Distributed Modal Logic

Chapter · April 2016

DOI: 10.1007/978-3-319-29300-4_16

2 authors, including:

William Lawrence Harrison

72 PUBLICATIONS **342** CITATIONS

SEE PROFILE

Distributed Modal 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

Abstract

Modal logics typically have only one domain of discourse—i.e., the collection of worlds or states. For distributed computing systems, however, it makes sense to have several collections of worlds, one collection for each component and to relate one component's local worlds to another using either relations or special maps. To this end, we introduce distributed modal logics. Distributed modal logics lift the distribution structure of a distributed system directly into the logic, thereby parameterizing the logic by the distribution structure itself. Each domain supports a "local logic" (which can itself be a modal logic). The connections between the local logics are realized as "distributed modal connectives" where these connectives take propositions in one logic to propositions in another. More generally, weak distributed systems require neighborhood semantics and hence the connection between domains becomes a neighborhood map linking each world in one domain to a collection of neighborhoods in another domain. In sufficiently strong distributed systems, the maps may be Kripke relations linking worlds from two different domains. We illustrate distributed logics with the outline of a security verification for a hardware distributed system (i.e., a system-on-a-chip) with components that must be woven into proofs of security statements. Distributed modal logics support probabilistic systems using stochastic relations.

1 Introduction

Modal logics typically have only one domain of discourse—i.e., the collection of worlds or states. For distributed computing systems, however, it makes sense to have several collections of worlds and to relate one domain's local worlds to anothers using either relations or special maps. To this end, we introduce distributed modal logics. Distributed modal logics lift the distribution structure of a distributed system directly into the logic, thereby parameterizing the logic by the distribution structure itself. Each domain supports a "local modal logic". The connections between local logics are realized as "distributed modal connectives" where these connectives take propositions in one logic to propositions in another. More generally, weak distributed logic systems require neighborhood semantics and, hence, the connection between domains becomes a neighborhood map linking each world in one domain to a collection of neighborhoods in another domain. In sufficiently strong distributed logic systems, the maps may be Kripke relations linking worlds from two different domains. We briefly illustrate distributed modal logics with the outline of a security verification for a hardware distributed system (i.e., a system-on-a-chip) with domains that must be woven into proofs of security statements.

Logic in 20th century had many parents. It settled upon a very linguistic base and many logical investigations concern exploring logical notions as represented via this linguistic base. The reasoning that we can perform using logic in this manner is filtered through this linguistic base. This has the tendency to force some notions to be expressed (if even possible) in higher abstract formal machinery and more complicated semantics than is desired if we intend for logics to be used by humans for reasoning (as opposed to machines). The problems encountered are not to be considered as artificially imposed via the linguistic base, but rather there is much more that could be represented directly of the world about which we use logic to reason. It is

in this sense that we present distributed logics, i.e., as an attempt to capture more of the work of reasoning that we need logic to support. The received linguistic syntactic structure should not be seen as paradigmatic for logic but rather a first attempt at coming to terms with logical reasoning. Distributed systems are commonplace in computing and engineering, yet they have been rather less so in the philosophical world. Distributed logic extends the notion of what is to be considered as logical, and yet we still rely heavily on the hard work of our predecessors in logic.

Much of the background in distributed logic owes a debt to J. Michael Dunn for his work in Gaggle Theory [**?**], [**?**]—Gaggle Theory's notion of residuation is essentially a notion of distribution, for example. Gaggle theory can be used to relate two different algebraic systems and it is but a short step to view logics through algebraic eyes as do most algebraic logicians (of which the first author considers himself). Another precursor to distributed logics is Barwise and Seligman [**?**]. The colloquial term used is *channel theory* and channel theory is billed as the logic of distributed systems. We have done work in channel theory [**?**] and, indeed, spent quite a bit of time learning how its notion of distribution is used in a logical setting. The notion of a *local logic* stems from channel theory. Channel theory itself relies heavily on classical logic.

The direct precursor to distributed logic is partially ordered modalities [**?**]. The partial order among modalities is generally sparse in any application and is modeled via a partial order on relations. Of course, there is, at least, a complete lattice of relations on a set. However, there are few relations in most applications and, consequently, the entire complete lattice is mostly noise, i.e., most of the relations have no realistic counterpart in an application. It was our desire to generalize partially ordered modalities that led directly to distributed logics. We concentrated on modal logics because we were attempting to generalize a modal base. In subsequent work, we will modify the modal distributed logics to intensional distributed logics in an analogous sense to how relevance logic modifies modal logic.

A traditional approach to distributed systems is Markov transition systems. Here, the notion of measurement is prominent. There has been some recent work (see [**?**] and his references) showing how stochastic relations (in place of Kripke relations) can be used in measuring Kripke systems expressed as coalgebras. We only present the notion here to show how distributed modal logics are appropriate logic systems for Markov transition systems. Modal systems as coalgebras require a single local logic, and, hence, do not really provide an adequate logical framework for Markov transition systems and stochastic relations. Our work in distributed logic did not arise from stochastic relations. However, in retrospect, the match is very tight and we can view the work on stochastic relations as giving us a continuous mathematics interpretation for distributed modal logic.

A *distributed modal logic* is a collection of *local modal logics* linked together by *distributed modal connectives*, each of which takes formulas in one logic and returns formulas in a different logic. Semantically, each local logic is interpreted over a collection of worlds. Let this collection be called the *local collection* for this local logic. A *local neighborhood (nbd) map* takes each world to a set of neighborhoods taken from the local collection and is used to interpret the modal connectives of the local logic. The distributed modal connectives are also interpreted using nbd maps; here, the nbd maps take worlds from a local collection of worlds to nbds of worlds from a different local collection.

Extra properties, via logical axioms and rules, can be imposed on the interpreting nbd maps. This is precisely analogous to imposing conditions on Kripke relations or nbd maps in traditional modal logic. Many of the usual conditions (e.g., normality or functionality) can be generalized from their traditional counterparts. The selection of axioms reflects the model theory one needs for an application. If one adds enough axioms to force the distributed modal connectives to be normal modal connectives (even though they map from one logic to another), the interpreting nbd maps can be defined to be Kripke relations that, here, span local collections.

There are other approaches to locality in logic: we have already mentioned channel theory [**?**, **?**], and institutions [**?**] and Chu spaces [**?**] are others. There are also multi-agent logic systems [**?**]. What distinguishes distributed logics from these are that the morphisms—i.e., the nbd maps—have been lifted into the logic and hence are given properties via logical axioms and rules.

A planned companion paper to the current paper will use Grothendieck fibrations (see Jacobs [**?**] and his sources). The notion of distribution matches well with the notion of distribution in these fibrations, namely

through the eyes of a total category over a base category. The stalk over an object of the base category represents a local logic. The pullback and other associated functors become distributed modal operators. Some of the conditions such as Beck-Chevalley and Frobenius are expressed as modal axioms.

An Application to Computer Security. The obvious practical question is "What are distributed logics good for?". Consider Figure 1; this is a simplified view of an actual system.

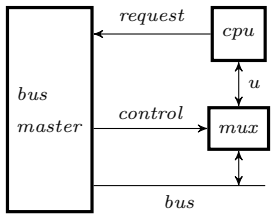


Figure 1: The Bus Master Example

The *cpu* issues a request to the *bus master* to read from the bus. The *mux* either connects line u to the bus or leaves it undefined as a "tri-state value", $\perp$, which will be used as a predicate in the security specification below. The control line tells the *mux* when to make the connection. The formulas are distributed logic statements that hold of the *bus master*:

$$(control = 0) \supset [c](\perp(u)), \quad (control = 1) \supset [c](bus = u)$$

This simplified view of a hardware bus system illustrates how reasoning in distributed logic supports formal verification of distributed computing systems. The *bus master* does not have access to the line u and, hence, u cannot be part of the *bus master*'s state. The two statements hold of any state in the *bus master* since the *control* line is either 0 or 1. Every state in the *bus master* is related to at least one state of the *cpu-mux* via the *control* line; this co-occurrence relation, which will be called C, is used in interpreting the (necessity) distributed modal connective $[c]$.

Let σ be a state in the *bus master*'s worlds where $control = 0$. The evaluation of the first statement is then

$$\begin{array}{l} \sigma \models_{bus\ master} (control = 0) \supset [c](\perp(u)) \\ \therefore\ \sigma \models_{bus\ master} [c](\perp(u)) \\ \therefore\ \text{for all } \tau \in \textit{cpu-mux}(C\sigma\tau \text{ implies } \tau \models_{cpu\text{-}mux} \perp(u)) \end{array}$$

Note how the appellation of the semantic turnstile changes from *bus master* to *cpu-mux* as the formula is evaluated.

