Bisimulation Quantified Modal Logics: Decidability

Tim French

ABSTRACT. Bisimulation quantifiers are a natural extension of modal logics. They preserve the bisimulation invariance of modal logic, while allowing monadic second-order expressivity. Unfortunately, it is not the case that extension by bisimulation quantifiers always preserves the decidability of a modal logic. Here we examine a general class of modal logics for which decidability is preserved under extension by bisimulation quantifiers.

1 Introduction

Bisimulation quantifiers extend pure modal logics with a form of weak propositional quantification. They were introduced by Visser [21] and Ghilardi and Zawadowski [11] to semantically characterize uniform interpolants for modal logics, and they are related to Pitt's quantifiers [18] in intuitionistic propositional logic. Bisimulation quantifiers can often increase expressivity allowing us to represent monadic second-order properties not expressible in pure modal logic. They also preserve many of the intuitions of pure modal logic, such as bisimulation invariance [4].

Standard propositional quantification for modal logics was considered by Fine [6], and was shown to make several simple modal logics, such as **K**, undecidable. Bisimulation quantifiers weaken standard propositional quantification by allowing us to quantify "modulo bisimulation". That is, instead of quantifying over all interpretations of a propositional atom in a given structure, we quantify over all interpretations of a propositional atom in all bisimilar structures.

Modal logics are bisimulation invariant. That is, two bisimilar models will validate exactly the same set of modal formulas. One of the advantages of modal logic has been the natural abstraction inherent in this bisimulation invariance. A model is not required to be a precise representation of a real-world system. A model simply has to faithfully represent the salient properties of the system up to bisimulation. When we seek to extend the expressive power of modal logic, it is therefore important to preserve this bisimulation invariance. If we recognize a model to be an *abstraction* of a system, then when we quantify over the interpretation of an atom, we should quantify over that atom, not only in the given abstraction, but in all equally good abstractions. That is, we apply bisimulation quantification.

Another advantage of bisimulation quantification is that it gives us some of the expressive capabilities of the modal μ-calculus, but retains a correspondence to natural language, not evident in fixed-point operators. A bisimulation quantifier, $\exists x$, can be read as "hypothetically, x could be such that...". The bisimulation quantifier preserves all modal properties that are independent of x, so they quantify over all interpretations of x consistent with all other properties of the model.

Bisimulation quantifiers have been investigated in a number of logics. In the case of **GL**, **Grz** and **K** they have been used to characterize uniform interpolants [21, 11]. In the case of **S4** and **K4** they have been used to show that the logics do not have uniform interpolation. In [1] bisimulation quantifiers have been used to characterize uniform interpolants for the modal μ-calculus. More recently, a sound and complete axiomatization has been provided for an expressive bisimulation quantified modal logic [2] and in [3] bisimulation quantifiers are considered for transitive modal logics. Also, in [10] bisimulation quantifiers are used to characterize conservative extensions in description logics.

In [7] it was shown that bisimulation quantifiers for modal logics defined over an arbitrary class of frames may not always be well-behaved. Particularly examples were given where

1. bisimulation quantifiers may not preserve the intuitions of propositional quantification (so the standard axioms for propositional quantification were not valid),

2. bisimulation quantifiers did not preserve the decidability of the modal logic.

A sufficient condition, amalgamation, was given for classes of frames where bisimulation quantifiers validate the axioms of propositional quantification. Unfortunately, no simple condition was found that ensures extension by bisimulation quantifiers preserve the decidability of the pure logic.

In this paper we describe an interpretation for bisimulation quantifiers in various modal logics and describe a relatively general class of modal logics which is well-behaved under extension by bisimulation quantifiers.

2 Preliminaries

In this section we give the basic notation, definitions and lemmas required throughout this paper.

2.1 Pure modal logic

DEFINITION 1. Let Λ be a non-empty set. A Λ-*frame* is a tuple (S, R) where S is some set and $R : \Lambda \longrightarrow \wp(S \times S)$.

A Λ-frame (also referred to as a *Kripke frame*) is the basic relational structure of our abstraction. The set S corresponds to set of possible worlds, and for each $a \in \Lambda$, $R(a)$ refers to some relation between the possible worlds. To define a model we are required to assign atomic propositions to the set of possible worlds. A model adds a valuation to a Λ-frame which defines which atomic propositions are true at each element of S, and identifies *current world*. The current world is required since modal logic is always evaluated with respect to a single world.

DEFINITION 2. A Λ-$\mathcal{V}$-*model* is a tuple (S, R, ρ, s) where (S, R) is a Λ-frame, $\rho : S \longrightarrow \wp(\mathcal{V})$, and $s \in S$.

To make use of such a model we must define a syntax. We will assume that $\mathcal{V}$ is fixed as some countably infinite set,and make Λ explicit in the logic by referring to the following syntax as $\mathcal{ML}_\Lambda$:

(1) $\alpha \ ::= \ x \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \Diamond_a \alpha$

where $x \in \mathcal{V}$ and $a \in \Lambda$. Correspondingly, we will refer to Λ-$\mathcal{V}$-models as Λ-models when there is no ambiguity about $\mathcal{V}$. We let the abbreviations $\wedge$, $\rightarrow$, $\leftrightarrow$, $\top$ and $\bot$ be defined as usual and let $\Box_a\alpha$ abbreviate $\neg\Diamond_a\neg\alpha$. We let $var(\alpha)$ be the set of propositional atoms appearing in α.

The modal logic L_Λ (referred to as the *pure modal logic of* Λ) augments the syntax $\mathcal{ML}_\Lambda$ with the following semantics. Given some Λ-model $M = (S, R, \rho, s)$ for each $t \in S$ let $M_t = (S, R, \rho, t)$. We then define:

$$\begin{aligned}
M \models x &\iff x \in \rho(s)\\
M \models \neg\alpha &\iff M \not\models \alpha\\
M \models \alpha \vee \alpha' &\iff M \models \alpha \text{ or } M \models \alpha'\\
M \models \Diamond_a\alpha &\iff \text{for some } (s,t) \in R(a),\ M_t \models \alpha.
\end{aligned}$$

From a purely semantic view of logic, we can define a variety of modal logics by restricting the class of models over which formulas are evaluated.

DEFINITION 3. Given some set Λ, a Λ-*class* $\mathcal{C}$ is a class of Λ-frames. The logic $\mathrm{L}_\mathcal{C}$ is the set of all $\mathcal{ML}_\Lambda$ formulas which are true on all Λ-$\mathcal{V}$-models, (S, R, ρ, s), where $(S, R) \in \mathcal{C}$.

In an abuse of notation we may refer to $M \in \mathcal{C}$, where $M = (S, R, \rho, s)$ is a Λ-frame, $\mathcal{C}$ is a Λ-class and $(S, R) \in \mathcal{C}$.

2.2 The modal μ-calculus

The language, μL_Λ is recursively defined as follows:

(2) $\alpha ::= x \mid \alpha \vee \alpha \mid \Diamond_a\alpha \mid \mu x\alpha \mid \neg\alpha$

where $x \in \mathcal{V}$, $a \in \Lambda$ and in the recursion for $\mu x\alpha$ we require that x only occurs in the scope of an even number of negations in α. In addition to the usual abbreviations for propositional and temporal logic, we also let $\nu x\alpha$ be an abbreviation for $\neg\mu x\neg\alpha[x\backslash\neg x]$, (where $\alpha[x\backslash\neg x]$ is α with every free occurrence of x replaced by $\neg x$). Note in the case that Λ is a singleton we will refer to μL_Λ as μL, and we let the phrase *modal μ-calculus* refer to μL_Λ for arbitrary sets Λ.

The models of the modal μ-calculus are identical to the models of modal logic. The operators μ and ν are the fixed point operators. In $\mu x\alpha$ the operator μ refers to the least of all assignments of x such that x is true exactly where α is true, and the operator ν refers to the greatest such assignment. Given a model $M = (S, R, \rho, s)$, $x \in \mathcal{V}$ and some set $T \subseteq S$, we let the model $M^{[x \to T]} = (S, R, \tau, s)$ where for all $u \in S$

$$(3) \quad \tau(u) = \begin{cases} \rho(u) \cup \{x\} \text{ if } u \in T \\ \rho(u) \backslash \{x\} \text{ otherwise.} \end{cases}$$

Let $\| \alpha \| = \{u \mid M_u \models \alpha\}$, and for $T \subseteq S$ let $\| \alpha \|_{[x \to T]} = \{u \mid M_u^{[x \to T]} \models \alpha\}$.

