Compositionality Made Up

Carlo Alberto Furia

December 2006

Abstract

This paper develops some methods and techniques for the practical use of compositionality to prove the correctness of modular systems, with reference to the metric temporal logic TRIO. It advocates a "lightweight" approach to compositionality, driven by the specifics of the system under development, where methodology is as important as technical solutions.

Contents

1 Framing the Compositionality Problem

This paper develops some methods and techniques for the practical use of compositionality to prove the correctness of modular systems specified with (metric) temporal logic.

At their core, compositional techniques consist in the application of a compositional inference rule which allows one to infer facts about the composition of some formulas from other — usually simpler — facts about the individual modules in isolation. There is a certain amount of work available in the literature about such compositional inference rules [Fur05]. The goal of this paper is to develop a practical approach to the problem, according to the guidelines that we are now going to lay down.

First, we provide a general definition of what we mean by compositionality (Section 1.1). Then, we summarize what are — from a practical viewpoint — some shortcomings of the compositional techniques that have been developed in a large part of the literature, and we correspondingly list the features of our approach to compositionality, in response to these shortcomings (Section 1.2). The approach is then developed in the remaining sections (2 through 6).

We point out that some parts of this general "plan" for compositionality have been developed elsewhere [FRMM06]; therefore, we refer the reader to [FRMM06] for such aspects that are not developed here.

1.1 Compositionality: A Definition

Our ultimate goal is pursuing a rather broad approach to the verification of modular systems. Therefore, we start from a definition of compositionality which is generic and of wide spectrum. More precisely, we recall the informal definition of compositionality that we provided in [Fur05].

> "Techniques and methods that permit the modularization of the verification process of a large system".

Some general aspects of what it means to "specialize" such a general definition have been presented and discussed in [Fur05], so we do not repeat them here.

Let us instead consider the problem from a slightly different perspective. To wit, compositionality according to the above definition prescribes two main areas of investigation: *techniques* and *methods*.

- The *technical* aspects of compositionality consist in the study of the rules of composition for the chosen formal language; in particular, we are interested in temporal logics, and real-time extensions of them.

  Even more concretely, studying the technical aspects of compositionality means formulating a number of *compositional inference rules*. These are logic inference rule whose hypotheses state some facts about modules of the system in isolation, and whose conclusions state some facts about the composite system working as a whole.

We remark that the technical aspect is the aspect that has been more deeply studied in the literature, in particular in the form of compositional inference rules.

- The *methodological* aspects of compositionality consist in the study of how compositional techniques can be successfully applied *in practice* to (models of) modular systems.

  Therefore, a compositional method should suggest how to organize the local specifications of each module so that they can be composed through a compositional inference rule. Moreover, it should also describe a general "attitude" in formalizing and reasoning about composite systems, one which helps in managing the complexity of formalizing systems composed of several modules, and of the resulting correctness proofs.

  Some parts of the methodological aspects of compositionality have been tackled in [FRMM06], where we described general guidelines to follow in modularizing a large specification, and in organizing the correctness proofs. In this paper, we focus on the aspects more related to the choice of the compositional rules to be used in the correctness proofs, and on the interplay between such methodological aspects and the technical aspects.

1.2 Compositionality for the Common Man

Let us now summarize some important aspects of the compositional problem. We point out what are some typical features that are often associated with several of the approaches to compositionality pursued so far [Fur05], with respect to the aspects we elicit.[1] Correspondingly, we lay out the general guidelines for our approach to compositionality.

- The first aspect is the "compose vs. decompose". In the literature, we find both *decompositional* and *compositional* approaches. Decompositional approaches are basically refinement techniques where a single module is refined into the composition of several smaller modules; compositional techniques ensure that the composition of the refined modules retains some properties of the original single module. Compositional approaches, instead, start from a set of individual modules and provide ways of inferring properties about the composition of these individual modules into a larger system.

  In this paper, we are mostly interested in the *compositional* approach. First, a decompositional approach would by definition involve refinement techniques, and thus aspects that are simply out of the scope of this paper, as they would require a different framing and a different focus. Second, compositionality allows one to *reuse* the specifications of single modules in different systems that instantiate the modules. Therefore, we argue

[1] Another interesting critique to compositionality is the one by Lamport [Lam98]. Indeed, some of the points raised by Lamport are also relevant to the content of this paper.

that compositionality is mostly useful in such contexts, where the possible additional burden introduced by the construction of a compositional specification [Lam98] is then amortized by reusing the same specification in multiple proofs for multiple different systems.

- Most of the literature deals with the *technical* aspects of compositionality, whereas *methodological* aspects are often neglected.

  In this paper (as well as in [FRMM06]) we try to develop not just some compositional inference rules, but also a general methodological approach to the issue of compositional verification. We already pointed out in [FRMM06] how the traditional practices of modularization, abstraction, and information hiding can be successfully applied also to the formal specification and verification of modular systems. This is integrated with a suitable notation such as TRIO, which gives to the user suitable constructs to implement these practices.

- Most works in the literature provide one unique compositional inference rule, with the tacit assumptions that all facts about the composition of modules should be inferred using the rule. This is often not very practical: the verification process is often a complex task that requires much flexibility on how it should be carried out. Therefore, constraining it to develop according to some prescribed "recipe" may lead to needless complications.

  On the contrary, we provide *several different rules*, we give a possible taxonomy according to which different rules can be categorized, and we discuss how the dimensions of the taxonomy mirror aspects of real systems. Our goal is not only to provide a variety of rules among which the user can choose, but also — and most importantly — to illustrate the attitude to reverse the usual approach to applying compositionality. Rather than having a rule (or a set of rules) and describing the system in a way which is amenable to the use of the rule, we suggest to first formalize the system according to the general modularization methods we discussed in the previous point. Then, after the system is formalized, one tries to find a suitable rule that fits the system in question; possibly, the user should also *develop* a new rule, based on the "lessons" learned from the existing rules and the formalization of the system, to suitably fit the system itself and its verification goals. Therefore, our catalog of compositional rules is also meant to provide an illustration on how compositional rules can be built and modified according to one's needs. In Section 6 we will provide an example of how this can be done in practice.

- Since, as we discussed in the previous point, one single rule is often assumed to be used to handle all possible cases, its *completeness* is usually regarded as an important feature [NT00, Fur05]. However, as we also discussed in [FRMM06], compositional rules are complete in a *trivial* way, with respect to compositionality. This means that there exist global properties which — although provable with the rule in question thanks to

completeness — cannot be usefully "split" among properties local to different modules, but require a trivial partitioning of the system where one module behaves as the whole system.

Therefore, we claim that completeness is not an non-renounceable feature of a compositional rule. More specifically, completeness must not be traded off with simplicity of a compositional rule or with its practical applicability. Compositional reasoning cannot ease all problems of modular verification: it is important that it solves effectively some of them, those likely to happen in practice. Therefore, in this paper we develop both complete and incomplete rules with equal interest, and we try to understand what features may render a rule (in)complete.

- Most, if not all, of the compositional rules in the literature are *circular*, i.e., informally, the property of some module references circularly (in general, indirectly) to itself. Circular reasoning is clearly unsound, in general, so developing circular rules is a much more challenging task than developing non-circular ones.

  Nonetheless, we claim that circularity is not always needed in practice, and indeed most compositional reasoning can be carried out with non-compositional rules. Formulating non-circular inference rules is technically not very challenging, as they are simple consequences of the basic logic rules of conjunction and implication [Lam98]. Nonetheless, we develop some of them in this paper, together with circular ones. In fact, we believe that the technical challenges involved in developing the latter rules should not make us favor them over non-circular rules. Realizing what is the role of non-circular rules in practice should help framing the compositionality problem from the right perspective, and with the right objectives, that is practical usefulness rather than purely technical appeal.

We conclude this section by stressing that we advocate a "lightweight" approach to compositionality. The approach is driven by the specifics of the system we are considering. Method comes first, as it gives general guidelines on how the system should be specified and organized. Then, according to the peculiar verification needs that arise, one should be able to pick a suitable technical solution to it, possibly developing new ones according to the problem at hand.

2 A Taxonomy of Compositional Rules

This section defines the notion of inference rule in general, and of compositional inference rule in particular. Then, it introduces some dimensions along which a variety of compositional inference rules can be characterized; in doing so, it tries to link the dimensions of the taxonomy to features of "real" systems they mirror.

2.1 Definition of (Compositional) Inference Rule

2.1.1 Inference Rules

Inference rules. Let us start by defining what is an inference rule, generally speaking. An *inference rule* [Men97] is defined by a relation between pairs of sets of formulas: $\Pi = \{\pi_1, \ldots, \pi_m\}$ and $\Xi = \{\xi_1, \ldots, \xi_n\}$ and usually denoted by:

$$\frac{\Pi = \{\pi_1, \ldots, \pi_m\}}{\Xi = \{\xi_1, \ldots, \xi_n\}}$$

Elements of Π are called the *premises* (or *antecedents*, or *hypotheses*) of the inference rule, while elements of Ξ are called the *conclusions* (or *consequents*, or *theses*). Π and Ξ are usually thought of as ordered sets, as our notation suggests.

Intuitively, whenever a set of premises $\widetilde{\Pi}$ and a set of conclusions $\widetilde{\Xi}$ belong to the relation defined by the inference rule, it means that the truth of $\widetilde{\Pi}$ entails the truth of $\widetilde{\Xi}$.

Soundness and completeness. An inference rule is *sound* (or *correct*) if: for any pair $\widetilde{\Pi}, \widetilde{\Xi}$ that belongs to the relation defined by the rule, if the premise $\widetilde{\Pi}$ is true, then the conclusion $\widetilde{\Xi}$ is also true.

We usually only care about sound inference rules, since the intuitive purpose behind defining and using an inference rule is exactly that of deducing true facts from other known true facts. Therefore, if we establish a proposition of the following form:

Proposition 1 (Soundness of an inference rule). *If:*

1. π_1
2. π_2

   $\vdots$

m. π_m

then:

1. ξ_1
2. ξ_2

   $\vdots$

n. ξ_n

we have in practice proved the soundness of the corresponding inference rule.

On the other hand, an inference rule is *complete* if: for any conclusion $\widetilde{\Xi}$ which is true, there is some premise $\widetilde{\Pi}$ such that the pair $\widetilde{\Pi}, \widetilde{\Xi}$ belongs to the relation defined by the rule, and $\widetilde{\Pi}$ is true. In other words, a complete rule allows the proof of any true fact.

If our reference logic language is an intrinsically incomplete one — as it is the case with TRIO, which includes arithmetic — then we cannot have a fully complete inference rule, that is one which is complete for any formula in the language. In such cases, when we speak about completeness we actually mean *relative completeness* [Coo78], that is, in a nutshell, completeness with respect to an oracle for the truth of arithmetic formulas.

While soundness is an indispensable property for an inference rule, completeness (even relative one) is a feature which is not strictly required, but may be desirable in some contexts.

2.1.2 Compositional Inference Rules

Let us now focus on *compositional* inference rules. The full practical significance of these rules is apparent with the compositional methodology presented in [FRMM06]. However, while introducing the technicalities necessary to discuss compositional inference rules, in this section we also try to illustrate the meaning of the inference rules, and their overall rationale.

Generally speaking, the goal of a compositional inference rule is to provide a way to prove some facts about the *composition* of some modules, from other known facts about each of the modules. Let us assume that we are dealing with systems made of some $N \geq 2$ modules.

Local rely/guarantee specifications. The starting point for the application of a compositional inference rule is given by a set of *local specifications*, one for each of the modules in the system. This means that we have some formalization of the elementary significant behavior of each module, in terms of a formula.

In general, the local specifications could be any kind of formula. In practice, a very natural way to specify a module is according to the *rely/guarantee paradigm*.With the rely/guarantee paradigm one makes some assumptions on the behavior of the *environment* of the module under specification, and he/she links these assumptions to the guaranteed behavior of the module itself: the module behaves as expected only if its environment does the same.

In order to formalize this in the inference rules, we associate to each module $1 \leq i \leq N$ two formulas: E_i and M_i. E_i is called the *local assumption* of module i, while M_i is called the *local guarantee* of module i. Therefore, the local specification of module i consists of some formula linking E_i to M_i.

In the simplest case, this link consists in logical implication: $E_i \Rightarrow M_i$ is a local specification describing a module that respects formula M_i as long as its environment respects formula E_i. For instance, let us consider a simple adder module which takes an integer a and returns it increased by 2 units: $b = a + 2$. A rely/guarantee local specification for such a module could be the following: if a is an even number, then so is b.

More generally — and in particular when dealing with the description of timing behavior — the link between assumption and guarantee is defined through a binary operator which we generically denote as $\succ$. We call it *compositional*

operator. So, implication is a simple example of compositional operator, but in general different inference rules use different compositional operators. We will provide several examples of compositional operators in the following sections.

Global specifications. As we said, the goal of a compositional inference rule is to infer some facts about the composition of modules. In the most general case, the N modules are composed into an open system, which can therefore also be characterized by a global rely/guarantee specification. The global rely/guarantee specification can be of the same form of the local specifications, or of a different form.

Anyway, we characterize it through two formulas called *global assumption* and *global guarantee*, denoted as E and M, respectively. As it is simple to understand, they play similar roles as the local formulas, but with reference to the whole system: E formalizes assumptions about the expected behavior of the system's environment, and M formalizes the guaranteed behavior of the composite system.

Parts of a compositional inference rule. Let us now illustrate the typical components of a compositional inference rules, through a simple example.

Recall the module "adding 2" described above; let us introduce a predicate $\mathsf{eve}(k)$ to denote the fact that k is even. Thus, the rely/guarantee specification of this module can take the form $\mathsf{eve}(a) \Rightarrow \mathsf{eve}(b)$. Let us now connect two instances $1, 2$ of this module such that the input of module 1 is the output of module 2, and *vice versa*. The rely/guarantee specification of module 2 would then be $\mathsf{eve}(b) \Rightarrow \mathsf{eve}(a)$. We now illustrate the components of compositional inference rules through this simple system.

Global specification. Let us start from the conclusion of the inference rule. In a nutshell, it consists of the *global specification*: $E \succ M$. In other words, the overall goal of the inference rule is to deduce the truth of the global specification.

For our simple example, let us assume that the global guarantee is $\mathsf{eve}(a) \wedge \mathsf{eve}(b)$, that is both values exchanged by the two modules are even numbers. Since the system is a closed one, we need not care about a global assumption; equivalently, we assume E to be identically true.

Local specifications. Let us now consider the formulas that appear among the antecedents of the rule. Obviously, we have the local specifications, as the ultimate goal of the rule is to infer the truth of the global specification from the truth of all the local specifications. Therefore all the local specifications $E_i \succ M_i$ for all $1 \leq i \leq N$ appear as antecedents of a compositional inference rule.

In our example, we have both $\mathsf{eve}(a) \Rightarrow \mathsf{eve}(b)$ and $\mathsf{eve}(b) \Rightarrow \mathsf{eve}(a)$ among the antecedents.

Assumption discharging formulas. Since we are considering local specifications in rely/guarantee form, we cannot say anything about the truth of the local guarantees unless we are able to prove the truth of the local assumptions. In other words, in order to use local guarantees as descriptions of the modules' behavior, we must first show that the modules' environments behave as required by the guarantees. But the system is made by the composition of the various modules; therefore, for each module i, the other modules constitute i's environment.

This suggests to introduce an hypothesis in the inference rule that allows one to show that the truth of the environment assumptions of the various modules follows logically from that of the modules' guarantees. Since the verb "discharge" is typically used to mean "prove an assumption", we call these hypotheses (one for each module) *assumption discharging formulas*. They are in the form $M_1 \wedge M_2 \wedge \cdots M_N \Rightarrow E_i$ for each module $1 \leq i \leq N$; notice that, in the most general case, we allow one to use any module j, including itself, to discharge the assumption. Later, we discuss how this gives rise to different kinds of rules.

The global environment E may take part in determining the properties of the environment of each module i. Therefore, in the most general case it may be required to use the global assumption E to discharge some local assumption E_i: the discharging formula becomes $E \wedge M_1 \wedge M_2 \wedge \cdots M_N \Rightarrow E_i$ in this case. In other words, the local environment of module i "inherits" some properties of the global environment.

In our running example, notice that the assumption of module 1 is the guarantee of module 2, and *vice versa*. Therefore, the assumption discharging formula for module 1 (resp. 2) coincides with the local specification of module 2 (resp. 1).

Circularity breaking and initialization. As it should be clear even from our trivial example, rely/guarantee compositional inference rules may involve a *circularity* between assumptions and guarantees of some modules. In our running example, this is apparent by the fact that each module relies on the other module's guarantee in order to behave correctly (i.e., according to its own guarantee). Circular reasoning is in general unsound, therefore we need to introduce an additional hypothesis in the inference rule to make the deduction sound. We call this additional hypothesis *initialization condition*.

Even if the rule does not involve a circularity, some form of initialization condition is required as "starting point" to apply the chain of inferences implied by the local specifications. In other words, if we have no circularities between specifications and dischargings, we nonetheless have to start from the truth of some local assumption or guarantee to exploit the rely/guarantee specifications. Therefore, the existence of some initialization condition does not depend on the circularity of the rule.

The actual form of the initialization condition can vary a lot from inference rule to inference rule, and it might even be implicit (i.e., subsumed by some

assumptions on the semantics of the language). In our simple example, it might simply consists of an assertion about a being even *a priori*: $\mathsf{eve}(a)$. This clearly breaks the circularity as it holds regardless of any assumption. In fact, in this case the application of the inference rule amounts to the application of the well-known *modus ponens* logic inference rule [Men97]: from the truth of $\mathsf{eve}(a)$ and $\mathsf{eve}(a) \Rightarrow \mathsf{eve}(b)$ we deduce $\mathsf{eve}(b)$, so overall we have $\mathsf{eve}(a) \wedge \mathsf{eve}(b)$. Although compositional inference rules will be much more complicated than this, being able to deal with temporal properties, our trivial example shows that the very basics of compositional reasoning are grounded in basic logic inference.

Global implementation. Combining the local specifications with the dischargings, and through the initialization condition, one should be able to infer the truth of some (or all) of the local guarantee formulas M_i's. One final ingredient is usually required: an hypothesis should link the truth of the local guarantees to that of the global guarantee; the latter is in fact the ultimate goal. Therefore, in general we have a formula $M_1 \wedge M_2 \wedge \cdots M_N \Rightarrow M$ among the antecedents of a compositional inference rule. We call this hypothesis *global implementation*.

In our simple running example, the global implementation is trivially true, as $M = M_1 \wedge M_2 = \mathsf{eve}(b) \wedge \mathsf{eve}(a)$.

As with the local dischargings, it may be necessary to exploit the assumptions about the global environment to infer the truth of the global guarantee. Therefore, a more general form of the global implementation formula is: $E \wedge M_1 \wedge M_2 \wedge \cdots M_N \Rightarrow M$

Time and initialization. Although the notions of compositionality and compositional inference rules is grounded in basic logic languages, our interest is about compositionality for temporal logic formalisms. Therefore, our compositional inference rules must deal with temporal descriptions of the behavior of systems.

In such cases, it happens often that the rely/guarantee behavior of a module involves a temporal relationship between the assumption and the guarantee of the module. For instance, consider a functional module that takes some input and returns an output after some time T. Our running example may be refined to have such a temporal behavior: the local specification would now be expressible through the TRIO formula $\mathsf{eve}(a) \Rightarrow \mathrm{Futr}(\mathsf{eve}(b), T)$.

In accordance, we may need to restrict the "temporal scope" of the antecedents, or of the consequent, or both. Formally, this can be achieved by making each formula of the inference rule the consequent of an implication, whose antecedent is a (temporal) formula that holds if and only if we are within the desired "temporal scope". We denote such antecedents with the letter I. In general, we may have a different I for each hypothesis or conclusion of the inference rule. We denote them by adding subscripts and superscripts according to the formula to which that I is to be applied.

Returning one more time to our simple example, let us assume that both

modules have a predicate s which is true exactly when the system is started. Then, for instance, the local rely/guarantee specification for module 1 would now be in the form $\mathsf{s} \Rightarrow (\mathsf{eve}(a) \Rightarrow \mathrm{Futr}(\mathsf{eve}(b), T))$, and similarly for the other formulas. The initialization condition would be naturally expressed by specifying that at when the system is started, a is even: $\mathsf{s} \Rightarrow \mathsf{eve}(a)$.

