Rules for trace consistent reasoning

R. Ramanujam

The Institute of Mathematical Sciences
C.I.T. campus
Madras 600 113
India

jam@imsc.ernet.in

ABSTRACT

Formulas of temporal logic which cannot distinguish between different interleavings of the same run are said to be *trace consistent*. So called partial-order methods can be applied for verification of such formulas, since checking such a property over an equivalence class of runs reduces to checking it for one representative.

In this paper, we present *inference rules* that typify this kind of reasoning. The rules lead us to a complete axiomatization of a linear time temporal logic, all formulas of which are trace consistent. The axiomatization is presented in a layered manner so that we can attempt to isolate the global reasoning required.

0 Introduction

In this paper, we study some aspects of *local reasoning* in the context of the Propositional Temporal Logic of Linear Time (PTL). Models for PTL are given by infinite runs (computations) of finite-state transition systems. For instance, the formula $\Box(p \supset \Diamond q)$ specifies the set of runs to be those such that every state along the run that satisfies the property p is eventually followed by a state that satisfies q. This has crucial implications for model checking, where we wish to check that every run of a given system satisfies a given formula: for a formula as above, we need to check every path issuing out of a node satisfying p and see if a node satisfying q is reachable. The use of next-time operators $\bigcirc$ in the formula can force this check along specific paths.

Such a view of PTL presents the models of formulas merely as finitely generated infinite sequences. This hides the fact that the transition systems which generate these computations themselves often possess some structure, which could perhaps considerably cut down the paths to be checked. PTL is mainly used to reason about behaviours of distributed systems, and transition systems modelling distributed programs have the following structure:

- States of the system are given as tuples of *local* states. If there are n processes in the system, a global state is an n-tuple of local states.
- A typical transition of the system is of the form $(q_1, \ldots, q_n) \stackrel{a}{\rightarrow} (r_1, \ldots, r_n)$, where a is an action of the system. Suppose $\theta(a) \subseteq \{1, 2, \ldots, n\}$ is the set of processes (components) in the system that participate in the action a. It is then the case that for all $j \notin \theta(a)$, $q_j = r_j$.

In addition, if the processes communicate only via synchronization rather than message passing, the following additional property holds as well:

- If $(q_1, \ldots, q_n) \stackrel{a}{\rightarrow} (r_1, \ldots, r_n) \stackrel{b}{\rightarrow} (s_1, \ldots, s_n)$, and $\theta(a) \cap \theta(b) = \emptyset$, then there exists a global state $(t_1, \ldots, t_n)$ such that $(q_1, \ldots, q_n) \stackrel{b}{\rightarrow} (t_1, \ldots, t_n) \stackrel{a}{\rightarrow} (s_1, \ldots, s_n)$.

We can now use this structure to reason as follows: whenever we are checking a property *local* to a specific process i, we can 'skip' over transitions along any run in which i is not taking part. For instance, in the formula $\Box(p \supset \Diamond q)$, suppose that the propositions p and q refer only to the local variables of process 1. Now consider the parallel program (intuitively) given by the expression $(ac)^\omega || (bc)^\omega$ whose behaviour is given as a pair of transition systems, $s_0 \stackrel{a}{\rightarrow} s_1 \stackrel{c}{\rightarrow} s_0$ and $t_0 \stackrel{b}{\rightarrow} t_1 \stackrel{c}{\rightarrow} t_0$. Assume that a is a local action of process 1, b that of 2 and c, a synchronization between them (perhaps the update of a shared variable). Clearly, there are infinitely many runs of the system starting at (s_0, t_0). Examples are runs on the words $(abc)^\omega$, $(bac)^\omega$, $(abcbac)^\omega$, $(abc)^{100}(bac)^{50}(abc)^\omega$ etc. However, the formula being considered cannot distinguish between all these runs, which are all different interleavings of the run on the word $(abc)^\omega$. Any of these runs satisfy this formula *iff* every one of these runs satisfy it. Thus checking the property on all runs reduces to checking it on any one of these runs.

Based on this observation, many *partial order methods* have been developed for speeding up verification of PTL formulas([V90], [KP92], [GW94]). In this approach, we define an equivalence relation on the set of runs so that to verify a property for an equivalence class of runs, it suffices to verify it for one run. However, this works only for those properties that are insensitive to interleavings, and it is only too easy to specify formulas in PTL which hold for some interleavings and not for others. For instance, in the system above, consider the formula $\Box(\lambda_c \supset \bigcirc \lambda_a)$, where the proposition λ_x denotes that the next transition of the run at that state is on action x. This singles out two runs, on the words $(abc)(abc)^\omega$ and $(bac)(abc)^\omega$, among the equivalence class of runs referred to. For such formulas, the methods suggested above for pruning the state space do not apply.

PTL formulas which cannot distinguish between different interleavings of the same run are called *equivalence robust* [P93], or *trace consistent*[T95]. Checking whether a given formula is trace consistent or not is decidable ([DGP95], [PWW96]), though of high complexity. In [R96c], we studied the alternative of considering stronger notions of satisfiability (and associated notions of validity) for PTL, where one asks whether there is an equivalence class of runs satisfying a formula, rather than a single such run. In this context, it is important to study trace consistent subsets of PTL, in which *every* formula, by choice, is trace consistent. When they can be identified syntactically as done with increasingly larger subsets in [T95], [T94] and [R96a], the structure of temporal connectives reflects the trace consistent reasoning involved. [TW97] presents an expressively complete logic where every formula is trace consistent.

While the papers mentioned above study trace consistency principally from the viewpoint of PTL satisfiability and model checking, in this paper, we take up the study from an *axiomatic* viewpoint. We believe that the insights gathered from deductive methods and model checking are both complementary and valuable. It has been argued that the major hopes for verification theory lie in combining deductive methods with model checking ([Ru96]). There is another motivation for the study of inference rules that involve trace consistent reasoning: they help us to isolate structurally those properties which we can reason about locally, componentwise in a distributed system, and those which must be analyzed globally. In this sense, this paper is a (modest) contribution to the theory of *compositional reasoning* about distributed systems.

