Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 305 (2014) 123–139

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On a Graph Approach to Modal Logics

Paulo A. S. Veloso [a,2] Sheila R. M. Veloso [b,3]
Mario R. F. Benevides [a,4]

[a] *Computing and System Enginneering Progr., COPPE, Federal University of Rio de Janeiro Rio de Janeiro, Brazil*

[b] *Computing and System Enginneering Dept, Engin. Fac. State University of Rio de Janeiro Rio de Janeiro, Brazil*

Abstract

We introduce a sound and complete graph calculus for multi-modal logics. This formalism internalizes the (Kripke) semantics of modal logics and provides uniform tools for expressing and manipulating modal formulas. We present the graph calculus for logic K and show how to extend it to handle some modalities, like the global and difference modalities, in a natural manner. We also indicate how it can be easily extended to other normal modal logics, such as T, S4, S5, etc.

Keywords: Modal logics, graph calculus, Kripke semantics, special modalities, refutation.

1 Introduction

We present a graph approach to multi-modal logics yielding sound and complete calculi. Such formalisms internalize the (Kripke) semantics of modal logics and provides uniform tools for expressing and manipulating modal formulas. Our derivation rules mimic the semantical properties of the modal operators, so their application is quite intuitive.

Using drawings for relations is a natural idea: represent the fact that a is related to b via relation r by an arrow $a \xrightarrow{r} b$. Then, one can reason about relations by manipulating their representations. This is a key idea underlying graph methods for reasoning about relations [3,4,5,6]. Graph representations and manipulations,

[1] Research partly sponsored by the Brazilian agencies CNPq and FAPERJ

[2] Email: pasveloso@gmail.com

[3] Email: sheila.murgel.bridge@gmail.com

[4] Email: mario@cos.ufrj.br

http://dx.doi.org/10.1016/j.entcs.2014.06.009

having precise syntax and semantics, are proof methods. In our graph calculi the derivations are very similar to the reasoning on modalities on the semantical level.

Modal logics and graphs are closely related. Kripke semantics is often presented by means of labeled graphs for the accessibility relation associated to each modality. Now, our approach here is as follows. [5] We associate a graphical representation to a modal formula and convert it to a graph, which will have empty extension if the formula is unsatisfiable, otherwise one can read a model from it (as much as in tableaux) [8].

The structure of this paper is as follows. Section 2 introduces and illustrates the main ideas underlying our graph approach to modal logics. Section 3 presents our graph language for modal logics: syntax, semantics and some constructions. Section 4 introduces our graph calculus for modal logics, with its conversion and expansion rules, discussing its correctness. In Section 5 we indicate some extensions of our graph calculus to handle some special modalities (like the global one and difference) as well as some modalities with properties like reflexivity, symmetry, transitivity and determinism. Section 6 presents some concluding remarks.

2 Graph Approach to Modalities: main ideas

We now introduce the main ideas of our graph approach to modal logics.

An interesting feature of the graph approach is its 2-dimensional notation providing pictorial representations that support manipulations. We now examine and illustrate these ideas (see Sections 3 and 4 for more details).

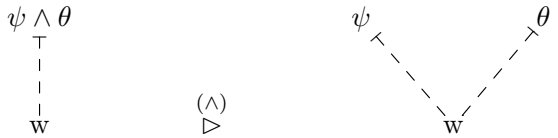
A graph is a finite set of (alternative) slices. A slice consists of a draft together with a distinguished node. A draft consists of finite sets of nodes and arcs.

Arcs may be binary or unary. A binary arc stands for accessibility among states; we represent that node v is accessible from node u by the relation of α by a solid arrow labelled α from u to v: u $\stackrel{\alpha}{\rightarrow}$ v. A unary arc is meant to capture the fact that a formula holds at state; we represent that formula φ holds at node w by a dashed line from w to φ: w – – $\dashv\varphi$. Thus, one may regard a set of arcs as a description of a model (see Examples 2.2 and 2.3).

We also mark the current state; we distinguish the current node: $\hat{\text{x}}$.

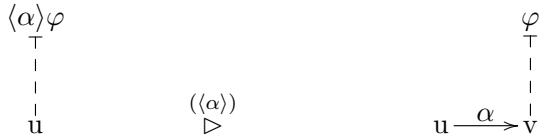
We will also have rules for manipulating these representations.

($\wedge$) We can eliminate conjunction by splitting a dashed line into two, as follows:



($\langle\alpha\rangle$) We can eliminate modality $\langle\alpha\rangle$, by adding a new node, via the conversion:

[5] Graphs will be defined in the paper: the extension of a graph will be a set.

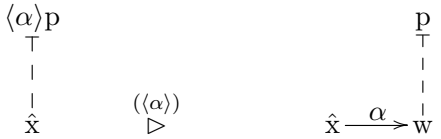


The above drawings are graphical representations of drafts.

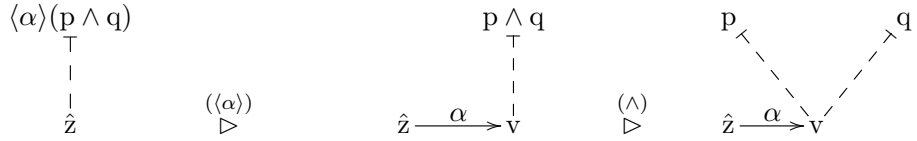
We now illustrate how we can establish consequence by means of drawings.

Example 2.1 We wish to show that $\langle\alpha\rangle p$ follows from $\langle\alpha\rangle(p \wedge q)$. We can represent the formulas $\langle\alpha\rangle p$ and $\langle\alpha\rangle(p \wedge q)$ by slices T and S, respectively as follows:

We can convert slice T (of formula $\langle\alpha\rangle p$) to slice T' as follows:



We can convert slice S (of $\langle\alpha\rangle(p \wedge q)$) to slice S' as follows:



Now, let us compare the resulting slices S' and T', namely:

We see that we can find within slice S' a copy of slice T'. The node mapping $x \mapsto z, w \mapsto v$ preserves unary and binary arcs as well as distinguished nodes: it gives a homomorphism from T' to S' (see 3.2). So, any state satisfying formula $\langle\alpha\rangle(p \wedge q)$ will also satisfy formula $\langle\alpha\rangle p$.

This example illustrates a graph approach to consequence. Given formulas ψ and θ, we represent them by single-node slices S and T and transform these slices to S' and T'. If we have a homomorphism from T' to S', then $\psi \models \theta$.

We now illustrate some other features of this approach.

Example 2.2 Consider the formulas p and $\langle\alpha\rangle p$.

(i) We do not expect $\langle\alpha\rangle p$ to follow from p in general. The single-node slices S and T (for p and $\langle\alpha\rangle p$) convert to slices S' and T', respectively as follows:

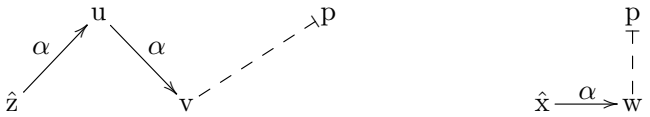
We cannot find a homomorphism from T' to S'. Now, from slice S' we can read the following natural model: $M = \{\mathrm{y}\}$, $\alpha^{\mathfrak{M}} = \emptyset$ and $\mathbf{V}(\mathrm{p}) = \{\mathrm{y}\}$. In this model $\mathfrak{M}$, state y satisfies p (as $\mathrm{y} \in \mathbf{V}(\mathrm{p})$), but not $\langle\alpha\rangle\mathrm{p}$ (as $\alpha^{\mathfrak{M}} = \emptyset$).

(ii) Now, we expect $\langle\alpha\rangle\mathrm{p}$ to follow from p if the relation corresponding to α is reflexive. In this case, every node is α-reachable from itself. So, we can transform the above slices S' and T' to $\mathsf{S}_{\mathtt{rf}}$ and $\mathsf{T}_{\mathtt{rf}}$ respectively as follows:

Now, the node mapping $\mathrm{x}, \mathrm{w} \mapsto \mathrm{y}$ gives a homomorphism from $\mathsf{T}_{\mathtt{rf}}$ to $\mathsf{S}_{\mathtt{rf}}$.

