A tableau for Many-Valued Multi-Modal Logic

41st Italian Conference on Computational Logic (CILC)

Guillermo Badia[1] Riccardo Monego[2] Carles Noguera[3] **Alberto Paparella**[4] Guido Sciavicco[4] Ionel Eduard Stan[5]

June 25th, 2026

[1]School of Historical and Philosophical Inquiry, University of Queensland, Brisbane, Australia

[2]Dept. of Mathematics, University of Torino, Torino, Italy

[3]Dept. of Information Engineering and Mathematics, University of Siena, Siena, Italy

[4]Dept. of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy

[5]Dept. of Informatics, Systems and Communication, University of Milano-Bicocca, Milano, Italy

Table of contents

Introduction

Motivation

Why Multi-Modal Logic?

- good **complexity** tradeoff (propositional logic $<$ modal logic $<$ f.o.)
- comprise many **temporal** and **spatial** logics (LTL, CL, HS, L^{Rec}_{RCC8})

Motivation

Why Multi-Modal Logic?

- good **complexity** tradeoff (propositional logic $<$ modal logic $<$ f.o.)
- comprise many **temporal** and **spatial** logics (LTL, CL, HS, L^{Rec}_{RCC8})

Why Many-Valued Logic?

- **Inacuraccies** in the data (e.g., due to sensing and discretization)
- **Unclear boundaries** due to treating numerical values semantically (when is a temperature high/low?)
- **Many-expert** systems[1]

[1]M. Fitting. **"Many-valued modal logics".** In: *Fundamenta Informaticae* 15.3-4 (1991), pp. 235–254

Problem: which Many-Valued Logic?

Many-Valued logics can be based on a variety of families of algebras:

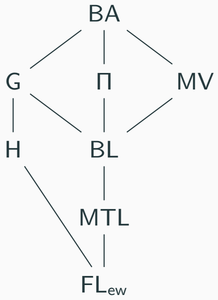

Figure 1: A partial taxonomy of well-known many-valued algebras, namely: Boolean algebra (BA), Gödel algebras (G), Product algebras (Π), MV-algebras (MV), Heyting algebras (H), Basic Fuzzy Logic algebras (BL), Monoidal t-norm logic algebras (MTL), and $\mathrm{FL_{ew}}$-algebras ($\mathrm{FL_{ew}}$).

The sweet spot: $\mathrm{FL_{ew}}$-algebras

Definition

$\mathrm{FL_{ew}}$-**algebras**[2]

$$\mathcal{A} = \langle A, \cap, \cup, \cdot, 0, 1 \rangle$$

are defined over **bounded commutative residuated lattices**, where:

- $\langle A, \cap, \cup, 0, 1 \rangle$ represents a **bounded complete lattice**
- $\langle A, \cdot, 1 \rangle$ is a **commutative monoid**
- We can define an **implication** $\hookrightarrow$ (the residuation property holds)

$$\alpha \hookrightarrow \beta = \sup\{\gamma \mid \alpha \cdot \gamma \preceq \beta\}$$

$\mathcal{A}$ is a **chain** if $\langle A, \preceq \rangle$ is a total order, **finite** if A is finite.

[2] Hiroakira Ono and Yuichi Komori. **"Logics without the contraction rule".** In: *The Journal of Symbolic Logic* 50.1 (1985), pp. 169–201

Many-Valued Multi-Modal Logic

Many-Valued Multi-Modal Logic

Definition

Let $\mathcal{A}P$ be a set of propositional letters, $\mathcal{A}$ a complete $\mathrm{FL}_{\mathrm{ew}}$-algebra. Well-formed formulas of *Multi-Modal Logic* $\mathrm{FL}_{\mathrm{ew}}$-$K_n$ are obtained as:

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

for $1 \leq i \leq n$, $p \in \mathcal{A}P$, and $\alpha \in \mathcal{A}$.

Problem: the double negation axiom ($\neg\neg\varphi \equiv \varphi$) is **not** always vaild.