More generally, we observe that the need for a notion of "initialization" (or "system start") arises especially when one uses bi-infinite time domains (such as $\mathbb{R}$ and $\mathbb{Z}$, which are the default choices for TRIO); otherwise, with mono-infinite domains, the infimum element of the time domain conventionally denotes an absolute time instant at which the system starts. Indeed, most compositional inference rules in the literature [Fur05] adopt a mono-infinite time domain, and therefore do not use initialization predicates (at least in the form we have just presented).

General form of a rely/guarantee compositional inference rule. According to our explanations, the form of a rely/guarantee compositional inference rule is summarized in Table 1.

$\bigwedge_{1 \le i \le N} (I_i^{\mathrm{sp}} \Rightarrow (E_i \succ M_i))$	local specifications
$\bigwedge_{1 \le i \le N} \left(I_i^{\mathrm{dsc}} \Rightarrow \left(E \wedge \bigwedge_j M_j \Rightarrow E_i \right) \right)$	(assumption) discharging formulas
$I^{\mathrm{imp}} \Rightarrow \left(E \wedge \bigwedge_j M_j \Rightarrow M \right)$	global implementation
$I^{\mathrm{init}} \Rightarrow \mathbf{init}$	initialization (condition)
$I^{\mathrm{glb}} \Rightarrow E \succ M$	global specification

Table 1: The components of a generic compositional inference rule.

The observant reader may argue that the general form is not really a specialization of a generic inference rule, as the initialization condition allows any formula among the antecedents, and the global specification allows any formula as consequent through an adequate choice of compositional operator and of E, M. This is technically true, but the goal of the above schema is not to strictly characterize all compositional rules, but simply to give an (partly intuitive) idea of what a form a compositional rule should have, and to serve as a yardstick for the following developments, taxonomy, and explanations.

Syntactically-restricted completeness. For compositional inference rules one is usually interested in a notion of completeness that is slightly narrower than the one introduced previously for general inference rules. Namely, it is common to consider only formulas expressible as a global specification, that

is through a compositional operator. Then, a compositional inference rule is complete — according to this syntactically-restricted notion of completeness — if: for any conclusion $\widetilde{\xi}$ that is expressible as $\widetilde{\xi} \equiv I^{\mathrm{glb}} \Rightarrow E \succ M$ and is true, there is some set of premises $\widetilde{\Pi}$ such that the pair $\widetilde{\Pi}, \widetilde{\xi}$ belongs to the relation defined by the rule, and $\widetilde{\Pi}$ is true. The difference between this definition and the more general one given above is in the fact that we now consider only formulas expressible according to the syntactic restrictions given by the statement of the compositional inference rule.

Notice that the syntactic restrictions may not impact semantic expressiveness at all: in such cases, when any formula can be written as a global specification, the two definitions coincide. This happens often in practice, as we will see with the compositional operators that we will introduce in the following sections. In the most general case, however, such a narrower notion of completeness is adopted since it is totally irrelevant to discuss the completeness of a compositional inference rule for formulas that are not expressible as specifications of a modular systems: in such cases compositionality is useless, and one should resort to traditional "flat" inference rules and standard deduction.

2.2 Dimensions for a Compositional Taxonomy

This section considers some common choices according to which the general compositional inference rule of Table 1 is instantiated into actual rules, and classifies the dimensions along which these instantiations are made. We stress again the fact that we do not aim at exhaustiveness — which would probably be of little practical interest anyway — but simply at giving a useful taxonomy which may guide the development of some actual rules.

Let us first list the dimensions we consider; afterward, we comment them.

- First, we distinguish between *circular* and *non-circular* (or *circle-free*) rules. A rule is circular if there are circularities between the discharging formulas and the local specifications: in other words, when the assumption of some module is discharged (directly or indirectly) through the guarantee of the same module.

- A *self-discharging* rule is one where the guarantee M_i of some module i appears on the left-hand side of the implication that discharges the assumption E_i of the same module. Notice that this is a different notion than circularity, as we discuss below.

- A rule is *fully compositional* when the global assumption E does not appear in any of the discharging formulas.

- A rule is *globally initialized* iff all of the formulas I_i^{sp}, I_i^{dsc}, I^{imp} are identically true; in other words, the truth of the corresponding formulas is not dependent of the occurrence of the start predicate. Otherwise, we call the rule *locally initialized*.

- Finally, another dimension along which to classify a compositional rule is the choice of the *compositional operator*: every choice of the compositional operator gives rise to a different "flavor" of inference rule.

Let us now give an idea of how the various features of inference rules may be related those of the systems whose correctness they are used to prove.

Circular vs. non-circular rules. This is a major feature according to which compositional inference rules can be classified. Circular rules are needed in practice when modeling systems whose components circularly rely on one another to function correctly when put together. For instance, the simple example of the two "adding 2" modules (shown previously) exhibits an obvious circularity, manifest by the fact that the guarantee $\mathsf{eve}(b)$ of module 1 coincides exactly with the assumption of module 2, and *vice versa*.

Indeed, our experience has shown that this is often *not* the case: designing a system whose components circularly rely on one another is most of the times an overly complicated choice, so it is not likely to be taken often. Of course, exceptions exist: for instance the dining philosophers example of [FRMM06], and the BitTorrent example of Section 6, are examples of systems that exhibit circularities. However, they have been chosen especially with this criterion in mind: to show circular inference rule in action. Therefore, in the remainder we also discuss non-circular inference rules, and argue that these are very commonly used in practice.

In general, the soundness of non-circular rules relies on very minimal (and simple) assumptions about the various elements of the rule. Indeed, their correctness is often a simple consequence of the elementary properties of the implication and of the conjunction, as we will show in Section 3.

On the other hand, showing the soundness of circular rules is in general more involved, in particular when dealing with the specification of temporal properties. In fact, circular reasoning is in general not sound, so that one has to show that there is some circularity breaking mechanism that resolves the issue (this is usually a result of the interplay between the initialization conditions and the properties of the compositional operator). This probably explains why most, if not all, the compositional rules that have been studied in the literature are circular [Fur05]: demonstrating their soundness has been deemed a sufficiently challenging problem to be of interest.[2]

Self-discharging rules. Whereas circularity is mostly a semantic property, self-discharging rules have a clear *syntactic* characterization.

In order to motivate the use of self-discharging rules, let us sketch a very simple example of a timed system where self-discharging is the natural way to go. For the sake of simplicity let us assume a discrete time domain, and let us just consider one single thermostat module.[3] Now, let us assume that the

[2] Indeed, Abadi and Lamport [AL95] call "strong" inference rules which are circular.

[3] Clearly, this would make us lose the point of compositional reasoning, but we do this for illustration purposes only.

thermostat is capable of influencing the temperature of a room. In particular, whenever the temperature is within a given "safe" range, the device is capable of guaranteeing that the temperature is also within the range in the next time instant. Thus, if we introduce a time-dependent predicate ok to indicate that the temperature is within the desired range, the rely/guarantee behavior of the module may be described with the formula $\mathsf{ok} \Rightarrow \mathrm{NowOn}(\mathsf{ok})$. Notice that in this case $E = M = \mathsf{ok}$ for the module. Therefore, the tautology $\mathsf{ok} \Rightarrow \mathsf{ok}$ would naturally be a self-discharging assumption discharging formula for the system.

As this little example suggests, self-discharging rules may be needed in describing the behavior of modules with some form of feedback. In such modules, the functionality itself of the module (represented by the guarantee) may contribute directly, together with the other modules in the system, to make the environment of the module behave as in the assumption. On the other hand, when there is no feedback, then the module relies directly solely on the other modules to have a "correct" environment, while its actions contribute to guaranteeing that the other modules' environments are "correct". Another, more realistic, example of a system where self-discharging is useful is the controlled reservoir example presented in [Fur03, Chap. 7].

Let us remark that self-discharging and circular are two distinct properties of a rule. More precisely, a self-discharging rule is always also a circular rule: this is apparent from our definition of circular rules. On the other hand, a rule may be circular without being self-discharging: in this case the circularity between assumption and guarantee of some module is not direct (i.e., within the same implication), but it requires some levels of indirection. The canonical example is that of two modules $1, 2$ whose local specifications are $E_1 \Rightarrow M_1$ and $E_2 \Rightarrow M_2$, and whose discharging formulas are $M_2 \Rightarrow E_1$ and $M_1 \Rightarrow E_2$. Then, the discharging formulas show that the rule is not self-discharging. However, the discharging of assumption E_1 (resp. E_2) relies on the guarantee M_2 (resp. M_1), which in turn depends on the assumption E_2 (resp. E_1) to hold, which relies on the guarantee M_1 (resp. M_2); thus all in all E_1 (resp. E_2) circularly depends on M_1 (resp. M_2) to be discharged.

Fully compositional rules. According to the stricter definition of compositionality, which we presented in Section 1.1, in a compositional rule the truth of some global fact should be inferred "on the basis of its constituent components only". This suggests that the discharging of the assumptions should not depend on anything which is "outside" the system, such as the global assumption E. This is why we call a rule "fully compositional" if this is the case.

Closed systems — such as our "adding 2" example of the previous section — are a particularly relevant class of systems for which fully compositional rules clearly suffice. In fact, closed systems do not communicate with any external environment, so that the global assumption E is identically equal to true. In general, controlled systems are usually modeled as closed systems, as the physical system under control and the controller constitute each other's environment. From this viewpoint, modeling a complete system in a rely/guarantee

style often results in a closed system, as one endeavors to include a model of the environment itself, according to the control standard paradigm.

On the contrary, when the modules are not "autonomous" in determining one other's environment, but they rely on some additional property of the global environment, one needs non-fully compositional rules. Notice also that an open system does not necessarily require a non-fully compositional inference rule, as the global assumption may be required only in the global implementation or in the global specification.

In the literature, several compositional inference rules are non-fully compositional [Fur05]. This is often driven more by an effort toward generality (and completeness), rather than by specific requirements of the systems of interests. In our analysis, we will consider both fully and non-fully compositional rules.

Globally vs. locally initialized rules. As we discussed in the previous Section 1.1, initialization predicates I are needed only when dealing with specifications describing the *temporal behavior* of a system. This is the focus of the present work, thus the notion of globally and locally initialized rules is needed.

If a system has a notion of "start" or initialization, then it may be that it behaves differently before than after it has started. Let us illustrate this on the thermostat example: let us postulate that the module can be started, and let us model this fact through a predicate s becoming true. Then, it may be that the rely/guarantee behavior $\mathsf{ok} \Rightarrow \mathrm{NowOn}(\mathsf{ok})$ holds only after the system has started, that is after s has been true at least once (for simplicity, let us assume that further starts are ignored). Then, the initialization predicate I^{sp} for the local specification should be $I^{\mathrm{sp}} = \mathrm{SomP}(\mathsf{s})$, and the full local specification would become: $\mathrm{SomP}(\mathsf{s}) \Rightarrow (\mathsf{ok} \Rightarrow \mathrm{NowOn}(\mathsf{ok}))$. Notice that this would also probably require to introduce the same initialization predicate for the other hypotheses and for the conclusion of the inference rule we would like to use. This corresponds to completely disregarding the behavior of the system before it has started.

Locally initialized rules subsume globally initialized ones, as in globally initialized rules one does not have to deal with initialization conditions except that in the initialization formula. Nonetheless, globally initialized rules usually present the local specifications in a simpler form, for modules whose rely/guarantee behavior does not depend on any initialization. Finally, notice that, in globally initialized rules, the initialization formula is only required for circularity breaking, in order to guarantee the soundness of the rule, but it does not influence the behavior of the single modules or the way they interact.

Plan of the following sections. The following Section 3 presents some compositional inference rules that are non-circular. Next, Section 4 presents several circular compositional inference rules, according to the taxonomy we have just introduced. Notice that Sections 3 and 4 present the rules in the simple case of systems composed on just two modules, in order for the presentation to be as terse as possible. Then, Section 5 shows how the previously introduced inference

rules can be generalized to handle the case of $N > 2$ modules as well.

3 Non-Circular Compositional Inference Rules

This section presents two non-circular compositional inference rules, and proves their soundness and completeness.

3.1 Non-Circular Inference Rules

For the sake of simplicity, let us first consider simple systems consisting of two modules only. The extension to the general case of $N \geq 2$ modules will be discussed separately, in Section 5.

First of all, the non-circular rules that we present here use regular implication as a compositional operator, as they are simple consequences of the logical rules of implication and conjunction. Even more generally, we are able to present their proofs in terms of two generic logical connectives denoted as $\sqsubseteq$ and $\sqcap$. $\sqsubseteq$ represents any operator which is an order relation (i.e., it is reflexive, anti-symmetric and transitive), while the $\sqcap$ connector represents an associative, commutative and idempotent operation which respects the order relation (i.e., such that if $A \sqsubseteq C$ and $B \sqsubseteq D$, then also $A \sqcap B \sqsubseteq C \sqcap D$ for any A, B, C, D). It is immediate to realize that logical implication $\Rightarrow$ and conjunction $\wedge$ respectively satisfy such properties.

Therefore, we have the two following proposition, that establish the soundness of the corresponding two compositional inference rules.

Proposition 2 (Non-Circular Rely/Guarantee Inference Rule 1). *If:*

1. $E_2 \sqsubseteq M_2$
2. $E \sqcap M_1 \sqsubseteq E_2$
3. $E \sqcap M_1 \sqcap M_2 \sqsubseteq M$
4. $E \sqsubseteq M_1$

then, $E \sqsubseteq M$

Proof. The following formal derivation proves the proposition.

$*$)	$E \sqcap M_1 \sqsubseteq M_2$	by 2, 1 and transitivity of $\sqsubseteq$
$**$)	$E \sqsubseteq E$	by reflexivity of $\sqsubseteq$
	$E \sqcap E \sqsubseteq M_1 \sqcap E$	by 4, the previous one and the order-preservation of $\sqcap$
	$E \sqsubseteq M_1 \sqcap E$	by the previous one and the idempotence of $\sqcap$
	$E \sqsubseteq E \sqcap M_1$	by the previous one and the commutativity of $\sqcap$
	$E \sqsubseteq M_2$	by the previous one, $*$) and the transitivity of $\sqsubseteq$
	$E \sqcap E \sqsubseteq M_1 \sqcap M_2$	by 4, the previous one, and the order-preservation of $\sqcap$
	$E \sqsubseteq M_1 \sqcap M_2$	by the previous one and the idempotence of $\sqcap$
	$E \sqcap E \sqsubseteq E \sqcap M_1 \sqcap M_2$	by $**$), the previous one and the order-preservation of $\sqcap$
	$E \sqsubseteq E \sqcap M_1 \sqcap M_2$	by the previous one and the idempotence of $\sqcap$
	$E \sqsubseteq M$	by the previous one, 3 and transitivity of $\sqsubseteq$

□

Notice that the rule of Proposition 2 is non fully compositional; instead, the following is.

Proposition 3 (Non-Circular Rely/Guarantee Inference Rule 2). *If:*

1. *(a)* $E_1 \sqsubseteq M_1$
   (b) $E_2 \sqsubseteq M_2$
2. $M_1 \sqsubseteq E_2$
3. $E \sqcap M_1 \sqcap M_2 \sqsubseteq M$
4. $E \sqsubseteq E_1$

then, $E \sqsubseteq M$

Proof. The proof is very similar to that of Proposition 2. In brief, just assume E holds; then, E_1 by 4, M_1 by 1a, E_2 by 2, M_2 by 1b, M by 3. □

We remark again that, however simple the two above non-circular rules may seem, they are those used in practice in a lot of formal reasoning about composite systems. Indeed, their simplicity and wide applicability show that what we call "compositional reasoning" is naturally embedded in mathematical logic reasoning [Lam98].

3.2 Completeness of the Non-Circular Rules

Let us show that the inference rules of Propositions 2 and 3 are relatively complete. We need an extra assumption on the set of formulas over which the order relation $\sqsubseteq$ is defined: we require that there exists a *maximum element* in the ordered set, indicated as $\top$, that is a formula such that for any formula P, $P \sqsubseteq \top$ is true.[4] Moreover, $\top$ must be an identity element for the $\sqcap$ operation, that is for any formula P, $P \sqcap \top = \top \sqcap P = P$.

Notice that these assumptions are obviously satisfied by the logical value *true*, with respect to the logical implication $\Rightarrow$ and conjunction $\wedge$ respectively.

Theorem 4 (Completeness of Non-Circular Inference Rules). *The non-circular inference rules in Proposition 2 and in Proposition 3 are relatively complete.*

Proof. Let us first consider Proposition 2. Assume $E \sqsubseteq M$ holds. Hence choose $M_1 = M$ and $E_2 = M_2 = \top$. Hypotheses 1 and 2 are trivially true because of the definition of $\top$ as maximum element of the order. Hypothesis 4 is $E \sqsubseteq M_1$, which follows from $E \sqsubseteq M$, $M \sqsubseteq M_1$ (since $M = M_1$) and transitivity of $\sqsubseteq$. Finally, hypothesis 3 is $E \sqcap M_1 \sqcap M_2 \sqsubseteq M$, which corresponds to $E \sqcap M \sqcap \top \sqsubseteq M$ because of the equivalences, and to $E \sqcap M \sqsubseteq M$ because of definition of identity. Now, since $E \sqsubseteq M$ and $M \sqsubseteq M$, the last hypothesis 3 also holds.

Let us now consider Proposition 3 and assume $E \sqsubseteq M$. If you choose $E_1 = E$, $M_1 = M$ and $M_2 = E_2 = \top$, conditions 1–4 correspond to:

[4] Recall that $P = Q$ iff $P \sqsubseteq Q$ and $Q \sqsubseteq P$

1. (a) $E \sqsubseteq M$, assumed true.
   (b) $\top \sqsubseteq \top$, trivially true.
2. $E \sqcap M \sqsubseteq \top$, trivially true.
3. $E \sqcap M \sqcap \top \sqsubseteq M$, equivalent to $E \sqsubseteq M$, assumed true.
4. $E \sqsubseteq E$, true by reflexivity of $\sqsubseteq$. □

3.3 Summary of Non-Circular Rules

Table 2 summarizes the two non-circular compositional inference rules we have presented above; for uniformity with the results of the following sections, we use implication and conjunction in place of the generic $\sqsubseteq$ and $\sqcap$ operators.

Rule 1 (Prop. 2)	**Rule 2 (Prop. 3)**
$E_2 \Rightarrow M_2$	$E_1 \Rightarrow M_1$, $E_2 \Rightarrow M_2$
$E \wedge M_1 \Rightarrow E_2$	$M_1 \Rightarrow E_2$
$E \wedge M_1 \wedge M_2 \Rightarrow M$	$E \wedge M_1 \wedge M_2 \Rightarrow M$
$E \Rightarrow M_1$	$E \Rightarrow E_1$
$E \Rightarrow M$	$E \Rightarrow M$

Table 2: Non-circular compositional inference rules for two modules.

4 Circular Compositional Inference Rules

This section presents several circular compositional inference rules, proves their soundness, and discusses their completeness. More precisely, we are going to start in Section 4.1 by introducing a compositional operator $\twoheadrightarrow$ and present fully and non-fully compositional rules that use this operator. Then, Section 4.2 explores variations of the $\twoheadrightarrow$ that may be more suited to model certain classes of systems; inference rules for these variations are derived from the analogous ones in Section 4.1. Finally, Section 4.3 studies the completeness of the previously introduced compositional rules.

4.1 The Time Progression Compositional Operator and Its Compositional Rules

4.1.1 The Time Progression Operator

Let us introduce a simple compositional operator that allows us to define sound circular inference rules. It is called *time progression* operator, and it is denoted by the symbol $\twoheadrightarrow$. Informally, $P \twoheadrightarrow Q$ is true for two time-dependent formulas P, Q if, whenever P has been true in the immediate past, then Q has also been

true in the immediate past, is true now, and will continue to hold for some time in the immediate future. This is formalized by the following definition.

$$P \twoheadrightarrow Q \equiv \begin{cases} \text{UpToNow}(P) \Rightarrow \text{UpToNow}(Q) \wedge Q \wedge \text{NowOn}(Q) & \text{if time is dense} \\ \text{UpToNow}(P) \Rightarrow \text{UpToNow}(Q) \wedge Q & \text{if time is discrete} \end{cases}$$

Thus, the rely/guarantee specifications in our inference rules are in the form $E \twoheadrightarrow M$. Notice that this is a reasonable way to express a rely/guarantee specification: in fact, we say that the behavior of the module in the immediate future is influenced only by the behavior of the environment in the immediate past, so that if the environment stops behaving correctly, then the module can also stop behaving correctly only after "a while".

From a technical viewpoint, notice that the semantics of the time progression operator allows us to build a sort of "temporal induction" over the timeline, where the inductive step is allowed by the $Q \wedge \text{NowOn}(Q)$ part that "makes time progress" past the current instant (hence, the name). This will become apparent in the soundness proofs of the inference rules that use this operator.