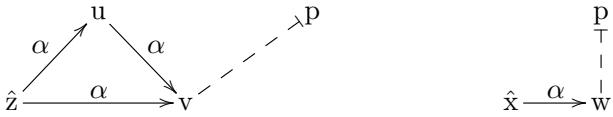
Example 2.3 The situation with $\langle\alpha\rangle\mathrm{p}$ and $\langle\alpha\rangle\langle\alpha\rangle\mathrm{p}$ is similar to Example 2.2.

(i) We do not expect $\langle\alpha\rangle\mathrm{p}$ to follow from $\langle\alpha\rangle\langle\alpha\rangle\mathrm{p}$ in general. Converting the slices for $\langle\alpha\rangle\langle\alpha\rangle\mathrm{p}$ and $\langle\alpha\rangle\mathrm{p}$, we obtain the following slices S and T:



Again, we cannot find a homomorphism from T to S and, from slice S, we can read a natural model: $M = \{\mathrm{z}, \mathrm{u}, \mathrm{v}\}$, $\alpha^{\mathfrak{M}} = \{(\mathrm{z}, \mathrm{u}), (\mathrm{u}, \mathrm{v})\}$ and $\mathbf{V}(\mathrm{p}) = \{\mathrm{v}\}$. In this model $\mathfrak{M}$, state z satisfies $\langle\alpha\rangle\langle\alpha\rangle\mathrm{p}$, but not $\langle\alpha\rangle\mathrm{p}$ (as $\mathrm{u} \notin \mathbf{V}(\mathrm{p})$).

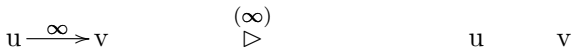
(ii) Now, we expect $\langle\alpha\rangle\mathrm{p}$ to follow from $\langle\alpha\rangle\langle\alpha\rangle\mathrm{p}$ if the relation corresponding to α is transitive. In this case, the above slices S and T will transform to slices $\mathsf{S}_{\mathtt{tr}}$ and $\mathsf{T}_{\mathtt{tr}}$, respectively as follows:



Now, the mapping $\mathrm{x} \mapsto \mathrm{z}$, $\mathrm{w} \mapsto \mathrm{v}$ gives a homomorphism from $\mathsf{T}_{\mathtt{tr}}$ to $\mathsf{S}_{\mathtt{tr}}$.

Examples 2.2 and 2.3 indicate how one can handle some properties of a relation. One can similarly handle some special relations, like the square relation ∞. As this relation connects any two states, it imposes no restriction.

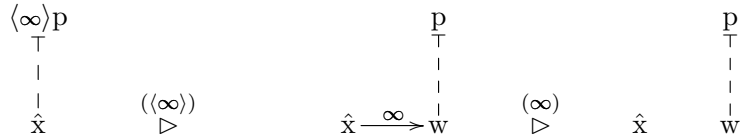
(∞) We can eliminate the square relation ∞ by the following conversion:



Example 2.4 Given the formulas p and $\langle\infty\rangle\mathrm{p}$, we wish to show $\mathrm{p} \models \langle\infty\rangle\mathrm{p}$.

(i) The single-node slice S for p converts to slice S' in Example 2.2.

(ii) The single-node T for $\langle\infty\rangle\mathrm{p}$ converts to slice T_{∞} as follows:



(iii) Now, the node mapping $x, w \mapsto y$ gives a homomorphism from T_∞ to S'.

We can see that $\langle\infty\rangle$ behaves as the global modality E [2] (see also 5.1).

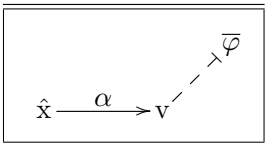
We now examine the representation for negation.

The extension of a formula is the set of states satisfying it. So, the extension of the negated formula $\neg\varphi$ is the complement of the extension of the formula φ. Thus, negation can be represented by an overbar (for complement).

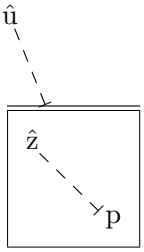
We thus have a larger syntactical category representing sets of states: the expressions, encompassing formulas, slices, graphs and their complements. So, if a formula φ converts to a slice S, then $\neg\varphi$ converts to $\overline{\mathsf{S}}$, where we can enclose the slice within a box for better readability.

We can represent formulas $\langle\alpha\rangle\neg\varphi$ and $\neg\langle\alpha\rangle\varphi$ respectively by the slices:

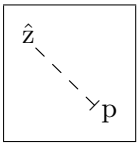
Thus, we can represent $[\alpha]\varphi$ by its equivalent $\neg\langle\alpha\rangle\neg\varphi$, i. e. by the slice:



Also, we can use expressions as labels for unary arcs. Expressions provide more flexible representations for formulas. For instance, we can represent formula p by the slice $\hat{z} - - \dashv p$. So, we can represent formula $\neg p$ by the expression $E = \overline{\hat{z} - - \dashv p}$ or by a slice like

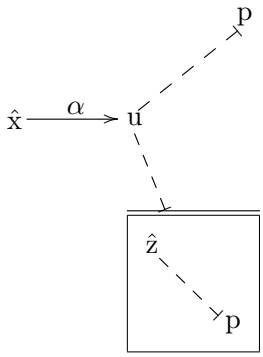


In the above slice representation of the formula $\neg p$, we have a negative arc: the arc having under complement the following slice T:

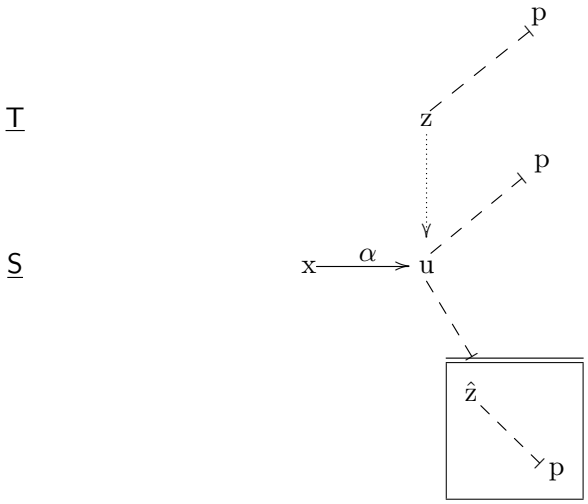


We can now establish that some formulas are not satisfiable in our approach.

Example 2.5 Consider the formula $\langle\alpha\rangle(p \wedge \neg p)$. Its slice converts to slice S:



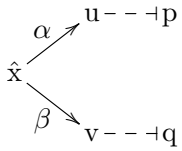
This slice S has a conflicting situation. For, on node u, we have two contradictory unary arcs: a positive arc (labelled p) and a negative arc (corresponding to $\overline{\text{p}}$). Indeed, within slice S we find under complement the above slice T. Now, consider the underlying drafts: $\underline{\text{T}}$ of T and $\underline{\text{S}}$ of S. By mapping z to u, we can map draft $\underline{\text{T}}$ into draft $\underline{\text{S}}$, as follows:

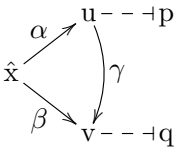


The preceding example also indicates how our graph approach handles the necessitation rule. It shows that formula $\langle\alpha\rangle(\text{p} \wedge \neg\text{p})$ is unsatisfiable because so is formula $\text{p} \wedge \neg\text{p}$. In other words, since formula $\neg(\text{p} \wedge \neg\text{p})$ is valid, so is formula $\neg\langle\alpha\rangle(\text{p} \wedge \neg\text{p})$, which is equivalent to $[\alpha]\neg(\text{p} \wedge \neg\text{p})$.

