

# A Theory of Distributed Systems

Norman R. Howes  
Institute for Defense Analyses  
[howes@ida.org](mailto:howes@ida.org)

## Abstract

The theory  $\theta$  presented here is the smallest theory in the temporal logic TLB [10] that all distributed systems, according to our definition of a distributed system, must satisfy.  $\theta$  is an instance of the classical modal logic S4.2. The central theorems of  $\theta$  are stated here without proof. Proofs will appear in [10]. Logics like TLA [14] and TLRCs [18] are used for specifying computer programs and reasoning about their behaviors. Their usefulness for large distributed systems has yet to be proven. Systems with multi-thousand node networks exhibit inherently asynchronous concurrency. Logics like TLA and TLRCs only provide a sequential (interleaved) concurrent execution model. Hence TLA and the future-tense part of TLRCs should be instances of the modal logic S4.3.1 [11, p. 179], but we are unaware if this has been proven. Not having a fully asynchronous concurrent execution model makes proofs about distributed systems within these logics suspect. Like quantum systems, distributed systems have pairs of observables that are not simultaneously measurable which leads to inherent uncertainty in their behaviors.  $\theta$  has an asynchronous concurrent execution model and accounts for this inherent uncertainty. Unlike S4.2, TLB has a second primitive modal operator called the “everywhere sometime” operator that mixes space and time. The reduction formulas of  $\theta$  allow us to reduce distributed correctness proofs to finitely many program proofs running on single computers. For distributed systems with the peer process architecture,  $\theta$  allow us to reduce distributed correctness proofs to a single program proof running on a single computer.

## 1. Introduction

For distributed systems, the precedence relationships between state transitions at different nodes cannot, in general, be known, i.e. we cannot always know if a state change at node  $A$  occurs before or after a state change at node  $B$ . We can still envision an instantaneous *global state*, we just cannot know what it is at any instant. All we can know at any instant is the *local view* of the global state that we can observe from a *local node*. The way a local node gets information about the states of other nodes is by receiving messages from them describing their states at certain instants. Many temporal logics model the execution of a program as a sequence of states (called a *behavior*) the program goes through as it executes. This

approach for modeling the executional behavior of a program has proven useful for specifying and proving theorems about programs. But sequences of states are not an appropriate executional model for a system where we cannot, in general, know the order that state transitions occur at two or more nodes. While we can design distributed systems to run synchronously, it is unnatural and inefficient. Even partially synchronous systems, such as ones with request-reply messaging semantics, such as client-server systems, tend to be a couple of orders of magnitude slower than ones with asynchronous messaging semantics [4]. Designers of temporal logics with interleaved execution models are often aware of this paradox.

"TLA is based on an interleaving model of concurrency, in which we assume that the execution of the system consists of a sequence of atomic events. It seems paradoxical to represent concurrent systems with a formalism in which events are never concurrent. We will not attempt to justify the philosophical correctness of interleaving models for reasoning about concurrent algorithms. Instead, we have tried to demonstrate the best reason we know for using TLA: it is a practical formalism for specifying and verifying safety and liveness properties." [14]

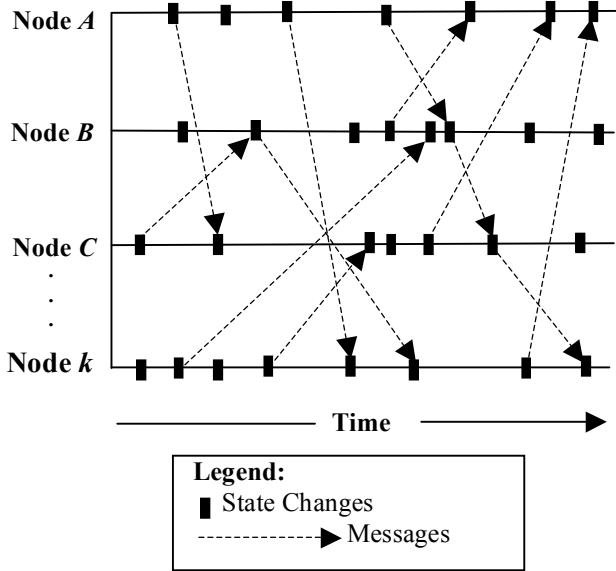
The distributed systems we consider are those whose execution model can be represented by a *net* as depicted in Figure 1 and defined in what follows. The example in Figure 1 has  $k$  nodes whose purpose is to keep each node informed of the *liveness* of each other node. It does this by having each node periodically send a *heartbeat* message to each other node. The horizontal lines in Figure 1 depict the *timelines* for the  $k$  nodes, and the small boxes on the timelines depict when and where state changes occur. If a box on one timeline appears to be to the left of a box on another timeline, this does not signify it precedes the state change indicated by the other box because, it is not always possible to know which state change occurs first. The state transition boxes on the various timelines in Figure 1 are not uniformly spaced to indicate that programs at different nodes may not progress at the same rate due to different processor speeds or different algorithms.

In certain cases we can know if a state change at one node precedes a state change at another node. For instance, consider the case in Figure 1 where Node  $C$  sends a heartbeat message to Node  $B$ . In this case, we can infer that the state change associated with the sending of

the message by  $C$  precedes the state change associated with the reception of the message by  $B$ . Since we can know when some state changes precede other state changes on different nodes, we can *partially order* the state changes of a distributed system by *precedence*. But we cannot, in general, *well order* these state changes by precedence as we can for sequential behaviors of programs that execute on a single node.

Not all state changes are associated with sending or receiving messages in Figure 1. These other state changes may be associated with other operations e.g. formatting heartbeat messages prior to their sending.

**Figure 1:** Distributed Heartbeat System  $\mathcal{D}$



Let  $\mathfrak{N}$  represent the set of nodes on which  $\mathcal{D}$  runs. For each  $A \in \mathfrak{N}$  let  $P^A$  denote the program executing on node  $A$ . Each  $P^A$  produces a sequence  $\sigma^A = \{s_n^A\}$  of states as it executes called a *sequential behavior* of  $P^A$ . The first element (state)  $s_0^A$  of  $\sigma^A$  is called the *initial state* of  $\sigma^A$ . We have not yet defined what we mean by a *state* of a program or distributed system, or of a *state space*. We do that shortly. Each behavior  $\sigma^A$  is a mapping  $\sigma^A: \mathbf{N}_0 \rightarrow \mathbf{St}^A$  from the non-negative integers  $\mathbf{N}_0$  into the state space  $\mathbf{St}^A$  of the program  $P^A$  such that  $\sigma^A(n) = s_n^A$ . Let  $\mathbf{D}_{\mathfrak{N}}$  be the Cartesian product  $\mathfrak{N} \times \mathbf{N}_0$  of the sets  $\mathfrak{N}$  and  $\mathbf{N}_0$ . We define the *distributed behavior*  $\beta$  of  $\mathcal{D}$  corresponding to the program behaviors  $\sigma^A$  to be the mapping  $\beta: \mathbf{D}_{\mathfrak{N}} \rightarrow \Sigma\{\mathbf{St}^A \mid A \in \mathfrak{N}\}$  defined by  $\beta(A, n) = \sigma^A(n) \in \mathbf{St}^A$  where  $\Sigma\{\mathbf{St}^A \mid A \in \mathfrak{N}\}$  is the disjoint (topological) sum [5, p. xxii] of the spaces  $\mathbf{St}^A$ . A behavior  $\beta$  can be thought of as “modeling” an individual execution of  $\mathcal{D}$ .

The domain  $\mathbf{N}_0$  of all the behaviors  $\sigma^A$  is ordered by the natural ordering  $\leq$  on the non-negative integers that is known to be a well ordering.  $\mathbf{D}_{\mathfrak{N}}$  is not well ordered but it can be ordered by a *directed ordering*. A pair  $(D, \leq)$ , where  $D$  is a set and  $\leq$  is a relation on  $D$ , is said to be a *directed set* if  $\leq$  is *reflexive* ( $d \in D$  implies  $d \leq d$ ),

*transitive* ( $d \leq a$  and  $a \leq b$  implies  $d \leq b$ ) and for each pair  $a, b \in D$  there is a  $d \in D$  with  $a \leq d$  and  $b \leq d$ . We define the *directed ordering*  $\leq_{\beta}$  on  $\mathbf{D}_{\mathfrak{N}}$  for a specific execution  $\beta$  of  $\mathcal{D}$  is as follows:

- (1.1) For each  $A \in \mathfrak{N}$  put  $(A, m) \leq_{\beta} (A, n)$  in  $\mathbf{D}_{\mathfrak{N}}$  if  $m \leq n$ .
- (1.2) For  $A, B \in \mathfrak{N}$  put  $(A, m) \leq_{\beta} (B, n)$  if the state change  $s_m^A$  in  $\sigma^A$  causes a message to be sent to  $B$  and if  $s_n^B$  in  $\sigma^B$  represents  $B$  receiving it.
- (1.3) For  $(A, m), (B, n), (C, k) \in \mathbf{D}_{\mathfrak{N}}$  with  $(A, m) \leq_{\beta} (B, n)$  and  $(B, n) \leq_{\beta} (C, k)$  put  $(A, m) \leq_{\beta} (C, k)$ .
- (1.4) For each  $(A, m), (B, n) \in \mathbf{D}_{\mathfrak{N}}$  there is a  $(C, k) \in \mathbf{D}_{\mathfrak{N}}$  with  $(A, m) \leq_{\beta} (C, k)$  and  $(B, n) \leq_{\beta} (C, k)$ .