Finally, let us also remark that a different operator was also named *time progression* and denoted with the same symbol in [FRMM06]. In Section 4.2 below we will show how the inference rule for that operator, also presented in [FRMM06], can be derived from those for the time progression operator introduced here.

In the remainder, recall that all TRIO formulas are implicitly universally quantified over time, that is closed with an Alw operator. In particular, this is the case for the hypotheses and conclusions of the rules that we are presenting.

4.1.2 Circular Inference Rules for the Time Progression Operator

Let us now present and prove the soundness of two circular inference rules for the time progression operator. The first rule is given in the following proposition; according to our taxonomy, it is a fully-compositional, non self-discharging, globally initialized, circular inference rule.

Let us also remark that, in practice, the initialization predicate S would be modeled in TRIO as a *unique event*, i.e., an event which happens exactly once in time. This is rendered by the two formulas $\text{Som}(S)$ and $S \Rightarrow \text{AlwP}(\neg S) \wedge \text{AlwF}(\neg S)$. However, the soundness of this inference rule, as well as that of the others that we present in the remainder of this section, do not depend on S being a unique event.

Proposition 5 (Rely/Guarantee Circular Inference Rule 1). *If:*

1. *(a)* $E_1 \twoheadrightarrow M_1$
   (b) $E_2 \twoheadrightarrow M_2$
2. *(a)* $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$

3. $M_1 \wedge M_2 \Rightarrow M$

4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$*, for some* $i \in \{1, 2\}$

then $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$.

Proof for dense time domains. Let t be the current instant, and let us assume that $\mathrm{SomP_i}(S)$ holds at t; thus, let $t' \leq t$ be (any) instant at which S held. We have to show that $E \twoheadrightarrow M$ holds at t.

First of all, let us prove that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ hold at t. To this end, we know that at t' we have $\mathrm{UpToNow}(E_i)$ or $\mathrm{UpToNow}(M_i)$ for some i, because of (4). Let us assume that $\mathrm{UpToNow}(E_i)$; this is without loss of generality, as if $\mathrm{UpToNow}(M_i)$, then (2) lets us conclude immediately that also $\mathrm{UpToNow}(E_{\hat{i}})$ at t', where we adopt the convention of denoting by $\hat{i}$ the "other" index $\{1, 2\} \setminus \{i\} \ni \hat{i}$. Then, let us consider the consequences of (1) at t': since $\mathrm{UpToNow}(E_i)$, then $\mathrm{UpToNow}(M_i)$, M_i, and $\mathrm{NowOn}(M_i)$ also at t'. Thus, let us consider (1) and (2) again and see that also $\mathrm{UpToNow}(E_{\hat{i}})$, $\mathrm{UpToNow}(M_{\hat{i}})$, $E_{\hat{i}} \wedge M_{\hat{i}}$, $\mathrm{NowOn}(E_{\hat{i}})$, and $\mathrm{NowOn}(M_{\hat{i}})$.

Therefore, we have that $M_1 \wedge M_2$ holds until some instant $t'' > t'$ in the future w.r.t. t'; in other words, we can say that $\mathrm{UpToNow}(M_1 \wedge M_2)$ holds at t''. But then, $\mathrm{UpToNow}(E_1 \wedge M_2)$ holds at t'', because of (2). It is not difficult to see that we can repeat everything that we did at t' again at t''. Thus, we will have a strictly monotonic sequence of points $t' < t'' < t''' < \cdots$ such that $M_1 \wedge M_2$ holds throughout.

Next, we have to show that the sequence gets (at least) until a point $u > t$. We show this by contradiction: assume to the contrary that $M_1 \wedge M_2$ holds until point $\overline{t} \leq t$ only; that is $\mathrm{NowOn}(M_1 \wedge M_2)$ is false at $\overline{t}$. Therefore, $\mathrm{UpToNow}(M_1 \wedge M_2)$ holds at $\overline{t}$. Then, (2) lets us deduce that also $\mathrm{UpToNow}(E_1 \wedge E_2)$ holds at $\overline{t}$. But then we consider (1) twice, once for module 1 and once for module 2, to conclude in particular that $\mathrm{NowOn}(M_1 \wedge M_2)$. This is in contradiction with the hypothesis that $M_1 \wedge M_2$ held up until $\overline{t}$, as expected.

All in all, we have shown that $M_1 \wedge M_2$ hold from before t' until some $u > t$. Therefore, in particular $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ all hold at t.

Finally, from (3) we easily infer that $\mathrm{UpToNow}(M) \wedge M \wedge \mathrm{NowOn}(M)$ at t. □

Proof for discrete time domains. We present a proof along the lines of the one for dense time domains. Alternatively, one could use induction to prove the same result for discrete time.

Let t be the current instant, and let us assume that $\mathrm{SomP_i}(S)$ holds at t; thus, let $t' \leq t$ be (any) instant at which S held. We have to show that $E \twoheadrightarrow M$ holds at t.

First of all, let us prove that $\mathrm{UpToNow}(M_1 \wedge M_2)$ and $M_1 \wedge M_2$ hold at t. To this end, we know that at t' we have $\mathrm{UpToNow}(E_i)$ or $\mathrm{UpToNow}(M_i)$ for some i, because of (4). Let us assume that $\mathrm{UpToNow}(E_i)$; this is without loss

of generality, as if UpToNow(M_i), then (2) lets us conclude immediately that also UpToNow($E_{\hat{i}}$) at t', where we adopt the convention of denoting by $\hat{i}$ the "other" index $\{1,2\} \setminus \{i\} \ni \hat{i}$. Then, let us consider the consequences of (1) at t': since UpToNow(E_i), then UpToNow(M_i) and M_i also at t'. Thus, let us consider (1) and (2) again and see that also UpToNow($E_{\hat{i}}$), UpToNow($M_{\hat{i}}$), $E_{\hat{i}}$, and $M_{\hat{i}}$.

Therefore, we have that $M_1 \wedge M_2$ holds until $t'' = t' + 1$ included; in other words, we can say that UpToNow($M_1 \wedge M_2$) holds at $t'' + 1$. But then, UpToNow($E_1 \wedge M_2$) holds at $t'' + 1$, because of (2). It is not difficult to see that we can repeat everything that we did at t' again at $t'' + 1$. Thus, we will have a strictly monotonic sequence of points $t' < t' + 1 < t' + 2 < \cdots$ such that $M_1 \wedge M_2$ holds throughout.

Next, we have to show that the sequence gets (at least) until a point $u > t$. We show this by contradiction: assume to the contrary that $M_1 \wedge M_2$ holds until point $\overline{t} \leq t$ included only; that is NowOn($M_1 \wedge M_2$) is false at $\overline{t}$. Therefore, UpToNow($M_1 \wedge M_2$) holds at $\overline{t} + 1$. Then, (2) lets us deduce that also UpToNow($E_1 \wedge E_2$) holds at $\overline{t} + 1$. But then we consider (1) twice, once for module 1 and once for module 2, to conclude in particular that $M_1 \wedge M_2$. This is in contradiction with the hypothesis that $M_1 \wedge M_2$ held up until $\overline{t}$, as expected.

All in all, we have shown that $M_1 \wedge M_2$ hold from $t' - 1$ until some $u > t$. Therefore, in particular UpToNow($M_1 \wedge M_2$) and $M_1 \wedge M_2$ all hold at t.

Finally, from (3) we easily infer that UpToNow(M) $\wedge M$ at t. □

Notice that in the above proofs we never actually introduced any fact about the value of E; in other words it is as if we had proved the conclusion true $\twoheadrightarrow M$. Indeed, in this rule — and in some of the following ones — the global assumption E is introduced in the conclusion only in conformance with the other rules; in other words we make no assumptions on the global environment. On the contrary, other rules actually require E to hold in order to be sound; this is the case, for instance, of the rules of Proposition 8 and Proposition 12, which we will present in the following sections.

The second rule we present is non fully compositional, self-discharging, globally initialized, and circular. Notice that our choice of compositional operator, combined with the fact that the rule is non fully compositional, requires to add a term in the conclusion that requires the global assumption E to hold "always in the past". Otherwise (i.e., if E was false somewhere in the past), it would be impossible, in general, to carry out the discharging of local assumptions, since the rule is non fully compositional. We prove the soundness of the rule only for dense time models; the proof for discrete time models can be developed along the same lines.

Proposition 6 (Rely/Guarantee Circular Inference Rule 2). *If:*

1. *(a)* $E_1 \twoheadrightarrow M_1$
   (b) $E_2 \twoheadrightarrow M_2$

2. $E \wedge M_1 \wedge M_2 \Rightarrow E_1 \wedge E_2$

3. $M_1 \wedge M_2 \Rightarrow M$

4. $S \Rightarrow \mathrm{UpToNow}(E_1 \wedge E_2) \vee \mathrm{UpToNow}(M_1 \wedge M_2)$

then $\mathrm{SomP_i}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$.

Proof for dense time domains. Let t be the current instant, and let us assume that $\mathrm{SomP_i}(S)$ and $\mathrm{AlwP_e}(E)$ hold at t. We show that $E \twoheadrightarrow M$ holds at t; to this end, let us first show that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ hold at t.

Let $t' \leq t$ be an instant at which S holds. From (4), let us assume that $\mathrm{UpToNow}(E_1 \wedge E_2)$ holds at t'. This is without loss of generality, since if $\mathrm{UpToNow}(M_1 \wedge M_2)$, then also $\mathrm{UpToNow}(E_1 \wedge E_2)$ at the same time, from (2) and the fact that $\mathrm{AlwP_e}(E)$ at $t \geq t'$.

Then, let us consider (1) at t'. We can infer that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ all hold at t'. Therefore, $M_1 \wedge M_2$ holds until some $t'' > t'$. If $t'' > t$, we are done proving the current goal; otherwise, we can iterate the reasoning and get to a new point $t''' > t''$.

The sequence of points $t' < t'' < t''' < \cdots$ must eventually reach a point $u > t$. The proof by contradiction goes just as in the case of the proof of Proposition 5.

So, finally we have that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ hold at t. The final step infers that also $\mathrm{UpToNow}(M)$, M, and $\mathrm{NowOn}(M)$ from (3). Thus, we have shown that $E \twoheadrightarrow M$ holds at t. □

4.1.3 Locally Initialized Circular Inference Rules

Let us now provide *locally initialized* variations of the two circular inference rules introduced above. They will be useful in discussing the completeness of the inference rules (see Section 4.3). Recall that, in locally initialized rules, the use of the predicate S "simulates" an origin on the time axis.

Notice that, in order to retain soundness, we have to introduce two small changes, other than local initializations. First, the initialization predicate in the conclusion excludes the current instant for S to occur; second, the initialization condition now requires an interval *in the future* where a local assumption or guarantee holds. This is required to ensure that all predicates are evaluated after the system has started, so that the locally initialized specification, dischargings, and global implementation formulas hold there.

Proposition 7 (Rely/Guarantee Circular Inference Rule 1(bis)). *If:*

1. *(a)* $\mathrm{SomP_e}(S) \Rightarrow (E_1 \twoheadrightarrow M_1)$
   (b) $\mathrm{SomP_e}(S) \Rightarrow (E_2 \twoheadrightarrow M_2)$
2. *(a)* $\mathrm{SomP_e}(S) \Rightarrow (M_1 \Rightarrow E_2)$
   (b) $\mathrm{SomP_e}(S) \Rightarrow (M_2 \Rightarrow E_1)$

3. $\mathrm{SomP_e}(S) \Rightarrow (M_1 \wedge M_2 \Rightarrow M)$

4. $S \Rightarrow \mathrm{NowOn}(E_i) \vee \mathrm{NowOn}(M_i)$, *for some* $i \in \{1, 2\}$

then $\mathrm{SomP_e}(S) \Rightarrow (E \twoheadrightarrow M)$.

Proof for dense time domains. Let t be the current instant, and let us assume that $\mathrm{SomP_e}(S)$ holds at t. To show that $E \twoheadrightarrow M$ holds at t, let us first show that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ hold at t.

Let $t' < t$ be an instant at which S holds. From (4), let us assume that $\mathrm{NowOn}(E_i)$ holds at t'. This is without loss of generality, since if $\mathrm{NowOn}(M_i)$, then also $\mathrm{NowOn}(E_{\hat{i}})$ at the same time, from (2) and the fact that in the right-neighborhood of t' $\mathrm{SomP_e}(S)$ holds. Since $\mathrm{NowOn}(E_i)$ at t', then equivalently $\mathrm{UpToNow}(E_i)$ holds at some $t' + \epsilon$, for some $\epsilon > 0$.

Then, let us consider (1) at $t' + \epsilon$. We can infer that $\mathrm{UpToNow}(M_i)$, M_i, and $\mathrm{NowOn}(M_i)$ all hold at $t' + \epsilon$. Then, also $\mathrm{UpToNow}(E_{\hat{i}})$ holds at $t' + \epsilon$ from (2); but then, by (1) again, also $\mathrm{UpToNow}(M_{\hat{i}})$, $M_{\hat{i}}$, and $\mathrm{NowOn}(M_{\hat{i}})$ all hold at $t' + \epsilon$.

At this point, the proof goes on exactly as for Proposition 5. In particular, we infer that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ are true at t, and we conclude that $E \twoheadrightarrow M$ by applying (3). □

Similar modifications are done to get a locally initialized circular inference rule analogous to that of Proposition 6. We omit the proof, which is however all similar to the one we have just provided.

Proposition 8 (Rely/Guarantee Circular Inference Rule 2(bis)). *If:*

1. *(a)* $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E_1 \twoheadrightarrow M_1)$
(b) $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E_2 \twoheadrightarrow M_2)$

2. $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \wedge M_1 \wedge M_2 \Rightarrow E_1 \wedge E_2)$

3. $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (M_1 \wedge M_2 \Rightarrow M)$

4. $S \Rightarrow \mathrm{NowOn}(E_1 \wedge E_2) \vee \mathrm{NowOn}(M_1 \wedge M_2)$

then $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$.

Finally, we present one more variation of the rule of Proposition 5. In this case, we write the global implementation formula in terms of the $\twoheadrightarrow$ operator, rather than using simple implication. We may argue that the original Proposition 5 is probably in a form which is more likely to be applicable in practice, whereas the following rule overloads one hypothesis with a large share of the complexity involved in compositional reasoning. Nonetheless, we will show that the following strengthening allows us to have a *complete* inference rule, and in any case it may be useful in practice with systems where the link between local and global guarantees is more involved than simple implication.

Proposition 9 (Rely/Guarantee Circular Inference Rule 1(ter)). *If:*

1. (a) $E_1 \twoheadrightarrow M_1$
   (b) $E_2 \twoheadrightarrow M_2$
2. (a) $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$
3. $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$, *for some* $i \in \{1, 2\}$

then $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$.

Proof. The proof is exactly the same as the one of Proposition 5 until we have established that $\mathrm{UpToNow}(M_1 \wedge M_2)$ and $M_1 \wedge M_2$ all hold at the current instant t. Then, if also $\mathrm{UpToNow}(E)$ then clearly we infer that $\mathrm{UpToNow}(M) \wedge M$ at t, from (3). This concludes the proof. □

4.2 Other Compositional Operators and Compositional Rules

This sections provides other circular compositional inference rules, exploring different compositional operators and other different features.

4.2.1 A Stronger Time Progression Operator

Let us provide a compositional inference rule very similar to that discussed and introduced in [FRMM06]. The rule exploits another compositional operator — also called "time progression" in [FRMM06] — that we denote with the symbol $\gg$ here. Its semantics is the following.

$$P \gg Q \equiv \begin{cases} \mathrm{AlwP_e}(P) \Rightarrow \mathrm{AlwP_i}(Q) \wedge \mathrm{NowOn}(Q) & \text{if time is dense} \\ \mathrm{AlwP_e}(P) \Rightarrow \mathrm{AlwP_i}(Q) & \text{if time is discrete} \end{cases}$$

Notice that $\gg$ is stronger than $\twoheadrightarrow$ in the following sense: if $P \twoheadrightarrow Q$ holds over the whole time axis, then $P \gg Q$ also holds, while the converse is not true, in general. In fact, if for instance P holds exactly over $(0, 10)$ and Q is always false, then it is true that $P \gg Q$ everywhere, as $\mathrm{AlwP_e}(P)$ is trivially false, but $P \twoheadrightarrow Q$ is false, as Q should hold over some set $(0, 10 + \epsilon)$, for some $\epsilon > 0$.

Despite $\gg$ being stronger than $\twoheadrightarrow$ in the above sense, we cannot develop a valid inference rule for the new operator by simply replacing the time progression operator in Proposition 5 or 6 with the new compositional operator. In fact, in general we also have to change the initialization condition with one which "triggers" the application of the new operator. In particular, the $\gg$ operator requires its left-hand argument to hold over always in the past from the current instant; therefore, we require that at the system initialization either some local assumption or some local guarantee are true always in the past. Thus, we obtain an inference rule which is a slight variation of the one presented in [FRMM06] for the $\gg$ operator.

Proposition 10 (Rely/Guarantee Circular Inference Rule 3). *If:*

1. *(a)* $E_1 \gg M_1$
   (b) $E_2 \gg M_2$
2. *(a)* $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$
3. $M_1 \wedge M_2 \Rightarrow M$
4. $S \Rightarrow \mathrm{AlwP_e}(E_i) \vee \mathrm{AlwP_e}(M_i)$, *for some* $i \in \{1, 2\}$

then $\mathrm{SomP_i}(S) \Rightarrow (E \gg M)$.

Proof sketch for dense time domains. Let us just sketch the beginning of the proof: the remainder is all similar to the previous proofs, as well as to that presented in [FRMM06].

Let t be the current instant and $t' \leq t$ be an instant at which S held. Then, without loss of generality we can assume that $\mathrm{AlwP_e}(E_i)$ at t'; otherwise, it would be $\mathrm{AlwP_e}(M_i)$, but then $\mathrm{AlwP_e}(E_{\hat{i}})$ would follow from (2). Then, from (1) we infer that $\mathrm{AlwP_i}(M_i)$ and $\mathrm{NowOn}(M_i)$ at t'. Moreover, from (2) and (1) it is simple to deduce that also $\mathrm{AlwP_i}(E_{\hat{i}} \wedge M_{\hat{i}})$ and $\mathrm{NowOn}(E_{\hat{i}} \wedge M_{\hat{i}})$. All in all we have "advanced" until some time $t'' > t'$.

Then, the proof proceeds by the usual non accumulation argument. When we have finally shown that $\mathrm{AlwP_i}(M_1 \wedge M_2) \wedge \mathrm{NowOn}(M_1 \wedge M_2)$ at t, then $E \gg M$ follows straight from (3). □

4.2.2 Implication as a Compositional Operator

As we also discussed elsewhere [FRMM06], there are basically two ways to make a circular inference rule sound. One relies on writing specifications using an *ad hoc* operator that allows one to "propagate" the validity of some predicate over time, thus allowing a sort of temporal induction; this is the way we have followed with the time progression operator or, more generally, with other compositional operators. The other way to achieve soundness in presence of circularity is to rely on semantic properties of the underlying model or, equivalently, of the formulas that describe that model.

We follow this second way with the following inference rule. Rather than using an *ad hoc* compositional operator, we simply write rely/guarantee guarantee specifications using implication $\Rightarrow$ to link assumption and guarantee of a module. On the other hand, we introduce assumptions on the behavior of the local assumptions and guarantee. More precisely, let us recall the definition of right-continuous and left-continuous states.

Definition 11 (States). A time-dependent predicate P is:

- A *state* iff $P \Rightarrow \mathrm{UpToNow}(P) \vee \mathrm{NowOn}(P)$ and $\neg P \Rightarrow \mathrm{UpToNow}(\neg P) \vee \mathrm{NowOn}(\neg P)$; we denote this by writing $P \in \mathsf{ST}\_$.

- A *left-continuous state* iff it is a state and moreover $P \Leftrightarrow \mathrm{UpToNow}(P)$; we denote this by writing $P \in \mathsf{ST}_{\rightarrow\bullet}$.
- A *right-continuous state* iff it is a state and moreover $P \Leftrightarrow \mathrm{NowOn}(P)$; we denote this by writing $P \in \mathsf{ST}_{\bullet\leftarrow}$.

Notice that a state which is both left-continuous and right-continuous is constant over time.