For the study, we have chosen a simple model of distributed systems and a structurally straightforward subset of PTL, presented earlier in [R96a]. This choice of logical language and models has been so that we can focus on the issues peculiar to local reasoning which are already quite nontrivial. The global next-time modality is the major culprit in violation of trace consistency, and hence is replaced by a weaker, less global, next-time modality that preserves trace consistency. Otherwise, the logic is basically PTL for each agent and globally defined to be boolean combinations of located formulas : for instance $\alpha@i \wedge \beta@j$ (read α holds at i). The model is a distributed network consisting of a fixed finite number of sequential components that synchronize by performing common actions together. Each component is a finite state automaton, and when a common ("handshake") action is performed, it is a 'lock-step' transition involving several automata at once.

This is a simple set-up, but axiomatizing the valid formulas is already quite non-trivial. The reason for this is that we cannot simply have an axiom system for each agent, namely that for PTL, and globally use the propositional calculus. When a is a synchronization between 1 and 2, the formulas 'the next action of agent 1 is a', and 'a is henceforth never enabled for agent 2' are by themselves satisfiable, but their conjunction is not. Such synchronization can in fact be seen as a local liveness property : if agent 1 needs to synchronize with 2, eventually both do so. This requires an induction principle for the logic, quite distinct from that required for eventuality. (Note that this is a kind of fairness property we reason about *in* the logic, as opposed to those which we assume externally.)

Consider the standard induction rule of linear time temporal logic, which asserts that the eventuality modality steps through the successor modality. This rule is crucial for obtaining a complete axiomatization.

$$(Ind)\ \frac{\alpha \supset \bigcirc\alpha}{\alpha \supset \Box\alpha}$$

To verify that a property α holds at s henceforth, it suffices to show that it is an invariant (preserved by every transition) at any state reachable from s. Now, this is the sort of reasoning we need to do on the *global transition system* of the distributed program, and we may expect that a rule of the above form, where instead of α, we have a global invariant (maintained by global transitions) of the form $\bigwedge_k \alpha_k$ will do the job. However, the global $\bigcirc$ is crucially used in the premise above, and it is not as such available in trace consistent logics. This forms the central technical issue addressed in obtaining a complete axiomatization in this paper.

On the other hand, a 'local' version of the above rule, as given below, is sound, but too weak. The rule uses local premises, each saying that α_k is a local invariant, to conclude globally that their conjunction holds henceforth.

$$(Ind)\ \frac{\vdash_k \alpha_k \supset \bigcirc\alpha_k}{(\bigwedge_k \alpha_k@k) \supset (\bigwedge_k \Box\alpha_k@k)}$$

Suppose we are at a global state $(s_1, \ldots, s_n)$ of a distributed program, and wish to assert that a property, say $\beta@i$ holds henceforth in Γ, the set of all global states of the program, reachable from this state. The formula $\bigwedge_k \alpha_k@k$ might well be an invariant in Γ, but applying the rule requires that $\alpha_k@k$ be a local invariant for each k. But this is too strong a requirement, since there may be local states satisfying $\alpha_k@k$ from which a transition to a local state satisfying $\neg\alpha_k@k$ may well be possible, except that they never form part of global states in Γ. Therefore, we go for a weaker premise, which involves a next-time modality less expressive than the global $\bigcirc$, but defines next-instants for groups of agents that synchronize.

$$(Ind)\ \frac{(\bigwedge_k \alpha_k@k) \supset \bigwedge_{a\in\Sigma}[a](\bigwedge_{i\in\theta(a)} \alpha_i@i)}{(\bigwedge_k \alpha_k@k) \supset (\bigwedge_k \Box\alpha_k@k)}$$

In fact, we will work with a logic where even such a restricted modality is not available globally, but we will get the same effect by an appropriately strengthened local next-state modality.

In [R96a], we have studied trace consistent logics with greater expressiveness than the one presented here, with present and past tense modalities. While we certainly need to study reasoning in these enriched logics, the basic difficulties in axiomatization are already present in the simpler logic studied here, so we content ourselves with it for this short presentation.

1 Preliminaries

We begin with the notion of a **distributed alphabet**. Fix a finite set of *locations* or *agents* $Loc = \{1, \ldots, n\}, n > 0$. A distributed alphabet is a tuple $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$, where each Σ_i is a finite nonempty set of *actions* of agent i. When an action a is in $\Sigma_i \cap \Sigma_j, i \neq j$, we think of it as a synchronization between i and j. Given a distributed alphabet $\widetilde{\Sigma}$, we often speak of the set

$\Sigma \stackrel{\text{def}}{=} \Sigma_1 \cup ... \cup \Sigma_n$ as the alphabet of the system. We also make implicit use of the associated function $\theta : \Sigma \to 2^{\{1,\ldots,n\}}$ defined by $\theta(a) \stackrel{\text{def}}{=} \{i \mid a \in \Sigma_i\}$. We will keep a fixed n and a fixed distributed alphabet for the discussion from now on.

The frames for our logics are networks of n transition systems over $\widetilde{\Sigma}$. As usual, a transition system over a finite nonempty alphabet Σ is a pair $T = (Q, \to)$, where Q is a **finite** set of states, $\to \subseteq Q \times \Sigma \times Q$ is the transition relation. We write $q \xrightarrow{a} q'$ to mean that $(q, a, q') \in \to$. A (finite or infinite) *run* from $q_0 \in Q$ is a sequence of transitions of the system: $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \ldots$. Let R_T denote the set of runs of T.

Let $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ be a distributed alphabet. A **system** over $\widetilde{\Sigma}$ is a tuple $\mathcal{T} = (T_1, \ldots, T_n)$, where $T_i = (Q_i, \to_i)$ is a transition system over Σ_i, for $i \in Loc$. The **product transition** relation of the system $\to \subseteq (Q_1 \times \ldots \times Q_n) \times \Sigma \times (Q_1 \times \ldots \times Q_n)$ is defined as usual : $(q_1, \ldots, q_n) \xrightarrow{a} (q'_1, \ldots, q'_n)$ iff $\forall i \in \theta(a) : q_i \xrightarrow{a}_i q'_i$, and $\forall j \notin \theta(a) : q_j = q'_j$. We can use the product transitions to define the *global runs* of $\mathcal{T}$, and the set of all global runs is denoted $R_{\mathcal{T}}$.