Condition (1.4) holds for the system of Figure 1 since each  $P^A$  periodically sends *heartbeat* messages to each other node. We assume all systems we consider have a heartbeat function so  $\mathbf{D}_{\mathfrak{N}}$  will always be directed. This is a reasonable assumption since the period between heartbeat messages can be as long as we like, so the overhead generated by heartbeat messages can be as small as we like. A mapping  $\beta$  (e.g. a behavior) from a directed set into a space  $\mathcal{S}$  (e.g.  $\Sigma\{\mathbf{St}^A \mid A \in \mathfrak{N}\}$ ) is called a *net*. A net is a generalization of a sequence. The theory of nets has an important history in mathematics [5].

Temporal logics are special types of *modal logic*. The history of modal logic dates to classical times. Modal logic as we know it today was formalized by Clarence Lewis [16] [17]. Modal logics are similar to the logic PC (Propositional Calculus) but in addition to the logical operators of PC, like  $\neg$  and  $\wedge$ , they have at least one *modal operator*. Unlike logical operators, the modal operators cannot be defined by simple truth tables. Instead they are defined by axioms that describe their operational behavior. In TLB the modal operators are defined within set theory and their operational behaviors are proven as theorems. TLB has two primitive modal operators  $\square$  and  $\blacklozenge$ . Both have *dual operators* denoted  $\blacklozenge$  and  $\square$  respectively.  $\square$  is called the *always operator*,  $\blacklozenge$  is the *sometime operator*,  $\blacklozenge$  is the *everywhere sometime operator* and  $\square$  is the *somewhere always operator*.

Since Lewis' time, modal operators have been interpreted many ways. There are over 50 interpretations that have received significant attention [11]. In TLB  $\square$  is interpreted as “ $p$  is true now and at each subsequent instant.” This interpretation has an important history that begins with Arthur Prior's influential book *Time and Modality* [19]. Prior conjectured Lewis' modal logic S4 was the appropriate logic to interpret  $\square$  with respect to time in the sense that  $\square p$  means “ $p$  is true now and at each subsequent moment.” Soon after it was shown S4 is too weak to allow this interpretation. In 1959, Dummett [3] added an additional axiom called **D1** to the axioms of S4 to get the logic S4.3. In 1964 Bull [1] showed S4.3 together with an axiom called **N1** (that he called S4.3.1) was the right modal logic for Prior's interpretation of  $\square$ . In [2] he showed S4.3.1 is complete. Today it is understood that temporal logics with sequential models of time that interpret  $\square p$  in Prior's sense should be

instances of S4.3.1. Later we see that  $\theta$  restricted to a single node, is an instance of S4.3.1.

## 2. State spaces and temporal formulae

Let  $\mathbf{Var}^A$  be the set of all variables that characterize the state of an executing computer program  $P^A$  (or perhaps a finite collection of programs  $P^A, \dots, P^n$ ) on a single node  $A \in \mathfrak{N}$ . For each  $x \in \mathbf{Var}^A$  let  $Val_x^A$  be the set of values the variable  $x$  can assume. If multiple programs  $P^k$  use  $x$  we say  $x$  is a *shared* variable. If  $y$  is a variable in  $\mathbf{Var}^B$  it is assumed to be formally distinct from every variable in  $\mathbf{Var}^A$  but not necessarily distinct in name. In other words variables on distinct nodes are not shared. Programs can also have *input parameters* that do not change during execution so we do not consider them to be variables. Changing a parameter may change the behavior of the program, but variables change as a result of program execution.

Soon we will be interested in the topologies these sets  $Val_x^A$  may have. Let  $\mathbf{St}^A = \prod \{Val_x^A \mid x \in \mathbf{Var}^A\}$ , the product space (endowed with the product topology) [5, page xxiii] of the individual spaces  $Val_x^A$  for each  $x \in \mathbf{Var}^A$ .  $\mathbf{St}^A$  is called the (local) *state space* for the program(s) running on node  $A$ . A *state* in  $\mathbf{St}^A$  is a “vector”  $s = (s_x, s_y, s_z, \dots) \in \mathbf{St}^A$ . Notice that it may not be possible for  $P^A$  to get into some of these states during an execution. In fact, it may not be possible for  $P^A$  to get into some of these states during *any* execution.

The *system state space*  $\mathcal{S}$  of a distributed system  $\mathcal{D}$ , sometimes called the *global state space*, is the Cartesian product of the state spaces  $\mathbf{St}^A$ , equipped with the product topology. Hence  $\mathcal{S}$  is a “product of product spaces.” It is easy to see how we can identify  $\mathcal{S}$  with the simpler product space  $\prod \{Val_x^A \mid A \in \mathfrak{N} \text{ and } x \in \mathbf{Var}^A\}$ . We call  $p \in \mathcal{S}$  a *system state*. Such a  $p$  is a “vector”  $(s^A, s^B, s^C, \dots)$  of local states where  $A, B, C, \dots$  are the nodes in  $\mathfrak{N}$  and  $s^A \in \mathbf{St}^A$  for each  $A \in \mathfrak{N}$ . Note that a behavior  $\beta$  of  $\mathcal{D}$  is a mapping  $\beta: \mathbf{D}_{\mathfrak{N}} \rightarrow \Sigma$ , where  $\Sigma = \Sigma \{\mathbf{St}^A \mid A \in \mathfrak{N}\}$ , not a mapping of  $\beta$  into  $\mathcal{S}$ ! This is different than for local behaviors where each  $\sigma^A$  is a mapping into the local state space  $\mathbf{St}^A$ . We might like to define  $\beta$  to be a net in  $\mathcal{S}$ . That would require knowing the local states of the behaviors  $\sigma^A$  simultaneously which, in general, is impossible. Hence we must settle for defining  $\beta$  based on what is knowable, namely the local states of the behaviors  $\sigma^A$ .

The meanings of *state functions*, *state predicates* and *actions* (atomic operations) on states in TLB are equivalent to their meanings in TLA [14], only their definitions and notation is different in TLB. To save space we will not develop those concepts here. Those readers familiar with TLA will already understand these concepts.

*Temporal formulae* in TLB are assertions about behaviors built from *state formulae* (assertions about states or sets of states) together with the modal operators  $\square$ ,  $\diamond$ ,  $\blacklozenge$  and  $\blacksquare$ . State formulae are built from *elementary expressions*, logical operators (like  $\wedge$  and  $\neg$ ), and the set-theoretic primitive  $\in$ . Elementary expressions

can involve states, sets of states, state functions, actions and state predicates. Some examples are:  $\sigma^A(n)$ ,  $\beta(A, n)$ ,  $U$  and  $\pi_A(U)$ . The expression  $\pi_A(U)$  indicates the projection of a set  $U$  of global states into the local state space  $\mathbf{St}^A$ . For each  $A \in \mathfrak{N}$  the projection mapping [5, page xxiv]  $\pi_A: \mathcal{S} \rightarrow \mathbf{St}^A$  plays an important role. We will find that  $\pi_A$  commutes with the modal operators so we call  $\pi_A$  a *projection operator*. Projection operators play a role similar to the role projection operators play in quantum systems. It is more accurate to say they are analogous to the *Hermitian operators* of quantum systems in that they correspond to observables of a system.

From elementary expressions we can build state formulae like  $\sigma^A(n) \in \pi_A(U)$ ,  $\beta(A, n) \in \pi_A(V)$  or  $\beta \subset V$ . The first formula says that the state of the program  $P^A$  at the  $n^{\text{th}}$  instant is contained in the projection of the set  $U$  of global states into  $\mathbf{St}^A$ . The second formula says the state of  $\mathcal{D}$  at the distributed instant  $(A, n)$  is contained in the projection of the set  $V$  of global states into  $\mathbf{St}^A$ . The third formula says the entire behavior  $\beta$  is contained in the global state set  $V$  (despite the fact that  $\beta \subset \Sigma$ ). We explain this convention shortly.