In general, our graph approach to refutation is as follows. Given a formula φ, we represent it by a single-node slice S and transform it to S′. If we can find a conflicting situation within S′ (see 3.2), then $\varphi \models \bot$.

We will be able to convert every modal formula to a graph (see Proposition 4.2: Conversion to basic graph). Consider, however, the following two slices S′ and S″:





Slice S' corresponds to the modal formula $\begin{pmatrix} \langle\alpha\rangle \mathrm{p} \\ \wedge \\ \langle\beta\rangle \mathrm{q} \end{pmatrix}$, but one does not have a modal formula corresponding to slice S''. So, slices (and graphs) will turn out to be more expressive than modal formulas.

3 Graph Language for Modalities

We now present our graph language for modal logics: first its syntax and semantics (in 3.1), and then some concepts and constructions (in 3.2).

3.1 Modal Graph Syntax and Semantics

We now present syntax and semantics of our graph language for modal logics.

We will consider a modal language L with set Ψ of formulas involving propositional letters from a set PS and modalities $\langle\alpha\rangle$ for $\alpha \in \Xi$. The corresponding graph language will also have a denumerably infinite set $\mathbb{Nd}$ of (state) *nodes* (we will use x and y for the first 2 nodes).

We introduce the syntax of our graph concepts by mutual recursion.

(E) The *expressions* are the formulas in Ψ, the slices and the graphs (see below), as well as $\overline{\mathrm{E}}$, for an expression E.

(a) An *arc* over a set $N \subseteq \mathbb{Nd}$ can be either unary or binary.
  (i) A unary arc is a pair $\mathrm{w}|\mathrm{E}$, where $\mathrm{w} \in N$ and E is an expression.
  (ii) A binary arc is a triple $\mathrm{u}\,\alpha\,\mathrm{v}$, where $\mathrm{u}, \mathrm{v} \in N$ and $\alpha \in \Xi$.

(Σ) A *sketch* $\Sigma = \langle \mathsf{N}; \mathsf{A} \rangle$ consists of 2 sets: a set $\mathsf{N} \subseteq \mathbb{Nd}$ (of nodes) and a set A of arcs over N. A *proper* sketch is one with non-empty set of nodes.

(D) A *draft* is a sketch with finite sets N of nodes and A of arcs.

(S) A *slice* $\mathsf{S} = \langle \mathsf{N}; \mathsf{A} : \mathrm{z}_{\mathsf{S}} \rangle$ consists of a draft $\underline{\mathsf{S}} = \langle \mathsf{N}; \mathsf{A} \rangle$, its *underlying draft*, together with a *distinguished node* $\mathrm{z}_{\mathsf{S}} \in \mathsf{N}$.

(G) A graph is a finite set of slices.

The *empty graph* $\{\,\}$ has no slice. Slices and graphs are finite objects, whereas sketches are useful in some contexts (see 3.2 and 4.3) The *slice of expression* E is the single-node single-arc slice $\mathsf{Sl}(\mathrm{E}) \coloneqq \langle \{\mathrm{x}\}; \{\mathrm{x}|\mathrm{E}\} : \mathrm{x} \rangle$ (where x is the first node in $\mathbb{Nd}$). For instance, Example 2.1 shows $\mathsf{Sl}(\langle\alpha\rangle \mathrm{p}) = \langle \{\mathrm{x}\}; \{\mathrm{x}|\langle\alpha\rangle \mathrm{p}\} : \mathrm{x} \rangle$.

We now examine the semantics of our graph language. We will use models for semantics: a *model* consists of an underlying frame and a valuation [2]. A *frame* $\mathfrak{F} = (M, (\alpha^{\mathfrak{F}})_{\alpha \in \Xi})$ assigns to each $\alpha \in \Xi$ a binary relation $\alpha^{\mathfrak{F}}$ on $M \neq \emptyset$. A *valuation* $\mathbf{V}$ assigns to each propositional letter $\mathrm{p} \in \mathsf{PS}$ a subset $\mathbf{V}(\mathrm{p})$ of M.

We now introduce the semantics of our graph concepts (again by mutual recursion). Consider a model $\mathfrak{M} = (M, (\alpha^{\mathfrak{F}})_{\alpha \in \Xi}, \mathbf{V})$ as above.

(E) The *extension of expression* E is the subset $[\mathrm{E}]_{\mathfrak{M}} \subseteq M$ defined as follows. For a

formula $\varphi \in \Psi$, $[\varphi]_{\mathfrak{M}} \coloneqq \{s \in M \,/\, \mathfrak{M}, s \Vdash \varphi\}$. For a slice or a graph, we use their extensions: $[\mathsf{S}]_{\mathfrak{M}} \coloneqq [\![S]\!]_{\mathfrak{M}}$ and $[\mathsf{G}]_{\mathfrak{M}} \coloneqq [\![G]\!]_{\mathfrak{M}}$ (as defined below). For $\overline{\mathrm{E}}$, we set $[\overline{\mathrm{E}}]_{\mathfrak{M}}$ as the complement of $[\mathrm{E}]_{\mathfrak{M}}$, i. e. $[\overline{\mathrm{E}}]_{\mathfrak{M}} \coloneqq M \setminus [\mathrm{E}]_{\mathfrak{M}}$.

(a) For an arc a, we define *satisfaction* under $\mathsf{g} : N \to M$ ($\mathsf{g} \Vdash_{\mathfrak{M}} \mathsf{a}$) as follows.
  (i) For a unary arc: $\mathsf{g} \Vdash_{\mathfrak{M}} \mathrm{w}|\mathrm{E}$ iff $\mathrm{w} \in N$ and $\mathrm{w}^{\mathsf{g}} \in [\mathrm{E}]_{\mathfrak{M}}$.
  (ii) For a binary arc: $\mathsf{g} \Vdash_{\mathfrak{M}} \mathrm{u}\,\alpha\,\mathrm{v}$ iff $\mathrm{u}, \mathrm{v} \in N$ and $(\mathrm{u}^{\mathsf{g}}, \mathrm{v}^{\mathsf{g}}) \in \alpha^{\mathfrak{F}}$.

(Σ) Assignment $\mathsf{g} : N \to M$ *satisfies* sketch $\Sigma = \langle \mathsf{N}; \mathsf{A} \rangle$ in $\mathfrak{M}$ (noted $\mathsf{g} : \Sigma \to \mathfrak{M}$) iff g satisfies all its arcs, i.e. $\mathsf{g} \Vdash_{\mathfrak{M}} \mathsf{a}$, for every arc $\mathsf{a} \in \mathsf{A}$.

(S) The *extension* of a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : \mathrm{z}_{\mathsf{S}} \rangle$ is the subset of M consisting of the values of its distinguished node z_{S} for the assignments satisfying its underlying draft $\underline{\mathsf{S}}$: $[\![\mathsf{S}]\!]_{\mathfrak{M}} := \{\mathrm{z}_{\mathsf{S}}{}^{\mathsf{g}} \in M \,/\, \mathsf{g} : \underline{\mathsf{S}} \to \mathfrak{M}\}$.

(G) The *extension* of a graph G is the union of the extensions of its slices, i. e. $[\![\mathsf{G}]\!]_{\mathfrak{M}} := \bigcup_{\mathsf{S} \in \mathsf{G}} [\![\mathsf{S}]\!]_{\mathfrak{M}}$.

Expressions E and F are *equivalent* (noted $\mathrm{E} \equiv \mathrm{F}$) iff $[\mathrm{E}]_{\mathfrak{M}} = [\mathrm{F}]_{\mathfrak{M}}$, for every model $\mathfrak{M}$. An expression E and its slice $\mathsf{Sl}(\mathrm{E})$ are equivalent. Also, a slice S and a singleton graph $\{\mathsf{S}\}$ are equivalent, so one may identify them. An expression E is *null* iff $\mathrm{E} \equiv \bot$. Clearly, the empty graph $\{\ \}$, having no slice, is null.

We will give a calculus for establishing that an expression is null in Section 4.

