finite number of elements, instead, is known as a *finite* algebra; finite algebras may or may not be chains. Choosing among these types depends on the intended applications and the availability of reasoning tools.

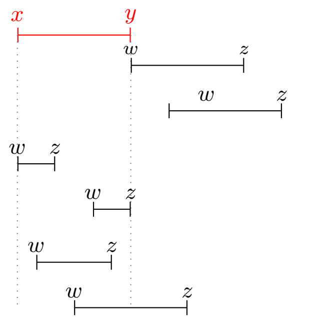
Many-valued logics are crucial in several mathematics and computer science areas, particularly within artificial intelligence and symbolic machine learning. These logics are predominantly employed in rule-based classifier learning [24], enhancing the flexibility and expressive power of such systems. Less frequently, many-valued logics are used to refine decision-tree classifiers by supporting more granular decision-making processes [9]. In the context of subsymbolic machine learning, many-valued logics find their use in fuzzy neural networks, which aid in managing the inherent uncertainty in data, thereby improving network adaptability and performance [7].

In terms of temporal logics, only a few attempts have been made to study point-based temporal languages in the many-valued case. These include early contributions, such as [14, 26, 33], which are characterized by the fact that, in the proposed languages, only the propositional side of formulas is fuzzified, and more recent proposals, such as [18], in which the authors provide a generalization of LTL that allows one to express uncertainty in both atomic propositions and temporal relations. In all such cases, however, standard underlying algebras are assumed.

While many-valued logics based on standard algebras are relatively common in practical applications, the practical necessity for an infinite set of truth values is often debatable; for example, in the context of machine learning, it is immediate to observe that datasets contain only a finite number of distinct values, and therefore give rise to a finite number of distinct situations. At the same time, the conventional reliance on chain algebras can sometimes restrict modeling capabilities, for example disallowing the possibility of reasoning about different, and incomparable, experts' viewpoints. A more general approach to many-valued modal (and therefore also temporal) logics is that of Fitting [16]. Fitting's formalization is applicable to any modal (temporal, spatial, and so on) logic and any algebra. In particular, in the case of interval temporal logic, this has been done in [11, 12, 13], where a many-valued extension of Halpern and Shoham's interval temporal logic (HS [23]) over a Heyting algebra (the resulting logic was denoted FHS, that is, *fuzzy* HS) has been introduced and studied, along with a tableau system for it in the case of finite algebras.

In this paper, we explore a further generalization of HS based on $\mathrm{FL}_{\mathrm{ew}}$-algebras [10]. An $\mathrm{FL}_{\mathrm{ew}}$-algebra is defined over a bounded commutative integral residuated lattice and naturally generalizes several common frameworks, including G, MV, Π, and H. In order to uniform the terminology, we shall use the term *Many-Valued Halpern and Shoham's interval temporal logic*, that is, MVHS. We extend Fitting's tableau system to deal with MVHS over some finite $\mathrm{FL}_{\mathrm{ew}}$-algebra, we provide a working implementation for it as a part of our open-source reasoning and learning framework, and we test it to assess its practical usefulness. It is important to notice that extending the tableau rules from H to $\mathrm{FL}_{\mathrm{ew}}$ does not require much work; on the other hand, implementing the resulting system is not trivial.

Table 1 Allen's interval relations.

relation	definition	example
after	$[x, y]R_A[w, z] \Leftrightarrow y = w$	
later	$[x, y]R_L[w, z] \Leftrightarrow y < w$	
begins	$[x, y]R_B[w, z] \Leftrightarrow x = w \wedge z < y$	
ends	$[x, y]R_E[w, z] \Leftrightarrow y = z \wedge x < w$	
during	$[x, y]R_D[w, z] \Leftrightarrow x < w \wedge z < y$	
overlaps	$[x, y]R_O[w, z] \Leftrightarrow x < w < y < z$	

This paper is organized as follows. In Section 2 we recall some basic notions of both the crisp version of the interval temporal logic HS and the most common algebras that have some role in the literature of many-valued logic. In Section 3 we present MVHS, its syntax and semantics, several application examples, and our adaptation of Fitting's tableau system for it. Finally, in Section 4 we present our implementation and the results of several systematic tests, before concluding.

2 Preliminaries

Halpern and Shoham's Interval Temporal Logic. Several different interval temporal logics have been proposed in the recent literature [21], mostly in the point-based setting. In the interval-based setting, however, *Halpern and Shoham's Modal Logic for Time Intervals* (HS) [23] can be considered the formalism that has received the most attention.

Let $\mathbb{D} = \langle D, < \rangle$ be a (strict) linear order with *domain* D; in the following, we shall use D and $\mathbb{D}$ interchangeably. A *strict interval* over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in \mathbb{D}$ and $x < y$. If we exclude the identity relation, there are 12 different binary ordering relations between two strict intervals on a linear order, often called *Allen's interval relations* [1]: the six relations R_A (*adjacent to*), R_L (*later than*), R_B (*begins*), R_E (*ends*), R_D (*during*), and R_O (*overlaps*), depicted in Tab. 1, and their *inverses*, that is, $R_{\overline{X}} = (R_X)^{-1}$, for each $X \in \{A, L, B, E, D, O\}$. We interpret interval structures as Kripke structures, with Allen's relations playing the role of accessibility relations. Thus, we associate an *existential modality* $\langle X \rangle$ with each one of Allen's relations R_X. Moreover, for each $X \in \{A, L, B, E, D, O\}$, the *transpose* of modality $\langle X \rangle$ is the modality $\langle \overline{X} \rangle$ corresponding to the inverse relation $R_{\overline{X}}$ of R_X. Now, let $\mathcal{X} = \{A, \overline{A}, L, \overline{L}, B, \overline{B}, E, \overline{E}, D, \overline{D}, O, \overline{O}\}$; well-formed HS formulas are built from a set of *propositional letters* $\mathcal{P}$, the classical connectives $\vee$ and $\neg$, and a modality for each Allen's interval relation, as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi,$$

where $p \in \mathcal{P}$ and $X \in \mathcal{X}$. The other propositional connectives and constants (i.e., $\psi_1 \wedge \psi_2 \equiv \neg\psi_1 \vee \neg\psi_2$, $\psi_1 \to \psi_2 \equiv \neg\psi_1 \vee \psi_2$ and $\top = p \vee \neg p$), as well as, for each $X \in \mathcal{X}$, the *universal modality* $[X]$ (e.g., $[A]\varphi \equiv \neg\langle A \rangle \neg\varphi$) can be derived in the standard way. The set of all subformulas of a given HS formula φ is denoted by $sub(\varphi)$.

The strict semantics of HS is given in terms of *interval models* of the type $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $\mathbb{D}$ is a linear order (in this context, an *interval frame*), $\mathbb{I}(\mathbb{D})$ is the set of all strict intervals over $\mathbb{D}$, and V is a *valuation function* $V : \mathcal{P} \to 2^{\mathbb{I}(\mathbb{D})}$ which assigns to every atomic proposition $p \in \mathcal{P}$ the set of intervals $V(p)$ on which p holds. The truth of a formula φ on a given interval $[x, y]$ in an interval model M, denoted by $M, [x, y] \Vdash \varphi$, is defined by structural induction on the complexity of formulas, as follows:

$$
\begin{array}{lll}
M, [x, y] \Vdash p & \text{if and only if} & [x, y] \in V(p), \text{ for each } p \in \mathcal{AP}, \\
M, [x, y] \Vdash \neg\psi & \text{if and only if} & M, [x, y] \not\Vdash \psi, \\
M, [x, y] \Vdash \psi_1 \vee \psi_2 & \text{if and only if} & M, [x, y] \Vdash \psi_1 \text{ or } M, [x, y] \Vdash \psi_2, \\
M, [x, y] \Vdash \langle X \rangle \psi & \text{if and only if} & \text{there is } [w, z] \text{ s.t. } [x, y] R_X [w, z] \text{ and } M, [w, z] \Vdash \psi,
\end{array}
$$

where $X \in \mathcal{X}$.