Formally, for $M = (S, R, \rho, s)$, the semantics for L_μ is given by:

$$\begin{aligned} M &\models x \Longleftrightarrow x \in \rho(s) \\ M &\models \Diamond_a \alpha \Longleftrightarrow M_t \models \alpha \text{ for some } (s,t) \in R(a) \\ M &\models \mu x \alpha \Longleftrightarrow s \in \bigcap \left\{ T \subseteq S \mid \ \| \alpha \|_{[x \to T]} \subseteq T \right\} \end{aligned}$$

where the propositional operators have their usual meaning.

The modal μ-calculus was introduced by Kozen in [14]. In the same work an axiomatization was proposed, and the satisfiability problem was shown to be decidable in non-elementary time. Emerson and Jutla presented an exponential decision procedure for the modal μ-calculus in [5].

2.3 Bisimulation Quantifiers

We will first present the syntax $\mathcal{QL}_\Lambda$ for a modal logic $\mathrm{QL}_\mathcal{C}$ (where $\mathcal{C}$ is a Λ-frame) and we will then give the semantics of $\mathrm{QL}_\mathcal{C}$. Let $\mathcal{V}$ be a countable infinite set of atomic propositions and suppose that $\mathcal{C}$ is a Λ-class for some countable, non-empty set Λ. We recursively define the formulas of $\mathcal{QL}_\Lambda$ as follows:

$$(4) \quad \alpha \ ::= \ x \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \Diamond_a \alpha \mid \exists x \alpha$$

where $x \in \mathcal{V}$ and $a \in \Lambda$. We let the abbreviations $\wedge$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$ and $\forall$ be defined as usual and let $\Box_a \alpha$ abbreviate $\neg \Diamond_a \neg \alpha$.

DEFINITION 4. Let $M = (S, R, \rho, s_0)$ and $N = (T, P, \lambda, t_0)$ be Λ-$\mathcal{V}$-models and suppose $\Theta \subseteq \mathcal{V}$. We say the models M and N are Θ-*bisimilar* (written $M \cong_\Theta N$) if there is some relation $B \subseteq S \times T$ such that:

1. $(s_0, t_0) \in B$;
2. for all $(s,t) \in B$, $\rho(s) \backslash \Theta = \lambda(t) \backslash \Theta$;
3. for all $(s,t) \in B$, for all $a \in \Lambda$, for all $u \in S$ such that $(s,u) \in R(a)$ there is some $v \in T$ such that $(u,v) \in B$ and $(t,v) \in P(a)$;
4. for all $(s,t) \in B$, for all $a \in \Lambda$, for all $v \in T$ such that $(t,v) \in P(a)$ there is some $u \in S$ such that $(u,v) \in B$ and $(s,u) \in R(a)$.

We call such a relation B a Θ-*bisimulation* from M to N and if $M \cong_{\{x\}} N$ we say M and N are x-bisimilar (written $M \cong_x N$). In the case $\Theta = \emptyset$ we refer to B as a bisimulation.

The semantics of $\mathrm{QL}_{\mathcal{C}}$ interpret formulas of $\mathcal{QL}_{\Lambda}$ as follows:

$$\begin{aligned} M_s \models_{\mathcal{C}} x &\iff x \in \rho(s) \\ M_s \models_{\mathcal{C}} \Diamond_a \alpha &\iff \text{for some } (s,t) \in R(a),\ M_t \models_{\mathcal{C}} \alpha \\ M \models_{\mathcal{C}} \exists x \alpha &\iff \text{for some } \mathcal{C}\text{−model } N,\ \text{s.t. } M \cong_x N,\ N \models_{\mathcal{C}} \alpha. \end{aligned}$$

We note that the class $\mathcal{C}$ appears explicitly in the semantics, and thus has a direct effect on the meaning of formulas. For example, we might have some Λ-model M and some formula α such that for two classes, $\mathcal{C}$ and $\mathcal{D}$ $M \models_{\mathcal{C}} \alpha$ and $M \models_{\mathcal{D}} \neg\alpha$. Consider, for example, the formula $\alpha = \forall x(x \to \Diamond_a x)$, and suppose M is a $\{a\}$-model consisting of a single world related to itself by $R(a)$.

- In the class $\mathcal{C}$ of all reflexive $\{a\}$-frames we would have $M \models_{\mathcal{C}} \alpha$.
- However, in the class of all $\{a\}$-frames, $\mathcal{D}$, we have $M \models_{\mathcal{D}} \neg\alpha$, since M is $\{x\}$-bisimilar to some model (T, P, τ, t) where $x \in \tau(t)$ and $(t,t) \notin P(a)$.

This is in contrast to the case of pure modal logic, where there is one semantics for all logics, so that $L_{\mathcal{C}}$ and $L_{\mathcal{D}}$ will always agree on the meaning of a formula (i.e. whether a given formula is satisfied by a given model), but they may differ on whether a formula is valid.

The definition of bisimulation quantifiers for pure modal logic can also be extended to propositional dynamic logic (PDL). We will find the added expressivity of dynamic modalities convenient for defining decidable bisimulation quantified modal logics. We let the syntax, $\mathcal{BQDL}_{\Lambda}$, for the *bisimulation quantified dynamic logic over* $\mathcal{C}$ ($\mathrm{BQDL}_{\mathcal{C}}$, where $\mathcal{C}$ is a Λ-class) be defined by the mutual recursions

(5) $\quad \alpha \ ::= \ x \mid \alpha \vee \alpha \mid \neg\alpha \mid \langle\pi\rangle\alpha \mid \exists x\alpha$

(6) $\quad \pi \ ::= \ a \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \mid \alpha?$

where $x \in \mathcal{V}$ and $a \in \Lambda$. In the mutual recursions we refer to α as the *proposition* and π as a *program*. We let $Prog_{\Lambda}$ refer to the set of programs defined with respect to Λ. Note, this logic (restricted to the class of all Λ-frames) was referred to as **BQL** in [1]. In [8] the language of $\mathrm{BQDL}_{\mathcal{C}}$ is extended to include the converse operator in the definition of programs.

The logic $\mathrm{BQDL}_{\mathcal{C}}$ extends basic propositional logic with modalities corresponding to programs and bisimulation quantifiers. The semantic interpretation of the formulas of $\mathrm{BQDL}_{\mathcal{C}}$ requires a concurrent induction over the complexity of programs and propositions. To define modalities corresponding to programs we extend the function R to map all programs, π, to a relation over S inductively where the base case is given by $R(a)$ for $a \in \Lambda$, and:

$$R(\pi;\pi') = \{(u,w) \mid \exists v \text{ s.t. } (u,v) \in R(\pi), (v,w) \in R(\pi')\}$$

$$R(\pi \cup \pi') = R(\pi) \cup R(\pi')$$

$$R(\pi^*) = \{(u_0, u_i) | \exists u_1, \ldots, u_{i-1} \in S \text{ s.t. } \forall j < i,\ (u_j, u_{j+1}) \in R(\pi)\}$$

$$R(\alpha?) = \{(u, u) | M_u \models_{\mathcal{C}} \alpha\}$$

Since $R(\alpha?)$ is defined with respect to the interpretation of α the following inductive definition is required

$$\begin{array}{rcl} M \models_{\mathcal{C}} x & \iff & x \in \rho(s) \\ M \models_{\mathcal{C}} \alpha \vee \beta & \iff & M \models_{\mathcal{C}} \alpha \text{ or } M \models_{\mathcal{C}} \beta \\ M \models_{\mathcal{C}} \neg\alpha & \iff & M \not\models_{\mathcal{C}} \alpha \\ M \models_{\mathcal{C}} \langle\pi\rangle\alpha & \iff & \exists(s,t) \in R(\pi) \text{ s.t. } M_t \models_{\mathcal{C}} \alpha \\ M \models_{\mathcal{C}} \exists x\alpha & \iff & \text{for some } \mathcal{C}-\text{model},\ N,\ \text{s.t. } M \cong_x N,\ N \models_{\mathcal{C}} \alpha. \end{array}$$

In the remainder of this paper we will let QL_Λ (respectively, BQDL_Λ) refer to the logic $\mathrm{QL}_{\mathcal{C}}$ (respectively, $\mathrm{BQDL}_{\mathcal{C}}$) where C is the class of all Λ-frames.