3.2 Modal Graph Concepts and Constructions

We will now examine some graph concepts and constructions.

We begin with the idea of natural model illustrated in Examples 2.2 and 2.3.

Consider a proper sketch $\Sigma = \langle \mathsf{N}; \mathsf{A} \rangle$. Its *natural frame* is $\mathfrak{F}[\Sigma] \coloneqq (\mathsf{N}, (\alpha^{\mathfrak{F}[\Sigma]})_{\alpha \in \Xi})$ with $\alpha^{\mathfrak{F}[\Sigma]} \coloneqq \{(\mathrm{u}, \mathrm{v}) \in \mathsf{N} \times \mathsf{N} \,/\, \mathrm{u}\,\alpha\,\mathrm{v} \in \mathsf{A}\}$. Now, the *natural valuation* for Σ has $\mathbf{V}_{\Sigma}(\mathrm{p}) \coloneqq \{\mathrm{w} \in N \,/\, \mathrm{w}|\mathrm{p} \in \mathsf{A}\}$. The *natural model* for Σ is $\mathfrak{N}[\Sigma] \coloneqq (\mathfrak{F}[\Sigma], \mathbf{V}_{\Sigma})$.

We use the notation '+' for adding nodes or arcs. Consider a sketch $\Sigma = \langle \mathsf{N}; \mathsf{A} \rangle$ and a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : \mathrm{z}_{\mathsf{S}} \rangle$. For a node $\mathrm{w} \in \mathbb{N}\mathsf{d}$, we set $\Sigma + \mathrm{w} \coloneqq \langle \mathsf{N} \cup \{\mathrm{w}\}; \mathsf{A} \rangle$ and $\mathsf{S} + \mathrm{w} \coloneqq \langle \underline{\mathsf{S}} + \mathrm{w} : \mathrm{z}_{\mathsf{S}} \rangle$. For arcs, we set $\Sigma + \mathrm{E}|\mathrm{w} \coloneqq \langle \mathsf{N} \cup \{\mathrm{w}\}; \mathsf{A} \cup \{\mathrm{E}|\mathrm{w}\} \rangle$ and $\Sigma + \mathrm{u}\,\alpha\,\mathrm{v} \coloneqq \langle \mathsf{N} \cup \{\mathrm{u}, \mathrm{v}\}; \mathsf{A} \cup \{\mathrm{u}\,\alpha\,\mathrm{v}\} \rangle$; then $\mathsf{S} + \mathsf{a} \coloneqq \langle \underline{\mathsf{S}} + \mathsf{a} : \mathrm{z}_{\mathsf{S}} \rangle$.

We extend a node translation $\nu : N \to \mathbb{N}\mathsf{d}$ naturally to arcs over N as follows: for a unary arc $\mathsf{a} = \mathrm{w}|\mathrm{E}$, we set $\mathsf{a}^{\nu} \coloneqq \mathrm{w}^{\nu}|\mathrm{E}$, and for a binary arc $\mathsf{a} = \mathrm{u}\,\alpha\,\mathrm{v}$, we set $\mathsf{a}^{\nu} \coloneqq \mathrm{u}^{\nu}\,\alpha\,\mathrm{v}^{\nu}$. We also extend it naturally to sets of nodes and of arcs, sketches and slices as follows: $N^{\nu} \coloneqq \{\mathrm{w}^{\nu} \,/\, \mathrm{w} \in N\}$ and $A^{\nu} \coloneqq \{\mathsf{a}^{\nu} \,/\, \mathsf{a} \in A\}$, for a sketch $\Sigma = \langle \mathsf{N}; \mathsf{A} \rangle$, $\Sigma^{\nu} \coloneqq \langle \mathsf{N}^{\nu}; \mathsf{A}^{\nu} \rangle$ and, for a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : \mathrm{z}_{\mathsf{S}} \rangle$, $\mathsf{S}^{\nu} \coloneqq \langle \underline{\mathsf{S}}^{\nu} : \mathrm{z}_{\mathsf{S}}{}^{\nu} \rangle$.

A useful node translation is renaming. Given node pair $(\mathrm{u}, \mathrm{v}) \in \mathbb{N}\mathsf{d}^2$, we define *rename* u to v as follows:

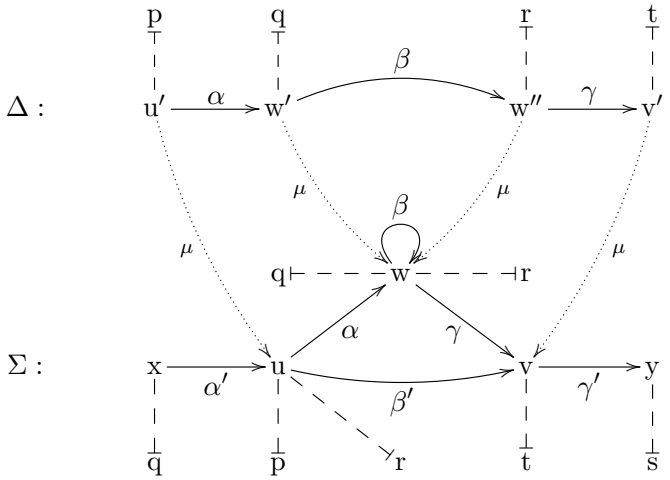
$$\mathrm{w}^{[\mathrm{u}/\mathrm{v}]} \coloneqq \begin{cases} \mathrm{v} & \text{if } \mathrm{w} = \mathrm{u} \\ \mathrm{w} & \text{if } \mathrm{w} \neq \mathrm{u} \end{cases}$$

We now introduce morphisms for comparing sketches.

Consider sketches $\Delta = \langle \mathsf{N}_{\Delta}; \mathsf{A}_{\Delta} \rangle$ and $\Sigma \coloneqq \langle \mathsf{N}_{\Sigma}; \mathsf{A}_{\Sigma} \rangle$. A node translation function $\mu : \mathsf{N}_{\Delta} \to \mathsf{N}_{\Sigma}$ is a *morphism* from Δ to Σ (noted $\mu : \Delta \dashrightarrow \Sigma$) iff it preserves arcs:

for every arc $\mathsf{a} \in \mathsf{A}_\Delta$, a^μ is an arc in A_Σ.

For the slices S and T of Example 2.5 (in Section 2), $z \mapsto u$ gives a morphism $\theta : \underline{\mathsf{T}} \dashrightarrow \underline{\mathsf{S}}$. Another example is as follows:



A morphism transfers satisfying assignments by composition.

Lemma 1 (Assignment transfer) *Given a morphism $\mu : \Delta \dashrightarrow \Sigma$ and a model $\mathfrak{M}$, for each assignment g satisfying Σ in $\mathfrak{M}$, the composite $\mathsf{g} \cdot \mu$ is an assignment satisfying Δ in $\mathfrak{M}$.*

Proof. For any arc a of Δ, a^μ is an arc of Σ, so that, for every g satisfying a^μ in a model $\mathfrak{M}$, $\mathsf{g} \cdot \mu$ is an assignment satisfying a in $\mathfrak{M}$. □

A sketch Σ is *zero* iff, for some slice $\mathsf{T} = \langle \underline{\mathsf{T}} : \mathsf{z}_\mathsf{T} \rangle$, there exists a morphism $\mu : \underline{\mathsf{T}} \dashrightarrow \Sigma$, such that $\mathsf{z}_\mathsf{T}{}^\mu | \overline{\mathsf{T}}$ is an arc of Σ. A slice S is *zero* iff its underlying draft $\underline{\mathsf{S}}$ is zero. A graph is *zero* iff all its slices are zero.

For instance, slice S of Example 2.5 is zero.

