PDF Download
3767738.pdf
05 February 2026
Total Citations: 2
Total Downloads: 258

Published: 15 November 2025
Online AM: 16 September 2025
Accepted: 31 August 2025
Revised: 01 May 2025
Received: 22 October 2024

Citation in BibTeX format

RESEARCH-ARTICLE

Centralized vs. Decentralized Monitors for Hyperproperties

LUCA ACETO, Reykjavík University, Reykjavik, Iceland

ANTONIS ACHILLEOS, Reykjavík University, Reykjavik, Iceland

ELLI ANASTASIADI, Aalborg University, Aalborg, Nordjylland, Denmark

ADRIAN FRANCALANZA, University of Malta, Msida, Malta

DANIELE GORLA, Sapienza University of Rome, Rome, RM, Italy

JANA WAGEMAKER, Radboud University, Nijmegen, Gelderland, Netherlands

Open Access Support provided by:

University of Malta

Radboud University

Aalborg University

Sapienza University of Rome

Reykjavík University

Centralized vs. Decentralized Monitors for Hyperproperties

LUCA ACETO, Department of Computer Science, Reykjavik University, Reykjavik, Iceland and Gran Sasso Science Institute, L'Aquila, Italy
ANTONIS ACHILLEOS, Department of Computer Science, Reykjavik University, Reykjavik, Iceland
ELLI ANASTASIADI, Department of Computer Science, Aalborg University, Aalborg, Denmark
ADRIAN FRANCALANZA, University of Malta, Msida, Malta
DANIELE GORLA, Department of Computer Science, "Sapienza" University of Rome, Rome, Italy
JANA WAGEMAKER, iCIS, Radboud University, Nijmegen, Netherlands

This article focuses on the runtime verification of hyperproperties expressed in Hyper-recHML, an expressive yet simple logic for describing properties of sets of traces. To this end, we consider a simple language of monitors that observe sets of system executions and report verdicts w.r.t. a given Hyper-recHML formula. We first employ a unique omniscient monitor that centrally observes all system traces. Since centralized monitors are not ideal for distributed settings, we also provide a language for decentralized monitors, where each trace has a dedicated monitor; these monitors yield a unique verdict by communicating their observations to one another. For both the centralized and the decentralized settings, we provide a synthesis procedure that, given a formula, yields a monitor that is correct (i.e., sound and violation complete). A key step in proving the correctness of the synthesis for decentralized monitors is a result showing that, for each formula, the synthesized centralized monitor and its corresponding decentralized one are weakly bisimilar for a suitable notion of weak bisimulation.

CCS Concepts: • **Theory of computation → Operational semantics**; **Modal and temporal logics**; • **Logic and verification**;

Additional Key Words and Phrases: Runtime Verification, hyperlogics, decentralization

ACM Reference format:
Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, Daniele Gorla, and Jana Wagemaker. 2025. Centralized vs. Decentralized Monitors for Hyperproperties. *ACM Trans. Comput. Logic* 27, 1, Article 2 (November 2025), 57 pages.
https://doi.org/10.1145/3767738

This work has been supported by the project 'Mode(l)s of Verification and Monitorability' (MoVeMent) (Grant No. 217987) of the Icelandic Research Fund. Elli Anastasiadi's research has been supported by the grant 'S4OS: Scalable analysis and Synthesis of Safe, Small, Secure and Optimal Strategies for Cyber-Physical Systems' (Grant No. 00037819 of the Villum Foundation).
Authors' Contact Information: Luca Aceto, Department of Computer Science, Reykjavik University, Reykjavik, Iceland and Gran Sasso Science Institute, L'Aquila, Italy; e-mail: luca@ru.is; Antonis Achilleos, Department of Computer Science, Reykjavik University, Reykjavik, Iceland; e-mail: antonios@ru.is; Elli Anastasiadi, Department of Computer Science, Aalborg University, Aalborg, Denmark; e-mail: ellia@cs.aau.dk; Adrian Francalanza, University of Malta, Msida, Malta; e-mail: adrian.francalanza@um.edu.mt; Daniele Gorla (corresponding author), Department of Computer Science, "Sapienza" University of Rome, Rome, Italy; e-mail: gorla@di.uniroma1.it; Jana Wagemaker, iCIS, Radboud University, Nijmegen, Netherlands; e-mail: jana.wagemaker@ru.nl.

ACM 1557-945X/2025/11-ART2
https://doi.org/10.1145/3767738

1 Introduction

Runtime verification (RV) [Bartocci et al., 2018] is a verification technique that observes system executions to determine whether some given specification is satisfied or violated. This runtime analysis is usually conducted by a computational entity called a *monitor* [Francalanza, 2021]. RV is a lightweight verification technique that is carried out as the system under observation executes, thereby avoiding scalability issues caused by the state-explosion problem, as is the case for model checking. Recently, RV has been extended to parallel set-ups [Bocchi et al., 2010; Cassar et al., 2017; Mezzina and Pérez, 2017], and a large body of work in that setting aims to verify *hyperproperties* at runtime [Aceto et al., 2022a; Bonakdarpour and Finkbeiner, 2016, 2018; Clarkson and Schneider, 2010; Finkbeiner et al., 2019].

Hyperproperties [Clarkson and Schneider, 2010] are sets of *hypertraces*, i.e., sets of traces that may be seen as describing different system executions or the contributions of different sequential processes to a system execution. As argued in Bonakdarpour et al. [2018], many properties of concurrent and distributed systems can be viewed as hyperproperties. When verifying hyperproperties at runtime, several traces (i.e., several execution sequences) can be observed instead of just one, possibly at the same time. Several extensions of temporal logics, such as HyperLTL, HyperCTL* [Clarkson et al., 2014], Hyper^{2}LTL [Beutner et al., 2023], have been defined to express hyperproperties. Extensions of standard logics to hyper properties also include variations of the μ-calculus, such as Aceto et al. [2022a], setting the basis for the logic used in this article, and Gutsfeld et al. [2021], which studies an asynchronous semantics.

Since they were proposed by Clarkson and Schneider [2010], hyperproperties have become a fundamental, trace-based formalism for expressing security and privacy properties, verified using static and dynamic techniques [Agrawal and Bonakdarpour, 2016; Beutner et al., 2023, 2024; Bonakdarpour and Finkbeiner, 2016; Bonakdarpour et al., 2018; Brett et al., 2017; Chalupa and Henzinger, 2023; Finkbeiner et al., 2019] implemented in a variety of tools [Beutner and Finkbeiner, 2022; Beutner et al., 2024; Finkbeiner et al., 2018]. There is a large body of work, such as Agrawal and Bonakdarpour [2016], Brett et al. [2017], and Hahn et al. [2019], detailing several algorithms for monitoring (fragments of) hyperlogics under different assumptions and providing several correctness guarantees. However, these proposals either construct a centralized monitoring algorithm that has access to all traces in the observed hypertrace, or verify single trace properties, over a distributed set-up.[1] Having an omniscient monitor simplifies the runtime analysis since the monitoring algorithm can compare all traces as needed by simply accessing different parts of its local memory. But this power comes with drawbacks. For starters, centralized monitors are unrealistic for distributed systems, where trace analysis is typically localized to network nodes so as to minimize communication across locations. Moreover, centralized monitors create single points of failure during verification [Aceto et al., 2024c]. Furthermore, it can be problematic to store all the traces locally, especially in light of the wide availability of multi-core systems. The goal of the decentralized monitor synthesis from logical specifications presented in this article is to permit distributed monitor choreographies with *local* trace views whose components communicate in order to verify *global* properties (such as hyperproperties). Decentralized monitors have been shown to avoid high contentions leading to vastly improved scalability [Aceto et al., 2024c]. They also offer better privacy guarantees whenever they are stationed locally at the nodes where the respective traces are generated [Francalanza et al., 2013; Jia et al., 2016]. To the best of our knowledge, such a message-passing monitoring set-up has never been studied for the purpose of verifying hyperproperties so far.

In this article, we study procedures for the *automated synthesis of centralized and decentralized monitors* from hyperproperties described in the logic Hyper-recHML [Aceto et al., 2022a]. This logic

[1]See e.g., Bonakdarpour et al. [2016, 2022], Fraigniaud et al. [2020], and Francalanza et al. [2013] for distributed monitoring algorithms for classic trace-based logics.

extends the linear-time [Vardi, 1988] μ-calculus [Kozen, 1983] (also known as Hennessy-Milner logic with recursion [Larsen, 1990]) with constructs to describe properties of hypertraces inspired by the work on HyperLTL (namely variables ranging over traces, modal operators parametrized by trace variables, matching/mismatching between trace variables, and existential and universal quantification over them). Hyper-recHML can describe hyperproperties not expressible in HyperLTL or HyperCTL*, such as properties that speak about consensus (see Example 2.2) and periodicity (see Example 2.4). Furthermore, Hyper-recHML supports a general, syntax-driven monitor synthesis that can handle both the aforementioned hyperproperties, at least in the centralized case (see also the discussion in Section 5).

In both the centralized and decentralized set-ups, we work in the parallel model [Finkbeiner et al., 2019], where a fixed number of system executions is processed in parallel by monitors in an online fashion. We specify monitors using a process-algebraic formalism that builds on the one presented in Aceto et al. [2019a] and Francalanza et al. [2017] to define a class of monitors called regular. Such monitors are easy to describe, resemble (alternating) automata, and have sufficient expressive power to provide standard monitoring guarantees. Moreover, their algebraic structure supports the compositional definition of their operational semantics and monitor synthesis procedures from formulas, building on previous work relating algebraic process calculi with RV [Aceto et al., 2019c, 2023; Bocchi et al., 2017; Francalanza, 2017, 2021; Inoue and Yamagata, 2017; Lanotte et al., 2021, 2023].