Given a model $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$ and a formula φ, we say that M *satisfies* φ if there exists an interval $[x, y] \in \mathbb{I}(\mathbb{D})$ such that $M, [x, y] \Vdash \varphi$. A formula φ is *satisfiable* if there exists an interval model that satisfies it. Moreover, a formula φ is *valid* if it is satisfiable at every interval of every (interval) model or, equivalently, if its negation $\neg\varphi$ is *unsatisfiable*.

$\mathbf{FL_{ew}}$-algebras. An algebraic structure

$$\langle A, \cap, \cup, \cdot, +, \hookrightarrow, 0, 1 \rangle,$$

where we define a binary relation $\preceq$ as $a \preceq b$ if and only if $a \cap b = a$, is called an $\mathrm{FL_{ew}}$-*algebra* if

- $\langle A, \cap, \cup, 0, 1 \rangle$ is a bounded lattice with upper bound 1 and lower bound 0 (and hence $\preceq$ is a partial order)
- $\langle A, \cdot, 1 \rangle$ and $\langle A, +, 0 \rangle$ are commutative monoids
- $\cdot$ and $+$ are monotone w.r.t. $\preceq$, i.e., if $\gamma \preceq \alpha$, $\delta \preceq \beta$, then $\gamma \cdot \delta \preceq \alpha \cdot \beta$, $\gamma + \delta \preceq \alpha + \beta$
- for each $a, b, c \in A$, $a \cdot b \preceq c$ if and only if $a \preceq b \hookrightarrow c$

We refer, as it is customary, to $\cap$ as *meet*, $\cup$ as *join*, $\hookrightarrow$ as *implication*, $\cdot$ as *t-norm*, and $+$ as *t-co-norm*. If the lattice order is complete (that is, each subset has infimum and supremum), we call the structure a *complete* $\mathrm{FL_{ew}}$-*algebra*. In this case, given a subset $A' \subseteq A$, we denote the infimum and the supremum of A' respectively as $\bigwedge A$ and $\bigvee A$. An $\mathrm{FL_{ew}}$-algebra is termed *linearly ordered* (or *chain*) if its lattice order is total, *standard* if its lattice reduct is the real unit interval $[0, 1]$, and *finite* if its lattice comprises only a finite number of elements. A many-valued logic, which generalizes Boolean logic, may derive its truth values from an $\mathrm{FL_{ew}}$-algebra, interpreting logical conjunctions, disjunctions, and implications as the t-norm, t-co-norm, and implication operations of the algebra, respectively.

In the context of many-valued propositional logics, the most common and well-known examples include *Gödel* algebra (G) [2], *MV* algebra (MV) [31], and *product* (Π) algebra [22].

In their typical formulation, they are based on the interval $[0, 1]$ in $\mathbb{R}$, and are, therefore, *chains*. However, they can all be considered special cases of $\mathrm{FL_{ew}}$-chains. In particular, Gödel algebra, interprets conjunction as the minimum; so, for example, if *high fever* has truth value $1/3$ and *pain* has truth value $2/3$, then the sentence *high fever and pain* evaluates to $1/3$, illustrating Gödel's conservative nature: if we know something about two facts, we know the minimum of them about their conjunction. In the case of MV, the conjunction $\alpha \cdot \beta$ is computed as $\max\{0, \alpha + \beta - 1\}$, implying that, in the same scenario as before, the sentence would evaluate to 0, which highlights MV pessimistic approach: if we do not know enough about two facts, we do not know anything about their conjunction. Conversely, conjunction

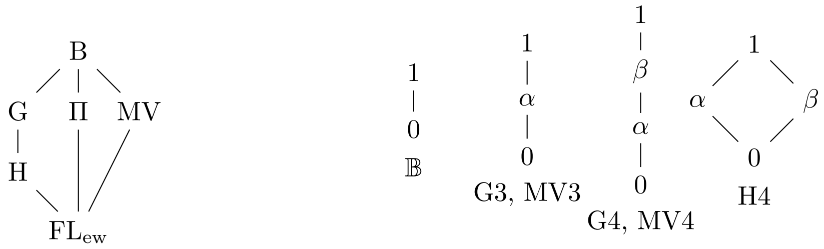


Figure 1 *On the left*, a taxonomy of well-known many-valued algebras. *On the right*, some examples of algebra lattices that will be used in our experiments. Note how G3 and MV3 (resp. G4 and MV4) differ because of the t-norm, but share the same lattice structure.

is the real product in product algebra, evaluating the same sentence as 2/9, thus reflecting a probabilistic interpretation of independent events. Similarly, a Heyting algebra (H) [15, 17] is an $\mathrm{FL_{ew}}$-algebra in which $\cdot$ is defined as $\cap$ and $+$ as $\cup$, and the underlying lattice is not necessarily a chain. For instance, if *high fever* is evaluated as α by some expert and *pain* as β some other expert, and α and β are not necessarily comparable to each other, then the sentence *high fever and pain* is evaluated as γ, where γ is the maximum value less than, and comparable with, both α and β. $\mathrm{FL_{ew}}$-algebras thus generalize all four prominent cases.

Fig. 1 (left-hand side) provides a partial taxonomy of algebra varieties corresponding to various many-valued logics, with the variety corresponding to classical logic (the one generated by the two-valued Boolean algebra) denoted by B. In this diagram, a higher element is less general than a lower one. So, Boolean algebras are generalized by algebras in the Gödel variety, the product variety, and the MV variety, whereas the last three are incomparable to each other. Similarly, a Heyting algebra is a generalization of a Gödel algebra (precisely, the class of Gödel algebras is the variety generated by linearly ordered Heyting algebras), and, finally, the variety of $\mathrm{FL_{ew}}$-algebras is more general than all of the above. In Fig. 1 (right-hand side), we give examples of specific algebra lattices from these varieties also used later in our experiments. For example, G3 is the Gödel algebra with 3 elements; it is linearly ordered, and meet/join are defined following the general specification for the Gödel variety. As we shall see, we can define a many-valued version of interval temporal logic by *plugging in* any specific algebra within $\mathrm{FL_{ew}}$; similarly, our tableau system works in any such case, for as long as the underlying lattice is finite.

3 Many-Valued Interval Temporal Logic

Many-valued linear orderings. In [11, 12, 13], a many-valued generalization of HS based on Heyting algebras has been proposed and studied. As we shall see, its further generalization to the case of $\mathrm{FL_{ew}}$-algebras is syntactically and semantically quite similar; however, decoupling conjunction (resp., disjunction) and t-norm (resp., t-co-norm) increases the flexibility and adaptability of the language to practical situations.

We start off by defining the many-valued generalization of a linearly ordered set.