Solution: explicit inclusion of all Boolean operators, as well as the universal version of every modality.

Many-Valued Kripke frames and structures

To reason in Many-Valued Multi-Modal Logic, we need to manify the notions of Kripke frame and Kripke structure.

Many-Valued Kripke frames and structures

To reason in Many-Valued Multi-Modal Logic, we need to manify the notions of Kripke frame and Kripke structure.

Definition

Given a non-empty set of *worlds* W and a complete $\mathrm{FL_{ew}}$-algebra $\mathcal{A}$, an $\mathrm{FL_{ew}}$-*Kripke frame* is an object $\widetilde{F} = \langle W, \widetilde{R}_1 \ldots, \widetilde{R}_n \rangle$, where each $\widetilde{R}_i : (W \times W) \to \mathcal{A}$ is an *accessibility* function.

Many-Valued Kripke frames and structures

To reason in Many-Valued Multi-Modal Logic, we need to manify the notions of Kripke frame and Kripke structure.

Definition

Given a non-empty set of *worlds* W and a complete $\mathrm{FL_{ew}}$-algebra $\mathcal{A}$, an $\mathrm{FL_{ew}}$-*Kripke frame* is an object $\widetilde{F} = \langle W, \widetilde{R}_1 \ldots, \widetilde{R}_n \rangle$, where each $\widetilde{R}_i : (W \times W) \to \mathcal{A}$ is an *accessibility* function.

Definition

An $\mathrm{FL_{ew}}$-*Kripke structure* (or *model*) is an $\mathrm{FL_{ew}}$-Kripke frame enriched with a *valuation function* $\widetilde{V} : (W \times \mathcal{A}P) \to \mathcal{A}$, and it is denoted by $\widetilde{M} = \langle \widetilde{F}, \widetilde{V} \rangle$.

Many-Valued Multi-Modal Logic semantics

Definition

Given a well-formed formula φ, we compute its *value in* $\widetilde{M}$ *at* w, for some $w \in W$, by extending $\widetilde{V}$ to formulas, as follows:

$$\begin{array}{lcl}
\widetilde{V}(\alpha, w) & = & \alpha, \\
\widetilde{V}(\varphi \wedge \psi, w) & = & \widetilde{V}(\varphi, w) \cdot \widetilde{V}(\psi, w), \\
\widetilde{V}(\varphi \vee \psi, w) & = & \widetilde{V}(\varphi, w) \cup \widetilde{V}(\psi, w), \\
\widetilde{V}(\varphi \rightarrow \psi, w) & = & \widetilde{V}(\varphi, w) \hookrightarrow \widetilde{V}(\psi, w), \\
\widetilde{V}(\langle X_i \rangle \varphi, w) & = & \bigcup \{\widetilde{R}_i(w, s) \cdot \widetilde{V}(\varphi, s)\}, \\
\widetilde{V}([X_i]\varphi, w) & = & \bigcap \{\widetilde{R}_i(w, s) \hookrightarrow \widetilde{V}(\varphi, s)\}.
\end{array}$$

The α-satisfiability problem

Definition

A formula φ of $\mathrm{FL_{ew}}$-K$_n$ is α-*satisfied* at world w in an $\mathrm{FL_{ew}}$-Kripke structure $\widetilde{M}$ if and only if

$$\widetilde{V}(\varphi, w) \succeq \alpha.$$

The α-satisfiability problem

Definition

A formula φ of $\mathrm{FL_{ew}}$-K_n is α-*satisfied* at world w in an $\mathrm{FL_{ew}}$-Kripke structure $\widetilde{M}$ if and only if

$$\widetilde{V}(\varphi, w) \succeq \alpha.$$

Definition

A formula φ is α-*satisfiable* if and only if there exists a structure and a world in which it is α-satisfied, and it is *satisfiable* if it is α-satisfiable for some $\alpha \in \mathcal{A}$, $\alpha \succ 0$; respectively, a formula is α-*valid* if it is α-satisfied at every world in every model, and *valid* if it is 1-valid.