More abstractly, some security properties of distributed systems can be expressed using these forms of logic statements. Distribution prevents taking large cross products of states which tend to degrade the performance of model checking algorithms beyond reasonable levels. Intuitively, although space prevents us from explicating it here, distributed logic statements can be paired with a process algebra where the terms yield something like a tensor product of states of the components.

There is another use for distributed logics in testing systems. The situation frequently arises where one is tasked with producing a distributed system for a system-on-a-chip where what is known as "foreign IP (intellectual property)" must be used. While in one state of a known component, tests are made to a foreign IP component. The tests generate neighborhoods about a state in which the test was made. The situation is similar to the non-normal diagram in the next section. The worlds are the states and the $\mathcal{R}$ neighborhood map indicates tests for each state (world).

2 The Logic

Distributed logics refer to all the logics with a distribution structure as we will specify it for non-normal and normal modal logics. A distributed logic starts with a directed graph where every node constitutes a *local*

logic. Each node is a classical propositional logic with a set of modal connectives, and any axioms and rules to govern behavior. The graph makes apparent the structure of the collection of the local logics. Using an arc for every modal connective can get a bit "noisy" due to classical negation and defining possibility from necessity or vice versa. Instead, arcs specify semantic maps that must exist in any interpretation. Each arc is then a bit of abstract syntax which, in an interpretation, will be turned in for a nbd map.

The collection of distributed modal connectives is specified in the axioms. These axioms can be mixed and matched depending upon the properties desired for the domain of discourse being modeled. One should look at one's axiom set as a control panel of switches and knobs which select the properties of the underlying Kripke relations. The distributed structure is typically lifted from the universe of discourse and is generally small. It is certainly possible to define meta-linguistically a very large graph of local logics and distributed modal operators. We do not do so in this paper to keep the level of abstraction to a minimum.

2.1 Conventions

The mental picture for models of two local logics h and k semantically connected by either a nbd map $\mathcal{R}$ or a relation $\mathcal{R}$ is the following diagram:

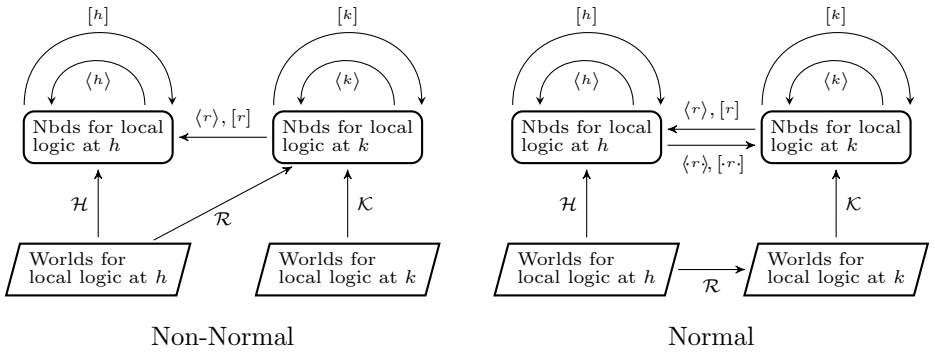


Figure 2: Mental Picture of Distributed Modal Logics

As depicted in the diagrams, the arrows labeled $\mathcal{R}$ are morphisms in a category, not functions. The $\langle r\rangle$ and $[r]$ are *forward looking* modal connectives in that their interpretation by the neighborhood map $\mathcal{R}$ looks forward along $\mathcal{R}$ from head to tail. The $\langle\cdot r\cdot\rangle$ and $[\cdot r\cdot]$ are backwards looking modal connectives. Let x be world for h and y be a world for k, then in the first diagram, $\mathcal{H}x$, $\mathcal{R}x$, and $\mathcal{K}y$ are each a collection of neighborhoods. One can add axioms for the distributed modal connectives to force the nbd maps to be simulation relations in the normal case and to respect a simulation condition for neighborhoods in the non-normal case.

Other axioms can require that the relations be functions. Using both simulation and function axioms requires that the relations be p-morphisms, and the resulting logic is simulation logic [**?**]. We simplify a bit and allow the indices h and k to refer to a local logic as well as indexing the local logic's modal connectives, and we also assume there are only the modal connectives $[k], \langle k\rangle$ in the logic for k and similarly for h. There are no problems adding more modal connectives and axioms and rules to govern their behavior. In particular, one can add conditions expressing the interaction between local modal connectives and distributed modal connectives. We use the simulation axiom (see Axiom F1 below) to illustrate this. There are a wealth of choices that are driven by the particular distributed system under consideration.

In sufficiently weak modal systems, it is not necessary that a point be a member of its neighborhoods. Here, it is almost a requirement or the notion of distribution is not present. Model theoretically, $\mathcal{R}$ relates two different neighborhood systems. These neighborhood maps, as morphisms, compose and there is an identity for each domain of worlds. In the normal case, the morphisms can be represented as relations with suitable modifications of the definitions.

The notation $\mathrm{dom}(r)$ refers to the domain or source of the arc r in a graph and $\mathrm{cod}(r)$ refers to the codomain or target of the arc, $r : \mathrm{dom}(r) \curvearrowright \mathrm{cod}(r)$. We use the locution $\langle h\rangle \in \mathrm{dom}(r)$ to refer to a modal

connective in the logic associated with the node which is the source for the arc $r : h \curvearrowright k$. The symbol $\equiv$ is used for *bi-implication*, i.e., $P \equiv Q$ stands for $(P \supset Q) \wedge (Q \supset P)$. We use the following letter conventions:

entity	description
h, k, l	nodes and endo-arcs in a graph $\mathfrak{G}$
$\langle h \rangle, [h], \langle k \rangle, [k]$	local modal connectives at nodes h and k
r, s	arcs in a graph $\mathfrak{G}$
$\langle r \rangle, [r], \langle s \rangle, [s]$	forward-looking modal connectives for arcs r and s
$\langle r \cdot \rangle, [r \cdot]$	backward-looking modal connectives for arc r
H, K	sets of worlds in interpretations for logics at h, k
$\mathcal{H}, \mathcal{K}$	interpret modal connectives for endo-arcs at h, k
$(H, \mathcal{H}, \mathbb{H}), (K, \mathcal{K}, \mathbb{K})$	neighborhood frames for the logics at h and k
$\mathcal{R}, \mathcal{S}$	interpret modal connectives for arcs r and s

We will assume, without loss of generality, that each local logic can be interpreted with a single neighborhood map. Hence, the node and its endo-arc can share the same label with use disambiguating meaning. This allows us to equate a node usually labeled h or k with the modal logic at that node.

2.2 Axioms and Rules

A local logic is "local" in that it is associated with one node in the graph. In this paper, the accompanying notion of a global logic does not entail formulas "spanning" two local logics in the sense of P in one logic implying Q in another where implying is reified as an implication connective (and similarly with other two place connectives). Each formula lives entirely within a single local logic although it may contain subformulas from others.

The distributed logic graphs we use have *endo-diagrams*, each of which is a labeled node and a single endo-arc (self-arc). Each endo-arc will be translated into an endo-morphism. Each node is required to have at least one endo-diagram whose arc will be translated into an identity morphism. This is necessary since the models for the logic will be a category. The graph axioms specify which local logics there are to be, which morphisms are to appear in any model, and force identity morphisms to exist. Each local logic may have its own propositional atoms and local modal connectives. The **S** specification and **A** and **B** axioms are not optional.

Graph Specification S:

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

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

Axiom Schemes A: For each node in $\mathfrak{G}$,

A1. all truth functional theorems of a propositional logic

A2. Modal axioms for a logic at this node

Each node h must contain an endo-diagram for each class of modal operators in its local logic. A class is the collection $\{[h], \langle h \rangle\}$ if the local logic is non-normal and $\{[h], \langle h \rangle, [h \cdot], \langle h \cdot \rangle\}$ if the local logic is normal.

Axiom Schemes B: These axioms force arcs to be interpreted as morphisms in a category. For arcs $r : h \curvearrowright k$ and $s : k \curvearrowright l$,

B1. $P \equiv [i]\, P$

B2. $[r]\, [s]\, P \equiv [s \circ r]\, P$

Axiom Schemes C: Taken all together these axioms would force the distributed modal connectives to be normal. Each may be optionally added.