Given $\delta \in R_{\mathcal{T}}$, we can speak of the *i-projection* of δ, given by projecting down to the i^{th} components of states and actions in Σ_i. (Essentially, erase all states not in Q_i, and all actions not in Σ_i.) It is easy to check that this gives us an element of R_{T_i}, and we will denote this by $\delta\lceil i$. Note that $\delta\lceil i$ can be finite even if δ is itself infinite.

We can now define an equivalence relation $\sim$ on runs of $\mathcal{T}$: $\delta \sim \delta'$ iff $\forall i, \delta\lceil i = \delta'\lceil i$. This equates different interleavings of the same run. We will denote the equivalence class of δ by $[\delta]$.

Below let δ be of the form $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \ldots$ (finite or infinite). Similarly let $\delta' = y_0 \xrightarrow{b_0} y_1 \xrightarrow{b_1} \ldots$. By $\delta(0 \ldots m)$, we denote the initial segment $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \ldots x_m$. We use the notation $x_m[j]$ to denote the j^{th} component (local state) of the n-tuple in the m^{th} instant.

We now define some special functions describing the views of agents at different points of global time and relating different agents' views.

The **local clock functions** $clock_{\delta,i}$: $\{0, 1, \ldots\} \to \{0, 1, \ldots\}$ are defined by: $clock_{\delta,i}(0) \stackrel{\text{def}}{=} 0$. $clock_{\delta,i}(k+1) \stackrel{\text{def}}{=} clock_{\delta,i}(k)+1$, if $a_k \in \Sigma_i$, and $clock_{\delta,i}(k+1) \stackrel{\text{def}}{=} clock_{\delta,i}(k)$, otherwise. We will speak of global instants and i-local instants with the implicit assumption of their relationship as given by these functions.

Proposition 1.1 *Suppose $\delta \sim \delta'$. If $clock_{\delta,i}(k) = clock_{\delta',i}(l)$, then we have $\delta(0 \ldots k) \lceil i = \delta'(0 \ldots l)\lceil i$.*

Thus, at any local instant m for agent i, the set $\{\delta'(0 \ldots k) \mid \delta' \in [\delta], clock_{\delta',i}(k) = m\}$, represents the global i-view of the system.

While we have presented the programs here to be obtained by products of transition systems, the framework applies equally well to the class of n-variable programs introduced in [R96c]. We can think of each agent as managing a group of program variables, and a synchronization as an update of a shared variable. Further, we can weaken the product assumption to consider systems which are defined by global transitions that respect the forward diamond condition referred to in the previous section. However, we keep the more restricted presentation for the sake of uniformity with [R96a], where the logic was originally introduced.

2 The logics

2.1 The basic logic L_0

Let $P = (P_1, \ldots, P_n)$ be a tuple of countable sets of atomic propositions. We will speak of $p \in P_i$ as a proposition of agent i. We do not assume disjointness of P_i and P_j for distinct agents i and j, as there can be common meaningful propositions. ("Has terminated" is one such.) The **i-formulas** are defined as follows:

$$\Phi_i \ ::= \ p \ \in \ P_i \mid \neg\, \alpha \mid \alpha \ \vee \ \beta \mid \langle a \rangle\, \alpha, \ a \in \Sigma_i \mid \alpha \mathbf{U} \beta$$

The **global formulas** are defined from these using boolean combinations: (We use the convention that α, β etc denote local formulas, whereas ϕ, ϕ' etc are used for global formulas.)

$$L_0 ::= \alpha @ i, \ \alpha \in \Phi_i, \ i \in Loc \mid \neg\, \phi \mid \phi_1 \ \vee \ \phi_2$$

Note that there are no modalities at the global level, and that formulas in Φ_i cannot refer to any other agent j in the system. Further note that L_0 is parametrized by the distributed alphabet.

Typically linear time logics have an $\bigcirc$ modality rather than the action-indexed successor modality, as the successor state at any instant is unique in linear time models. Here, the main purpose of using $\langle a \rangle$ is to describe synchronizations. When we encounter a formula of the form $(\langle a \rangle \alpha)@1 \wedge (\langle a \rangle \beta)@2$, where $\{1, 2\} \subseteq \theta(a)$, we understand that there will be a future instant when $\alpha@1 \wedge \beta@2$ will hold.

A **frame** for the logic is a pair $F = (\mathcal{T}, \delta)$, where $\mathcal{T} = (T_1, \ldots, T_n)$ is a system over $\widetilde{\Sigma}$ and $\delta \in R_{\mathcal{T}}$ is an infinite run of system $\mathcal{T}$. A **model** is a pair $M = (F, V)$, where F is a frame, and $V = (V_1, \ldots, V_n), V_i : Q_i \rightarrow 2^{P_i}$ give the local valuations. Given a model M, we define its i^{th} projection to be the tuple $M_i = ((T_i, \delta\lceil i), V_i)$.

We first define satisfaction of local formulas : $M_i, k \models_i \alpha$ denotes that the i-formula α is satisfied at (local) instant k for agent i. Below let $\delta\lceil i = q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \ldots$.

- $M_i, k \models_i p$ iff $p \in V_i(q_k)$.
- $M_i, k \models_i \neg\alpha$ iff $M_i, k \not\models_i \alpha$.
- $M_i, k \models_i \alpha \vee \beta$ iff $M_i, k \models_i \alpha$ or $M_i, k \models_i \beta$.
- $M_i, k \models_i \langle a \rangle \alpha$ iff q_{k+1} exists, $a_k = a$ and $M_i, k+1 \models_i \alpha$.
- $M_i, k \models_i \alpha \mathbf{U} \beta$ iff $\exists m$ such that $k \leq m, M_i, m \models_i \beta$ and for every $l : k \leq l < m : M_i, l \models_i \alpha$.

We now define the semantics of L_0 formulas. Since there are no global modalities, the formulas are not defined at every instant, but only at the initial state of the run.

- $M, 0 \models \alpha@i$ iff $M_i, 0 \models_i \alpha$.
- $M, 0 \models \neg\phi$ iff $M, 0 \not\models \phi$.
- $M, 0 \models \phi_1 \vee \phi_2$ iff $M, 0 \models \phi_1$ or $M, 0 \models \phi_2$.

As usual, we say that a formula ϕ is satisfiable if there exists a model M such that $M, 0 \models \phi$. A formula is valid if and only if its negation is not satisfiable.