▶ **Definition 1.** *Let* $\boldsymbol{A} = \langle A, \cap, \cup, \cdot, +, 0, 1\rangle$ *a complete* $\mathrm{FL_{ew}}$*-algebra. A* many-valued linear order *is a structure of the type*

$$\widetilde{\mathbb{D}} = \langle D, \widetilde{<}, \widetilde{=}\rangle,$$

where D *is a* domain *(again, we identify* D *with* $\widetilde{\mathbb{D}}$*) enriched with two functions* $\widetilde{<}, \widetilde{=} \colon D \times D \to A$*, for which the following conditions apply for every* x, y*, and* z*:*

$\widetilde{=}(x, y) = 1$ *iff* $x = y$,
$\widetilde{=}(x, y) = \widetilde{=}(y, x)$,
$\widetilde{<}(x, x) = 0$,
$\widetilde{<}(x, z) \succeq \widetilde{<}(x, y) \cdot \widetilde{<}(y, z)$,
if $\widetilde{<}(x, y) \succ 0$ *and* $\widetilde{<}(y, z) \succ 0$, *then* $\widetilde{<}(x, z) \succ 0$,
if $\widetilde{<}(x, y) = 0$ *and* $\widetilde{<}(y, x) = 0$, *then* $\widetilde{=}(x, y) = 1$,
if $\widetilde{=}(x, y) \succ 0$, *then* $\widetilde{<}(x, y) \prec 1$.

There are several possible definitions of non-crisp linear orders. For example, Zadeh [34], defines a similarity relation in a set, imposing that it is reflexive, symmetric, and transitive, as well as a notion of many-valued ordering, antisymmetry, and totality. Similarly, Bodenhofer [3] advocates for the use of similarity-based many-valued orderings, in which the linearity is in a strong form; the same notion is also used in [25]. Ovchinnikov [30] proposes a notion of many-valued ordering with a non-strict ordering relation. A common denominator to all such proposals is the definition of a very weak version of the transitivity property, which allows one to obtain very general definitions. A many-valued linear order defined as above is similar to Zadeh's, only slightly modified to take into account both the many-valued linear order and the many-valued equality in the same structure. This is motivated by the fact that many-valued Allen's relations involve both equality and linear order.

Given a many-valued linear order $\widetilde{\mathbb{D}} = \langle D, \widetilde{<}, \widetilde{=} \rangle$, we define the *crispification of* $\widetilde{\mathbb{D}}$ to be the crisp linear order $\mathbb{D} = \langle D, < \rangle$, where $x < y$ if and only if $\widetilde{<}(x, y) \neq 0$. It is easy to verify that $\mathbb{D} = \langle D, < \rangle$, so defined, is, in fact, a linear order.

Later we shall refer to the set of all possible many-valued linear orders as $\mathfrak{D}$.

Many-valued Halpern and Shoham's Interval Temporal Logic. As in the crisp case, formulas of the many-valued version of Halpern and Shoham's interval temporal logic are based on a set of propositional letters.

▶ **Definition 2.** *Let $\mathcal{P}$ be a set of propositional letters, and let $\boldsymbol{A}$ be a complete* $\mathrm{FL_{ew}}$*-algebra. Then, a well-formed* many-valued Halpern and Shoham's interval temporal logic formula *(or* MVHS-formula*, for short) is obtained by the following grammar:*

$$\varphi ::= \alpha \mid p \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \to \psi \mid \langle X \rangle \varphi \mid [X]\varphi,$$

where $\alpha \in A$, $p \in \mathcal{P}$, and, $X \in \mathcal{X}$.

In the following, we shall use $\neg\varphi$ to denote the formula $\varphi \to 0$, and $\widetilde{\mathcal{L}}_{MVHS}$ to denote the smallest set that contains all formulas generated by the previous definition.

As for the semantics of MVHS-formulas, given a many-valued linearly ordered set we define the set of fuzzy strict intervals in $\widetilde{\mathbb{D}}$ as

$$\mathbb{I}(\widetilde{\mathbb{D}}) = \{[x, y] \mid \widetilde{<}(x, y) \succ 0\},$$

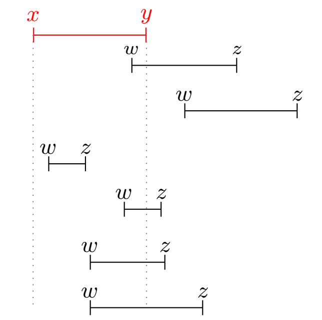
and, generalizing classical Boolean evaluation, propositional letters are directly evaluated in the underlying algebra $\boldsymbol{A}$ by defining a *many-valued valuation function*, as follows:

$$\widetilde{V} : \mathcal{P} \times \mathbb{I}(\widetilde{\mathbb{D}}) \to \boldsymbol{A}.$$

Moreover, given a fixed many-valued linear order we also need to define the many-valued generalization of Allen's relations between intervals (*many-valued Allen's relations*), which is obtained by substituting every $=$ with $\widetilde{=}$ and every $<$ with $\widetilde{<}$ in the original, crisp definition, as shown in Tab. 2.

Table 2 Many-valued version of Allen's interval relations.

relation	definition	example
after	$\widetilde{R}_A([x,y],[w,z]) \quad = \quad \widetilde{=}(y,w)$	
later	$\widetilde{R}_L([x,y],[w,z]) \quad = \quad \widetilde{<}(y,w)$	
begins	$\widetilde{R}_B([x,y],[w,z]) \quad = \quad \widetilde{=}(x,w) \cdot \widetilde{<}(z,y)$	
ends	$\widetilde{R}_E([x,y],[w,z]) \quad = \quad \widetilde{<}(x,w) \cdot \widetilde{=}(y,z)$	
during	$\widetilde{R}_D([x,y],[w,z]) \quad = \quad \widetilde{<}(x,w) \cdot \widetilde{<}(z,y)$	
overlaps	$\widetilde{R}_O([x,y],[w,z]) \quad = \quad \widetilde{<}(x,w) \cdot \widetilde{<}(w,y) \cdot \widetilde{<}(y,z)$	



Many-valued Allen's relations should be interpreted with caution. A many-valued linear order cannot be graphically represented as a crisp linear order, but it is abstractly defined by explicitly listing the value of each relation for each pair of points. Similarly, for example, one interval $[w,z]$ that is *after* an interval $[x,y]$ cannot be simply depicted "after" $[x,y]$ itself. However, to help the intuition, many-valued Allen's relations can be used to interpret a crisp linear order; in this case, it would be natural to consider, for instance, as "equal" two points that are in fact "close" to each other, leading to the possibility of picturing Allen's relations similarly to the crisp case.