In the centralized case, for each formula in the fragment of Hyper-recHML limited to greatest-fixed-point operators, our synthesis procedure yields a monolithic monitor that has access to all the traces in an observed hypertrace. However, in order to synthesize decentralized monitors for a sufficiently expressive fragment of the logic, it is necessary to extend the monitor capabilities with communication, as shown already in Aceto et al. [2022a]. For instance, to monitor for the property "If there is a trace where event a occurs, then there exists another trace where event b does not occur thereafter," monitors observing different traces need to communicate to record that event a occurred in some trace at some point and that there is some trace where b does not occur from that point onwards. Allowing monitors to send and receive messages significantly complicates their operational semantics (see Section 4), the monitor synthesis procedure (see Section 4.3), and all consequent proofs. The operational semantics for communicating monitors is one of the main contributions of the article since its design is crucial to obtain the correctness guarantees provided by the synthesis procedure for decentralized monitors. In particular, the semantics of decentralized monitors and their synthesis from formulas have to be designed carefully to ensure that monitors are reactive (they are always ready to process any system event) and input-enabled (they can always receive any input from other monitors in their environment), properties that are desirable in any decentralized RV set-up.

We show that both *the centralized and the decentralized monitor synthesis procedures are correct.* More precisely, the monitors synthesized from formulas are *sound* and *violation-complete*, meaning that (1) if the monitor synthesized from a formula φ reports a positive (resp., negative) verdict when observing a hypertrace T, then T does (resp., does not) satisfy φ, and (2) if T does not satisfy φ, then its associated monitor will report a negative verdict when observing T (see Theorems 3.2 and 3.5, and Corollaries 4.5 and 4.6). The proof of correctness in the decentralized case is considerably more technical than the corresponding proof in the centralized setting, due to the intricate communication semantics. To address the resulting technical challenges, we develop a proof strategy where we prove the correctness of the decentralized monitor synthesis procedure using the centralized one as a yardstick.

This methodology is one of the key contributions we offer in this study. More precisely, in Section 4.2 *we identify six properties of a decentralized monitor synthesis that make it 'principled'* (see Definition 4.8) and we show that, when a decentralized monitor synthesis is principled, the centralized and decentralized monitors synthesized from a formula are related by a suitable notion of weak bisimulation (Theorem 4.10). Apart from supporting the definition of decentralized monitor synthesis procedures, this result allows us to reduce the correctness of our decentralized monitor synthesis to that of the centralized one, which can in turn drive the definition of further synthesis procedures in future work. We also conjecture that our methodology provides a path to proving similar results for other models of communicating monitors independent of the monitoring strategy. In summary, our contributions are the following:

— a framework for monitoring hyperproperties by a central monitor that has access to all locations (Section 3) and a decentralized monitoring set-up for hyperproperties, with monitors that communicate (Section 4);
— a synthesis function that returns a correct centralized monitor for every formula without least fixed-points (Section 3);
— a synthesis function that returns a correct (decentralized) choreography of communicating monitors for every formula without least fixed-points that has no location quantifier within a fixed-point operator (Section 4); and
— a methodology to prove the correctness of a synthesis of communicating monitors, by establishing a list of desirable properties and relating the behavior of the decentralized monitors to that of the corresponding centralized monitor (Definition 4.8 and Theorem 4.10).

This is an extended and revised version of Aceto et al. [2024a]. With respect to the extended abstract, here we provide the most crucial details on proofs (for a complete account of all proofs, we refer the interested reader to Aceto et al. [2025]), we better justify Hyper-recHML (through the formulation of a version of the non-interference property taken from Clarkson and Schneider [2010]—see Example 2.3), we elaborate on the safety properties monitorable in our framework (Section 3.3), and we also explicitly include the decentralized synthesis of the diamond operator, that was omitted from the extended abstract (see however the drawback of omitting it in Remark 4.13).

2 The Model and the Logic

Let Act be a finite set of actions with at least two elements,[2] ranged over by a, b; the set of (infinite) traces over Act is $\mathsf{Trc} = \mathsf{Act}^\omega$, ranged over by t. Given a finite and non-empty set of locations $\mathcal{L}$ ranged over by ℓ, a hypertrace T on $\mathcal{L}$ is a function from $\mathcal{L}$ to Trc; the set of hypertraces on $\mathcal{L}$ is denoted by $\mathsf{HTrc}_{\mathcal{L}}$. $\mathcal{L}$ and Act are fixed throughout this article. A hypertrace describes a (distributed) system with $|\mathcal{L}|$ users, and every user is located at a unique location chosen from $\mathcal{L}$. A system behavior is captured by a hypertrace T on $\mathcal{L}$, mapping every user to the trace they perform.

For $t, t' \in \mathsf{Trc}$, we write $t \xrightarrow{a} t'$ whenever $t = at'$. Let $A : \mathcal{L} \to \mathsf{Act}$; for $T, T' \in \mathsf{HTrc}_{\mathcal{L}}$, we write $T \xrightarrow{A} T'$ whenever $T(\ell) \xrightarrow{A(\ell)} T'(\ell)$, for every $\ell \in \mathcal{L}$. Notice that, for each T, there is a *unique* pair A and T' such that $T \xrightarrow{A} T'$: more precisely, for every $\ell \in \mathcal{L}$, we have that $A(\ell) = a$ and $T'(\ell) = t'$, whenever $T(\ell) = at'$. We denote the A and T' just defined by $hd(T)$ and $tl(T)$, respectively. For a partial function $f : D \rightharpoonup E$ (where D and E are sets ranged over by d and e, respectively), we denote by $\mathsf{dom}(f)$ the set $\{d \in D \mid f(d) \text{ is defined}\}$ and by $\mathsf{rng}(f)$ the set $\{e \mid \exists d \in \mathsf{dom}(f).\ f(d) = e\}$. Notation $f[d \mapsto e]$ denotes the (partial) function mapping d to e and behaving like f otherwise.

2.1 The Logic Hyper-recHML

We consider Hyper-recHML as the logic to specify *hyperproperties*. We assume two disjoint and countably infinite sets Π and V of *location variables* and *recursion variables*, ranged over by π and x,

[2] When Act is a singleton, every property in the logic becomes equivalent to true or false.

Table 1. The Semantics of Hyper-recHML

$[\![\mathrm{tt}]\!]^{\rho}_{\sigma} = \mathsf{HTrc}_{\mathcal{L}}$	$[\![\mathrm{ff}]\!]^{\rho}_{\sigma} = \emptyset$	$[\![x]\!]^{\rho}_{\sigma} = \rho(x)$
$[\![\varphi \wedge \varphi']\!]^{\rho}_{\sigma} = [\![\varphi]\!]^{\rho}_{\sigma} \cap [\![\varphi']\!]^{\rho}_{\sigma}$		$[\![\varphi \vee \varphi']\!]^{\rho}_{\sigma} = [\![\varphi]\!]^{\rho}_{\sigma} \cup [\![\varphi']\!]^{\rho}_{\sigma}$
$[\![\max x.\psi]\!]^{\rho}_{\sigma} = \bigcup\{S \mid S \subseteq [\![\psi]\!]^{\rho[x\mapsto S]}_{\sigma}\}$		$[\![\min x.\psi]\!]^{\rho}_{\sigma} = \bigcap\{S \mid S \supseteq [\![\psi]\!]^{\rho[x\mapsto S]}_{\sigma}\}$
$[\![\exists\pi.\varphi]\!]^{\rho}_{\sigma} = \bigcup_{\ell\in\mathcal{L}} [\![\varphi]\!]^{\rho}_{\sigma[\pi\mapsto\ell]}$		$[\![\forall\pi.\varphi]\!]^{\rho}_{\sigma} = \bigcap_{\ell\in\mathcal{L}} [\![\varphi]\!]^{\rho}_{\sigma[\pi\mapsto\ell]}$
$[\![\pi = \pi']\!]^{\rho}_{\sigma} = \begin{cases} \mathsf{HTrc}_{\mathcal{L}} & \text{if } \sigma(\pi) = \sigma(\pi') \\ \emptyset & \text{otherwise} \end{cases}$		$[\![\pi \neq \pi']\!]^{\rho}_{\sigma} = \begin{cases} \mathsf{HTrc}_{\mathcal{L}} & \text{if } \sigma(\pi) \neq \sigma(\pi') \\ \emptyset & \text{otherwise} \end{cases}$
$[\![[a_\pi]\varphi]\!]^{\rho}_{\sigma} = \{T \mid hd(T)(\sigma(\pi)) = a \text{ implies } tl(T) \in [\![\varphi]\!]^{\rho}_{\sigma}\}$		$[\![\langle a_\pi\rangle\varphi]\!]^{\rho}_{\sigma} = \{T \mid hd(T)(\sigma(\pi)) = a \wedge tl(T) \in [\![\varphi]\!]^{\rho}_{\sigma})\}$

respectively. Formulas of Hyper-recHML are constructed as follows:

$$\varphi ::= \mathrm{tt} \mid \mathrm{ff} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \max x.\varphi \mid \min x.\varphi \mid x \mid \exists\pi.\varphi \mid \forall\pi.\varphi \mid \pi = \pi \mid \pi \neq \pi \mid [a_\pi]\varphi \mid \langle a_\pi\rangle\varphi.$$

Apart from the basic Boolean constructs, we include the greatest and least fixed-point operators to describe unbounded and/or infinite behaviors in a finitary manner,[3] existential/universal quantifiers and equality/inequality tests on location variables, and the usual Hennessy-Milner modalities where $[a_\pi]$ stands for 'necessarily after a at the location bound to π,' and $\langle a_\pi\rangle$ denotes 'possibly after a at the location bound to π.' A formula is said to be *guarded* if every recursion variable appears within the scope of a modality within its fixed-point binding. All formulas are assumed to be guarded (without loss of expressiveness [Kupferman et al., 2000]). We write $\mathsf{fv_l}(\varphi)$ to denote the free location variables of φ, and $\mathsf{fv_r}(\varphi)$ for the free recursion variables.