From state formulae we create temporal formulae like  $\square(\sigma^A(n) \in \pi_A(U))$  or  $\diamond(\beta \subset V)$  by combining state formulae with modal operators. We have not yet defined what applying these operators to state formulae means. We do that now. If  $\mathcal{E}$  is a state formula containing a global state  $p$  or a set of global states  $V$ , by  $\pi_A(\mathcal{E})$ , for each  $A \in \mathfrak{N}$ , we mean the formula  $\mathcal{E}$  with every occurrence of  $p$  replaced by  $\pi_A(p)$  and every occurrence of  $V$  replaced by  $\pi_A(V)$ . We adopt the convention that a behavior  $\beta$  *satisfies* a state formula  $\mathcal{E}$  at an instant  $(A, n)$  if and only if  $\pi_A(\mathcal{E})$  is true at the state  $\sigma^A(n)$  of  $\mathbf{St}^A$ , i.e.

$$(2.1) \quad \beta(A, n)[\mathcal{E}] \equiv \sigma^A(n)[\pi_A(\mathcal{E})].$$

If  $\beta$  satisfies  $\mathcal{E}$  at each instant  $(A, n)$  we say  $\beta$  *satisfies*  $\mathcal{E}$  and write  $\beta[\mathcal{E}]$ , i.e.

$$(2.2) \quad \beta[\mathcal{E}] \equiv \forall(A, n) : \beta(A, n)[\mathcal{E}].$$

For state formulae  $\mathcal{E}_1$  and  $\mathcal{E}_2$  we define:

$$(2.3) \quad \beta(A, n)[\neg\mathcal{E}_1] \equiv \neg\beta(A, n)[\mathcal{E}_1]$$

$$(2.4) \quad \beta(A, n)[\mathcal{E}_1 \wedge \mathcal{E}_2] \equiv \beta(A, n)[\mathcal{E}_1] \wedge \beta(A, n)[\mathcal{E}_2].$$

Notice  $\beta[\neg\mathcal{E}_1] \Leftrightarrow \neg\beta[\mathcal{E}_1]$  may not hold since  $\neg\beta[\mathcal{E}_1]$  holds if there is a single  $(B, k)$  such that  $\beta(B, k)$  holds whereas  $\beta[\neg\mathcal{E}_1] \Leftrightarrow \forall(A, n) : \neg\beta(A, n)[\mathcal{E}_1]$ . But

$$(2.5) \quad \beta[\mathcal{E}_1 \wedge \mathcal{E}_2] \Leftrightarrow \beta[\mathcal{E}_1] \wedge \beta[\mathcal{E}_2].$$

TLB uses the *multiple worlds* approach [11] [15] to define temporal operators. *Distributed instants*  $(A, n) \in \mathbf{D}_{\mathfrak{N}}$  are considered to be “alternate worlds” (e.g. the world on planet  $A$  at time  $n$ ) that are related by the *visibility relation*  $\leq_{\beta}$  on  $\mathbf{D}_{\mathfrak{N}}$ . If  $(A, n), (B, k) \in \mathbf{D}_{\mathfrak{N}}$  with  $(A, n) \leq_{\beta} (B, k)$ , we say world  $(A, n)$  can “see” world  $(B, k)$ . We want the formula  $\square\mathcal{E}$ , read “always  $\mathcal{E}$ ,” to be true at  $(A, n)$  if  $\mathcal{E}$  is true in every world  $(B, k)$  it can see. We want  $\diamond\mathcal{E}$ , read “sometime  $\mathcal{E}$ ,” to be true at  $(A, n)$  if there is a  $(B, k)$  that  $(A, n)$  can see such that  $\mathcal{E}$  is true at  $(B, k)$ . Clearly each  $(A, n)$  can see itself. Hence the following definitions:

$$(2.6) \quad \beta(A, n)[\Box \mathcal{E}] \equiv \forall (B, k) \geq_{\beta} (A, n) : \beta(B, k)[\mathcal{E}]$$

$$(2.7) \quad \beta(A, n)[\Diamond \mathcal{E}] \equiv \exists (B, k) \geq_{\beta} (A, n) : \beta(B, k)[\mathcal{E}].$$

Hence

$$(2.8) \quad \beta(A, n)[\Box \mathcal{E}] \Leftrightarrow \beta(A, n)[\neg \Diamond \neg \mathcal{E}] \text{ and}$$

$$(2.9) \quad \beta(A, n)[\Diamond \mathcal{E}] \Leftrightarrow \beta(A, n)[\neg \Box \neg \mathcal{E}].$$

For a temporal formula  $\mathcal{F}$  containing  $\Box$  or  $\Diamond$  and a distributed instant  $(A, n)$  we define  $\beta(A, n)[\Box \mathcal{F}]$ ,  $\beta(A, n)[\Diamond \mathcal{F}]$  and  $\beta[\mathcal{F}]$  recursively as:

$$(2.10) \quad \beta(A, n)[\Box \mathcal{F}] \equiv \forall (B, k) \geq_{\beta} (A, n) : \beta(B, k)[\mathcal{F}]$$

$$(2.11) \quad \beta(A, n)[\Diamond \mathcal{F}] \equiv \exists (B, k) \geq_{\beta} (A, n) : \beta(B, k)[\mathcal{F}]$$

$$(2.12) \quad \beta[\mathcal{F}] \equiv \forall (A, n) : \beta(A, n)[\mathcal{F}].$$

These definitions lead to *iterated* temporal operators since  $\mathcal{F}$  may contain terms like  $\Box \Diamond \mathcal{E}$  or  $\Diamond \Box \mathcal{E}$  or  $\Box \Box \mathcal{E}$ . In such cases we cannot use (2.1) to evaluate  $\beta(A, n)[\mathcal{F}]$ . But we can do the following: if  $\mathcal{F} \equiv \Box \Diamond \mathcal{E}$  then

$$(2.13) \quad \beta(A, n)[\mathcal{F}] \Leftrightarrow \beta(A, n)[\Box \Diamond \mathcal{E}]$$

$$\Leftrightarrow \forall (B, k) \geq (A, n) : \beta(B, k)[\Diamond \mathcal{E}]$$

$$\Leftrightarrow \forall (B, k) \geq (A, n) : \exists (C, m) \geq (B, k) :$$

$$\sigma^C(m)[\pi_C(\mathcal{E})].$$

In other words, we must evaluate each modal operator in  $\mathcal{F}$  before we can apply (2.1). By (2.1) and (2.2)

$$(2.14) \quad \beta[\mathcal{C}V] \Leftrightarrow \forall (A, n) : \sigma^A(n)[\mathcal{C}\pi_A(V)].$$

But what does  $\sigma^A(n)[\mathcal{C}\pi_A(V)]$  mean?  $\sigma^A(n)$  is not a set but we can identify  $\sigma^A(n)$  with  $\{\sigma^A(n)\}$  and  $\{\sigma^A(n)\} \subset \pi_A(V)$  is meaningful. It is also equivalent to  $\sigma^A(n) \in \pi_A(V)$  and leads naturally to the following definitions:

$$(2.15) \quad \beta(A, n)[\mathcal{C}V] \equiv \beta(A, n)[\in V] \text{ and}$$

$$(2.16) \quad \beta[\mathcal{C}V] \equiv \beta[\Box \in V]$$

We leave it to the reader to show (2.14) and (2.16) are equivalent. If  $\beta[\mathcal{C}V]$  then to an observer at some node  $B$  it looks like  $\beta$  is a subset of  $V$  because each  $\beta(B, k)$  lies in  $\pi_B(V)$  and this is how an observer at node  $B$  sees  $\beta$  with respect to  $V$ . Next, for temporal formulae  $\mathcal{F}_1$  and  $\mathcal{F}_2$  we define

$$(2.17) \quad \beta(A, n)[\neg \mathcal{F}_1] \equiv \neg \beta(A, n)[\mathcal{F}_1]$$

$$(2.18) \quad \beta(A, n)[\mathcal{F}_1 \wedge \mathcal{F}_2] \equiv \beta(A, n)[\mathcal{F}_1] \wedge \beta(A, n)[\mathcal{F}_2].$$

It is easily shown that for any temporal formula  $\mathcal{F}$

$$(2.19) \quad \beta(A, n)[\neg \Box \neg \mathcal{F}] \Leftrightarrow \beta(A, n)[\Diamond \mathcal{F}]$$

$$(2.20) \quad \beta(A, n)[\neg \Diamond \neg \mathcal{F}] \Leftrightarrow \beta(A, n)[\Box \mathcal{F}]$$

$$(2.21) \quad \beta[\neg \Box \neg \mathcal{F}] \Leftrightarrow \beta[\Diamond \mathcal{F}]$$

$$(2.22) \quad \beta[\neg \Diamond \neg \mathcal{F}] \Leftrightarrow \beta[\Box \mathcal{F}] \text{ and}$$

$$(2.23) \quad \beta[\mathcal{F}] \Leftrightarrow \beta[\Box \mathcal{F}].$$

### 3. Convergence and clustering

Convergence and clustering of behaviors to specific states are fundamental concepts about systems, usually not considered in other temporal logics. The concepts of *liveness*, *fairness* and *safety* receive most attention in other temporal logics. These properties can be stated in

terms of clustering and convergence, but the concepts of convergence and clustering are more general and allow us to characterize global system properties, e.g. *global correctness*, *distributed fairness* and *liveness* and the *limit* of a distributed algorithm.

#### 3.1 Convergence and clustering of programs

Let  $P^A$  be a program executing on a node  $A$ . We can think of  $P^A$  as a *degenerate* distributed system running on a single node, in which case a behavior  $\beta$  of  $P^A$  reduces to a sequence  $\sigma^A$  since  $\mathbf{D}_n$  reduces to  $\mathbf{N}_0$ . For instance,  $\beta[\Box \mathcal{F}]$  is now  $\sigma^A[\Box \pi_A(\mathcal{F})]$ .