Lemma 2 (Zero sketches) *No assignment can satisfy a zero sketch.*

Proof. By Lemma 1. If $\mathsf{g} : \Sigma \to \mathfrak{M}$, then we have a contradiction: $\mathsf{z}_\mathsf{T}{}^{\mathsf{g}\cdot\mu} \notin [\![\mathsf{T}]\!]_\mathfrak{M}$ (as $\mathsf{z}_\mathsf{T}{}^\mu | \overline{\mathsf{T}} \in \mathsf{A}_\Sigma$) and $\mathsf{z}_\mathsf{T}{}^{\mathsf{g}\cdot\mu} \in [\![\mathsf{T}]\!]_\mathfrak{M}$ (as $\mathsf{g} \cdot \mu : \underline{\mathsf{T}} \to \mathfrak{M}$). □

Corollary 1 (Zero slices and graphs) *Zero slices and graphs are null.*

Proof. By Lemma 2: if $[\![\mathsf{H}]\!]_\mathfrak{M} \neq \emptyset$, then $[\![\mathsf{T}]\!]_\mathfrak{M} \neq \emptyset$, for some slice $\mathsf{T} \in \mathsf{H}$, thus some M-assignment satisfies the underlying draft $\underline{\mathsf{T}}$. □

We can also introduce homomorphisms for comparing slices as follows. Given slices $\mathsf{T} = \langle \mathsf{N}_\mathsf{T}; \mathsf{A}_\mathsf{T} : \mathsf{z}_\mathsf{T} \rangle$ and $\mathsf{S} = \langle \mathsf{N}_\mathsf{S}; \mathsf{A}_\mathsf{S} : \mathsf{z}_\mathsf{S} \rangle$, a *homomorphism* $\eta : \mathsf{T} \to \mathsf{S}$ is a node translation $\eta : \mathsf{N}_\mathsf{T} \to \mathsf{N}_\mathsf{S}$ such that $\eta : \underline{\mathsf{T}} \dashrightarrow \underline{\mathsf{S}}$ and η preserves distinguished nodes: $\eta(\mathsf{z}_\mathsf{T}) = \mathsf{z}_\mathsf{S}$. For a homomorphism $\eta : \mathsf{T} \to \mathsf{S}$ and a model $\mathfrak{M}$: $[\![\mathsf{S}]\!]_\mathfrak{M} \subseteq [\![\mathsf{T}]\!]_\mathfrak{M}$ (by Lemma 1). Also, node renaming is a homomorphism from S to $\mathsf{S}^{[\mathsf{u}/\mathsf{v}]}$.

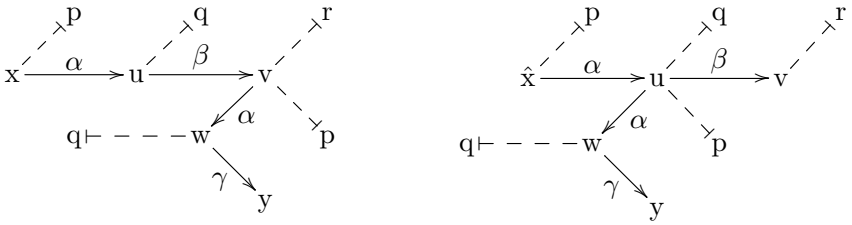
We now wish to glue a slice T onto a sketch Σ (or slice S) via a designated node $\mathsf{w} \in \mathbb{N}\mathsf{d}$. We do this by identifying the distinguished node of T with the designated

node w, to obtain a sketch $\Sigma^{w}\mathsf{T}$ and a slice $\mathsf{S}^{w}\mathsf{T}$ with the following aspects:

The next example illustrates these ideas.

Example 3.1 (Gluing) Consider slices S and T respectively as follows:

The glued draft $\underline{\mathsf{S}}^{v}\mathsf{T}$ and slice $\mathsf{S}^{u}\mathsf{T}$ are respectively as follows:



More precisely, consider a slice $\mathsf{T} = \langle \mathsf{N}; \mathsf{A} : z \rangle$ and a node $w \in \mathbb{N}\mathsf{d}$.

(Σ) Given a sketch $\Sigma = \langle \mathsf{N}_\Sigma; \mathsf{A}_\Sigma \rangle$, to glue T onto sketch Σ, we proceed as follows.
- (i) Add node w to Σ, obtaining sketch $\Sigma_0 \coloneqq \Sigma + w$ (with $\mathsf{N}_0 = \mathsf{N}_\Sigma \cup \{w\}$).
- (ii) Take a copy $\mathsf{T}_1 = \langle \mathsf{N}_1; \mathsf{A}_1 : z_1 \rangle$ of slice T with $\mathsf{N}_1 \cap \mathsf{N}_0 = \emptyset$.
- (iii) Now, rename z_1 to w in slice T_1 to obtain slice $\mathsf{T}_2 \coloneqq \mathsf{T}_1{}^{[z_1/w]}$ (with sets $\mathsf{N}_2 = \mathsf{N}_1{}^{[z_1/w]}$ and $\mathsf{A}_2 = \mathsf{A}_1{}^{[z_1/w]}$).
- (iv) Then, form the *glued sketch* $\Sigma^{w}\mathsf{T} \coloneqq \langle \mathsf{N}_0 \cup \mathsf{N}_2; \mathsf{A}_\Sigma \cup \mathsf{A}_2 \rangle$.

(S) Given a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : z_\mathsf{S} \rangle$, we glue slice T onto S, by gluing T onto the underlying draft $\underline{\mathsf{S}}$ and using the distinguished node: the *glued slice* is $\mathsf{S}^{w}\mathsf{T} \coloneqq \langle \underline{\mathsf{S}}^{w}\mathsf{T} : z_\mathsf{S} \rangle$.

Lemma 3 (Arc addition and gluing) *For slices* S *and* T*:* $\mathsf{S} + w|\mathsf{T} \equiv \mathsf{S}^{w}\mathsf{T}$.

Proof. By construction (we can combine satisfying assignments). □

4 Graph Calculus for Modalities

We now introduce our graph calculus for modal logics. We will first introduce basic objects and then the rules, conversion (in 4.1) and expansion (in 4.2), and examine correctness (soundness and completeness) of the calculus in 4.3.

To establish that an expression is null, we convert it to a graph (by conversion rules) and then try to obtain a zero graph by repeatedly applying the expansion rule (see Example 4.3 in 4.2). Each rule will transform an expression to an equivalent one, so we can use it in any context. As usual, we employ '$-^*$' for the reflexive-transitive closure of a relation.

We define basic expressions, arcs, sketches, slices and graphs by mutual recursion. An expression E is *basic* iff it is a propositional letter or $\overline{\mathsf{T}}$, where T is a basic slice (see below). An unary arc E|w is *basic* iff E is a basic expression and a binary

arc $\mathrm{u}\,\alpha\,\mathrm{v}$ is *basic*. A sketch is *basic* iff all its arcs are basic. A slice S is *basic* iff its underlying draft $\underline{\mathsf{S}}$ is a basic sketch. A graph is *basic* iff all its slices are basic.

In Example 2.1, slices S' and T' are basic, whereas slices S and T are not basic.

4.1 Modal Graph Conversion

We now examine conversion in our calculus. The aim of the conversion rules is to transform each expression to an equivalent basic graph.

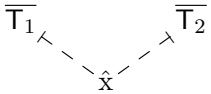
The conversion rules come from equivalences between expressions. They will be of three kinds: formula rules, complementation rules and structural rules. [6]

The *formula rules* eliminate logical symbols by converting an expression to an equivalent expression. For $\neg$: $\neg\varphi \overset{(\neg)}{\rhd} \overline{\varphi}$, and the rules for $\wedge$ and $\langle\alpha\rangle$ give the conversions (cf. Section 2 (Graph Approach to Modalities: main ideas)):

The rule for $\bot$ replaces it by the empty graph. The rule for $\vee$ replaces $\psi \vee \theta$ by the graph with the following 2 slices:

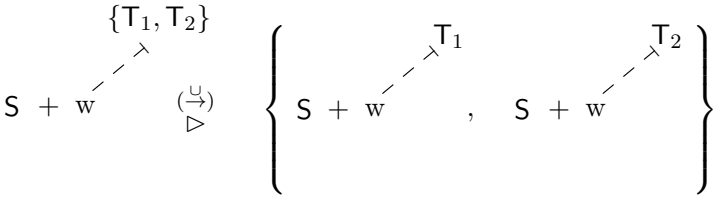
By applying these formula rules in any context, one can eliminate connectives and modalities. We may still have complement as well as slices or graphs and their complements may appear in arcs. The remaining 5 rules will address these cases.