3 Bisimulations in modal logic

Bisimulations were introduced in the context of process algebras by Milner [16] and Park [17]. Prior to this, an equivalent notion of the *p-relation* was given in the context of modal logic by van Benthem [4]. The concept of bisimulation is very natural in modal logic. Bisimilar models are, for the purposes of modal logic, equivalent. That is, bisimilar models will validate exactly the same set of modal formulas.

DEFINITION 5. Let P be some *property* of Λ-models, so that for every Λ-model, P is either true or false. We say P is *bisimulation invariant* if the set of models for which P is true is closed under bisimulation. Given a logic L defined over Λ-models we say L is *bisimulation invariant* if for every L-formula, α, for all L-models, M and N where $M \cong_{\mathcal{V}-var(\alpha)} N$, $M \models \alpha$ if and only if $N \models \alpha$.

It is well-known that modal logic is bisimulation invariant. Furthermore, in [4] it is shown that the properties definable in modal logic are exactly the bisimulation invariant properties that are definable in first order logic. This result has been extended by Janin and Walukiewicz [13] to show that properties definable in the modal μ-calculus, μL_Λ, are exactly the bisimulation invariant properties definable in monadic second-order logic.

LEMMA 6. *A property is expressible in the modal μ-calculus if and only if it is expressible in monadic second-order logic and it is bisimulation invariant.*

Ironically, it is not necessary that bisimulation quantified modal logics will be bisimulation invariant. In [7] a class $\mathcal{C}$ is defined such that two $\mathcal{C}$-models may be Θ-bisimilar, but still disagree on the interpretation of some formula that does not contain atoms from Θ. However, the restriction to amalgamative classes below (Section 3.2) is sufficient to ensure bisimulation invariance.

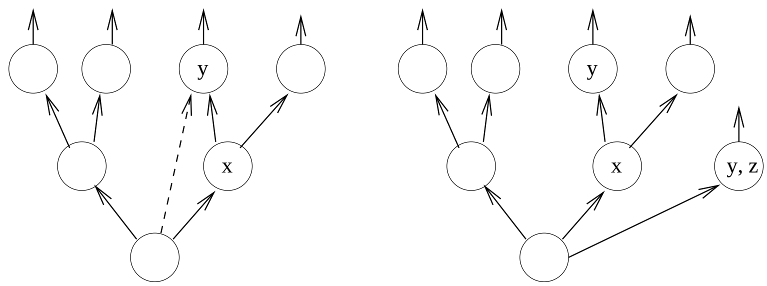

Figure 1. Two z-bisimilar $S4$-models, illustrating the validity of formula (7). The transitive arrows have been ignored, except for the salient one which is dashed.

3.1 Examples

Having established the formal definitions for bisimulation quantified modal logics, we can now examine them in more detail. An example of the interpretation of bisimulation quantifiers is given in Figure 1, where the class of frames is taken to be the class of all reflexive, transitive frames (those frames corresponding to **S4**). The figure illustrates the validity of the formula

(7) $\forall x \forall y((\neg x \wedge \Diamond(x \wedge \neg y \wedge \Diamond(y \wedge \neg x))) \rightarrow \exists z(\Diamond(z \wedge y) \wedge \Box(x \rightarrow \Box\neg z)))$

Another interesting example of the application of bisimulation quantifiers is in the logic **KD45**, which is Kripke complete for the class of serial, transitive, Euclidean frames (a frame is Euclidean if any two successors of a world are related to each other). In $\mathrm{QL}_{\mathbf{KD45}}$,

(8) $\Diamond(\exists x(x \wedge \Box\neg x))$

is a validity, but $\exists x \Diamond(x \wedge \Box\neg x)$ is a contradiction. The unsatisfiability of the second follows from the fact that $\Box(x \rightarrow \Diamond x)$ is a validity of **KD45**. Figure 2 demonstrates how formula (8) is satisfied. Here, M is a **KD45** model, and $M_t \cong_x N_{t'}$, but there is no **KD45**-model $K = (U, Q, \gamma, u)$ where $K \cong_x M$ and $K_v \cong N_{t'}$ for some v where $(u, v) \in Q$. That is, the bisimulation quantifier does not necessary commute with modalities. This may appear to be a flaw in the interpretation of bisimulation quantifiers. However, **KD45** is the logic of belief and in this context it makes sense. An agent *believes* that some set of worlds are possible, and refuses to believe any other world is possible. Suppose that we are dealing with an informed agent that knows the difference between belief and knowledge. The agent would *believe* that in principle it is *possible* that he believes some (hypothetical) property to be true, even though that property is not satisfied at the current world. That is, $\Diamond\exists x(\neg x \wedge \Box x)$ is valid. However the agent would still not concede that any of the real, non-hypothetical properties (i.e. unquantified propositions) that he believes are false. For example, I believe that whenever Australia loses a football match, the referee is at fault. I concede that it is

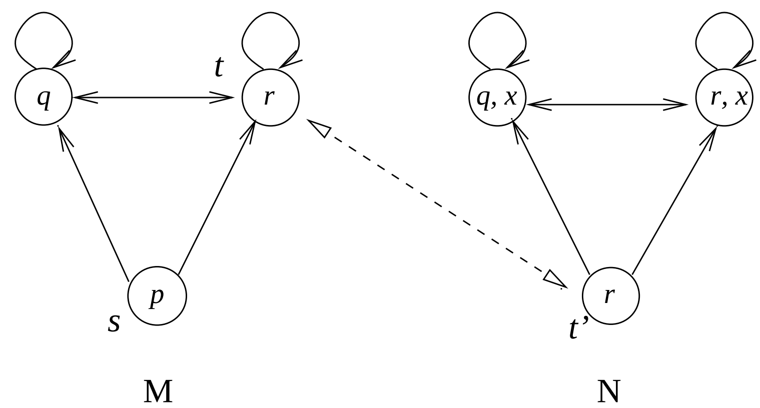

Figure 2. A depiction of the validity (8) with respect to the class of **KD45** frames.

hypothetically possible that Australia might lose a match because it is not the better team, and I still blame the refereeing. However, I'm not aware of any such case.

3.2 Amalgamation

It was shown in [7] that bisimulation quantifiers are not well behaved for all classes of frames. Particularly we can define a class of frames $\mathcal{C}$ such that $\mathrm{QL}_{\mathcal{C}}$ does not validate the standard rules of propositional quantification (see for example [18]). These rules are:

1. *Existential elimination*: If $\phi \to \psi$ is a validity and ψ does not contain free occurrences of the variable x, then $\exists x\phi \to \psi$ should also be a validity.
2. *Existential introduction*: Suppose α is a formula such that β is free for x in α. Then $\alpha[x\backslash\beta] \to \exists x\alpha$ is a validity.

Here $\alpha[x\backslash\beta]$ is the formula α with every free occurrence of the variable x replaced by the formula β, and β is *free for x in α* if and only if for every free variable, y, of β the variable x is not in the scope of a quantifier, $\exists y$, in α.

Several classes of frames that do not validate these rules are presented in [7]. These rules are sound for all logics, $\mathrm{QL}_{\mathcal{C}}$, where $\mathcal{C}$ has the *amalgamation property.*

DEFINITION 7. We say the class of frames $\mathcal{C}$ has the *amalgamation property* (or $\mathcal{C}$ is *amalgamative*) if and only if for any $\Theta_1, \Theta_2 \subset \mathcal{V}$, for any $\mathcal{C}$-models M and N such that $M \cong_{\Theta_1 \cup \Theta_2} N$, there is some $\mathcal{C}$-model, K such that $M \cong_{\Theta_1} K$ and $N \cong_{\Theta_2} K$.

A related concept of amalgamation has been studied in the context of modal algebras (see, for example, [15]). The formulation of amalgamation above appeared in Lemma 3.3 of [1], and is also presented in [3]. The same property is referred to as *safety* in [7].

LEMMA 8. *For any Λ, the class of all Λ-frames is amalgamative.*

Proof. Suppose that $M = (S, R, \rho, s)$ and $N = (T, P, \tau, t)$ are Λ-models such that $M \cong_{\Theta\cup\Gamma} N$ via the $\Theta \cup \Gamma$-bisimulation, $B \subseteq S \times T$. We construct a Λ-model $K = (U, Q, \upsilon, u)$ where:

- $U = B$;
- for all $a \in \Lambda$, $(v, w) \in Q(a)$ if and only if $v, w \in B$, $(v_1, w_1) \in R(a)$ and $(v_2, w_2) \in P(a)$ (where $v = (v_1, v_2)$ and $w = (w_1, w_2)$);
- for all $v \in B$, $\upsilon(v) = (\rho(v_1)\backslash\Theta) \cup (\tau(v_2)\backslash\Gamma)$ (where $v = (v_1, v_2)$);
- $u = (s, t)$

It is then straightforward to check that $K \cong_\Theta M$ via the Θ-bisimulation $\{(v, w) \mid v_1 = w\}$ and $K \cong_\Gamma N$ via the Γ-bisimulation $\{(v, w) \mid v_2 = w\}$. ■

The following was shown in [7].