For each  $x \in \mathbf{Var}^A$  let  $\tau_x$  be the topology for  $Val_x^A$ . For instance if  $Val_x^A = \mathbf{R}$  (the real numbers), let  $\tau_x$  be the usual topology for  $\mathbf{R}$ . If for some  $y \in \mathbf{Var}^A$ ,  $Val_y^A$  is not a topological space or if we are not interested in the topology of  $Val_y^A$ , let  $\tau_y$  be the *indiscrete* topology, i.e. the topology whose only open sets are the empty set  $\emptyset$  and the set  $Val_y^A$  itself. Let  $\tau^A$ , the topology of  $\mathbf{St}^A$ , be the *product* topology of all the  $\tau_x$ . *Sub-basic open* sets of  $\tau^A$  are sets of the form

$$V_z = \{s \in \mathbf{St}^A \mid s_z \in V\}$$

where  $V$  is an open set in  $\tau_z$  for some  $z \in \mathbf{Var}^A$  and *basic open* sets are finite intersections of the sub-basic open sets. We now consider additional modal operators derived from  $\Box$  that correspond to the classical convergence theory (theory of limits) concepts of *frequently* and *eventually*. The *frequently* operator, denoted  $\nabla$ , and the *eventually* operator, denoted  $\Delta$ , are defined as:

$$(3.1) \quad \nabla \mathcal{F} \equiv \Box \Diamond \mathcal{F} \text{ and}$$

$$\Delta \mathcal{F} \equiv \Diamond \Box \mathcal{F}.$$

For a sequential behavior  $\sigma^A$  it is easily shown that

$$(3.2) \quad \sigma^A[\nabla \mathcal{F}] \Leftrightarrow \forall n : \exists m \geq n : \sigma^A(m)[\mathcal{F}] \text{ and}$$

$$\sigma^A[\Delta \mathcal{F}] \Leftrightarrow \exists n : \forall m \geq n : \sigma^A(m)[\mathcal{F}].$$

A *cofinal* subset of  $\mathbf{N}_0$  is a set of the form  $C = \{n \in \mathbf{N}_0 \mid \sigma^A(n)[\mathcal{F}]\}$  where  $\sigma^A[\nabla \mathcal{F}]$  and a set  $R$  of the form  $R = \{n \in \mathbf{N}_0 \mid \sigma^A(n)[\mathcal{F}]\}$  where  $\sigma^A[\Delta \mathcal{F}]$  is called *residual* in  $\mathbf{N}_0$ . A behavior  $\sigma^A$  *converges* to  $s^A \in \mathbf{St}^A$  if for each open  $U \subset \mathbf{St}^A$  containing  $s^A$ , there is a residual  $R \subset \mathbf{N}_0$  with  $\sigma^A(R) \subset U$ .  $\sigma^A$  *clusters* to  $s^A$  if for each open  $U \subset \mathbf{St}^A$  containing  $s^A$ , there is a cofinal  $C \subset \mathbf{N}_0$  with  $\sigma^A(C) \subset U$ . Hence the *convergence operator*  $\rightarrow$  and the *cluster operator*  $\rightsquigarrow$  are defined as follows:

$$(3.3) \quad \sigma^A[\rightarrow s^A] \equiv \forall U \in \tau^A(s^A) : \sigma^A[\Delta(\in U)] \text{ and}$$

$$(3.4) \quad \sigma^A[\rightsquigarrow s^A] \equiv \forall U \in \tau^A(s^A) : \sigma^A[\nabla(\in U)]$$

where  $\tau^A(s^A)$  is the neighborhood base of  $\tau^A$  at  $s^A$ .<sup>”</sup>  $\sigma^A[\rightarrow s^A]$  is read “ $\sigma^A$  converges to  $s^A$ ,” and  $\sigma^A[\rightsquigarrow s^A]$  is read “ $\sigma^A$  clusters to  $s^A$ .”

Even though we can prove  $\sigma^A$  converges to a state  $s$ , we may not be able to *implement* a program  $P^A$  that generates a sequence  $\sigma^A$  of states that actually gets arbitrarily close to  $s$  because current programming environments cannot handle arbitrarily large or small values or execute infinitely many steps. Yet taking a limit

involves an infinite process of executing operations and computing arbitrarily large or small values. This issue is dealt with in [10].

### 3.2 Distributed convergence and clustering

Convergence of nets is a generalization of the theory of convergence of sequences. That behaviors are not nets in global state space as in the degenerate case of sequential behaviors makes defining convergence more complicated for them than in the classical theory of nets. We need a new theory called the *theory of convergence of distributed behaviors* [7]. The definition of distributed convergence is

(3.5) *A behavior  $\beta$  converges to a system state  $p$  if for each open set  $U$  containing  $p$ , there is a residual  $R \subset \mathbf{D}_{\mathfrak{N}}$  with  $\beta(A, n) \in \pi_A(U)$  for each  $(A, n) \in R$ .*

$R \subset \mathbf{D}_{\mathfrak{N}}$  is residual in  $\mathbf{D}_{\mathfrak{N}}$  if  $\forall(A, n) \in R, (B, k) \geq (A, n)$  implies  $(B, k) \in R$ .  $C \subset \mathbf{D}_{\mathfrak{N}}$  is cofinal in  $\mathbf{D}_{\mathfrak{N}}$  if  $\forall(A, n) \in \mathbf{D}_{\mathfrak{N}}$ , there is a  $(B, k)$  in  $C$  with  $(B, k) \geq (A, n)$ . From Section 2 we recall that  $\beta[\mathbf{C}U]$  if for each instant  $(A, m)$ ,  $\sigma^A(m) \in \mathbf{C}\pi_A(U)$ . (3.5) says that if  $\beta$  converges to  $p$  then it appears to an observer at any node  $A$  that  $\sigma^A$  converges to  $\pi_A(p)$ . The proof of Theorem (3.6) is given in [10].

(3.6) *A behavior  $\beta$  converges to a state  $p$ , if and only if each local behavior  $\sigma^A$  converges to  $\pi_A(p)$ .*

From Theorem (3.6) we see how to define the *distributed convergence operator*  $\rightarrow$  as:

$$(3.7) \quad \beta[\rightarrow p] \equiv \forall A \in \mathfrak{N} : \sigma^A[\rightarrow \pi_A(p)].$$

Distributed clustering is more complex than clustering of ordinary nets. In the classical theory of nets we replace the residual set  $R$  in the definition of convergence with a cofinal set  $C$  to get the definition of clustering. If we did this for a behavior  $\beta$ , then if a local behavior  $\sigma^A$  clustered to  $\pi_A(p)$ ,  $\beta$  would cluster to  $p$  regardless of what the other  $\sigma^B$  do. In fact,  $\beta$  would cluster to every distributed state  $q$  in  $\mathcal{S}$  whose  $A^{\text{th}}$  coordinate is  $\pi_A(p)$ . This does not agree with our intuition. The problem is that any cofinal subset of a  $\sigma^A$  is also cofinal in  $\mathbf{D}_{\mathfrak{N}}$ . We get around this problem by strengthening the definition of clustering by using *fully cofinal* subsets of  $\mathbf{D}_{\mathfrak{N}}$ .  $C \subset \mathbf{D}_{\mathfrak{N}}$  is fully cofinal in  $\mathbf{D}_{\mathfrak{N}}$  if for each  $(A, n) \in \mathbf{D}_{\mathfrak{N}}$  and each  $B \in \mathfrak{N}$  there is a  $(B, k) \in C$  with  $(A, n) \leq (B, k)$ . The definition of clustering becomes:

(3.8) *A behavior  $\beta$  clusters to a state  $p$  if for each open  $U$  containing  $p$ , there is a fully cofinal  $C \subset \mathbf{D}_{\mathfrak{N}}$  with  $\beta(A, n) \in \pi_A(U)$  for each  $(A, n) \in C$ .*

The following theorem is proven in [10].

(3.9) *A behavior  $\beta$  clusters to a state  $p$ , if and only if each local behavior  $\sigma^A$  clusters to  $\pi_A(p)$ .*

Hence we define the *distributed cluster operator* as:

$$(3.10) \quad \beta[\rightsquigarrow p] \equiv \forall A \in \mathfrak{N} : \sigma^A[\rightsquigarrow \pi_A(p)]$$

We now show how to define the distributed convergence and cluster operators for  $\beta$  within TLB without recourse

to the sequential behaviors  $\sigma^A$  that comprise  $\beta$ . First we need the *everywhere sometime* operator  $\blacklozenge$ . Modal logics with multiple primitive modal operators are called *multi-modal logics* [11]. It is the interplay between  $\square$  and  $\blacklozenge$  that allows us to define  $\rightarrow$  and  $\rightsquigarrow$  directly within TLB.

Unfortunately  $\square$  and  $\blacklozenge$  cannot express the idea of distributed clustering because (3.9) involves the concept of a *fully cofinal* set whereas the operator  $\nabla \equiv \square \blacklozenge \mathcal{F}$  only expresses the concept of a *cofinal* subset of  $\mathbf{D}_{\mathfrak{N}}$ . It is  $\blacklozenge$  that allows us to characterize fully cofinal subsets. Define

$$(3.11) \quad \beta(A, n)[\blacklozenge \mathcal{F}] \equiv \forall B \in \mathfrak{N} : \exists(B, k) \geq_{\beta}(A, n) : \beta(B, k)[\mathcal{F}]$$

$$(3.12) \quad \beta(A, n)[\square \mathcal{F}] \equiv \exists B \in \mathfrak{N} : \forall(B, k) \geq_{\beta}(A, n) : \beta(B, k)[\mathcal{F}]$$

where it is assumed  $\mathcal{F}$  does not contain the new operators  $\blacklozenge$  or  $\square$ . Next we define what  $\beta(A, n)$  satisfies formulae  $\mathcal{F}$  and  $\mathcal{G}$  and the logical operators  $\neg$  and  $\wedge$  means, where  $\mathcal{F}$  and  $\mathcal{G}$  may contain the new operators  $\blacklozenge$  and  $\square$ .