C1. $[r]\, P \wedge [r]\, Q \supset [r](P \wedge Q)$

C2. $[r](P \wedge Q) \supset [r]\, P \wedge [r]\, Q$

C3. $\top \supset [r]\, \top$

The Axiom Schemes **C** should be present to specify simulation logic [**?**]; they also allow the specification of backward looking connectives residuated (see [**?**]) with their forward looking counterparts. Simulation logic could also be built on a non-normal basis using the same main simulation axiom. However, the semantic conditions then involve neighborhoods, not relations.

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,

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

Rule B: For each $r : h \curvearrowright k$ arc in $\mathfrak{G}$,

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

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

We will only be concerned with the forward versions of necessity and possibility connectives since the backwards versions are so similar. The backward versions are only present for normal systems.

2.3 Options

Axiom Schemes D: The **D** axioms are examples of extra properties to be enforced on the interpreting morphisms. Other axioms can be added as well, we use these as paradigm examples:

D1. $[r]\, P \supset \langle r \rangle\, P$ D2. $\langle r \rangle\, P \supset [r]\, P$

In non-normal systems, the axiom D1 specifies consistency and the axiom D2 specifies completeness, both with respect to the collection of neighborhoods about any world when the world is in the source of the nbd map used in interpreting $[r]$ and $\langle r \rangle$. In normal systems, the first specifies the interpreting relation be total on its domain and the second that it act functionally (see Section 3.1).

Axiom Schemes E: The axiom E1 is only necessary if you wish the classical proposition logic at $\mathrm{dom}(r)$ to be included in the logic at $\mathrm{cod}(r)$. This condition is part of the definition of simulation [**?**] although it is not strictly necessary in that it can be removed without damaging the logic.

For all propositional letters p,

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

From now on, a distributed logic contains at least the specification **S** and axiom schemes **A** and **B**, and the Definition of Possibility, and the rules **A** and **B**. Normal distributed logics include the non-normal axioms and rules and the Axioms Schemes **C**. Axiom Schemes **C** can also be added individually rather than en masse if only a subset of the properties of normality are desired. The Axiom Schemes **D** are of interest and we have modeling conditions for them. The Axiom Scheme **E** must be handled quite separately in the semantics. Other axioms can be added, we stop with the list chosen for the purposes of this presentation.

Axiom Scheme F: Simulation logic [**?**] requires for an arc $r : h \curvearrowright k$ in $\mathfrak{G}$, and modal connectives $[h] \in \mathrm{dom}(r), [k] \in \mathrm{cod}(r)$,

F1. $\langle r \rangle\, [k]\, P \supset [h]\, \langle r \rangle\, P$

In normal distributed logics, the axiom F1 forces the arcs in the graph to be interpreted as simulation relations and B2 forces composition of relations to hold, where a simulation relation is one "half" of a bisimulation [**?**]. One common use of the simulation relation is when the interpretation of $\langle r \rangle$ via a relation $\mathcal{R}$ is a p-morphism. To force this, add the Axiom Schemes **C** and **D** to the simulation axiom.

3 Frames and Algebras

In keeping with our simplifications, assume there is only one local modality per frame, including both a $\Box$ and $\Diamond$ since they are inter-definable. More modal connectives can be added if needed if needed by the particular distributed system under consideration.

3.1 Frames

Definition 3.1.1 *A* neighborhood frame *is a structure $\mathcal{H} = (H, \mathcal{H}, \mathbb{H})$ such that H is a collection of worlds, $\mathbb{H}$ is a collection of* neighborhoods *which are subsets of H and the entire collection is closed under the Boolean operations and under the operations $[h], \langle h \rangle : \mathbb{H} \to \mathbb{H}$ given by:*

$$[h]\, C \stackrel{def}{=} \{x \in H \mid C \in \mathcal{H}x\}, \quad \langle h \rangle\, C \stackrel{def}{=} \{x \in H \mid -C \notin \mathcal{H}x\},$$

with where $-C$ is the set complement of C in H. $\mathcal{H} : H \to \mathcal{P}\mathbb{H}$ is a nbd map taking every world of H into a collection of neighborhoods. We use the same symbol for the frame and its nbd map, and let use disambiguate what is meant.

Each node in a distributed logic's graph has a local logic associated with it. That local logic, in turn, must have a neighborhood frame associated with it.

Definition 3.1.2 *Let $\mathcal{H}$ and $\mathcal{K}$ be neighborhood frames. A* nbd map *$\mathcal{R} : \mathcal{H} \to \mathcal{K}$ is a map (also using the symbol $\mathcal{R}$) $\mathcal{R} : H \to \mathcal{P}\mathbb{K}$ such that for any $C \in \mathbb{K}$,*

$$[r]\, C \stackrel{def}{=} \{x \in H \mid C \in \mathcal{R}x\} \in \mathbb{H}, \quad \langle r \rangle\, C \stackrel{def}{=} \{x \in H \mid -C \notin \mathcal{R}x\} \in \mathbb{H}.$$

Let $\mathcal{R} : \mathcal{H} \to \mathcal{K}$ and $\mathcal{S} : \mathcal{K} \to \mathcal{L}$ be morphisms. The identity morphism $I : \mathcal{H} \to \mathcal{H}$ and the composition $\mathcal{S} \circ \mathcal{R} : \mathcal{H} \to \mathcal{L}$ are defined with ($x \in H$)

$$Ix \stackrel{def}{=} \{C \in \mathbb{H} \mid x \in C\}, \quad (\mathcal{S} \circ \mathcal{R})_x \stackrel{def}{=} \{C \in \mathbb{L} \mid \{y : C \in \mathcal{S}y\} \in \mathcal{R}_x\}.$$

Each arc $r : h \curvearrowright k$ of the graph must be associated with a *semantic morphism* in the interpretation. The semantic morphisms are *neighborhood maps* $\mathcal{R} : H \to \mathcal{P}\mathbb{K}$ where $\mathbb{K}$ is the collection of neighborhoods, i.e., the $\mathbb{K}$ in $(K, \mathcal{K}, \mathbb{K})$. In the normal case, the neighborhood maps can be replaced with relations. These relations are derivable in the usual way [**?**], i.e., $\mathcal{R}xy$ iff $y \in \bigcap \mathcal{R}x$; that is, take intersection of all the neighborhoods at x under $\mathcal{R}$.

Note that the definition for composition can be rewritten as

$$(\mathcal{S} \circ \mathcal{R})_x \stackrel{def}{=} \{C \in \mathbb{L} \mid [s]\, C \in \mathcal{R}_x\}$$

using the Definition 3.1.2 for $[s]\, C$. The definition is found in Manes [**?**] for the Kleisli category of the double power set monad. Our models are always in the category of neighborhood frames.

Each node representing a distinct local logic must be mapped to a distinct frame object in any interpretation. This informal way of restricting interpretations is the result of treating the graph as not defining everything in a distributed logic, but the alternative would make the logic impenetrable.

The corresponding Kripke frame conditions for the logical axioms are

Frame Conditions S:

FS1. A category of *local neighborhood frames* and neighborhood maps	FS2. An identity morphism for the i arc in $D \in \mathfrak{D}$

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

FA1. A set of classical worlds

FA2. Frame conditions for a local logic at this node

Frame Conditions B: For $I : H \to \mathbb{H}$, $\mathcal{R} : H \to \mathbb{K}$ and $\mathcal{S} : K \to \mathbb{L}$ in $\mathfrak{G}$

FB1. $I_x = \{C \in \mathbb{H} \mid x \in C\}$

FB2. $(\mathcal{S} \circ \mathcal{R})_x = \{C \in \mathbb{L} \mid [s]\, C \in \mathcal{R}_x\}$

Frame Conditions C:

FC1. $B, C \in \mathcal{R}_x$ implies $B \cap C \in \mathcal{R}_x$

FC2. $B \in \mathcal{R}_x$ and $B \subseteq C$ implies $C \in \mathcal{R}_x$

FC3. $\top \in \mathcal{R}_x$

Frame Conditions D:

FD1. $C \in \mathcal{R}_x$ implies $-C \notin \mathcal{R}_x$

FD2. $C \notin \mathcal{R}_x$ implies $-C \in \mathcal{R}_x$

Frame Condition F:

FF1. $-\{y \mid C \in \mathcal{K}y\} \notin \mathcal{R}x$ implies $\{z \mid -C \notin \mathcal{R}z\} \in \mathcal{H}x$

with the convention that the nbd maps that use upper case script relation letters will interpret modal connectives that use the corresponding lower case Roman letters. Each distributed frame category interpreting a distributed 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.