Remark 2.1. We consider formulas where bound location variables are all pairwise distinct (and different from the free variables); hence, the formula $\forall\pi.[a_\pi]\exists\pi.\varphi$ denotes the formula $\forall\pi.[a_\pi]\exists\pi'.(\varphi\{^{\pi'}/_{\pi}\})$, where $\varphi\{^{\pi'}/_{\pi}\}$ stands for the capture-avoiding substitution of π' for π in φ. A similar notation for other kinds of substitutions is used throughout the article. ◀

The semantics for formulas φ in Hyper-recHML is defined over $\mathsf{HTrc}_{\mathcal{L}}$ through the function $[\![-]\!]^{\rho}_{\sigma}$, as shown in Table 1, by exploiting two partial functions: $\rho\colon V \rightharpoonup 2^{\mathsf{HTrc}_{\mathcal{L}}}$, which assigns a set of hypertraces on $\mathcal{L}$ to all free recursion variables of φ, and $\sigma\colon \Pi \rightharpoonup \mathcal{L}$, which assigns a location to all free location variables of φ. In what follows, we tacitly assume that the free recursion and location variables in a formula φ are always included in $\mathsf{dom}(\rho)$ and $\mathsf{dom}(\sigma)$, respectively. Function ρ gives the semantics of recursion variables. Intuitively, $\rho(x)$ is the set of hypertraces that are assumed to satisfy x, as formally described on the first line of Table 1. Formulae of the form $\max x.\psi$ and $\min x.\psi$ are interpreted as the greatest and least fixed-points of the function induced by their body ψ. Function σ keeps track of the location associated with each location variable; it is extended every time a new variable π is introduced (through $\exists\pi$ or $\forall\pi$, resp.) to check whether the body of the quantification holds for some (resp., for all) locations.

All the other operators have quite a straightforward semantics. In particular, a formula $\langle a_\pi\rangle\varphi$ holds true at hypertrace T if the trace in T at the location bound to π starts with an a and $tl(T)$ satisfies φ; by contrast, a formula $[a_\pi]\varphi$ can also hold true if the trace in T at the location associated to π does not start with an a.

[3]In LTL, this behavior is captured by the 'Until' and 'Release' operators, but these are less expressive than fixed-points; see Aceto et al. [2021].

Whenever φ is *closed* (i.e., without any free variable), the semantics is given by $[[\varphi]]^{\emptyset}_{\emptyset}$, where $\emptyset$ denotes the partial function with empty domain. Notationally, we shall simply write $[[\varphi]]$ instead of $[[\varphi]]^{\emptyset}_{\emptyset}$. We say that T satisfies the closed formula φ if $T \in [[\varphi]]$.

Example 2.2. For example, consider the set of actions $\{a, b\}$; then, the hyperproperty:

$$\varphi_a = \forall\pi.\max x.\big(\langle b_\pi\rangle x \;\vee\; \exists\pi'.(\pi' \neq \pi \;\wedge\; \langle a_{\pi'}\rangle x)\big), \tag{1}$$

is a consensus-type property stating that, at every position of every trace, whenever there is an a there is another trace that also has a. Using the semantic definition of the logic, it is not hard to see that the hypertrace T_1 over the set of locations $\{\ell_1, \ell_2, \ell_3\}$ that maps ℓ_1 to a^ω, ℓ_2 to ba^ω and ℓ_3 to $(ba)^\omega$ does not satisfy the property φ_a: what breaks the property is the first position. On the other hand, the hypertrace T_2 that maps ℓ_1 to a^ω, ℓ_2 to $(ab)^\omega$ and ℓ_3 to $(ba)^\omega$ does satisfy φ_a because at each position there are two traces that exhibit an a. ◀

Example 2.3. As a second example, we show how the well-known property of *non-interference*, in the formulation by Goguen and Meseguer from Clarkson and Schneider [2010], can be written in our logic. The property requires that "commands issued by users holding high clearances be removable without affecting observations of users holding low clearances." They model this security policy as a hyperproperty by treating commands as inputs and observations as outputs, and by requiring that a system contains, for any trace t, a corresponding trace t' with no high inputs, yet with the same low outputs as t:

$$GMNI = \{T \mid \forall t \in T.\exists t' \in T\;.\;(H_{in}(t') = \epsilon \wedge L_{out}(t) = L_{out}(t'))\}.$$

In our setting, we assume that actions can be either high or low, i.e., that Act can be bipartitioned into $\{l^1, \dots, l^m\}$ and $\{h^1, \dots, h^n\}$. Then, we define the auxiliary formula:

$$low\_agree(\pi, \pi') = max\ x.\left(\bigwedge_{i=1}^{m}\left(\langle l^i_\pi\rangle \mathsf{tt} \leftrightarrow \langle l^i_{\pi'}\rangle \mathsf{tt}\right) \;\wedge\; \bigvee_{a\in\mathsf{Act}} [a_\pi]x\right).$$

This formula says that the two given locations π and π' exhibit the same low behavior, and this holds repeatedly, after any possible action (being it high or low). By using it, *GMNI* can be formulated in Hyper-recHML as follows:

$$\varphi_{GMNI} = \forall\pi.\exists\pi'.\left(low\_agree(\pi,\pi') \;\wedge\; max\ x.\left(\bigwedge_{i=1}^{n}[h^i_{\pi'}]\mathsf{ff} \;\wedge\; \bigwedge_{i=1}^{m}[l^i_{\pi'}]x\right)\right).$$

Formula φ_{GMNI} states that, for every trace, there must exist another trace such that: (1) the two completely agrees on the low actions (as expressed by the *low_agree* conjunct); and (2) the second one cannot perform any high action (as expressed by the *max x* conjunct).

2.2 On the Expressiveness of Hyper-recHML

The logic Hyper-recHML adapts linear-time μHML [Larsen, 1990] to express properties of hypertraces, just as HyperLTL and HyperCTL* [Clarkson et al., 2014] are variations on LTL [Pnueli, 1977] and CTL* [Emerson and Halpern, 1986], respectively, interpreted over hypertraces. It is well known that μHML is more expressive than LTL and CTL* [Wolper, 1983]; in a companion paper, we are working to formally show that the same result holds also in the hyper-setting, i.e., that Hyper-recHML can express all properties that can be described using HyperLTL and HyperCTL*. Here we confine ourselves to show that Hyper-recHML can express properties that cannot be described using HyperLTL and HyperCTL*.

First, we recall that Wolper showed in Wolper [1983] that the property "event a occurs at all even positions in a trace" cannot be expressed in LTL (see Wolper [1983, Corollary 4.2] that is based on Theorem 4.1 in that reference). We will refer to this property as φ_e, where "e" stands for even, and adapt it to a hypertrace setting for proving strictness of the inclusion of HyperLTL in Hyper-recHML.

Example 2.4. Let φ_e^h be the hyperproperty on the set of actions $\{a, b\}$ that results from adding an existential trace quantifier $\exists\pi$ at the beginning of φ_e, and replacing all modalities with π-indexed ones:

$$\varphi_e^h = \exists\pi.\, \max x.\big([a_\pi]\langle a_\pi\rangle x \wedge [b_\pi]\langle a_\pi\rangle x\big). \tag{2}$$

This is a liveness property that describes the periodicity of events; when evaluated over singleton hypertraces, it coincides with the evaluation of φ_e. ◀

Theorem 2.5. *Hyper-*rec*HML is more expressive than HyperLTL.*

Proof. By contradiction, assume that a formula φ_{h-LTL} expresses exactly that there exists a trace in a hypertrace such that a holds on even positions. We cannot trivially claim that this would be expressed with a single quantifier $\exists\pi$, but we know that over singleton hypertraces, say for $T = \{t_0\}$, $T \models \varphi_{h-LTL}$ iff $T \models \varphi_e^h$. However, since T contains only a single trace, we know that all the trace variables in φ_{h-LTL} must be mapped to t_0. Consequently, all propositional variables that occur in φ_{h-LTL} are now interpreted to hold on the same trace. Therefore, for this variable mapping, we get an LTL formula which expresses exactly that the single trace (that all trace variables have been mapped to) satisfies Wolper's property.

Therefore, we can express the same formula in plain LTL, by replacing all propositional variables with non-trace quantified ones, and arrive at a contradiction. □

For HyperCTL* we exploit the possibility of the latter to use quantifiers in any part of a formula. This has already been shown in the hyperproperty φ_a defined in Equation (1): such a formula can potentially spawn an unbounded number of quantifiers, by unfolding the recursion when encountering a events.

Theorem 2.6. *Hyper-*rec*HML is more expressive than HyperCTL*.*

Proof. To prove this, we refer the reader to Bozzelli et al. [2015], where it is shown that the property "there is an $n \geq 0$ such that $a \neq t(n)$ for every $t \in T$" is not expressible in HyperCTL* (notice that this property was also shown not to be expressible in HyperLTL by Finkbeiner and Zimmermann [2017]). In our logic this property is expressible (over the set of actions $\{a, b\}$) with the formula:

$$\min x.((\forall\pi.\langle b_\pi\rangle \mathrm{tt}) \vee (\forall\pi'.([a_{\pi'}]x \wedge [b_{\pi'}]x)).$$

Note that this formula can also be simplified. We choose to write it like this so that it is explicit that either all traces have b, or all traces take a step. However the following formulation is also valid, and only uses one quantifier:

$$\min x.((\forall\pi.((\langle b_\pi\rangle \mathrm{tt}) \vee ([a_\pi]x \wedge [b_\pi]x)).$$ □

To conclude, we remark that this additional expressiveness of Hyper-recHML is present in the fragments for which we synthesize monitors, as we shall see in the next sections.

Table 2. The Operational Semantics for Centralized Monitors, Where $\odot \in \{\otimes, \oplus\}$

$$v \xrightarrow{A} v \qquad \frac{A(\ell) = a}{a_\ell.m \xrightarrow{A} m} \qquad \frac{A(\ell) \neq a}{a_\ell.m \xrightarrow{A} \mathsf{end}} \qquad \frac{m\{^{\mathsf{rec}\,x.m}/_x\} \xrightarrow{A} m'}{\mathsf{rec}\,x.m \xrightarrow{A} m'} \qquad \frac{m \xrightarrow{A} m'}{m + n \xrightarrow{A} m'} \qquad \frac{n \xrightarrow{A} n'}{m + n \xrightarrow{A} n'}$$

$$\frac{m \xrightarrow{A} m' \qquad n \xrightarrow{A} n'}{m \odot n \xrightarrow{A} m' \odot n'}$$

3 Centralized Monitoring

We first present a centralized, omniscient monitor that enforces hyperproperties expressed in Hyper-recHML; here, the monitoring algorithm can compare all traces as needed by simply accessing different parts of its local memory.