$$(3.13) \quad \beta(A, n)[\neg \square \mathcal{F}] \equiv \neg \beta(A, n)[\square \mathcal{F}]$$

$$(3.14) \quad \beta(A, n)[\neg \blacklozenge \mathcal{F}] \equiv \neg \beta(A, n)[\blacklozenge \mathcal{F}]$$

$$(3.15) \quad \beta(A, n)[\blacklozenge \mathcal{F} \wedge \square \mathcal{G}] \equiv \beta(A, n)[\blacklozenge \mathcal{F}] \wedge \beta(A, n)[\square \mathcal{G}]$$

$$(3.16) \quad \beta(A, n)[\square \mathcal{F} \wedge \blacklozenge \mathcal{G}] \equiv \beta(A, n)[\square \mathcal{F}] \wedge \beta(A, n)[\blacklozenge \mathcal{G}]$$

It is not difficult to show that:

$$(3.17) \quad \beta(A, n)[\neg \square \neg \mathcal{F}] \Leftrightarrow \beta(A, n)[\blacklozenge \mathcal{F}] \text{ and}$$

$$(3.18) \quad \beta(A, n)[\neg \blacklozenge \neg \mathcal{F}] \Leftrightarrow \beta(A, n)[\square \mathcal{F}].$$

For any formula  $\mathcal{F}$ , we define

$$(3.19) \quad \beta[\mathcal{F}] \equiv \forall(A, n) : \beta(A, n)[\mathcal{F}] \text{ and}$$

For a formula  $\mathcal{E}$  with no modal operators we define:

$$(3.20) \quad \beta(A, n)[\mathcal{E}] \equiv \sigma^A(n)[\pi_A(\mathcal{E})].$$

We define the *distributed frequently* operator  $\blacktriangledown$  and the *distributed eventually* operator  $\blacktriangle$  as:

$$(3.21) \quad \blacktriangledown \mathcal{F} \equiv \square \blacklozenge \mathcal{F} \text{ and}$$

$$(3.22) \quad \blacktriangle \mathcal{F} \equiv \blacklozenge \square \mathcal{F}.$$

In [10] it is proved that

$$(3.23) \quad \beta[\blacktriangledown \mathcal{F}] \Leftrightarrow \forall A \in \mathfrak{N} : \sigma^A[\blacktriangledown \pi_A(\mathcal{F})] \text{ and}$$

$$(3.24) \quad \beta[\blacktriangle \mathcal{F}] \Leftrightarrow \forall A \in \mathfrak{N} : \sigma^A[\blacktriangle \pi_A(\mathcal{F})].$$

Formulae like (3.23) and (3.24) are called *reduction formulae* because they allow us to reduce a formula like  $\beta[\blacktriangledown \mathcal{F}]$  to finitely many formulae about sequential behaviors of programs. In [10] it is also shown that:

$$(3.25) \quad \beta[\square \mathcal{F}] \Leftrightarrow \beta[\neg \blacklozenge \neg \mathcal{F}] \text{ and}$$

$$(3.26) \quad \beta[\blacklozenge \mathcal{F}] \Leftrightarrow \beta[\neg \square \neg \mathcal{F}].$$

Applying (3.23) and (3.24) to the state formula  $\in U$  gives:

$$(3.27) \quad \beta[\blacktriangle(\in U)] \Leftrightarrow \forall A \in \mathfrak{N} : \sigma^A[\blacktriangle(\in \pi_A(U))].$$

Let  $p \in \mathcal{S}$  and for each  $A \in \mathfrak{N}$  put  $p_A = \pi_A(p)$ . Also, for each  $A$  let  $\tau^A(p_A)$  denote a neighborhood base for  $p_A$  in the topology  $\tau^A$  of  $\mathbf{St}^A$ . From (3.7) and (3.3) we can derive:

$$(3.28) \beta[\rightarrow p] \Leftrightarrow \forall A \in \mathfrak{N} : \forall U_A \in \tau^A(p_A) : \sigma^A[\blacktriangle(\in U_A)].$$

Now  $U = \prod_{A \in \mathfrak{N}} U_A$  is a neighborhood of  $p$  in  $\mathcal{S}$ . Hence:

$$(3.29) \beta[\rightarrow p] \Leftrightarrow \forall U \in \mathcal{N}_p : \forall A \in \mathfrak{N} : \sigma^A[\blacktriangle(\in \pi_A(U))].$$

where  $\mathcal{N}_p$  represents a basis for the open sets at  $p \in \mathcal{S}$ . Then from (3.27) we can derive:

$$(3.30) \beta[\rightarrow p] \Leftrightarrow \forall U \in \mathcal{N}_p : \beta[\blacktriangle(\in U)].$$

that is analogous to (3.3) and a similar result:

$$(3.31) \beta[\rightsquigarrow p] \Leftrightarrow \forall U \in \mathcal{N}_p : \beta[\blacktriangledown(\in U)]$$

that is analogous to (3.4). This shows we could have developed the concepts of distributed convergence and clustering directly from the operators  $\blacksquare$  and  $\blacklozenge$  if we had the necessary intuition. Instead the intuition came from classical convergence theory as found in [5], and we worked from there to figure out how to define  $\blacktriangle$  and  $\blacktriangledown$ .

### 3.3 Uncertainty in distributed systems

We first consider the intuitive meaning of a behavior  $\beta$  being eventually or frequently in a subset  $V$  of  $\mathcal{S}$ . From (2.14) we get:

$$(3.32) \beta[\subset V] \Leftrightarrow \forall A \in \mathfrak{N} : \sigma^A[\subset \pi_A(V)].$$

Theorem (3.32) expresses the *uncertainty* in knowing if a behavior  $\beta$  of a system  $\mathcal{Q}$  produces *global states* that lie in  $V$ , by means of observing if the values of each  $\sigma^A$  lies in  $\pi_A(V)$  for each  $A \in \mathfrak{N}$ . The right side of (3.32) can be interpreted as the sequences  $\sigma^A$  being subsets of the projections  $\pi_A(V)$  since the range of  $\sigma^A$  is a subset of  $\mathbf{St}^A$ . The left side of (3.32) cannot be interpreted as  $\beta$  actually being a subset of  $V$  since the range of  $\beta$  lies in  $\Sigma$ , not  $\mathcal{S}$ . So how does (3.32) say anything about the global states produced by  $\mathcal{Q}$  during an execution for which  $\beta$  is the model? This is the question we now investigate.

The concept of a distributed system having global states during an execution is not incompatible with the fact that we cannot know all components of the global state simultaneously. A similar situation occurs with quantum systems. We can, in certain interpretations of quantum logic, consider the system to have a global state even though there are observables of the system we cannot measure simultaneously. So not being able to know all the components of the global state at the same instant does not mean it does not exist, at least in these interpretations

What can be known about a distributed system are the *local states* that can be observed at each node. We can imagine an experiment where the clocks of the computers in a distributed system  $\mathcal{Q}$  are synchronized to the best accuracy attainable consistent with the capabilities of the processing and networking resources available for  $\mathcal{Q}$  to execute on. Then a time interval  $T$  is selected and at a predetermined time the following experiment is begun. At the end of each successive interval  $T$  each program  $P^A$  at node  $A$  takes a “snapshot” of its state and sends it to each other node where it is logged and time stamped. We assume  $A$  sends a copy of its snapshots to itself so they

will also appear in  $A$ 's log file. After some predetermined time period the experiment is completed and the log files are all sent to a central location where they are correlated as follows.

For each node  $A$ , a snapshot of the state of  $P^A$  should appear only once in each time period  $T$  in  $A$ 's log file, but multiple state snapshots may exist for processes  $P^B$  at other nodes  $B$  in a given time period due to messaging delays. For each node  $A$  the latest state snapshot from each other node  $B$  will be retained as the “state at node  $B$  during that time period. If no state snapshot is received for a node  $B$  during a time interval, then the last state snapshot received from node  $B$  for a previous time period will be considered to be the state at node  $B$  during that time period. After this correlation, a *node  $A$  system state view* can be constructed for each node  $A$  by concatenating the state snapshots for each of the nodes  $B \in \mathfrak{N}$  logged at node  $A$  during each time period  $T$ .

These *local views of the (global) system state history* can now be compared. None of them are likely to be the “real” system state history and any pair of them are likely to be different. Yet each one is an approximation of the real system state history. How good are these histories? For most distributed systems they probably are not very good. There is a way to build a much better history, but not one that can be viewed locally in “real-time” as the *local view approximations* just discussed. It is this better approximation that the right side of (3.32) relates to.

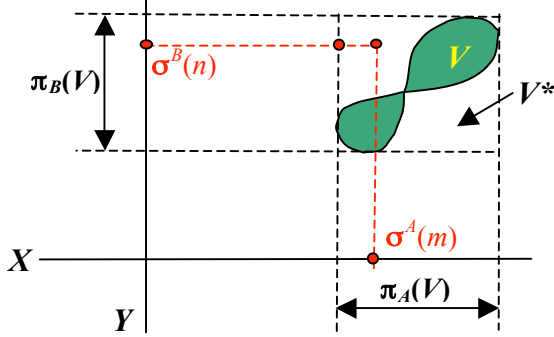
The only way node  $A$ 's local view history could be as good as the history we now explain is if all the local state snapshots from other nodes are sent to  $A$  infinitely fast (in zero time). This *best* history method omits sending the snapshots. Instead, each node logs its own state at the end of each time period  $T$  and at the end of the experiment sends its log file to a central location where a single global system state history is constructed. This history differs from the real global state history only by the amounts of time it takes to monitor and log the value of each local state variable at each node at the end of each period  $T$  and the differences in time at each node due to imperfect clock synchronization. The right hand side of (3.32) refers to observing if  $\sigma^A[\subset \pi_A(V)]$  holds at each  $A$  in this experiment.