Slightly different frames are used for the axiom E1; the local frames will contain functions to interpret constants, one for every atomic proposition of the local logic for which the local frame provides a model.

The following proposition allows for the use of one neighborhood frame per local logic.

Proposition 3.1.3 *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 and rules. The consequence is that no formula in the logic has a binary connective between formulas in two different local logics.

Note that we stated the 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 would give more than one instance of the logic in the entire distributed logic.

Using the semantics conditions, it is easy to show that

$$x \models_{\mathcal{H}} \neg\, [r] \neg P \text{ iff } x \models_{\mathcal{H}} \langle r \rangle\, P,$$

hence the definition of $\langle r \rangle$ in terms of $[r]$ makes sense. A distributed category model has neighborhood frames for every node with a valuation for each node. The morphisms are neighborhood maps.

Definition 3.1.4 *A* distributed category model *is a neighborhood frame category with a valuation and a local frame for each local logic. The local frame and its valuation are called a* local model. *A valuation specifies a collection of points in the local frame where the atomic propositions are true.*

3.2 Algebras

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 3.2.1 (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 3.2.2 *A* distributed algebra *appropriate for a distributed 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* distributed operators $\langle r \rangle$ *and* $[r]$ *for every arc r of the graph. For $r : h \curvearrowright k$ in the graph,*

- $[r]\,[s]\,a = [s \circ r]\,a$;
- $[i]\,a = a$, *for the i arc in an endo-diagram;*
- *if the Axiom Schemes* ***C*** *are used*
  - $[r]\,a \wedge [r]\,b \leq [r](a \wedge b)$;
  - $[r](a \wedge b) \leq [r]\,a \wedge [r]\,b$;
  - $\top_{\mathbb{H}} = [r]\,\top_{\mathbb{K}}$, *for $\top$ the top of a Boolean lattice;;*
- *if Axiom Schemes* ***D*** *are used*
  - $[r]\,a \leq \langle r \rangle\,a$;
  - $\langle r \rangle\,a \leq [r]\,a$;
- $\langle r \rangle\,[k]\,a \leq [h]\,\langle r \rangle\,a$, *if Axiom Scheme* ***F*** *is used.*

The axiom E1 will be handled in the next subsection where we must add constant operations and functions to help interpret the propositional atoms.

Appropriate distributed 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 $r : h \curvearrowright k$ of the graph map from a local modal algebra to a local modal algebra. This stratifies the heterogeneous distributed algebra and treats every local modal algebra as an object in the surrounding distributed 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 distributed operators.

Lemma 3.2.3 *The distributed operators respect bi-equivalence.*

The connective $[r]$ respects bi-equivalence because of the Rule **B**. Using Boolean negation, it is easy to show that $\langle r \rangle$ does as well.

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

Lemma 3.2.4 *The LT distributed operators satisfy the required properties for a distributed algebra.*

The equivalence classes for the LT algebras are defined (as usual) with $[\![P]\!] = \{Q \mid \vdash_{\mathcal{H}} P \equiv Q\}$. The operators are defined inductively, i.e., $[\![P]\!] \wedge [\![Q]\!] = [\![P \wedge Q]\!]$, $[r][\![P]\!] = [\![[r]\, P]\!]$.

Corollary 3.2.5 *The LT heterogeneous algebra is a distributed 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 3.1.3 shows that no additional sorts over and above the local modal algebra carrier sets are necessary. Lemma 3.2.3 shows that the replacement property for the bi-implication congruence holds for each operator. Finally, Lemma 3.2.4 shows each of LT operators satisfy the distributed algebra axioms. ∎

Theorem 3.2.6 *Distributed Logic is sound with respect to the algebraic and distributed frame category models.*

outline: Soundness over the algebraic models is an induction starting with a valuation into a distributed algebra and then using the fact that the LT algebra is a free algebra for the heterogeneous class of distributed algebras. From this, it is easy to see that $\supset$ interprets to $\leq$ in the algebra. The axioms of the LT algebra clearly interpret to the axioms of the logic, and the rules of the logic preserve truth in the algebra. The free heterogeneous algebras are then used to generate the universal morphism for any interpretation into a heterogeneous modal algebra thus validating the axioms and rules.

The Frame Conditions FS1, FS2, FB1, and FB2, given the work in Manes [**?**] on the double power set monad restricted to neighborhoods, show that the neighborhood maps are the Kleisli morphisms and hence form a category, so the identity and associative laws of categories are met. In the presence of the normal axioms, the previous prescription for manufacturing relations from neighborhood maps shows these frame conditions ensure the maps act like Kleisli morphisms for the power set monad restricted to neighborhoods.

The rest of the axioms and rules are easily checked.

∎

The *canonical frame* is generated by the LT algebra; the frame's neighborhoods are the output of representation function for the LT algebra. The representation function β is defined by

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

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 space containing all the maximal filters of each local algebra and define the local neighborhood maps with:

$$\beta a \in \mathcal{H}x \text{ iff } [h]\, a \in x.$$

Since $[h]$ and $\langle h \rangle$ are DeMorgan duals of each other and β is a homomorphism,

$$-\beta a \notin \mathcal{H}x \text{ iff } \beta\neg a \notin \mathcal{H}x \text{ iff } [h]\, \neg a \notin x \text{ iff } \neg\, [h]\, \neg a \in x \text{ iff } \langle h \rangle \in x.$$

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

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 $r : h \curvearrowright k$.

Lemma 3.2.7 *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: $x \in \beta\, [r]\, a$ iff $[r]\, a \in x$ iff $\beta a \in \mathcal{R}x$ iff $x \in [r]\, \beta a$.The proof for $\langle r \rangle$ is similar. ∎

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 distributed algebra for each carrier set and the operations using the above prescriptions.

Theorem 3.2.8 *Distributed Logic is complete with respect to the distributed algebras and the distributed category models.*

Proof: From Proposition 3.1.3, 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 3.2.5 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 distributed category model falsifying the non-theorem.

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

3.3 The Axiom Schemes E

The axiom E1 requires some special treatment. The algebra will now have a collection of constant operators, one for each propositional atom in the language.

Definition 3.3.1 *An* E local modal algebra *is a local modal algebra with a collection of (local) constant operations. Note that two constant operations, being functions, can point to the same element of the local modal algebra. The Lindenbaum-Tarski 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 arc 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 $\mathrm{dom}(r)$.

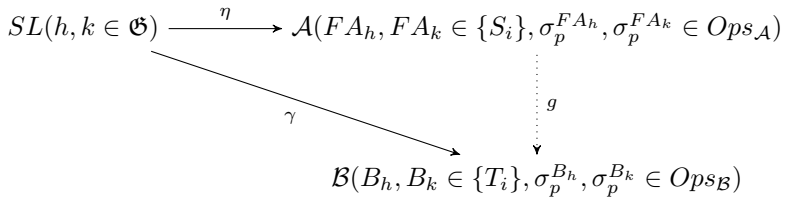
Definition 3.3.2 *A* E neighborhood frame *is a neighborhood frame with a collection of constant functions, f_p, one for each propositional atom. A constant function selects an element of the set algebra, i.e., a neighborhood.*

Fix a distributed algebra with any necessary E local modal algebra. 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 algebras which agree on everything except the nullary operations. So the variation gets satisfied at a slightly higher level. A similar statement holds for E neighborhood frames. The inductive definition generating interpretations from valuations remains the same and hence the restriction on valuations gets transferred to interpretations.

Definition 3.3.3 *An* E local 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 neighborhood frame and valuations $[\![-]\!]$, we demand $[\![p]\!] = C$ if $f_p = C$. Also, we demand that for*

$r : h \curvearrowright k$, the r interpreting relation $\mathcal{R}$ must respect the constant functions in the sense that $x \in f_p$ at the neighborhood frame for h and $f_p \in \mathcal{R}x$ at the neighborhood 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 algebras to which the nullary operations point for the respective propositional atoms. In the freeness diagram below, p indicates some propositional atom in the language, FA_h is the carrier set of the local modal logic for h inside of the free algebra $\mathcal{A}$. The algebra $\mathcal{B}$ is some other appropriate distributed algebra, and γ 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, are preserved by g. Hence, $\eta(p) = \sigma_p^{FA_h}$ and $g(\eta(p)) = g(\sigma_p^{FA_h}) = \sigma_p^{FB_h}$. Since the diagram commutes, $\gamma(p) = \sigma_p^{FB_h}$.

