Simulation Logic

Gerard Allwein
Naval Research Laboratory
Code 5543
Washington, DC 20375, U.S.A.
gerard.allwein@nrl.navy.mil

William L. Harrison
Dept. of Computer Science
University of Missouri
Columbia, Missouri, U.S.A.
harrisonwl@missouri.edu

David Andrews
Dept. of Computer Science and Computer Engineering
University of Arkansas
Fayetteville, Arkansas 72701
dandrews@uark.edu

Abstract

Simulation relations have been discovered in many areas: Computer Science, philosophical and modal logic, and set theory. However, the simulation condition is strictly a first-order logic statement. We extend modal logic with modalities and axioms, the latter's modeling conditions are the simulation conditions. The modalities are normal, i.e., commute with either conjunctions or disjunctions and preserve either Truth or Falsity (respectively). The simulations are considered arrows in a category where the objects are descriptive, general frames. One can augment the simulation modalities by axioms for requiring the underlying modeling simulations to be bisimulations or to be p-morphisms. The modal systems presented are multi-sorted and both sound and complete with respect to their algebraic and Kripke semantics.

1 Introduction

A beneficial property of simulation relations is their ability to allow the transfer of modal formulas between logic representations while preserving validity. Simulation relations satisfy a model theoretic condition; the latter is expressed as a quantified first-order logic statement which underwrites the transfer. A problem arises in that the simulation condition is strictly a meta-logic condition. Thus, the original logics involved in the transfer cannot take advantage of this condition. The consequence is that the model theoretic condition restricting the modal logics has no visibility in the logics. This runs counter to the notion that model theoretic restrictions or features should answer to some feature of the logics being modeled. Simulation logic lifts the transfer from a model theoretic phenomenon to a modal logic phenomenon and does not require any quantifiers, thus providing the ability to use simulation reasoning at the modal logical level without having to drop down into the semantic meta-logic. Additionally, modal axioms can be added to further constrain the underlying condition, say, to force it to be a function or p-morphism or bisimulation. This provides the advantage of expressing simulations such that the modal logic supporting the simulations can be made multi-sorted; each sort is a "local logic". The local logics may each be different

and hence targeted at a particular domain. The simulations, via simulation operators, provide the ability to transfer logical statements among the domains.

A simulation relation $\mathcal{R}$ is a binary relation between two relations $\mathcal{H}$ and $\mathcal{K}$ in the following form:

$$\mathcal{R}xx' \text{ and } \mathcal{H}xy \text{ implies for some } y'(\mathcal{K}x'y' \text{ and } \mathcal{R}yy'),$$

If we consider $\mathcal{H}$ as a relation on a set X and $\mathcal{K}$ as a relation on a set Y then $\mathcal{R}$ can be seen to be a mapping, or more accurately a *morphism*, from $(X, \mathcal{H})$ to $(Y, \mathcal{K})$. Thinking of these latter two tuples as Kripke frames, and seeing as $\mathcal{R}$ is a relation, it is natural to ask whether $\mathcal{R}$ can be used to interpret a modal operator which connects two logics, one whose interpretation involves $(X, \mathcal{H})$ and one whose interpretation involves $(Y, \mathcal{K})$. The simulation condition allows necessary modal formulas holding at $(Y, \mathcal{K})$ to be translated back along the simulation relation to hold at $(X, \mathcal{H})$. However, this translation occurs at the meta-logic level of the semantics. Simulation logic lifts or formalizes this situation so that it can be stated in the object language of a logic through a *simulation axiom*. The axiom's validation condition is precisely the definition of a simulation relation.

Typically, one would also require the propositional atoms for both logics related by a simulation axiom to evaluate the same in the frame models. This is known as the *totality condition* and amounts to identifying the classical propositional fragments of the modal logics at either end of a simulation. We leave this element to a separate axiom that can either be added or left off depending upon the application.

A simulation logic is a collection of modal logics, not necessarily all of the same type, connected by simulation operators whose interpretation will be handled by simulation relations. One adds axioms for the simulation operators, in addition to the simulation axiom, to specify additional model theoretic properties of the interpreting morphisms. We chose simulation relations as arrows, and hence the term simulation logic, because simulation relations work so well with modal logics. We believe the general idea can be applied to many different kinds of logics by suitably changing the axioms although the axioms will still remain modal axioms in essence.

Simulations, and their symmetric kin *bisimulations*, were discovered independently within Computer Science, philosophical logic and set theory [**?**]. Bisimulations capture a strong notion of behavioral equivalence. Two systems are bisimilar precisely when their observable behavior is identical. In the context of labelled transition systems, for example, a relation $\mathcal{R}$ between systems P and Q is a bisimulation if, and only if, two conditions hold: (1) for all labels μ and states p and q of P and Q, respectively, $p\,\mathcal{R}\,q$ and $p \xrightarrow{\mu} p'$ implies there is a Q-state q' such that $p'\,\mathcal{R}\,q'$ and its "converse", (2) $p\,\mathcal{R}\,q$ and $q \xrightarrow{\mu} q'$ implies there is a P-state p' such that $p'\,\mathcal{R}\,q'$. The label μ may be an action, in the case of computer science oriented systems, or a Kripke relation in the case of modal frames. The relation $\mathcal{R}$ is a simulation of P in Q if, and only if, (1) holds. Thus any bisimulation is a simulation (while the converse does not hold).

Within theoretical Computer Science, bisimulation and refinements to it have received considerable attention; simulation has received relatively little attention by comparison (although some study has occurred [**?**]; see Sangiorgi [**?**], Chapter 6, for an overview).

Simulations, we believe, can have a useful role in computer security in particular. This motivates their study in this paper. Consider, for instance, the following intuitive scenario. Let P be a program in a procedural programming language like C and let M be the machine code implementing P produced by a compiler. "M simulates P" is tantamount to a statement of correctness for the

compiler. If M performing an operation of P reaches state m, there must be a corresponding semantic state of P, say s, related to m in the simulation relation.

