A Matrix Characterization of Validity for Multimodal Logics

SAMULI HEILALA AND BRIGITTE PIENTKA

ABSTRACT. We present a new matrix characterization of validity for a family of propositional multimodal logics with interacting modalities. Unlike previous matrix characterizations for modal logics, which could only cope with a few well-behaved unimodal logics, our formulation is not based on prefixed tableaus in the style of Fitting, but has a more direct relationship with the semantics of the logics and their defining frame conditions. The resulting formalism uniformly and elegantly characterizes all 15 basic unimodal logics, as well as a number of multimodal logics with interacting modalities.

Keywords: Matrix characterization, multimodal logic.

1 Introduction

Reasoning in modal and multimodal logics has been formalized using all manner of proof theoretic apparatus, from the early Hilbert-style axiomatizations proposed and studied by the likes of Lewis and Prior to the plethora of natural deduction, tableau, and sequent calculus formalisms that have since been explored. A slightly less-known formalism is the *matrix* or *connection-based characterization*, pioneered for first-order logic by Prawitz [16] and developed further by Bibel [4] and Andrews [2]. Wallen [18, 19] subsequently generalized and extended the matrix characterization to unimodal and intuitionistic logics, while similar extensions to multiplicative linear logic and multiplicative exponential linear logic were carried out by Kreitz et al. [12] and Kreitz and Mantel [13], respectively.

One of the motivations for developing matrix characterizations is automated theorem proving, since a matrix characterization of validity for a logic generally induces a very compact proof space for it that theorem provers can efficiently explore. Reasoning in most natural deduction, tableau, and sequent calculus systems involves taking formulas apart from the outside in, one connective at a time, resulting in nondeterminism during proof search and redundancies in the finished proofs, arising from irrelevance and proof permutatibility [19]. The idea behind matrix characterizations, on the other hand, is to analyze the entire structure of a formula down to its atomic subformulas, extract *paths* through it, and find *connections* that simultaneously *span* the paths. As a result, the search is in some sense global, rather than local to a particular connective or subformula. Regardless of the origins

Paper submitted to Advances in Modal Logic 2008

of matrix characterizations in proof search, they are beautiful in their own right, as they are able to collectively express validity in a myriad of different logics in a very uniform and modular way.

Matrix characterizations have been found for a number of logics, and Wallen's conjecture is that matrix methods can be developed for any logic with the same primary properties as classical logic [19]. Wallen himself has carried out the development for the unimodal logics **K**, **K4**, **D**, **D4**, **T**, **S4**, and **S5**. The natural first step to completing his work is to extend the characterization to the remaining 8 basic unimodal logics (see for instance Chellas [7]), including symmetric and euclidean logics, and then to logics with multiple interacting modalities. However, the roots of Wallen's work are in Fitting's prefixed tableau systems [8, 9, 14], and because of the way in which frame conditions such as symmetry are encoded by prefixes in these tableau systems, it is not clear how to adapt Wallen's formulation to symmetric logics, for instance. The problem is compounded when dealing with multimodal logics with modalities interacting in particular ways, where prefixed tableaus with the flavour of Fitting's can run into difficulties of their own (Baldoni [3] provides some examples).

Our solution is to develop a matrix characterization of validity for modal and multimodal logics that, rather than being based on Fitting's prefixed tableaus, has a basis closer to the semantics of the logic and the frame conditions that define it. Instead of manipulating prefixes that stand for notional worlds in a putative countermodel, with the accessibility relation between worlds encoded by the structures of the prefixes, we will manipulate atomic world names that represent worlds directly, separately maintaining an accessibility relation on the world names. Although this semantic shift costs us efficiency in proof search, we gain the ability to uniformly and elegantly characterize validity in all 15 basic unimodal logics, as well as a number of multimodal logics with interacting modalities, something not easily done using Wallen's characterization.

In this paper, we will depend largely on informal discussion and examples to describe our characterization, stating correctness results but omitting their proofs. For details and complete proofs, see [11]. The rest of the paper is structured as follows. In Section 2, we describe our matrix characterization for the 15 basic unimodal logics, generalizing our approach to multimodal logics with interacting modalities in Section 3. It is a testament to the modularity of our approach that almost all of the technical apparatus we develop for unimodal logics carries over to multimodal logics unchanged. Related work is discussed in Section 4, and we conclude with future work in Section 5.

2 Unimodal Logics

2.1 Syntax and Semantics

The syntax and semantics of propositional unimodal logics are as usual (see for instance Blackburn et al. [5] or Chellas [7]), but worth reviewing briefly. *Formulas* are constructed from propositional letters p_1, p_2, $\dots$, the binary

L	**L's frame conditions**	**Other names for L**
K	none	
KD	seriality	**D**
KT	reflexivity	**T**
KB	symmetry	
K4	transitivity	
K5	euclideanness	
KDB	seriality, symmetry	
KD4	seriality, transitivity	**D4**
KD5	seriality, euclideanness	
KTB	reflexivity, symmetry	**B**
KT4	reflexivity, transitivity	**S4**
KB4	symmetry, transitivity	
K45	transitivity, euclideanness	
KD45	seriality, transitivity, euclideanness	
KTB4	reflexivity, symmetry, transitivity	**S5**, **KT5**

Table 1. The 15 basic unimodal logics and their frame conditions.

connectives $\wedge$, $\vee$, and $\supset$, and the unary connectives $\neg$, $\Box$, and $\Diamond$. A *frame* is a pair (W, R), where W is a nonempty set whose elements are called *worlds*, and R is a binary relation on W, called the *accessibility relation*. A *model* is a triple (W, R, V), where (W, R) is a frame and V is a function from propositional letters to subsets of W, called the *valuation*. The *truth relation* $\Vdash$ relates a model $M = (W, R, V)$ and worlds $w \in W$ to formulas as follows.