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

Theorem 3.3.4 *Distributed logics with the **E** axioms are sound and complete with respect to **E** distributed algebras and **E** distributed category models.*

4 Cheap Entailment Arrows

We use the word "cheap" because we embed the entailment arrow into a distributed modal logic. This is as opposed to treating the entailment arrows as more like relevance logic's entailment. In later work, we will show how to distribute relevance logic's entailment arrow; for now we stick to modal logic due to its simplicity and that its models are relatively free of auxiliary partial orders such as is necessary for relevance logic.

Gödel's embedding of intuitionistic entailment into S4 leaves open the possibility that an intuitionistic entailment might have a distributed counterpart. Barwise and Seligman [**?**] used a similar encoding but into S5 for an entailment. In their setup, worlds are essentially three-valued; they use a partial function from a world and a proposition into $\{+, -\}$. S4 and S5 suffer from the problem that the Kripke relations used in interpretation are required to be reflexive and transitive which would cause too much of the distributed structure of the interpreting category to collapse. However, one can can still use the general idea of modal embedding.

We take the property of residuation to be the primary feature of an entailment arrow. The residuation partner of an entailment we will term (following Dunn) *fusion*. The care and feeding of fusion dictates the distributed nature of the entailments.

In the sequel, we sometimes use superscripts over operators to indicate to which logic they belong, i.e., $\overset{h}{\supset}$ is classical implication for the local logic at node h in the graph of a distributed logic. The general rule of thumb is that classical logical connectives live entirely within a single logic whereas distributed intensional and modal connectives take their arguments from various logics. The result is always a formula that lives entirely within a logic, i.e., it is not a formula of some (non-existent) global logic.

In the sequel, we use a notion from [**?**] (although it is not original with them) of a *classification* containing two sets, a set containing the logic *over* a containing the models or worlds. They are connected with a

satisfaction relation $\models$. We use the terminology *classification of* h to refer to the classification at a node h in a distributed logic's graph and $\models_h$ is the satisfaction relation at h.

4.1 Gödel Entailment and Fusion

This version arose from insisting that the interpretation of the intuitionistic entailment retain the form of its first order logic evaluation, although dropping the restrictions of S4 on the interpreting relation. A picture will display nicely where the various formulas live.

Referring to the diagram below, we use $\Rightarrow$ and $\Leftarrow$ for the entailment operators. Later it will turn out these are identical, but for now we will leave them as separate operators. The $\circledcirc$ operator is a distributed intensional conjunction. The following diagram shows the logic (top) layer for two logics h and k and the model (bottom) layer of their respective sets of worlds H and K. The arrows between the logics are logical operators. As before, $[r]$ is the forward looking necessity interpreted with the relation $\mathcal{R}$ and $\langle r\rangle$ is the backwards looking possibility operator interpreted with $\mathcal{R}$.

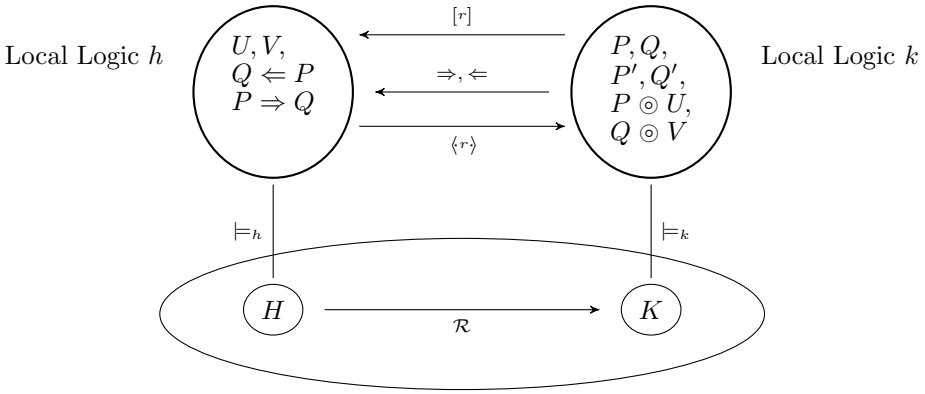


Figure 3: Gödel Entailment and Fusion

The evaluation condition is for $\circledcirc$ is

$$y \models_k P \circledcirc U \text{ iff } \exists x(\mathcal{R}xy \text{ and } x \models_h U \text{ and } y \models_k P).$$

The representation of $\Rightarrow$ using the S4 necessity is lost to us because reflexivity of the relation would necessitate that there be a single logic, i.e., no distribution. Transitivity also tends to destroy distribution. The evaluation condition for $\Rightarrow$ is

$$x \models_h P \Rightarrow Q \text{ iff } \forall y(\mathcal{R}xy \text{ and } y \models_k P \text{ implies } y \models_k Q).$$

Note that these modeling conditions are precisely the modeling conditions for the coding

$$P \Rightarrow Q \stackrel{def}{=} [r](P \supset Q).$$

and the conditions for $\circledcirc$ generate the encoding

$$P \circledcirc U \stackrel{def}{=} P \wedge \langle r\rangle U.$$

Theorem 4.1.1 *The two residuation axioms*

$$U \stackrel{h}{\supset} \big(P \Rightarrow (P \circledcirc U)\big), \qquad \big(P \circledcirc (P \Rightarrow Q)\big) \stackrel{k}{\supset} Q,$$

which dictate into which logics the formulas fall, are valid.

One adds the following monotonicity rules:

$$\frac{U \vdash_h V \qquad P \vdash_k Q}{P \circledcirc U \vdash_k Q \circledcirc V} \circledcirc -monotonicty \qquad\qquad \frac{P \vdash_k Q \qquad P' \vdash_k Q'}{P \Rightarrow Q \vdash_h P' \Rightarrow Q'} \Rightarrow -monotonicty$$

From the axioms and rules, the usual (bidirectional) form of residuation is derivable:

$$\frac{U \overset{h}{\supset} (P \Rightarrow Q)}{P \circledcirc U \overset{k}{\supset} Q} \; residuation$$

One might conjecture there is a second fusion operator with the evaluation condition

$$y \models_k U \circledcirc' P \text{ iff } \exists x(\mathcal{R}xy \text{ and } x \models_h U \text{ and } y \models_k P).$$

and a second entailment $\Leftarrow$ operator with the evaluation condition

$$x \models_h Q \Leftarrow P \text{ iff } \forall y(\mathcal{R}xy \text{ and } y \models_k P \text{ implies } y \models_k Q)$$

is the same as $\Rightarrow$ simply because there is no additional freedom to alter it given the Gödel evaluation form we are following and where the formulas must fall in the distribution.

The $\Leftarrow$ and $\circledcirc'$ have the following two residuation properties

$$U \overset{h}{\supset} \big((U \circledcirc' P) \Leftarrow P\big), \qquad \big((Q \Leftarrow P) \circledcirc' P\big) \overset{k}{\supset} Q,$$

which again dictate into which logics the formulas fall. Residuation in the following form holds:

$$\frac{U \overset{h}{\supset} (Q \Leftarrow P)}{U \circledcirc' P \overset{k}{\supset} Q} \; residuation$$

It is then easy to show that $\circledcirc$ is the same operator as $\circledcirc'$.

This would leave the embedding of $\circledcirc'$ and $\Leftarrow$ as the same as for $\circledcirc$ and $\Rightarrow$. If one were to treat these operators as native rather than defined, then one needs to add the axioms

$$(P \circledcirc U) \overset{k}{\supset} (U \circledcirc' P) \qquad\qquad (U \circledcirc' P) \overset{k}{\supset} (P \circledcirc U)$$

and

$$(P \Rightarrow Q) \overset{h}{\supset} (Q \Leftarrow P) \qquad\qquad (Q \Leftarrow P) \overset{h}{\supset} (P \Rightarrow Q)$$

because the semantics is going to identify $\circledcirc$ with $\circledcirc'$ and $\Rightarrow$ with $\Leftarrow$.

It is necessary to have an intensional conjunction, $\circledcirc$, rather than an extensional conjunction, $\wedge$, because in the statement of residuation, (rewritten) the lower premise $P \wedge U \overset{k}{\supset} Q$ would require that P and U be in the same classification for $\wedge$ to make sense and also that Q be in that same classification for $\overset{k}{\supset}$ to make sense.