If one wishes to distinguish secure systems from insecure ones meaningfully, one must admit the expression of insecure systems into the domain of discourse and the relative weakness of simulation compared to bisimulation is an advantage. A buffer overflow exploit [**?**] occurs when the argument to a procedure call, $proc(arg)$, exceeds a certain size threshold, making it possible for a malicious agent to inject and then execute arbitrary machine instructions encoded within arg. While the mechanics underlying such exploits need not concern us here, it should be noted that, were M bisimilar to P, such exploits would not be possible because the behavior of implementation of $proc(arg)$ would necessarily be identical to its behavior according to the language semantics.

From a strictly logical perspective, simulation logic can be seen as an extension to partially ordered modalities [**?**]. As an example of partially ordered modalities, if in any model we wish to require that the Kripke relation $\mathcal{H}$ is a subset of the Kripke relation $\mathcal{K}$, then this is specified in the logic by requiring the axiom $[k]\,P \supset [h]\,P$. Any model validating this axiom will be such that $\mathcal{H}$ interpreting $[h]$ is a subset of $\mathcal{K}$ interpreting $[k]$.

There are two generalizations to the partially ordered modality picture made possible by simulation logic. The subset relation between two binary relations is a simulation relation and as a consequence, simulation logic includes the logic of partially ordered modalities. The second is that a simulation relation can relate two different modal frames whereas the partial order required the relations be on a common modal frame. Thinking categorically for a moment, a simulation relation represents an arrow from one model to another. In partially ordered modalities, there is a very slim category, the partial order. So in effect, we are asking for more than one reason to map between modalities beyond mere subset inclusion in their interpreting relations.

Modal frames and simulation relations form a category where the simulations are the arrows. We wished for a logic in which we could reason about these simulations because the applications indicated above are most naturally expressed using simulations in the reasoning. Since simulations are relations, the question becomes then, which parts of our logical tool box can be used to represent this arrow? A simulation relation is just that, a relation. Naturally, it occurs to think of a modal operator for which the simulation relation can be used for its interpretation. We modify an axiom in Chellas [**?**] and Lemmon [**?**], which is attributed to Geach in an earlier work [**?**]. The idea is to use what Chellas termed incestual relations. They yield a series of axioms of a particular form. The main axiom for simulation logic has one of these forms. However, the Geach axiom uses a single relation whereas simulation relations use at least three relations, the two modal relations and the simulation relation.

Simulation logic is a federated or modular logic in that a simulation axiom links two modal logics where each logic is allowed to have its own axiom set. We term these modal logics "local logics". In terms of our compiler example above, one logic is for the source code and one logic is for the compiled code. The simulation logic allows for logical statements about the source to be "transferred" to logical statements about the compiled code. We work with normal modal logics, an extension to non-normal modal logics is planned.

2 The Logic

The logic is, in the language of Chellas, a twist on the logic for incestual relations. We assume normal modal systems in this paper.

2.1 Incestual and Simulation Relations

Chellas describes the general schema:

$$G^{k,l,m,n} : \Diamond^k \Box^l P \supset \Box^m \Diamond^n P,$$

where k, l, m, n are natural numbers. $\Diamond^k$ stands for a string of k instances of the operator $\Diamond$, and similarly for the rest.

As an example,

$$T_c = G^{0,0,1,0} : P \supset \Box P$$

axiomatizes vacuous frames where Rxx' implies $x = x'$.

The schema G is valid in all incestual models, or models which satisfy

$$\alpha R^k \beta \text{ and } \alpha R^m \gamma \text{ implies for some } \delta, \beta R^l \delta \text{ and } \gamma R^n \delta,$$

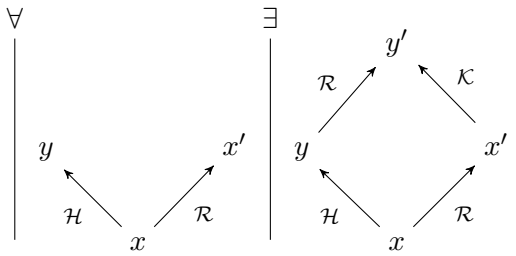
where, in sympathy with the operators, R^k stands for k fold relational composition $R \circ \cdots \circ R$ with relational composition defined as $\alpha R \circ R\gamma$ iff there is a β such that $\alpha R\beta$ and $\beta R\gamma$. A modification of this condition defines a simulation from modal logic (now changing notation from Chellas to the notation to be used throughout this paper).

Definition 2.1.1 *When a relation $\mathcal{R}$ from the relation $\mathcal{H}$ to the relation $\mathcal{K}$ satisfies the following condition,*

$$\mathcal{R}xx' \text{ and } \mathcal{H}xy \text{ implies } \textit{for some } y'(\mathcal{K}x'y' \text{ and } \mathcal{R}yy'),$$

it is known as a simulation relation.

We prefer to depict the information from this definition in the following confluent diagram with apologies to Freyd and Scedrov [**?**] for appropriating their categorical notation for a diagram that is not category theoretic



where one reads from left to right,

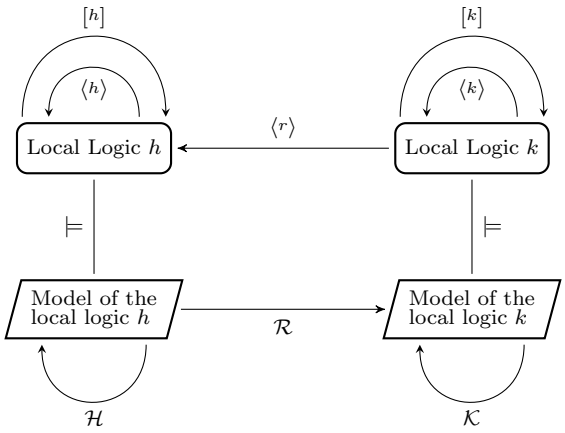
for all x, y, x' such that $\mathcal{R}xx'$ and $\mathcal{H}xy$, there exists a y' such that $\mathcal{K}x'y'$ and $\mathcal{R}yy'$.