3.1 Centralized Monitors: Syntax and Operational Semantics

The set of centralized monitors CMon is given by the following grammar:

$$\mathsf{CMon} \ni m ::= \mathsf{yes} \mid \mathsf{no} \mid \mathsf{end} \mid a_\ell.m \mid m + m \mid m \oplus m \mid m \otimes m \mid \mathsf{rec}\,x.m \mid x.$$

Verdicts, taken from the set {yes, no, end} and ranged over by v, are the possible outcomes of the monitor observation: it can be yes, if the hyperproperty associated to the monitor is satisfied, no, it if is not, or end, if the monitor has not enough evidence to conclude either yes or no. The prefixed monitor $a_\ell.m$ checks if ℓ is currently performing an a and, in case, continues as m. We can compose monitors by using non-deterministic choice, used to specify alternative and mutually exclusive observation capabilities, and parallel-and and parallel-or, used to build parallel monitors that combine their verdicts according to a conjunctive or disjunctive strategy, respectively. Notationally, we denote with $\odot$ any of $\otimes$ and $\oplus$. Finally, monitors can also be recursively defined.

The operational semantics of centralized monitors is given in Table 2 and follows the intuitive explanations we gave above for the different operators. In particular, a monitor that waits for an action at some location (as prescribed by a_ℓ) can either see that action at ℓ (as stated by A) and proceed in its observation, or it does not observe that action and stops its monitoring activity, by reporting end. Recursive monitors should be unfolded to evolve. A non-deterministic choice $m + n$ can initially behave like m and discard n in doing so or vice versa. Finally, once issued, a verdict never changes; possibly different verdicts obtained in different parallel branches of the monitor can be composed conjunctively or disjunctively. This is represented by the judgment $\Rrightarrow$, whose intended use is to evaluate monitors and reach a verdict, whenever possible. The rules are given in Table 3. They state that an end verdict can be obtained only when combining two (or more) end verdicts, that yes and no are the absorbing elements for $\oplus$ and $\otimes$, respectively, and the neutral elements for $\otimes$ and $\oplus$, respectively, that verdict evaluation of a non-deterministic monitor is non-deterministic as well, and that recursive monitors must be unfolded before they can be properly evaluated. Also notice that there can be multiple ways to infer the same verdict for the same monitor: e.g., for yes $\oplus$ no we can either use the third rule of the first line or the (symmetric version of the) second rule from the second line of Table 3. However, the inferred value is of course the same (i.e., yes, in the previous situation).

We instrument a monitor m on a hypertrace T based on the rules of Table 4. As usual, we write $\rightarrowtail^*$ for the reflexive-transitive closure of $\rightarrowtail$.

Table 3. Verdict Evaluation for Centralized Monitors (up to Commutativity of +, ⊗, and ⊕)

$v \Rrightarrow v$	$\dfrac{m \Rrightarrow \mathsf{end} \quad n \Rrightarrow \mathsf{end}}{m \odot n \Rrightarrow \mathsf{end}}$	$\dfrac{m \Rrightarrow \mathsf{yes}}{m \oplus n \Rrightarrow \mathsf{yes}}$	$\dfrac{m \Rrightarrow \mathsf{no}}{m \otimes n \Rrightarrow \mathsf{no}}$
$\dfrac{m \Rrightarrow v}{m + n \Rrightarrow v}$	$\dfrac{m \Rrightarrow \mathsf{no} \quad n \Rrightarrow v}{m \oplus n \Rrightarrow v}$	$\dfrac{m \Rrightarrow \mathsf{yes} \quad n \Rrightarrow v}{m \otimes n \Rrightarrow v}$	$\dfrac{m\{^{\mathsf{rec}\ x.m}/_{x}\} \Rrightarrow v}{\mathsf{rec}\ x.m \Rrightarrow v}$

Table 4. The Instrumentation Rules for Centralized Monitors

$\dfrac{m \xrightarrow{A} m' \qquad T \xrightarrow{A} T'}{m \triangleright T \rightarrowtail m' \triangleright T'}$
$\dfrac{m \Rrightarrow v}{m \triangleright T \rightarrowtail v}$

Table 5. Centralized Monitor Synthesis

$\mathsf{cm}_\sigma(\mathsf{tt}) = \mathsf{yes} \qquad \mathsf{cm}_\sigma(\mathsf{ff}) = \mathsf{no}$	$\mathsf{cm}_\sigma(x) = x \qquad \mathsf{cm}_\sigma(\max x.\varphi) = \mathsf{rec}\ x.\mathsf{cm}_\sigma(\varphi)$
$\mathsf{cm}_\sigma(\varphi \wedge \varphi') = \mathsf{cm}_\sigma(\varphi) \otimes \mathsf{cm}_\sigma(\varphi')$	$\mathsf{cm}_\sigma(\varphi \vee \varphi') = \mathsf{cm}_\sigma(\varphi) \oplus \mathsf{cm}_\sigma(\varphi')$
$\mathsf{cm}_\sigma(\forall \pi.\varphi) = \bigotimes_{\ell \in \mathcal{L}} \mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\varphi)$	$\mathsf{cm}_\sigma(\exists \pi.\varphi) = \bigoplus_{\ell \in \mathcal{L}} \mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\varphi)$
$\mathsf{cm}_\sigma(\pi = \pi') = \begin{cases} \mathsf{yes} \text{ if } \sigma(\pi) = \sigma(\pi') \\ \mathsf{no} \text{ otherwise} \end{cases}$	$\mathsf{cm}_\sigma(\pi \neq \pi') = \begin{cases} \mathsf{yes} \text{ if } \sigma(\pi) \neq \sigma(\pi') \\ \mathsf{no} \text{ otherwise} \end{cases}$
$\mathsf{cm}_\sigma([a_\pi]\varphi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\varphi) + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{yes}$	$\mathsf{cm}_\sigma(\langle a_\pi \rangle \varphi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\varphi) + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{no}$

3.2 From Formulas to Centralized Monitors

We derive monitors for the subset of formulas without least fixed-points, denoted with Hyper-maxHML. More precisely, given a formula φ, we want to derive a monitor that, when monitoring a hypertrace T, returns no if and only if T does not belong to the semantics of φ; furthermore, if it returns yes, then T belongs to the semantics of φ. All regular properties of infinite traces that can be monitored for violations with the aforementioned guarantees can be expressed without using least fixed-point operators (see the maximality results presented in Aceto et al. [2019a, Proposition 4.18] and Aceto et al. [2021, Theorem 5.2] in the setting of logics interpreted over infinite traces). Intuitively, we use least fixed-points to describe liveness properties, whose violation does not have a finite witness in general.

The definition of the synthetized monitor is given by induction on φ. This definition is parametrized by a partial function σ, assigning a location to all the free location variables of φ; when φ is closed, we consider $\mathsf{cm}_\emptyset(\varphi)$. The formal definition is given in Table 5. A monitor synthetized from a greatest-fixed-point formula is itself recursive and intuitively checks whether some unfolding of the formula is violated. The monitor for $\varphi \wedge \varphi'$ (respectively, $\varphi \vee \varphi'$) is obtained as the parallel-and (respectively, the parallel-or) of the monitors synthesized from φ and φ'. Universal and existential quantifiers are treated as conjunctions and disjunctions, respectively, and use the function σ to assign values to the newly introduced location variable. Finally, the monitors for formulae of the form $[a_\pi].\varphi$ and $\langle a_\pi \rangle.\varphi$ look for an occurrence of action a at (the location bound to) π; if such action is observed at that location, then the monitor proceeds by checking the rest of the formula, otherwise the monitor for $\langle a_\pi \rangle.\varphi$ returns no whereas the monitor for $[a_\pi].\varphi$ returns yes. Notice that we are using the choice operator '+' here to consider only one of the possible observations (corresponding to the action occurring at the location $\sigma(\pi)$) and discarding all the other ones.

Example 3.1. Let $\mathcal{L} = \{1, 2\}$ and $\mathsf{Act} = \{a, b\}$, and consider the formula (2). The monitor synthesis in Table 5 produces the following monitor m when applied to that formula:

$$m = \bigoplus_{\ell \in \{1,2\}} \mathsf{rec}\ x.((a_\ell.(a_\ell.x + b_\ell.\mathsf{no}) + b_\ell.\mathsf{yes}) \otimes (b_\ell.(a_\ell.x + b_\ell.\mathsf{no}) + a_\ell.\mathsf{yes})).$$

When monitor m is instrumented with the hypertrace T mapping location 1 to a^ω and location 2 to $(ab)^\omega$, the verdict no cannot be reached: indeed, T satisfies the formula φ since the trace at location 1 has a at all positions. On the other hand, when m is instrumented with the hypertrace T' mapping location 1 to b^ω and location 2 to $(ab)^\omega$, the no verdict is reached after the monitor has observed the first two actions at locations 1 and 2; this is in line with the fact that T' does not satisfy φ^h_e. ◀

The main results of this section are that the centralized monitors synthesized from formulas report sound verdicts and their verdicts are complete for formula violations. We refer the reader to Aceto et al. [2021] for a discussion on notions of correctness for monitors and the significance of soundness and violation-completeness. The main point is that, already in the setting of μHML, it is possible to have soundness and completeness only for formulae without fixed-points; hence, satisfaction completeness and violation completeness were introduced to weaken the completeness requirement and having the possibility of monitoring also some recursive formulae.

To prove soundness (Theorem 3.5), we first need three lemmata: the first one claims soundness of synthesized monitors when they have not taken any transition yet; the second one claims that synthesized monitors always transition to other synthesized monitors; the last one claims that the relation between the hypertrace and the semantics of the formula are carried over through the instrumentation steps. The proofs of these results have been relegated to Appendix A, to streamline reading.