▶ **Definition 3.** *Let $\mathcal{P}$ be a set of propositional letters, and let $\boldsymbol{A}$ be a complete* $\mathrm{FL}_{\mathrm{ew}}$-*algebra. Then, a* many-valued Halpern and Shoham's interval temporal logic interpretation *(*MVHS-interpretation*, for short) is a tuple of the type:*

$$\widetilde{M} = \langle \mathbb{I}(\widetilde{\mathbb{D}}), \widetilde{V} \rangle$$

where $\widetilde{\mathbb{D}}$ is a many-valued linear order and $\widetilde{V}$ is a fuzzy valuation function. Given an MVHS-formula φ and an MVHS- interpretation, the valuation of φ *on $\widetilde{V}$ and $[x,y] \in \mathbb{I}(\widetilde{D})$ is computed as follows:*

$$\begin{array}{lcl}
\widetilde{V}(\alpha,[x,y]) & = & \alpha, \\
\widetilde{V}(\varphi \wedge \psi,[x,y]) & = & \widetilde{V}(\varphi,[x,y]) \cdot \widetilde{V}(\psi,[x,y]), \\
\widetilde{V}(\varphi \vee \psi,[x,y]) & = & \widetilde{V}(\varphi,[x,y]) + \widetilde{V}(\psi,[x,y]), \\
\widetilde{V}(\varphi \rightarrow \psi,[x,y]) & = & \widetilde{V}(\varphi,[x,y]) \hookrightarrow \widetilde{V}(\psi,[x,y]), \\
\widetilde{V}(\langle X \rangle \varphi,[x,y]) & = & \bigvee\{\widetilde{R}_X([x,y],[w,z]) \cdot \widetilde{V}(\varphi,[w,z])\}, \\
\widetilde{V}([X]\varphi,[x,y]) & = & \bigwedge\{\widetilde{R}_X([x,y],[w,z]) \hookrightarrow \widetilde{V}(\varphi,[w,z])\},
\end{array}$$

*where $X \in \mathcal{X}$ and where $[w,z]$ varies in $\mathbb{I}(\widetilde{\mathbb{D}})$. We say that a formula of FHS φ is α-*satisfied *at an interval $[x,y]$ in a fuzzy interval model $\widetilde{M}$ if and only if*

$$\widetilde{V}(\varphi,[x,y]) \succeq \alpha.$$

*Moreover, a formula φ is α-*satisfiable *if and only if there exists a fuzzy interval model and an interval in that model in which it is α-satisfied, and it is* satisfiable *if it is α-satisfiable for some $\alpha \in \mathcal{H}$, $\alpha \neq 0$; similarly, a formula is α-*valid *if it is α-satisfied at every interval in every model, and* valid *if it is 1-valid.*

Observe that since an $\mathrm{FL}_{\mathrm{ew}}$-algebra, in general, does not encompass classical negation, and since our definition of satisfiability is graded, instead of absolute, then the usual duality of satisfiability and validity does not hold anymore.

Applications. Interval temporal logic is designed to describe situations in which events have a duration, and are therefore not necessarily instantaneous. Even if a pure qualitative language such as the one of HS does not allow the specification of metric duration, the relative temporal position of different events can be expressed.

Temporal datasets are typically provided in the form of sets of multivariate time series. *Time series* are observations interpreted over a linear order; in some data science contexts, they are also called *single dimension* data. Observations can be *univariate* if there is only one measurement, or *multivariate* if there is more than one; observed data type can be numerical or categorical. Temporal datasets are common in several research areas, spanning from medicine to industry, among many others. In some cases, information can be extracted from a multivariate time series in the form of point-based temporal logic formulas, in which propositional letters symbolically represent instantaneous values (e.g., *the fever is over 37.5 degrees*, or *the vibration sensor marks below 3 mm/s*). More commonly, however, it is often the case that relevant events are better described as intervals so that a propositional letter represents some properties of some function of some variable (e.g., *the average fever is over 37.5 degrees*). Such functions are generally referred to as *feature extraction functions.* In this sense, interesting properties of a time series, or, better, of a temporal dataset of time series, could be expressed in propositional interval temporal logic.

One relevant example of how MVHS can be used to describe interesting situations is that of multivariate time series in psychology and psychiatry. A single patient can be described by several independent variables that help physicians identify his/her psychological status across some observation time. Each variable may be specific to some behavior treat, so that several symptoms can be identified. In the case of clinical depression, for example, such symptoms range from *depressed mood* to *diminished interest*, to *insomnia/hypersomnia*, among many others. Each one of them, clearly, can be associated with one or more periods of time during the observation. In major depression, in particular, the role of *overlapping* symptoms has been debated [4]. The presence of two such overlapping symptoms in a patient can be expressed, for example, as the HS formula

$$\langle G\rangle(\text{depressed mood} \wedge \langle O\rangle\text{insomnia}),$$

where $\langle G\rangle$ is the (definable) existential operator, that makes true a formula on some interval in the model. Statements such as the one above can, in fact, be automatically learned by a system such as [6, 32, 28], from suitable datasets of patients, and the problem of distinguishing, for instance, which pairs of overlapping symptoms more often lead to a major depression diagnosis can be solved as a temporal classification problem.

Operating with a crisp logic, however, one may incur in extracting less-than-optimal knowledge, for two reasons. First, single events labeled with symptoms may not necessarily be clearly identified, as is often the case in medicine. Second, minimal temporal variations may cause sharp changes in the labeled temporal relation (e.g., a patient with both symptoms, not overlapping but only by a relatively small amount of time, would not satisfy the above formula). Extracting many-valued logic statements, expressed in MVHS for example, would allow one to avoid both potential problems, as both the propositional letters and the temporal relations are given a degree of truth (in the above example, a patient with both symptoms, not overlapping but only by relatively small amounts of time, would still satisfy the formula, only to a slightly lower degree).

4 Reasoning in Many-Valued Halpern and Shoham's Interval Temporal Logic

A tableau system. In this section, we consider the problem of reasoning with MVHS formulas. Tableau systems have been introduced in [5, 19, 20, 29] for variants, fragments, and generalizations of crisp HS, and in [17] for many-valued modal logics. The case of MVHS interpreted on a finite Heyting algebra has been tackled in [13]. Here, we generalize the rules to cover the case of a finite $\mathrm{FL_{ew}}$-algebra. From a theoretical point of view, the difference between the two systems is limited; therefore, we shall focus on the problems that having a functioning implementation raises, and, then, on the results of a set of systematic tests.

A tableau for an MVHS formula is a directed tree, in which every node is associated with a truth judgment, a pair formula/interval, and a finite many-valued linear order; these elements, altogether, form a decoration.

▶ **Definition 4** (decoration). *Given an* $\mathrm{FL_{ew}}$*-algebra* $\boldsymbol{A}$*, an MVHS formula* φ*, and a finite many-valued linear order* $\widetilde{\mathbb{D}}$*, a* decoration *is an object of the type*

$$Q(\alpha \preceq \varphi, [x,y], \widetilde{\mathbb{D}}), \text{ or } Q(\varphi \preceq \alpha, [x,y], \widetilde{\mathbb{D}}),$$

where $\alpha \in \boldsymbol{A}$ *and* $Q \in \{T, F\}$ *is a* judgment*. The expression* $\alpha \preceq \varphi$ *(*$\varphi \preceq \alpha$*) is an* assertion *on* $[x,y] \in \mathbb{I}(\widetilde{\mathbb{D}})$*. The universe of all possible decorations is denoted by* $\mathcal{D}$*.*