1. For every propositional letter p, $M, w \Vdash p$ iff $w \in V(p)$.
2. $M, w \Vdash \neg A$ iff $M, w \not\Vdash A$.
3. $M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$.
4. $M, w \Vdash A \vee B$ iff $M, w \Vdash A$ or $M, w \Vdash B$.
5. $M, w \Vdash A \supset B$ iff $M, w \Vdash A$ implies that $M, w \Vdash B$.
6. $M, w \Vdash \Box A$ iff for every $x \in W$ such that $(w, x) \in R$, $M, x \Vdash A$.
7. $M, w \Vdash \Diamond A$ iff there is some $x \in W$ such that $(w, x) \in R$ and $M, x \Vdash A$.

A formula A is said to be *valid* in a frame (W, R) if it is true at every world in every model with (W, R) as an underlying frame.

A frame (W, R) or a model (W, R, V) is said to be *serial*, *reflexive*, *symmetric*, *transitive*, or *euclidean* if its accessibility relation R has the named property. Recall that a relation R over W is serial if for every $w \in W$, there is an $x \in w$ such that $(w, x) \in R$, and euclidean if for every $w, x, y \in W$,

$(w, x) \in R$ and $(w, y) \in R$ together imply that $(x, y) \in R$. Different modal logics are characterized by different *frame conditions*. Exactly 15 distinct modal logics can be obtained by combining the five frame conditions mentioned above in various ways (see for instance Chellas [7]), and in Table 1, we assign names to these 15 basic unimodal logics and give their frame conditions. If **L** is the name of a logic, then a formula A is said to be ***L**-valid* if A is valid in every frame whose accessibility relation satisfies **L**'s frame conditions.

The form of question that our matrix characterization of validity for unimodal logics seeks to answer is, "given a logic **L** and a formula A, is A **L**-valid?" As in most tableau and sequent calculus systems, we will answer this question by attempting to construct a countermodel for A that satisfies **L**'s frame conditions. If this construction results in a necessary contradiction, then A is **L**-valid. In tableau and sequent calculus systems, this countermodel construction is plagued by nondeterminism throughout proof search. One of the advantages of matrix proof methods is that some of this nondeterminism disappears, since part of the countermodel construction in matrix proof methods occurs in a purely deterministic initial phase in which the query formula is analyzed down to its atomic subformulas, giving us a rough idea of potential countermodels before we even perform any kind of search.

The main result of this paper is the correctness of our matrix characterization. We state its soundness and completeness results here, and subsequently guide the reader through them, providing definitions, explanations, intuition, and examples for the technical terms used below.

THEOREM 1 (Soundness). *If there is a multiplicity μ for a signed formula $X = (0, A)$ and an **L**-admissible world realization σ for the indexed signed formula X^μ such that all atomic paths through X^μ are σ-complementary, then A is **L**-valid.*

THEOREM 2 (Completeness). *If a formula A is **L**-valid, then there is a multiplicity μ for the signed formula $X = (0, A)$ and an **L**-admissible world realization σ for the indexed signed formula X^μ such that all atomic paths through X^μ are σ-complementary.*

For brevity, we omit the full proofs here (see [11] for details), but it is worth emphasizing that the matrix characterization is closely related to tableau systems, and correctness is consequently proved in much the same way as it conventionally is for tableaus. Namely, soundness follows from a contrapositive argument, using a lemma of preservation of ***L**-satisfiability*, while completeness follows from a *systematic construction* and a notion of ***L**-Hintikka sets* (see [8] for outlines of the corresponding proofs for Fitting's tableau system).

2.2 Formula Trees

An important proof theoretic extension of a formula is a *signed formula*, a pair (s, A), where s, called the *sign* of the signed formula, is either 0

α	α_1	α_2
$(1, A \wedge B)$	$(1, A)$	$(1, B)$
$(0, A \vee B)$	$(0, A)$	$(0, B)$
$(0, A \supset B)$	$(1, A)$	$(0, B)$
$(0, \neg A)$	$(1, A)$	$(1, A)$
$(1, \neg A)$	$(0, A)$	$(0, A)$

β	β_1	β_2
$(u, 0, A \wedge B)$	$(u, 0, A)$	$(u, 0, B)$
$(u, 1, A \vee B)$	$(u, 1, A)$	$(u, 1, B)$
$(u, 1, A \supset B)$	$(u, 0, A)$	$(u, 1, B)$