2.2 The logic L_1

L_0 is a very simple logic, but expressively weak. Specifically, the fact that agents in the system *learn* about others' states on synchronization is not reflected in L_0-formulas. Indeed, the whole purpose of synchronizing is to exchange information, and hence we should be able to utilize this in reasoning. Below we propose an extension, where agents can "see" the local states of their partners in synchronization at the time they synchronize. Of course, this also means that they can see the other's future at that instant, and hence this gives a good deal of expressive power.

Formally, the logic L_1 is defined as follows:

$$\Psi_i \ ::= \ p \ \in \ P_i \mid \neg\, \alpha \mid \alpha \ \vee \ \beta \mid \langle a \rangle \ \phi, \ a \in \Sigma_i, \ AV(\phi) \subseteq \theta(a) \mid \alpha \mathbf{U} \beta$$

$$L_1 ::= \alpha@i, \ \alpha \in \Psi_i, \ i \in Loc \mid \neg\, \phi \mid \phi_1 \ \vee \ \phi_2$$

$AV(\phi)$, the *agent vocabulary* of ϕ, is the set of agents mentioned in ϕ, and is inductively defined in the obvious manner:

$$AV(\alpha@i) \stackrel{\text{def}}{=} \{i\}; \ AV(\neg\phi) \stackrel{\text{def}}{=} AV(\phi); \ AV(\phi_1 \vee \phi_2) \stackrel{\text{def}}{=} AV(\phi_1) \cup AV(\phi_2)$$

The semantics needs to be suitably changed. Below we give only those parts of the definition that require change. Fix a model $M = (F, V)$, where $F = (\mathcal{T}, \delta)$ and $\delta = x_0 \xrightarrow{a_0} x_1 \ldots$.

- $M_i, k \models_i \langle a \rangle \phi$ iff $\exists l$: $clock_i(l) = k$, $clock_i(l+1) = k+1$, $a_l = a$ and $M, l+1 \models \phi$.

Note that the expression $\exists l$ in the definition can be replaced by '$\exists$ a unique l'; this means that the formula $\langle a \rangle \phi$ implies its dual formula $[a]\phi$, as we would expect in linear time temporal logics.

- $M, k \models \alpha@i$ iff $M_i, clock_i(k) \models_i \alpha$.
- $M, k \models \neg\phi$ iff $M, k \not\models \phi$.
- $M, k \models \phi_1 \vee \phi_2$ iff $M, k \models \phi_1$ or $M, k \models \phi_2$.

In L_1, we not only have both local and global formulas in the syntax, but local formulas can refer to global formulas as well. Therefore, to avoid confusion, the reader is requested to note the following **Notational Conventions**: We will use α, β etc to refer to local formulas and ϕ, ϕ' etc denote global formulas. Let $a \in \Sigma_i$, and ϕ be a global formula. Then $\langle a \rangle \phi$ is an i-formula, and $(\langle a \rangle \phi)@i$ is a global formula, which we often abbreviate as $\langle a \rangle_i \phi$. We use $\bigcirc \phi$ to denote the i-formula $\bigvee_{a \in \Sigma_i} \langle a \rangle \phi$.

The i-formula $True_i$ is defined to be $p_0 \vee \neg p_0$, where p_0 is a fixed propositional letter in P_i. Now $True_i@i$ is a global formula. Let $a \in \Sigma$. The abbreviation $en_i(a)$ is used to denote the i-formula $\langle a \rangle (True_i@i)$. We further use the abbreviation $en_G(a)$ to denote the global formula $\bigwedge_{j \in \theta(a)} (en_j(a)@j)$. Note that $True_i@i$ is a valid formula, and that $en_i(a)$ asserts that the action a is enabled locally. The stronger assertion $en_G(a)$ says that a is enabled at the global state where it is asserted. We also define $dis_G(a) \stackrel{\text{def}}{=} \neg en_G(a)$, and $dis_i(a) \stackrel{\text{def}}{=} \neg en_i(a)$, to refer to disabled actions globally and locally.

Derived connectives and dual modalities $(\alpha \wedge \beta, \alpha \supset \beta, [a]\phi, \Diamond\alpha, \Box\alpha)$ are standard.

Note that global dependencies can be expressed nicely in this logic. For instance, suppose the distributed alphabet $\widetilde{\Sigma}$ is such that $a \in \Sigma_1 \cap \Sigma_2, b \in \Sigma_2 \cap \Sigma_3$, $c \in \Sigma_3 \cap \Sigma_4$ and $p \in P_4$. Then a formula like $\langle a \rangle_1 \langle b \rangle_2 \langle c \rangle_3 p@4$ describes a sequence of dependent communications. In this manner one can 'walk down' the global run as in $TrPTL$ [T94]. The following propositions will be useful later on. The latter asserts that every formula of the logic L_1 is trace consistent.

Proposition 2.1 *Let $k \geq 0$ and $i \in \theta(a)$ such that $M, k \models (\langle a \rangle \phi)@i$. Let m be the least instant such that $m \geq k$ and $a_m = a$. Then $M, m+1 \models \phi$.*

Proposition 2.2 *Consider models $M = (\mathcal{T}, \delta, V)$ and $M' = (\mathcal{T}, \delta', V)$, where $\delta \sim \delta' \in R_{\mathcal{T}}$. For any formula ϕ of L_1, $M, 0 \models \phi$ iff $M', 0 \models \phi$.*

3 Axiom system for L_1

We present the axiomatization only for the more expressive logic. The system is presented in a layered manner. We have one axiom system Ax_i for each agent i in the system, and in addition a global system Ax_g to reason about synchronization. We use the notation $\vdash_i \alpha$ to mean that the formula $\alpha \in \Psi_i$ is a theorem of system Ax_i. Similarly, $\vdash \phi$ means that ϕ is a theorem of the global system Ax_g. Both the systems use the theorems of each other recursively, and we refer to the combined system as AX.