The schema G for $k = l = m = n = 1$ is also known as the Geach axiom (left):

$$\Diamond \Box P \supset \Box \Diamond P, \qquad \langle r \rangle [k] P \supset [h] \langle r \rangle P.$$

Working backwards from the relations in the diagram and the proof of the validity of this modal formula, we get (now using index k to refer to a particular modal operator and not a natural

number) the formula on the right, where the modal operators $\langle r \rangle$, $[k]$, and $[h]$ are interpreted by the relations $\mathcal{R}$, $\mathcal{K}$, and $\mathcal{H}$ respectively. If we simplify a bit and allow the indices h and k to refer to a logic as well as indexing the logic's modal operators, and we also assume there are only the modal operators $[k], \langle k \rangle$ in the logic at k, the mental picture of two local logics h and k semantically connected by a simulation $\mathcal{R}$ is



which should look familiar from duality theory for modal logics where typically $\mathcal{R}$ is a p-morphism. Simulation relations are generalizations of p-morphisms. By allowing the p-morphisms to be relations, we allow for picking up this bit of the model theory of modal logics and expressing it in the logic.

Notice that the $\mathcal{R}$ relation appears to map between two sets of worlds, those involved with $\mathcal{H}$ and those involved with $\mathcal{K}$. We wish to build this observation into the core of the logic. Model theoretically, $\mathcal{R}$ can be made to relate two different logics. Simulation relations compose and the identity relation is a simulation relation. The specification of the logic needs to make this apparent. We do this by using a *graph*. Our models are always in the category of descriptive, general frames (DG frames) with simulation relations as morphisms. The indices h and k above can then be taken to label nodes in a graph for a simulation logic.

The axiom $\langle r \rangle [k] P \supset [h] \langle r \rangle P$ is somewhat curious. P is a formula at k and so $[k]$ may be applied to it. $\langle r \rangle$ maps the result to a formula at h. The consequent first maps P to h and then applies $[h]$. In effect, h gets to import some of the language of k but with no access to the internal structure of the formulas involved. Hence, $\langle r \rangle$ is a sort of wrapper that allows the importation into h of formulas some formulas from k.

2.2 Simulation Logic

Observing the validity proof of the Geach axiom and seeing that the form of the semantic condition is the same as the form of the definition for simularity minus the totality condition, we now axiomitize our base logic. We assume the modal logics are normal. A simulation logic requires we specify which local logics are to be present. This is done with a graph. There will be a node for each local logic. The graph also contains an arrow for every simulation relation to be present in any interpretation of the simulation logic.

A local logic is local in that it is associated with one node in the graph. The accompanying notion of a global logic does not entail formulas "spanning" two local logics in the sense of P in one

logic entailing Q in another where entailment is an implication arrow (and similarly with other two place connectives). Each formula lives entirely within a single local logic although it may contain subformulas from others.

The graph makes apparent the structure of the collection of the local logics and allows us to use the arrows to specify which simulation relations must exist in any interpretation of the logical structure. Since there are several modal operators that can be defined for any single relation, assigning an arrow for every modal operator can get a bit "noisy". Using the arrows to specify which relations must exist in an interpretation simplifies this.

The simulation logic graphs we use have *endo-diagrams*, each of which is a labeled node and a single endo-arrow (self-arrow). Under interpretation, this arrow is in a commutative diagram with an unmentioned empty arrow, in effect, giving the identity arrow a name and indicating that it is to be a simulation arrow. This is necessary since the models for the logic will be a category whose identity arrows will be simulations.

The notation $\mathrm{dom}(r)$ refers to the domain or source of the arrow r in a graph and $\mathrm{cod}(r)$ refers to the codomain or target of the arrow r. We use the locution $\langle k \rangle \in \mathrm{dom}(r)$ to refer to a modal connective in the logic associated the node which is the source for the arrow r. The symbol $\equiv$ is used for *bi-implication*, i.e., $P \equiv Q$ stands for $\vdash P \supset Q$ and $\vdash Q \supset P$.

We use the following letter conventions:

entity	description
h, k, l	nodes in a graph $\mathfrak{G}$
$\langle h \rangle, [h]$	modalities at node h and similarly for modalities at nodes k, l
r, s	arrows in a graph $\mathfrak{G}$
$\langle r \rangle, [r]$	forward-looking modalities associated to arrow r and similarly for the arrow s
$\langle\!\langle r \rangle\!\rangle, [\![r]\!]$	backward-looking modalities associated to arrow r and similarly for the arrow s
H, K, L	sets of worlds in an interpretation for the modalities at h, k, l respectively
$\mathcal{H}, \mathcal{K}, \mathcal{L}$	modal relations interpreting the modalities at h, k, l respectively
$(H, \mathcal{H}, A)$	general, descriptive frame for the logic at h
$(K, \mathcal{K}, B)$	general, descriptive frame for the logic at k
$\mathcal{R}, \mathcal{S}$	simulation relations interpreting modalities for arrows r and s

2.3 Axioms and Rules

The "axiom" **??** is a bow to Meyer [**?**] who uses this "axiom" to simplify his axiom set. The rest of the axioms are arranged in groups. The graph axioms just tell us that to specify a simulation logic, we must first specify a simulation graph detailing which local logics there are to be, which simulation arrows are to appear in any model, and forcing identity arrows to exist as simulation arrows.

The **A** axioms are not optional; they describe the form of the local logics. They are to all be normal modal logics and there is one per each node in the graph $\mathfrak{G}$. Each local logic may have its own propositional atoms and local modalities. The **B** axiom **??** forces the arrows in the graph to be interpreted as simulation relations and **??** forces composition of arrows to hold. These two

axioms are not optional. The **C** axioms are optional; they may be added to the **B** axioms to force the simulation relations to be bisimulations. The **D** axioms may be added to the **B** axioms to force the simulation relation to be *p-morphisms*, i.e., simulation relation which are also functions. One can also add their backwards looking counterparts to force bisimulations to be p-morphic in reverse direction. The **E** axioms are optional; **??** forces propositional atoms at the domain of a simulation arrow to be interpreted as they are at the codomain. **??** forces propositional atoms at the codomain of an arrow be interpreted as they are at the domain.

Graph

S1. A graph $\mathfrak{G}$ of nodes and arrows
A set $\mathfrak{D}$ of endo-diagrams

S2. An endo-diagram for each node in $\mathfrak{G}$

Axiom Schemes A: For each node in $\mathfrak{G}$, and each endo-diagram arrow i,

A1. all truth functional theorems of a propositional logic

A2. Modal axioms for a logic at this node

A3. $P \equiv [i]\, P$

Axiom Schemes B: For each arrow each $r : h \rightarrow k$ and $s : k \rightarrow l$ in $\mathfrak{G}$, and each modal operator $[h] \in \mathrm{dom}(r), [k] \in \mathrm{cod}(r)$,

B1. $\langle r \rangle\, [k]\, P \supset [h]\, \langle r \rangle\, P$

B2. $\langle r \rangle\, \langle s \rangle\, P \equiv \langle s \circ r \rangle\, P$

Notice the axiom **??** while being a statement of the logic at h uses subformulas $[k]\, P$ and P at k.