We have 2 *complementation rules*: one erases double complement ($\overline{\overline{\mathsf{E}}} \overset{(=)}{\rhd} \mathsf{E}$) and rule ($\overline{\mathsf{H}}$) converts a complemented graph to a single-node slice, e. g. $\overline{\{\mathsf{T}_1, \mathsf{T}_2\}}$ to the single-node slice:



The 3 *structural rules* eliminate graphs and slices occurring as arc labels and introduce slice of expression.

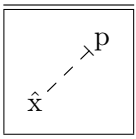
($\overset{\sqcup}{\to}$) Addition of a graph-arc converts to alternative additions of its slice-arcs, in view of the equivalence $\mathsf{S} + \mathrm{w}|\mathsf{H} \equiv \{\mathsf{S} + \mathrm{w}|\mathsf{T} \,/\, \mathsf{T} \in \mathsf{H}\}$. For instance, for $\{\mathsf{T}_1, \mathsf{T}_2\}$:



[6] Recall that x and y are the first 2 nodes (cf. 3.1).

($\overset{\mathsf{T}}{\rightarrow}$) Addition of a slice-arc converts to the glued slice: $\mathsf{S} + \mathrm{w}|\mathsf{T} \overset{(\overset{\mathsf{T}}{\rightarrow})}{\triangleright} \mathsf{S}^{\mathrm{w}}\mathsf{T}$.[7]

($\uparrow$ E) An expression converts to its slice: $\mathrm{E} \overset{(\uparrow \mathrm{E})}{\triangleright} \mathsf{Sl}(\mathrm{E})$ (cf. 3.1). So, we can replace a complemented propositional letter $\overline{\mathrm{p}}$ by the basic expression



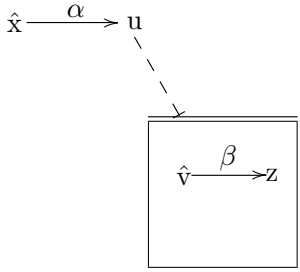
We can convert expressions modularly as the next example illustrates.

Example 4.1 Let φ be the formula $\langle\alpha\rangle\top \wedge \neg\langle\alpha\rangle\langle\beta\rangle\top \wedge \neg\langle\alpha\rangle\neg\langle\beta\rangle\top$, with $\top \coloneqq \neg\bot$.

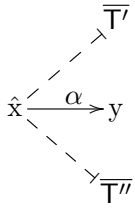
(i) Formula $\langle\alpha\rangle\top$ converts to the slice $\hat{\mathrm{x}} \xrightarrow{\alpha} \mathrm{y}$.

(ii) Formula $\langle\alpha\rangle\langle\beta\rangle\top$ converts to the slice $\mathsf{T}' \coloneqq \hat{\mathrm{x}} \xrightarrow{\alpha} \mathrm{y} \xrightarrow{\beta} \mathrm{z}$.

(iii) Formula $\langle\alpha\rangle\neg\langle\beta\rangle\top$ converts to the following slice T'':



So, formula $\langle\alpha\rangle\top \wedge \neg\langle\alpha\rangle\langle\beta\rangle\top \wedge \neg\langle\alpha\rangle\neg\langle\beta\rangle\top$ converts to the following slice S:



Proposition 4.2 (Conversion to basic graph) *Every expression* E *can be effectively converted to an equivalent basic graph* E^{bs}.

4.2 Modal Graph Expansion

We now examine graph expansion and its rule in our calculus for modal logics.

The next example introduces expansion and its usefulness.

Example 4.3 (Expansion) Recall the slice S in Example 4.1. This slice S is not yet zero. Consider, however, the following slice T:

$$\hat{\mathrm{y}} \xrightarrow{\beta} \mathrm{z}$$

[7] Cf. Lemma 3 (Arc addition and gluing) in 3.2: Modal Graph Concepts and Constructions.

We can use this slice T to expand slice S to a graph G consisting of 2 alternative slices $\mathsf{S}_+ = \mathsf{S}^{\mathrm{y}}\mathsf{T}$ and $\mathsf{S}_- = \mathsf{S} + \mathrm{y}|\overline{\mathsf{T}}$, respectively as follows:

Both slices S_+ and S_- can be seen to be zero, so slice S is null (cf. 3.1).
We thus have (cf. Example 4.1) $\langle\alpha\rangle\top \wedge \neg\langle\alpha\rangle\langle\beta\rangle\top \wedge \neg\langle\alpha\rangle\neg\langle\beta\rangle\top \models \bot$, whence $\langle\alpha\rangle\top \wedge [\alpha]\langle\beta\rangle\top \models \langle\alpha\rangle\langle\beta\rangle\top$.

We can expand a slice S on a node w under a slice T, giving 2 alternative slices $\mathsf{S}^{\mathrm{w}}\mathsf{T}$ and $\mathsf{S} + \mathrm{w}|\overline{\mathsf{T}}$, which will be basic if both S and T are basic. Note that $\mathsf{S} \equiv \{\mathsf{S}^{\mathrm{w}}\mathsf{T}, \mathsf{S} + \mathrm{w}|\overline{\mathsf{T}}\}$. The *expansion rule* is as follows:

$$\text{(Exp)}\quad \mathsf{G} \cup \{\mathsf{S}\} \lhd \mathsf{G} \cup \{\mathsf{S}^{\mathrm{w}}\mathsf{T}, \mathsf{S} + \mathrm{w}|\overline{\mathsf{T}}\} \qquad \mathrm{w} \in \mathsf{N}_{\mathsf{S}} \qquad \text{expand slice } \mathsf{S}$$

4.3 Modal Graph Calculus Correctness

We now examine the correctness of our graph calculus for modal logics.

A *derivation* is a sequence of applications of conversion and expansion rules: $\vdash\!\!\!= (\rhd \cup \lhd)^*$. In Example 4.3 we have the derivation $\varphi \rhd^\star \{\mathsf{S}\} \lhd \{\mathsf{S}_+, \mathsf{S}_-\}$. Call a derivation *smooth* iff applications of conversion rules precede applications of expansion. [8] An expression is *derivably zero* iff it derives some zero graph and *expansively zero* iff it eventually expands to some zero graph.

We have soundness and completeness of (smooth) derivations.

Proposition 4.4 *Soundness: every derivably zero expression is null. Completeness: if a basic graph* H *is null, then* H *is expansively zero.*

Soundness is not difficult to see. We now indicate how one can establish completeness. The idea is saturating a slice by applications of the expansion rule. [9] Consider a basic slice S that is not expansively zero. Then, for every $\mathrm{w} \in \mathsf{N}_{\mathsf{S}}$ and basic slice T, $\mathsf{S}^{\mathrm{w}}\mathsf{T}$ or $\mathsf{S} + \mathrm{w}|\overline{\mathsf{T}}$ is not expansively zero. Thus, we can then obtain a chain of non-zero basic slices whose union is a basic sketch Σ. Now, this sketch Σ *discriminates* basic drafts: satisfying assignments into the natural model $\mathfrak{N}[\Sigma]$ are morphisms into Σ. [10] Hence, in $\mathfrak{N}[\Sigma]$, slice S has non-empty extension.

We thus have a correct calculus for null expressions.

Theorem 4.5 (Correctness) *An expression* E *is null iff* E *is derivably zero, i. e. its basic form* E^{bs} *is expansively zero.*

Proof. By Propositions 4.2 (Conversion to basic graph) and 4.4. □

---

[8] The preceding examples use smooth derivations: of the form $\mathrm{E} \rhd^\star \mathsf{G} \lhd^* \mathsf{H}$.