ν	ν_1
$(1, \Box A)$	$(1, A)$
$(0, \Diamond A)$	$(0, A)$

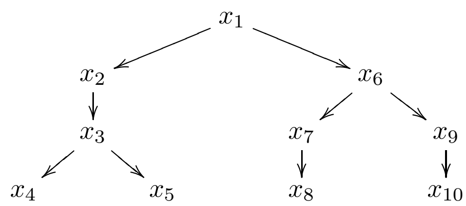
π	π_1
$(0, \Box A)$	$(0, A)$
$(1, \Diamond A)$	$(1, A)$

Figure 1. Types and components of signed formulas.

or 1, and A is a formula. Every nonatomic signed formula has a *type* of either α, β, ν, or π, determined by its sign and its top-level connective, as well as one or more *components*. The types and components of nonatomic signed formulas are defined as shown in Figure 1. For instance, the signed formula $(0, p_1 \wedge (p_2 \vee p_3))$ has type β, and its two components are $(0, p_1)$ and $(0, p_2 \vee p_3)$. Intuitively, a signed formula denotes the truth (if its sign is 1) or falsehood (if its sign is 0) of a formula, and a signed formula of type α "holds" (i.e. is true or false, depending on its sign) if both of its components hold, while one of type β holds if either one of its components holds. Within some modal context (i.e. at some world), a signed formula of type ν or π holds if its component holds at all accessible worlds or at some accessible world, respectively. Note that in practice, signed formulas of the form $(s, \neg A)$ would be treated as having only a single component, say α_1, but in the theory, they can be uniformly handled in the same way as any other signed formula of type α.

A *formula tree* for a signed formula (s, A) is a representation of A as a tree of names, called *positions*, each position corresponding to a distinct subformula occurrence of A. The subformula corresponding to a position x is called its *label*—denoted $\mathrm{lab}(x)$—and a position is called *atomic* or *nonatomic* when its label is atomic or nonatomic, respectively. Each position x of a formula tree for a signed formula (s, A) is also associated with a *sign*—denoted $\mathrm{sgn}(x)$—of s if the subformula occurrence corresponding to x occurs positively in A, and $s + 1 \bmod 2$ otherwise. The *tree ordering* $\ll$ is the partial ordering induced by the formula tree, that is, $x_i \ll x_j$ if position x_i strictly dominates position x_j. As an example, the formula tree for the signed formula $X = (0, \Diamond(p_1 \supset p_2) \supset (\Box p_1 \supset \Diamond p_2))$ is shown in Figure 2.

In addition to labels and signs, each position x of a formula tree has a *primary type*—denoted $\mathrm{ptp}(x)$—and a *secondary type*—denoted $\mathrm{stp}(x)$. The primary type of a nonatomic position x is simply the type of the signed formula $(\mathrm{sgn}(x), \mathrm{lab}(x))$. Atomic positions consequently have no primary type. The secondary type of a position x is determined by the primary type of its parent in the formula tree, if it has one, and its relationship with its



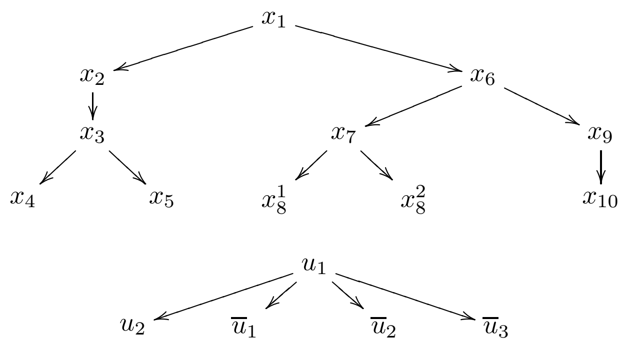
x	$\mathrm{lab}(x)$	$\mathrm{sgn}(x)$	$\mathrm{ptp}(x)$	$\mathrm{stp}(x)$
x_1	$\Diamond(p_1 \supset p_2) \supset (\Box p_1 \supset \Diamond p_2)$	0	α	π_1
x_2	$\Diamond(p_1 \supset p_2)$	1	π	α_1
x_3	$p_1 \supset p_2$	1	β	π_1
x_4	p_1	0		β_1
x_5	p_2	1		β_2
x_6	$\Box p_1 \supset \Diamond p_2$	0	α	α_2
x_7	$\Box p_1$	1	ν	α_1
x_8	p_1	1		ν_1
x_9	$\Diamond p_2$	0	ν	α_2
x_{10}	p_2	0		ν_1

Figure 2. The formula tree for the signed formula X.

parent. If x's parent has primary type α (resp. β), then x has secondary type α_1 or α_2 (resp. β_1 or β_2), depending on whether it is the first or second child of its parent. If x's parent has primary type ν (resp. π), then x has secondary type ν_1 (resp. π_1). The secondary type of the root position is defined to be π_1 for technical reasons. Figure 2 also shows the types of the positions of our example.