▶ **Definition 5** (tableau). *Given a finite* $\mathrm{FL_{ew}}$*-algebra* $\boldsymbol{A}$ *and an MVHS formula* φ*, a* tableau τ *for* φ *and* $\alpha \in \boldsymbol{A}$ *is defined as a tuple*

$$\tau = \langle \mathcal{V}, \mathcal{E}, d, f, c \rangle,$$

where $\langle \mathcal{V}, \mathcal{E} \rangle$ *is a tree with nodes in* $\mathcal{V}$ *and edges in* $\mathcal{E}$*, and whose set of branches is denoted by* $\mathcal{B}$*. The vertices are partially ordered by a relation* $\lhd$*, which is induced by the edges. The function*

$$d \colon \mathcal{V} \to \mathcal{D}$$

is a node labeling function *that assigns each node* ν *a decoration of the form* $Q(\beta \preceq \psi, [w,z], \widetilde{\mathbb{D}})$ *or* $Q(\psi \preceq \beta, [w,z], \widetilde{\mathbb{D}})$*, where* $\beta \in \boldsymbol{A}$ *and* ψ *is a subformula of* φ *(denoted as* $\psi \in sub(\varphi)$*). The function*

$$f \colon \mathcal{V} \to \{0,1\}$$

is a node flag function *indicating whether nodes are expanded (1) or not expanded (0). The function*

$$c \colon \mathcal{B} \to \mathfrak{D}$$

is a branch labeling function *that associates to every branch the finite many-valued linear order that belongs to the decoration of its leaf. The initial tableau* τ_0 *is*

$$\langle \{\nu_0\}, \emptyset, \{\nu_0 \mapsto (T(\alpha \preceq \varphi), [x,y], \{\widetilde{<}(x,y) = \beta\})\}, \{\nu_0 \mapsto 0\}, \{\nu_0 \mapsto \{\widetilde{<}(x,y)) = \beta\}\} \rangle$$

or

$$\langle \{\nu_0\}, \emptyset, \{\nu_0 \mapsto (F(\alpha \preceq \varphi), [x,y], \{\widetilde{<}(x,y) = \beta\})\}, \{\nu_0 \mapsto 0\}, \{\nu_0 \mapsto \{\widetilde{<}(x,y)) = \beta\}\} \rangle$$

for some $\beta \in \boldsymbol{A}$*, and it evolves by iteratively applying the* branch expansion rule *(see Fig. 2) or, if not applicable, the* reverse rule *in (see Fig. 3). These rules are applied to the node* ν *closest to the root such that* $f(\nu) = 0$*, affecting every descendant leaf* ν' *for which* $\nu \lhd \nu'$*. An application of either rule to* ν *where* $f(\nu) = 0$ *will result in setting* $f(\nu)$ *to 1.*

$$(T\wedge)\ \frac{T(\beta \preceq (\psi \wedge \xi), [x,y], C)}{\begin{array}{c} T(\beta_1 \preceq \psi, [x,y], C(B)) \mid \ldots \mid T(\beta_n \preceq \psi, [x,y], C(B)) \\ T(\gamma_1 \preceq \xi, [x,y], C(B)) \mid \ldots \mid T(\gamma_n \preceq \xi, [x,y], C(B)) \end{array}}$$

where $\beta \neq 0$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta \preceq \beta_i \cdot \gamma_i$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta \preceq \beta'_i \cdot \gamma'_i, \beta'_i \preceq \beta_i$ and $\gamma'_i \preceq \gamma_i$.

$$(F\wedge)\ \frac{F(\beta \preceq (\psi \wedge \xi), [x,y], C)}{\begin{array}{c} T(\psi \preceq \beta_1, [x,y], C(B)) \mid \ldots \mid T(\psi \preceq \beta_n, [x,y], C(B)) \\ T(\xi \preceq \gamma_1, [x,y], C(B)) \mid \ldots \mid T(\xi \preceq \gamma_n), [x,y], C(B) \end{array}}$$

where $\beta \neq 0$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta \npreceq \beta_i \cdot \gamma_i$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta \npreceq \beta'_i \cdot \gamma'_i, \beta_i \preceq \beta'_i$ and $\gamma_i \preceq \gamma'_i$.

$$(T\vee)\ \frac{T((\psi \vee \xi) \preceq \beta, [x,y], C)}{\begin{array}{c} T(\psi \preceq \beta_1, [x,y], C(B)) \mid \ldots \mid T(\psi \preceq \beta_n, [x,y], C(B)) \\ T(\xi \preceq \gamma_1, [x,y], C(B)) \mid \ldots \mid T(\xi \preceq \gamma_n, [x,y], C(B)) \end{array}}$$

where $\beta \neq 1$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta_i + \gamma_i \preceq \beta$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta'_i + \gamma'_i \preceq \beta, \beta_i \preceq \beta'_i$ and $\gamma_i \preceq \gamma'_i$.

$$(F\vee)\ \frac{F((\psi \vee \xi) \preceq \beta, [x,y], C)}{\begin{array}{c} T(\beta_1 \preceq \psi, [x,y], C(B)) \mid \ldots \mid T(\beta_n \preceq \psi, [x,y], C(B)) \\ T(\gamma_1 \preceq \xi, [x,y], C(B)) \mid \ldots \mid T(\gamma_n \preceq \xi, [x,y], C(B)) \end{array}}$$

where $\beta \neq 1$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta_i + \gamma_i \npreceq \beta$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta'_i + \gamma'_i \npreceq \beta, \beta'_i \preceq \beta_i$ and $\gamma'_i \preceq \gamma_i$.

$$(T \hookrightarrow)\ \frac{T(\beta \preceq (\psi \hookrightarrow \xi), [x,y], C)}{\begin{array}{c} T(\psi \preceq \beta_1, [x,y], C(B)) \mid \ldots \mid T(\psi \preceq \beta_n, [x,y], C(B)) \\ T(\gamma_1 \preceq \xi, [x,y], C(B)) \mid \ldots \mid T(\gamma_n \preceq \xi, [x,y], C(B)) \end{array}}$$

where $\beta \neq 0$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta \preceq \beta_i \hookrightarrow \gamma_i$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta \preceq \beta'_i \hookrightarrow \gamma'_i, \beta_i \preceq \beta'_i$ and $\gamma'_i \preceq \gamma_i$.

$$(F \hookrightarrow)\ \frac{F(\beta \preceq (\psi \hookrightarrow \xi), [x,y], C)}{\begin{array}{c} T(\beta_1 \preceq \psi, [x,y], C(B)) \mid \ldots \mid T(\beta_n \preceq \psi, [x,y], C(B)) \\ T(\xi \preceq \gamma_1, [x,y], C(B)) \mid \ldots \mid T(\xi \preceq \gamma_n, [x,y], C(B)) \end{array}}$$

where $\beta \neq 0$, $(\beta_i, \gamma_i) \in \boldsymbol{A} \times \boldsymbol{A}$ so that $\beta \npreceq \beta_i \hookrightarrow \gamma_i$ and there is no other $(\beta'_i, \gamma'_i) \in \boldsymbol{A} \times \boldsymbol{A}$ such that $\beta \npreceq \beta'_i \hookrightarrow \gamma'_i, \beta'_i \preceq \beta_i$ and $\gamma_i \preceq \gamma'_i$.

Figure 2 Propositional rules.

$$(T\succeq)\ \frac{T(\beta\preceq\psi,[x,y],C)}{F(\psi\preceq\gamma,[x,y],C(B))}$$
where $\varphi\neq\alpha$, $\beta\neq 0$ and γ is any maximal element not above β, i.e., $\gamma\not\succeq\beta$

$$(F\succeq)\ \frac{F(\beta\preceq\psi,[x,y],C}{T(\psi\preceq\gamma_i,[x,y],C(B))\mid\ldots\mid T(\psi\preceq\gamma_n,[x,y],C(B))}$$
where $\varphi\neq\alpha$, $\beta\neq 0$ and $\gamma_1,\ldots,\gamma_n$ are all maximal elements not above β, i.e., $\gamma_1,\ldots,\gamma_n\not\succeq\beta$

$$(T\preceq)\ \frac{T(\psi\preceq\beta,[x,y],C)}{F(\gamma\preceq\psi,[x,y],C(B))}$$
where $\varphi\neq\alpha$, $\beta\neq 1$ and γ is any minimal element not below β, i.e., $\gamma\not\preceq\beta$

$$(F\preceq)\ \frac{F(\psi\preceq\beta,[x,y],C)}{T(\gamma_i\preceq\psi,[x,y],C(B))\mid\ldots\mid T(\gamma_i\preceq\psi,[x,y],C(B))}$$
where $\varphi\neq\alpha$, $\beta\neq 1$ and $\gamma_1,\ldots,\gamma_n$ are all minimal elements not below β, i.e., $\gamma_1,\ldots,\gamma_n\not\preceq\beta$