Axiom Schemes C: To force the simulation arrows to be bisimulations, use the following axioms using backward looking possibility, $\langle\cdot r\rangle$, on the simulation arrow. For each arrow $r : h \rightarrow k$ and $s : k \rightarrow l$ in $\mathfrak{G}$, and each modal operator $[k] \in \mathrm{cod}(r), [h] \in \mathrm{dom}(r)$

C1. $\langle\cdot r\rangle [h]\, P \supset [k]\, \langle\cdot r\rangle P$

C2. $\langle\cdot s\rangle \langle\cdot r\rangle P \equiv \langle\cdot r \circ s\rangle P$

Axiom Schemes D: To force simulation arrows to be *p-morphisms* use the following **D** axioms; they force the interpreting relation to be a function (just as they would in any normal modal logic systems):

D1. $[r]\, P \supset \langle r \rangle\, P$

D2. $\langle r \rangle\, P \supset [r]\, P$

Axiom Schemes E: The axiom **??** is only necessary if you wish the classical proposition logic at $\mathrm{dom}(r)$ to be included in the logic at $\mathrm{cod}(r)$. It is not strictly necessary although it does pick up the clause in the definition of simulation [**?**] requiring this of a simulation.

For all propositional letters p,

E1. $p \supset [r]\, p$

E2. $p \supset [\cdot r\cdot] p$

As a simplification which will not detract from the generality of the logic, we will assume that each local logic can be interpreted with a single modal relation. This allows us to equate a node usually labeled h or k with the modal logic at that node.

Definitions and Rules (Normal Systems)

Definition of Possibility: $\langle m \rangle\, P \stackrel{def}{=} \neg\, [m]\, \neg P, \quad m \in \{k, r\}$

Rules A: For each local logic k,

$$\frac{\vdash_k P \qquad \vdash_k P \supset Q}{\vdash_k Q}$$

$$\frac{\vdash_k (P_1 \wedge \cdots \wedge P_n) \supset P}{\vdash_k ([k]\, P_1 \wedge \cdots \wedge [k]\, P_n) \supset [k]\, P}$$

Rules B: For each $r : h \to k$ arrow in $\mathfrak{G}$,

$$\frac{\vdash_k (P_1 \wedge \cdots \wedge P_n) \supset P}{\vdash_h ([r]\, P_1 \wedge \cdots \wedge [r]\, P_n) \supset [r]\, P}$$

Rules C: For each $r : h \to k$ arrow in $\mathfrak{G}$,

$$\frac{\vdash_h (P_1 \wedge \cdots \wedge P_n) \supset P}{\vdash_k ([\cdot r \cdot] P_1 \wedge \cdots \wedge [\cdot r \cdot] P_n) \supset [\cdot r \cdot] P}$$

where the subscripted $\vdash$ indicates to the local logic to which the proof sign attaches.

We will only be concerned with necessity and possibility operators and mostly the forward versions since the backwards versions (for bisimulation) are so similar. Other modal operators definable for the same relation (see [**?**]) require other kinds of relations than simulation relations for their "movement" among logics. We relegate these sorts of complications to a sequel paper.

From now on, a simulation logic refers to the axiom schemes **S**, **A**, and **B**, and the Definition of Possibility, and and the rules **A** and **B**. We will ignore the axiom schemes and rules **C** in the interest of conciseness and because handling those is so similar to **B**. The axiom schemes **D** are of interest and we have modeling conditions for them. The axiom schemes **E** must be handled quite separately in the semantics.

2.4 Frames and Models

Each node in a simulation logic's graph has a local logic associated with it and that local logic must have a DG frame associated with the local logic. Each arrow of the graph must be associated with a simulation relation. Each node representing a distinct local logic must be mapped to a distinct frame object in any interpretation. This sort of informal way of restricting interpretations is the result of not treating the graph as defining everything in a simulation logic, but the alternative would make the logic impenetrable.

We work with DG frames, and in keeping with our simplification, assume there is only one modal relation per frame. More relations for more modalities can be added if needed.

Definition 2.4.1 (Kupke, Kurz, and Venema [?]) *A* general frame *is a structure* $\mathcal{H} = (H, \mathcal{H}, A)$ *such that* $(H, \mathcal{H})$ *is a Kripke frame and* A *is a collection of so-called* admissible *subsets of* H *that is closed under the Boolean operations and under the operation* $[h] : A \to A$ *given by:*

$$[h]\, C \stackrel{def}{=} \{x \in X \mid \mathcal{H}xy \text{ implies } y \in C\}$$

with $\langle h \rangle\, C = \neg\, [h]\, \neg C$ *where* $\neg$ *is set complement. A general frame* $\mathcal{H}$ *is called* differentiated *if for all distinct* $x, y \in H$ *there is a 'witness'* $a \in A$ *such that* $y \in a$ *while* $x \notin a$*;* tight *if whenever* y *is not an* $\mathcal{H}$*-successor of* x*, then there is a 'witness'* $a \in A$ *such that* $y \in a$ *while* $x \notin \langle h \rangle\, a$*; and* compact *if* $\bigcap A_0 \neq \emptyset$ *for every subset* A_0 *of* A *which has the finite intersection property. A general frame is* descriptive *if it is differentiated, tight, and compact.*

We use the same symbol to refer to the general frame and its relation letting context disambiguate the use. As it is noted in [**?**], tightness can be re-expressed as a relation being point closed, i.e., $\mathcal{R}x$ (equal to $\mathcal{R}\{x\}$), the forward image of x under $\mathcal{R}$, is a closed set in the topology generated by the algebra A of clopen sets.

We use a category of DG frames for a simulation logic.