Notice that all of the pieces of information associated with positions are obtained deterministically. Given a signed formula, there is a unique formula tree for it (modulo renaming of positions), with corresponding label, sign, and type functions. The intuition is that every position of a formula tree corresponds to a subformula occurrence of the query formula, and in constructing a putative countermodel for it, the signed subformulas corresponding to positions are required to hold for the query formula to be falsifiable. The types of positions determine, for instance, whether their corresponding signed subformula occurrences are *conjunctively* or *disjunctively* related, where conjunctively related signed subformula occurrences must all hold at once, while of a set of disjunctively related signed subformula occurrences, only one must hold. Formally, two positions x and x' of a formula tree are said to be conjunctively related, or *α-related*, if their $\ll$-greatest common ancestor has primary type other than β. Otherwise, they are said to be disjunctively related, or *β-related*.

Modally, positions of secondary type ν_1 have necessary force, meaning



y	lab(y)	sgn(y)	ptp(y)	stp(y)	wn(y)
x_1	$\Diamond(p_1 \supset p_2) \supset (\Box p_1 \supset \Diamond p_2)$	0	α	π_1	u_1
x_2	$\Diamond(p_1 \supset p_2)$	1	π	α_1	u_1
x_3	$p_1 \supset p_2$	1	β	π_1	u_2
x_4	p_1	0		β_1	u_2
x_5	p_2	1		β_2	u_2
x_6	$\Box p_1 \supset \Diamond p_2$	0	α	α_2	u_1
x_7	$\Box p_1$	1	ν	α_1	u_1
x_8^1	p_1	1		ν_1	$\overline{u}_1$
x_8^2	p_1	1		ν_1	$\overline{u}_2$
x_9	$\Diamond p_2$	0	ν	α_2	u_1
x_{10}	p_2	0		ν_1	$\overline{u}_3$

Figure 3. The indexed formula tree and formula frame for the indexed signed formula X^μ.

that they can be instantiated at multiple worlds of a purported countermodel. They can be forced to hold at any world, in fact, accessible from the world at which their parents of primary type ν hold. This is encoded by a *multiplicity* for a signed formula $X = (s, A)$, a function from the set of positions of secondary type ν_1 in X to the natural numbers. The intuition is that $\mu(x)$ stipulates the number of different ways in which the signed subformula occurrence corresponding to position x can be used (i.e. at how many different worlds). An *indexed signed formula* is an expression X^μ, where X is a signed formula and μ is a multiplicity for X. An *indexed formula tree* for an indexed signed formula X^μ is an extension of the formula tree for X whose nodes are *indexed positions*. The indexed formula tree for X^μ is simply the formula tree for X, with every subtree rooted at a position of secondary type ν_1 replicated as many times as its multiplicity requires. Indexed positions differ from regular positions only in that they feature superscript indices to differentiate between several instances derived from the same position. They inherit the signs, labels, and types of the positions from which they are derived. As an example, the indexed formula tree for the indexed signed formula X^μ is shown in Figure 3, where $X = (0, \Diamond(p_1 \supset p_2) \supset (\Box p_1 \supset \Diamond p_2))$, $\mu(x_8) = 2$, and $\mu(x_{10}) = 1$. (x_8 and x_{10} are the only positions of X of secondary type ν_1.)

Finally, each indexed position y of an indexed formula tree is also associated with a *world name*—denoted $\mathrm{wn}(y)$. World names come in two disjoint classes, *constant world names* u_1, u_2, ... and *variable world names* $\overline{u}_1$, $\overline{u}_2$, ... A function from indexed positions to world names is a valid world name assignment if

1. every indexed position of secondary type ν_1 is mapped to a unique variable world name,
2. every indexed position of secondary type π_1 is mapped to a unique constant world name, and
3. every indexed position of secondary type other than ν_1 and π_1 is mapped to the same world name as its parent.

A world name function for a formula tree induces a *formula frame* (also a tree) whose worlds are the world names of the indexed positions in the indexed formula tree. The formula frame for the indexed signed formula X^μ of our running example and the corresponding world name function are also shown in Figure 3.

2.3 World Realizations

Indexed signed formula trees encode all the structural properties of a query formula, while formula frames encode the minimal accessibility relations required for putative countermodels to actually be permissible as countermodels, that is, for the forced holding of the signed subformulas corresponding to indexed positions to be coherent. However, variable world names in formula frames are merely placeholders for concrete worlds, as they are world names introduced by indexed positions of necessary force, standing in lieu of any world accessible from the world of the parent position. To obtain a concrete countermodel, variable world names must either themselves be made concrete or mapped to other concrete world names.

This mapping is formalized as a *world realization*, a function σ from the set of world names of an indexed signed formula to itself, with the property that every constant world name and every variable world name that is the image, under σ, of another world name is mapped to itself. For instance, using our running example, σ_1 such that $\sigma_1(\overline{u}_1) = \sigma_1(\overline{u}_2) = \sigma_1(\overline{u}_3) = \sigma_1(u_2) = u_2$ is a world realization. So is σ_2 such that $\sigma_2(\overline{u}_1) = \sigma_2(\overline{u}_2) = \overline{u}_2$ and $\sigma_2(\overline{u}_3) = \sigma_2(u_2) = u_2$. A world realization is called *strict* if it maps every variable world name to a constant world name, so σ_1 is a strict world realization, while σ_2 is not. We will use σ_1 and σ_2 as concrete world realizations with our running example.