[9] One may regard this as an analogue of Lindenbaum's Lemma: extending a consistent theory to a maximally consistent one.

[10] For every basic draft D, $\mathsf{g} : \mathsf{D} \to \mathfrak{N}[\Sigma]$ iff $\mathsf{g} : \mathsf{D} \dashrightarrow \Sigma$ (see 5.3).

5 Graph Calculus for Modalities: some extensions

We now indicate some extensions of our graph calculus. We will examine some special modalities (in 5.1) and some modalities with special properties (in 5.2).

In Section 4, we have presented a sound and complete graph calculus for validity on arbitrary models. We will now examine some cases of frames with special accessibility relations, as well as some details in 5.3.

5.1 Special Modalities

We now indicate how we handle modalities like the global one and difference.

The global modality and difference are interesting cases [2]: $\mathfrak{M}, s \Vdash \mathsf{E}\varphi$ iff, for some $t \in M$, $\mathfrak{M}, t \Vdash \varphi$ and $\mathfrak{M}, s \Vdash \mathsf{D}\varphi$ iff, for some $t \in M \setminus \{s\}$, $\mathfrak{M}, t \Vdash \varphi$.

We handle them by considering special relations ∞ and $\neq$ as logical, in the sense that in any frame frame $\mathfrak{F} = (M, (\alpha^{\mathfrak{F}})_{\alpha \in \Xi})$, $\infty^{\mathfrak{F}}$ is the square $M^2 \coloneqq M \times M$ (if $\infty \in \Xi$) and $\neq^{\mathfrak{F}} = \{(s,t) \in M \times M \,/\, s \neq t\}$ (if $\neq \in \Xi$).

We can see that this achieves the desired effect: modality $\langle \infty \rangle$ behaves as the global E ($\mathfrak{M}, s \Vdash \langle \infty \rangle \varphi$ iff $\mathfrak{M}, s \Vdash \mathsf{E}\varphi$) and modality $\langle \neq \rangle$ behaves as difference D ($\mathfrak{M}, s \Vdash \langle \neq \rangle \varphi$ iff $\mathfrak{M}, s \Vdash \mathsf{D}\varphi$).

It remains to provide rules for manipulating ∞ and $\neq$.

The case of ∞ has already been indicated in Example 2: we can use the conversion $\overset{(\infty)}{\rhd}$ to eliminate occurrences of ∞.

We can similarly handle logical identity $=$, where $=^{\mathfrak{F}}$ is taken as the diagonal $\{(s,t) \in M \times M \,/\, s = t\}$. Now, we can eliminate an arc $\mathsf{u} \overset{=}{\to} \mathsf{v}$ from a slice S by renaming v to u throughout S (cf. 3.2), i. e. by the conversion $\mathsf{S} + \mathsf{u} \overset{=}{\to} \mathsf{v} \overset{(=)}{\rhd} \mathsf{S}^{[\mathsf{v}/\mathsf{u}]}$.

We cannot eliminate diversity $\neq$, so we allow basic sketches to have occurrences of $\neq$. We do however have rules for manipulating $\neq$.

($\neq\uparrow$) First (as $s = t$ or $s \neq t$) we have the following expansion rule

$$(\neq\uparrow) \quad \mathsf{G} \cup \{\mathsf{S}\} \lhd \mathsf{G} \cup \{\mathsf{S}^{[\mathsf{v}/\mathsf{u}]}, \mathsf{S} + \mathsf{u} \overset{\neq}{\to} \mathsf{v}\} \qquad (\mathsf{u}, \mathsf{v}) \in \mathsf{N_S}^2$$

($\neq\downarrow$) Also (as $s = s$) we have the following contraction rule

$$(\neq\downarrow) \quad \mathsf{G} \cup \{\mathsf{S} + \mathsf{w} \overset{\neq}{\to} \mathsf{w}\} \lhd \mathsf{G} \qquad \text{erase slice } \mathsf{S} + \mathsf{w} \overset{\neq}{\to} \mathsf{w}$$

With these two rules, we can establish some properties of these special modalities, such as $\varphi \to [\neq]\langle \neq \rangle \varphi$ and $\langle \infty \rangle \varphi \leftrightarrow (\varphi \vee \langle \neq \rangle \varphi)$.

Also, proceeding as in 4.3, we obtain a union sketch Σ that is $\neq$-*complete* in the sense: for $(\mathsf{u}, \mathsf{v}) \in \mathsf{N}_\Sigma{}^2$, $\mathsf{u} \neq \mathsf{v}$ iff $\mathsf{u} \overset{\neq}{\to} \mathsf{v} \in \mathsf{A}_\Sigma$. So, Σ will be discriminating for basic drafts (which may now have $\neq$-arcs).

5.2 Modalities with Special Properties

We now indicate how one can handle modalities with some simple properties, like reflexivity, symmetry, transitivity and determinism. In some cases, a categorical approach is convenient: we will outline such ideas in 5.3.

We will introduce the terminology for the case of reflexivity, the terminology for the other cases being analogous.

The case of reflexivity has already been indicated in Example 2.2. Given $\rho \in \Xi$, consider frames where relation $\rho^{\mathfrak{F}}$ is reflexive: $(s, s) \in \rho^{\mathfrak{F}}$, for every $s \in M$. We are interested in the formulas holding in every model with a reflexive frame.

Consider a sketch Σ. Call Σ *ρ-reflexive* iff the relation $\rho^{\mathfrak{F}[\Sigma]}$ of its natural frame $\mathfrak{F}[\Sigma]$ (cf. 3.2) is reflexive. Also, call Σ *ρ-reflexive basic* iff Σ is basic and ρ-reflexive. We use similar terminology for a slice S via its underlying draft $\underline{\mathsf{S}}$.

Example 2.2 suggests the conversion rule (Rf_ρ): $\mathsf{S} \rhd \mathsf{S} + \mathrm{w}\,\rho\,\mathrm{w}$, for $\mathrm{w} \in \mathsf{N_S}$. Note that S and $\mathsf{S} + \mathrm{w}\,\rho\,\mathrm{w}$ have the same extensions on a reflexive model.

With this rule, we can convert every slice S to a ρ-reflexive slice $\mathsf{S_{rf}}$. Thus, as in Proposition 4.2 (in 4.1), we can convert every expression to a graph whose slices are ρ-reflexive basic. We thus have a correct calculus as in 4.3.

The case for a symmetric σ is similar. It suffices to consider the conversion rule (Sm_σ): $\mathsf{S} \rhd \mathsf{S} + \mathrm{v}\,\sigma\,\mathrm{u}$, for an arc $\mathrm{u}\,\sigma\,\mathrm{v} \in \mathsf{A_S}$.

In the case of a transitive τ, Example 2.3 already suggests the conversion rule (Tr_τ): $\mathsf{S} \rhd \mathsf{S} + \mathrm{u}\,\tau\,\mathrm{w}$, for arcs $\mathrm{u}\,\tau\,\mathrm{v}, \mathrm{v}\,\tau\,\mathrm{w} \in \mathsf{A_S}$.

Repeated applications of this rule will convert every slice S to a τ-transitive slice $\mathsf{S_{tr}}$. But, we now have to adapt gluing and the construction in 4.3 to τ-transitive sketches (see 5.3). With these adaptations, we have a correct calculus as in 4.3.

The case of a deterministic δ is similar to the transitive one. We use a conversion rule with renaming (as in 5.1), rule (Dt_δ): $\mathsf{S} \rhd \mathsf{S}^{[\mathrm{v}''/\mathrm{v}']}$, for arcs $\mathrm{u}\,\delta\,\mathrm{v}', \mathrm{u}\,\delta\,\mathrm{v}'' \in \mathsf{A_S}$. As above, repeated applications of this rule will convert every slice S to a δ-deterministic slice $\mathsf{S_{dt}}$, but we need appropriate adaptations. With such adaptations, we will have a correct calculus as in 4.3.