LEMMA 9. *Given $\mathcal{C}$ enjoys amalgamation, the rules* existential elimination *and* existential introduction *are sound for* $\mathrm{QL}_\mathcal{C}$.

3.3 Decidability of BQDL_Λ

In this section we give a number of translations between the modal μ-calculus and bisimulation quantified modal logics.

From [1] we have:

LEMMA 10. *The logics* BQDL_Λ *and* μL_Λ *are expressively equivalent. Furthermore there is a computable translation in each direction that preserves the meaning of formulas. That is, for every* BQDL_Λ *formula α there is a μL_Λ formula α^μ that is satisfied by exactly the models that satisfy α.*

The translation from μL_Λ formulas to BQDL_Λ formulas is relatively straightforward and a generalization of this construction is presented below in Lemma 11. The equivalent BQDL_Λ formula is linear in the length of the original μL_Λ formula. The translation from BQDL_Λ formulas to μL_Λ formulas is not as simple. The translation is given via μ-automata [13], and the resulting μL_λ formula may be non-elementary in the length of the original BQDL_Λ formula. In [8] it is shown that there is no elementary translation from μL_Λ to BQDL_Λ.

As a corollary to Lemmas 6 and 10 we have that BQDL_Λ can express exactly the properties definable in monadic-second order logic that are bisimulation invariant, and that BQDL_Λ is decidable. An alternative proof of the decidability of BQDL_Λ (including converse operators) is presented in [8].

We can generalize one direction of Lemma 10 from the class of all Λ-frames to any class of Λ-frames that enjoys amalgamation. Since μL_Λ and $\mathrm{BQDL}_\mathcal{C}$ agree on the semantics for all common operators, it suffices to show that the fixed-point formula $\mu x\alpha$ is equivalent to some BQDL$_\mathcal{C}$ formula, α^*. The translation can then be applied recursively. More precisely, we let $(\cdot)^*$ be a transformation such that $x^* = x$, $(\Diamond_a \alpha)^* = \langle a\rangle(\alpha^*)$ and likewise for the propositional operators. We define

(9) $(\mu x.\alpha)^* = \forall x([*](\alpha^* \to x) \to x)$,

where $\langle * \rangle$ is an abbreviation for $\langle (\bigcup_{a\in\Lambda} a)^* \rangle$. We note the translation given from $\mu L_\mathcal{C}$ to BQDL$_\mathcal{C}$ is linear.

LEMMA 11. *Let $\mathcal{C}$ be any Λ-class. Then for every μL_Λ formula, α, for all $\mathcal{C}$-models M, $M \models_\mathcal{C} \alpha^*$ if and only if $M \models \alpha$.*

Proof. This lemma is proven by induction over the complexity of formulas where our induction hypothesis is that for all $\mathcal{C}$-models N we have $N \models \alpha$ if and only if $N \models_\mathcal{C} \alpha^*$. The only non-trivial case of this induction is to show that $\mu x\alpha$ and $(\mu x\alpha)^*$ have the same interpretation. First, suppose that $M \not\models_\mathcal{C} (\mu x\alpha)^*$. Then there is some $\mathcal{C}$-model $N = (T, P, \tau, t)$ such that $N \cong_x M$ and $N \models_\mathcal{C} [*](\alpha^* \to x) \wedge \neg x$. Let $U \subseteq T$ be the set $\{u \in T \mid x \in \tau(u)\}$, and let $V = \{v \in T \mid (t, v) \in P(*)\}$. Applying the induction hypothesis we have, $N \models_\mathcal{C} [*](\alpha \to x)$ so for all $v \in V$, $N_v \models \alpha \to x$. Let $N' = (V, P, \tau, t)$ (where P and τ are appropriately restricted). Clearly $N' \cong N$ so by the bisimulation invariance of μL_Λ (Lemma 6) we have for all $v \in V$, $N'_v \models \alpha \to x$. By the semantic definition (Section 2.2) of $\mu x\alpha$ it follows that for all $v \in V$, $N'_v \models \mu x\alpha$ only if $v \in U$. Since $x \notin \tau(t)$, we have $N'_v \models \neg\mu x\alpha$. Finally, since $M \cong_x N'$ and μL_Λ is bisimulation invariant it follows that $M \models \neg\mu x\alpha$.
Conversely suppose $M \models_\mathcal{C} (\mu x\alpha)^*$. Therefore, $M \models_\mathcal{C} \forall x([*](\alpha^* \to x) \to x)$ and thus for every $T \subset S$, if $M_u^{[x\to T]} \models_\mathcal{C} \alpha^*$ only for $u \in T$, then $s \in T$. By the induction hypothesis, $M_u^{[x\to T]} \models_\mathcal{C} \alpha^*$ if and only if $M_u^{[x\to T]} \models_\mathcal{C} \alpha$. Thus, for every $T \subseteq S$ where $M_u^{[x\to T]} \models \alpha$ implies $u \in T$, we have $s \in T$. By the semantic definition of $\mu x\alpha$ it follows that $M \models \mu x\alpha$. ■

Note that the other direction of Lemma 10 cannot be generalized to arbitrary, or even amalgamative classes of frames. In [3] a class of transitive well-founded frames, $GL3$, is given and it is shown that QL$_{GL3}$ is strictly more expressive than μL_{GL3}.

4 Idempotent transductions

We now investigate classes of frames, $\mathcal{C}$ such that QL$_\mathcal{C}$ is decidable and amalgamative. The classes we define, the *idempotent transduction classes*, are built by using dynamic modalities to define constraints on frames. This technique is often used for pure modal logics (see [9]), where some logic may be translated into PDL. Bisimulation quantifiers add an extra layer of complexity. It is not enough to be able to express a transformation as a set of dynamic modalities, since we also have to ensure that the transformation commutes with bisimulations.

DEFINITION 12. Given some finite set Λ, a Λ-*transduction* is a function, $\Pi : \Lambda \longrightarrow Prog_\Lambda$, where $Prog_\Lambda$ is the set of programs defined over the set of atomic programs Λ (see Section 2.3). We say a Λ-transduction, Π, is *closed* if for every $a \in \Lambda$, $\Pi(a)$ does not contain any free propositions.

A Λ-transduction induces a function that acts on Λ-frames. For every $a \in \Lambda$ the transduction $\Pi(a)$ replaces the relation $R(a)$ in a frame with the relation $R(\Pi(a))$. In this way we are able to "construct" frames satisfying certain constraints.

DEFINITION 13. Given a closed Λ-transduction, Π, and a Λ-frame $F = (S, R)$ we define $F^\Pi = (S, R^\Pi)$ where for all $u, v \in S$, $(u, v) \in R^\Pi(a)$ if and only if for some Λ-model $M = (S, R, \rho, s)$, $(u, v) \in R(\Pi(a))$ (see Section 2.3). We define the *transduction class of* Π (written $\mathcal{C}_\Pi$) to be the class of Λ-frames F^Π where F is any Λ-frame.

Note that if a program $\Pi(a)$ contains a bisimulation quantifier, the bisimulation quantifier is interpreted with respect to the class of all Λ-frames. Given a closed Λ-transduction, Π, and a Λ-model $M = (S, R, \rho, s)$ we define M^Π to be the $\mathcal{C}_\Pi$-model $(S, R(\Pi), \rho, s)$.

DEFINITION 14. A closed Λ-transduction Π is an *idempotent transduction* if for all Λ-models M, we have $(M^\Pi)^\Pi = M^\Pi$. If Π is an idempotent transduction, then $\mathcal{C}_\Pi$ is an *idempotent transduction class* of frames and the logic (Q)$\mathrm{L}_{\mathcal{C}_\Pi}$ is a *(bisimulation quantified) idempotent transduction logic.*

In [8] a weaker definition of idempotent transductions is given that only required that $(M^\Pi)^\Pi \cong M^\Pi$. This allowed the *converse operator* to appear in transductions, which increases the complexity of the required proofs of decidability, but it does allow us to define idempotent transductions corresponding to logics such as **S5** and **KD45**. It was shown that for programs not containing the converse operator, both notions of idempotence are equivalent.