A tableau for Many-Valued Multi-Modal Logic

Overview of the Tableau Structure

- Tree-like structure with nodes and branches
- Each node is associated with a **decoration**

$$Q(\alpha \preceq \varphi, w, \widetilde{F}) \text{ or } Q(\varphi \preceq \alpha, w, \widetilde{F}),$$

  where Q is a judgment (either *true* or *false*), $\alpha \in A$ is a truth value from the chosen $\mathrm{FL}_{\mathrm{ew}}$-algebra, φ is a formula, w is a world, and $\widetilde{\mathbb{F}}$ is a many-valued Kripke structure
- Branches represent possible **evaluations** and are associated with a many-valued Kripke structure

Overview of the Tableau Procedure

- Systematically apply **expansion rules** to generate new nodes
- Close branches that contain contradictions using **closing rules**
- Systematically explore possible valuations to determine:
  - α-**satisfiability**, if starting from *true*$(\alpha \preceq \varphi)$ it finds an open branch
  - α-**validity**, if starting from *false*$(\alpha \preceq \varphi)$ it closes all branches

Example: $\Diamond(p \wedge \alpha) \wedge \alpha$ is α-satisfiable in G3

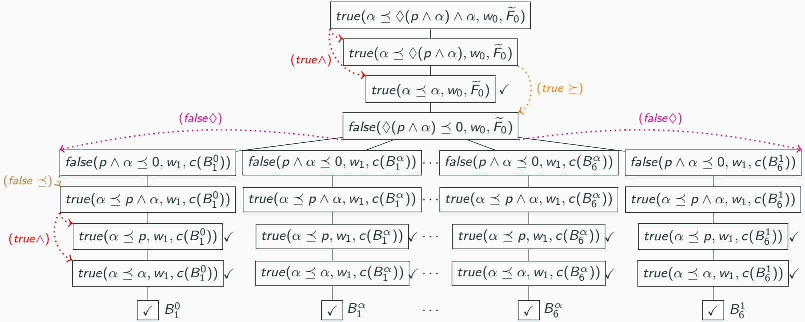

Figure 2: Fully evaluated SAT-tableau for $\alpha \preceq \Diamond(p \wedge \alpha) \wedge \alpha$ in G3. Note that we only needed to fully expand one branch (e.g., B_1^0), to prove α-satisfiability; here, we choose to show all possible branches for example purposes.

Example: $\Diamond(p \wedge \alpha) \wedge \alpha$ is not α-satisfiable in MV3

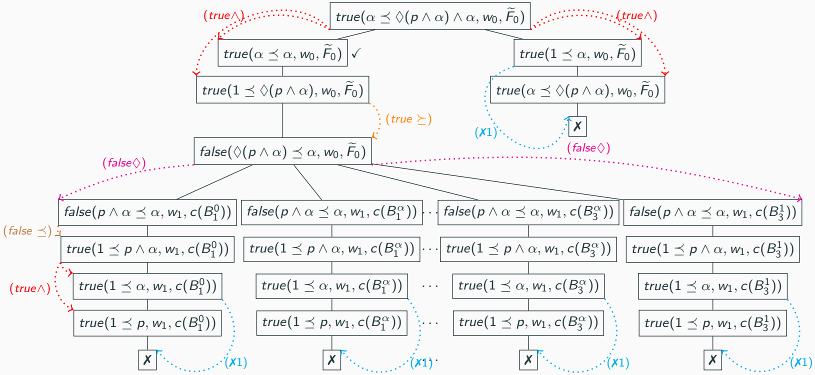

Figure 3: Fully evaluated SAT-tableau for $\alpha \preceq \Diamond(p \wedge \alpha) \wedge \alpha$ in MV3. Applying the rules of the tableau system results in closing all branches, proving that the formula is not α-satisfiable in MV3.