Lemma 3.2. *If* $\mathsf{cm}_\sigma(\varphi) \Rightarrow v$, *then* $[[\varphi]]_\sigma = \mathsf{HTrc}_\mathcal{L}$, *if* $v = \mathsf{yes}$, *and* $[[\varphi]]_\sigma = \emptyset$, *if* $v = \mathsf{no}$.

Lemma 3.3. *If* $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} m'$, *then* $m' = \mathsf{cm}'_\sigma(\varphi')$ *for some* $'_\sigma$ *and* φ'.

Lemma 3.4. *Let* $\mathsf{cm}_\sigma(\varphi) \triangleright T \rightarrowtail \mathsf{cm}'_\sigma(\varphi') \triangleright T'$. *If* $T' \notin [[\varphi']]'_\sigma$ *then* $T \notin [[\varphi]]_\sigma$; *if* $T' \in [[\varphi']]'_\sigma$ *then* $T \in [[\varphi]]_\sigma$.

Theorem 3.5 (Soundness). *Let* $\varphi \in$ *Hyper-maxHML be a closed formula and* $T \in \mathsf{HTrc}_\mathcal{L}$. *If* $\mathsf{cm}_\emptyset(\varphi) \triangleright T \rightarrowtail^* \mathsf{no}$, *then* $T \notin [[\varphi]]$; *if* $\mathsf{cm}_\emptyset(\varphi) \triangleright T \rightarrowtail^* \mathsf{yes}$, *then* $T \in [[\varphi]]$.

Proof. By definition, there exist an integer $h > 0$ and $m_1 \triangleright T_1, \dots, m_h \triangleright T_h$ such that $m_1 = \mathsf{cm}_\emptyset(\varphi)$, $T_1 = T$, $m_i \triangleright T_i \rightarrowtail m_{i+1} \triangleright T_{i+1}$ (for every $i = 1, \dots, h-1$), and $m_h \triangleright T_h \rightarrowtail v$. We proceed by induction on h. The base case holds because of Lemma 3.2 The inductive case holds because of Lemma 3.4, which can be applied because of Lemma 3.3. □

We now move to violation completeness, i.e., that all hypertraces that do not belong to $[[\varphi]]$ are rejected by the monitor for φ. This is proved in Theorem 3.8, that requires two preliminary easy lemmata, whose proofs hold by definition of the operational semantics.

Lemma 3.6. *If* $m \triangleright T \rightarrowtail^* \mathsf{no}$, *then* $(m_1 \otimes \dots \otimes m_h \otimes m \otimes m_{h+1} \otimes \dots \otimes m_k) \triangleright T \rightarrowtail^* \mathsf{no}$, *for every* $m_1, \dots, m_h, m_{h+1}, \dots, m_k$ *without free variables.*

Lemma 3.7. *If* $m_i \triangleright T \rightarrowtail^* \mathsf{no}$ *for every* $i = 1, \dots, k$, *then* $(m_1 \oplus \dots \oplus m_k) \triangleright T \rightarrowtail^* \mathsf{no}$.

Theorem 3.8 (Violation Completeness). *Let* $\varphi \in$ *Hyper-maxHML be a closed formula and* $T \in \mathsf{HTrc}_\mathcal{L}$. *If* $T \notin [[\varphi]]$, *then* $\mathsf{cm}_\emptyset(\varphi) \triangleright T \rightarrowtail^* \mathsf{no}$.

Proof. We assume that in φ every recursive variable x appears in the scope of a unique (max) fixed-point subformula of φ of the form $\mathtt{fix}(x) \triangleq \max x.\psi$. Let $\leq_\varphi$ be a partial order of the recursive variables in φ, such that $x \leq_\varphi y$ if $\mathtt{fix}(x)$ is a subformula of $\mathtt{fix}(y)$ in φ. We now define the closure $\mathtt{cls}(\psi)$ of a subformula ψ of φ by induction on the number of the free recursion variables in ψ: if ψ is closed, then $\mathtt{cls}(\psi) \triangleq \psi$; otherwise, $\mathtt{cls}(\psi) \triangleq \mathtt{cls}(\psi\{^{\mathtt{fix}(x)}/_x\})$, where x is $\leq_\varphi$-minimal in ψ.

For each σ, let ρ_σ be such that $\rho_\sigma(x) = \{T \mid \mathsf{cm}_\sigma(\mathtt{cls}(x)) \triangleright T \not\rightarrowtail^* \mathsf{no}\}$ for every x in its domain. By induction on the structure of ψ, we prove that:

> for every (not necessarily closed) subformula ψ of φ, $T \notin [[\psi]]_\sigma^{\rho_\sigma}$ implies $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \rightarrowtail^*$ no, for σ and ρ_σ such that $\mathsf{fv}_\mathsf{l}(\psi) \subseteq \mathsf{dom}(\sigma)$ and $\mathsf{fv}_\mathsf{r}(\psi) \subseteq \mathsf{dom}(\rho_\sigma)$.

Clearly, the theorem is proved if we consider φ. There are seven possible base cases:

- $\psi = \mathsf{tt}$, or $\psi = (\pi = \pi')$ for $\sigma(\pi) = \sigma(\pi')$, or $\psi = (\pi \neq \pi')$ for $\sigma(\pi) \neq \sigma(\pi')$: In these cases, all T's belong to $[[\psi]]_\sigma^{\rho_\sigma}$, so there is nothing to prove.
- $\psi = \mathsf{ff}$, or $\psi = (\pi = \pi')$ for $\sigma(\pi) \neq \sigma(\pi')$, or $\psi = (\pi \neq \pi')$ for $\sigma(\pi) = \sigma(\pi')$: In these cases, $\mathsf{cm}_\sigma(\psi) \triangleq \mathsf{no}$ and the claim is then trivial.
- $\psi = x$: Immediate from the definition of ρ_σ.

For the inductive step, we distinguish the outmost operator in ψ:

- $\psi = \psi_1 \wedge \psi_2$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho_\sigma}$ if and only if $T \notin [[\psi_i]]_\sigma^{\rho_\sigma}$, for some $i \in \{1,2\}$. By inductive hypothesis, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi_i)) \triangleright T \rightarrowtail^* \mathsf{no}$. By definition of $\mathsf{cm}_\sigma(\mathtt{cls}(\psi))$ and Lemma 3.6, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \rightarrowtail^* \mathsf{no}$.
- $\psi = \psi_1 \vee \psi_2$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho_\sigma}$ if and only if $T \notin [[\psi_i]]_\sigma^{\rho_\sigma}$, for $i \in \{1,2\}$. By inductive hypothesis, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi_i)) \triangleright T \rightarrowtail^* \mathsf{no}$. By definition of $\mathsf{cm}_\sigma(\mathtt{cls}(\psi))$ and Lemma 3.7, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \rightarrowtail^* \mathsf{no}$.
- $\psi = [a_\pi]\chi$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho_\sigma}$ if and only if $hd(T)(\sigma(\pi)) = a$ and $tl(T) \notin [[\chi]]_\sigma^{\rho_\sigma}$. By definition, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\mathtt{cls}(\chi)) + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{yes}$; hence, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \rightarrowtail \mathsf{cm}_\sigma(\mathtt{cls}(\chi)) \triangleright tl(T)$. By inductive hypothesis, $\mathsf{cm}_\sigma(\mathtt{cls}(\chi)) \triangleright tl(T) \rightarrowtail^*$ no, and we easily conclude.
- $\psi = \langle a_\pi \rangle \chi$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho_\sigma}$ if and only if either $hd(T)(\sigma(\pi)) \neq a$ or $tl(T) \notin [[\chi]]_\sigma^{\rho_\sigma}$. By definition, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\mathtt{cls}(\chi)) + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{no}$. If $hd(T)(\sigma(\pi)) \neq a$, then $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \rightarrowtail \mathsf{no} \triangleright tl(T)$ and we easily conclude. Otherwise, we work like in the previous case.
- $\psi = \forall \pi.\chi$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho}$ if and only if there exists $\ell \in \mathcal{L}$ such that $T \notin [[\chi]]_{\sigma[\pi \mapsto \ell]}^{\rho}$. By inductive hypothesis, $\mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\mathtt{cls}(\chi)) \triangleright T \rightarrowtail^* \mathsf{no}$. We conclude by definition of $\mathsf{cm}_\sigma(\mathtt{cls}(\psi))$ and Lemma 3.6.
- $\psi = \exists \pi.\chi$: By definition of the semantics, $T \notin [[\psi]]_\sigma^{\rho}$ if and only if for all $\ell \in \mathcal{L}$ it holds that $T \notin [[\chi]]_{\sigma[\pi \mapsto \ell]}^{\rho}$. By inductive hypothesis, $\mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\mathtt{cls}(\chi)) \triangleright T \rightarrowtail^* \mathsf{no}$, for all ℓ. We conclude by definition of $\mathsf{cm}_\sigma(\mathtt{cls}(\psi))$ and Lemma 3.7.
- $\psi = \max x.\chi$: We prove the contrapositive, i.e., that $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \not\rightarrowtail^* \mathsf{no}$ implies that $T \in [[\psi]]_\sigma^{\rho_\sigma}$. Let us consider $S \triangleq \{T' \mid \mathsf{cm}_\sigma(\mathtt{cls}(\chi)) \triangleright T' \not\rightarrowtail^* \mathsf{no}\}$. We observe that $\rho_\sigma[x \mapsto S]$ satisfies the assumptions required by the inductive statement; so, the inductive hypothesis gives us that $S \subseteq [[\chi]]_\sigma^{\rho_\sigma[x \mapsto S]}$, and therefore $S \subseteq [[\psi]]_\sigma^{\rho_\sigma}$. By the definition of the monitor semantics, $\mathsf{cm}_\sigma(\mathtt{cls}(\psi)) \triangleright T \not\rightarrowtail^* \mathsf{no}$ yields $\mathsf{cm}_\sigma(\mathtt{cls}(\chi)) \triangleright T \not\rightarrowtail^* \mathsf{no}$, meaning that $T \in S$, and therefore $T \in [[\psi]]_\sigma^{\rho_\sigma}$, which completes the proof. □

3.3 On the Monitorable Hypersafety Properties

The monitor synthesis we presented above does not cover the whole Hyper-recHML, but just Hyper-maxHML; nevertheless, this fragment is able to capture all *hypersafety properties*, as defined in Clarkson and Schneider [2010]. Using this definition, one can show that hypersafety properties are subset-closed, as shown in Clarkson and Schneider [2010, Theorem 1].