Idempotence is a strong condition that ensures that bisimulations and transductions can commute. Particularly we would like to have, if $M \cong_\Theta N$, then $M^\Pi \cong_\Theta N^\Pi$. This is shown in Lemma 15. This leads to the main results of this paper: that idempotent transduction classes are amalgamative; and that idempotent transduction logics are decidable.

4.1 Examples

Let us now consider some examples of idempotent transduction logics. The pure modal logic **S4** is evaluated over frames with a single reflexive, transitive relation. We will suppose that this single modality is labelled as a. Then the class of **S4** frames can be seen to be the idempotent transduction class corresponding to the $\{a\}$-transduction $\Pi(a) = a^*$. We note that for all $\{a\}$-models, M, M^Π is transitive and reflexive. It is also easy to see that $(M^\Pi)^\Pi = M^\Pi$.

As a less trivial example of an idempotent transduction logic we consider the *Gödel Löb* logic, **GL**. The logic **GL** was shown by Segerberg [19] to be Kripke complete for all frames (S, R) where R is a Noetherian strict partial orders (i.e. all irreflexive, transitive frames containing no infinite path). Let us refer to this class of frames as GL. We can define an idempotent $\{a\}$-transduction Γ such that $\mathcal{C}_\Gamma$ is exactly the class of frames GL. This transduction is defined by

(10) $\Gamma(a) = (\forall x(x \to \langle a^*\rangle(x \wedge [a]\neg x)))?; a; a^*$

Given a model $M = (S, R, \rho, s)$, for all $u, v \in S$, we have $(u, v) \in R^\Gamma(a)$ if and only if

1. $M_u \models_\Lambda \forall x(x \to \langle a^*\rangle(x \wedge [a]\neg x))$ (it is easy to check that this is so if and only if every path beginning at u is well-founded), and

2. v is reachable from u via a path of length at least one.

Therefore, the transduction Γ will remove every pair $(u, v) \in R(a)$ where u appears in some infinite path, and take the transitive closure of the remaining pairs. An example of this transduction is given in Figure 3. We can see that the transduction Γ removes the pairs (a, b), (a, c) and (b, b) from the model M since both a and b belong to an infinite path. The transduction Γ takes the transitive closure of the remaining pairs so the pairs (c, f) and (c, g) are added to create the model M^Γ. The logic **GL** has uniform interpolation [20] and has been shown to be equivalent to QL_{GL} in [21, 11].

As a final example, we note that given any idempotent Λ-transduction, Π, we can always add the reflexive, transitive closure of all the modalities and still have an idempotent transduction. That is, we can define the $\Lambda \cup \{o\}$ transduction, Π', where $\Pi'(o) = (\bigcup_{a\in\Lambda} \Pi(a))^*$ and $\Pi'(a) = \Pi(a)$ for all $a \in \Lambda$.

4.2 Amalgamation for idempotent transductions

In this section we show that all bisimulation quantified idempotent transduction logics preserve our intuitions of propositional quantification. That is, they enjoy amalgamation (Definition 7) and thus satisfies the rules of existential elimination and existential introduction (Section 3.2). The following lemma is well-known.

LEMMA 15. *Suppose that Π is a Λ-transduction, and M and N are Λ-models such that $N \cong_\Theta M$ for some Θ where for all $a \in \Lambda$, no atom of Θ appears free in $\Pi(a)$. Then $N^\Pi \cong_\Theta M^\Pi$.*

Proof. Suppose that $M = (S, R, \rho, s)$ and $N = (T, P, \tau, t)$ are Θ-bisimilar Λ-models and $B \subseteq S \times T$ is a Θ-bisimulation from M to N. We will show that B is also a bisimulation from M^Π to N^Π by induction over the complexity of programs. Suppose that for all sub-programs π' of π we have, for all $(u, u') \in B$,

1. for all $v \in S$ where $(u, v) \in R(\pi')$, there is some $v' \in T$ where $(u', v') \in P(\pi')$ and $(v, v') \in B$, and

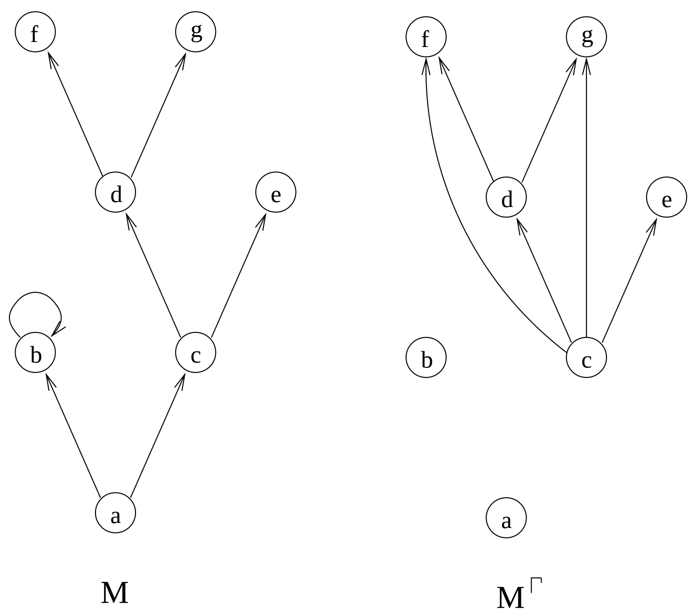

Figure 3. An example of how the transduction Γ defines the class of frames GL.

2. for all $v' \in T$ where $(u', v') \in P(\pi')$, there is some $v \in S$ where $(u, v) \in R(\pi')$ and $(v, v') \in B$.

Then we show by induction that for all $(u, u') \in B$,

Zig: for all $v \in S$ where $(u, v) \in R(\pi)$, there is some $v' \in T$ where $(u', v') \in P(\pi)$ and $(v, v') \in B$, and

Zag: for all $v' \in T$ where $(u', v') \in P(\pi)$, there is some $v \in S$ where $(u, v) \in R(\pi)$ and $(v, v') \in B$.

The inductive cases are for the *Zig* clauses below, and the proofs for *Zag* are symmetric.

1. Let $\pi = \pi_1; \pi_2$. Then if $(u, u') \in B$ and $(u, v) \in R(\pi)$, there is some $w \in S$ such that $(u, w) \in R(\pi_1)$ and $(w, v) \in R(\pi_2)$. By the inductive hypothesis there exists $w' \in T$ where $(u', w') \in P(\pi_1)$ and $(w, w') \in B$. Applying the induction hypothesis a second time there is some $v' \in T$ such that $(w', v') \in P(\pi_2)$ and $(v, v') \in B$. Therefore, there is some $v' \in T$ such that $(u', v') \in P(\pi)$ and $(v, v') \in B$, as required.

2. Let $\pi = \pi_1 \cup \pi_2$. Then if $(u, u') \in B$ and $(u, v) \in R(\pi)$, there either $(u, v) \in R(\pi_1)$ or $(u, v) \in R(\pi_2)$. Suppose without loss of generalization, that $(u, v) \in R(\pi_1)$. Then by the induction hypothesis, there

must be some $v' \in T$ such that $(u', v') \in P(\pi_1)$ and $(v, v') \in B$. Therefore, there is some $v' \in T$ such that $(u', v') \in P(\pi)$ and $(v, v') \in B$, as required.

3. Let $\pi = \pi_1^*$. Then if $(u_0, u_0') \in B$ and $(u_0, v) \in R(\pi)$, for some $k \geq 0$ there exists $u_1, \ldots, u_k \in S$ such that for all $i < k$, $(u_i, u_{i+1}) \in R(\pi_1)$ and $u_k = v$. By the induction hypothesis, there exists $u_1' \in T$ such that $(u_0', u_1') \in P(\pi_1)$ and $(u_1, u_1') \in B$. Likewise, we can show there exists $u_2' \in T$ such that $(u_1', u_2') \in P(\pi_1)$ and $(u_2, u_2') \in B$, and by induction it follows that there exists $u_1', \ldots, u_k' \in T$ where for all $i < k$, $(u_i', u_{i+1}') \in P(\pi_1)$ and for all $i \leq k$, $(u_i, u_i') \in B$. By the definition of $P(\pi)$ we have $(u_0', u_k') \in P(\pi)$ and $(v, u_k') \in B$.