The inference rule of the following proposition requires that a module has a local assumption or local guarantee that behaves as a left-continuous state, whereas the other module has a local assumption or local guarantee that behaves as a right-continuous state. These two facts combined allows us to substitute an "alternation" of left- and right-continuous predicates to the use of a time progression operator, in order to set up an induction over time. In a sense, the interplay between left- and right-continuous states and the way they are logically implied in the local specifications results in a constancy over time of the local specifications. Notice that the rule is similar to the one in Proposition 5, except for the use of $\Rightarrow$ a compositional operator and for the semantic assumption on the local formulas.

Proposition 12 (Rely/Guarantee Circular Inference Rule 4). *If:*

1. *(a)* $E_1 \Rightarrow M_1$
   (b) $E_2 \Rightarrow M_2$
2. *(a)* $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$
3. $E \wedge M_1 \wedge M_2 \Rightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$, *for some* $i \in \{1, 2\}$
5. $E_1 \in \mathsf{ST}_{\bullet\leftarrow}$ *or* $M_1 \in \mathsf{ST}_{\bullet\leftarrow}$[5]
6. $E_2 \in \mathsf{ST}_{\rightarrow\bullet}$ *or* $M_2 \in \mathsf{ST}_{\rightarrow\bullet}$

then $\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$.

Proof for dense time domains. Let t be the current instant, and let us assume that $\mathrm{SomP_e}(S)$ holds at t. To show that $E \twoheadrightarrow M$ holds at t, let us first show that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ hold at t.

Let $t' < t$ be an instant at which S holds. From (4), let us assume that $\mathrm{NowOn}(E_i)$ holds at t'. This is without loss of generality, since if $\mathrm{NowOn}(M_i)$, then also $\mathrm{NowOn}(E_{\hat{i}})$ at the same time, from (2) and the fact that in the right-neighborhood of t' $\mathrm{SomP_e}(S)$ holds. Since $\mathrm{NowOn}(E_i)$ at t', then equivalently $\mathrm{UpToNow}(E_i)$ holds at some $t' + \epsilon$, for some $\epsilon > 0$.

[5] Obviously, fixing which module is described by a right-continuous state is without loss of generality.

Then, let us consider (1) at $t' + \epsilon$. We can infer that $\mathrm{UpToNow}(M_i)$, M_i, and $\mathrm{NowOn}(M_i)$ all hold at $t' + \epsilon$. Then, also $\mathrm{UpToNow}(E_{\hat{i}})$ holds at $t' + \epsilon$ from (2); but then, by (1) again, also $\mathrm{UpToNow}(M_{\hat{i}})$, $M_{\hat{i}}$, and $\mathrm{NowOn}(M_{\hat{i}})$ all hold at $t' + \epsilon$.

At this point, the proof goes on exactly as for Proposition 5. In particular, we infer that $\mathrm{UpToNow}(M_1 \wedge M_2)$, $M_1 \wedge M_2$, and $\mathrm{NowOn}(M_1 \wedge M_2)$ are true at t, and we conclude that $E \twoheadrightarrow M$ by applying (3). □

Since in the above proof we never evaluated hypothesis (3) at instants in which $\mathrm{SomP_i}(S)$ is false, we can actually strengthen that hypothesis and get the following variation. By doing this, the rule becomes complete, as we will show in Section 4.3.

Proposition 13 (Rely/Guarantee Circular Inference Rule 4(bis)). *If:*

1. *(a)* $E_1 \Rightarrow M_1$
   (b) $E_2 \Rightarrow M_2$
2. *(a)* $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$
3. $\mathrm{SomP_i}(S) \Rightarrow (E \wedge M_1 \wedge M_2 \Rightarrow M)$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$, *for some* $i \in \{1, 2\}$
5. $E_1 \in \mathsf{ST}_{\bullet\leftarrow}$ *or* $M_1 \in \mathsf{ST}_{\bullet\leftarrow}$
6. $E_2 \in \mathsf{ST}_{\rightarrow\bullet}$ *or* $M_2 \in \mathsf{ST}_{\rightarrow\bullet}$

then $\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$.

We conclude this section by pointing out that the notion of states (and right-continuous or left-continuous ones) is a meaningful one only when dealing with dense time models [GM01]. Therefore, the above rules are applicable to such time models only, and have no discrete-time counterparts.

4.2.3 A Circular Inference Rule Without Compositional Operator

Let us present one more circular inference rule. This rule differentiates itself from the other ones, in that it does not use a compositional operator and it does not make any assumptions on the behavior of the assumption and guarantee items. Moreover, it is substantially different than all the previous rules because it does not assert the validity of some global guarantee over a time interval that goes from the system initialization to the current time. Instead, it asserts that the system always responds in a timely manner by making true the global guarantee, provided the global assumption holds continuously for some time, and some of the local assumption or guarantee are periodically initialized. Thus, in a sense, it describes a system with a fixed bounded response time to

periodic initializations. In such systems the circularities are resolved *periodically*, rather than once for all at the beginning. Therefore, the rule is suitable to prove properties about systems composed of modules periodically responding to stimuli. In Section 6 we will demonstrate how to use the rule with an example of communication protocol.

According to our taxonomy, the rule is circular, non self-discharging, fully compositional, globally initialized (actually, not initialized at all), without compositional operator. Its soundness proof is rather simple, and requires no temporal induction, contrarily to the other rules we have presented.[6]

Proposition 14 (Rely/Guarantee Circular Inference Rule 5). *If, for some fixed duration* $\mathrm{T_B} > 0$*:*

1. *(a)* $E_1 \Rightarrow M_1$
   (b) $E_2 \Rightarrow M_2$
2. *(a)* $M_1 \Rightarrow E_2$
   (b) $M_2 \Rightarrow E_1$
3. $E \wedge M_1 \wedge M_2 \Rightarrow M$
4. $\exists i \in \{1,2\} : \mathrm{WithinF}(E_i, \mathrm{T_B}) \vee \mathrm{WithinF}(M_i, \mathrm{T_B})$

then $\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$.

Proof. Let t be a generic time instant at which $\mathrm{Lasts}(E, \mathrm{T_B})$ holds, and prove that $\mathrm{WithinF}(M, \mathrm{T_B})$. Notice that the same proof works for both dense and discrete time domains.

From (4), let us assume that $\mathrm{WithinF}(E_1, \mathrm{T_B})$ at t. This is without loss of generality: in fact, if $\mathrm{WithinF}(M_2, \mathrm{T_B})$, then $\mathrm{WithinF}(E_1, \mathrm{T_B})$ from (2b); if $\mathrm{WithinF}(E_2, \mathrm{T_B})$, then $\mathrm{WithinF}(M_2, \mathrm{T_B})$ from (1b); if $\mathrm{WithinF}(M_1, \mathrm{T_B})$, then $\mathrm{WithinF}(E_2, \mathrm{T_B})$ from (2a).

Thus, there exists a t' such that $t < t' < \mathrm{T_B}$ and E_1 holds at t'. Therefore, $E_1 \wedge M_1 \wedge E_2 \wedge M_2$ holds at at t' by (1) and (2). Moreover, since $t < t' < \mathrm{T_B}$ and $\mathrm{Lasts}(E, \mathrm{T_B})$ at t, then E also holds at t'. But then, M holds at t' by (4). Since $t < t' < \mathrm{T_B}$, this implies that $\mathrm{WithinF}(M, \mathrm{T_B})$ at t. □

4.3 Completeness of Circular Inference Rules

This section analyzes the completeness of the circular inference rules presented above.

Theorem 15. *The rely/guarantee circular inference rule 1 of Proposition 5 is incomplete.*

[6] Notice that the conclusion formula $\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$ is a weaker version of the bounded *release* operator $\mathrm{Releases}_{<T}(P, Q)$. The operator is usually defined as $\forall 0 < t < T : \mathrm{Futr}(Q, t) \vee \mathrm{WithinF}(P, t)$. Therefore $\mathrm{Releases}_{<\mathrm{T_B}}(M, \neg E)$ implies $\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$, but not *vice versa*, as the latter is true whenever E holds only over intervals of length less than $\mathrm{T_B}$.

Proof. Let σ, ϵ, μ be three Boolean basic time-dependent items; let us take $S = \sigma$, $E = \epsilon$, and $M = \mu$. Then, in order to show incompleteness, we provide a history for σ, ϵ, μ such that the conclusion of the inference rule holds, but no choice of E_i, M_i makes true all the premises of the rule.

The history is as follows. μ and ϵ are false everywhere, whereas σ is true only at some point s internal to the time domain (and it is false everywhere else). Notice that this history is definable in TRIO by the two formulas $\mathrm{Alw}(\neg\mu \wedge \neg\epsilon)$ and $\mathrm{Som}(\sigma \wedge \mathrm{AlwP}(\neg\sigma) \wedge \mathrm{AlwF}(\neg\sigma))$.[7]

Let us now realize that $E \twoheadrightarrow M$ is always true, since E is always false; therefore the conclusion of the inference rule holds *a fortiori*.

Let us now consider what happens at s. In order to make (4) true, let us assume that it is $\mathrm{UpToNow}(E_1)$ at s. As usual, this is without loss of generality, since if $\mathrm{UpToNow}(M_2)$ then $\mathrm{UpToNow}(E_1)$ in order to satisfy (2b); if $\mathrm{UpToNow}(E_2)$ then $\mathrm{UpToNow}(M_2)$ in order to satisfy (1b); if $\mathrm{UpToNow}(M_1)$ then $\mathrm{UpToNow}(E_2)$ in order to satisfy (2a).

Then, $\mathrm{UpToNow}(E_1)$ at s implies $\mathrm{UpToNow}(M_1)$ at s to satisfy (1a), and therefore also $\mathrm{UpToNow}(M_2)$ at s to satisfy (1b, 2). Since $\mathrm{UpToNow}(M_1 \wedge M_2)$ at s, then (3) requires that $\mathrm{UpToNow}(M)$ at s as well. But $M = \mu$ is false everywhere by assumption, so there is no choice of E_1, E_2, M_1, M_2 that is compatible with all the premises of the rule. □

Theorem 16. *The rely/guarantee circular inference rule 2 of Proposition 6 is incomplete.*

Proof. Let σ, ϵ, μ be three Boolean basic time-dependent items; let us take $S = \sigma$, $E = \epsilon$, and $M = \mu$. Then, in order to show incompleteness, we provide a history for σ, ϵ, μ such that the conclusion of the inference rule holds, but no choice of E_i, M_i makes true all the premises of the rule.

The history is as follows. μ is false everywhere, whereas σ is true only at some point s internal to the time domain (and it is false everywhere else). At s, ϵ is true on a non-empty interval on the left which is strictly contained within the time domain; that is, let us say that ϵ holds exactly on the interval $(s-\gamma, s)$ for some suitable $\gamma > 0$. Notice that the history is definable in TRIO.

Let us now realize that $\mathrm{AlwP_e}(E)$ is always false, since $(s-\gamma, s)$ is strictly contained in the time domain by hypothesis. Therefore, the conclusion of the inference rule coincides with an implication with false antecedent, and it is therefore trivially true.

Let us now consider what happens at s. In order to make (4) true, it must be either $\mathrm{UpToNow}(E_1 \wedge E_2)$ or $\mathrm{UpToNow}(M_1 \wedge M_2)$ at s. However, if $\mathrm{UpToNow}(E_1 \wedge E_2)$ then also $\mathrm{UpToNow}(M_1 \wedge M_2)$ by (1), so let us assume the latter without loss of generality.

Finally, notice that it is also $\mathrm{UpToNow}(E)$ at s, therefore it must be $\mathrm{UpToNow}(M)$ at s. But $M = \mu$ is false everywhere, thus we have a false premise. □

[7] For simplicity, assume a bi-infinite time domain such as $\mathbb{R}$ or $\mathbb{Z}$, so that there are no (finite) boundaries. Extensions to mono-infinite or even infinite time domains are routine.

If we look carefully at the above incompleteness proofs, we notice that the main obstacle to achieving completeness is the impossibility of choosing suitable values for E_i, M_i *before the system is initialized*, that is when $\mathrm{SomP_i}(S)$ is false. As a consequence, by switching to locally initialized inference rules, it may be possible to achieve completeness. Nonetheless, having a locally initialized inference rule is neither a sufficient nor a necessary condition for completeness. Indeed, in the remainder we show that the locally initialized inference rule 1(bis) of Proposition 7 is incomplete, whereas the locally initialized inference rule 2(bis) of Proposition 8 is complete. Moreover, we show that the globally initialized rule 1(ter) of Proposition 9 is also complete. Therefore, in general the relationship between the features of a compositional rule and its completeness is subtle and non straightforward.

Theorem 17. *The rely/guarantee circular inference rule 1(bis) of Proposition 7 is incomplete.*

Proof. As we did above, let σ, ϵ, μ be three Boolean basic time-dependent items; let us take $S = \sigma$, $E = \epsilon$, and $M = \mu$. In order to show incompleteness, we provide a history for σ, ϵ, μ such that the conclusion of the inference rule holds, but no choice of E_i, M_i makes true all the premises of the rule.

The history is as follows. μ and ϵ are false everywhere, whereas σ is true only at some point s internal to the time domain (and it is false everywhere else). Notice that the history is definable in TRIO.

Let us now realize that $E \twoheadrightarrow M$ is always true, since E is always false; therefore the conclusion of the inference rule holds *a fortiori*.

Let us now consider what happens at s. In order to make (4) true, let us assume that it is $\mathrm{NowOn}(E_1)$ at s. As usual, this is without loss of generality, since if $\mathrm{NowOn}(M_2)$ then $\mathrm{NowOn}(E_1)$ in order to satisfy (2b); if $\mathrm{NowOn}(E_2)$ then $\mathrm{NowOn}(M_2)$ in order to satisfy (1b); if $\mathrm{NowOn}(M_1)$ then $\mathrm{NowOn}(E_2)$ in order to satisfy (2a).

Then, $\mathrm{NowOn}(E_1)$ at s implies $\mathrm{NowOn}(M_1)$ at s to satisfy (1a), and therefore also $\mathrm{NowOn}(M_2)$ at s to satisfy (1b, 2). Since $\mathrm{NowOn}(M_1 \wedge M_2)$ at s, then (3) requires that $\mathrm{NowOn}(M)$ at s as well. But $M = \mu$ is false everywhere by assumption, so there is no choice of E_1, E_2, M_1, M_2 that is compatible with all the premises of the rule. □

Notice that in the proofs of the following Theorem (18) we assume to deal only with non-Zeno items. We conjecture that the theorem holds also for Zeno items, but the proof would be even more involved — and not practically very interesting, as Zenoness is anyway a source of incompleteness on its own [GM01] — so we omit it for simplicity.

Theorem 18. *The rely/guarantee circular inference rule 1(ter) of Proposition 9 is complete.*

Proof. Let us assume that the conclusion $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$ holds. Then,

let us define M_1 as follows.

$$M_1 \equiv \begin{cases} E \twoheadrightarrow M & \text{if } \mathrm{SomP_i}(S) \\ \text{true} & \text{if } \mathrm{Until}(M,S) \vee \mathrm{Until}(\neg M \wedge \neg E, S) \text{ and } \mathrm{AlwP_i}(\neg S) \\ \text{false} & \text{otherwise} \end{cases}$$

Equivalently, we can express the definition above with the TRIO formula:

$$\begin{aligned} &(\mathrm{SomP_i}(S) \Rightarrow (M_1 \Leftrightarrow (E \twoheadrightarrow M))) \quad \wedge \\ &\quad (\mathrm{AlwP_i}(\neg S) \wedge (\mathrm{Until}(M,S) \vee \mathrm{Until}(\neg M \wedge \neg E, S)) \Rightarrow M_1) \quad \wedge \\ &\quad\quad (\mathrm{AlwP_i}(\neg S) \wedge \neg\mathrm{Until}(M,S) \wedge \neg\mathrm{Until}(\neg M \wedge \neg E, S) \Rightarrow \neg M_1) \end{aligned}$$

or, more concisely, with the equivalent formula:

$$\begin{aligned} M_1 \Leftrightarrow\ & (\mathrm{SomP_i}(S) \wedge (E \twoheadrightarrow M)) \\ & \vee (\mathrm{AlwP_i}(\neg S) \wedge (\mathrm{Until}(M,S) \vee \mathrm{Until}(\neg M \wedge \neg E, S))) \end{aligned}$$

Similarly, let us choose $M_2 = E_1 = E_2$ as follows.

$$M_2 = E_1 = E_2 \equiv \begin{cases} \text{true} & \text{if } \mathrm{Until}(M,S) \text{ or } \mathrm{Until}(\neg M \wedge \neg E, S) \text{ or } \mathrm{SomP_i}(S) \\ \text{false} & \text{otherwise} \end{cases}$$

Then:

- Let us consider hypothesis (1a) at some instant t. We distinguish the following cases:
  - If $\mathrm{SomP_e}(S)$ holds at t, then notice that the three formulas: $\mathrm{UpToNow}(E \twoheadrightarrow M)$, $\mathrm{NowOn}(E \twoheadrightarrow M)$, and $\mathrm{UpToNow}(E_1)$ also hold at t. This is because $\mathrm{SomP_e}(S)$ implies $\mathrm{UpToNow}(\mathrm{SomP_e}(S))$ and $\mathrm{NowOn}(\mathrm{SomP_e}(S))$. Therefore, (1a) is equivalent to $\mathrm{UpToNow}(E_1) \Rightarrow \mathrm{UpToNow}(E \twoheadrightarrow M) \wedge (E \twoheadrightarrow M) \wedge \mathrm{NowOn}(E \twoheadrightarrow M)$, and it does hold at t.
  - If S at t (and $\neg\mathrm{SomP_e}(S)$), then without loss of generality let us assume that $\mathrm{UpToNow}(E_1)$ holds as well; if not, (1a) is trivially true since $\mathrm{UpToNow}(\neg E_1)$ would follow for properties of Zeno items. But from the definition of M_1 notice that whenever $\mathrm{UpToNow}(E_1)$ and S, then also $\mathrm{UpToNow}(M_1)$. Moreover, after t it is $\mathrm{SomP_e}(S)$, and thus $M_1 = (E \twoheadrightarrow M)$ holds there by assumption. Thus, at t it is also M_1 and $\mathrm{NowOn}(M_1)$. This establishes (1a) in this case.
  - Otherwise $\mathrm{AlwP_i}(\neg S)$ at t. Again, without loss of generality we can assume that $\mathrm{UpToNow}(E_1)$. But then, whenever $\mathrm{AlwP_i}(\neg S)$, M_1 holds iff E_1 holds. So, if also $\mathrm{NowOn}(\neg S)$, (1a) holds at t. Otherwise, it must be $\mathrm{NowOn}(S)$ and $\neg S$ at t. In this case, $\mathrm{NowOn}(\mathrm{SomP_i}(S))$ holds at t, and thus $\mathrm{NowOn}(M_1)$ also holds at t (from the assumed conclusion of the inference rule and the definition of M_1). Thus finally, (1a) holds at t in this case as well.

- The reasoning for hypothesis (1b) is similar to that for hypothesis (1b), only a bit simpler since E_2 and M_2 are everywhere equal. We omit the details which are the same as in the previous step.

- Let us consider hypothesis (2a): if it is evaluated when $\mathrm{SomP_i}(S)$, then is reduces to $(E \twoheadrightarrow M) \Rightarrow$ true, which is true by hypothesis. If instead $\mathrm{AlwP_i}(\neg S)$, then either $\mathrm{Until}(M, S) \vee \mathrm{Until}(\neg M \wedge \neg E, S)$ or not. In the former case, (2a) reduces to an implication with true consequent; in the latter case it reduces to an implication with false antecedent. Both are tautologies.

- A similar reasoning goes for hypothesis (2b), which reduces to either true $\Rightarrow$ true or to false $\Rightarrow$ false.

- Let us consider hypothesis (3) when $\mathrm{SomP_i}(S)$. Then, it can be rewritten as $E \wedge (E \twoheadrightarrow M) \twoheadrightarrow M$. Now, notice that if $E \twoheadrightarrow M$ then *a fortiori* $E \wedge (E \twoheadrightarrow M) \twoheadrightarrow M$, since the latter can be written as an implication with the same consequent but a more demanding antecedent. Since we are assuming that $E \twoheadrightarrow M$, we are done in this case.

  Otherwise, $\mathrm{AlwP_i}(\neg S)$, and let t be the instant at which we are evaluating (3). Now the analysis gets more involved. First of all, let us consider the case in which $\mathrm{AlwF_e}(\neg S)$. Thus S is always false; therefore, $\mathrm{Until}(M, S) \vee \mathrm{Until}(\neg M \wedge \neg E, S)$ is also false (since we are dealing with strong until), and thus $M_2 = E_1 = E_2 =$ false everywhere. Therefore $\mathrm{UpToNow}(M_2)$ is always false, which implies that $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$ is always true.