If σ is a world realization for an indexed signed formula X^μ, then the *modal ordering* $\sqsubset$ is a partial ordering on the indexed positions of X^μ such that $y \sqsubset y'$ if $\sigma(\mathrm{wn}(y')) = \mathrm{wn}(y)$ and $\mathrm{wn}(y') \neq \mathrm{wn}(y)$. For a world realization to be semantically sound, letting a variable world name represent some other concrete world name is only meaningful if the concrete world name is known to exist. In other words, the modal ordering imposes an order in which indexed positions must be taken apart during countermodel construction for a world realization to be meaningful.

2.4 Paths and Matrices

A central concept in Theorems 1 and 2 is the idea of atomic paths. In general, a *path* through an indexed signed formula X^μ is a subset of its positions, and an *atomic path* is a particular kind of path. While the technical definitions of paths and atomic paths are slightly cumbersome, the matrix characterization gets its name from an intuitive and appealing method of displaying indexed signed formulas in such a way as to visually reveal their atomic paths. Formally, the *matrix* of an indexed position y of an indexed signed formula X^μ is defined inductively as follows.

1. If y is atomic, then the matrix of y is y itself.
2. If y has primary type α, then the matrix of y is a 1×2 array with the matrices of the two components of y as the entries in the two columns.
3. If y has primary type β, then the matrix of y is a 2×1 array with the matrices of the two components of y as the entries in the two rows.
4. Otherwise, the matrix of y is a $1 \times n$ array with the matrices of the components of y as the entries in the columns.

The matrix of an indexed signed formula X^μ is the matrix of the root indexed position of X^μ, and the atomic paths through X^μ are then the horizontal paths through the matrix of X^μ. As an example, the matrix for X^μ from our running example is

$$\left[\begin{bmatrix} x_4 \\ x_5 \end{bmatrix} \quad \left[\begin{bmatrix} x_8^1 & x_8^2 \end{bmatrix} \quad x_{10} \right] \right]$$

For clarity, we have omitted brackets around 1×1 arrays. In this case, the atomic paths through X^μ are $\{x_4, x_8^1, x_8^2, x_{10}\}$ and $\{x_5, x_8^1, x_8^2, x_{10}\}$. It is worth emphasizing at this point that matrices encode only the conjunctivity or disjunctivity between indexed positions, they say nothing about modal relationships amongst their world names. Modal information is always separately maintained by the formula frame and the world name mapping.

If σ is a world realization for an indexed signed formula X^μ, then an atomic path ϕ through X^μ is said to be *σ-complementary* if ϕ contains two indexed positions y and y' such that $\sigma(\mathrm{wn}(y)) = \sigma(\mathrm{wn}(y'))$, $\mathrm{sgn}(y) \neq \mathrm{sgn}(y')$, and $\mathrm{lab}(y) = \mathrm{lab}(y')$. In our running example, with σ_1 and σ_2 defined as previously, $\{x_4, x_8^1, x_8^2, x_{10}\}$ is σ_1-complementary by virtue of the indexed positions x_4 and either x_8^1 or x_8^2, while $\{x_5, x_8^1, x_8^2, x_{10}\}$ is σ_1-complementary by virtue of the indexed positions x_5 and x_{10}. On the other hand, $\{x_4, x_8^1, x_8^2, x_{10}\}$ is not σ_2-complementary. Pairs of positions that are complementary are sometimes called *connections*, and a set of connections is said to *span* an indexed signed formula X^μ if every atomic path through X^μ contains at least one connection.

In our example, our attempt to build a countermodel for $\Diamond(p_1 \supset p_2) \supset (\Box p_1 \supset \Diamond p_2)$ involves reasoning that for this formula to be false at some world named by u_1, $\Diamond(p_1 \supset p_2)$ must be true and $\Box p_1 \supset \Diamond p_2$ must be false at u_1. For $\Diamond(p_1 \supset p_2)$ to be true at u_1, there must be some world named by u_2 and accessible from u_1 at which $p_1 \supset p_2$ is true, which in turn requires p_1 to be false or p_2 to be true at u_2. In either case, for $\Box p_1 \supset \Diamond p_2$ to be false at u_1, $\Box p_1$ must be true and $\Diamond p_2$ must be false at u_1, so at any worlds named by $\overline{u}_1$ and $\overline{u}_2$ and accessible from u_1, p_1 must be true, and at any world named by $\overline{u}_3$ and accessible from u_1, p_2 must be false. According to the world realization σ, we will choose to let p_1 be true at $\sigma(\overline{u}_1) = u_2$ and $\sigma(\overline{u}_2) = u_2$, and we will let p_2 be false at $\sigma(\overline{u}_3) = u_2$. So regardless of whether p_1 is false or p_2 is true at u_2, we obtain a contradiction, as either p_1 is both true and false or p_2 is both true and false at the world named by u_2. Observe that this argument remains valid if we let $\mu(x_8) = 1$, eliminating position x_8^2 and world name $\overline{u}_2$.