Figure 3 Reverse rules.

$$(T\Box)\ \frac{T(\beta\preceq[X]\psi,[x,y],C)}{\begin{array}{c}T((\beta\cdot\gamma_1)\preceq\psi,[z_1,t_1],c(B))\\ \cdots\\ T((\beta\cdot\gamma_n)\preceq\psi,[z_n,t_n],c(B))\\ T(\beta\preceq[X]\psi,[x,y],c(B))\end{array}}$$
where $\gamma_i=R_X([x,y],[z_i,t_i])$, $[z_i,t_i]\in o(c(B))$, $\gamma_i\succ 0$, and $\beta\cdot\gamma_i\neq 0$

$$(T\Diamond)\ \frac{T(\langle X\rangle\psi\preceq\beta,[x,y],C)}{\begin{array}{c}T((\psi\preceq(\gamma_1\hookrightarrow\beta),[z_1,t_1],c(B))\\ \cdots\\ T(\psi\preceq(\gamma_n\hookrightarrow\beta),[z_n,t_n],c(B))\\ T(\langle X\rangle\psi\preceq\beta,[x,y],c(B))\end{array}}$$
where $\gamma_i=R_X([x,y],[z_i,t_i])$, $[z_i,t_i]\in o(c(B))$, $\gamma_i\succ 0$, and $\gamma_i\hookrightarrow\beta\neq 1$

$$(F\Box)\ \frac{F(\beta\preceq[X]\psi,[x,y],C)}{F((\beta\cdot\gamma_1)\preceq\psi,[z_1,t_1],c(B))\mid\ldots\mid F((\beta\cdot\gamma_n)\preceq\psi,[z_n,t_n],c(B))}$$
where $\gamma_i=R_X([x,y],[z_i,t_i])$,$[z_i,t_i]\in o(c(B))\cup n(c(B))$, $\gamma_i\succ 0$, and $\beta\cdot\gamma_i\neq 0$

$$(F\Diamond)\ \frac{F(\langle X\rangle\psi\preceq\beta,[x,y],C)}{F(\psi\preceq(\gamma_1\hookrightarrow\beta),[z_1,t_1],c(B))\mid\ldots\mid F(\psi\preceq(\gamma_n\hookrightarrow\beta),[z_n,t_n],c(B))}$$
where $\gamma_i=R_X([x,y],[z_i,t_i])$,$[z_i,t_i]\in o(c(B))\cup n(c(B))$, $\gamma_i\succ 0$, and $\gamma_i\hookrightarrow\beta\neq 1$

Figure 4 Temporal rules.

$$(\text{✗}1)\ \frac{T(\beta\preceq\gamma,[x,y],C)}{\text{✗}}$$
where $\beta\not\preceq\gamma$

$$(\text{✗}2)\ \frac{F(\beta\preceq\gamma,[x,y],C)}{\text{✗}}$$
where $\beta\neq 0$, $\gamma\neq 1$, and $\beta\preceq\gamma$

$$(\text{✗}3)\ \frac{F(0\preceq\psi,[x,y],C)}{\text{✗}}$$

$$(\text{✗}4)\ \frac{F(\psi\preceq 1,[x,y],C)}{\text{✗}}$$

$$(\text{✗}5)\ \frac{\begin{array}{c}T(\gamma\preceq\psi,[x,y],C)\\ F(\beta\preceq\psi)\end{array}}{\text{✗}}$$
where $\beta\preceq\gamma$

$$(\text{✗}6)\ \frac{Q(\cdot,\cdot,C)}{\text{✗}}$$
where C is inconsistent

Figure 5 Branch closing rules.

▶ **Definition 6** (open and closed tableau). *Given a finite* $\mathrm{FL}_{\mathrm{ew}}$*-algebra* $\boldsymbol{A}$ *and an* $\boldsymbol{A}$*-formula* φ*, a tableau* τ *for* φ *and* $\alpha \in \boldsymbol{A}$ *is is* closed *if the* branch closing rules *(see Fig. 5), can be applied to all of its branches. Conversely,* τ *is* open *if there exists at least one branch to which the branch closing rules cannot be applied.*

An open (resp., closed) tableau for φ and α, initiated with the decoration $T(\alpha \preceq \varphi)$ (resp., $F(\alpha \preceq \varphi)$), effectively demonstrates that φ is α-satisfiable (resp., α-valid). Conversely, the reversed decorations $T(\varphi \preceq \alpha)$ and $F(\varphi \preceq \alpha)$ are generally not employed as starting decorations to prove substantial statements. For example, $T(\varphi \preceq \alpha)$ might suggest the possibility of constructing a structure where φ has a value less than α. In classical logic, with a two-element Boolean algebra and $\alpha = 1$, this condition would imply that $\neg\varphi$ is valid, as φ would consistently take the value 0. However, in the many-valued case, such duality is absent, and there is no guarantee of a formula that consistently attains the value 1 when φ is valued strictly less than 1.

Despite their limited use as initial decorations, these reversed decorations, in conjunction with the reverse rule, facilitate reducing the number of necessary rules. For instance, a decoration including the judgment $T(p \preceq \beta)$ appearing in a branch of a tableau starting with the judgment $T(\alpha \preceq \varphi)$ informs us that in any model validating α-satisfiability of φ, the propositional variable p must take a value less than or equal to β. For any given formula φ and value α, a tableau starting with $T(\alpha \preceq \varphi)$ (resp., $F(\varphi \preceq \alpha)$) is termed a *SAT-tableau* (resp., *VAL-tableau*).

Two distinct notions of soundness and completeness are applicable: one for the SAT-tableau system and another for the VAL-tableau system. Although the foundational arguments for both systems bear strong similarities, our statement is focused on the former.

▶ **Theorem 7** (soundness and completeness for α-satisfiability). *The tableau system for the MVHS of finite* $\mathrm{FL}_{\mathrm{ew}}$*-algebras is sound and complete for proving* α*-satisfiability, that is, given a finite* $\mathrm{FL}_{\mathrm{ew}}$*-algebra* $\boldsymbol{A}$*, an* $\boldsymbol{A}$*-formula* φ*, and a value* $\alpha \in A$*,* φ *is* α*-satisfiable if and only if some SAT-tableau for* φ *and* α *is open.*

Technically speaking, the proof of this theorem is essentially identical to that of soundness and completeness of the version of this tableau given in [13], and it is therefore omitted.

Implementation. The tableau system for MVHS with finite $\mathrm{FL}_{\mathrm{ew}}$-algebras has been implemented using the Julia programming language as part of a broader open-source project aimed at representing, reasoning, and learning from structured and unstructured data [27].

In our implementation, formulas are represented as syntax trees with leaves consisting of either propositional letters or algebra values. Finite $\mathrm{FL}_{\mathrm{ew}}$-algebras are configured by establishing a finite domain, defining operators through their tables, and ensuring, via a one-time check, that all axioms proper to an $\mathrm{FL}_{\mathrm{ew}}$-algebra are satisfied. Formulas can be generated randomly, while algebras are systematically created with progressively larger domains to explore the impact of algebra size on system performance.