Figure 3-1 gives an example of this “best quality” observation for a distributed system  $\mathcal{Q}$  consisting of two programs  $P^A$  and  $P^B$  running on nodes  $A$  and  $B$ . It illustrates a *measure*  $\Delta V$  of the uncertainty of  $\beta$ 's global states being contained in a  $V \subset \mathcal{S}$  given each  $\sigma^A$  is observed to be contained in  $\pi_A(V)$ .  $P^A$  and  $P^B$  have variables  $x$  and  $y$  respectively that both represent real numbers. Let  $X$  and  $Y$  be copies of the real numbers  $\mathbf{R}$  over which  $x$  and  $y$  range. Then  $\mathcal{S} = X \times Y$ .

The meaning of  $\beta$  satisfying  $\subset V$  in (3.32) is that both  $\sigma^A$  and  $\sigma^B$  are subsets of  $\pi_A(V)$  and  $\pi_B(V)$  respectively. The states  $\sigma^A(m)$  and  $\sigma^B(n)$  in Figure 3-1 are observed at instants  $(A, m)$  and  $(B, n)$  respectively. They define a global state  $p = (\sigma^A(m), \sigma^B(n))$  in  $\mathcal{S}$ . In Figure 3-1  $p$  does not belong to  $V$ . Since  $\pi_A(p) = \sigma^A(m)$  and  $\pi_B(p) = \sigma^B(n)$ ,  $p$

$\in V^* = \pi_A^{-1}(\pi_A(V)) \times \pi_B^{-1}(\pi_B(V))$ . Hence  $p$  is “close” to  $V$  in the sense  $V^*$  is the “smallest” set in  $\mathcal{S}$  of the form  $Z = \pi_A^{-1}(\pi_A(Z)) \times \pi_B^{-1}(\pi_B(Z))$  containing  $V$ .

**Figure 3 – 1:** Distributed uncertainty example



In this example,  $\mathcal{S}$  is a metric space and the distance  $d(p, Cl(V))$  from  $p$  to the closure  $Cl(V)$  of  $V$  is defined and finite. We can use the distance  $d(p, Cl(V))$  to define the *uncertainty* that system states  $p$  produced by an execution  $\beta$  of  $\mathcal{Q}$  are in  $V$  given  $\sigma^A[\llbracket \pi_A(V) \rrbracket]$  and  $\sigma^B[\llbracket \pi_B(V) \rrbracket]$  as follows. Put

$$(3.33) \quad \Delta V = \max\{d(p, Cl(V)) \mid p \in V^*\}.$$

Note that if  $V = V^*$  then  $V^* \setminus V = \emptyset$  and  $\Delta V = 0$ , i.e. there is no uncertainty. Otherwise  $\Delta V > 0$ . If  $\mathcal{S}$  is not a metric space, but  $\mathcal{S}$  is a *measurable space* [Howes 95, p. 243] with measure  $\mu$  we define  $\Delta V$  as

$$(3.34) \quad \Delta V = \mu(V^* \setminus V).$$

Again if  $V = V^*$  then  $V^* \setminus V = \emptyset$  and  $\Delta V = 0$ . Also  $\Delta V \geq 0$ . Even if  $\mathcal{S}$  is not a metric space or measurable space we can consider  $V^* \setminus V$  to be a measure of uncertainty and in such cases we refer to  $V^* \setminus V$  as the *uncertainty set*.

Since open sets in  $\mathcal{S}$  of the form  $U = \prod_{A \in \mathfrak{N}} \pi_A^{-1}(U_A)$  where each  $U_A$  is open in  $\mathbf{St}^A$  form a neighborhood base for  $\mathcal{S}$  we can use them to shrink down on states  $p \in \mathcal{S}$  when considering a limit of  $\beta$  at  $p$ . This can be seen from the definition of the product topology [Howes 95, p. xxiv] and the fact  $\mathcal{S}$  is a finite product space. In fact, there is a basis  $\tau_{\mathcal{S}}$  for the topology of  $\mathcal{S}$  consisting of opens sets of this form. So considering  $\beta$  to eventually be in every neighborhood of  $p$  is reasonable if  $p$  is a limit of  $\beta$  (even though  $\beta$  lies in  $\Sigma$  rather than  $\mathcal{S}$ ) since a neighborhood  $N$  of  $p$  contains an open  $U \subset \mathcal{S}$  with  $p \in U \subset U^* \subset N$ , which is essentially what (3.30) says.

To conclude this discussion of uncertainty we notice that uncertainty in knowing if global states produced by a distributed system are in specific subsets of state space is expressed in terms of observables and projection operators analogously to the way uncertainty is expressed in terms of observables and *Hermitian* operators in quantum mechanics. Moreover, the phenomenon that causes uncertainty about quantum states is the same one that causes uncertainty about global system states, namely these systems (quantum and distributed) admit pairs of observables that cannot be measured simultaneously.

## 4. Soundness, completeness and consistency

In [11, p. 36], we find the following remark:

“We shall in fact come across many cases in which we have an axiomatic modal system defined without any reference to an account of validity, and a definition of validity formulated without any reference to the theoremhood in a system, and yet the theorems of that system are precisely the well formed formulas which are valid by that definition; but this is something which has to be proved in every case, . . . we have to prove two things: (A) that every theorem of the system is valid by that definition, and (B) that every well formed formula valid by that definition is a theorem of the system. If (A) holds we say the system is *sound*, and if (B) holds, we say that it is *complete*, in each case with respect to the validity-definition in question.”

### 4.1 Validity of TLB formulae

The concept of validity in TLB is based on Kripkean semantics [12] [13] as interpreted by [3]. Validity of a formula  $\mathcal{F}$  with respect to a behavior  $\beta$ , denoted  $\models_{\beta} \mathcal{F}$ , is first defined. Then validity of  $\mathcal{F}$  with respect to a system  $\mathcal{Q}$ , denoted  $\models_{\mathcal{Q}} \mathcal{F}$ , is defined. Each behavior  $\beta$  gives rise to a *validity model*. Validity models involve the concept of a *frame*. A frame is an ordered pair  $\langle W, R \rangle$  where  $W$  is a set and  $R$  is a relation in  $W \times W$ .  $W$  is called the set of *possible worlds* for the frame and  $R$  is a *visibility* relation among worlds that tells which worlds determine the truth of formulae of the type  $\Box \mathcal{F}$  or  $\Diamond \mathcal{F}$  in a given world  $w$ . An important difference between TLB and temporal logics with sequential execution models is that each execution  $\beta$  has a distinct frame  $\langle W, R_{\beta} \rangle$  because each  $\beta$  induces a distinct directed ordering  $R_{\beta}$  on  $\mathbf{D}_{\mathfrak{N}}$ . The *validity model generated by*  $\beta$  is the ordered triple  $M_{\beta} = \langle W, R_{\beta}, T_{\beta} \rangle$  where  $W = \mathbf{D}_{\mathfrak{N}}$ ,  $R_{\beta}$  is  $\leq_{\beta}$  on  $\mathbf{D}_{\mathfrak{N}}$  and  $T_{\beta}$  is a *truth-valued* function from  $\mathfrak{F} \times W$  onto  $\{0, 1\}$  that tells which formulae, in the set  $\mathfrak{F}$  of all TLB formulae, are true at each world  $w \in W$  in the model  $M_{\beta}$ . The function  $T_{\beta}$  is defined recursively as:

- (4.1)  $T_{\beta}(\neg \mathcal{F}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w) = 0$  for each  $(\mathcal{F}, w) \in \mathfrak{F} \times W$ .
- (4.2)  $T_{\beta}(\mathcal{F} \vee \mathcal{G}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w) = 1$  or  $T_{\beta}(\mathcal{G}, w) = 1$  for  $\mathcal{F}, \mathcal{G} \in \mathfrak{F}$  and  $w \in W$ .
- (4.3)  $T_{\beta}(\mathcal{F} \wedge \mathcal{G}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w) = 1 = T_{\beta}(\mathcal{G}, w)$  for  $\mathcal{F}, \mathcal{G} \in \mathfrak{F}$ .
- (4.4)  $T_{\beta}(\mathcal{F} \Rightarrow \mathcal{G}, w) = 1$  iff  $T_{\beta}(\mathcal{G}, w) = 1$  or  $T_{\beta}(\mathcal{F}, w) = 0$  for  $\mathcal{F}, \mathcal{G} \in \mathfrak{F}$ .
- (4.5)  $T_{\beta}(\mathcal{F} \Leftrightarrow \mathcal{G}, w) = 1$  iff  $T_{\beta}(\mathcal{G}, w) = T_{\beta}(\mathcal{F}, w)$  for  $\mathcal{F}, \mathcal{G} \in \mathfrak{F}$ .
- (4.6)  $T_{\beta}(\Box \mathcal{F}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w') = 1$  for each  $w' \in W$  with  $wR_{\beta}w'$ .
- (4.7)  $T_{\beta}(\Diamond \mathcal{F}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w') = 1$  for some  $w'$  in  $W$  with  $wR_{\beta}w'$ .
- (4.8)  $T_{\beta}(\blacklozenge \mathcal{F}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w') = 1$  for some  $w' = (A, n)$  with  $wR_{\beta}w'$  for each  $A \in \mathfrak{N}$ .
- (4.9)  $T_{\beta}(\blackbox \mathcal{F}, w) = 1$  iff  $T_{\beta}(\mathcal{F}, w') = 1$  for each  $w' = (A, n)$  with  $wR_{\beta}w'$  for some  $A \in \mathfrak{N}$ .