Definition 2.4.2 *A* simulation frame category *has a descriptive, general frame, called a* local frame*, for each node in the graph, and a point closed relation for each arrow. That is, for $r : h \to k$ in the graph, r's interpretation $\mathcal{R} : \mathcal{H} \to \mathcal{K}$ must be point closed.*

The corresponding Kripke frame conditions for the logical axioms are

Frame Conditions S:

FS1.	A category of *local modal frames* and simulations	FS2.	An identity arrow for the arrow in $D \in \mathfrak{D}$

Frame Conditions A: For each node in $\mathfrak{G}$,

FA1.	A set of classical worlds	FA2.	Modal frame conditions for a logic at this node

Frame Conditions B:

FB1.	$\mathcal{R}xx'$ and $\mathcal{H}xy$ implies $\exists y'(\mathcal{K}x'y'$ and $\mathcal{R}yy')$	FB2.	Frame category contains all compositions

Frame Conditions C:

FC1.	$\mathcal{R}xx'$ and $\mathcal{K}x'y'$ implies $\exists y(\mathcal{H}xy$ and $\mathcal{R}yy')$	FC2.	Frame category contains all compositions

Frame Conditions D:

FD1.	$\exists y(\mathcal{R}xy)$	FD2.	$(\mathcal{R}xy$ and $\mathcal{R}xz)$ implies $y = z$

with the convention that upper case script relation letters interpret modalities using lower case Roman letters. Each simulation frame category interpreting a simulation logic will have the conditions matching the axioms. The frame conditions **S**, **A**, and **B** are always assumed, the others are required if the corresponding axioms are present in the modeled local logic.

We will use slightly different frames for axioms **??** and **??**; the algebras in the local frames will contain constants, one for every atomic proposition of the local logic for which the local frame provides a model.

The following proposition allows us to use one DG frame per local logic.

Proposition 2.4.3 *There are no provable instances of formulas of the form $P \bullet Q$ for $\bullet \in \{\supset, \wedge, \vee\}$ with P in one local logic and Q in different local logic.*

The proof is an easy induction on the axiom schemes **A**-**E** and rules. The axioms **??** and **??** are the most interesting. Considering **??**, P and $[k]\,P$ is in the local logic at k while $\langle r\rangle\,[k]\,P$ is in the local logic at h. The formulas $\langle r\rangle\,P$ and $[h]\,\langle r\rangle\,P$ are in the local logic at h. The consequence is that no formula in the logic has a binary connective between formulas in two different local logics.

Note, we stated that above proposition in terms of formula "instances" rather than formulas because it is possible to attach a local logic to more than one node in the graph. In effect, this gives two instances of the logic in the entire simulation logic.

In preparation for soundness, we must show that the definition of $\langle r\rangle$ in terms of $[r]$ makes sense in that it does not clash with the definition of interpretation. To wit, let $\mathcal{R} : h \rightarrow k$ be a simulation

$$\begin{aligned} x \in H \models \neg\,[r]\,\neg P \text{ iff } & x \in H \not\models [r]\,\neg P \\ & \text{iff there exists } y \in K(\mathcal{R}xy \text{ and } y \not\models \neg P) \\ & \text{iff there exists } y \in K(\mathcal{R}xy \text{ and } y \models P) \\ & \text{iff } x \in H \models \langle r\rangle\, P \end{aligned}$$

A simulation category model has the usual sort of Kripke DG frames for every node with a valuation for each node. The arrows are the simulation relations.

Definition 2.4.4 *A* simulation category model *is a simulation frame category with valuation and a local frame for each local logic. The local frame and its valuation are called a* local model. *The valuation specifies a collection of points in the local frame where the atomic propositions are true.*

We rely on heterogeneous (multisorted) algebras [**?**] for the free algebra construction. The categorical version is most easily accessible in [**?**] who attribute the multisorted (non-categorical) case to [**?**].

Definition 2.4.5 (Birkhoff and Lipson [?]) *A heterogeneous algebra is a system $A = [\mathscr{L}, F]$ in which*

1. *$\mathscr{L} = \{S_i\}$ is a family of non-void sets S_i of different types of elements, each called a* phylum *of the algebra A. The phyla S_i are indexed by some set I; i.e., $S_i \in \mathscr{L}$ for $i \in I$ (or are called by appropriate names).*

2. *$F = \{f_\alpha\}$ is a set of finitary operations operations, where each f_α is a mapping*

$$f_\alpha : S_{i(1,\alpha)} \times S_{i(2,\alpha)} \times \cdots \times S_{i(n(\alpha),\alpha)} \rightarrow S_{p(\alpha)}$$

for some non-negative integer $n(\alpha)$, function $i_\alpha : j \rightarrow i(j,\alpha)$ from $n(\alpha) = \{1, 2, \ldots, n(\alpha)\}$ to I, and $p(\alpha) \in I$. The operations f_α are indexed by some set Ω; i.e., $f_\alpha \in F$ for $\alpha \in \Omega$ (or are called by appropriate names).

Definition 2.4.6 *A* simulation algebra *appropriate for a simulation logic is a heterogeneous algebra with a modal algebra, called a* local modal algebra, *for each node of a graph, identity modal operators for each node, and* simulation operators $\langle r\rangle$ *and* $[r]$ *for every arrow r (including identity arrows) of the graph. If the bisimulation axioms are used, then there are operators $\langle\!\langle r\rangle\!\rangle$ and $[\![r]\!]$ in the opposite direction than $\langle r\rangle$ and $[r]$. For $r : h \rightarrow k$ in the graph,*

- $[r](a \wedge b) = [r]\,a \wedge [r]\,b$;
- $[r]\,\top_k = \top_h$, *for $\top$ the top of Boolean lattice;*
- $\langle r\rangle\,[k]\,a \leq [h]\,\langle r\rangle\,a$;

- $\langle r \rangle \langle s \rangle a = \langle s \circ r \rangle a$;
- $[i]\, a = a$, *for* i *the arrow in an endo-diagram;*
- $\langle\!\cdot r \cdot\!\rangle [h]\, a \leq [k] \langle\!\cdot r \cdot\!\rangle a$, *if axiom schemes C are used;*
- $\langle\!\cdot s \cdot\!\rangle \langle\!\cdot r \cdot\!\rangle a = \langle\!\cdot r \circ s \cdot\!\rangle a$, *if axiom schemes C are used;*
- $\langle r \rangle\, a = [r]\, a$, *if axiom schemes E are used.*

The axioms **??** and **??** will be handled in the next subsection where we must add constant operations and functions to help interpret the propositional atoms.