Soundness and Completeness

- (**Soundness**) If a formula φ is α-satisfiable, then there exists an opened tableau for φ and α
- (**Completeness**) If a tableau is opened for φ and α, then φ is α-satisfiable (i.e., the method explores all necessary valuations)
- Similar statements can be made for α-validity

Experimental results

SoleReasoners.jl

The reasoning system has been implemented in the **Julia** programming language as part of **SOLE**, a framework for learning, reasoning and post hoc analysis, and is provided through the **SoleReasoners.jl** package.

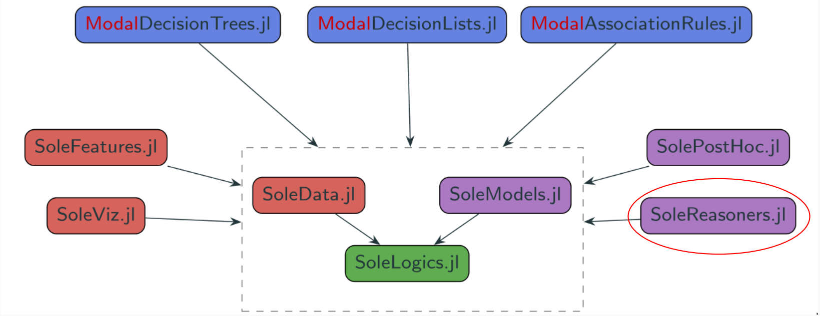

Figure 4: Symbolic Oriented Learning Environment (SOLE) ecosystem.

Tested $\mathrm{FL_{ew}}$-algebras

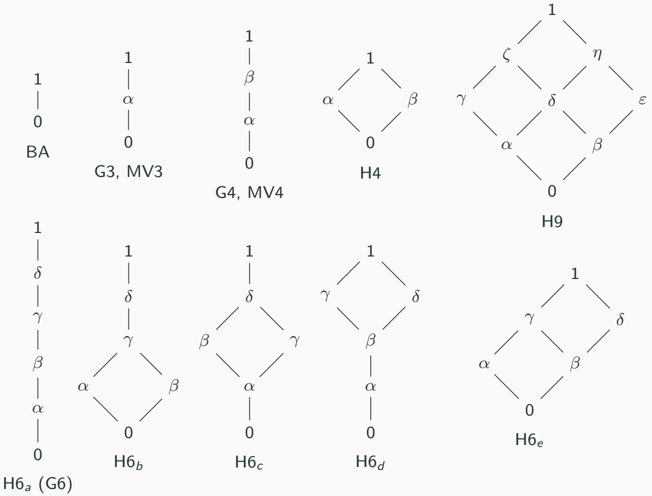

Figure 5: Algebras used in our experiments. G* and MV* differ for the monoid.

Performance results

height	BA	G3	MV3	G4	MV4	H4	$H6_a$	$H6_b$	$H6_c$	$H6_d$	$H6_e$	H9
						α-satisfiability						
1	200	200	200	200	200	200	200	200	200	200	200	176
2	198	199	199	199	199	185	198	195	188	190	181	157
3	191	195	196	194	191	171	190	183	178	180	168	134
4	179	182	181	164	165	140	158	145	139	142	131	112
5	168	165	157	147	136	128	136	126	123	126	111	100
6	153	137	129	115	111	104	111	109	100	101	99	95
						α-validity						
1	200	200	200	200	200	200	200	200	200	200	200	200
2	198	194	194	194	194	194	193	193	193	193	193	182
3	181	183	184	184	184	182	174	175	173	172	172	151
4	174	165	161	142	141	145	129	136	134	131	138	123
5	151	124	115	102	99	108	95	94	91	87	95	101
6	119	95	87	86	80	91	78	79	75	79	78	86

Figure 6: Estimation of how the computation time for *many-valued linear temporal logic with future and past* is influenced by the height of the formula and the type of algebra: how many formulas can be computed within a 60-second timeout.