Notice how this reasoning is partially encoded in the indexed formula tree and the matrix for X^μ. The disjunctive relationship between positions x_4 and x_5 stipulates that only one of the corresponding signed subformulas needs to hold. By requiring that all paths through X^μ be σ-complementary, we ensure that regardless of which disjunctive set of positions holds, a contradiction is derived, from which the validity of the root position's label might follow. Notice also, however, that the required accessibilities between worlds have not yet been verified. The "accessibility template" described by the formula frame must be used to validate a potential spanning set of connections, which brings us to the final and most complex technical definition in Theorems 1 and 2, namely, the ***L**-admissiblity* of world realizations.

2.5 L-Admissibility

Up to now, we have not made any mention of individual logics, and the reader will be wondering how a formula such as $\Diamond\Diamond p_1 \supset \Diamond p_1$ can be characterized as **K4**-valid but **K**-invalid if, in both cases, the indexed formula trees and formula frames are identical. The key is that the world realization that makes all paths through the query formula complementary must be admissible for the logic at hand, and the conditions of admissibility differ for each logic. In particular, a world realization σ for an indexed signed formula X^μ with formula frame (V, Q) is said to be ***L**-admissible* if it has the following properties.

1. If **L** is not serial, then σ is strict. Recall that a world realization is strict if it maps all variable world names to constant world names.

2. The *reduction ordering* $\lhd = (\ll \cup \sqsubset)^+$, that is, the transitive closure of the tree and modal orderings, is irreflexive.

3. For all indexed positions y and y' of X^μ, $\sigma(\mathrm{wn}(y')) = \mathrm{wn}(y)$ implies that
   (a) y and y' are α-related, and
   (b) $(\sigma(\mathrm{par}(\mathrm{wn}(y'))), \sigma(\mathrm{wn}(y))) \in \mathrm{clo}(\mathbf{L}, Q^*)$, where $\mathrm{par}(\mathrm{wn}(y'))$ is the parent of $\mathrm{wn}(y')$ in the formula frame,
   $$\begin{aligned} Q^* = \{(\sigma(\mathrm{wn}(z)), \sigma(\mathrm{wn}(z'))) :& \\ &(\mathrm{wn}(z), \mathrm{wn}(z')) \in Q, \\ &z, z \lhd y', \text{ and } z \text{ and } z' \text{ are } \alpha\text{-related to } y'.\} \end{aligned}$$
   and $\mathrm{clo}(\mathbf{L}, Q^*)$ is the $\mathbf{L}$-closure of the binary relation Q^*. This is the smallest binary relation containing Q^* and satisfying $\mathbf{L}$'s frame conditions, *with the exception of seriality.*

The first condition states that a world realization may only be non-strict if $\mathbf{L}$ is serial. A non-strict world realization can effectively instantiate variable world names to be new concrete world names. Intuitively, this corresponds to postulating the existence of some successor to a world containing a signed formula of necessary force, then requiring its component to hold in the world just assumed to exist. The second condition states that it must be possible to order the indexed positions of X^μ in such a way that they respect both the structure of the formula and the modal requirement that every concrete world is known to exist when it is used. The third definition looks daunting, but simply states that when a signed formula corresponding to a position is forced to hold at some world, then that world is an $\mathbf{L}$-successor of the parent position in the countermodel constructed so far. In other words, when a concrete world is used, it is used in a way that respects $\mathbf{L}$'s frame conditions.

3 Multimodal Logics

Multimodal formulas are a generalization of unimodal formulas in which the modal connectives $\Box$ and $\Diamond$ are replaced by any number n of pairs of connectives $\Box_1$, $\Diamond_1$, $\Box_2$, $\Diamond_2$, ..., and $\Box_n$, $\Diamond_n$. A multimodal frame is a tuple $(W, R_1, R_2, \ldots, R_n)$, where W is a nonempty set of worlds, and every R_i is a binary accessibility relation on W. A multimodal model is a tuple $(W, R_1, R_2, \ldots, R_n, V)$, where $(W, R_1, R_2, \ldots, R_n)$ is a frame and V is, as before, a valuation from propositional letters to subsets of W. The multimodal truth relation is the usual extension of the unimodal one, with the modal cases defined as follows.

1. $M, w \Vdash \Box_i A$ iff for every $x \in W$ such that $(w, x) \in R_i$, $M, x \Vdash A$.
2. $M, w \Vdash \Diamond_i A$ iff there is some $x \in W$ such that $(w, x) \in R_i$ and $M, x \Vdash A$.

Every accessibility relation R_i can be individually serial, reflexive, symmetric, transitive, or euclidean, but multimodal logics are most interesting when their modalities interact. We consider the following interactions between relations.

1. R_i and R_j are said to be *mutually symmetric* if for every $w, x \in W$, $(w, x) \in R_i$ implies that $(x, w) \in R_j$,

2. R_i, R_j, and R_k are said to be *mutually transitive* if for every $w, x, y \in W$, $(w, x) \in R_i$ and $(x, y) \in R_j$ together imply that $(w, y) \in R_k$, and

3. R_i, R_j, and R_k are said to be *mutually euclidean* if for every $w, x, y \in W$, $(w, x) \in R_i$ and $(w, y) \in R_j$ together imply that $(x, y) \in R_k$.