4. Suppose $\pi = \alpha?$. Then if $(u, u') \in B$ and $(u, v) \in R(\pi)$, we have $v = u$ and $M_u \models \alpha$. Since $M_u \cong_\Theta N_{u'}$ and no atom from Θ appears free in α, we have $N_{u'} \models \alpha$ (as BQDL_Λ is bisimulation invariant). Therefore $(u', u') \in P(\pi)$, and $(u, u') \in B$ as required.

We can apply this inductive argument for all programs $\Pi(a)$ where $a \in \Lambda$. It follows from the definitions that for any $(u, u') \in B$, for all $v \in S$, if $(u, v) \in R^\Pi(a)$, we must have $(u, v) \in R(\Pi(a))$. By the reasoning above, there is some $v' \in T$ such that $(u', v') \in P(\Pi(a))$ (so also $(u', v') \in P^\Pi(a)$) and $(v, v') \in B$. As the *Zag* case is symmetric we have shown

- $(s, t) \in B$, and for all $(u, v) \in B$, $\rho(u)\backslash\Theta = \tau(v)\backslash\Theta$ (since B is a bisimulation from M to N),
- (Zig) for all $(u, u') \in B$, for all $v \in S$ where $(u, v) \in R^\Pi(a)$, there is some $v' \in T$ such that $(u', v') \in P^\Pi(a)$ and $(v, v') \in B$, and
- (Zag) for all $(u, u') \in B$, for all $v' \in T$ where $(u', v') \in P^\Pi(a)$, there is some $v \in S$ where $(u, v) \in R^\Pi(a)$ and $(v, v') \in B$.

Therefore B is a Θ-bisimulation from M^Π to N^Π. ■

An equivalent result was first shown by Hollenberg [12]. In addition, Hollenberg found that the set of all binary relations that are definable in monadic second-order logic and preserve bisimulation (the *bisimulation-safe fragment of MSO*) are exactly the binary relations that can be described by the programs of BQDL_Λ (Section 2.3). Lemma 15 does not use the full bisimulation-safe fragment of MSO as the programs $\Pi(a)$ cannot contain any free variables, and the notion of bisimulation is generalized to Θ-bisimulation, for some set of atoms Θ. Lemma 15 is applied below to show that idempotent transduction classes are amalgamative, and idempotent transduction logics are decidable.

LEMMA 16. *Every idempotent transduction class is amalgamative.*

Proof. Suppose that Π is an idempotent Λ-transduction, $M = (S, R, \rho, s)$ and $N = (T, P, \tau, t)$ are $\mathcal{C}_\Pi$-models and $M \cong_{\Theta\cup\Gamma} N$. Since M and N are

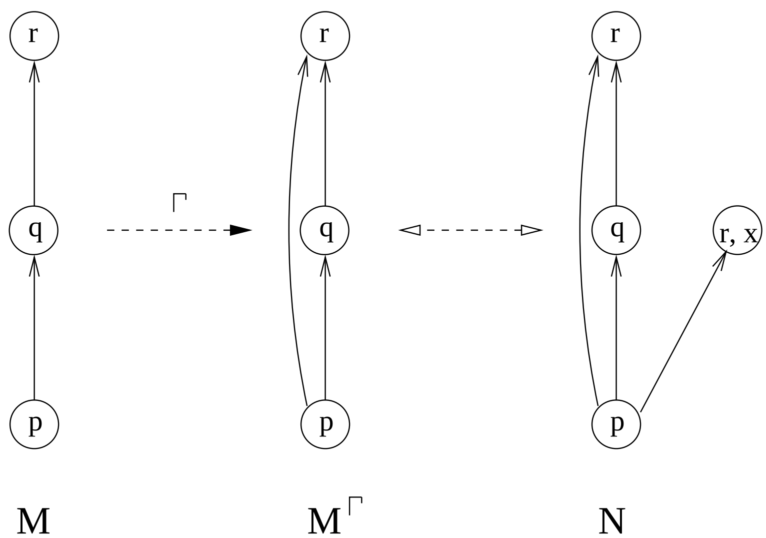

Figure 4. An example of the transduction for $K4$, Γ. We note that there is no Λ-model, K, x-bisimilar to M such that K^Γ is bisimilar to N.

Λ-models and the class of all Λ-models is amalgamative (Lemma 8) there is some Λ-model K such that $M \cong_\Theta K$ and $K \cong_\Gamma N$. By Lemma 15 and the fact that Π is idempotent,

$$M = M^\Pi \cong_\Theta K^\Pi \cong_\Gamma N^\Pi = N,$$

so $\mathcal{C}_\Pi$ is amalgamative. ■

4.3 Decidability for idempotent transduction logics

In this section we apply the results of Sections 3.3 and 4.2 to show the decidability of all bisimulation quantified idempotent transduction logics. To do this we define a translation taking a QL_Λ formula α to a BQDL_Λ formula, α^Π, such that for all Λ-models, M, $M \models_\Lambda \alpha^\Pi$ if and only if $M^\Pi \models_{\mathcal{C}_\Pi} \alpha$. Therefore the satisfiability problem for $\mathrm{QL}_{\mathcal{C}_\Pi}$ reduces to the satisfiability problem for BQDL_Λ which is known to be decidable (see [8] or [1]).

Before we proceed to this quite complicated translation, we will first see why the more obvious translation (that simply replaces modalities $\Diamond_a$ with programs $\langle\Pi(a)\rangle$) will not work. Consider the set of all transitive Λ-frames $K4$, where $\Lambda = \{a\}$. It is not hard to see that this set of frames is equal to $\mathcal{C}_\Gamma$ where Γ is the idempotent Λ-transduction defined by, $\Gamma(a) = a; a^*$. In Figure 4 we see how the Λ model M is modified by Γ. We can also see that

(11) $M^\Gamma \models_{\mathcal{C}_\Gamma} \exists x(\Diamond_a(x \wedge r) \wedge \Box_a(q \to \Box_a \neg x))$

as the $K4$ model N satisfies the unquantified fragment of the formula, and N and M^Γ are clearly x-bisimilar. However, it is not hard to show that

(12) $M \not\models_\Lambda \exists x(\langle a; a^*\rangle(x \wedge r) \wedge [a; a^*](q \to [a; a^*]\neg x))$.

We can see that there is no Λ-model K that is x-bisimilar to M such that K^Γ is bisimilar to N.

Naively replacing every modality $\Diamond_a$ with the program $\langle\Pi(a)\rangle$ is not sufficient to encode the satisfiability problem for $\mathrm{QL}_{\mathcal{C}_\Pi}$. However, the translation can encode the satisfiability problem for a related class.

DEFINITION 17. Given an idempotent Λ-transduction, Π, let $\overline{\Lambda}$ be the set $\{a, a' \mid a \in \Lambda\}$ where for each $a \in \Lambda$, a' is a new distinct element . We define the *rigid class of* Π to be the $\overline{\Lambda}$-class of frames (S, R) where for all $a \in \Lambda$, $R(a) = R(\Pi(a'))$. We will denote this class as $\mathcal{D}_\Pi$.

The rigid class of an idempotent transduction still allows us to define new modalities corresponding to Λ-programs, but it retains the original modalities that are used to define the Λ-programs. This removes a lot of the flexibility associated with the bisimulation quantifiers, and makes the satisfiability problem a lot easier. Given the QL_Λ formula α, let $\alpha[\Lambda\backslash\Pi(\Lambda)$ be the BQDL_Λ formula that results when every occurrence of any subformula $\Diamond_a\gamma$ is replaced by $\langle\Pi(a)\rangle\gamma$. Likewise for every BQDL_Λ formula α we can define $\alpha[\Lambda\backslash\Pi(\Lambda)]$ to be the BQDL_Λ formula α where every occurrence of a in every program in α is replaced by $\Pi(a)$.