Performance results

height	BA	G3	MV3	G4	MV4	H4	$H6_a$	$H6_b$	$H6_c$	$H6_d$	$H6_e$	H9
						α-satisfiability						
1	200	200	200	200	200	200	200	195	190	189	179	176
2	198	199	199	197	198	179	189	178	177	172	160	148
3	192	192	192	185	188	159	167	158	152	153	136	124
4	185	171	174	155	158	128	128	127	123	116	115	104
5	168	152	148	134	132	118	113	114	106	103	103	92
6	150	129	130	104	111	94	104	98	99	95	93	83
						α-validity						
1	200	200	200	200	200	200	200	200	200	200	200	200
2	197	194	194	190	193	191	175	175	176	175	181	174
3	180	182	183	163	176	179	147	147	144	145	150	136
4	166	144	137	122	119	135	106	108	103	105	115	103
5	137	109	102	98	93	102	86	85	84	84	85	91
6	116	88	83	88	83	93	75	77	78	76	79	80

Figure 7: Estimation of how the computation time for *many-valued compass logic* is influenced by the height of the formula and the type of algebra: how many formulas can be computed within a 60-second timeout.

Performance results

height	BA	G3	MV3	G4	MV4	H4	$H6_a$	$H6_b$	$H6_c$	$H6_d$	$H6_e$	H9
α-satisfiability												
1	200	200	200	147	147	147	152	152	152	152	152	146
2	198	183	183	117	116	113	117	117	114	116	112	106
3	186	156	156	100	103	97	100	99	99	96	97	91
4	161	142	141	88	85	83	88	84	89	83	80	79
5	141	124	121	89	86	79	82	82	83	82	80	72
6	125	116	112	82	80	77	81	80	79	75	79	73
α-validity												
1	200	200	200	140	140	140	144	144	144	144	144	144
2	197	171	172	92	90	94	88	90	87	87	89	94
3	177	138	139	80	75	78	73	76	70	72	71	77
4	142	104	104	66	66	68	66	66	64	66	66	70
5	114	87	90	64	64	64	60	64	62	63	58	68
6	97	80	83	69	66	67	59	61	58	61	61	65

Figure 8: Estimation of how the computation time for *many-valued Halpern and Shoham's interval temporal logic* is influenced by the height of the formula and the type of algebra: how many formulas can be computed within a 60-second timeout.

Performance results

height	BA	G3	MV3	G4	MV4	H4	$H6_a$	$H6_b$	$H6_c$	$H6_d$	$H6_e$	H9
α-satisfiability												
1	200	146	146	147	147	147	152	152	152	152	152	146
2	194	116	115	115	116	113	116	117	117	115	113	109
3	170	101	104	103	103	99	105	103	103	101	99	97
4	146	97	94	89	93	88	85	89	87	85	81	85
5	121	87	89	92	89	78	88	87	90	81	84	82
6	116	83	81	93	89	78	84	86	86	79	81	80
α-validity												
1	200	142	142	140	140	140	144	144	144	144	144	144
2	189	100	100	93	92	92	89	88	88	89	87	94
3	151	79	81	80	78	79	71	74	70	73	72	78
4	122	71	69	68	65	66	65	66	66	67	66	72
5	101	66	66	66	68	68	61	63	64	62	63	70
6	90	64	65	67	66	67	60	62	62	61	62	68

Figure 9: Estimation of how the computation time for *many-valued linear Lutz and Wolter's modal logic of topological relations* is influenced by the height of the formula and the type of algebra: how many formulas can be computed within a 60-second timeout.

Conclusions

Conclusions

In conclusion:

- We presented a Many-Valued Multi-Modal Logic based on $\mathrm{FL_{ew}}$-algebras
- This logic provides a framework to treat common Temporal and Spatial logics
- We introduced a reasoning system based on analytic tableau
- We provided an implementation for the reasoning system, as well as experimental results

Next steps:

- Tableau system for (not necessary finite) BL-algebras
- Studying applications leveraging the reasoning system, including formula minimization, symbolic learning, and so on...