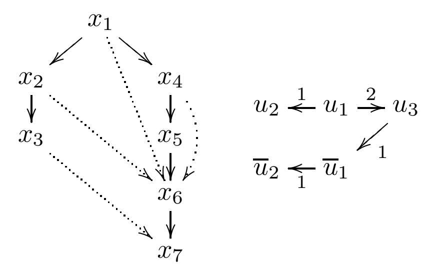
A wide family of multimodal logics can be defined in this way. For instance, the basic temporal language [5] has two modalities R_1 and R_2. In a particular interpretation of temporal logic, both relations are mutually symmetric in both directions, i.e. for every $w, x \in W$, $(w, x) \in R_1$ implies that $(x, w) \in R_2$ and vice versa. In addition, both R_1 and R_2 are serial and transitive. Rather than assigning names to multimodal logics, we will simply associate logics with their frame conditions. A multimodal formula A is again said to be **L**-valid if it is true at all worlds in all models whose underlying frames satisfy **L**'s frame conditions. (Here, **L** simply represents any collection of frame conditions.)

Thanks to the modularity of our approach, very few modifications need to be made to the definitions presented in the previous section to make them applicable to multimodal logics. Indeed, the multimodal soundness and completeness results should look very familiar:

THEOREM 3 (Soundness). *If there is a multiplicity μ for a signed formula $X = (0, A)$ and an **L**-admissible world realization σ for the indexed signed formula X^μ such that all atomic paths through X^μ are σ-complementary, then A is **L**-valid.*

THEOREM 4 (Completeness). *If a formula A is **L**-valid, then there is a multiplicity μ for the signed formula $X = (0, A)$ and an **L**-admissible world realization σ for the indexed signed formula X^μ such that all atomic paths through X^μ are σ-complementary.*

In fact, the results are identical. The only differences are in the meanings of **L**-validity, which we have already mentioned, and **L**-admissibility for world realizations. Even in the latter case, the definition of **L**-admissibility does not change from the unimodal case, since the properties of the logic are encoded in world realizations by means of closures. We will demonstrate the uniformity of our approach by working through a small example, namely, showing the validity of the temporal formula $\Diamond_1 p_1 \supset \Box_2 \Diamond_1 \Diamond_1 p_1$. In temporal terms, this formula states that if p_1 was true in the past, then at every point in the future, there will have been a point in the past at which p_1 was true in the past. Since we are now dealing with multiple modalities, we will decorate types ν, ν_1, π, and π_1 with indices to differentiate which modality they refer to, viz. ν_i, $\nu_{i,1}$, π_i, and $\pi_{i,1}$.



y	lab(y)	sgn(y)	ptp(y)	stp(y)	wn(y)
x_1	$\Diamond_1 p_1 \supset \Box_2 \Diamond_1 \Diamond_1 p_1$	0	α	π_1	u_1
x_2	$\Diamond_1 p_1$	1	π_1	α_1	u_1
x_3	p_1	1		$\pi_{1,1}$	u_2
x_4	$\Box_2 \Diamond_1 \Diamond_1 p_1$	0	π_2	α_2	u_1
x_5	$\Diamond_1 \Diamond_1 p_1$	0	ν_1	$\pi_{2,1}$	u_3
x_6	$\Diamond_1 p_1$	0	ν_1	$\nu_{1,1}$	$\overline{u}_1$
x_7	p_1	0		$\nu_{1,1}$	$\overline{u}_2$

Figure 4. The indexed formula tree and formula frame for the indexed signed formula X^μ.

Let $X = (0, \Diamond_1 p_1 \supset \Box_2 \Diamond_1 \Diamond_1 p_1)$ and let $\mu(x) = 1$ for each position x of X of secondary type $\nu_{i,1}$. Then the indexed formula tree and formula frame for X^μ are shown in Figure 4, and the matrix for X^μ is

$$\begin{bmatrix} x_3 & x_7 \end{bmatrix}$$

Notice that the formula frame shown in Figure 4 is now also multimodal, and we have decorated its edges to indicate which modality each one represents. Let σ be a world realization such that $\sigma(\overline{u}_1) = u_1$ and $\sigma(\overline{u}_2) = u_2$. The modal ordering $\sqsubset$ induced by σ on the indexed positions of X^μ is shown as dotted arcs in the indexed formula tree in Figure 4. Now the single path $\{x_3, x_7\}$ through X^μ is σ-complementary, since $\sigma(\mathrm{wn}(x_3)) = u_2 = \sigma(\mathrm{wn}(x_7))$, and since σ is **L**-admissible (the three conditions of **L**-admissibility are easy to verify), $\Diamond_1 p_1 \supset \Box_2 \Diamond_1 \Diamond_1 p_1$ is **L**-valid.

4 Related Work

Wallen's work [18], based on the prefixed modal tableaus of Fitting [8, 9], is the only matrix characterization for modal logics we are aware of. Representing worlds using prefixes is convenient for proof search, since the structures of prefixes automatically encode the required accessibility relation amongst the worlds they represent. Since prefixes are nothing more than sequences of positive integers, world realizations become string substitutions, while proof search becomes a string unification problem, where each logic allows different kinds of string substitutions. Our characterization is far more semantically motivated, since our world names are simply identifiers for worlds, and a separate formula frame is required to represent the required accessibility relation amongst the named worlds. In a way, we have traded efficiency during proof search for uniformity and semantic clarity.