Appropriate simulation algebras give a "localization" view of heterogeneous algebras which is isomorphic to the definition given above. Each phylum S_i with operators defined only upon S_i is a local modal algebra. The operations associated with $r : h \rightarrow k$ of the graph map from a local modal algebra to a local modal algebra. This stratifies the heterogeneous simulation algebra and treats every local modal algebra as an object in the surrounding simulation algebra.

Algebraic versions of soundness and completeness depend on the Lindenbaum-Tarski (LT) algebra. We must first show that the operators all respect the congruence of bi-implication induced on the local word algebras by the local logics. The only operators not already covered in previous modal algebraic work are the simulation operators.

Lemma 2.4.7 *The simulation operators respect bi-equivalence.*

Proof: We only show the proof for $\langle r \rangle$, the other simulation operators use similar principles.

$\vdash_k P \supset Q$ implies $\vdash_k \neg Q \supset \neg P$	contraposition
implies $\vdash_h [r] \neg Q \supset [r] \neg P$	rule **??**
implies $\vdash_h \neg [r] \neg P \supset \neg [r] \neg Q$	contraposition
iff $\vdash_h \langle r \rangle P \supset \langle r \rangle Q$	Definition of Possibility

It easily follows that $\vdash_k P \equiv Q$ implies $\vdash_h \langle r \rangle P \equiv \langle r \rangle Q$. ∎

Next, we must show that the LT algebra is actually a simulation algebra. The only operators that are at issue are the simulation operators.

Lemma 2.4.8 *The LT simulation operators satisfy the required properties for a simulation algebra.*

Proof: We only show the proofs for $[r]$, the rest use similar principles. Setting n to 0 in the rule **??**, yields $\vdash_k \top$ implies $\vdash_h [h] \top$. Hence $\vdash_h \top \supset [h] \top$. It is always the case that $\vdash_h [h] \top \supset \top$, hence $[h] \top \equiv \top$ and so $[h][\top] = [\top]$.

That $\vdash_h [r] P \wedge [r] Q \supset [r](P \wedge Q)$ follows immediately from $\vdash_k P \wedge Q \supset P \wedge Q$ and the rule **??**. Next, $\vdash_k P \wedge Q \supset P$ and $\vdash_k P \wedge Q \supset Q$, so $\vdash_h [r](P \wedge Q) \supset [r] P$ and $\vdash_h [r](P \wedge Q) \supset [r] Q$. Hence, $\vdash_h [r](P \wedge Q) \supset [r] P \wedge [r] Q$, and so $[r][P \wedge Q] = [r][P] \wedge [r][Q]$. ∎

Corollary 2.4.9 *The LT heterogeneous algebra is a simulation algebra.*

Proof: (Proof Outline) The free heterogeneous algebra is the usual algebra of equivalence classes of terms in the variables as generators. One runs the induction procedure to get the word algebras over all the local logics simultaneously [**?**], then divide out by the equalities in each algebra. Proposition **??** shows that no additional sorts over and above the local modal algebra carrier sets are necessary. Lemma **??** shows that the replacement property for the bi-implication congruence holds for each operator. Finally, Lemma **??** shows each LT operators satisfy the simulation algebra axioms. ■

Theorem 2.4.10 *Simulation Logic is sound with respect to the algebraic and simulation frame category models.*

Proof: We need only check the simulation axiom. Soundness over the algebraic models is the usual induction starting with a valuation into a simulation algebra and then using the fact that the LT algebra is the free algebra in the heterogeneous class of simulation algebras. The free heterogeneous algebras are then used to generate the universal arrow for any interpretation into a heterogeneous modal algebra thus validating the axioms and rules.

We check axiom **??** for Kripke models, the rest use similar principles. Assume $x \models \langle r \rangle [k] P$. To show, $x \models [h] \langle r \rangle P$, we further assume $\mathcal{H}xy$ for arbitrary y. From the first assumption, there is some x' such that $\mathcal{R}xx'$ and $x' \models [k] P$. Since, $\mathcal{H}xy$ and $\mathcal{R}xx'$, the frame condition **??** yields $\mathcal{K}x'y'$ and $\mathcal{R}yy'$ for some y' and hence $y' \models P$. By definition, $y \models \langle r \rangle P$. Since the choice of y was arbitrary, by definition $x \models [h] \langle r \rangle P$. ■

The *canonical frame* is generated by the LT algebra; the frame's admissible sets are those of the carrier set of the representation algebra (of the LT algebra) using the representation function β (see below). Let $\mathrm{MA}(h), \mathrm{MA}(k)$ stand for the local modal algebras and $\mathrm{CF}(h), \mathrm{CF}(k)$ stand for the canonical frames at h and k respectively. To get a frame category from the LT modal algebra requires that one take the (dual) Stone spaces containing all the maximal filters of each local algebra and define the Kripke relations with:

$$\mathcal{H}xy \text{ iff } [h] a \in x \text{ implies } a \in y.$$

Since $[h]$ and $\langle h \rangle$ are DeMorgan duals of each other, $\mathcal{H}$ admits an equivalent definition:

$$\mathcal{H}xy \text{ iff } a \in y \text{ implies } \langle h \rangle a \in x.$$

These same definitions work for the canonical relation $\mathcal{R}$ for $r : h \nrightarrow k$ where now $a \in \mathrm{MA}(k)$, $[r] a, \langle r \rangle a \in \mathrm{MA}(h)$, $x \in \mathrm{CF}(h)$ and $y \in \mathrm{CF}(k)$.

The modal completeness argument is the usual algebraic argument [**?**] using contraposition and the frame argument uses the canonical frame derived from a representation theorem [**?**, **?**]. The modal representation theorem represents a modal algebra as an algebra of sets using the canonical frame (Stone space) of the algebra. One defines the 1-1 homomorphism β on the Boolean algebra $\mathcal{A} = (A, \wedge, \vee, \neg)$ by:

$$\beta a = \{x \mid a \in x \text{ and } x \text{ is a maximal filter}\}.$$

It is not hard to show that $\beta [h] a = [h] \beta a$ and $\beta \langle h \rangle a = \langle h \rangle \beta a$. Set union, intersection, and set complement interpret the classical logic logic connectives $\vee$, $\wedge$, and $\neg$. The only question is the status of $\langle r \rangle, [r]$ for $r : h \nrightarrow k$.