Thank you for your attention!

Questions?

Propositional rules

$$(true\ \wedge)\ \frac{true(\beta \preceq (\psi \wedge \xi), w, \widetilde{F})}{\begin{array}{c} true(\beta_1 \preceq \psi, w, c(B)) \mid \ldots \mid true(\beta_n \preceq \psi, w, c(B)) \\ true(\gamma_1 \preceq \xi, w, c(B)) \mid \ldots \mid true(\gamma_n \preceq \xi, w, c(B)) \end{array}}$$

where $\beta \succ 0$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta \preceq \beta_j \cdot \gamma_j$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta \preceq \beta'_j \cdot \gamma'_j, \beta'_j \preceq \beta_j$ and $\gamma'_j \preceq \gamma_j$

$$(false\ \wedge)\ \frac{false(\beta \preceq (\psi \wedge \xi), w, \widetilde{F})}{\begin{array}{c} true(\psi \preceq \beta_1, w, c(B)) \mid \ldots \mid true(\psi \preceq \beta_n, w, c(B)) \\ true(\xi \preceq \gamma_1, w, c(B)) \mid \ldots \mid true(\xi \preceq \gamma_n, w, c(B)) \end{array}}$$

where $\beta \succ 0$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta \npreceq \beta_j \cdot \gamma_j$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta \npreceq \beta'_j \cdot \gamma'_j, \beta_j \preceq \beta'_j$ and $\gamma_j \preceq \gamma'_j$

$$(true\ \vee)\ \frac{true((\psi \vee \xi) \preceq \beta, w, \widetilde{F})}{\begin{array}{c} true(\psi \preceq \beta_1, w, c(B)) \mid \ldots \mid true(\psi \preceq \beta_n, w, c(B)) \\ true(\xi \preceq \gamma_1, w, c(B)) \mid \ldots \mid true(\xi \preceq \gamma_n, w, c(B)) \end{array}}$$

where $\beta \prec 1$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta_j \cup \gamma_j \preceq \beta$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta'_j \cup \gamma'_j \preceq \beta, \beta_j \preceq \beta'_j$ and $\gamma_j \preceq \gamma'_j$

$$(false\ \vee)\ \frac{false((\psi \vee \xi) \preceq \beta, w, \widetilde{F})}{\begin{array}{c} true(\beta_1 \preceq \psi, w, c(B)) \mid \ldots \mid true(\beta_n \preceq \psi, w, c(B)) \\ true(\gamma_1 \preceq \xi, w, c(B)) \mid \ldots \mid true(\gamma_n \preceq \xi, w, c(B)) \end{array}}$$

where $\beta \prec 1$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta_j \cup \gamma_j \npreceq \beta$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta'_j \cup \gamma'_j \npreceq \beta, \beta'_j \preceq \beta_j$ and $\gamma'_j \preceq \gamma_j$

$$(true\ \rightarrow)\ \frac{true(\beta \preceq (\psi \rightarrow \xi), w, \widetilde{F})}{\begin{array}{c} true(\psi \preceq \beta_1, w, c(B)) \mid \ldots \mid true(\psi \preceq \beta_n, w, c(B)) \\ true(\gamma_1 \preceq \xi, w, c(B)) \mid \ldots \mid true(\gamma_n \preceq \xi, w, c(B)) \end{array}}$$

where $\beta \succ 0$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta \preceq \beta_j \hookrightarrow \gamma_j$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta \preceq \beta'_j \hookrightarrow \gamma'_j, \beta_j \preceq \beta'_j$ and $\gamma'_j \preceq \gamma_j$

$$(false\ \rightarrow)\ \frac{false(\beta \preceq (\psi \rightarrow \xi), w, \widetilde{F})}{\begin{array}{c} true(\beta_1 \preceq \psi, w, c(B)) \mid \ldots \mid true(\beta_n \preceq \psi, w, c(B)) \\ true(\xi \preceq \gamma_1, w, c(B)) \mid \ldots \mid true(\xi \preceq \gamma_n, w, c(B)) \end{array}}$$