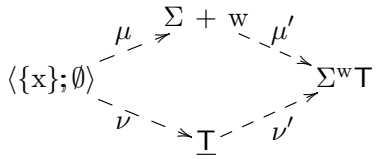
The cases examined above are relatively simple: the rules do not involve addition of new nodes. Consider, however, the case of seriality: the natural rule will add a new node. Such cases are best left for the expansion stage (cf. 4.3). One can handle several cases by pairs of sketches, i. e. a rule replacing Δ by Σ.

5.3 Some Details: categorical aspects and discrimination

We will now examine some details concerning 5.2 and 4.3.

We first consider some categorical constructions: co-limits and pushouts [7].

The category of sketches and morphisms has co-limits. The co-limit of a diagram of sketches can be obtained as expected: obtain the co-limit of the sets of nodes and then transfer arcs (by using the functions to the co-limit node set). Thus, the pushout of drafts is a draft. Also, gluing can be described as a pushout. Consider a slice T. Given a sketch Σ and a node $\mathrm{w} \in \mathbb{Nd}$, the glued sketch $\Sigma^{\mathrm{w}}\mathsf{T}$ is the pushout of sketches $\Sigma + \mathrm{w}$ (cf. 3.2) and $\underline{\mathsf{T}}$ over the arcless sketch $\langle\{\mathrm{x}\};\emptyset\rangle$ under the natural morphisms ($\mu : \mathrm{x} \mapsto \mathrm{w}$ and $\nu : \mathrm{x} \mapsto \mathrm{z_T}$) as follows:



Given a slice $\mathsf{S} = \langle \underline{\mathsf{S}} : z_{\mathsf{S}} \rangle$, we obtain the glued slice $\mathsf{S}^{\mathrm{w}}\mathsf{T}$ by transferring the distinguished node z_{S} of S to the glued sketch $\underline{\mathsf{S}}^{\mathrm{w}}\mathsf{T}$: $\mathsf{S}^{\mathrm{w}}\mathsf{T} \coloneqq \langle \underline{\mathsf{S}}^{\mathrm{w}}\mathsf{T} : \mu'(z_{\mathsf{S}}) \rangle$.

One can adapt these constructions to particular categories of sketches, like the transitive ones (cf. 5.2). The co-limit sketch Σ may fail to be τ-transitive, but one can transform Σ to a τ-transitive sketch $\Sigma_{\mathtt{tr}}$, which will be a co-limit in the category of τ-transitive sketches. Transitive gluing is the pushout in the category of transitive sketches. [11] This is the gluing used in the rule $(\overset{\mathsf{T}}{\rightarrow})$ for adding a slice-arc (cf. 4.1) and in the expansion rule (Exp) (cf. 4.2). Also, the construction in 4.3 now gives a family of slices connected by homomorphisms and we use its co-limit sketch Σ to obtain the natural model $\mathfrak{N}[\Sigma]$. The deterministic case is similar.

We now examine discrimination (cf. 4.3)

The crucial result is discrimination of the co-limit sketch Σ: for every basic draft D, $\mathsf{g} : \mathsf{D} \to \mathfrak{N}[\Sigma]$ iff $\mathsf{g} : \mathsf{D} \dashrightarrow \Sigma$. This can be established by induction on the complement-complexity of a basic draft. [12] Then, since we have a morphism $\mu : \underline{\mathsf{S}} \dashrightarrow \Sigma$, we can conclude $\mu(z_{\mathsf{S}}) \in [\![\mathsf{S}]\!]_{\mathfrak{N}[\Sigma]}$, whence $[\![\mathsf{S}]\!]_{\mathfrak{N}[\Sigma]} \neq \emptyset$.

6 Concluding Remarks

We now present some remarks on our graph approach to modal logics.

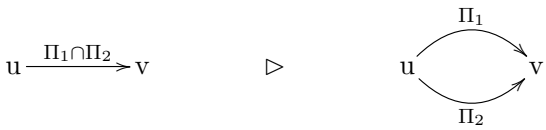
We have examined a graph approach to multi-modal logics giving correct calculi. Such formalisms, which internalize the (Kripke) semantics of modal logics, provide uniform tools for expressing and manipulating modal formulas. Soundness and completeness indicate that our rules capture the intended meanings of the concepts. The uniformity of the approach yields a flexibility that is very useful in extensions: they can be presented naturally by expressing the corresponding intuitive ideas.

We have presented the graph calculus for logic K in Section 4. In Section 5, we have shown how to extend it, in a natural manner, to handle some modalities, like the global and difference modalities (in 5.1), as well as some modalities with properties like reflexivity, symmetry, transitivity and determinism (in 5.2). This illustrates some benefits of our uniform approach: the graph rules we formulate express the intuitive ideas, which contrasts with the unorthodox rules often used [2].

Similar ideas can be used in a graph approach to PDL for structured data. This illustrates the naturalness of our approach in expressing ';'. We can also express intersection of programs (cf. [1]) simply as follows:

[11] Note that Lemma 3 (Arc addition and gluing), in 3.2 (Modal Graph Concepts and Constructions), still holds because of the pushout propery.

[12] We introduce ranks by mutual recursion. For expressions: for $\mathrm{r} \in \mathrm{R}n$, $\mathrm{r}k[\mathrm{r}] \coloneqq 0$ and, for a basic slice T, $\mathrm{r}k[\overline{\mathsf{T}}] \coloneqq \mathtt{rk}[\mathsf{T}] + 1$. For arcs: $\mathtt{rk}(\mathrm{w}|\mathrm{E}) \coloneqq \mathrm{r}k[\mathrm{E}]$ and $\mathtt{rk}(\mathrm{u}\,\alpha\,\mathrm{v}) \coloneqq 0$. For a draft $\mathsf{D} = \langle \mathsf{N}; \mathsf{A} \rangle$: $\mathtt{rk}(\mathsf{D}) \coloneqq \sum_{\mathsf{a} \in \mathsf{A}} \mathtt{rk}(\mathsf{a})$. For a slice S: $\mathtt{rk}[\mathsf{S}] \coloneqq \mathtt{rk}(\underline{\mathsf{S}})$.



As the examples in Section 2 suggest, we have finite counter-models. The model provided by the construction outlined in 4.3 is not guaranteed to be finite, but we think that we can extend the filtration method [2] to our approach, thereby obtaining finite models. In a similar vein, it seems that familiar modal logic ideas, such as bi-simulation [2], have appropriate analogues for graphs.

References

[1] Balbiani, P., Vakarelov, D.: PDL with Intersection of Programs: a Complete Axiomatization. J. Appl. Non-Classical Logics, 13, 231–276 (2003)

[2] Blackburn, P., de Rijke, M., Venema, Y. : Modal Logic. Cambridge University Press, Cambridge (2001)

[3] Curtis, S., Lowe, G.: Proofs with Graphs. Sci. Computer Progr. 26, 197–216 (1996)

[4] Freitas, R., Veloso, P. A. S., Veloso, S. R. M., Viana, P. J.: Reasoning with Graphs. In: Mints, G., de Queiroz, R.J.G.B. (eds.) ENTCS, vol. 165, pp. 201–212. Elsevier (2006)

[5] Freitas, R., Veloso, P. A. S., Veloso, S. R. M., Viana, P. J.: On Graph Reasoning. Information and Computation, 207, 1000–1014 (2009)

[6] Freitas, R., Veloso, P. A. S., Veloso, S. R. M., Viana, P. J.: A Calculus for Graphs with Complement. In: Goel, A. K., Jamnik, M., Narayanan, N. H. (eds.) Diagrams 2010, LNCS, vol. 6170, pp. 84–98. Springer (2010)

[7] MacLane, S.: Categories for the Working Mathematician (2nd edition). Springer-Verlag, Berlin (1998)

[8] Veloso, P. A. S., Veloso, S. R. M.: On Graph Refutation for Relational Inclusions. In: Rocca, S. R. D., Pimentel, E. (eds.) LSFA 2011, EPTCS, vol. 81, pp. 47–62. (2011)