  Otherwise S is true somewhere in the future. Without loss of generality, since we are assuming non-Zeno items, let $s > t$ be the *next* time instant at which S or $\mathrm{NowOn}(S)$ holds. Now we further consider two cases:

  - If $\mathrm{Until}(M, S)$ at t, then in particular M holds over the interval (t, s). Let us now consider the next point in the past from t at which M becomes false; let $u \leq t$ be this instant. Thus, M holds continuously over the interval (u, s) and is false before u (the value of M exactly at u is not relevant now). Notice that for all instants in (u, s), the formula $\mathrm{UpToNow}(M) \wedge M \wedge \mathrm{NowOn}(M)$ holds, since we are within an open interval. This implies that (3) can be rewritten as an implication with true consequent (by considering the definition of the $\twoheadrightarrow$ operator), which is therefore true throughout (u, s).

    We still have to evaluate (3) in the interval $(-\infty, u]$. Notice that in all these instants, the formula $\mathrm{UpToNow}(M_1 \wedge M_2)$ is false; in fact, M_1 and M_2 are true at most at u and after it, but false before it since $\mathrm{Until}(M, S)$ no longer holds (recall that $\mathrm{AlwP_i}(\neg S)$ holds at $t \geq u$). Therefore, *a fortiori* $\mathrm{UpToNow}(E \wedge M_1 \wedge M_2)$ is false at these instants. But then $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$ is equivalent to an implication with false antecedent, which is trivially true.

– If instead $\mathrm{Until}(\neg M \wedge \neg E, S)$ at t, then let us repeat the above reasoning for $\neg M \wedge \neg E$ in place of M. So, assume that M and E are false throughout (u, s) for some $u \leq t$. As a consequence, $\mathrm{UpToNow}(M) \wedge M \wedge \mathrm{NowOn}(M)$ is now false throughout (u, s), and so is $\mathrm{UpToNow}(E)$. Therefore, *a fortiori* $\mathrm{UpToNow}(E \wedge M_1 \wedge M_2)$ is false at these instants. But then $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$ is equivalent to an implication with false antecedent, which is trivially true.

Similarly as the previous case, in the interval $(-\infty, u]$ the formula $\mathrm{UpToNow}(M_1 \wedge M_2)$ is false. Thus, $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$ is equivalent to an implication with false antecedent, which is trivially true.

– Notice that we have no other cases to consider. In particular if both $\mathrm{Until}(M, S)$ and $\mathrm{Until}(\neg M \wedge \neg E, S)$ are false at t, then $\mathrm{UpToNow}(E \wedge \neg M)$ holds at s. But this contradicts the assumption that the conclusion of the inference rule is true there.

- Finally, hypothesis (4). Let us consider any instant s at which S holds. At s, either $\mathrm{UpToNow}(M)$ or $\mathrm{UpToNow}(\neg M)$, since we are dealing with non-Zeno items. If $\mathrm{UpToNow}(M)$, then $\mathrm{UpToNow}(M_2 \wedge E_1 \wedge M_1)$, since $\mathrm{Until}(M, S)$ holds in a left-neighborhood of s. If $\mathrm{UpToNow}(\neg M)$ then it must also be $\mathrm{UpToNow}(\neg E)$, otherwise the conclusion $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$ would be false at s. Therefore, $\mathrm{UpToNow}(M_2 \wedge E_1 \wedge M_1)$, since $\mathrm{Until}(\neg M \wedge \neg E, S)$ holds in a left-neighborhood of s. □

Theorem 19. *The rely/guarantee circular inference rule 2(bis) of Proposition 8 is complete.*

Proof. The proof is all the same as the previous one: let us assume that the conclusion $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$ holds. Then, let us define M_1 as $M_1 = E \twoheadrightarrow M$, and let $M_2 = E_1 = E_2 = \mathrm{true}$ be all identically equal to true. Therefore:

- Hypothesis (1a) is equivalent to $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$ which is exactly the conclusion.

- Hypotheses (1b,2) are all equivalent to $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow \mathrm{true}$, which is trivially true.

- Let us consider hypothesis (3). Without loss of generality, let us take any instant at which $\mathrm{SomP_e}(S)$ and $\mathrm{AlwP_e}(E)$ hold. Then, we have to establish that $(E \twoheadrightarrow M) \wedge \mathrm{true} \Rightarrow M$. Clearly, this is the same as just M, since $E \twoheadrightarrow M$ holds from the conclusion. Now notice that $\mathrm{AlwP_e}(E)$ implies *a fortiori* that $\mathrm{UpToNow}(E)$.[8] Therefore, the conclusion allows us to assert that $\mathrm{UpToNow}(M) \wedge M \wedge \mathrm{NowOn}(M)$, so that M is, in particular, true.

[8] We are skipping a little technicality here: we are tacitly assuming that we are at a point internal to the time domain, otherwise it could be that $\mathrm{AlwP_e}(E)$ is trivially true, but $\mathrm{UpToNow}(E)$ is false since there is no non-empty interval to the left of the boundary. However, this is not a problem, as we just have to notice that $\mathrm{SomP_e}(S)$ being true implies that we are indeed in an internal point.

- Hypothesis (4) is trivially satisfied for $E_1 = E_2 = \text{true}$. □

The following result has been already proved in [FRMM06].

Theorem 20. *The rely/guarantee circular inference rule 3 of Proposition 10 is incomplete.*

Proof. Let σ, ϵ, μ be three Boolean basic time-dependent items; let us take $S = \sigma$, $E = \epsilon$, and $M = \mu$. Then, in order to show incompleteness, we provide a history for σ, ϵ, μ such that the conclusion of the inference rule holds, but no choice of E_i, M_i makes true all the premises of the rule.

The history is as follows. μ and ϵ are false everywhere, whereas σ is true only at some point s internal to the time domain (and it is false everywhere else). Notice that the history is definable in TRIO.

Let us now realize that $E \gg M$ is always true, since $\text{AlwP}_\text{e}(E)$ is always false; therefore the conclusion of the inference rule holds *a fortiori*.

Let us now consider what happens at s. Without adding details (as they are all similar as in the other proofs), one realizes that in order for (4) to be true, we must have $\text{AlwP}_\text{e}(E_1 \wedge E_2 \wedge M_1 \wedge M_2)$ at s.

Then, (3) requires that $\text{AlwP}_\text{e}(M)$ at s. But $M = \mu$ is false everywhere by assumption, so there is no choice of E_1, E_2, M_1, M_2 that is compatible with all the premises of the rule. □

Theorem 21. *The rely/guarantee circular inference rule 4 of Proposition 12 is incomplete, whereas the rule 4(bis) of Proposition 13 is complete.*

Proof. The incompleteness proof is along the usual lines. Let S be true exactly at some time s, M false everywhere, and E be true on some open interval $(s - \epsilon, s)$, and false everywhere else (in particular, this implies that $\text{UpToNow}(E) \wedge \neg E$ holds at s). Notice that this implies that whenever $\text{SomP}_\text{i}(S)$, then E is false, so $E \Rightarrow M$ holds; thus the conclusion of the inference rule holds everywhere. Then, by (4) and (1–2) it must be $\text{UpToNow}(E_1 \wedge E_2 \wedge M_1 \wedge M_2)$ at s. Moreover, by (3) $\text{UpToNow}(M)$ must also hold at s, a contradiction.

The completeness proof instead is as follows. Let us assume that the conclusion $\text{SomP}_\text{i}(S) \Rightarrow (E \Rightarrow M)$ holds. Then, let us define M_1 as $M_1 = \text{SomP}_\text{i}(S) \Rightarrow (E \Rightarrow M)$, and let $M_2 = E_1 = E_2 = \text{true}$ be all identically equal to true.

(1a) is the same as $\text{SomP}_\text{i}(S) \Rightarrow (E \Rightarrow M)$, which holds by assumption. (1b) is trivially true, being an implication with true consequent. (2) are both trivially true, also being implications with true consequents. (3) can be rewritten as $\text{SomP}_\text{i}(S) \Rightarrow (E \Rightarrow M)$, again true by assumption. (4) is satisfied everywhere by $E_1 = E_2 = \text{true}$. Finally, notice that a constant Boolean time-dependent item is both a right-continuous and a left-continuous state, so (5–6) is satisfied by E_1, E_2. □

Theorem 22. *The rely/guarantee circular inference rule 5 of Proposition 14 is complete.*

Proof. Let us assume that the conclusion $\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$ holds. Then, let us define $E_1 = E_2 = M_1 = M_2$ implicitly as follows.

$$\begin{aligned}(\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{Lasts}(E_1 = E_2 = M_1 = M_2 = M, \mathrm{T_B})) \quad \wedge \\ (\mathrm{WithinF}(\neg E, \mathrm{T_B}) \Rightarrow \\ \mathrm{Until}(E_1 = E_2 = M_1 = M_2 = \neg E, \mathrm{Lasts}(E, \mathrm{T_B}) \wedge E_1 = E_2 = M_1 = M_2 = \neg E) \vee \\ \mathrm{AlwF_e}(\mathrm{WithinF}(\neg E, \mathrm{T_B}) \wedge E_1 = E_2 = M_1 = M_2 = \neg E))\end{aligned}$$

Thus clearly (1–2) all reduce to propositional tautologies of the form $A \Rightarrow A$, since $E_1 = E_2 = M_1 = M_2$ everywhere.

Then, let us consider (4) first, and let t be any instant. At t, either $\mathrm{Lasts}(E, \mathrm{T_B})$ or $\mathrm{WithinF}(\neg E, \mathrm{T_B})$. In the former case, the conclusion lets us deduce that $\mathrm{WithinF}(M, \mathrm{T_B})$. But since also $\mathrm{Lasts}(E_1 = E_2 = M_1 = M_2 = M, \mathrm{T_B})$ at t, then $\mathrm{WithinF}(E_1 \wedge E_2 \wedge M_1 \wedge M_2, \mathrm{T_B})$, which satisfies (4). Otherwise, we have a $t < t' < t + \mathrm{T_B}$ such that $\neg E$ holds at t'. Then, it is either $\mathrm{AlwF_e}(E_1 = E_2 = M_1 = M_2 = \neg E \wedge \mathrm{WithinF}(\neg E, \mathrm{T_B}))$ at t, or not. In the former case, we have $E_1 = E_2 = M_1 = M_2$ at t', and thus (4) is satisfied. In the latter case, there exists an instant $t'' > t$ such that $E_1 = E_2 = M_1 = M_2 = \neg E$ holds until t'' *included*. If $t'' \geq t'$, then (4) is implied; otherwise $t'' < t'$, $\mathrm{Lasts}(E, \mathrm{T_B})$ holds at t'', which contradicts the fact that E is false at t'.

Next, let us discuss hypothesis (3), at a generic time instant t. If E is false at t, then the implication holds trivially.

Otherwise E is true at t. Let us distinguish two cases: (a) if there exists an interval $(p, p+\mathrm{T_B})$ such that $t \in (p, p+\mathrm{T_B})$ and E is true throughout $(p, p+\mathrm{T_B})$; (b) if this is not the case. Notice that if (b), then in particular there must exist an instant $t - \mathrm{T_B} < q < t$ such that $\mathrm{WithinF}(\neg E, \mathrm{T_B})$ holds at q.

Thus, if (a) is the case, then let us consider the instant p. At p $\mathrm{Lasts}(E, \mathrm{T_B})$ holds, therefore $\mathrm{Lasts}(E_1 = E_2 = M_1 = M_2 = M, \mathrm{T_B})$ also holds at p. Thus at $t \in (p, p + \mathrm{T_B})$: if M, then also $M_1 \wedge M_2$, thus (3) holds; if $\neg M$, then also $\neg M_1 \wedge \neg M_2$, thus (3) also holds, being an implication with false antecedent.

Otherwise, (b) is the case, and $\mathrm{WithinF}(\neg E, \mathrm{T_B})$ holds at q. Let us distinguish whether $\mathrm{Lasts}(\mathrm{WithinF}(\neg E, \mathrm{T_B}), t - q)$ holds at q or not. If it does, then surely $E_1 = E_2 = M_1 = M_2 = \neg E$ at t, and therefore (4) reduces to $E \wedge \neg E \Rightarrow M$, which is trivially true having a contradiction in the antecedent. If it does not, then there exists some $q < q' < t$ such that $\mathrm{Lasts}(E, \mathrm{T_B})$ holds at q'. But this corresponds to case (a), a contradiction which concludes this branch as well. □

4.4 Summary of Circular Rules

Table 3 summarizes the various circular compositional inference rules we have presented above; an I/C letter after before the proposition number denotes if the rule is incomplete/complete.

Rule 1 (Prop. 5 I)	**Rule 1(bis) (Prop. 7 I)**	**Rule 1(ter) (Prop. 9 C)**
$E_1 \twoheadrightarrow M_1$, $E_2 \twoheadrightarrow M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $M_1 \wedge M_2 \Rightarrow M$ $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$	$\mathrm{SomP_e}(S) \Rightarrow (E_1 \twoheadrightarrow M_1) \wedge (E_2 \twoheadrightarrow M_2)$ $\mathrm{SomP_e}(S) \Rightarrow (M_1 \Rightarrow E_2) \wedge (M_2 \Rightarrow E_1)$ $\mathrm{SomP_e}(S) \Rightarrow (M_1 \wedge M_2 \Rightarrow M)$ $S \Rightarrow \mathrm{NowOn}(E_i) \vee \mathrm{NowOn}(M_i)$	$E_1 \twoheadrightarrow M_1$, $E_2 \twoheadrightarrow M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $E \wedge M_1 \wedge M_2 \twoheadrightarrow M$ $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$
$\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_e}(S) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$
Rule 2 (Prop. 6 I)	**Rule 2(bis) (Prop. 8 C)**	**Rule 3 (Prop. 10 I)**
$E_1 \twoheadrightarrow M_1$, $E_2 \twoheadrightarrow M_2$ $E \wedge M_1 \wedge M_2 \Rightarrow E_1 \wedge E_2$ $M_1 \wedge M_2 \Rightarrow M$ $S \Rightarrow \mathrm{UpToNow}(E_1 \wedge E_2) \vee \mathrm{UpToNow}(M_1 \wedge M_2)$	$\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E_1 \twoheadrightarrow M_1) \wedge (E_2 \twoheadrightarrow M_2)$ $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \wedge M_1 \wedge M_2 \Rightarrow E_1 \wedge E_2)$ $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (M_1 \wedge M_2 \Rightarrow M)$ $S \Rightarrow \mathrm{NowOn}(E_1 \wedge E_2) \vee \mathrm{NowOn}(M_1 \wedge M_2)$	$E_1 \gg M_1$, $E_2 \gg M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $M_1 \wedge M_2 \Rightarrow M$ $S \Rightarrow \mathrm{AlwP_e}(E_i) \vee \mathrm{AlwP_e}(M_i)$
$\mathrm{SomP_i}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \gg M)$
Rule 4 (Prop. 12 I)	**Rule 4(bis) (Prop. 13 C)**	**Rule 5 (Prop. 14 C)**
$E_1 \Rightarrow M_1$, $E_2 \Rightarrow M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $E \wedge M_1 \wedge M_2 \Rightarrow M$ $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$ E_1 or $M_1 \in \mathsf{ST}_{\bullet\leftarrow}$, E_2 or $M_2 \in \mathsf{ST}_{\rightarrow\bullet}$	$E_1 \Rightarrow M_1$, $E_2 \Rightarrow M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $\mathrm{SomP_i}(S) \Rightarrow (E \wedge M_1 \wedge M_2 \Rightarrow M)$ $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$ E_1 or $M_1 \in \mathsf{ST}_{\bullet\leftarrow}$, E_2 or $M_2 \in \mathsf{ST}_{\rightarrow\bullet}$	$E_1 \Rightarrow M_1$, $E_2 \Rightarrow M_2$ $M_1 \Rightarrow E_2$, $M_2 \Rightarrow E_1$ $E \wedge M_1 \wedge M_2 \Rightarrow M$ $\mathrm{WithinF}(E_i \vee M_i, \mathrm{T_B})$
$\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$	$\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$

Table 3: Circular compositional inference rules for two modules.

5 Generalization to More Than Two Modules

This section discusses how the compositional inference rules presented in Section 3 and 4 can be generalized to handle a system with any number $N \geq 2$ of modules.

Let us point out that these generalizations are not strictly necessary to apply our compositional rules to systems with more than two modules. In fact, the rules can be applied by recursively partitioning the set of modules into two suitable sets, and apply the compositional rules by considering the two subsets as two (macro) modules. Nonetheless, the structure of a multi-modular system is often best exploited by considering the composition of all the modules at the same level. In fact, a well-designed system often has a modular partitioning where it is "natural" to associate some local properties to each of its modules. Therefore, we extend our compositional rules to handle such commonly encountered cases.

Throughout this section N is an integer greater than one, that represents the number of modules.

5.1 Non-Circular Inference Rules

The non-circular inference rules of Propositions 2 and 3 can be extended to handle any number of modules by introducing a way to describe a set of discharging formulas where the local assumption of each module is discharged by the local guarantee of another module, and no circularities are introduced. To this end, it is sufficient to describe a permutation of the set of modules where the local assumption of each module is discharged by the local guarantee of the module which follows in the permutation.

Thus, let us denote by π_N permutations of the set $\{1, \ldots, N\}$. By $\pi_N(i)$ we mean the ith element of the permutation, for $i \in \{1, \ldots, N\}$. For example if $N = 3$ and $\pi_3 = [3, 1, 2]$, then $\pi_3(1) = 3$, $\pi_3(2) = 1$ and $\pi_3(3) = 2$.

Let us first consider the extension of rule 1 of Proposition 2.

Proposition 23 (Non-Circular Rely/Guarantee Inference Rule 1N). *If there exists a permutation π_N such that:*

1. *for all $i \in \{1, \ldots, N\}$: $E_i \sqsubseteq M_i$*
2. *for all $i \in \{1, \ldots, N-1\}$: $E \sqcap M_{\pi_N(i)} \sqsubseteq E_{\pi_N(i+1)}$*
3. *$E \sqcap M_1 \sqcap \cdots \sqcap M_N \sqsubseteq M$*
4. *$E \sqsubseteq M_{\pi_N(1)}$*

then $E \sqsubseteq M$

Proof sketch. Let us just provide a sketch, since the proof is easily derivable from that of Proposition 2. Assume that E holds, then $M_{\pi_N(1)}$ by (4). Then, by successively applying (1) and (2) for $i = \pi_N(1), \pi_N(2), \ldots, \pi_N(N)$, we get that all $E_{\pi_N(2)}$, $M_{\pi_N(2)}$, $\ldots$, $M_{\pi_N(N)}$ hold. Therefore, M holds by (3). □

The extension of Proposition 3 is very similar, the only difference being that we now have a fully compositional rule. For brevity, we omit the (obvious) proof.

Proposition 24 (Non-Circular Rely/Guarantee Inference Rule 2N). *If there exists a permutation π_N such that:*

1. *for all $i \in \{1, \ldots, N\}$: $E_i \sqsubseteq M_i$*
2. *for all $i \in \{1, \ldots, N-1\}$: $M_{\pi_N(i)} \sqsubseteq E_{\pi_N(i+1)}$*
3. *$E \sqcap M_1 \sqcap \cdots \sqcap M_N \sqsubseteq M$*
4. *$E \sqsubseteq E_{\pi_N(1)}$*

then $E \sqsubseteq M$

Completeness. Let us remark that the completeness results about the two rules of Propositions 2 and 3 clearly carry on to Propositions 23 and 24, which therefore define *complete* inference rules. In fact, if we assume $E \sqsubseteq M$, then both Propositions 23 and 24 define a complete rule if we choose $M_{\pi_N(1)} = M$, $E_{\pi_N(1)} = E$, and all other E_i' s and M_i's equal to $\top$.

Summary. Table 4 summarizes the two above non-circular inference rules.

Rule 1N (Prop. 23 C)	**Rule 2N (Prop. 24 C)**
$\forall i \in \{1, \ldots, N\} : (E_i \Rightarrow M_i)$ $\forall i \in \{1, \ldots, N-1\} : (E \wedge M_{\pi_N(i)} \Rightarrow E_{\pi_N(i+1)})$ $E \wedge M_1 \wedge \cdots \wedge M_N \Rightarrow M$ $E \Rightarrow M_{\pi_N(1)}$	$\forall i \in \{1, \ldots, N\} : (E_i \Rightarrow M_i)$ $\forall i \in \{1, \ldots, N-1\} : (M_{\pi_N(i)} \Rightarrow E_{\pi_N(i+1)})$ $E \wedge M_1 \wedge \cdots \wedge M_N \Rightarrow M$ $E \Rightarrow E_{\pi_N(1)}$
$E \Rightarrow M$	$E \Rightarrow M$

Table 4: Non-circular compositional inference rules for $N \geq 2$ modules.