A TLB formula  $\mathcal{F}$  is said to be *true in a model*  $M_\beta$ , denoted  $\models_\beta \mathcal{F}$ , if it is *true* at every world of the model, i.e. if  $T_\beta(\mathcal{F}, w) = 1$  for each  $w \in W$ .  $\mathcal{F}$  is said to be *valid* for  $\mathcal{Q}$ , denoted  $\models_{\mathcal{Q}} \mathcal{F}$ , if it is *true* in every model  $M_\beta$  of  $\mathcal{Q}$ . Finally we define  $\models \mathcal{F}$  to mean that  $\mathcal{F}$  is valid for every distributed system  $\mathcal{Q}$ .

Some formulae in logics are true by virtue of their syntactic form. Their truth value is independent of what other formulae are uniformly substituted for their variable (formula) symbols. We call such formulae *tautologies*. An example of a TLB tautology is the formula *schema*:  $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$ . By a formula schema we mean a collection of formulae that all have the same syntactic form like the set  $\{(A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C) \mid A, B, C \in \mathfrak{F}\}$ . It does not matter which formulae  $\mathcal{F}$ ,  $\mathcal{G}$  and  $\mathcal{H}$  we substitute for  $A$ ,  $B$  and  $C$  respectively,  $(\mathcal{F} \Rightarrow \mathcal{G}) \wedge (\mathcal{G} \Rightarrow \mathcal{H}) \Rightarrow (\mathcal{F} \Rightarrow \mathcal{H})$  will still be true. We define a tautology  $\mathcal{T}$  to be a formula (or formula schema) that is true for every truth-value function  $T_\beta$  on the *modal-atomic* formulae  $\mathfrak{M}$  of  $\mathfrak{F}$ . A formula is said to be modal-atomic if it contains no modal operators or if it is of the form  $\Box \mathcal{F}$ ,  $\Diamond \mathcal{F}$ ,  $\blacklozenge \mathcal{F}$  or  $\Box \mathcal{F}$  where the formula  $\mathcal{F}$  contains no modal operators.

Since  $T_\beta$  can be extended uniquely from  $\mathfrak{M}$  to  $\mathfrak{F}$ ,  $T_\beta(\mathcal{T}, w) = 1$  for each  $w \in W$  and each validity model  $M_\beta$  of  $\mathcal{Q}$ . Thus  $\models_{\mathcal{Q}} \mathcal{T}$  for each tautology  $\mathcal{T}$ . A formula  $\mathcal{F}$  is a *tautological consequence* of formulae  $\mathcal{F}_1 \dots \mathcal{F}_n$  if for each  $T_\beta$  with  $T_\beta(\mathcal{F}_i, w) = 1$  for each  $i = 1 \dots n$  and  $w \in W$ ,  $T_\beta(\mathcal{F}, w) = 1$ .

## 4.2 Provability and theories in TLB

A *theory* in TLB is a subset  $\Theta$  of the set  $\mathfrak{F}$ , of all TLB formulae, that contains the set  $\mathfrak{T}$  of all tautologies in  $\mathfrak{F}$  and is *closed* under the *inference rule* Modus Ponens (MP), i.e. if  $\mathcal{F} \in \Theta$  and  $\mathcal{F} \Rightarrow \mathcal{G} \in \Theta$  then  $\mathcal{G} \in \Theta$ . TLB also satisfies the inference rule *Necessitation* (N) which is necessary for TLB to be a normal modal logic which S4.2 is. In TLB N can be derived from MP which is not the case with all modal logics, so formally TLB's only inference rule is MP.

Clearly  $\mathfrak{T}$  and  $\mathfrak{F}$  are theories in TLB.  $\mathfrak{T}$  is the smallest theory and  $\mathfrak{F}$  is the largest. For each theory  $\Theta$  we have  $\mathfrak{T} \subset \Theta \subset \mathfrak{F}$ . If  $\{\Theta_\alpha \mid \alpha \in A\}$  is a collection of theories, then their intersection  $\bigcap \{\Theta_\alpha \mid \alpha \in A\}$  is also a theory. For a subset  $\mathfrak{G}$  of  $\mathfrak{F}$  there is a smallest theory containing  $\mathfrak{G}$ . It is the intersection of all TLB theories containing  $\mathfrak{G}$ .

Let  $\Theta$  be a theory and  $\mathcal{F} \in \mathfrak{F}$ . We say  $\mathcal{F}$  is *provable in*  $\Theta$  or  $\mathcal{F}$  is a *theorem* of  $\Theta$ , denoted  $\vdash_\Theta \mathcal{F}$ , if there are finitely many  $\mathcal{F}_1 \dots \mathcal{F}_n \in \Theta$  with  $(\mathcal{F}_1 \wedge \dots \wedge \mathcal{F}_n \Rightarrow \mathcal{F}) \in \Theta$ . If  $\mathfrak{G} \subset \mathfrak{F}$  we say  $\mathcal{F}$  is *provable* from  $\mathfrak{G}$  in  $\Theta$  and write  $\mathfrak{G} \vdash_\Theta \mathcal{F}$  if there are finitely many  $\mathcal{G}_1 \dots \mathcal{G}_n \in \mathfrak{G}$  with  $(\mathcal{G}_1 \wedge \dots \wedge \mathcal{G}_n \Rightarrow \mathcal{F}) \in \Theta$  or if  $\vdash_\Theta \mathcal{F}$ . So if  $\mathfrak{G} = \emptyset$  and  $\vdash_\Theta \mathcal{F}$  we have  $\mathfrak{G} \vdash_\Theta \mathcal{F}$ . Also, if  $\mathfrak{G} \neq \emptyset$  but there are no  $\mathcal{G}_1 \dots \mathcal{G}_n \in \mathfrak{G}$  with  $(\mathcal{G}_1 \wedge \dots \wedge \mathcal{G}_n \Rightarrow \mathcal{F}) \in \Theta$  and  $\vdash_\Theta \mathcal{F}$ , we still have  $\mathfrak{G} \vdash_\Theta \mathcal{F}$ . Clearly all members of  $\Theta$  are theorems in  $\Theta$  since  $\mathcal{F} \Rightarrow \mathcal{F}$  is a tautology for each  $\mathcal{F} \in \Theta$ . The question naturally arises: Are there theorems of  $\Theta$  not

contained in  $\Theta$ ? The answer is no. The following are proven in [10].

(4.10) *Theorem* If  $\vdash_\Theta \mathcal{F}$  then  $\mathcal{F} \in \Theta$ .

(4.11) *Corollary*  $\vdash_\Theta \mathcal{F}$  if and only if  $\mathcal{F} \in \Theta$ .

The method of proof for (4.10) can be used to show that tautological consequence of  $\mathcal{F}_1 \dots \mathcal{F}_n \in \Theta$  belong to  $\Theta$ . We are interested in theories about distributed systems  $\mathcal{Q}$  and in particular the theory of all distributed systems that we call  $\theta$ . For a behavior  $\beta$  of  $\mathcal{Q}$  let  $\Theta_\beta = \{\mathcal{F} \in \mathfrak{F} \mid \models_\beta \mathcal{F}\}$ . Clearly each tautology is in  $\Theta_\beta$ . If  $\models_\beta \mathcal{F}$  and  $\models_\beta (\mathcal{F} \Rightarrow \mathcal{G})$  then for each world  $w$ ,  $T_\beta(\mathcal{F}, w) = 1$  and  $T_\beta(\mathcal{F} \Rightarrow \mathcal{G}, w) = 1$ . By (4.4)  $T_\beta(\mathcal{G}, w) = 1$ . Hence  $\mathcal{G} \in \Theta_\beta$ . Thus  $\Theta_\beta$  is closed under MP, so  $\Theta_\beta$  is a theory.

For a distributed system  $\mathcal{Q}$  let  $\Theta_{\mathcal{Q}} = \{\mathcal{F} \in \mathfrak{F} \mid \models_{\mathcal{Q}} \mathcal{F}\}$ . To show  $\Theta_{\mathcal{Q}}$  is a theory notice that for each tautology  $\mathcal{T}$ ,  $\models_{\mathcal{Q}} \mathcal{T}$  for each behavior  $\beta$ , so  $\models_{\mathcal{Q}} \mathcal{T}$  and hence  $\mathcal{T} \in \Theta_{\mathcal{Q}}$ . If  $\mathcal{F}$  and  $\mathcal{F} \Rightarrow \mathcal{G} \in \Theta_{\mathcal{Q}}$  then  $\mathcal{F}$  and  $\mathcal{F} \Rightarrow \mathcal{G} \in \Theta_\beta$  for each  $\beta$  so  $\mathcal{G} \in \Theta_\beta$  for each  $\beta$ . Hence  $\models_{\mathcal{Q}} \mathcal{G}$  for each  $\beta$  which means  $\models_{\mathcal{Q}} \mathcal{G}$ . Thus  $\mathcal{G} \in \Theta_{\mathcal{Q}}$  so  $\Theta_{\mathcal{Q}}$  is closed under MP. Hence  $\Theta_{\mathcal{Q}}$  is a theory. Note that  $\Theta_{\mathcal{Q}}$  is the intersection of the theories  $\Theta_\beta$  for each behavior  $\beta$  of  $\mathcal{Q}$ . Finally there is the theory  $\theta = \{\mathcal{F} \in \mathfrak{F} \mid \models \mathcal{F}\}$  which is the intersection of all the theories  $\Theta_{\mathcal{Q}}$ .