Ax_i, **The axiom schemes for agent** i

$(A0_i)$ All the substitutional instances of the tautologies of PC

$(A1_i)$ $[a](\phi_1 \supset \phi_2) \supset ([a]\phi_1 \supset [a]\phi_2)$

$(A2_i)$ $en_i(a) \supset dis_i(b) \quad (a \neq b)$

$(A3_i)$ $\langle a\rangle\phi \supset [a]\phi$

$(A4_i)$ $\alpha\mathbf{U}\beta \equiv \beta \vee (\alpha \wedge \bigcirc(\alpha\mathbf{U}\beta)@i)$

Inference rules

$(MP_i)\ \dfrac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (NG_i)\ \dfrac{\vdash \phi, AV(\phi) \subseteq \theta(a), a \in \Sigma_i}{[a]\phi}$

Before we proceed to present the global axiom system, some remarks are in order. As mentioned in Section 0, for eventuality, we need induction rules that step through the more expressive next-time modality in L_1. Moreover, when we are at a global state where agent i is waiting for a synchronization with agent j, eventually such a synchronization must occur, and this again necessitates a form of induction. Such rules add to the complexity of reasoning in the logic.

Ax_g, **Global axiom schemes**

$(B0)$ $(\neg\alpha)@i \equiv \neg\alpha@i$

$(B1)$ $(\alpha \vee \beta)@i \equiv (\alpha@i \vee \beta@i)$

$(B2)$ $\bigvee\limits_{a \in \Sigma} en_G(a)$

$(B3)$ $en_G(a) \supset (([a]\phi)@i \supset \bigwedge\limits_{j \in \theta(a)} ([a]\phi)@j) \qquad a \in \Sigma_i$

Inference rules

$(MP)\ \dfrac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (GG)\ \dfrac{\vdash_i \alpha}{\alpha@i}$

$$(GM)\ \dfrac{\bigwedge\limits_{i \in \theta(a)} \phi_i \supset \bigvee\limits_{j \notin \theta(a)} \alpha_j@j}{\bigwedge\limits_{i \in \theta(a)} (\langle a\rangle\phi_i)@i \supset \bigvee\limits_{j \notin \theta(a)} \alpha_j@j}$$

Below let ψ *be of the form* $\bigwedge\limits_{k} \alpha_k@k$, *let* ψ_b *denote* $\bigwedge\limits_{j \in \theta(b)} \alpha_j@j$.

$$(Sy)\ \dfrac{\begin{array}{c}\psi \supset dis_G(a) \quad a \in \Sigma_i \\ \psi \wedge en_G(b) \supset ([b]\psi_b)@k \quad b \in \Sigma_k - \Sigma_i\end{array}}{\psi \supset dis_i(a)@i}$$

$$(Ev)\ \dfrac{\begin{array}{c}\psi \supset \alpha@i \\ \psi \wedge en_G(b) \supset ([b]\psi_b)@k \quad b \in \Sigma\end{array}}{\psi \supset (\Box\alpha)@i}$$

Note that we do not have an analog of (A0) in Ax_g to derive tautologies, but "@-versions" of tautologies can be got as follows: for instance consider $p \supset \neg\neg p$, $p \in P_i$. This is a theorem of Ax_i, and hence by rule (GG), $(p \supset \neg\neg p)@i$ is derived. Now we use (B0) and (B1) to derive $(p@i \supset (\neg\neg p)@i)$. (B2) asserts that models are infinite runs of the system. (B3) says that at any global state when a is enabled (all the synchronizing agents being ready to perform a together), they must all agree on the global state resulting after the synchronization. The soundness of this axiom directly follows from Proposition 2.1.

The rule (GM) describes joint moves in the system. In particular, when an a-move is made, this leaves the states of agents outside $\theta(a)$ unchanged, and this is stated in (GM). This rule typifies the pattern of reasoning in a "true concurrency" based logic.

Note that the validity of premise $\psi \supset dis_G(a)$ in rule (Sy) *cannot* by itself give the conclusion $\psi \supset dis_i(a)@i$, despite such an appearance. The premise merely states that in all ψ-states, *some one* of the agents in $\theta(a)$ is unwilling to do an a, whereas the latter is an assertion that at ψ-states, the specific agent $i \in \theta(a)$ cannot decide on doing a, and hence implies that at ψ-states, no agent in $\theta(a)$ may commit to a as its next action. This does require the extra premise leading to an induction argument.

Theorem 3.1 *The combined system AX provides a sound and complete axiomatization of the valid formulas of L_1.*

The axioms are easily seen to be sound. Among the rules, checking that (Sy) preserves validity is worth remarking on. (Soundness of (Ev) follows similarly.) Assume ψ such that the premises of (Sy) are valid, but not the conclusion. Then there exists a model M based on a run $\delta = x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \ldots$ such that $M, 0 \models \psi \wedge en_i(a)@i$. By the semantics of the modality, there exists k such that $clock_i(k) = clock_i(0)$, $a_k = a$, and for all l such that $0 \leq l < k$, $a_l \notin \Sigma_i$. $M, k \models en_G(a)$.

Note that $M, 0 \models \psi$ and ψ is of the form $\bigwedge\limits_j \alpha_j@j$. By the semantics of @-formulas, for every $j \notin \theta(a_0)$, $M, 1 \models \alpha_j@j$ as well. Now consider $j \in \theta(a_0)$. $M, 0 \models en_G(a_0)$ and by the validity of the premise, $M, 0 \models [a_0]\psi_{a_0}@j$. But then, by Proposition 2.1, $M, 1 \models \psi_{a_0}@j$ and hence $M, 1 \models \alpha_j@j$. Thus, we have shown that $M, 1 \models \psi$. Proceeding this way, we get $M, k \models \psi$, and by validity of premise, $M, k \models dis_G(a)$, contradicting the fact that $M, k \models en_G(a)$.

The crucial use of the enriched L_1 modality in the rules (Sy) and (Ev) should be noted. The formula $[b]\psi_b$ above is much stronger than the L_0 formula $\bigwedge\limits_{j \in \theta(b)} ([b]\alpha_j)@j$. The latter form would also give a sound rule, but proves too weak to give completeness. With an infinitary version of the (Sy)-scheme, not in the sense of having infinitely many premises, but in parametrizing the rule by a number m (thus yielding one scheme for each m), we can obtain an axiomatization of L_0-valid formulas. Details can be found in [R96b].