5.2 Circular Inference Rules

Let us now generalize the circular compositional inference rules of Section 4 to the case of more than two modules. There are basically two modifications to introduce in each rule to generalize it; the underlying rationale for these two changes is common.

The first change is to the discharging formulas. Whereas in the two module case each module's assumption was usually discharged by means of the other module's guarantee, with $N > 2$ modules it is simpler and more natural to specify that the conjunction of all the modules' guarantees discharges the conjunction of all the modules' assumptions: this makes the rules self-discharging, but simpler to state as we do not have to specify complicated discharging relations. Clearly, variations are possible, but they may render the statement of the rule more involved. According to our general approach to compositionality,

we recommend the formulation of such variations to be driven by the particular systems being modeled, rather than *a priori* for the sake of exploring different rules.

The second change is related to the first one. In fact, if the discharging of the assumptions depends on *all* the guarantees being true, we have to modify the initialization condition so that one can infer from S being true that all the guarantees are also true at the same time. In practice, this can be usually implemented by requiring that the guarantee or the assumption of each module holds when S holds.

In the remainder of this section we presents generalizations of the inference rules of Section 4 along these lines. We provide proofs for just a few of the propositions, the other proofs being routine. In the remainder, we denote the finite set $\{1, \ldots, N\}$ as $\mathcal{I}_N$.

Circular inference rules for the time progression operator.

Proposition 25 (Rely/Guarantee Circular Inference Rule 1N). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \twoheadrightarrow M_i$
2. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i$
3. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$*, for all* $i \in \mathcal{I}_N$

then $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$.

Proof. Let t be the current instant, and let us assume that $\mathrm{SomP_i}(S)$ holds at t; thus, let $t' \leq t$ be (any) instant at which S held. We have to show that $E \twoheadrightarrow M$ holds at t.

For any $i \in \mathcal{I}_N$, either $\mathrm{UpToNow}(E_i)$ or $\mathrm{UpToNow}(M_i)$. In particular, let $I \subseteq \mathcal{I}_N$ the set of modules' indexes for which $\mathrm{UpToNow}(E_i)$ holds. Then, from (1) we can infer that also $\mathrm{UpToNow}(M_i)$ for all $i \in I$. Therefore, all in all we have that $\mathrm{UpToNow}\big(\bigwedge_{i \in \mathcal{I}_N} M_i\big)$ at t'. From (2) this implies that $\mathrm{UpToNow}\big(\bigwedge_{i \in \mathcal{I}_N} E_i\big)$ also holds at t'. Therefore, we exploit (1) to deduce that $\bigwedge_{i \in \mathcal{I}_N} M_i$ and $\mathrm{NowOn}\big(\bigwedge_{i \in \mathcal{I}_N} M_i\big)$ both hold at t' as well.

This provides the usual "temporal inductive step" as in all the other proofs for two modules. Therefore, we omit the remainder of the proof which amounts to the usual "non accumulation" argument, which is unaffected by the fact that we are now dealing with $N \geq 2$ modules. □

Since we now introduced self-discharging in the rule of Proposition 25, the only difference between Proposition 25 and the following is that we now have a non-fully compositional rule. Also notice that hypothesis (4) below could actually be relaxed to the same as in Proposition 25; however, we leave as it is to conform with the original rule 2 of Proposition 6.

Proposition 26 (Rely/Guarantee Circular Inference Rule 2N). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \twoheadrightarrow M_i$
2. $E \wedge \bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i$
3. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}\big(\bigwedge_{i \in \mathcal{I}_N} E_i\big) \vee \mathrm{UpToNow}\big(\bigwedge_{i \in \mathcal{I}_N} M_i\big)$

then $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$.

The following rules 1N(bis), 1N(ter), and 2N(bis) are all straightforward extensions, so we introduce them without any comment or proof.

Proposition 27 (Rely/Guarantee Circular Inference Rule 1N(bis)). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $\mathrm{SomP_e}(S) \Rightarrow (E_i \twoheadrightarrow M_i)$
2. $\mathrm{SomP_e}(S) \Rightarrow (\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i)$
3. $\mathrm{SomP_e}(S) \Rightarrow (\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M)$
4. $S \Rightarrow \mathrm{NowOn}(E_i) \vee \mathrm{NowOn}(M_i)$*, for all* $i \in \mathcal{I}_N$

then $\mathrm{SomP_e}(S) \Rightarrow (E \twoheadrightarrow M)$.

Proposition 28 (Rely/Guarantee Circular Inference Rule 2N(bis)). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E_i \twoheadrightarrow M_i)$
2. $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \wedge \bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i)$
3. $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M)$
4. $S \Rightarrow \mathrm{NowOn}\big(\bigwedge_{i \in \mathcal{I}_N} E_i\big) \vee \mathrm{NowOn}\big(\bigwedge_{i \in \mathcal{I}_N} M_i\big)$

then $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$.

Proposition 29 (Rely/Guarantee Circular Inference Rule 1N(ter)). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \twoheadrightarrow M_i$
2. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i$
3. $E \wedge \bigwedge_{i \in \mathcal{I}_N} M_i \twoheadrightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$*, for all* $i \in \mathcal{I}_N$

then $\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$.

A stronger time progression operator. The extension of rule 3 of Proposition 10 is also straightforward, and gives a rule which is very similar to that introduced in [FRMM06].

Proposition 30 (Rely/Guarantee Circular Inference Rule 3N). *If:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \gg M_i$
2. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow \bigwedge_{i \in \mathcal{I}_N} E_i$
3. $\bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M$
4. $S \Rightarrow \mathrm{AlwP_e}(E_i) \vee \mathrm{AlwP_e}(M_i)$*, for all* $i \in \mathcal{I}_N$

then $\mathrm{SomP_i}(S) \Rightarrow (E \gg M)$.

Implication as a compositional operator. The circular rule 4 of Proposition 12 requires some more work to be generalized. In this case, in fact, the soundness of the rule relies on a mutual interplay between the two modules, where one is required to have an assumption (or guarantee) behaving as a left-continuous state, and the other module to have it behaving as a right-continuous state.

The idea to generalize this is thus to implement the same interplay between two sets of all modules in the system. More precisely, we can assume that the set of all modules $\mathcal{I}_N$ can be partitioned into two non empty subsets $L, R \subset \mathcal{I}_N$ such that any module $i \in L$ (resp. $i \in R$) has its assumption E_i or its guarantee M_i which is a left-continuous (resp. right-continuous) state. Then, we require that the guarantees of the modules in L can discharge all the assumptions of the modules in R, and *vice versa*. All in all, we have the following inference rule.

Proposition 31 (Rely/Guarantee Circular Inference Rule 4N). *If, for two non empty subsets* $L, R \subset \mathcal{I}_N$ *that partition* $\mathcal{I}_N$*:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \Rightarrow M_i$
2. *(a)* $\bigwedge_{i \in L} M_i \Rightarrow \bigwedge_{i \in R} E_i$
   (b) $\bigwedge_{i \in R} M_i \Rightarrow \bigwedge_{i \in L} E_i$
3. $E \wedge \bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$*, for all* $i \in \mathcal{I}_N$
5. $E_i \in \mathsf{ST}_{\bullet\leftarrow}$ *or* $M_i \in \mathsf{ST}_{\bullet\leftarrow}$*, for all* $i \in R$
6. $E_i \in \mathsf{ST}_{\rightarrow\bullet}$ *or* $M_i \in \mathsf{ST}_{\rightarrow\bullet}$*, for all* $i \in L$

then $\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$.

Proof for dense time domains. Let t be the current instant, and let us assume that $\mathrm{SomP_i}(S)$ holds at t; thus, let $t' \leq t$ be (any) instant at which S held. We have to show that $E \twoheadrightarrow M$ holds at t.

As usual, we focus on showing one "temporal inductive step", the non accumulation argument being all similar to that of the previous proofs. More precisely, let us show that $\bigwedge_{i\in\mathcal{I}_N}(E_i \wedge M_i)$ and $\mathrm{NowOn}\big(\bigwedge_{i\in\mathcal{I}_N}(E_i \wedge M_i)\big)$ hold at t'.

First of all, let us show that $\bigwedge_{i\in L}\mathrm{UpToNow}(M_i)$. In fact, let i be any index $i \in L$; from (4) it is either $\mathrm{UpToNow}(E_i)$ or $\mathrm{UpToNow}(M_i)$ at t'. But, if $\mathrm{UpToNow}(E_i)$ for some i, then also $\mathrm{UpToNow}(M_i)$ from (1).

Next, from $\bigwedge_{i\in L}\mathrm{UpToNow}(M_i)$ and (2a) it follows that $\bigwedge_{i\in R}\mathrm{UpToNow}(E_i)$. Then, we consider (1) to deduce that $\bigwedge_{i\in R}\mathrm{UpToNow}(M_i)$. Thus, from (2b), also $\bigwedge_{i\in L}\mathrm{UpToNow}(E_i)$. So, all in all we have that $\bigwedge_{i\in\mathcal{I}_N}\mathrm{UpToNow}(E_i \wedge M_i)$ holds at t'.

For all $i \in L$, (6) lets us infer that also $E_i \vee M_i$ at t'. Moreover, from (1) we can actually state that $\bigwedge_{i\in L} M_i$ at t', and from (2a) it also follows that $\bigwedge_{i\in R} E_i$ at t'.

But then, for all $i \in R$, (5) — combined with (1) — lets us infer that also $\mathrm{NowOn}(M_i)$ at t'. Thus, also $\bigwedge_{i\in L}\mathrm{NowOn}(E_i)$ from (2b). (1) then implies that also $\bigwedge_{i\in L}\mathrm{NowOn}(M_i)$, and (2a) that $\bigwedge_{i\in R}\mathrm{NowOn}(E_i)$.

All in all, we have shown that $E_i \wedge M_i$ and $\mathrm{NowOn}(E_i \wedge M_i)$ both hold at t', for all $i \in \mathcal{I}_N$. The remainder of the proof is as for the other propositions. □

Proposition 32 (Rely/Guarantee Circular Inference Rule 4N(bis)). *If, for two non empty subsets $L, R \subset \mathcal{I}_N$ that partition $\mathcal{I}_N$:*

1. *for all $i \in \mathcal{I}_N$: $E_i \Rightarrow M_i$*
2. *(a) $\bigwedge_{i\in L} M_i \Rightarrow \bigwedge_{i\in R} E_i$*
   (b) $\bigwedge_{i\in R} M_i \Rightarrow \bigwedge_{i\in L} E_i$
3. $\mathrm{SomP_i}(S) \Rightarrow (E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M)$
4. $S \Rightarrow \mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i)$, *for all $i \in \mathcal{I}_N$*
5. $E_i \in \mathsf{ST}_{\bullet\leftarrow}$ *or* $M_i \in \mathsf{ST}_{\bullet\leftarrow}$, *for all $i \in R$*
6. $E_i \in \mathsf{ST}_{\rightarrow\bullet}$ *or* $M_i \in \mathsf{ST}_{\rightarrow\bullet}$, *for all $i \in L$*

then $\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$.

A circular inference rules without compositional operator. The extension of rule 5 of Proposition 14 is along the same lines of those we have just discussed. Moreover, as it was the case for other rules, the initialization formula could be made less restrictive at the price of complicating the discharging formulas. Indeed, in the version we present, the discharging formulas are not needed at all in the soundness proof.

Proposition 33 (Rely/Guarantee Circular Inference Rule 5N). *If, for some fixed duration* $\mathrm{T_B} > 0$*:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \Rightarrow M_i$
2. $E \wedge \bigwedge_{i \in \mathcal{I}_N} M_i \Rightarrow M$
3. $\mathrm{WithinF}\big(\bigwedge_{i \in \mathcal{I}_N} (E_i \vee M_i), \mathrm{T_B}\big)$

then $\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$.

Proof. Let t be a generic time instant at which $\mathrm{Lasts}(E, \mathrm{T_B})$ holds, and prove that $\mathrm{WithinF}(M, \mathrm{T_B})$.

From (3), let us assume that $\mathrm{WithinF}\big(\bigwedge_{i \in \mathcal{I}_N} M_i, \mathrm{T_B}\big)$ at t. This is without loss of generality: in fact, if $\mathrm{WithinF}(E_i, \mathrm{T_B})$ for some $i \in \mathcal{I}_N$, then $\mathrm{WithinF}(M_i, \mathrm{T_B})$ from (1). Thus, there exists a t' such that $t < t' < t + \mathrm{T_B}$ and $\bigwedge_{i \in \mathcal{I}_N} M_i$ holds at t'. Moreover, since $t < t' < t + \mathrm{T_B}$ and $\mathrm{Lasts}(E, \mathrm{T_B})$ at t, then E also holds at t'. But then, M holds at t' by (3). Since $t < t' < \mathrm{T_B}$, this implies that $\mathrm{WithinF}(M, \mathrm{T_B})$ at t. □

Notice the following: the above rule works for any variation of the Lasts and WithinF operators, provided the two variations coincide. In other words, the above rule works for any choice of $\mathrm{Lasts}_{\mathrm{lr}}$ and $\mathrm{WithinF}_{\mathrm{lr}}$ operator, where $l, r \in \{e, i\}$, provided l and r are the same in the two operators.

Completeness. For the sake of brevity, we do not repeat the (in)completeness analysis performed in Section 4.3 on the generalized rules for $N \geq 2$ modules. Nonetheless, it is not difficult to realize that the very same results can be lifted from the two module to the N module case. In fact, firstly an incompleteness proof for a two module system implies the incompleteness of the generalization of the same rule (i.e., of the rule which reduces to the two module one if we choose $N = 2$). Secondly, the completeness proofs can be extended to the N module case by suitably instantiating all the E_i's and M_i's for $i > 2$ with the same values of E_2, M_2 in the two module case.

Summary. Table 5 summarizes the circular inference rules for N modules.

6 An Illustrative Example: the BitTorrent Protocol

This section illustrates our practical approach to compositionality through an example: the compositional verification of the peer to peer communication protocol BitTorrent (BT) [Coh01, Coh03]. Actually, we model and verify only a very small part of the behaviors allowed by the complex protocol, in a specific situation — which is nonetheless likely to happen in practice —, and we provide

Rule 1N (Prop. 25 I)	**Rule 1N(bis) (Prop. 27 I)**	**Rule 1N(ter) (Prop. 29 C)**
$\forall i \in \mathcal{I}_N : (E_i \twoheadrightarrow M_i)$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i))$	$\mathrm{SomP_e}(S) \Rightarrow \forall i \in \mathcal{I}_N : (E_i \twoheadrightarrow M_i)$ $\mathrm{SomP_e}(S) \Rightarrow (\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i)$ $\mathrm{SomP_e}(S) \Rightarrow (\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M)$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{NowOn}(E_i) \vee \mathrm{NowOn}(M_i))$	$\forall i \in \mathcal{I}_N : (E_i \twoheadrightarrow M_i)$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i$ $E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \twoheadrightarrow M$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i))$
$\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_e}(S) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$
Rule 2N (Prop. 26 I)	**Rule 2N(bis) (Prop. 28 C)**	**Rule 3N (Prop. 30 I)**
$\forall i \in \mathcal{I}_N : (E_i \twoheadrightarrow M_i)$ $E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M$ $S \Rightarrow \mathrm{UpToNow}\Big(\bigwedge_{i\in\mathcal{I}_N} E_i\Big) \vee \mathrm{UpToNow}\Big(\bigwedge_{i\in\mathcal{I}_N} M_i\Big)$	$\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow \forall i \in \mathcal{I}_N : (E_i \twoheadrightarrow M_i)$ $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i)$ $\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M)$ $S \Rightarrow \mathrm{NowOn}\Big(\bigwedge_{i\in\mathcal{I}_N} E_i\Big) \vee \mathrm{NowOn}\Big(\bigwedge_{i\in\mathcal{I}_N} M_i\Big)$	$\forall i \in \mathcal{I}_N : (E_i \gg M_i)$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{AlwP_e}(E_i) \vee \mathrm{AlwP_e}(M_i))$
$\mathrm{SomP_i}(S) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_e}(S) \wedge \mathrm{AlwP_e}(E) \Rightarrow (E \twoheadrightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \gg M)$
Rule 4N (Prop. 31 I)	**Rule 4N(bis) (Prop. 32 C)**	**Rule 5N (Prop. 33 C)**
$\forall i \in \mathcal{I}_N : (E_i \Rightarrow M_i)$ $\bigwedge_{i\in L} M_i \Rightarrow \bigwedge_{i\in R} E_i$, $\bigwedge_{i\in R} M_i \Rightarrow \bigwedge_{i\in L} E_i$ $E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i))$ $\forall i \in R : E_i$ or $M_i \in \mathsf{ST}_{\bullet\leftarrow}$, $\forall i \in L : E_i$ or $M_i \in \mathsf{ST}_{\rightarrow\bullet}$	$\forall i \in \mathcal{I}_N : (E_i \Rightarrow M_i)$ $\bigwedge_{i\in L} M_i \Rightarrow \bigwedge_{i\in R} E_i$, $\bigwedge_{i\in R} M_i \Rightarrow \bigwedge_{i\in L} E_i$ $\mathrm{SomP_i}(S) \Rightarrow (E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M)$ $S \Rightarrow \forall i \in \mathcal{I}_N : (\mathrm{UpToNow}(E_i) \vee \mathrm{UpToNow}(M_i))$ $\forall i \in R : E_i$ or $M_i \in \mathsf{ST}_{\bullet\leftarrow}$, $\forall i \in L : E_i$ or $M_i \in \mathsf{ST}_{\rightarrow\bullet}$	$\forall i \in \mathcal{I}_N : (E_i \Rightarrow M_i)$ $\bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow \bigwedge_{i\in\mathcal{I}_N} E_i$ $E \wedge \bigwedge_{i\in\mathcal{I}_N} M_i \Rightarrow M$ $\mathrm{WithinF}\Big(\bigwedge_{i\in\mathcal{I}_N} (E_i \vee M_i), \mathrm{T_B}\Big)$
$\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$	$\mathrm{SomP_i}(S) \Rightarrow (E \Rightarrow M)$	$\mathrm{Lasts}(E, \mathrm{T_B}) \Rightarrow \mathrm{WithinF}(M, \mathrm{T_B})$

Table 5: Circular compositional inference rules for $N \geq 2$ modules.

a high-level description which abstracts away from most implementation details, focusing on timing aspects.[9]

The overall goal of this example is not only to show the application of the inference rules in practice, but also to demonstrate our general approach to compositionality, as laid out in the previous sections and in [FRMM06]. To this end, we provide two different, alternate verifications of the protocol. The first approach is the most straightforward and standard: we select a suitable inference rule, among those provided in the previous sections, and we apply it to verify the system. Although compositionality succeeds in shifting the most part of the verification burden from global to local, the application of the inference rules is nonetheless highly nontrivial, and requires some considerable ingenuity.

The second approach is instead more unconventional, in that it does not use any previously defined inference rule. On the contrary, it builds a new inference rule by varying the existing rules in a way which fits the particular specification we are trying to verify. In this case the verification effort required in applying the new rule is minimal, although this is of course traded-off against ingenuity required in designing the new inference rule, and in verifying its correctness.

Then, we conclude this section with a brief comparison between the two approaches that highlights the advantages and disadvantages of both. We believe that our example, however small, constitutes an evidence of the importance of having a flexible approach to compositionality, and of the necessity of focusing on real examples of applications on a case-by-case basis, rather than looking for a "one-size-fit-all" compositional rule.

In practice, the example is carried out as follows. After providing suitable axiomatic specification of the classes involved in describing the protocol, we state a global theorem that we want to prove about the composition of the various classes. Then, we provide two different proofs of the theorem, both according to the general methodology of [FRMM06]. The first proof is an application of the inference rule 5N of Proposition 33, that we have discussed above. The second proof, instead, pursues in full the practical approach to compositionality that we have advocated in Section 2.2. Therefore, instead of figuring out how to fit our system specification within a predefined inference rule, we develop a new rule, combining the known results about the previously introduced rules with the specific structure of the formulas that model our system.

6.1 Protocol Basics

Let us describe the basic features of the BT protocol, namely those that we are going to specify and use for verification.

BT is a peer-to-peer communication protocol for distributing files over networks among a number of hosts. Hosts are partitioned among seeds and peers. Let N_s be the number of *seeds*, and N_p be the number of *peers* in the system.