By now the observant reader will have noticed we have shown soundness and completeness for the theories  $\Theta_\beta$ ,  $\Theta_{\mathcal{Q}}$  and  $\theta$ . This follows from the way they are defined. For example  $\theta$  consists of all valid formulae, but since it is a theory, all its formulae are also provable.

## 4.3 Consistency of $\theta$ and relation to S4.2

The results in this section are all proved in [10]. We say a theory  $\Theta$  is *consistent* if there is no formula  $\mathcal{H}$  in  $\Theta$  such that  $\vdash_\Theta \mathcal{H}$  and  $\vdash_\Theta \neg \mathcal{H}$ .

(4.12) *Theorem*  $\Theta_\beta$  is consistent for each behavior  $\beta$ .

(4.13) *Theorem*  $\Theta_{\mathcal{Q}}$  is consistent for any system  $\mathcal{Q}$ .

(4.14) *Theorem* The theory  $\theta$  is consistent.

We may also want to know if a formula  $\mathcal{G}$  or set  $\{\mathcal{G}_\alpha\} \subset \mathfrak{F}$  is consistent with a theory  $\Theta$  or if  $\mathcal{G}$  or the  $\mathcal{G}_\alpha$  are contained in  $\Theta$ . We define  $\mathfrak{G} \subset \mathfrak{F}$  to be *consistent with*  $\Theta$  if there is no  $\mathcal{H}$  with  $\mathfrak{G} \vdash_\Theta \mathcal{H}$  and  $\mathfrak{G} \vdash_\Theta \neg \mathcal{H}$ .

(4.15) *Proposition*  $\mathcal{H} \in \mathfrak{G}$  implies  $\mathfrak{G} \vdash_\Theta \mathcal{H}$ .

(4.16) *Proposition* If  $\mathfrak{G} \subset \mathfrak{H}$  then  $\mathfrak{G} \vdash_\Theta \mathcal{H} \Rightarrow \mathfrak{H} \vdash_\Theta \mathcal{H}$ .

(4.17) *Proposition* If  $\Theta \subset \Phi$  then  $\mathfrak{G} \vdash_\Theta \mathcal{H} \Rightarrow \mathfrak{G} \vdash_\Phi \mathcal{H}$ .

(4.18) *Lemma* If  $\mathfrak{G} \vdash_\Theta \mathcal{H}$  and  $\{\mathcal{H}\} \vdash_\Theta \mathcal{K}$  then  $\mathfrak{G} \vdash_\Theta \mathcal{K}$ .

(4.19) *Lemma* If  $\mathfrak{G} \vdash_\Theta \mathcal{H}$  and  $\mathfrak{G} \vdash_\Theta (\mathcal{H} \Rightarrow \mathcal{K})$  then  $\mathfrak{G} \vdash_\Theta \mathcal{K}$ .

(4.20) *Proposition*  $\mathfrak{G} \cup \{\mathcal{H}\} \vdash_\Theta \mathcal{F} \Leftrightarrow \mathfrak{G} \vdash_\Theta (\mathcal{H} \Rightarrow \mathcal{F})$ .

(4.21) *Proposition* The smallest theory containing  $\Theta$  and  $\mathfrak{G}$  is  $\{\mathcal{H} \in \mathfrak{F} \mid \mathfrak{G} \vdash_\Theta \mathcal{H}\}$ .

(4.22) *Theorem* If  $\mathfrak{G} \subset \Theta$  then  $\mathfrak{G}$  is consistent with  $\Theta$  if and only if  $\Theta \neq \mathfrak{F}$ .

A corollary to (4.22) is that all TLB theories except  $\mathfrak{F}$  are consistent. Hence we should expect  $\mathfrak{G}$  to be consistent with  $\Theta$  only if  $\{\mathcal{H} \in \mathfrak{F} \mid \mathfrak{G} \vdash_\Theta \mathcal{H}\} \neq \mathfrak{F}$ .

- (4.23) *Proposition*  $\mathfrak{G}$  is consistent with  $\Theta$  iff there is an  $\mathcal{F} \in \mathfrak{F}$  that is not provable from  $\mathfrak{G}$  in  $\Theta$ .
- (4.24) *Theorem*  $\mathfrak{G} \cup \{\mathcal{F}\}$  is consistent with  $\Theta$  iff  $\neg\mathcal{F}$  is not provable from  $\mathfrak{G}$  in  $\Theta$ .
- (4.25) *Theorem* If  $\mathfrak{G}$  is consistent with  $\Theta$  then for each  $\mathcal{F} \in \mathfrak{F}$ , either  $\mathfrak{G} \cup \{\mathcal{F}\}$  or  $\mathfrak{G} \cup \{\neg\mathcal{F}\}$  is consistent with  $\Theta$ .

In [10] it is shown that the (consistent) TLB theories are instances of S4.2. Thus  $\Theta$  is an instance of S4.2. Also proved in [10] is that when a system  $\mathcal{Q}$  has only one node, the theories associated with  $\mathcal{Q}$  are instances of S4.3.1. Much is known about S4.2 and S4.3.1. For instance S4.2 is the modal logic that describes special-relativistic space-time. S4.2 characterizes (normal) temporal logics with a directed model of time (execution) whereas S4.3.1 characterizes those with a sequential model of time. Consequently if a temporal logic is not an instance of S4.2, either it is not normal or it does not have a directed execution model so the type of distributed systems we have been discussing do not satisfy its axioms or inference rules in some way. If a temporal logic is not an instance of S4.3.1, either it is not normal or it does not have a sequential (or interleaved) model of time.

## 5. Experience with $\Theta$

To date, our experience with  $\Theta$  (and  $\Theta_{\mathcal{Q}}$ ) has been limited to some of the distributed system services discussed in [8]. [10] includes examples of distributed systems having a peer process architecture and how their correctness proofs can be reduced to single proofs of the correctness of a single program executing on a single node. The peer process architecture is especially useful for distributed real-time systems such as the one described in [8]. It results in system performance that is often over two orders of magnitude faster than conventional designs [4]. Further experience with  $\Theta$  will no doubt cause our knowledge of  $\Theta$  to grow as we prove additional theorems. But structurally  $\Theta$  will remain the smallest theory in TLB that all distributed systems must satisfy.

In summary the central ideas in  $\Theta$  include distributed convergence and clustering introduced in [7] that grew out of the ideas in [6] about convergence of algorithms to states, the idea to develop  $\Theta$  within Set Theory which simplifies its axioms and rules of inference and the concept of uncertainty in distributed systems and how to factor the uncertainty out of the correctness proofs by means of the *reduction formulae* in TLB.

## 6. References

- [1] R. Bull, "A note on the modal calculi S4.2 and S4.3," *Zeit. für math. Logik und Grund. der Math.*, No. 10, 1964, pp.53-55.
- [2] R. Bull, "An algebraic study of Diodorean modal systems," *Jour. of Symbolic Logic*, No. 30, 1965, pp. 58-64.
- [3] M. Dummett and E. Lemmon, "Modal logics between S4 and S5," *Zeit. für math. Logik und Grund. der Math.*, No. 5, 1959, pp. 250-264.
- [4] E. Feustel and N. Howes, "Broker Performance," *IDA Paper*, P-3439, 1998.
- [5] N. Howes, *Modern Analysis and Topology*, Springer-Verlag, New York, 1995.
- [6] N. Howes, "State Space Topology and a Theory of Convergence for Temporal Logics," *Workshop on Topology in Computer Science*, New York, 2002.
- [7] N. Howes, "On Distributed Convergence," *2003 Summer Topology Conference*, Brooklyn College, NY.
- [8] N. Howes, M. Mezzino and J. Sarkesain, "On Cyber Warfare Command and Control Systems," *9th International IEEE Command and Control Symp*, Copenhagen, 2004.
- [9] N. Howes, "Applications of Modal Logics to Computer Science," *2005 Topology Conference on applications of topology to computer science*, Dennison Univ, Ohio.
- [10] N. Howes, "A Temporal Logic for Distributed Systems," to appear.
- [11] G. Hughes and M. Cresswell, "A New Introduction to modal logic," *Routledge*, London, 1968.
- [12] S. Kripke, "A Completeness Theorem in Modal Logic," *Journal of Symbolic Logic*, Vol. 24, No. 1, 1959.
- [13] S. Kripke, "Semantic Analysis of Modal Logic I: normal propositional calculi," *Zeit. Math. Logik Grund. Math.*, Vol 9, 1963, pp. 67-96.
- [14] L. Lamport, "The Temporal Logic of Actions," *ACM Trans. on Prog. Lang. and Systems*, Vol. 16, No. 3, 1994.
- [15] E. Lemmon and D. Scott, "An Introduction to Modal Logic," *Am. Phil. Quarterly*, Monograph No. 11, 1977.
- [16] C. Lewis, "Symbolic Logic," *The Century Company*, New York, 1932.
- [17] C. Lewis, "A Survey of Symbolic Logic," 1918.
- [18] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*, Springer-Verlag, New York, 1992.
- [19] A. N. Prior, *Time and Modality*, Oxford Univ. Press, 1957.