where $\beta \succ 0$, $(\beta_j, \gamma_j) \in \mathcal{A} \times \mathcal{A}$ so that $\beta \npreceq \beta_j \hookrightarrow \gamma_j$ and there is no other $(\beta'_j, \gamma'_j) \in \mathcal{A} \times \mathcal{A}$ such that $\beta \npreceq \beta'_j \hookrightarrow \gamma'_j, \beta'_j \preceq \beta_j$ and $\gamma_j \preceq \gamma'_j$

Figure 10: Propositional rules. Every rule is applied to every branch $B \in \mathcal{B}$ containing the node ν which is being expanded.

Modal rules

$$(\mathit{true}\ \Box)\ \frac{\mathit{true}(\beta \preceq [X_i]\psi, w, \widetilde{F})}{\begin{array}{c}\mathit{true}((\beta \cdot \gamma_1) \preceq \psi, w_1, c(B))\\ \cdots\\ \mathit{true}((\beta \cdot \gamma_m) \preceq \psi, w_m, c(B))\\ \mathit{true}(\beta \preceq [X_i]\psi, w, c(B))\end{array}}$$

where $\gamma_j = \widetilde{R}_i(w, w_j)$, $w_j \in o(c(B))$, $\gamma_j \succ 0$, and $\beta \cdot \gamma_j \succ 0$

$$(\mathit{true}\ \Diamond)\ \frac{\mathit{true}(\langle X_i\rangle\psi \preceq \beta, w, \widetilde{F})}{\begin{array}{c}\mathit{true}(\psi \preceq (\gamma_1 \hookrightarrow \beta), w_1, c(B))\\ \cdots\\ \mathit{true}(\psi \preceq (\gamma_m \hookrightarrow \beta), w_m, c(B))\\ \mathit{true}(\langle X_i\rangle\psi \preceq \beta, w, c(B))\end{array}}$$

where $\gamma_j = \widetilde{R}_i(w, w_j)$, $w_j \in o(c(B))$, $\gamma_j \succ 0$, and $\gamma_j \hookrightarrow \beta \prec 1$

$$(\mathit{false}\ \Box)\ \frac{\mathit{false}(\beta \preceq [X_i]\psi, w, \widetilde{F})}{\mathit{false}((\beta \cdot \gamma_1) \preceq \psi, w_1, c(B')) \mid \ldots \mid \mathit{false}((\beta \cdot \gamma_m) \preceq \psi, w_m, c(B'))}$$

where $\gamma_j = \widetilde{R}_i(w, w_j)$, $w_j \in o(c(B)) \cup n(c(B))$, $\gamma_j \succ 0$, and $\beta \cdot \gamma_j \succ 0$

$$(\mathit{false}\ \Diamond)\ \frac{\mathit{false}(\langle X_i\rangle\psi \preceq \beta, w, \widetilde{F})}{\mathit{false}(\psi \preceq (\gamma_1 \hookrightarrow \beta), w_1, c(B')) \mid \ldots \mid \mathit{false}(\psi \preceq (\gamma_m \hookrightarrow \beta), w_m, c(B'))}$$

where $\gamma_j = \widetilde{R}_i(w, w_j)$, $w_j \in o(c(B)) \cup n(c(B))$, $\gamma_j \succ 0$, and $\gamma_j \hookrightarrow \beta \prec 1$

Figure 11: Modal rules. Every rule is applied to every branch $B \in \mathcal{B}$ containing the node ν which is being expanded. When expanding a node of the type $\mathit{false}(\cdot, \cdot, \widetilde{F})$, a new branch B' is generated for each possible way a new world can be added to the frame $c(B)$. $o(c(B))$ is the set of all existing worlds in the frame $c(B)$, while $n(c(B))$ is the new world added to $c(B)$ to obtain $c(B')$.