The tableau itself is structured as a rooted tree with variable arity. Each node in the tableau has a decoration and maintains two pointers, one to its parent node and the other to the head of a list of its children, which can be empty. Nodes also carry two flags to track whether they have been expanded (or not) and whether they are open (or closed). Specifically, a closed node indicates that all branches containing it are also closed. The tableau is linked to an array of priority queues where each queue node references a tableau node and carries a value determined by a priority function. This priority setting is crucial during the expansion phase to determine the next branch for expansion.

The construction of the tableau follows a standard search algorithm, beginning with an empty node to which several immediate children are attached; each child corresponds to a different algebra value for the relation $\widetilde{<}$ between the two initial points. At each cycle, we evaluate all nodes at the top of any priority queue. We select one node from this group based on a predefined meta-policy (e.g., random or majority). The selected node, immediately removed from other queues where it was at the top, identifies a set of branches chosen for expansion. We first verify whether this node, denoted as ν, is already closed; if so, the cycle proceeds without action. Next, if ν has already been expanded, the process skips to the next cycle unless ν is a leaf, signaling an open branch in a fully expanded tableau. For a SAT-tableau, such an open branch identifies a model that α-satisfies the starting formula; for a VAL-tableau, it represents a counterexample to its α-validity.

If ν has not been expanded, we identify the nearest unexpanded ancestor ν' of ν and mark it as expanded. The subsequent steps are to apply the closing rule, thereby closing ν' and its descendants (including ν). If this is not applicable, the propositional rule extends the tree from all leaves containing ν' and inserts the fresh nodes into all priority queues checking that each fresh node was not already present in the branch; in such case, if no node is added to that branch, a virtual node containing $T(\top \preceq \top)$ is inserted to keep track of the presence of a branch. If no propositional rule is applicable, the temporal rule is applied with the only difference that when applying the $T\Box$ and the $T\Diamond$ rule, if at least one node has been added to the branch before $T(\beta \preceq [X]\psi, [x, y], C(B))$ (resp. $T(\beta \preceq \langle X\rangle\psi, [x, y], C(B)))$ we still add the latter, as we may have new points in $c(B)$ w.r.t. C; otherwise, we end the tableau procedure, as we have found an open branch with no contradictions within that is no longer expandable. Again, if this is not applicable, the reverse rule is applied with implications for all branch leaves containing ν', inserting all fresh nodes in all priority queues with the same precautions taken for the expansion rules. Finally, if none of the above apply, ν is reinserted into the priority queues from which was extracted. The process concludes when all priority queues are empty, indicating no open branches remain, and the tableau is fully expanded and declared closed. For a SAT-tableau, this outcome proves the α-unsatisfiability of the starting formula; for a VAL-tableau, it demonstrates α-validity.

Optimizations. To improve both time and space efficiency of our implementation, we periodically perform a cleaning operation on all priority queues. For a parameter $K \in \mathbb{N}$, every K cycles all nodes within a queue that have already been expanded (unless they are leaves) or that are already closed are removed. This operation helps manage the computational overhead and optimizes the performance of the tableau construction. Moreover, since the creation of each new branch in these rules is independent from the others and has a non-negligible cost, as for each new node we have to check that no inconsistencies are introduced, this process has been parallelized, introducing also a locking system to manage the concurrent writing to both the tableau structure and the heaps.

All standard search strategies can be used within our system; different search strategies can be used at the same time, in a round-robin policy, or even with a further level of parallelization, in order to implement a virtual best solver policy. Given the high computation complexity of these problems, however, we designed (but not tested) a specific search strategy based on a weighting function that quantifies how ‘representative, that is, how different a branch is from others already tested. Such a strategy can be used in a complete form (branches are explored in order of representativeness), or an incomplete form (only the most representative branches are explored). Future work includes a systematic assessment of the usefulness of such strategies.

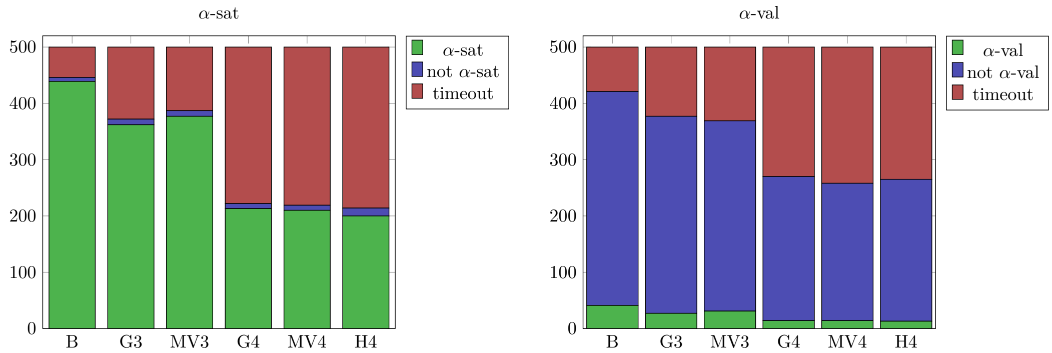


Figure 6 Results on Fig.1 algebras for formulas of height up to 5 with a timeout of 30 seconds.

5 Experiments

The objective of our experiments was to investigate the impact of different finite $\mathrm{FL_{ew}}$-algebras as well as growth in formula height on computational performance. This analysis was conducted involving both SAT-tableaux and VAL-tableaux.

Initially, we generated six representative finite $\mathrm{FL_{ew}}$-algebras: the Boolean algebra ($\mathbb{B}$), used as the baseline, the Gödel algebras and MV-algebras with $n \in \{3, 4\}$ values (Gn and MVn), and the diamond algebra with 4 values (H4). The lattice structures for these algebras are depicted in Fig. 1 (right-hand side). It is important to note that while G3 and MV3 (as well as G4 and MV4) are based on identical lattices, they are distinguished by their respective norms. We generated 100 random formulas for each algebra and for each height up to 5 (i.e., up to 32 symbols). Formula generation was governed by a weighted selection process: connectives $\{\wedge, \vee, \rightarrow\}$ were assigned a weight of 8, modalities $\{\langle A\rangle, \langle L\rangle, \langle B\rangle, \langle E\rangle, \langle D\rangle, \langle O\rangle, [A], [L], [B], [E], [D], [O]\}$ as well as their inverse were assigned a weight a 1 (so that one has the same probability to get a connective or a modality), algebra values received a weight of $1/|A|$ where A is the domain of the algebra, and each propositional letter in the set $\mathcal{P}$ was weighted by $1/|\mathcal{P}|$. Selection probabilities for a symbol s were calculated as

$$P(s) = \frac{W[s]}{\Sigma_{\{t|t\in\{\wedge,\vee,\rightarrow\}\cup\{\langle X\rangle|X\in\mathcal{X}\}\cup A\cup\mathcal{P}\}} W[t]},$$

with formula expansion ceasing upon reaching the designated height or if further expansion is impossible (in which case it is discarded). The performance of each formula in terms of α-satisfiability and α-validity, with α chosen randomly, was analyzed; the results are depicted in Fig. 6. Throughout these experiments, the branch priority policy was kept random (and complete), and the choice of α was also randomized. All tests were conducted on a machine equipped with 2 Intel Xeon Gold 28-Core CPUs and 224GB of RAM.