4.2 Simple Entailment and Fusion

A simple version of relevant-like entailment is definable. Let there be the following setup:

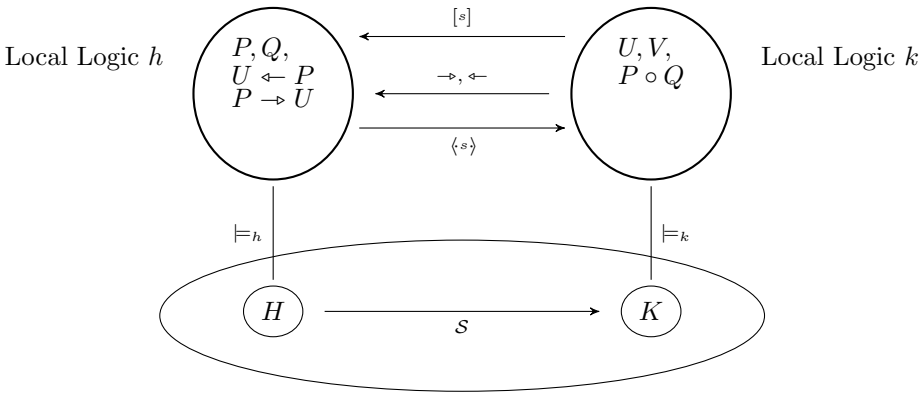


Figure 4: Simple Entailment and Fusion

It turns out that there is no distinction between $\rightarrowtriangle$ and $\multimap$ mainly because $\circ$ must be symmetric. This latter is so because the evaluation condition is

$$y \models_k P \circ Q \text{ iff } \exists x(\mathcal{S}xy \text{ and } x \models_h P \text{ and } x \models_h Q).$$

The $\rightarrowtriangle$ connective has the following evaluation condition:

$$x \models_h P \rightarrowtriangle U \text{ iff } \forall y(\mathcal{S}xy \text{ and } x \models_h P \text{ implies } y \models_k U).$$

Since x is a free variable and $x \models_h P$ does not rely on the $\forall y$ quantifier, we can rewrite this as

$$x \models_h P \rightarrowtriangle U \text{ iff } x \models_h P \text{ and } \forall y(\mathcal{S}xy \text{ implies } y \models_k U).$$

The two evaluation conditions show the following definitions can be made

$$P \rightarrowtriangle U \stackrel{def}{=} P \stackrel{h}{\supset} [s]\, U, \quad P \circ Q \stackrel{def}{=} \langle s \rangle (P \wedge Q).$$

Theorem 4.2.1 *The two residuation properties*

$$Q \stackrel{h}{\supset} \big(P \rightarrowtriangle (P \circ Q)\big), \qquad \big(P \circ (P \rightarrowtriangle U)\big) \stackrel{k}{\supset} U,$$

which dictate into which logics the formulas fall, are valid.

The following monotonicity rules must be added:

$$\frac{P \vdash_h P' \qquad Q \vdash_h Q'}{P \circ P' \vdash_k Q \circ Q'} \circ - \textit{monotonicty} \qquad \frac{P \vdash_h Q \qquad U \vdash_k V}{Q \rightarrowtriangle U \vdash_h P \rightarrowtriangle V} \rightarrowtriangle - \textit{monotonicty}$$

Residuation holds in this distributed setting with the bidirectional rule:

$$\frac{Q \stackrel{h}{\supset} (P \rightarrowtriangle U)}{P \circ Q \stackrel{k}{\supset} U} \textit{ residuation}$$

It is clear there can be only one fusion operator and not a second $\circ'$ given the symmetry of the semantics. Due to the symmetry and residuation, there can be only a single entailment connective and so $\rightarrowtriangle$ and $\multimap$ collapse into a single entailment. Also, if these operators are treated as native, one also needs

$$(P \circ Q) \stackrel{k}{\supset} (Q \circ' P) \qquad (Q \circ' P) \stackrel{k}{\supset} (P \circ Q)$$

and

$$(P \rightarrowtriangle U) \stackrel{h}{\supset} (U \multimap P) \qquad (U \multimap P) \stackrel{h}{\supset} (P \rightarrowtriangle U)$$

since the semantics will identify the respective operators.

4.3 Preservation Conditions

We work first with Gödel Entailment. Assume the following confluent diagram, appropriating the categorical notation of Freyd and Scedrov [**?**] (although the diagram is not category theoretic)

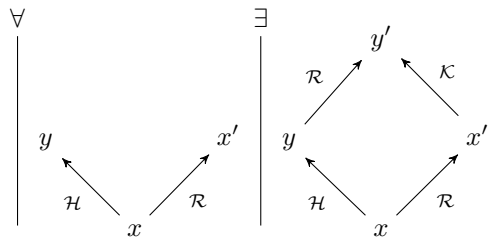


Figure 5: Simulation Condition

where one reads from left to right the Simulation Condition,

> 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):

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

From Simulation Logic [**?**], the condition is the first-order Simulation Condition for modal logics. The axiom on the right is the Simulation Axiom. The Simulation Condition validates the axiom. The modal operators $\langle r \rangle$, $[k]$, and $[h]$ are interpreted by the relations $\mathcal{R}$, $\mathcal{K}$, and $\mathcal{H}$ respectively in the Simulation Condition. Again, we allow the indices h and k to refer to a local logic as well as indexing the local logic's modal operators, and we also assume there are only the modal operators $[k], \langle k \rangle$ in the logic at k and similarly at h, the mental picture of two local logics h and k semantically connected by a simulation $\mathcal{R}$ is

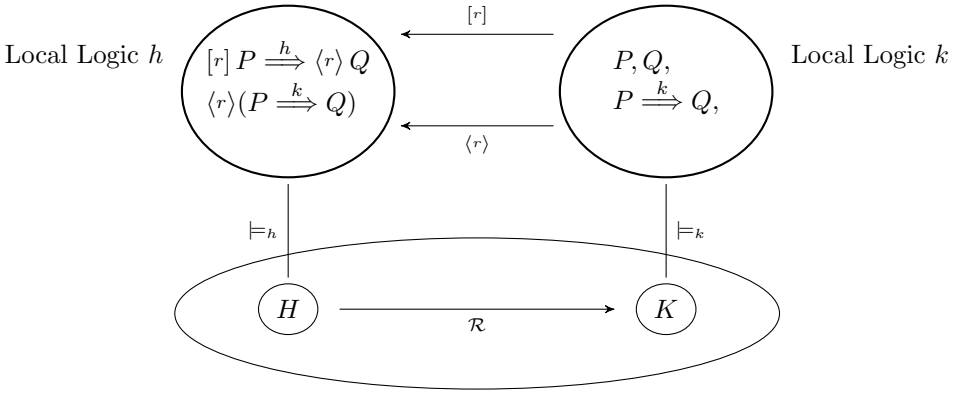


Figure 6: Preserving Gödel Entailment

The confluent condition underwrites the Gödel entailment preservation in the form of the following axiom replacing the Simulation Logic axiom:

$$\langle r \rangle (P \overset{k}{\Longrightarrow} Q) \overset{h}{\supset} \big([r]\, P \overset{h}{\Longrightarrow} \langle r \rangle\, Q \big)$$

The following is a derived rule by virtue of residuation and the fact that the $\langle r \rangle$ operator is monotone:

$$\frac{\langle r \rangle (P \overset{k}{\Longrightarrow} Q) \overset{h}{\supset} \big([r]\, P \overset{h}{\Longrightarrow} \langle r \rangle\, Q \big)}{\big([r]\, U \overset{h}{\circ} \langle r \rangle\, V \big) \overset{h}{\supset} \langle r \rangle (U \overset{k}{\circ} V)}$$

One can also use residuation to take the conclusion of this derived rule as an axiom and derive the premise.

Now for the simple entailment preservation of $\overset{h}{\rightarrowtriangle}$. The Freyd-Scedrov diagram is

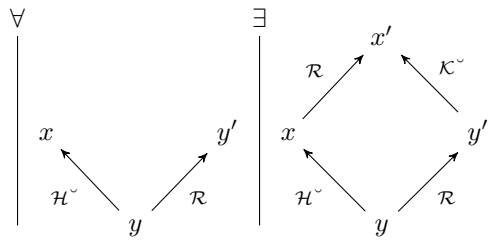


Figure 7: Simple Simulation Condition

where $\mathcal{H}^\smile$ and $\mathcal{K}^\smile$ refer to the converse of $\mathcal{H}$ and $\mathcal{K}$, and are used because $\circ$ is a backward's looking operator. This condition validates

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

The confluent condition also underwrites the simple fusion preservation in the form of the following axiom:

$$\big([r]\, P \overset{h}{\circ} [r]\, Q \big) \overset{h}{\supset} [r](P \overset{k}{\circ} Q)$$

The following is a derived rule by virtue of residuation and the fact that the $[r]$ operator is monotone:

$$\frac{\big([r]\, P \overset{h}{\circ} [r]\, Q \big) \overset{h}{\supset} [r](P \overset{k}{\circ} Q)}{[r](P \overset{k}{\longrightarrow\!\!\!\triangleright} Q) \overset{h}{\supset} [r]\, P \overset{h}{\longrightarrow\!\!\!\triangleright} [r]\, Q}$$

As before, one can also use residuation to take the conclusion of this derived rule as an axiom and derive the premise.