Although matrix characterizations for modal logics have not been widely studied, they are closely related to tableau systems, which have (see [10] for an overview). The connection is that paths correpond naturally to tableau branches, complementary paths to closed branches, and matrices to compact representations of the atomic formulas on all potential complete branches. So far, matrix characterizations have been based on *explicit* tableau systems that refer to concrete worlds in their rules, either in the form of integer prefixes [8, 14] or atomic world symbols similar to our world names [15, 3]. *Implicit* tableau systems, on the other hand, do not refer to worlds. Instead, a tableau node is understood to be bound to a single world in the countermodel, and the semantic properties of each logic are built directly into the tableau rules [17, 10]. In translating a tableau system into a corresponding matrix characterization, the nondeterminism arising from tableau rule orderings disappears, and the incremental search for a closed tableau becomes a simultaneous search to make all paths complementary (i.e. make all potential branches closed). We have seen how to abstract world assignments from explicit tableau systems, but it remains to be seen if implicit tableau systems can give rise to equivalent matrix characterizations.

5 Conclusions and Future Work

We have presented a uniform and modular matrix characterization of validity for propositional multimodal logics. However, we have not touched on proof search, other than conceding that we have ostensibly traded suitability for automated theorem proving for expressivity, uniformity, and thereotical clarity. Like Baldoni [3], we can use our formulation to show the decidability of the multimodal logics we have considered, but it is not intended as an efficient proof search mechanism. Investigating possible proof search techniques for our characterization remains to be done.

Natural extensions to our work include matrix characterizations for a wider class of multimodal logics, such as the incestual multimodal logics of Catach [6]. A different extension would be to first-order multimodal logics, something that would conceivably follow much the same route as Wallen's generalization of his system from propositional unimodal to first-order unimodal logics [19]. Finally, the idea that certain intuitionistic modal logics may be expressible as classical bimodal logics [20, 1] opens up the interesting possibility of matrix characterizations for intuitionistic modal and multimodal logics, an area which is still largely unexplored.

BIBLIOGRAPHY

[1] Natasha Alechina, Michael Mendler, Valeria de Paiva, and Eike Ritter. Categorical and Kripke semantics for constructive S4 modal logic. In *15th International Workshop*

on Computer Science Logic, pages 292–307. Springer-Verlag, 2001.

[2] Peter B. Andrews. Theorem proving via general matings. *Journal of the Association for Computing Machinery*, 28:193–214, 1981.

[3] Matteo Baldoni. Normal multimodal logics with interaction axioms. In *Labelled Deduction*, pages 33–57. Kluwer Academic Publishers, 2000.

[4] Wolfgang Bibel. On matrices with connections. *Journal of the Association for Computing Machinery*, 28:633–645, 1981.

[5] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[6] Laurent Catach. Normal multimodal logics. In *7th National Conference on Artificial Intelligence*, pages 491–495. MIT Press, 1988.

[7] Brian F. Chellas.*Modal Logic: An Introduction*. Cambridge University Press, 1980.

[8] Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13:237–247, 1972.

[9] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel, 1983.

[10] Rajeev Goré. Tableau methods for modal and temporal logics. In *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[11] Samuli Heilala. A matrix characterization of validity for multimodal logics. Master's thesis, McGill University, forthcoming May 2008.

[12] Christoph Kreitz, Heiko Mantel, Jens Otten, and Stephan Schmitt. Connection-based proof construction in linear logic. In *14th International Conference on Automated Deduction*, pages 207–221. Springer-Verlag, 1997.

[13] Christoph Kreitz and Heiko Mantel. A matrix characterization for multiplicative exponential linear logic. *Journal of Automated Reasoning*, 32:121–166, 2004.

[14] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In *12th International Conference on Automated Deduction*, pages 723–737. Springer-Verlag, 1994.

[15] Anil Nerode. Some lectures on modal logic. In *Logic and Computation*, pages 281–334. Springer-Verlag, 1991.

[16] Dag Prawitz. An improved proof procedure. *Theoria*, 26:102–139, 1960.

[17] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12:403–423, 1983.

[18] Lincoln A. Wallen. Matrix proof methods for modal logics. In *10th International Joint Conference on Artificial Intelligence*, pages 917–923. Morgan Kaufmann, 1987.

[19] Lincoln A. Wallen. *Automated Deduction in Nonclassical Logics*. MIT Press, 1990.

[20] Frank Wolter and Michael Zakharyaschev. Intuitionistic modal logics as fragments of classical bimodal logics. In *Logic at Work*, pages 168–186. Springer-Verlag, 1999.

Samuli Heilala
McConnell Engineering Building, Room 318
School of Computer Science, McGill University
3480 University Street, Montreal, QC H3A 2A7, Canada
`sheila1@cs.mcgill.ca`

Brigitte Pientka
McConnell Engineering Building, Room 107N
School of Computer Science, McGill University
3480 University Street, Montreal, QC H3A 2A7, Canada
`bpientka@cs.mcgill.ca`