LEMMA 18. *Let* Π *be an idempotent* Λ*-transduction and let* $M = (S, R, \rho, s)$ *be a* Λ*-model. We define the* $\mathcal{D}_\Pi$ *model* $M^{\overline{\Pi}} = (S, \overline{R}, \rho, s)$ *where for all* $a \in \Lambda$, $\overline{R}(a) = R(\Pi(a))$ *and* $\overline{R}(a') = R(a)$. *Then for every* QL_Λ *(*BQDL_Λ*) formula* α,

$$M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \alpha \iff M \models_\Lambda \alpha[\Lambda\backslash\Pi(\Lambda)]$$

Proof. The proof is given by induction over the complexity of formulas. The cases for propositional atoms and propositional operators are trivial, so suppose that for all models M, $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \beta$ if and only if $M \models_\Lambda \beta[\Lambda\backslash\Pi(\Lambda)]$. If $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \Diamond_a\beta$, then there is some $t \in S$ such that $(s, t) \in R(\Pi(a))$ and $M_t^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \beta$. By the induction hypothesis, $M_t \models_\Lambda \beta[\Lambda\backslash\Pi(\Lambda)]$, and as $(s, t) \in R(\Pi(a))$ we have $M \models_\Lambda (\Diamond_a\beta)[\Lambda\backslash\Pi(\Lambda)]$. Conversely, if $M \models_\Lambda (\Diamond_a\beta)[\Lambda\backslash\Pi(\Lambda)]$, then there is some $(s, t) \in R(\Pi(a))$ such that $M_t \models_\Lambda \beta[\Lambda\backslash\Pi(\Lambda)]$. From the induction hypothesis and the definition of $M^{\overline{\Pi}}$ it follows that $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \Diamond_a\beta$. In the case of BQDL_Λ formulas, (where we have programs rather than modalities) a similar proof can be applied. The details are left to the reader.

If $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \exists x\beta$, then there is some $\mathcal{D}_\Pi$-model $N = (T, P, \tau, t)$ such that $N \cong_x M^{\overline{\Pi}}$ and $N \models_{\mathcal{D}_\Pi} \beta$. Let $K = (T, P', \tau, t)$ be the Λ-model where for all $a \in \Lambda$, $P'(a) = P(a')$. It follows from Definition 4 that if B is an x-bisimulation from N to $M^{\overline{\Pi}}$ then B is also an x-bisimulation from K to M, so $K \cong_x M$. Also, since $K^{\overline{\Pi}} = N$ we can apply the induction hypothesis to show $K \models_\Lambda \beta[\Lambda\backslash\Pi(\Lambda)]$. Therefore $M \models_\Lambda (\exists x\beta)[\Lambda\backslash\Pi(\Lambda)]$ as required. Conversely, if $M \models_\Lambda (\exists x\beta)[\Lambda\backslash\Pi(\Lambda)]$, then there is some Λ-model N such that

$M \cong_x N$ and $N \models \beta[\Lambda\backslash\Pi(\Lambda)$. By the same argument used in Lemma 15, if $M \cong_x N$, then $M^{\overline{\Pi}} \cong_x N^{\overline{\Pi}}$. Applying the induction hypothesis we have $N^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \beta$, so $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} \exists x\beta$, completing the proof. ■

Lemma 18 can be applied to show any logic $\mathrm{QL}_{\mathcal{D}_\Pi}$ is decidable where Π is idempotent. This is not a surprising result as translating formulas of $\mathrm{QL}_{\mathcal{D}_\Pi}$ to BQDL_Λ is a trivial exercise. By combining Lemma 18 with Lemmas 10 and 11 we get a much stronger result. Specifically, we already know of the translations: $(\cdot)^\mu$ (from Lemma 10) that translates a BQDL_Λ formula to an equivalent μL_Λ formula; and $(\cdot)^*$ (from Lemma 11) that translates a μL_Λ formula to an equivalent BQDL_Λ formula. These two translations are the inverse of each other, so it follows that $(\alpha^\mu)^*$ is simply equivalent to α in BQDL_Λ. However, when we consider $\mathrm{BQDL}_{\mathcal{C}}$ for different classes of $\mathcal{C}$ the equivalence does not necessarily hold. Rather, we have the following lemma.

LEMMA 19. *Let $\mathcal{C}$ be a Λ-class, suppose that α is some formula of* BQDL_Λ*, and let the translations $(\cdot)^\mu$ and $(\cdot)^*$ be the translations described in Lemma 10 and Lemma 11 respectively. Then for any $\mathcal{C}$-model M,*

$$M \models_{\mathcal{C}} (\alpha^\mu)^* \iff M \models_\Lambda \alpha.$$

Proof. From Lemma 11 we have $M \models_{\mathcal{C}} (\alpha^\mu)^*$ if and only if $M \models \alpha^\mu$. We note here that α^μ is a μL_Λ formula and consequently it is independent of the Λ-class $\mathcal{C}$. From Lemma 10 we have $M \models \alpha^\mu$ if and only if $M \models_\Lambda \alpha$. Therefore $M \models_{\mathcal{C}} (\alpha^\mu)^*$ if and only if $M \models_\Lambda \alpha$. ■

The complexity of Lemma 19 is hidden in the proof of Lemma 10. The result is quite powerful, allowing us to syntactically "unwind" any $\mathcal{C}$-model into a Λ-model. That is, we can consider the interpretation of a formula independent of the restraints of the Λ-class $\mathcal{C}$. Furthermore, the translation $(\alpha^\mu)^*$ is independent of the class $\mathcal{C}$. We let $(\cdot)^{\mu*}$ refer to the composition of the translations $(\cdot)^\mu$ and $(\cdot)^*$ (so that $\alpha^{\mu*} = (\alpha^\mu)^*$).

We define the translation $(\cdot)^\Pi$ from QL_Λ to BQDL_Λ inductively as follows

1. $x^\Pi = x$, where x is an atomic proposition;
2. $(\alpha \vee \beta)^\Pi = \alpha^\Pi \vee \beta^\Pi$;
3. $(\neg\alpha)^\Pi = \neg(\alpha^\Pi)$;
4. $(\Diamond_a \alpha)^\Pi = \langle\Pi(a)\rangle\alpha^\Pi$;
5. $(\exists x\alpha)^\Pi = ((\exists x(\alpha^\Pi))^{\mu*}[\Lambda\backslash\Pi(\Lambda)]$.

We will first examine how the translation $(\cdot)^\Pi$ works with respect to the example of Figure 4. We let Γ be the transduction for $K4$ defined as before, and suppose $\alpha = \Diamond_a(x \wedge r) \wedge \Box_a(q \to \Box_a\neg x)$. Given the model M, to decide if $M^\Gamma \models_{\mathcal{C}_\Gamma} \exists x\alpha$ we ask whether there is any $\overline{\Lambda}$-model N that is both

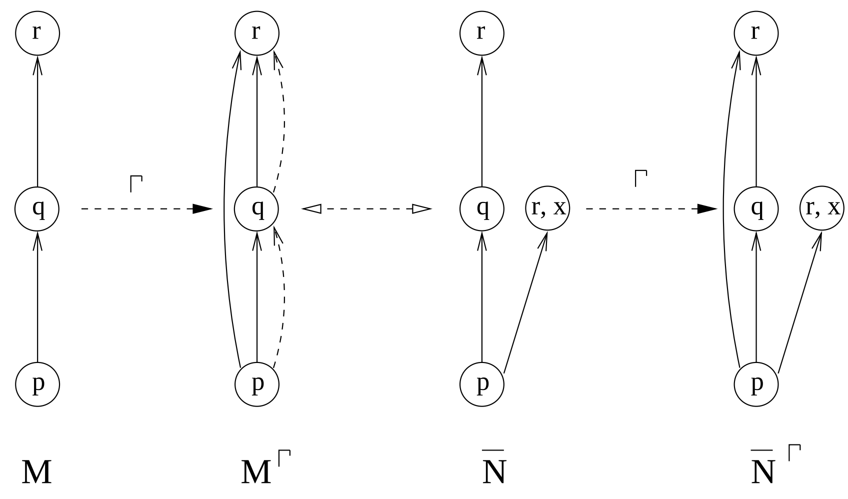

Figure 5. An example of models used to evaluate the translation $(\cdot)^\Gamma$ applied to the idempotent transduction logic $\mathrm{QL}_{\mathcal{C}_\Gamma}$. The dashed arrows correspond to the modality a'.

x-bisimilar to $M^{\overline{\Gamma}}$ and that satisfies α^Γ. By Lemma 20 (below) this implies $\overline{N}^\Gamma \models_{\mathcal{C}_\Gamma} \alpha$ (where $\overline{N}$ is the model N restricted to the modality a.) Finally, it can be shown that by the idempotence of Γ, $M^\Gamma \cong_x \overline{N}^\Gamma$ as required. This is represented in Figure 5.

LEMMA 20. *Let Π be an idempotent Λ-transduction and let M be a Λ-model. Then for every formula α of* QL_Λ,

$$M^\Pi \models_{\mathcal{C}_\Pi} \alpha \iff M \models_\Lambda \alpha^\Pi.$$

Proof.