6 Conclusions

In this paper, we expanded previous work on a tableau-based reasoning system tailored for many-valued interval temporal logic. In particular, we provided a reasoning system for MVHS in a very general case, and we focused on its implementation, which we made available as part of a comprehensive symbolic learning and reasoning framework [27]. We have also designed and carried out a series of tests to study the scalability of the system. As we have mentioned, future work includes exploring more elaborate search strategies and testing their effectiveness.

Moving forward, our aim is to build on the integration of symbolic learning models and reasoning systems and to provide end-to-end solutions for full data-driven learning, reasoning, and decision-making processes.

References

1 J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.

2 M. Baaz, N. Preining, and R. Zach. First-order Gödel logics. *Annals of Pure and Applied Logic*, 147:23–47, 2007. doi:10.1016/J.APAL.2007.03.001.

3 U. Bodenhofer. Representations and constructions of similarity-based fuzzy orderings. *Fuzzy Sets and Systems*, 137(1):113–136, 2003. doi:10.1016/S0165-0114(02)00436-0.

4 G. E. Brancati, E. Vieta, J. M. Azorin, J. Angst, C. L. Bowden, S. Mosolov, A. H. Young, and G. Perugi. The role of overlapping excitatory symptoms in major depression: are they relevant for the diagnosis of mixed state? *Journal of Psychiatric Research*, 115:151–157, 2019.

5 D. Bresolin, D. Della Monica, A. Montanari, and G. Sciavicco. A tableau system for right propositional neighborhood logic over finite linear orders: An implementation. In *Proc. of the 22th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX)*, volume 8123 of *LNCS*, pages 74–80. Springer, 2013. doi:10.1007/978-3-642-40537-2_8.

6 A. Brunello, G. Sciavicco, and I. E. Stan. Interval temporal logic decision tree learning. In *Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA)*, volume 11468 of *LNCS*, pages 778–793. Springer, 2019. doi:10.1007/978-3-030-19570-0_50.

7 J. J. Buckley and Y. Hayashi. Fuzzy neural networks: A survey. *Fuzzy Sets and Systems*, 66:1–13, 1994.

8 C. C. Chang. Algebraic analysis of many valued logics. *Transactions of the American Mathematical society*, 88(2):467–490, 1958.

9 Y. Chen, T. Wang, B. Wang, and Z. Li. A survey of fuzzy decision tree classifier. *Fuzzy Information and Engineering*, 1(2):149–159, 2009.

10 P. Cintula, P. Hájek, and C. Noguera, editors. *Handbook of Mathematical Fuzzy Logic*, volume 37-38 of *Studies in Logic. Mathematical Logic and Foundation*. College publications, 2011.

11 W. Conradie, D. Della Monica, E. Muñoz-Velasco, and G. Sciavicco. An approach to fuzzy modal logic of time intervals. In *Proc. of the 24th European Conference on Artificial Intelligence (ECAI)*, volume 325 of *FAIA*, pages 696–703. IOS Press, 2020. doi:10.3233/FAIA200156.

12 W. Conradie, D. Della Monica, E. Muñoz-Velasco, G. Sciavicco, and I. E. Stan. Fuzzy Halpern and Shoham's interval temporal logics. *Fuzzy Sets and Systems*, 456:107–124, 2023. doi:10.1016/J.FSS.2022.05.014.

13 W. Conradie, R. Monego, E. Muñoz-Velasco, G. Sciavicco, and I. E. Stan. A sound and complete tableau system for fuzzy Halpern and Shoham's interval temporal logic. In *Proc. of the 30th International Symposium on Temporal Representation and Reasoning (TIME)*, volume 278 of *LIPIcs*, pages 9:1–9:14. Schloss Dagstuhl, 2023. doi:10.4230/LIPICS.TIME.2023.9.

14 S. Dutta. An event based fuzzy temporal logic. In *Proc. of the 18th International Symposium on Multiple-Valued Logic*, pages 64–71, 1988.

15 L. Esakia, G. Bezhanishvili, W. H. Holliday, and A. Evseev. *Heyting Algebras: Duality Theory*. Springer, 2019.

16 M. Fitting. Many-valued modal logics. *Fundamenta Informaticae*, 15(3-4):235–254, 1991.

17 M. Fitting. Tableaus for many-valued modal logic. *Studia Logica*, 55(1):63–87, 1995. doi:10.1007/BF01053032.

18 A. Frigeri, L. Pasquale, and P. Spoletini. Fuzzy time in linear temporal logic. *ACM Transactions on Computational Logic*, 15:1–22, 2014. doi:10.1145/2629606.

19 V. Goranko, A. Montanari, P. Sala, and G. Sciavicco. A general tableau method for propositional interval temporal logics: Theory and implementation. *Journal of Applied Logics*, 4(3):305–330, 2006. doi:10.1016/J.JAL.2005.06.012.

20 V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003. doi:10.3217/JUCS-009-09-1137.

21 V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1–2):9–54, 2004. doi:10.3166/JANCL.14.9-54.

22 P. Hájek. *The Metamathematics of Fuzzy Logic*. Kluwer, 1998.

23 J. Y. Halpern and Y. Shoham. A Propositional Modal Logic of Time Intervals. *Journal of the ACM*, 38(4):935–962, 1991. doi:10.1145/115234.115351.

24 L. I. Kuncheva. *Fuzzy Classifier Design*, volume 49 of *Studies in Fuzziness and Soft Computing*. Springer, 2000. doi:10.1007/978-3-7908-1850-5.

25 S. Kundu. Similarity relations, fuzzy linear orders, and fuzzy partial orders. *Fuzzy Sets and Systems*, 109(3):419–428, 2000. doi:10.1016/S0165-0114(97)00370-9.

26 K. B. Lamine and F. Kabanza. Using fuzzy temporal logic for monitoring behavior-based mobile robots. In *Proc. of the IASTED International Conference, Robotics and Applications*, pages 116–121, 2000.

27 F. Manzella, G. Pagliarini, A. Paparella, G. Sciavicco, and I. E. Stan. Sole.jl – Symbolic Learning in Julia. https://github.com/aclai-lab/Sole.jl, 2024.

28 F. Manzella, G. Pagliarini, G. Sciavicco, and I. E. Stan. Interval Temporal Random Forests with an Application to COVID-19 Diagnosis. In *Proceedings of the 28th International Symposium on Temporal Representation and Reasoning (TIME)*, volume 206 of *LIPIcs*, pages 7:1–7:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.TIME.2021.7.

29 E. Muñoz-Velasco, M. Pelegrín-Garcí, P. Sala, G. Sciavicco, and I. E. Stan. On coarser interval temporal logics. *Artificial Intelligence*, 266:1–26, 2019. doi:10.1016/J.ARTINT.2018.09.001.

30 S. Ovchinnikov. Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems*, 40(1):107–126, 1991.

31 A. Rose. Formalisations of further $\aleph_0$-valued Łukasiewicz propositional calculi. *Journal of Symbolic Logic*, 43(2):207–210, 1978. doi:10.2307/2272818.

32 G. Sciavicco and I. E. Stan. Knowledge extraction with interval temporal logic decision trees. In *Proceedings of the 27th International Symposium on Temporal Representation and Reasoning (TIME)*, volume 178 of *LIPIcs*, pages 9:1–9:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.TIME.2020.9.

33 H. Thiele and S. Kalenka. On fuzzy temporal logic. In *Proc. of the 2nd International Conference on Fuzzy Systems*, pages 1027–1032. IEEE, 1993.

34 L. A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3(2):177–200, 1971. doi:10.1016/S0020-0255(71)80005-1.