4 Completeness

For technical convenience, assume that Ψ_i is enriched with $\bigcirc\alpha, \alpha \in \Psi_i$, where $\bigcirc\alpha$ is semantically equivalent to $\bigvee\limits_{a \in \Sigma_i} \langle a \rangle \alpha@i$. Further assume that we have $en_i(a) \in \Psi_i$ for $a \in \Sigma_i$, and $en_G(a) \in L_1$, for $a \in \Sigma$, with $Voc(en_G(a)) = \theta(a)$.

We can define, for any L_1-formula ϕ and any $\alpha \in \Psi_i$, the sets of their subformulas $CL(\phi)$ and $CL_i(\alpha)$ by simultaneous induction in such a way that :

- $\phi \in CL(\phi)$ and $\alpha \in CL_i(\alpha)$.
- if $\alpha@i \in CL(\phi)$ then $\{\beta@i | \beta \in CL_i(\alpha)\} \subseteq CL(\phi)$.
- if $(\langle a\rangle\phi')@i \in CL(\phi)$ then $CL(\phi') \subseteq CL(\phi)$ and $\forall j \in \theta(a), en_j(a)@j \in CL(\phi)$.
- if $\langle a\rangle\phi' \in CL_i(\alpha)$ then $\{\langle a\rangle\beta@i, \beta | \beta@i \in CL(\phi')\} \subseteq CL_i(\alpha)$.
- if $\phi' \in CL(\phi)$ then $\neg\phi' \in CL(\phi)$; a similar condition holds for $CL_i(\alpha)$ and here $\neg\neg$ is treated as identity.
- if $\phi_1 \vee \phi_2 \in CL(\phi)$ then $\phi_1, \phi_2 \in CL(\phi)$.
- if $\beta_1 \vee \beta_2 \in CL_i(\alpha)$ then $\beta_1, \beta_2 \in CL_i(\alpha)$.
- if $\beta_1 \mathbf{U} \beta_2 \in CL_i(\alpha)$ then $\beta_1, \beta_2, \bigcirc(\beta_1 \mathbf{U} \beta_2) \in CL_i(\alpha)$.

It can be checked that $CL(\phi)$ is linear in the size of ϕ. Fix a formula $\phi_0 \in L_1$. We will refer to $CL(\phi_0)$ simply as CL. By CL_i, we refer to the union of sets $CL_i(\alpha)$, where $\alpha@i \in CL$. Let $X \subseteq CL$. By $X\lceil i$, we refer to the subset $\{\alpha \mid \alpha@i \in X\}$ of CL_i.

Call $A \subseteq CL_i$ an *i-atom* iff it is propositionally consistent, and satisfies the conditions :

- if both $en_i(a)$ and $en_i(b)$ are in A, then $a = b$.
- $\alpha\mathbf{U}\beta \in A$ iff ($\beta \in A$ or both α and $\bigcirc(\alpha\mathbf{U}\beta)$ are in A).

Call $X \subseteq CL$ an *atom* if and only if it is propositionally consistent, and:

- $en_G(a) \in X$ iff for every $i \in \theta(a)$, $en_i(a)@i \in X$.
- for every i, $X\lceil i$ is an i-atom.

Let AT denote the set of all atoms and AT_i denote the set of i-atoms.

For $X, Y \in AT$, define $X \overset{a}{\Rightarrow}{}' Y$ iff for every $i \in \theta(a)$, the following conditions hold : for every $\langle a\rangle\phi$ in CL_i, $(\langle a\rangle\phi)@i \in X$ iff $\phi \in Y$, and for every $\bigcirc\alpha$ in CL_i, $(\bigcirc\alpha)@i \in X$ iff $\alpha@i \in Y$, and for every $j \notin \theta(a)$, $X\lceil j = Y\lceil j$. This induces a relation on i-atoms: $A \overset{a}{\rightarrow}_i B$ iff there exist atoms X and Y such that $X \overset{a}{\Rightarrow}{}' Y$, $X\lceil i = A$ and $Y\lceil i = B$. We also say $a \in \Sigma$ is enabled at X if $en_G(a) \in X$.

Let $G \subseteq AT, \Rightarrow \subseteq \Rightarrow'$. Call the graph $(G, \Rightarrow)$ a *pseudo-model* for ϕ_0, if the following conditions hold :

1. There exists $X \in G$ such that $\phi_0 \in X$.
2. Every $X \in G$ has a successor in G.
3. Consider $X \in G$ and $a \in \Sigma$ such that $en_G(a) \in X$. Then there exists $Y \in G$ such that $X \overset{a}{\Rightarrow} Y$.
4. Consider $X \in G$ and $a \in \Sigma$ such that for some $i \in \theta(a)$, $en_i(a) \in X\lceil i$. Then there exists a sequence $X = X_0 \overset{b_0}{\Rightarrow} X_1 \ldots \overset{b_k}{\Rightarrow} X_k \overset{a}{\Rightarrow} Y$, $k \geq 0$.
5. Consider $X \in G$ such that for some i, $(\alpha\mathbf{U}\beta)@i \in X$. Then there exists a sequence $X = X_0 \overset{b_0}{\Rightarrow} X_1 \ldots \overset{b_k}{\Rightarrow} X_k$, $k \geq 0$ such that $\beta@i \in X_k$.

Lemma 4.1 *If ϕ_0 is AX-consistent, then there exists a pseudo-model for ϕ_0 in $(AT, \Rightarrow')$.*

Suppose the lemma holds. Then we can use the pesudo-model $(G, \Rightarrow)$ to construct an infinite atom-run $X_0 \stackrel{a_0}{\Rightarrow} X_1 \stackrel{a_1}{\Rightarrow} \ldots$ which has the following properties : $\phi_0 \in X_0$; whenever there exists $en_i(a) \in X_k \lceil i$ there exists an $l \geq k$ such that l is the next index after k with $a = a_l$; whenever there exists $(\alpha \mathbf{U} \beta)@i \in X_k$ there exists an $l \geq k$ such that $\beta@i \in X_l$ and for all $k \leq j < l, \alpha@i \in X_j$. Now consider $Q_i = \{X\lceil i | X \text{ occurs in the constructed infinite run}\}$. $Q_i \subseteq AT_i$. Consider the system defined by $\mathcal{T} \stackrel{\text{def}}{=} (T_1, \ldots, T_n)$, where $T_i = (Q_i, \rightarrow_i \cap (Q_i \times \Sigma_i \times Q_i))$. Clearly, $R_{\mathcal{T}}$ contains a run $\delta = Y_0 \stackrel{a_0}{\longrightarrow} Y_1 \stackrel{a_1}{\longrightarrow} \ldots$, where $Y_j = (X_j\lceil 1, \ldots, X_j\lceil n)$. Consider the model $M = ((\mathcal{T}, \delta), V)$, where for $A \in Q_i$, $V_i(A) \stackrel{\text{def}}{=} A \cap P_i$. We can then show by (double) induction that for every k and for every $\phi \in CL$, $M, k \models \phi$ if and only if $\phi \in X_k$. But then, since $\phi_0 \in X_0$, we get $M, 0 \models \phi_0$, and we have a model for the consistent formula ϕ_0 and completeness of AX is proved.