Lemma 2.4.11 *For $a \in \mathrm{MA}(k)$ and $\langle r \rangle a \in \mathrm{MA}(h)$,*

$$\beta [r] a = [r] \beta a \text{ and } \beta \langle r \rangle a = \langle r \rangle \beta a.$$

Proof: We show the proof of $\beta \langle r \rangle a = \langle r \rangle \beta a$, the other is similar. From right to left, let $x \in \mathrm{CF}(h)$ and $x \in \langle r \rangle \beta a$. There is some $y \in \mathrm{CF}(k)$ such that $\mathcal{R}xy$ and $y \in \beta a$, hence $a \in y$. Since $\mathcal{R}xy$ iff $b \in y$ implies $\langle r \rangle b \in x$, we have $\langle r \rangle a \in x$. So $x \in \beta \langle r \rangle a$. Going the other direction, let $x \in \beta \langle r \rangle a$, so that $\langle r \rangle a \in x$. Further, let $y_2 = \langle r \rangle^{-1} -x$ where $-x$ refers to the set complement of x and is a maximal ideal. y_2 is an ideal: $b_1, b_2 \in y_2$ iff $\langle r \rangle b_1, \langle r \rangle b_2 \in -x$ iff $\langle r \rangle b_1 \vee \langle r \rangle b_2 \in -x$ ($\langle r \rangle$ is a normal operator) iff $\langle r \rangle (b_1 \vee b_2) \in -x$ iff $b_1 \vee b_2 \in y_2$. Hence if $a \in y_2$ then $\langle r \rangle a \in -x$ which is a contradiction. $a{\uparrow}$ (the principle filter determined by a) and y_2 are disjoint, we can separate them with a maximal filter-ideal pair $(y, -y)$ with $a \in y$ and $y_2 \subseteq -y$ using Zorn's Lemma; so $y \in \beta a$. Let $b \in y$. If $\langle r \rangle b \in -x$, then $b \in -y$ and that is a contradiction. So $\langle r \rangle b \in x$. By definition, $\mathcal{R}xy$ holds and $a \in y$, so $x \in \langle r \rangle \beta a$. ∎

Theorem 2.4.12 *Simulation Logic is complete with respect to the simulation algebras and the simulation category models.*

Proof: From Proposition **??**, we need only concern ourselves with formula (instances) which sit entirely within a single local logic. So one presents the formula instance at issue and then picks the local logic for which it must be determined whether it is a theorem. The argument is a contraposition argument. Using the LT heterogeneous algebra and its canonical frame category.

Note that any theorem without an implication as the main connective can be outfitted with one because $\vdash P$ iff $\vdash T \supset P$ where T is the truth constant in a local logic. Hence we need only check implications. Suppose $\nvdash P \supset Q$, then $[P] \not\leq [Q]$ in the LT algebra where $[P], [Q]$ are the bi-implicational equivalence classes. This along with Corollary **??** is enough for algebraic completeness.

For frame completeness, there is maximal separating filter x such that $[P] \in x$ and $[Q] \notin x$, i.e., $x \in \beta[P]$ and $x \notin \beta[Q]$, so $x \models P$ and $x \not\models Q$. Therefore there is a local model falsifying the non-theorem, and hence a simulation category model falsifying the non-theorem.

Taking the contrapositive in the algebraic and frame cases yields the required result. ∎

2.5 The Axiom Schemes E

The axioms **??** and **??** requires some special treatment. We concentrate on **??** since **??** is entirely similar. The algebra will now have a collection of constant operators, one for each propositional atom in the language.

Definition 2.5.1 *An* E local modal algebra *is a local modal algebra with a collection of constant operations. Note that two constant operations, being functions, can point to the same element of the local modal algebra. The LT E local modal algebra has each constant operation pointing out the equivalence class of the propositional atom to which it attached. In symbols, if p is a propositional atom, then its constant, nullary operation, σ_p, is such that $\sigma_p = p$ in the word algebra of the logic and $\sigma_p = [p]$ in the LT algebra. In addition to any axioms necessary for the local modal logic, we add the axiom*

$$\sigma_p \leq [r] \sigma_p$$

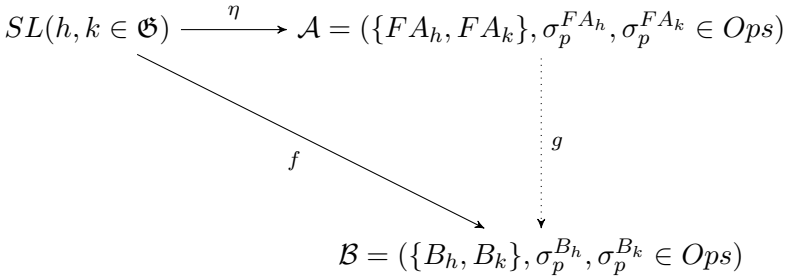
for an arrow r in the diagram to another node. This effectively forces $[\![p]\!] \leq [r][\![p]\!]$ for any interpretation $[\![-]\!]$. We also require the logic at $\mathrm{cod}(r)$ to contain at least the same propositional atoms as those at $\mathrm{dom}(r)$.

Definition 2.5.2 *A E local modal frame is a descriptive, general frame with a collection of constant functions, f_p, one for each propositional atom. A constant function selects an element of the set algebra.*

Fix a local modal algebra. A modal valuations vary over what gets assigned to the propositional atoms. Here, the valuations must be consistent with the nullary operations associated with each atom. We get the variation necessary for valuations by choosing different local modal algebras which agree on everything except the nullary operations. So the variation gets satisfied at a slightly higher level. A similar statement holds for local modal frames. The inductive definition generating interpretations from valuations remains the same and hence the restriction on valuations gets transferred to interpretations.