In our setting, S ($\subseteq \mathsf{HTrc}_{\mathcal{L}}$) is a hypersafety property whenever:

$$\forall T \in \mathsf{HTrc}_{\mathcal{L}}.\, T \notin S \implies \Big(\exists M \in \mathsf{HTrc}^{f}_{\mathcal{L}} .\ (M \leq T \ \wedge\ \forall T' \in \mathsf{HTrc}_{\mathcal{L}}.\, M \leq T' \implies T' \notin S)\Big),$$

where $\mathsf{HTrc}^{f}_{\mathcal{L}}$ denotes the set of mappings from $\mathcal{L}$ to Act^* and $M \leq T$ means that, for every $\ell \in \mathcal{L}$, there exists a $\ell' \in \mathcal{L}$ such that $M(\ell)$ is a prefix of $T(\ell')$. The formula above tells that a set of hypertraces S is a hypersafety property if every hypertrace T that does not belong to S admits a “finite justification” M for this; M is a set of finte traces that can be extended to become T (by possibly changing the traces-to-locations assignment—as the ‘$\leq$’ relation allows) and that, however we extend it, we obtain some hypertrace that is not in S.

Notice that, in general, Clarkson and Schneider [2010] do not associate traces to locations and, most importantly, they allow hypertraces to be possibly infinite sets of traces. By contrast, our T’s are finite sets, since $|\mathcal{L}| = k$, which also implies an upper bound on any possible “bad thing” M. For such M’s it has been proven (see Bonakdarpour and Finkbeiner [2016]) that, for each $k > 1$, these properties coincide with the so-called *k-safety hyperproperties*, which in turn correspond to HyperLTL sentences with at most k universal quantifiers.

However, our monitor synthesis also handles formulae with quantifier alternation; as an example, consider Goguen and Meseguer ’s classic definition of the non-interference property given in Clarkson and Schneider [2010] (see *GMNI* in Example 2.3). Clearly, the hyperproperty *GMNI* can be expressed in the fragment of Hyper-recHML for which we have synthesized centralized monitors in Section 3.2 (see φ_{GMNI} in Example 2.3): indeed, to detect violations for a fixed set of traces, it suffices to check all pairs for such a violation, which is what the formula expresses and the monitor synthesized from it does. However, *GMNI* is known not to be a hypersafety property, since it is not subset-closed.

Hence, if we want to characterize all the properties monitorable for violations in our centralized monitoring set-up, where verdicts are irrevocable, what is needed is a variation on the notion of hypersafety properties that takes the fact that hypertraces are $\mathcal{L}$-indexed families of traces into account. We believe that the following proposal better corresponds to our set-up, and hence to the power of the monitors we have discussed so far. Intuitively, by following the characterization of safety property given in Alpern and Schneider [1985] (i.e., the well-known motto “nothing bad happens”), the two key characteristics of the notion of the “bad thing” witnessing a violation of a hypersafety property are that it must be finitely observable and irremediable for a given *fixed* set of locations $\mathcal{L}$.

Definition 3.9. A hyperproperty S is *violation-monitorable* whenever:

$$\forall T \in \mathsf{HTrc}_{\mathcal{L}}.\, T \notin S \implies \Big(\exists M \in \mathsf{HTrc}^{f}_{\mathcal{L}} .\ (M \sqsubseteq T \ \wedge\ \forall T' \in \mathsf{HTrc}_{\mathcal{L}}.\, M \sqsubseteq T' \Rightarrow T' \notin S)\Big),$$

where $M \sqsubseteq T$ means that, for every $\ell \in \mathcal{L}$, it holds that $M(\ell)$ is a prefix of $T(\ell)$.

Essentially, this definition allows for the “bad thing” M to be infinitely extended (as T'), but, differently from hypersafety, no new trace can be added (i.e., M and T' are defined on the very same set of locations $\mathcal{L}$). Notice that *GMNI* is a violation-monitorable property. Indeed, let $T \notin GMNI$; then, there is some location ℓ such that, for every $\ell' \in \mathcal{L}$, it holds that either $T(\ell')$ contains some high action, or the low parts of $T(\ell)$ and $T(\ell')$ are different. For each $\ell' \in \mathcal{L}$, let $i_{\ell'}$ be the first index such that $T(\ell')(i_{\ell'})$ is high or $T(\ell)(i_{\ell'}) \neq T(\ell')(i_{\ell'})$, and let i be the maximum of all the $i_{\ell'}$. Let M be the $\mathcal{L}$-indexed set of finite traces mapping each $\ell' \in \mathcal{L}$ to the prefix of $T(\ell')$ of length i. Now, for every T' such that $M \sqsubseteq T'$, we have that $T' \notin GMNI$ because its prefix M provides a witness breaking the definition of GMNI for $T'(\ell)$.

We conjecture that violation-monitorable hypersafety corresponds to the properties that are monitorable for violations in the parallel monitor set-up. Moreover, such a definition should also allow us to prove maximality results for a specific monitoring set-up; for example, any violation-monitorable property is monitorable for violations by the monitors produced by our synthesis, and vice versa. This is a challenging direction for future research. Furthermore, the classic definition of hypersafety property and that of violation-monitorable hyperproperty given above are parametrized on a relation between $\mathcal{L}$-index sets of finite traces and $\mathcal{L}$-indexed sets of traces: the relations $\leq$ and $\sqsubseteq$. Another variation on such a relation could be defined modulo a permutation of $\mathcal{L}$, so that the names of the locations do not really matter:

$$M \preceq T \text{ iff there is a permutation } \gamma \text{ over } \mathcal{L} \text{ such that } M(\ell) \text{ is a prefix of } T(\gamma(\ell)), \text{for each } \ell \in \mathcal{L}.$$

The study of the notion of safety that emerges by using $\preceq$ and how it compares with the above notions of safety is another direction for future work.

4 Decentralized Monitoring

When verifying a distributed system, having a central authority that performs any type of RV is a strong assumption, as it reduces the appeal of distribution, creates single points of failure during verification and can pose problems in storing all the traces locally, especially in light of the wide availability of multi-core systems. Thus, we study to what extent hyperproperties can be monitored by decentralized monitors; these avoid high contentions (leading to vastly improved scalability [Aceto et al., 2024c]) and also offer better privacy guarantees (whenever they are stationed locally at the nodes where the respective traces are generated [Francalanza et al., 2013; Jia et al., 2016]).

4.1 Decentralized Monitors: Syntax and Operational Semantics

We associate monitors to locations, denoted by ℓ, and monitors associated to ℓ monitor only actions required to happen at ℓ, thus allowing the processing of events to happen locally. This imposes some form of coordination between monitors at different locations. For this reason, we introduce the possibility for monitors to communicate.

We define a communication alphabet Com, ranged over by c, over some finite alphabet of communication constants Con (that contains Act), ranged over by γ, as:

$$\mathsf{Com} \ni c \ ::= \ (!G, \gamma) \ \mid \ (?G, \gamma),$$

where $G \subseteq \mathcal{L}$ and $\gamma \in \mathsf{Con}$. We have a communication action $(!G, \gamma)$ for sending γ to group G (multicast communication), and one $(?G, \gamma)$ for receiving γ from any monitor from the set G. Point-to-point communication can be represented by taking singleton sets for G.

The syntax of decentralized monitors is given by the following grammar:

$$\mathsf{DMon} \ni M ::= [m]_\ell \ \mid \ M \vee M \ \mid M \wedge M$$
$$\mathsf{LMon} \ni m ::= \mathsf{yes} \mid \mathsf{no} \mid \mathsf{end} \mid a.m \mid c.m \mid m + m \mid m \oplus m \mid m \otimes m \mid \mathsf{rec}\, x.m \mid x.$$

Notationally, in what follows we shall sometimes use $\diamond$ to denote any of $\wedge$ and $\vee$. Monitor $[m]_\ell$ denotes that m monitors the trace located at location ℓ, so, it is 'localized' at ℓ (this justifies the name LMon). Monitors assigned to the same trace run in parallel and observe identical events; contrary to Aceto et al. [2022a], monitors assigned to different traces are no longer completely isolated from each other, but can now communicate, which is the main new feature of the decentralized set-up.

Table 6. The Operational Semantics for Decentralized Local Monitors (up to Commutativity of +, ⊗ and ⊕), Where we Let λ Denote Either a, $(!G,\gamma)$ or $(?\ell,\gamma)$ for $\ell \in \mathcal{L}, G \subseteq \mathcal{L}$

$a.m \xrightarrow{a} m$	$\dfrac{\ell \in G}{(?G,\gamma).m \xrightarrow{(?\ell,\gamma)} m}$	$(!G,\gamma).m \xrightarrow{(!G,\gamma)} m$	$v \xrightarrow{a} v$	$\dfrac{m\{^{\mathsf{rec}\ x.m}/_x\} \xrightarrow{\lambda} m'}{\mathsf{rec}\ x.m \xrightarrow{\lambda} m'}$	$\dfrac{m \xrightarrow{a} m' \qquad n \xrightarrow{a} n'}{m \odot n \xrightarrow{a} m' \odot n'}$
$\dfrac{m \xrightarrow{(?\ell,\gamma)} m' \qquad n \xrightarrow{(?\ell,\gamma)} n'}{m \odot n \xrightarrow{(?\ell,\gamma)} m' \odot n'}$	$\dfrac{m \xrightarrow{\lambda} m'}{m + n \xrightarrow{\lambda} m'}$	$\dfrac{m \xrightarrow{(!G,\gamma)} m'}{m \odot n \xrightarrow{(!G,\gamma)} m' \odot n}$	$\dfrac{m \xrightarrow{(?\ell,\gamma)} m' \qquad n \not\xrightarrow{(?\ell,\gamma)}}{m \odot n \xrightarrow{(?\ell,\gamma)} m' \odot n}$		

The operational rules for $m \in \mathsf{LMon}$ are given in Table 6. Notice that, when we have parallel monitors, only one of them at a time can send; by contrast, all those that can receive from some location ℓ are forced to do so.