We now run through the proof of the lemma. Fix ϕ_0, a consistent formula. Consider $G \stackrel{\text{def}}{=} \{X \in AT | X \text{ is a maximal } AX\text{-consistent subset of } CL\}$.

For any finite set of formulas Z, $\overline{Z}$ denotes the conjunction of all formulas in Z. For $X \in G$, let X^i denote the conjunction of formulas in the i-atom $X\lceil i$. Clearly, because of rule (GG), X^i is a maximal Ax_i-consistent subset of CL_i. Further, for $X \in G$, let $\widehat{X}$ denote the conjunction $\bigwedge_k X^k@k$, and for $a \in \Sigma, X \in G$, let X^a denote $\bigwedge_{i \in \theta(a)} X^i@i$. We see, thanks to axioms (B0) and (B1) that $\vdash \overline{X} \equiv \widehat{X}$. For $X, Y \in G$ define $X \stackrel{a}{\Rightarrow} Y$ iff $\widehat{X} \wedge \bigwedge_{i \in \theta(a)} (\langle a \rangle Y^a)@i$ is consistent and for every $j \notin \theta(a), X\lceil j = Y\lceil j$. It is easy to check that every $X \in G$ is indeed an atom and that $\Rightarrow \subseteq \Rightarrow'$. The claim is that $H = (G, \Rightarrow)$ is a pseudo-model for ϕ_0.

The first two conditions for H being a pseudo-model for ϕ_0 are easily seen to be satisfied. ϕ_0 being a consistent formula and in CL, there exists a maximal consistent subset of CL containing ϕ_0. If we prove the third condition, the second one follows from global axiom (B2). Suppose there exists $X \in G$ and $a \in \Sigma$ such that a is enabled in X. Working within Ax_i, we can derive i-theorems of the form $\overline{X^i} \supset \langle a \rangle \widetilde{\Gamma}_i$, where Γ_i is a non-empty subset of G, and $\widetilde{\Gamma_i}$ denotes the disjunction of Y^a, $Y \in \Gamma_i$. We can then show that $\bigwedge_{i \in \theta(a)} (\langle a \rangle \widetilde{\Gamma_i})@i \wedge \bigwedge_{j \notin \theta(a)} X^j$ is consistent. By rule (GM), we can show that there exists Y such that $\bigwedge_{i \in \theta(a)} Y^i@i \wedge \bigwedge_{j \notin \theta(a)} X^j$ is consistent. Now the union of i-formulas from Y, $i \in \theta(a)$, and j-formulas from X, $j \notin \theta(a)$ has been shown to be consistent and is a subset of CL. Hence there exists $Z \in G$ containing this subset, and by definition $X \stackrel{a}{\Rightarrow} Z$, as required.

To prove condition 4 above, suppose $X \in G$ and $en_i(a) \in X\lceil i$ for some i. Form the least subset of Γ of G such that $X \in \Gamma$, and if $X_1 \in \Gamma$ and there exists a path in $(G, \Rightarrow)$ from X_1 to some X_2 which involves no action in Σ_i, then $X_2 \in \Gamma$. If there exists $Z \in \Gamma$ such that a is enabled in Z, we are done. We claim that this is indeed the case, by showing a contradiction otherwise.

Define $\psi \stackrel{\text{def}}{=} \bigwedge_k Y_k@k$, where $Y_k = \bigvee_{Z\in\Gamma} Z^k$. By assumption and condition (3) for pseudo-models, for every $Z \in \Gamma$, we have $\vdash \overline{Z} \supset dis_G(a)$, and hence $\vdash \psi \supset dis_G(a)$. Suppose we can show that for each k and $b \in \Sigma_k - \Sigma_i$, the following formula is a theorem: $\vdash \psi \wedge en_G(b) \supset ([b]\psi_b)@k$. Then we get from rule (Sy) and the fact that $X \in \Gamma$ that $\overline{X} \supset dis_i(a)@i$, which contradicts the assumption that $en_i(a) \in X\lceil i$.

Now, suppose that the above formula is not a theorem for some k and b in $(\Sigma_k - \Sigma_i)$. Then $\psi \wedge en_G(b) \wedge (\langle b\rangle\neg\psi_b)@k$ is consistent. If $\widetilde{\Gamma}$ denotes the disjunction of all atoms in Γ, we have $\vdash \widetilde{\Gamma} \supset \psi$, and hence we find that for some $Z \in \Gamma$, $\overline{Z} \wedge en_G(b) \wedge (\langle b\rangle\neg\psi_b)@k$ is consistent. From this we can conclude that b is enabled in Z, and (using rule GM) that there is a b-successor of Z which is consistent with $\bigvee_{j\in\theta(b)} \neg Y_j@j$. But then this b-successor of Z must be an atom outside Γ. We thus have an atom Z' such that $Z \stackrel{b}{\Rightarrow} Z'$ where $Z' \notin \Gamma$. But then $Z \in \Gamma$, there is a (one-step) path from Z to Z' using $b \notin \Sigma_i$, and $Z' \notin \Gamma$, clearly contradicting the closure condition on Γ.

The proof that condition (5) holds is similar. This completes the proof of the lemma and establishes the completeness theorem for AX.

5 System Validity

The axiom system presented here is meant for establishing pure logical validities, in the sense that theorems are formulas valid in all runs of all systems. This is of theoretical interest, whereas in practice, we are more interested in **system validity** where we are concerned with formulas satisfied by all runs of a *given system*. We now show that the presented system AX is valid for system validity as well.