5 Noninterference as a Simulation

Noninterference is a security property that is frequently imposed on state-based systems that process a combination of high security and low security data. Note that we refer to data here and not information. Information requires a more sophisticated typing scheme beyond mere values. To go one step further, knowledge requires a certain relationship between an agent and information. So we are concerned here with the most basic form of processing.

High level security properties are generally expressed informally using distributed notions. However, when coerced into formal models, they frequently lose their distribution and the notion of a cross-product of system states replaces the notion of distribution. This has the effect of making the analysis complicated because then the distribution structure has to be disentangled from the combined system. Distributed logic cuts through the encode-the-distribution and subsequent decode-the-distribution steps. In this section, we show how to view noninterference in a distributed setting.

A high security process (one that processes high security data) influences a low security process just when ignoring the effects of the high security process changes the behavior of the low security process. The high and low processes are run together and compared against the running the low process with high's output deleted. The reason for running the high and low processes together for the comparison is to account for any covert channels from high to low that are not accounted for merely by looking at high's output. Even then, using Shannon information theory, it is possible through the measurement process to pass as much information as you like through a channel of capacity zero [**?**]. This occurs because capacity is defined via a mathematical limiting process. Put quickly, sending one bit every 2^n time periods results in sending the ability to send 0 bits per time period at the limit regardless of how many were sent getting there.

Noninterference can be expressed using a particular simulation relation which builds in the deletion of high's output to what low can see. So in effect, we are choosing a canonical simulation relation for the combined high-low system; this is a simulation relation intimately tied to non-interference.

Let there be a High and Low system h and a Low system k. If h noninterferes with k then there is a simulation relation so that every move that h can make is simulated by k. However, this is not enough. It could be that k is simply reading h's output and that would be interfering. So it is not enough to claim that noninterference means the existence of a simulation relation. The simulation relation must have some other properties. In [**?**], they state that noninterference exists when a particular equation holds. Let *out* be an output function, w is a finite sequence of commands issued by users, each command and user pair denoted (b, v), v is a user in k, and b is a read command which reads the state (we presume) but can only be executed by a Low user. This latter is important because in the equation below, the left hand side almost appears as though either High or Low can execute b when in fact, it can only be Low. b is in a collection B of Low read commands. (Due to a nomenclature clash, we have changed Goguen and Meseguer's r to a b and $P_{G,A}$ to $S_{G,A}$).

$$\text{out}\Big([[w]], v, b\Big) = \text{out}\Big([[S_{G,A}(w)]], v, b\Big),$$

where $S_{G,A}$ purges the (b, v) for any v in High alone executing a command in A. Commands in A are the ones requiring protection from snooping.

5.1 Derivation of the Simulation Relation

Let $\mathcal{H}$ be the next state relation on the classification h side and $\mathcal{K}$ be the next state relation on the classification k. The next state relation is a combination derived from all the commands a particular side can produce. Since we are abstracting over all commands, it is okay the effect of a set of commands to be represented by a relation, i.e., we do not care about any one particular command.

The simulation axiom is

$$\langle r \rangle\, [k]\, Q \supset [h]\, \langle r \rangle\, Q$$

for all properties Q of k. The left hand side becomes

$$\exists y \in k(\mathcal{R}xy \text{ and } \forall z \in k(\mathcal{K}yz \text{ implies } z \models_k Q))$$

and the right hand side is

$$\forall u \in h(\mathcal{H}xu \text{ implies } \exists z \in k(\mathcal{R}uz \text{ and } z \models_k Q)).$$

It helps to observe that $\mathcal{K}$ is the result of purging the output of High's commands. The fact that b is a low command means that Q is a statement made about what Low can observe. In effect, the combination of out and b stands for $z \models_k P$. More to the point, b stands for a reification of a state z and out stands for $\models_k$. P is something we can observe from out.

$\mathcal{R}$ stands for stripping out High's state information since it relates h states with k states. If x is a state in h, then it must contain both High and Low information. y is only a state in k and hence can contain only Low's information. Now we attempt to rewrite Goguen and Meseguer. Recall

$$\text{out}\Big([[w]], v, b\Big) = \text{out}\Big([[S_{G,A}(w)]], v, b\Big),$$

Here, v is more or less a useless parameter, we assume the user is some v. So we can drop v. Every command must have start and stop states and $[[w]]$ relates start and stop states. Hence we can rewrite with $[[w]]$ standing for a relation in infix notation:

$$x\ [[w]]\ u \text{ and out}(b(u)) \text{ iff } y\ [[S_{G,A}(w)]]\ z \text{ and out}(b(z)).$$

where b reads the state and out produces some output value. We can abstract the output value into some predicate, Q, and assume the predicate is being held true or false of a k state. The relation $\mathcal{R}$ represents stripping a $h + k$ state of its high component and returning its low component. Hence

$$x\ [[w]]\ u \text{ and } \mathcal{R}uz' \text{ and } \text{out}(b(z')) \text{ iff } y\ [[S_{G,A}(w)]]\ z \text{ and } \text{out}(b(z)).$$

The out and b combination is awkward, we replace that with a predicate

$$x\ [[w]]\ u \text{ and } \mathcal{R}uz' \text{ and } Q(z') \text{ iff } y\ [[S_{G,A}(w)]]\ z \text{ and } Q(z).$$

Both computations should start out in the same state and they would were they both in the same state space. However, they are not, and we use $\mathcal{R}$ again to strip out h's information from x:

$$x\ [[w]]\ u \text{ and } \mathcal{R}uz' \text{ and } Q(z') \text{ iff } \mathcal{R}xy \text{ and } y\ [[S_{G,A}(w)]]\ z \text{ and } Q(z).$$

All the commands in h together can be abstracted into a next state relation $\mathcal{H}$ and similarly for k using $\mathcal{K}$:

$$\mathcal{H}xu \text{ and } \mathcal{R}uz' \text{ and } Q(z') \text{ iff } \mathcal{R}xy \text{ and } \mathcal{K}yz \text{ and } Q(z).$$

If $z \neq z'$ then l would be interfered with, so we make them equal:

$$\mathcal{H}xu \text{ and } \mathcal{R}uz \text{ and } Q(z) \text{ iff } \mathcal{R}xy \text{ and } \mathcal{K}yz \text{ and } Q(z).$$

Since $Q(z)$ is parametric and appears on both sides of the iff, we can remove it:

$$\mathcal{H}xu \text{ and } \mathcal{R}uz \text{ iff } \mathcal{R}xy \text{ and } \mathcal{K}yz.$$

This statement expresses the following confluent diagram

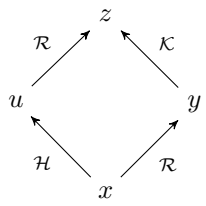


Figure 8: Confluent Diagram

We have ignored quantification over states. Certainly the confluent diagram does not hold for any u, x, y, and z, only the states in the proper relations. u is restricted by $\mathcal{H}$ and y by $\mathcal{R}$. z is determined by $\mathcal{K}$ and $\mathcal{R}$. The first is of the form "for any x and u such that $\mathcal{H}xu$, and the latter by "for any x and y such that $\mathcal{R}xy$, even if $\mathcal{R}$ appears as stripper function. z is determined by an existential which is hidden by using a function, namely being a state determined by a stripper function but satisfying $\mathcal{K}yz$.

Now we must motivate the reformulation in terms of a simulation. We need only be concerned with whether h can make a move that k possibly cannot mimic. There are two parts, h must make a move recorded by $\mathcal{H}xu$ and the beginning state for the move, x, must be strippable into a state y of k. If h can make no such move or the beginning state is not strippable into a state for k, then we have "do not car" situation. The equality is really masking the fact that the statement should read

> Were h to execute the sequence of commands w, then k could not detect the difference from executing the sequence of w with h commands stripped out.