Reverse rules

$$(true \succeq)\ \frac{true(\beta \preceq \psi, w, \widetilde{F})}{false(\psi \preceq \gamma, w, c(B))}$$

where $\beta \succ 0$ and γ is any maximal element not above β, i.e., $\gamma \not\succeq \beta$

$$(false \succeq)\ \frac{false(\beta \preceq \psi, w, \widetilde{F})}{true(\psi \preceq \gamma_j, w, c(B)) \mid \ldots \mid true(\psi \preceq \gamma_n, w, c(B))}$$

where $\beta \succ 0$ and $\gamma_1, \ldots, \gamma_n$ are all maximal elements not above β, i.e., $\gamma_1, \ldots, \gamma_n \not\succeq \beta$

$$(true \preceq)\ \frac{true(\psi \preceq \beta, w, \widetilde{F})}{false(\gamma \preceq \psi, w, c(B))}$$

where $\beta \prec 1$ and γ is any minimal element not below β, i.e., $\gamma \not\preceq \beta$

$$(false \preceq)\ \frac{false(\psi \preceq \beta, w, \widetilde{F})}{true(\gamma_j \preceq \psi, w, c(B)) \mid \ldots \mid true(\gamma_j \preceq \psi, w, c(B))}$$

where $\beta \prec 1$ and $\gamma_1, \ldots, \gamma_n$ are all minimal elements not below β, i.e., $\gamma_1, \ldots, \gamma_n \not\preceq \beta$

Figure 12: Reverse rules. Every rule is applied to every branch $B \in \mathcal{B}$ containing the node ν which is being expanded.

Branch closing rules

$$(\text{✗}1)\ \frac{\mathit{true}(\beta \preceq \gamma, w, \widetilde{F})}{\text{✗}}\ \text{where } \beta \not\preceq \gamma \qquad (\text{✗}2)\ \frac{\mathit{false}(\beta \preceq \gamma, w, \widetilde{F})}{\text{✗}}\ \text{where } \beta \succ 0,\ \gamma \prec 1, \text{ and } \beta \preceq \gamma \qquad (\text{✗}3)\ \frac{\mathit{false}(0 \preceq \psi, w, \widetilde{F})}{\text{✗}}$$

$$(\text{✗}4)\ \frac{\mathit{false}(\psi \preceq 1, w, \widetilde{F})}{\text{✗}} \qquad (\text{✗}5)\ \frac{\begin{array}{c}\mathit{true}(\gamma \preceq \psi, w, \widetilde{F})\\ \mathit{false}(\beta \preceq \psi, w, \widetilde{F}')\end{array}}{\text{✗}}\ \text{where } \beta \preceq \gamma \qquad (\text{✗}6)\ \frac{Q(\cdot, \cdot, \widetilde{F})}{\text{✗}}\ \text{where } \widetilde{F} \text{ is inconsistent}$$

Figure 13: Branch closing rules. Every branch $B \in \mathcal{B}$ containing the node ν which is being expanded must be closed.

An alternative: translating to F.O.

One could also translate the satisfiability problem for many-valued modal logic to a two-sort first-order logic and leverage an SMT solver (e.g., $Z3$)

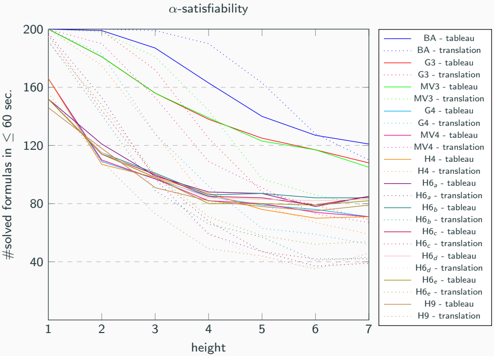

Figure 14: Tableau (strict lines) vs translation (dotted lines) performance for solving α-satisfiability for *many-valued Halpern and Shoham's interval temporal logic*: how many formulas can be computed within a 60-second timeout.