[9] For another, completely different example of verification of the BitTorrent protocol, based on the approximation of process algebras through differential equations, we refer the reader to [Dug06].

Usually, it is $N_p \gg N_s$, although this is not necessary for the correct functioning of the protocol.

When the protocol is started, each seed possesses an entire copy of the file to be distributed; let us represent the file as split into P packets, numbered from 0 to $P-1$. The peers, instead, do not initially have any part of the file; their goal is to get it. In order to do that, each peer sends periodically a (random) packet to any peer it is connected to. After a peer has received (and thus stored) at least a packet, it also periodically sends a packet to any other peer it is connected to. The connections among peers and between seeds and peers change dynamically and are coordinated by a dedicated host called *tracker*. For simplicity, we do not represent explicitly the tracker in our model, but simply specify minimal assumptions about the dynamics of connections between hosts.

Let us describe how the connections between hosts are managed in our model; recall that this is a special case of the real BT protocol. At any time, the set of all peers is partitioned into a number of clusters. In each cluster, the peers are connected in a circle, and there is at least one peer which is connected to a seed and can receive data from it. Moreover, the connections change only periodically. More precisely, we will ensure that, before the clusters are changed, every peer in the cluster has received at least a packet (that is there has been at least a "passing around" of packets over the circle of peers in the cluster).

Notice that the connections do not represent physical connections, which would hardly be dynamic; instead, they represent coordination between peers. Moreover, we neglect the transmission time over the connections, which are considered instantaneous (this is acceptable, since the transmission times can be modeled directly in the specification of each host as re-transmission delays).

6.2 System Specification

Let us now introduce the classes of our BT specification. Notice that in the remainder we describe a real-time strengthening of the real BT protocol; the latter does not have any hard real time constraint built in, as it relies on the TCP/IP protocol which is "best-effort".

The seed class. The seeds in the system are represented by instances of the **seed** class, which is pictured in Figure 1. This class is parametric with respect to the number P of packets of the file, and the send time T_s. Moreover, it has a single event item $\mathsf{send}(i)$, which is true whenever the seed sends the packet number $i \in \{0, \ldots, P-1\}$ over its connections. The axiomatization of the class is very simple: every T_s time units (at most), a packet is nondeterministically sent over the connections. This is represented by the following axiom.[10]

Axiom 1 (seed.sending). $\exists i : \mathrm{WithinF}(\mathsf{send}(i), T_s)$

[10] In the remainder of the specification, we do not mention explicitly the types of the predicates arguments whenever this is unambiguous. For example, we simply write $\mathsf{send}(i)$ assuming implicitly that $i \in \{0, \ldots, P-1\}$.

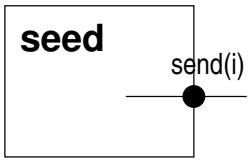

Figure 1: Interface of the **seed** class

The peer class. The peers are represented by instances of the **peer** class, pictured in Figure 2. The **peer** class is also parametric with respect to the number P of packets of the file, and the send time T_s. It has a send event, representing its sending a packet over its connection, and a recv event, representing the receiving of a packet (from another peer or from a seed). Moreover, the class has a non-visible stored state: $\mathsf{stored}(i)$ being true represents the fact that the packet i has been received and is therefore stored locally; this is formalized by the following axioms (i is a variable of type $\{0, \ldots, P-1\}$).

Axiom 2 (peer.sending). $\mathsf{stored}(i) \Leftrightarrow \mathrm{SomP_i}(\mathsf{recv}(i))$

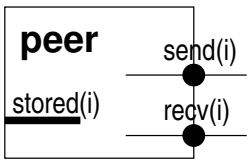

Figure 2: Interface of the **peer** class

The send and recv actions happen according to axioms **peer.recv_to_send** and **peer.send_to_recv**: the former states that a recv triggers a send of a (nondeterministically chosen) packet within T_s time units; the latter conversely states that a send only happens if a packet has been received in the past, and the packet is stored locally.

Axiom 3 (peer.recv_to_send). $\exists i : \mathsf{recv}(i) \Rightarrow \exists j : \mathrm{WithinF}(\mathsf{send}(j), T_s)$

Axiom 4 (peer.send_to_recv). $\mathsf{send}(i) \Rightarrow \mathsf{stored}(i) \wedge \exists j : \mathrm{WithinP}(\mathsf{recv}(j), T_s)$

The system class. Now, let us describe how the modules are composed into a global system, representing the network. The seeds are N_s instances of the **seed** class that populate an array named Se, and the peers are N_p instances of the **peer** class that populate an array named Pe.

To represent the connections among hosts in the system, we cannot exploit TRIO connections which are static. On the contrary, we introduce two state predicates $\mathsf{connected_p}(p_1, p_2)$ and $\mathsf{connected_s}(s, p)$ to indicate connections between two peers, and between a seed and a peer, respectively. The meaning of the two predicates is obvious: whenever i is connected to j, it means that what

i sends, it is received by j. This is stated by the global axioms **sp_connections** and **pp_connections**.

Axiom 5 (sp_connections). $\mathsf{connected_s}(s,p) \Rightarrow (\mathsf{Se}[s].\mathsf{send}(i) = \mathsf{Pe}[p].\mathsf{recv}(i))$

Axiom 6 (pp_connections). $\mathsf{connected_p}(p_1,p_2) \Rightarrow (\mathsf{Pe}[p_1].\mathsf{send}(i) = \mathsf{Pe}[p_2].\mathsf{recv}(i))$

Finally, we have to define how the connections change over time. This is an important part of the specification, and requires some nontrivial details.

Let $K = \{0,\ldots,N_p-1\}$ be the set of all peers. We postulate that, at any time, there exist k (time-dependent) sets $K_1,\ldots,K_k$, named *clusters*, such that:

1. The clusters form a partition of K;
2. Each cluster is such that all its peers are connected to form a ring;
3. For each cluster, there is *at least* a peer in the cluster which is connected to some seed.

Conditions C(1–3) are formalized by axiom **clustering** below. In order to present it, we first introduce the following notation.[11]

- The successor of index p modulo m (i.e., over a set of m elements numbered from 0 to $m-1$) is denoted by $\mathrm{next}_m(p) \equiv (p+1) \bmod m$.
- We denote a permutation of the m-element set $\{0,\ldots,m-1\}$ as a function $\pi_m : \{0,\ldots,m-1\} \to \{0,\ldots,m-1\}$.
- We associate to every cluster K_i its size $k_i \equiv |K_i|$.

Axiom 7 (clustering).

$$\exists 0 < k \le N_p : \exists K_1,\ldots,K_k :$$

$$\textit{(C1)} \left(\begin{array}{l} \forall i \in \{1,\ldots,k\} : K_i \subseteq K \ \wedge \bigcup_{i=1,\ldots,k} K_i = K \\ \qquad \wedge\ \forall i,j \in \{1,\ldots,k\} : i \neq j \Rightarrow K_i \cap K_j = \emptyset \end{array} \right) \wedge$$

$$\textit{(C2)} \left(\begin{array}{l} \forall i \in \{1,\ldots,k\} : \exists \widetilde{\pi}_{k_i} : \forall j \in \{0,\ldots,k_i-1\} : \\ \qquad \mathsf{connected_p}(\widetilde{\pi}_{k_i}(j), \widetilde{\pi}_{k_i}(\mathrm{next}_{k_i}(j))) \end{array} \right) \wedge$$

$$\textit{(C3)} \left(\begin{array}{l} \forall i \in \{1,\ldots,k\} : \exists j \in \{0,\ldots,k_i-1\} : \\ \qquad \exists s \in \{0,\ldots,N_s-1\} :\ \mathsf{connected_s}(s,j) \end{array} \right)$$

Notice that, from now on, with a little abuse of notation, we will treat the K_i's as globally available items, that is we will be able to reference them in any module of the system; notice that this is only a shortcut to represent easily the sets, whose precise formalization in TRIO would not be difficult but would introduce some additional notational overhead.

[11] In the following axiom, we slightly abuse TRIO notation by using higher-order quantifications, namely on the functions/sets K_i and π_{k_i}. Notice, however, that the same meaning can be retained with suitable first-order predicates and quantifications, but with a much more cumbersome notation, which we prefer to avoid for clarity.

6.3 Rely/Guarantee Specifications

Let us now describe a global specification property of the system. Then, we also provide suitable specifications local to each peer module, from which the correctness of the global specification can be inferred compositionally.

The form of the local specifications depends on the inference rule that we intend to use. Therefore, in this section we first of all introduce the global specification, and then we provide two different sets of local specifications: Section 6.3.2 describes those suitable to apply the inference rule of Proposition 33, whereas the following Section 6.3.3 introduces a new *ad hoc* inference rule, and shows how the basic specification of the system can be seen as a local rely/guarantee specification suitable to be used with this new inference rule.

6.3.1 Global Specification

Our overall verification goal is to show a liveness property about the peers, that is that every peer eventually sends a packet, within an upper bound equal to the value $N_p T_s$. We choose this particular property as it allows us to focus on temporal properties, and thus to demonstrate best the use of our inference rules for temporal logic. An orthogonal concern, which we omit for brevity, would be the proof of the the correctness of the protocol. However, let us notice that similar, compositional techniques could be used for this purpose as well.

Intuitively, such a property cannot be guaranteed without introducing constraints on how the connections among peers vary. Basically, we need that the connections do not change "too often", if we want the system to actually converge towards some goal. More precisely, let us introduce a predicate $\mathrm{const}(p, t)$ which holds whenever its first argument keeps its current value over an interval of given length given by its second argument. Therefore, for any time-dependent predicate F with domain D, we have the following definition (as usual, *bb* can be any of *ee*, *ei*, etc.).

$$\mathrm{const}_{\mathrm{bb}}(F, T) \;\equiv\; \exists d \in D : \mathrm{Lasts}_{\mathrm{bb}}(F(d), T)$$

Through this notation, we formulate the following global specification in a rely/guarantee form: if the connections stay unchanged over a time interval of length $N_p T_s$, then every peer eventually sends a packet.

Theorem 8 (global_send_liveness).

$$\mathrm{const}_{\mathrm{ie}}(\mathsf{connected_p}(p_1, p_2) \wedge \mathsf{connected_s}(s, p), N_p T_s) \quad \Rightarrow \\ \mathrm{WithinF}_{\mathrm{ie}}(\exists i : \mathsf{Pe}[p].\mathsf{send}(i)\,, N_p T_s)$$

6.3.2 Local Specifications for a Predefined Inference Rule

In order to apply the inference rule 5N of Proposition 33, we now set up suitable rely/guarantee specifications, local to each **peer** class.

The following theorem **peer.bounded_send_recv** provides such a specification. As usual, $i, j \in \{0, \ldots, P-1\}$ and k indicates the size of the cluster the current module is in.

Theorem 9 (peer.bounded_send_recv).
$\exists j : \text{WithinF}(\mathsf{recv}(j), kT_s) \Rightarrow \exists j : \text{WithinF}(\mathsf{send}(j), kT_s)$

This formula is probably a bit counterintuitive, since it establishes a relationship between two events that are both in the future. However remember that we are dealing with an abstract specification, and this description is therefore logically acceptable.

Now that we have introduced the local rely/guarantee specification through theorem **peer.bounded_send_recv**, we have to prove it by means of the other formulas of the **peer** class, to show that it is indeed a valid local specification. In order to do that, we introduce an assumption formula which we will discharge by means of axioms of other **peer** classes. The formula, named **peer.late_recv**, states that if a late (i.e., later than $(k-1)T_s$ time units) recv is received, a send is also present within the overall kT_s time period.

Assumption 10 (peer.late_recv).
$\exists j : \text{Futr}(\text{WithinF}(\mathsf{recv}(j), T_s), (k-1)T_s) \Rightarrow \exists j : \text{WithinF}(\mathsf{send}(j), kT_s)$

The need for this assumption will be clear in the following proof.

*Proof of Theorem **peer.bounded_send_recv**.* For simplicity, let 0 be the current time instant. Assume the antecedent of the implication holds, that is that $\mathsf{recv}(i)$ holds at a time instant $0 < t < kT_s$ from the current time instant, for any i. We distinguish two cases: whether $t > (k-1)T_s$ or not.

In the former case, $\mathsf{recv}(i)$ holds at a time instant $(k-1)T_s < t < T_s$. This means that at $(k-1)T_s$ the formula $\text{WithinF}(\mathsf{recv}(i), T_s)$ holds, which corresponds to the antecedent of formula **peer.late_recv**. Therefore the consequent holds, which is what we have to prove.

The latter case is the one in which the time instant in which $\mathsf{recv}(i)$ holds is $t \leq (k-1)T_s$. Let us consider axiom **peer.recv_to_send** at time t. We can infer that $\text{WithinF}(\mathsf{send}(j), T_s)$ at t, for some j. In other words, there is a time instant $t < u < t + T_s$ such that $\mathsf{send}(j)$ at u. It is simple to realize that $u < (k-1)T_s + T_s = kT_s$, in this branch of the proof. Therefore, it is true that $\text{WithinF}(\mathsf{send}(j), kT_s)$ at the current time. □

6.3.3 Making Up a New Inference Rule

In accordance with the guidelines we have proposed in the previous sections, let us find suitable rely/guarantee inference rule and local specifications according to try to the goal of our verification, stated in Section 6.3.1.

Local specifications. Let us seek a suitable formula to serve as local specification of the modules. More precisely, we need a specification for each peer, as we have many of them in each cluster. We notice that axiom **peer.recv_to_send**

is a local specification about the behavior of the send item in response to a behavior of the recv. Therefore, it is a suitable rely/guarantee local specification.

A suitable compositional inference rule. Let us now build a suitable inference rule to prove the truth of the global specification theorem **globalsend_liveness** from the truth of the local specifications axioms **peer.recv_to_send**. Let us summarize the characteristics that it should possess.

- it should be a rely/guarantee rule, as we have both a global specification and the local specifications that are in rely/guarantee form;
- the link between the assumptions and the guarantees of the formulas should be given by a temporal relationship expressed through the WithinF operator;
- the rule must be circular, as this mirrors the circular nature of the connections among the peers in each cluster, each of whose assumption can be satisfied by the preceding peer;
- the rule need not be self-discharging, as the assumption of each module is discharged solely by the guarantee of the module that precedes it;
- it must be non fully compositional, as the global assumption about the constancy of the connections over time defines what is the state of the connections, and thus it is required in discharging local assumptions;
- there is no notion of initialization required, since we are proving a "liveness" property, similarly as in the inference rule of Proposition 33.

All in all, let us introduce — and prove the soundness of — the following circular inference rule, whose structure of assumptions and guarantee mirrors closely the one of the local and global specification formulas in our example.

Proposition 34. *If, for some fixed duration* $\mathrm{T_B} > 0$*:*

1. *for all* $i \in \mathcal{I}_N$*:* $E_i \Rightarrow \mathrm{WithinF}(M_i, \mathrm{T_B})$
2. *for all* $i \in \mathcal{I}_N$*:* $E \wedge M_i \Rightarrow E_{\texttt{if } i<N \texttt{ then } i+1 \texttt{ else } 1}$
3. $\bigwedge_{i \in \mathcal{I}_N} \mathrm{WithinF}(M_i, N\mathrm{T_B}) \Rightarrow M$
4. $\mathrm{WithinF}(E_i, \mathrm{T_B})$*, for some* $i \in \mathcal{I}_N$

then $\mathrm{Lasts}(E, N\mathrm{T_B}) \Rightarrow M$*.*

Proof. Let us assume that $\mathrm{Lasts}(E, N\mathrm{T_B})$ holds at the current time t, that is E holds throughout the interval $(t, t+N\mathrm{T_B})$. By (4), there exists a $t < t_1 < t+\mathrm{T_B}$ such that E_i holds at t' for some i. Without loss of generality, let $i = N$: this amounts just to a re-numbering of the modules.

Next, let us show that there exists N instants $t_1, t_2, \ldots, t_N$ such that M_i holds at t_i, and $t < t_i < t + i\mathrm{T_B}$, for all $i \in \mathcal{I}_N$. The proof goes by (finite)

induction. The base case $i = 1$ is obvious: since E_N at t_1, then also M_1 at t_1 by (2), and $t < t_1 < t + \mathrm{T_B}$ by assumption.

For the inductive step, assume that M_{i-1} holds at $t < t_{i-1} < t + (i-1)\mathrm{T_B}$; notice that E holds at t_{i-1} as well. Therefore, E_i holds at t_{i-1} from (2). Let us evaluate (1) at t_{i-1}; it follows that there exists a $t_{i-1} < t_i < t_{i-1} + \mathrm{T_B}$ such that M_i holds at t_i. Given the bounds on t_{i-1}, it is clear that $t < t_{i-1} < t_i < t_{i-1} + \mathrm{T_B} < t + (i-1)\mathrm{T_B} + \mathrm{T_B} = t + i\mathrm{T_B}$. This concludes the induction.

All in all we have shown that $\bigwedge_{i \in \mathcal{I}_N} \mathrm{WithinF}(M_i, N\mathrm{T_B})$ holds at t. (3) lets us conclude that M holds at t. □

6.4 Application of the Compositional Rule

In this section, we provide the proof of theorem **global_send_liveness** through the applications of compositional inference rules with the local rely/guarantee specifications we have presented in the previous subsection. As we already mentioned, we provide two such proofs. The first one is the object of Section 6.4.1 and uses inference rule 5N of Proposition 33; the second one is the object of Section 6.4.2 and exploits the inference rule of Proposition 34 introduced beforehand. We warn the reader that the very final part of the proof in Section 6.4.1 will actually be also used in the other proof of Section 6.4.2, so we do not repeat this part in the latter section.

6.4.1 Using a Predefined Inference Rule

Our goal is to compose the various theorems **peer.bounded_send_recv** (one for each instance of the **peer** class) using the compositional inference rule of Proposition 33 to deduce the validity of the global specification.

In our system, we can identify two levels at which to prove the liveness property for the peers: intra-cluster and inter-cluster. To wit: since at any time the set of all peers is divided into clusters, we can perform the global proof in two steps:

- first, prove the liveness property within each cluster, independently of the others;
- then, show that the composition of the intra-cluster liveness properties yields the overall desired liveness property of the system.

Therefore, we are going to first apply the inference rule to each cluster separately.

Intra-cluster proofs. Consider the (generic) cluster of size k and the the following choices for assumption and guarantee formulas. Notice that modules in the inference rule are numbered from 1, while peers (and seeds) are numbered from 0. However, for simplicity, we now assume to have a 0-based numbering in the modules: filling the gap is straightforward.[12]

[12] We add the superscript k for notational clarity.

- $E_i^k = \exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{recv}(j)\,, kT_s)$;
- $M_i^k = \exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{send}(j)\,, kT_s)$;
- $\text{Lasts}_{\text{ie}}\big(E^k, T_s\big) = \text{const}_{\text{ie}}(\mathsf{connected_p}(p_1, p_2) \wedge \mathsf{connected_s}(s, p), kT_s)$, for all p_1, p_2, p in the cluster and some seed s;
- $M^k = \bigwedge_{i=1,\ldots,k} M_i^k$;
- $N = k$ and $T_B^k = kT_s$.

Let us now apply the inference rule of Proposition 33 for the Lasts_{ie} and $\text{WithinF}_{\text{ie}}$ operators. Therefore, the we split the proof into lemmas, each corresponding to a condition of the inference rule.

Lemma 35 (Proof step 1).
$\exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{recv}(j)\,, kT_s) \Rightarrow \exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{send}(j)\,, kT_s)$

Lemma 36 (Proof step 2).
$E^k \wedge \bigwedge_i \exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{send}(j)\,, kT_s) \Rightarrow \bigwedge_i \exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{send}(k)\,, kT_s)$

Lemma 37 (Proof step 3).
$\text{WithinF}_{\text{ie}}(\bigwedge_i((\exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{recv}(j)\,, kT_s)) \vee (\exists j : \text{WithinF}(\mathsf{Pe}[i].\mathsf{send}(j)\,, kT_s))), kT_s)$

Let us now consider the proofs of the lemmas. Notice that, in proving each lemma, we can safely assume that we are in a time interval where $\text{Lasts}_{\text{ie}}\big(E^k, kT_s\big)$ holds, that is where the connections are constant. In fact, the conclusion of the whole compositional proof holds under the condition $\text{Lasts}_{\text{ie}}\big(E^k\big)\, kT_s$. Now, for the proofs.