This situation is neatly handled by using a conditional

$$\mathcal{H}xu \text{ and } \mathcal{R}xy \text{ implies } \exists z(\mathcal{K}yz \text{ and } \mathcal{R}uz).$$

which is the Simulation Condition. Now we can bring back the parametric $Q(z)$. Since z is a world, and the worlds are being abstracted away in modal logic, what is left over is just the proposition Q. Here there a choice, in order to be a logic, Q should be a metalinguistic variable and range over all propositions. However, there is no need to go this far, Q can be restricted to only range over propositions that are critical to a particular implementation. Either way, the axiom

$$\langle r \rangle [k] Q \supset [h] \langle r \rangle Q$$

is certainly apropos in the analysis since the simulation condition is necessary to validate this axiom. The data can be recorded using

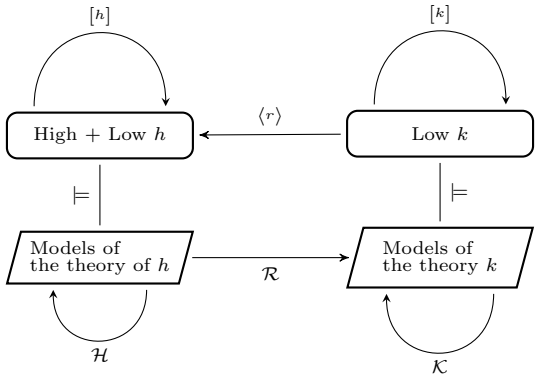


Figure 9: Non-interference Simulation

6 Stochastic Relations

The relation $\mathcal{R}$ as a set-valued map returns a set of all the points y to which an x is related. If instead there is some ambiguity about whether y is in $\mathcal{R}x$, then the notion of map must be relaxed much in the same way as that presented with neighborhood maps. The prescription is then

$$\mathcal{R}(x)(Q) \in [0,1]$$

where Q is a neighborhood at k and $[0,1]$ is the continuum from 0 to 1.

From [**?**], the following, with nomenclature changes to match this paper, defines stochastic relations. The models are now promoted to measurable spaces. In particular, the clopen sets of the Stone topology are now measurable and that topology is promoted to a σ-algebra. $\mathcal{H} = (H, \mathbb{H})$ and $\mathcal{K} = (K, \mathbb{K})$ are now measurable spaces and $\mathbb{H}$ and $\mathbb{K}$ are promoted to measurable relations as below:

Definition 6.0.1 *A* stochastic relation $\mathcal{R} : \mathcal{H} \twoheadrightarrow \mathcal{K}$ *is a measurable map* $\mathcal{H} \twoheadrightarrow \mathcal{G}(\mathcal{K})$ *where* $\mathcal{G}(\mathcal{K})$ *is the collection of subprobability measures and the initial* σ*-algebra on the subprobability measures.*

The following proposition from [**?**] characterizes stochastic relations:

Proposition 6.0.2 *Given measurable spaces* $\mathcal{H}$ *and* $\mathcal{K}$*, the following are equivalent:*

(i) $\mathcal{R} : \mathcal{H} \twoheadrightarrow \mathcal{K}$ *is a stochastic relation.*

(ii) $\mathcal{R}(x)$ *is a subprobability measure on* $\mathcal{K}$ *for each* $x \in H$ *such that the map* $x \mapsto \mathcal{R}(m)(Q)$ *is* $\mathcal{H}$ *measurable for each measurable set* $Q \in \mathbb{K}$*.*

Stochastic relations also compose, we refer to [**?**] for that. Stochastic relations give the means for measuring the possibility operator $\langle r \rangle$ as

$$(\mu \langle r \rangle)(Q) = \int_{x \in H} \mathcal{R}(x, Q) d\mu(x).$$

The point is that distributed logics lend themselves to be measured. This is important because many security properties are second-order properties but may be apprehended with modal logics. As is well-known, modal logics are capable of expressing some second-order properties. In particular, non-interference is a second-order property. By measuring the system over which non-interference is desired, the goal is to measure the security via non-interference.

7 Conclusions and Future Work

Distributed logic is best viewed as a logical toolbox for integrating many different logics which are themselves configured by axioms. One specifies the "connectivity" of local logics as a graph structure and then configures these "connections" with axioms and rules based upon a particular application. Many of the common modal axioms can be altered to fit distributed modal connectives. The simulation axiom shows this. As a further example, consider the Euclidean axiom (in a normal modal logic) $\langle h \rangle P \supset [h] \langle h \rangle P$ and its validating condition $\mathcal{H}xy$ and $\mathcal{H}xz$ implies $\mathcal{H}yz$. In distributed form for $r : h \curvearrowright k$ in Figure 10, this becomes $\langle r \rangle P \supset [r] \langle k \rangle P$ and the condition becomes ($\mathcal{R}xy$ and $\mathcal{R}xz$) implies $\mathcal{K}yz$.

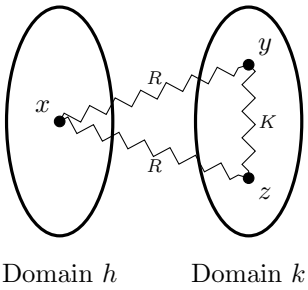


Figure 10: Distributed Euclidean Axiom

Figure 10 represents a common situation: the relation $\mathcal{R}$ between domain h and k is an artifact of the model and as such, deserves to be represented in a logic over the model. This is the sense in which distributed logic could be considered a model theoretic logic [**?**]. One must make choices up front before parts of the toolbox come together for a logic; the choices are made because models of a particular kind are needed for an application.

More philosophically speaking, modal logics come with a model theory which includes morphisms between models. The logic is abstracted over the model theory giving valid axioms and rules for reasoning about the models. Since morphisms are used in the model theory to describe critical aspects of the model, the obvious question is why these aspects are not formalized in the logics? The work in this paper (and its predecessor [**?**]) represents the first steps in this direction.

Part of the challenge of including morphisms in a logic is deciding which morphisms to include and how the included morphisms should be structured. Category theory presents us with the theory of morphisms, and considering modal logic, one could have started with p-morphisms. The approach we have taken is to generalize the notion of what should be considered a model theoretic morphism and then use logical axioms to give the morphisms the properties desired. In effect, we are choosing logical morphisms that preserve only some desired structure (but not all structure). The axiom system is then used as an array of control switches to configure distributed logics. In addition, the morphisms can be fine-tuned between some local logics but not imposed between all local logics within a distributed logic. This accords well with the notion that

distributed logics should be useful for representing reasoning about distributed computing systems where there is much variation and nuance that must be represented formally.

Space prevents us from also covering two-place intensional connectives such as entailment in relevance logic. That, too, has a pleasant reconstruction in distributed logic, although the three place relations require an extended notion of categorical morphism. Distributed logic was originally formulated with relations and consideration of testing for externally defined components in system-on-a-chip designs required the use of neighborhood systems. The ease of modification of distributed logic forced by two place intensional connectives and weak modal connectives requiring a neighborhood semantics is part of a larger theme for distributed logic: many model theoretic notions are "orthogonal" to distribution in that they do not seem to cause any significant hurdles to their re-expression in a distributed logic. Some model theoretic notions, such as morphism, are inherently distributed. Some, such as, Kripke relations, can be re-expressed as distributed notions. The bounds of what is possible seems to be related to the question of what is modality.

A good source of applications which require distributed reasoning are the security guarantees necessary for system-on-a-chip (SoC) architectures. In on-going and future work, we are expanding the use of distributed logics to provide a programming logic for a hardware specification language called ReWire [**?**]. Formal logic for SoCs almost demands a distributed logic. The sub-components are scattered across the chip and each is a small universe of internal states or worlds. One sub-component's connections with other sub-components can be either tight or very loose. Distributed Kripke relations provide the right kind of flexibility in this environment for interpreting logical properties of the SoC.

Most systems in the engineering world have some kind of distribution, whether it is concretely in space or abstractly as a mathematical distribution of modeling conditions. As is typical of the real world, many properties are not binary (i.e., that either the system or component has the property or it does not) but, rather, only admit to a probability of holding. Much of the goal of engineering is to have systems perform up a certain tolerance. Chip companies recognize this and have produced designs with error correction circuitry although, even then, they realize that they cannot achieve perfection. The advantage distributed logic holds, given the structural similarity with stochastic relations and Markov modeling, is that the these mathematical modeling techniques can now be seen as a direct weakening of logical properties.