For $M \in \mathsf{DMon}$, the operational semantics can be found in Table 7 (the rules concerning communication) and Table 8 (the rules concerning action steps). The operational semantics in Table 7 defines multicast, where a monitor located at ℓ sends a message to group G and every monitor at a location in G that can receive from ℓ does so; every monitor that cannot, or that is not in G, does not change its state. The first four rules capture the judgment for inferring when all components of a monitor which are able to receive a certain γ sent from a location do so. Intuitively, ℓ is the location from which message γ was sent to group G, and $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$ indicates that every monitor in M located at a location in G that can receive γ from ℓ indeed has received γ and transitioned appropriately in N. The last two rules then actually define communication. In particular, the last rule in Table 7 implements multicast by stipulating that the outcome of the synchronization between a send action $\ell : (!G,\gamma)$ and a receive one of the form $G : (?\ell,\gamma)$ is the send action itself, which can be received by other monitors at locations in G in a larger monitor of which $M \diamond N$ is a subterm. We note, in passing, that monitors $M \in \mathsf{DMon}$ are 'input-enabled': for each M, G, ℓ and γ, there is always some M' such that $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} M'$. So the last rule in Table 7 (and its symmetric version) can always be applied when the send transition in its premise is available.

Monitors can also locally observe an action, as prescribed by a location-to-action function A; the rules are given in Table 8. Monitors at the same location observe the same action. If a monitor cannot take the action prescribed by A at its location, the monitor becomes end, as stipulated by the second rule given in Table 8. Note that it is not sufficient to trigger that rule when m cannot exhibit action $A(\ell)$: we also require that m cannot communicate. Note that the inability of m to exhibit action $A(\ell)$ is not sufficient to trigger that rule: we also require that m cannot communicate. Intuitively, this is because monitors exhibit an 'alternating' behavior in which they observe the next action produced by a system hypertrace and then embark in a sequence of communications with other monitors to inform them of what they observed. The order in which these communication messages are received is immaterial, since for receive actions a confluence property holds (at least for the monitors synthetized from formulae—see Lemma B.22).

As will be made clear in our definition of a weak bisimulation relation presented in Definition 4.2, such communications are interpreted as internal actions in monitor behavior. Therefore, the inability of some monitor $[m]_\ell$ to perform action $A(\ell)$ can only be gauged in 'stable states'—that is, monitor states in which no communication is possible. This design choice is akin to that underlying the definition of refusal testing presented in Phillips [1987] and of the stable-failures model for (Timed) CSP defined in Reed and Roscoe [1999] and Roscoe [1997], where the inability of a process to perform some action can only be determined in states that afford no internal computation steps.

Table 7. Operational Semantics for Communication of $M \in \mathsf{DMon}$ (up to Commutativity of $\wedge$ and $\vee$—Denoted by $\diamond$)

$$\frac{m \xrightarrow{(!G,\gamma)} m'}{[m]_\ell \xrightarrow{\ell:(!G,\gamma)} [m']_\ell} \qquad \frac{m \xrightarrow{(?\ell',\gamma)} m' \qquad \ell \in G}{[m]_\ell \overset{G\,:\,(?\ell',\gamma)}{\rightsquigarrow} [m']_\ell}$$

$$\frac{m \not\xrightarrow{(?\ell',\gamma)}}{[m]_\ell \overset{G\,:\,(?\ell',\gamma)}{\rightsquigarrow} [m]_\ell} \qquad \frac{\ell \notin G}{[m]_\ell \overset{G\,:\,(?\ell',\gamma)}{\rightsquigarrow} [m]_\ell}$$

$$\frac{M \overset{G\,:\,(?\ell,\gamma)}{\rightsquigarrow} M' \qquad N \overset{G\,:\,(?\ell,\gamma)}{\rightsquigarrow} N'}{M \diamond N \overset{G\,:\,(?\ell,\gamma)}{\rightsquigarrow} M' \diamond N'}$$

$$\frac{M \xrightarrow{\ell:(!G,\gamma)} M' \qquad N \overset{G\,:\,(?\ell,\gamma)}{\rightsquigarrow} N'}{M \diamond N \xrightarrow{\ell:(!G,\gamma)} M' \diamond N'}$$

Table 8. Operational Semantics for Actions of $M \in \mathsf{DMon}$ (up to Commutativity of $\wedge$ and $\vee$—Denoted by $\diamond$)

$$\frac{A(\ell) = a \qquad m \xrightarrow{a} m'}{[m]_\ell \xrightarrow{A} [m']_\ell}$$

$$\frac{A(\ell) = a \qquad m \not\xrightarrow{a} \qquad m \not\xrightarrow{\varepsilon}}{[m]_\ell \xrightarrow{A} [\mathsf{end}]_\ell}$$

$$\frac{M \xrightarrow{A} M' \qquad N \xrightarrow{A} N'}{M \diamond N \xrightarrow{A} M' \diamond N'}$$

Table 9. The Verdict Combination Rules for Decentralized Monitors (up to Commutativity of $\wedge$ and $\vee$—Denoted by $\diamond$)

$$\frac{m \Rrightarrow v}{[m]_\ell \Rrightarrow v} \qquad \frac{M \Rrightarrow \mathsf{end} \qquad N \Rrightarrow \mathsf{end}}{M \diamond N \Rrightarrow \mathsf{end}}$$

$$\frac{M \Rrightarrow \mathsf{no}}{M \wedge N \Rrightarrow \mathsf{no}} \qquad \frac{M \Rrightarrow \mathsf{yes} \qquad N \Rrightarrow v}{M \wedge N \Rrightarrow v}$$

$$\frac{M \Rrightarrow \mathsf{yes}}{M \vee N \Rrightarrow \mathsf{yes}} \qquad \frac{M \Rrightarrow \mathsf{no} \qquad N \Rrightarrow v}{M \vee N \Rrightarrow v}$$

Table 10. The Evolution of a Decentralized Monitor Instrumented on a Hypertrace

$$\frac{M \xrightarrow{A} M' \qquad T \xrightarrow{A} T'}{M \triangleright T \rightarrowtail M' \triangleright T'}$$

$$\frac{M \xrightarrow{\ell:(!G,\gamma)} M'}{M \triangleright T \rightarrowtail M' \triangleright T}$$

$$\frac{M \Rrightarrow v}{M \triangleright T \rightarrowtail v}$$

Verdict evaluation for $M \in \mathsf{DMon}$ is defined in Table 9 and relies on that for $m \in \mathsf{CMon}$ provided in Table 3. Finally, given a decentralized monitor M and a hypertrace T, the instrumentation of the monitor on the trace is described by the rules of Table 10. As before, we denote with $\rightarrowtail^*$ the reflexive-transitive closure of $\rightarrowtail$.

To conclude, let us show how the operational semantics works, by considering the following toy example:

$$\begin{aligned}
m_1 &\triangleq a_1.m_1' && \text{where} && m_1' \triangleq (!\{\ell_2,\ell_3\},a).m_1'' \\
m_2 &\triangleq a_2.m_2' && \text{where} && m_2' \triangleq (?\ell_1,a).m_2'' \\
m_3 &\triangleq a_3.m_3' && \text{where} && m_3' \triangleq (?\ell_1,a).m_3'' \\
m_4 &\triangleq a_4.m_4'.
\end{aligned}$$

Assume that we have four locations $\ell_1, \ldots, \ell_4$ and the 4-tuple of actions $A \triangleq [\ell_1 \mapsto a_1, \ell_2 \mapsto a_2, \ell_3 \mapsto a_3, \ell_4 \mapsto a]$. Then:

$$m_i \xrightarrow{a_i} m_i',$$

for all $i \in \{1, \ldots, 4\}$ and so:

$$[m_i]_{\ell_i} \xrightarrow{A} [m_i']_{\ell_i} \qquad \text{for every } i \in \{1,2,3\} \text{ (since } a_i = A(\ell_i))$$
$$[m_4]_{\ell_4} \xrightarrow{A} [\mathsf{end}]_{\ell_4} \qquad \text{(since } a_4 \neq A(\ell_4) = a).$$

Thus:

$$\bigwedge_{i=1}^{4} [m_i]_{\ell_i} \xrightarrow{A} \bigwedge_{i=1}^{3} [m_i']_{\ell_i} \;\;\wedge\;\; [\mathsf{end}]_{\ell_4}.$$

Now, we can handle communication. First, we can infer that:

$$m_1' \xrightarrow{(!\{\ell_2,\ell_3\},a)} m_1'' \qquad \text{and} \qquad m_i' \xrightarrow{(?\ell_1,a)} m_i'' \quad \text{for } i \in \{2,3\},$$

and so:

$$[m_1']_{\ell_1} \xrightarrow{\ell_1:(!\{\ell_2,\ell_3\},a)} [m_1'']_{\ell_1} \qquad \text{and} \qquad [m_i']_{\ell_i} \overset{\{\ell_2,\ell_3\}:(?\ell_1,a)}{\rightsquigarrow} [m_i'']_{\ell_i} \quad \text{for } i \in \{2,3\}$$

By observing that end $\overset{(?\ell_1,a)}{\not\longrightarrow}$, we can also infer that:

$$[\mathsf{end}]_{\ell_4} \overset{\{\ell_2,\ell_3\}:(?\ell_1,a)}{\rightsquigarrow} [\mathsf{end}]_{\ell_4},$$

and so:

$$[m_2']_{\ell_2} \wedge [m_3']_{\ell_3} \wedge [\mathsf{end}]_{\ell_4} \overset{\{\ell_2,\ell_3\}:(?\ell_1,a)}{\rightsquigarrow} [m_2'']_{\ell_2} \wedge [m_3'']_{\ell_3} \wedge [\mathsf{end}]_{\ell_4}.$$

Finally, we have that:

$$\bigwedge_{i=1}^{3} [m_i']_{\ell_i} \;\;\wedge\;\; [\mathsf{end}]_{\ell_4} \xrightarrow{\ell_1:(!\{\ell_2,\ell_3\},a)} \bigwedge_{i=1}^{3} [m_i'']_{\ell_i} \;\;\wedge\;\; [\mathsf{end}]_{\ell_4}.$$

The final verdict of this monitor will be end if all m_i'' (for $i \in \{1,2,3\}$) evaluate to either end or yes; in contrast, if any of them evaluates to no, then the monitor evaluates to no. In particular, because of the failure in observing a in ℓ_4 (as stipulated by A), there is no way to obtain a yes verdict.