Proof of Lemma 35. For every i in a cluster of size k, this lemma is equivalent to theorem **peer.bounded_send_recv**. □

Proof of Lemma 36. Trivial, since the consequent is immediately subsumed by the antecedent. □

Modular distance and properties. The proof of Lemma 37 is more complicated than the others, and requires an additional definition for a *distance* between two peers in the cluster, together with a property of this new item. The property is not hard to prove, but it requires a somewhat involved case discussion.

Let us first define the distance between two peers p_1, p_2 in a cluster as: $\text{dist}_N(p_1, p_2) \equiv (p_2 - p_1) \bmod N$. Then, we have the following property.

Lemma 38.

$$\text{dist}_N(p_1, \text{next}_N(p_2)) = \begin{cases} 0 & \textit{if } \text{next}_N(p_2) = p_1 \\ \text{dist}_N(p_1, p_2) + 1 & \textit{otherwise} \end{cases}$$

Proof. Let us first consider the simple case $\text{next}_N(p_2) = p_1$. We then conclude, by the definition of distance, $\text{dist}_N(p_1, \text{next}_N(p_2)) = \text{dist}_N(p_1, p_1) = 0$.

Let us now take the other case $\text{next}_N(p_2) \neq p_1$. Considering the definitions of distance and next element, we have to prove

$$(((p_2 + 1) \bmod N) - p_1) \bmod N = ((p_2 - p_1) \bmod N) + 1. \tag{1}$$

The proof is split into two branches, whether $p_2 + 1 < N$ or not.

1. $p_2 + 1 < N$. In this case, (1) rewrites to

$$((p_2 + 1) - p_1) \bmod N = ((p_2 - p_1) \bmod N) + 1 \tag{2}$$

   Let us distinguish whether $p_2 - p_1 + 1 \geq 0$ or not.

   (a) $p_2 - p_1 \geq -1$. Notice that $p_2 - p_1 = -1$ iff $\text{next}_N(p_2) = p_1$, so we have $p_2 - p_1 \geq 0$ in this branch of the proof. Hence, (2) is established

$$p_2 - p_1 = (p_2 - p_1) \bmod N = p_2 - p_1$$

   (b) $p_2 - p_1 < -1$. Notice that, for any natural number $k \in [1..N-1]$, we have $-k \bmod N = N - k$. So, we can rewrite (2) as

$$N - (-p_2 + p_1 - 1) = (N - (-p_2 + p_1)) + 1$$

   which concludes this branch of the proof.

2. $p_2 + 1 \geq N$. It is simple to realize that it must be $p_2 + 1 = N$, since $p_2 < N$. Hence, (1) becomes

$$(0 - p_1) \bmod N = N - p_1 = ((N - 1 - p_1) \bmod N) + 1$$

   It is also obvious that $N - 1 - p_1 \geq 0$, so we finally conclude

$$N - p_1 = (N - 1 - p_1) + 1$$

□

To simplify the notation, we also introduce a notion of distance among peers in a cluster that refers to the permutation of the indexes induced by the connections. Namely, the distance between two peers with respect to a permutation is given by the distance between their indexes in the permutation and denoted as: $\text{dist}_{\pi_N}(p_1, p_2) \equiv \text{dist}_N(\pi_N^{-1}(p_1), \pi_N^{-1}(p_2))$. Consequently, a similar notion for the next peer is introduced. That is, the next of a peer with respect to a permutation is given by the element next to the peer in the permutation, and it is denoted as: $\text{next}_{\pi_N}(p) \equiv \pi_N(\text{next}_N(\pi_N^{-1}(p)))$. Notice that Lemma 38 can be shown to hold also for the functions $\text{dist}_{\pi_N}(p_1, p_2)$ and $\text{next}_{\pi_N}(p)$ we have just defined; we omit the straightforward proof of this fact.

Proof of Lemma 37. We still need to prove an intermediate lemma, before actually performing the proof of Lemma 37. Notice that, for the sake of brevity, we now write simply $\mathrm{dist}(p_1, p_2)$ and $\mathrm{next}(p)$ instead of $\mathrm{dist}_{\widetilde{\pi}_k}(p_1, p_2)$ and $\mathrm{next}_{\widetilde{\pi}_k}(p)$ respectively, whenever the permutation $\widetilde{\pi}_k$ is understood to be the one defined by axiom **clustering** at a given time instant.

Lemma 39 (futr_recv). $\mathsf{Pe}[p_1].\mathsf{recv}(i) \Rightarrow \forall 0 < d < k_i : \forall p_2 : (\mathrm{dist}(p_1, p_2) = d \Rightarrow \exists j : \mathrm{WithinF}(\mathsf{Pe}[p_2].\mathsf{recv}(j), dT_s))$

Proof. The proof goes by induction on the distance $d > 0$.

The *base case* $d = 1$ requires us to show that, for all p_2 such that $\mathrm{dist}(p_1, p_2) = 1$, $\mathsf{Pe}[p_1].\mathsf{recv}(i) \Rightarrow \exists i : \mathrm{WithinF}(\mathsf{Pe}[p_2].\mathsf{recv}(i), T_s)$. Now, as a consequence of axiom **peer.recv_to_send**, it follows from $\mathsf{Pe}[p_1].\mathsf{recv}(i)$ that $\mathrm{WithinF}(\mathsf{Pe}[p_1].\mathsf{send}(j), T_s)$ for some packet j. It is not hard to realize that, if $\mathrm{dist}(p_1, p_2) = 1$, then $p_2 = \mathrm{next}(p_1)$. Moreover, we are considering a time interval in the future where the connections are stable, so in particular we have that $\mathsf{connected_p}(p_1, p_2)$, that is $\mathsf{Pe}[p_1].\mathsf{send}(j)$ is equivalent to $\mathsf{Pe}[p_2].\mathsf{recv}(j)$. In other words, we have shown that $\mathrm{WithinF}(\mathsf{Pe}[p_2].\mathsf{recv}(j), T_s)$, concluding the base step.

Now, let us consider a $d > 1$. The *inductive step* requires us to assume that the property holds for d and to show that it holds for $d + 1$ as a consequence. So, let us consider two generic peers such that $\mathrm{dist}(p_1, p_2) = d + 1$, and assume that $\mathsf{Pe}[p_1].\mathsf{recv}(i)$ for some packet i. Let us also consider the peer p_3 such that $\mathrm{next}(p_3) = p_2$, that is the peer immediately preceding p_2 in the ring of connections. We can apply Lemma 38 to show that either $\mathrm{dist}(p_1, \mathrm{next}(p_3)) = \mathrm{dist}(p_1, p_2) = 0$ or $\mathrm{dist}(p_1, p_2) = \mathrm{dist}(p_1, p_3) + 1$. However, it is not the case that $p_2 = p_1$, otherwise it would be $\mathrm{dist}(p_1, p_2) = 0 < 1$. Hence $\mathrm{dist}(p_1, p_3) = d$.

The inductive step lets us imply that $\exists j : \mathrm{WithinF}(\mathsf{Pe}[p_3].\mathsf{recv}(j), dT_s)$, assuming $\mathsf{Pe}[p_1].\mathsf{recv}(i)$. Considering the definition of the WithinF operator, and assuming 0 as the current time, this means that there exists a time instant $0 < t < dT_s$ time units in the future,when $\mathsf{Pe}[p_3].\mathsf{recv}(j)$ holds. With a proof similar to that of the base case, considering the fact that $\mathsf{connected_p}(p_3, p_2)$ we can show that there exists a time instant q, occurring no later than T_s time units after t, when $\mathsf{Pe}[p_2].\mathsf{recv}(l)$ for some packet l. Since $q - t < T_s$, then $0 < q < (d+1)T_s$ and therefore we can write $\mathrm{WithinF}(\mathsf{Pe}[p_3].\mathsf{recv}(l), (d+1)T_s)$, thus establishing the property for $d+1$ and concluding the inductive proof. □

We are now ready for the final proof.

Proof of Lemma 37. We are actually going to prove $\exists i : \mathrm{WithinF}(\mathsf{Pe}[j].\mathsf{recv}(i), kT_s)$ for all j in the cluster, which subsumes the statement of the lemma.

Let us start by noting that axiom **clustering** implies that there exist a peer p and a seed s such that $\mathsf{connected_s}(s, p)$, and the connections stays stable over the next kT_s time units, because of the usual assumption. In the remainder, let us assume 0 as the current instant. As a consequence of axiom **seed.sending** for the seed s, it is true that $\mathrm{WithinF}(\mathsf{Se}[s].\mathsf{send}(i), T_s)$ for some packet i, that

is there exists a time instant $0 < t < T_s$ in the future when $\mathsf{Se}[s].\mathsf{send}(i)$ holds. Therefore, because of the connection, $\mathsf{Pe}[p].\mathsf{recv}(i)$ at the same time t.

Let us consider any generic peer j in the cluster. We can assume that $\mathrm{dist}(p,j) > 0$ without loss of generality. In fact, if $\mathrm{dist}(p,j) = 0$, then it is simple to realize that $p = j$ and therefore the proof is concluded. Otherwise, we can apply lemma **futr_recv**, substituting p, j and $\mathrm{dist}(p,j)$ for p_1, p_2 and d respectively and considering t as the base time instant. We infer that $\mathrm{Futr}(\mathrm{WithinF}(\mathsf{Pe}[j].\mathsf{recv}(l)\,, \mathrm{dist}(p,j)T_s)\,, t)$ for some packet l. Considering the definitions of the corresponding TRIO operators, this means that there exists a time instant $t < q < \mathrm{dist}(p,j)T_s + t$ in the future when $\mathsf{Pe}[j].\mathsf{recv}(l)$ happens. It is simple to realize that $\mathrm{dist}(p,j) < k$, since this is true of any pair of peers in a cluster of size k. So $c = \mathrm{dist}(p,j) + 1 \leq k$, $t < q < cT_s$ and $q < kT_s$. This is the same as saying that $\mathrm{WithinF}(\mathsf{Pe}[j].\mathsf{recv}(i)\,, kT_s)$ for the generic peer j in the cluster, what we had to prove. □

Conclusion of intra-cluster proofs. So far, we have proved that, for each cluster K_i of size k_i:

$$\mathrm{const}_{\mathrm{ie}}(\mathsf{connected_p}(p_1,p_2) \wedge \mathsf{connected_s}(s,p), k_iT_s) \Rightarrow \mathrm{WithinF}_{\mathrm{ie}}\left(\bigwedge_{j\in K_i} \exists l : \mathrm{WithinF}(\mathsf{Pe}[j].\mathsf{send}(l)\,, k_iT_s)\,, k_iT_s\right)$$

Actually, an analysis of the performed proofs shows that all the considered events occurred within a time bound of k_iT_s time units, that is, equivalently, the outermost WithinF is always instantiated at the current time. In other words, we can actually claim a strengthening of the above formula, and namely

$$\mathrm{const}_{\mathrm{ie}}(\mathsf{connected_p}(p_1,p_2) \wedge \mathsf{connected_s}(s,p), k_iT_s) \Rightarrow \bigwedge_{j\in K_i} \exists l : \mathrm{WithinF}_{\mathrm{ie}}(\mathsf{Pe}[j].\mathsf{send}(l)\,, k_iT_s)$$

Finally, we also notice that we can discharge the Assumption **peer.late_recv** intra-cluster. The proof of it is similar to that of Lemma 37, and is therefore not discussed here. Moreover, notice that the discharging does not involve any circularity, as it ought to be.

Inter-cluster proofs. We are now ready to put together the various intra-cluster proofs to deduce the validity of theorem **global_send_liveness**.

Before doing that, let us state simple proper of the operators $\mathrm{const}(\cdot,\cdot)$ and WithinF. The proofs are very simple and are therefore omitted.

Lemma 40. *For any n predicates $P_1, \ldots, P_n$, time distances $T_1, \ldots, T_n$ and $T \geq \max_{i=1,\ldots,n} T_i$:*

$$\mathrm{const}_{\mathrm{bb}}\left(\bigwedge_{i=1,\ldots,n} P_i, T\right) \Rightarrow \bigwedge_{i=1,\ldots,n} \mathrm{const}_{\mathrm{bb}}(P_i, T_i)$$

Lemma 41. *For any nm predicates P_i^j for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, time distances $T_1, \ldots, T_n$ and $T \geq \max_{i=1,\ldots,n} T_i$:*

$$\bigwedge_{i=1,\ldots,n} \bigwedge_{j=1,\ldots,m} \mathrm{WithinF}_{\mathrm{bb}}\left(P_i^j, T_i\right) \Rightarrow \bigwedge_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} \mathrm{WithinF}_{\mathrm{bb}}\left(P_i^j, T\right)$$

Finally, we prove the global liveness theorem.

*Proof of theorem **global_send_liveness**.* Assume the antecedent

$$\mathrm{const}_{\mathrm{ie}}(\mathsf{connected_p}(p_1, p_2) \wedge \mathsf{connected_s}(s, p), N_p T_s)$$

By axiom **clustering**, it is obvious that $k_i \leq N_p$, for every i and at any time. Therefore, $\max_i k_i T_s \leq N_p T_s$, and we can apply Lemma 40, so $\bigwedge_{i=1,\ldots,k} \mathrm{const}_{\mathrm{ie}}(\mathsf{connected_p}(p_1, p_2) \wedge \mathsf{connected_s}(s, p), k_i T_s)$ holds.

This means that the antecedents of every intra-cluster result is satisfied. Therefore, we have that:

$$\bigwedge_{i=1,\ldots,k} \bigwedge_{j \in K_i} \exists l : \mathrm{WithinF}_{\mathrm{ie}}(\mathsf{Pe}[j].\mathsf{send}(l), k_i T_s)$$

Now, we apply Lemma 41 and finally conclude:

$$\bigwedge_{p=1,\ldots,N_p} \exists l : \mathrm{WithinF}_{\mathrm{ie}}(\mathsf{Pe}[p].\mathsf{send}(l), N_p T_s)$$

where the last rearrangement of indices is justified by the fact that the K_i's are a partition of the set of all N_p peers (axiom **clustering**). □

6.4.2 Using a Made Up New Inference Rule

We have all the ingredients to apply the compositional inference rule of Proposition 34 in order to prove theorem **global_send_liveness**.

As we did in the previous proof, let us split the proof between intra-cluster and inter-cluster proofs. Actually, the inter-cluster proof is simple, and it is just like as it was done in Section 6.4.1, the only difference being that we now deal with *ee* variations of the $\mathrm{const}_{\mathrm{ee}}(\cdot, \cdot)$ and $\mathrm{WithinF}_{\mathrm{ee}}$ operators. This is however immediately reducible to the previously seen case, as it is simple to check that $\mathrm{const}_{\mathrm{ie}}(\cdot, \cdot)$ implies $\mathrm{const}_{\mathrm{ee}}(\cdot, \cdot)$ for the same arguments, and $\mathrm{WithinF}_{\mathrm{ee}}$ implies $\mathrm{WithinF}_{\mathrm{ie}}$, for the same argument. Thus, let us just carry out the intra-cluster proof.

Intra-cluster proof. Let us consider a generic cluster of size k; let us instantiate the inference rule of Proposition 34 as follows. Notice that modules in the rule are numbered from 1, while peers (and seeds) are numbered from 0. Thus, we associate the number $i+1$ in the rule to peer number i.

- $E_i = \exists j : \mathsf{Pe}[i-1].\mathsf{recv}(j)$;
- $M_i = \exists j : \mathsf{Pe}[i-1].\mathsf{send}(j)$;
- $E = \mathrm{const}_{\mathrm{ee}}(\mathsf{connected_p}(p_1, p_2) \wedge \mathsf{connected_s}(s, p), kT_s)$, for all p_1, p_2, p in the cluster, and some seed s;
- $M = \bigwedge_{i \in \mathcal{I}_N} \mathrm{WithinF}(M_i, kT_s)$;
- $N = k$ and $\mathrm{T_B} = T_s$.

Applying the inference rule of Proposition 34 amounts to the following steps:

1. Prove that: $\exists j : \mathsf{Pe}[i-1].\mathsf{recv}(j) \Rightarrow \mathrm{WithinF}(\exists j : \mathsf{Pe}[i-1].\mathsf{send}(j), T_s)$, for all $i \in \{1, \ldots, k\}$.
2. Prove that:
   $E \wedge \exists j : \mathsf{Pe}[i-1].\mathsf{send}(j) \Rightarrow \exists j : \mathsf{Pe}[\mathtt{if}\ i-1 < k-1\ \mathtt{then}\ i\ \mathtt{else}\ 0].\mathsf{recv}(j)$,
   for all $i \in \{1, \ldots, k\}$.
3. Prove that:
   $\bigwedge_{i \in \mathcal{I}_N} \mathrm{WithinF}(\exists j : \mathsf{Pe}[i-1].\mathsf{send}(j), kT_s) \Rightarrow \bigwedge_{i \in \mathcal{I}_N} \mathrm{WithinF}(\exists j : \mathsf{Pe}[i-1].\mathsf{send}(j), kT_s)$.
4. Prove that: $\mathrm{WithinF}(\exists j : \mathsf{Pe}[i-1].\mathsf{recv}(j), \mathrm{T_B})$, for some $i \in \{1, \ldots, k\}$.

We now orderly provide the proof justifications.

1. For any $i \in \{1, \ldots, k\}$, this is exactly axiom **peer.recv_to_send** for module $i-1$.
2. For any $i \in \{1, \ldots, k\}$, this is immediately implied by the connection axiom **clustering**, together with the definition of the connection predicate in axiom **pp_connections**, for module $i-1$ and its successor in the clustering.
3. This is trivial as the left-hand side of the implication is identical to the right-hand side.
4. This is a direct consequence of axiom **clustering** again, together with axiom **seed.sending**.

Discussion. Let us compare the two different verification processes we have detailed in this section.

The first approach used a previously defined inference rule, namely that of Proposition 33. As we have seen, the application of the rule has been non-trivial, and has required some considerable ingenuity, as well as the proof of several details. On the other hand, the application of the inference rule in the second verification has been straightforward and with little technical complications. This does not mean that in this case the complexity of verification has "disappeared"; in fact, we had to invest considerable ingenuity in devising a new compositional inference rule, one that was simple to apply for the system being verified. So, the two different solutions have resolved the trade-off between difficulty in building a rule and difficulty in applying it in two different ways.

In general, it is difficult — and perhaps highly subjective as well — to assess which of the two approaches is preferable. Ideally, one should consider both approaches, and choose according to the particular features of the system under consideration, as well as his/her specific skills and attitudes. All in all, the most important "lesson" that we can draw from this example is about the importance of flexibility in pursuing compositional verification, in accordance with what we have extensively advocated in the previous sections.

References

[AL95] Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–535, 1995.

[Coh01] Bram Cohen. BitTorrent. `http://www.bittorrent.org`, 2001.

[Coh03] Bram Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, May 2003.

[Coo78] Stephen A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM Journal on Computing*, 7(1):70–90, 1978. Corrigendum in [Coo81].

[Coo81] Stephen A. Cook. Corrigendum: Soundness and completeness of an axiom system for program verification. *SIAM Journal on Computing*, 10(3):612, 1981.

[Dug06] Adam Duguid. Coping with the parallelism of BitTorrent: Conversion of PEPA to ODEs in dealing with state space explosion. In Eugene Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 156–170. Springer-Verlag, 2006.

[FRMM06] Carlo A. Furia, Matteo Rossi, Dino Mandrioli, and Angelo Morzenti. Automated compositional proofs for real-time systems. *Theoretical Computer Science*, 2006. To appear.

[Fur03] Carlo Alberto Furia. Compositional proofs for real-time modular systems. Master's thesis, Politecnico di Milano, December 2003. (Tesi di Laurea).

[Fur05] Carlo Alberto Furia. A compositional world: a survey of recent works on compositionality in formal methods. Technical Report 2005.22, Dipartimento di Elettronica e Informazione, Politecnico di Milano, March 2005.

[GM01] Angelo Gargantini and Angelo Morzenti. Automated deductive requirement analysis of critical systems. *ACM Transactions on Software Engineering and Methodology*, 10(3):255–307, 2001.

[Lam98] Leslie Lamport. Composition: A way to make proofs harder. In Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli, editors, *Proceedings of the International Symposium: "Compositionality: The Significant Difference" (COMPOS'97)*, volume 1536 of *Lecture Notes in Computer Science*, pages 402–423. Springer-Verlag, 1998.

[Men97] Elliott Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall, 4th edition, 1997.

[NT00] Kedar S. Namjoshi and Richard J. Trefler. On the completeness of compositional reasoning. In E. Allen Emerson and A. Prasad Sistla, editors, *Proceedings of the 12th International Conference on Computer Aided Verification (CAV'00)*, volume 1855 of *Lecture Notes in Computer Science*, pages 139–153. Springer-Verlag, 2000.