Let $\widetilde{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ be a distributed alphabet. By abuse of notation, we say a system over $\widetilde{\Sigma}$ is a pair $S = (\mathcal{T}, V)$, where $\mathcal{T} = (T_1, \ldots, T_n)$, $T_i = (Q_i, \rightarrow_i)$ is a transition system over Σ_i, and $V = (V_1, \ldots, V_n)$, $V_i : Q_i \rightarrow 2^{P_i}$ give the local valuations. Let $R_{\mathcal{T}}$ denote the set of all infinite runs of $\mathcal{T}$. For an L_1 formula ϕ, we say $S \models \phi$ iff for every $\delta \in R_{\mathcal{T}}$, $M, 0 \models \phi$, where $M = (\mathcal{T}, \delta, V)$ is the associated model. Such a formula is said to be S-valid. For an i-formula α, we say $S \models_i \alpha$ iff $S \models \alpha@i$. Such an i-formula is said to be S_i-valid.

Theorem 5.1 *Let S be a given system. If α is a theorem of Ax_i, then $S \models_i \alpha$, and if ϕ is a theorem of Ax_g, we have $S \models \phi$.*

Fix a system S. It is easy that every axiom of Ax_i is S_i-valid, and every axiom of Ax_g is S-valid. To see that the inference rules of Ax_i preserve S_i-validity, first observe the following: let $\delta = x_0 \stackrel{a_0}{\rightarrow} x_1 \stackrel{a_1}{\rightarrow} \ldots$ be a run of S and M the associated model. Now, for any L_1 formula ϕ, and any $k \geq 0$, $M, k \models \phi$ iff $M', 0 \models \phi$, where M' is the model based on $\delta' = x_k \stackrel{a_k}{\rightarrow} x_{k+1} \stackrel{a_{k+1}}{\rightarrow} \ldots$. Now, consider rule NG_i

and assume that ϕ is S-valid, but that $[a]\phi$ is not S_i-valid. Let M be a model based on a run δ of S such that $M_i, 0 \models_i \langle a \rangle \neg\phi$. Let k be the first instant in δ such that $a_k = a$. Then, $M, k+1 \models \neg\phi$. But then by the observation above, we can construct a run $\delta' = x_{k+1} \stackrel{a_{k+1}}{\rightarrow} \ldots$ such that the induced model $M', 0 \not\models \phi$. contradicting S-validity of ϕ, and we are done.

We now show that the rule GM preserves S-validity. For this first observe that if M is any model based on run $\delta = x_0 \stackrel{a_0}{\rightarrow} x_1 \ldots$ and $a \in \Sigma$ such that $M, 0 \models en_G(a)$, then there exists a run $\delta' = x_0 \stackrel{a}{\rightarrow} y_0 \stackrel{b_0}{\rightarrow} y_1 \ldots$ such that $\delta' \sim \delta$. This follows from the fact that we are considering product systems; if k is the earliest instant in δ such that $a_k = a$, then all the actions $a_j, j < k$ are 'independent' of a and hence can be 'commuted'. Now consider rule GM and suppose the premise is S-valid but the conclusion is not. Then there is a model M based on a run δ such that $M, 0 \models \bigwedge_{i \in \theta(a)} (\langle a \rangle \phi_i)@i \wedge \bigwedge_{j \notin \theta(a)} \neg\alpha_j@j$. Note that $M, 0 \models en_G(a)$, and by the observation above, we can consider the model M' based on the equivalent run δ'. By proposition 2.2 (trace consistency) the formula above is satisfied at instant 0 in M'. But then $M', 1 \models \bigwedge_{i \in \theta(a)} \phi_i \wedge \bigwedge_{j \notin \theta(a)} \neg\alpha_j@j$. We now consider the model M'' based on the run δ' but starting at 1, and this violates the S-validity of the premise, and we are done.

The other rules are proved to preserve S-validity in a similar fashion. Thus, the presented system is sound for S-validity. It is clear that as yet it is not complete for S-validity (since we need to add system dependent axioms capturing the given transition structure). Obtaining such a complete system poses an interesting question for future study.

Acknowledgement: I thank the anonymous referees for most helpful comments. Section 5 was added thanks to a referee's suggestion.

References

[DGP95] Diekert, V., Gastin, P. and Petit, A., "Rational and recognizable complex trace languages", *Information and Computation*, vol 116, #1, 1995, 134-153.

[GW94] Godefroid, P. and Wolper, P., "A partial approach to model checking", *Information and Computation*, vol 110, 1994, 305-326.

[KP92] Katz, S. and Peled, D., "Interleaving set temporal logic", *TCS*, vol. 73, #3, 1992, 21-43.

[P93] Peled, D., "All from one and one from all: on model checking using representatives", Proc. CAV, *LNCS 697*, 1993, 409-423..

[PWW96] Peled, D., Wilke, T. and Wolper, P., "An algorithmic approach to proving closure properties of ω-regular languages", Proc. CONCUR, *LNCS 1119*, 1996.

[R96a] Ramanujam, R., "Locally linear time temporal logic", *Proc. IEEE LICS*, 1996, 118-127.

[R96b] Ramanujam, R., "Axiomatization of a partial order based temporal logic", *Bericht Nr 9605*, Christian-Albrechts Universität Kiel, June 1996.

[R96c] Ramanujam, R., "Trace consistency and inevitability", Proc. FST and TCS, *LNCS 1180*, 1996, 250-261.

[Ru96] Rushby, J., "Mechanized formal methods: progress and prospects", Proc. FST and TCS, *LNCS 1180*, 1996, 43-51.

[T94] Thiagarajan, P.S., "A trace based extension of propositional linear time temporal logic", *Proc. IEEE LICS*, 1994, 438-447.

[T95] Thiagarajan, P.S., "A trace consistent subset of PTL", Proc. CONCUR, *LNCS 962*, 1995, 438-452.

[TW97] Thiagarajan, P.S. and Walukiewicz, I., "An expressively complete linear time temporal logic for Mazurkiewicz traces", *Proc. IEEE LICS*, 1997.

[V90] Valmari, A., "A stubborn attack on state explosion", Proc. CAV, *LNCS 531*, 1990, 156-165.