We are required to show $M \models_\Lambda \alpha^\Pi$ if and only if $M^\Pi \models_{\mathcal{C}_\Pi} \alpha$. Proceeding by induction over the complexity of α, it is trivial to see that the induction is correct for the first three cases cases (atoms, disjunctions and negations).

For the case of modalities, suppose for all models $M = (S, R, \rho, s)$, $M^\Pi \models_{\mathcal{C}_\Pi} \beta$ if and only if $M \models_\Lambda \beta^\Pi$. Then if $M^\Pi \models_{\mathcal{C}_\Pi} \Diamond_a \beta$, for some $t \in S$ where $(s,t) \in R^\Pi(a)$ we have $M_t^\Pi \models_{\mathcal{C}_\Pi} \beta$. By the induction hypothesis it follows $M_t \models_\Lambda \beta^\Pi$ and by the definition of M^Π, we have $(s,t) \in R(\Pi(a))$. Therefore, $M \models_\Lambda \langle \Pi(a) \rangle \beta^\Pi$. Conversely, if $M \models_\Lambda \langle \Pi(a) \rangle \beta^\Pi$, then there is some $t \in S$ where $(s,t) \in R(\Pi(a))$ and $M_t \models \beta^\Pi$. By the induction hypothesis, $(M_t)^\Pi \models_{\mathcal{C}_\Pi} \beta$. Clearly $(M_t)^\Pi = (M^\Pi)_t$, so since $(s,t) \in R(\Pi(a)) = R^\Pi(a)$ we have $M^\Pi \models_{\mathcal{C}_\Pi} \Diamond_a \beta$.

We are now left to deal with bisimulation quantifiers. Assume inductively that $M \models_\Lambda \beta^\Pi$ if and only if $M^\Pi \models_{\mathcal{C}_\Pi} \beta$. We will prove that $M \models_\Lambda (\exists x \beta)^\Pi$

if and only if $M^\Pi \models_{\mathcal{C}_\Pi} \exists x\beta$ through a chain of equivalences. The first equivalence is

(13) $M^\Pi \models_{\mathcal{C}_\Pi} \exists x\beta \iff M^\Pi \models_\Lambda \exists x(\beta^\Pi)$.

To show this, first assume that $M^\Pi \models_{\mathcal{C}_\Pi} \exists x\beta$. Then there is some $\mathcal{C}_\Pi$-model, N such that $N \models_{\mathcal{C}_\Pi} \beta$ and $N \cong_x M^\Pi$. Since N is a $\mathcal{C}_\Pi$-model, we have $N^\Pi = N$, so $N^\Pi \models_{\mathcal{C}_\Pi} \beta$ and by the induction hypothesis $N \models_\Lambda \beta^\Pi$. From this it is immediate that $M^\Pi \models_\Lambda \exists x(\beta^\Pi)$.

Conversely, suppose that $M^\Pi \models_\Lambda \exists x(\beta^\Pi)$. Then for some Λ-model N with $N \cong_x M^\Pi$ we have $N \models_\Lambda \beta^\Pi$. By the induction hypothesis we obtain $N^\Pi \models_{\mathcal{C}_\Pi} \beta$. Also by Lemma 15 we have $N^\Pi \cong_x (M^\Pi)^\Pi = M^\Pi$, so $M^\Pi \models_{\mathcal{C}_\Pi} \exists x\beta$. This proves (13).

Since only the modalities from Λ appear in the formula β^Π it follows from the semantics of BQDL_Λ that

(14) $M^\Pi \models_\Lambda \exists x(\beta^\Pi) \iff M^{\overline{\Pi}} \models_{\overline{\Lambda}} \exists x(\beta^\Pi)$.

Applying Lemma 19 to the $\overline{\Lambda}$-class, $\mathcal{D}_\Pi$ we have

(15) $M^{\overline{\Pi}} \models_{\overline{\Lambda}} \exists x(\beta^\Pi) \iff M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} (\exists x(\beta^\Pi))^{\mu *}$.

From [1] we know that $(\exists x(\beta^\Pi))^{\mu *}$ will be a formula of BQDL_Λ so by applying Lemma 18 we have

(16) $M^{\overline{\Pi}} \models_{\mathcal{D}_\Pi} (\exists x(\beta^\Pi))^{\mu *} \iff M \models_\Lambda (\exists x(\beta^\Pi))^{\mu *})[\Lambda \backslash \Pi(\Lambda)]$

From the equivalences (13), (14), (15) and (16) we have $M^\Pi \models_{\mathcal{C}_\Pi} \exists x\beta$ if and only if $M \models_\Lambda (\exists x\beta)^\Pi$, as required. This completes the proof. ■

As a corollary to Lemma 20 we have main theorem of this paper.

THEOREM 21. *Given any idempotent* Λ*-transduction,* Π*,* $\mathrm{QL}_{\mathcal{C}_\Pi}$ *is decidable.*

To decide whether a formula α of $\mathrm{QL}_{\mathcal{C}_\Pi}$ is satisfiable (where Π is idempotent), we simply compute α^Π and apply a decision procedure for BQDL_Λ. (See for example [1, 8]).

5 Conclusion

Here we have just begun to investigate the range of logics for which bisimulation quantifiers can meaningfully applied. In [8] the class of idempotent transductions is extended to allow for programs containing the converse operator. This allows us to define idempotent transduction classes corresponding to logics such as **KD45** and **S5**. However, it is shown that even with the converse operator some simple logics, such as **Grz** are not definable using idempotent transductions. In future work we will examine further generalizations of the idempotent transduction classes.

Acknowledgements: The author would like to thank Mark Reynolds, Yde Venema, Ian Hodkinson and the anonymous reviewers for their many helpful comments and suggestions.

BIBLIOGRAPHY

[1] G. D'Agostino and M. Hollenberg. Logical questions concerning the μ-calculus: interpolation, Lyndon and Łos-Tarski. *J. Symb. Log.*, 65(1):310–332, 2000.

[2] G. D'Agostino and G. Lenzi. An axiomatization of bisimulation quantifiers via the μ-calculus. *Theor. Comput. Sci.*, 338:64–95, 2005.

[3] G. D'Agostino and G. Lenzi. Bisimulation quantifiers and fixed points over K4, 2005. unpublished.

[4] J. van Benthem. Correspondence theory. *Handbook of Philosophical Logic*, 2:167–247, 1984.

[5] E. A. Emerson and C. S. Jutla. Tree automata, μ-calculus and determinacy. In *Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 368–377, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[6] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36:336–346, 1970.

[7] T. French. Bisimulation quantified logics: undecidability. In *Proceedings of FSTTCS '05*, volume 3821 of *LNCS*, pages 396–407, 2005.

[8] T. French. *Bisimulation quantifiers for modal logic*. PhD thesis, The University of Western Australia, 2006. submitted, available from `http://people.csse.uwa.edu.au/tim/`.

[9] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many Dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.

[10] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR2006)*, Lake District, UK, 2006. To appear.

[11] S. Ghilardi and M. Zawadowski. Undefinability of propositional quantifiers in the modal system S4. *Studia Logica*, 55:259–271, 1995.

[12] M. Hollenberg. *Logic and Bisimulation*. PhD thesis, University of Utrecht, 1998.

[13] D. Janin and I. Walukiewicz. Automata for the modal μ-calculus and related results. In *MFCS '95: Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science*, pages 552–562, London, UK, 1995. Springer-Verlag.

[14] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27, 1983.

[15] L. Maksimova. Craig's theorem in superintuitionistic logics and amalgamable varieties of pseudoboolean algebras. *Algebra Logika*, 16(6):643–681, 1977.

[16] R. Milner. A calculus of communicating systems. *Lecture Notes in Computer Science*, 92, 1980.

[17] D. Park. Concurrency and automata on infinite sequences. *Lecture Notes in Computer Science*, 104:167–183, 1981.

[18] A. M. Pitts. On an interpretation of second order quantification in first order intuitionistic propositional logic. *Jour. Symbolic Logic*, 57:33–52, 1992.

[19] K. Segerberg. An essay in classical modal logic. *Philosophical Studies*, 13, 1971.

[20] V. Yu. Shavrukov. *Subalgebras of diagonalizable algebras of theories containing arithmetic*. Dissertationes Mathematicae CCCXXIII. Polska Akademia Nauk, Mathematical Institute, Warszawa, 1993.

[21] A. Visser. Uniform interpolation and layered bisimulation. In *Gödel '96*, volume 6 of *Lecture Notes in Logic*, pages 139–164, 1996.

Tim French

School of Computer Science and Software Engineering,

The University of Western Australia

tim@csse.uwa.edu.au