Definition 2.5.3 *A E local modal algebra valuation, $[\![-]\!]$ must take every propositional atom to an element of the carrier set pointed to by the nullary operation for that atom, i.e., if $\sigma_p = a$, then $[\![p]\!] = a$. Similarly, for a E local modal frame and valuations $[\![-]\!]$, we demand $[\![p]\!] = C$ if $f_p = C$. Also, we demand that for $r : h \rightarrow k$, the r interpreting relation $\mathcal{R}$ must respect the constant functions in the sense that $x \in f_p$ at the local frame for h and $\mathcal{R}xy$ implies $y \in f_p$ at the local frame for k.*

For the LT algebra, $\sigma_p = p$ in the word algebra forces $\sigma_p = [p]$ in the LT algebra. The result is that we get the same LT algebra as we would have without the nullary constants. The universal property of the free algebra with respect to unique maps to the other E local modal algebras are unaffected since the restriction on interpretations will force the unique maps to choose the same elements of the local modal algebras to which the nullary operations point for the respective propositional atoms. In freeness diagram, p indicates some propositional atom in the language, σ^{FA_h} is the carrier set of the local modal logic for h inside of the free algebra $\mathcal{A}$, $\mathcal{B}$ is some other appropriate simulation algebra, and f is an induced interpretation from the freeness property of $\mathcal{A}$,



The algebra $\mathcal{B}$ has no notion of propositional atoms, the σ_p being operations and preserved by k, force the valuations to value Boolean terms at h to be the same as those at k for $r : h \rightarrow k$.

The extension to simulation algebras and simulation category models are called **E** simulation algebras and **E** simulation category models.

Theorem 2.5.4 *Simulation logics with the **E** axioms are sound and complete with respect to **E** simulation algebras and **E** simulation category models.*

3 Some Extensions of the Concept

3.1 Stone Spaces and Their Morphisms

The general scheme of lifting morphisms to create multi-sorted logics connected via modal operators appears to have a wider potential than the simulation logic presented in this paper. In effect, there was nothing special per se about using simulation relations other than they work very well in terms of relating modal logics. Suppose we consider the category of Stone spaces and their morphisms and ask what would the multi-sorted logic look like. We clearly want the axioms **??** and **??** and we always want to use descriptive general frames. The multi-sorted logic has a modal operator for every morphism in the graph, that morphism is always to be a continuous map. Since the one-point sets are closed in Stone spaces, the continuous maps are closed relations.

3.2 Partially-ordered Modalities

In [**?**], we presented a logic system with partially-ordered modalities. That system can be presented in the current framework by setting all the simulation relations to the identity relation. The main axiom

$$[k]\, P \supset [h]\, P$$

then is a direct consequence since $\langle i \rangle\, P \equiv P$ for i being the sole arrow in an endo-diagram.

3.3 Other Frame Relating Principles

The axiom

$$\langle s \rangle\, [k]\, P \supset [h]\, \langle r \rangle\, P$$

holds just when for every $r, s : h \rightarrow k$ in the graph, the frame category has

$$\mathcal{S}xx' \text{ and } \mathcal{H}xy \text{ implies there exists } y'(\mathcal{K}x'y' \text{ and } \mathcal{R}yy').$$

The axiom

$$[r]\, [k]\, P \supset [h]\, [r]\, P$$

holds just when for every $r, s : h \rightarrow k$ in the graph, the frame category has

$$\mathcal{R}yz \text{ and } \mathcal{H}xy \text{ implies there exists } x'(\mathcal{K}x'z \text{ and } \mathcal{R}xx').$$

3.4 Relating the Simulation Relations

We can use partially-ordered modalities [**?**] in the many-sorted case by using the axiom from that paper

$$[s]\, P \supset [r]\, P$$

for $r, s : h \rightarrow k$ in the graph of the logic with the interpreting relations mapping between local frames where $\mathcal{R} \subseteq \mathcal{S}$ in any interpretation.

Let $(H, \mathcal{H}, A)$ be the local frame at h and $(K, \mathcal{K}, B)$ be the local frame at k. If we wish to relate $\mathcal{R}$ with $\mathcal{S}$ via a *higher-order* simulation, this can be accomplished by reinterpreting the first

axiom for the previous subsection. A higher-order simulation from $\mathcal{R}$ to $\mathcal{S}$ is a pair of relations, $\mathcal{T} \subseteq H \times H$ and $\mathcal{T}' : K \times K$ such that

$$\mathcal{T}xx' \text{ and } \mathcal{R}xy \text{ implies there exists } y'(\mathcal{S}x'y' \text{ and } \mathcal{T}'yy').$$

This underwrites the axiom

$$\langle t \rangle [s] P \supset [r] \langle t' \rangle P.$$

The property of Proposition **??** still obtains. Using $\log(h)$ and $\log(k)$ to refer to the logics at those nodes of the graph, $\langle t \rangle : \log(h) \rightarrow \log(h)$ and $\langle t' \rangle : \log(k) \rightarrow \log(k)$. $P \in \log(k)$; hence $\langle t' \rangle P \in \log(k)$. Since $r : h \rightarrow k$ in the graph, $[r] : \log(k) \rightarrow \log(h)$ and so $[r] \langle t' \rangle P \in \log(h)$. Since $s : h \rightarrow k$, $[s] P \in \log(h)$ and $\langle t \rangle [s] P \in \log(h)$. The entire formula is in $\log(h)$.

4 Conclusion

We have shown that modal logic can be extended to include simulation relations as morphisms in their Kripke semantics. This necessitates a multi-sorted logic composed of local logics linked by modal operators for which the simulation relations provide morphisms between models. The general approach appears to be transferrable to other logics by observing that their models usually sit within a category of models which has its own notion of morphism. The model morphisms are to be looked at as Kripke relations spanning models. The fact that many times these morphisms are functions and may have other properties leads to axioms constraining their modal connective counterparts. In fact, many of the relational properties used in modal logics and their related modal axioms can be lifted to provide a toolbox of axioms for modalities that span logics.

One of the central features of the logics presented in this paper is that every formula sat entirely within a single local logic. Complete generality would require that feature to be relaxed and allow formulas to span local logics. We anticipate the graphs that we used in formulating simulation logics will be made more complicated, in effect, promoting them to be the categorical notion of *sketch* (originally due to Ehresmann but as detailed in [**?**]). In this paper, we only use graphs and identity arrows to help specify a simulation logic and have no need for general diagrams, cones, and cocones of a sketch. For instance, one might have formulas of the form $P \wedge Q$ where P and Q are in separate logics. Providing models for these sorts of logics will require constructions that produce new frames from old frames.