4.2 Synthesizing Decentralized Monitors Correctly

In this section we describe how to synthesize decentralized monitors 'correctly' from formulas, i.e., such that their behavior corresponds to that of the corresponding centralized monitors. The advantage of this approach is that it simplifies the proof that monitors synthesized via a 'correct' decentralized synthesis function are sound and violation-complete, by utilizing the correspondence to centralized monitors. Moreover, it identifies desirable properties of a 'correct' decentralized synthesis function that can guide the development of further automated decentralized-monitor synthesis algorithms.

We first define the correspondence between centralized and decentralized monitors and show that this correspondence is sufficient to obtain soundness and violation-completeness in the decentralized setting from the corresponding results in the centralized setting (Theorems 3.5 and 3.8). In the remainder of the section, given a synthesis function which takes as inputs a formula φ and a mapping σ from location variables to locations, and outputs a monitor $\mathcal{M}_\sigma(\varphi) \in \mathsf{DMon}$, we specify criteria that allow us to derive this correspondence.

The correspondence between the centralized and the decentralized monitors is characterized as a weak bisimulation; this first requires defining sequences of communication actions, as the analogous of unobservable transitions in process calculi.

Definition 4.1. We write $M \twoheadrightarrow M'$ to denote the existence of an integer $h > 0$, of h monitors $M_1, \ldots, M_h$ and of $h-1$ locations $\ell_1, \ldots, \ell_{h-1}$ and communication actions $c_1, \ldots, c_{h-1}$ such that $M_1 = M$, $M_h = M'$, and $M_i \xrightarrow{\ell_i : c_i} M_{i+1}$ (for every $i = 1, \ldots, h-1$).

Similarly, we write $m \twoheadrightarrow m'$ to denote the existence of an integer $h > 0$, of h local monitors $m_1, \ldots, m_h$ and of $h-1$ actions $c_1, \cdots c_{h-1} \in \{(!G, \gamma), (?\ell, \gamma) \mid G \subseteq \mathcal{L}, \ell \in \mathcal{L}, \gamma \in \mathsf{Con}\}$ such that $m_1 = m$, $m_h = m'$ and $m_i \xrightarrow{c_i} m_{i+1}$ (for every $i = 1, \ldots, h-1$).

Notice that, by definition of $\rightarrow$ on communicating monitors, each c_i is $(!G_i, \gamma_i)$, for some $G_i \subseteq \mathcal{L}$ and $\gamma_i \in \mathsf{Con}$.

Definition 4.2. A binary relation $\mathcal{R}$ over $\mathsf{DMon} \times \mathsf{CMon}$ is a weak bisimulation if and only if, whenever $M\mathcal{R}m$, it holds that:

(1) $\exists M' \in \mathsf{DMon}$ such that $M \twoheadrightarrow M'$ and $M' \Rrightarrow v$ if and only if $m \Rrightarrow v$.
(2) If $M \xrightarrow{A} M'$ then $\exists m' \in \mathsf{CMon}$ such that $m \xrightarrow{A} m'$ and $M'\mathcal{R}m'$.
(3) If $M \xrightarrow{c} M'$ then $M'\mathcal{R}m$, where $c = \ell : (!G, \gamma)$ for some $\ell \in \mathcal{L}$, $G \subseteq \mathcal{L}$, $\gamma \in \mathsf{Con}$.
(4) If $m \xrightarrow{A} m'$ then there exist M_1, M_2, M' such that $M \twoheadrightarrow M_1 \xrightarrow{A} M_2 \twoheadrightarrow M'$ and $M'\mathcal{R}m'$.

We now prove four facts concerning the instrumentation rules of weakly bisimilar monitors, which we use to prove Proposition 4.4, which shows that centralized and decentralized monitors behaviorally agree.

Lemma 4.3. *Let $\mathcal{R}$ be a weak bisimulation relation such that $M\mathcal{R}m$. Then the following statements hold*:

(1) *If there exists M' such that $M \triangleright T \rightarrowtail M' \triangleright T'$, then there exists m' such that $m \triangleright T \rightarrowtail^* m' \triangleright T'$ and $M'\mathcal{R}m'$.*
(2) *If there exists m' such that $m \triangleright T \rightarrowtail m' \triangleright T'$, then there exists M' such that $M \triangleright T \rightarrowtail^* M' \triangleright T'$ and $M'\mathcal{R}m'$.*
(3) *If $M \triangleright T \rightarrowtail v$ then $m \triangleright T \rightarrowtail v$.*
(4) *If $m \triangleright T \rightarrowtail v$ then $M \triangleright T \rightarrowtail^* v$.*

Proof. For the first item, we assume that $M \triangleright T \rightarrowtail M' \triangleright T'$. Hence we have that $M \xrightarrow{A} M'$ and $T \xrightarrow{A} T'$ or $M \xrightarrow{\ell:(!G,\gamma)} M'$ and $T = T'$. In the first case, we use the definition of weak bisimulation to obtain that $m \xrightarrow{A} m'$ for $M'\mathcal{R}m'$, from which we conclude that $m \triangleright T \rightarrowtail m' \triangleright T'$. In the second case, we use the definition of weak bisimulation to obtain that $M'\mathcal{R}m$ and we also know that $m \triangleright T \rightarrowtail^* m \triangleright T$.

For the second item, we assume that $m \triangleright T \rightarrowtail m' \triangleright T'$. Hence we have that $m \xrightarrow{A} m'$ and $T \xrightarrow{A} T'$. From the definition of weak bisimulation, there exist M_1, M_2, M' such that $M \twoheadrightarrow M_1 \xrightarrow{A} M_2 \twoheadrightarrow M'$ and $M'\mathcal{R}m'$. Thus, we can conclude that $M \triangleright T \rightarrowtail^* M_1 \triangleright T$, $M_1 \triangleright T \rightarrowtail M_2 \triangleright T'$ and $M_2 \triangleright T' \rightarrowtail^* M' \triangleright T'$, from which we obtain that $M \triangleright T \rightarrowtail^* M' \triangleright T'$.

For the third item we assume that $M \triangleright T \rightarrowtail v$, from which we conclude that $M \Rrightarrow v$. From the definition of weak bisimulation we obtain immediately that $m \Rrightarrow v$ and thus $m \triangleright T \rightarrowtail v$.

For the fourth item we assume that $m \triangleright T \rightarrowtail v$, from which we conclude that $m \Rrightarrow v$. From the definition of weak bisimulation we obtain immediately that $M \twoheadrightarrow M'$ and $M' \Rrightarrow v$ and thus $M \triangleright T \rightarrowtail^* v$. □

Proposition 4.4. *Let $\mathcal{R}$ be a weak bisimulation such that $M\mathcal{R}m$. Then, $M \triangleright T \rightarrowtail^* v$ if and only if $m \triangleright T \rightarrowtail^* v$.*

Proof. For the left to right direction, assume that $M \triangleright T \rightarrowtail^* v$. By definition there exist an integer $h > 0$ and $M_1 \triangleright T_1, \dots, M_h \triangleright T_h$ such that $M_1 = M$, $T_1 = T$, $M_i \triangleright T_i \rightarrowtail M_{i+1} \triangleright T_{i+1}$ (for every $i = 1, \dots, h-1$), and $M_h \triangleright T_h \rightarrowtail v$. We proceed by induction on h. The base case holds because of Lemma 4.3 (Item 3). In the inductive step we use Lemma 4.3 (Item 1) to obtain $m \triangleright T \rightarrowtail^* m_2 \triangleright T_2$ for some m_2 and T_2 such that $M_2\mathcal{R}m_2$. Then we can use the induction hypothesis to conclude that $m_2 \triangleright T_2 \rightarrowtail^* v$. Thus, $m \triangleright T \rightarrowtail^* v$.

For the right to left direction, the proof is identical but uses Lemma 4.3 (Item 4) and Lemma 4.3 (Item 2) instead. □

This allows us to obtain violation-completeness and soundness for decentralized monitors from the corresponding results for centralized monitors:

Corollary 4.5 (Soundness). *Let $T \in \mathsf{HTrc}_{\mathcal{L}}$, $\varphi \in$ Hyper-maxHML be a closed formula such that $\mathcal{M}_{\emptyset}(\varphi)$ is defined, and $\mathcal{R}$ a weak bisimulation such that $(\mathcal{M}_{\emptyset}(\varphi), \mathsf{cm}_{\emptyset}(\varphi)) \in \mathcal{R}$. If $\mathcal{M}_{\emptyset}(\varphi) \triangleright T \rightarrowtail^*$ no, then $T \notin [[\varphi]]$; if $\mathcal{M}_{\emptyset}(\varphi) \triangleright T \rightarrowtail^*$ yes, then $T \in [[\varphi]]$.*

Corollary 4.6 (Violation Completeness). *Let $T \in \mathsf{HTrc}_{\mathcal{L}}$, $\varphi \in$ Hyper-maxHML be a closed formula such that $\mathcal{M}_{\emptyset}(\varphi)$ is defined, and $\mathcal{R}$ a weak bisimulation such that $(\mathcal{M}_{\emptyset}(\varphi), \mathsf{cm}_{\emptyset}(\varphi)) \in \mathcal{R}$. If $T \notin [[\varphi]]$, then $\mathcal{M}_{\emptyset}(\varphi) \triangleright T \rightarrowtail^*$ no.*

We now describe sufficient conditions for any decentralized synthesis function such that there is a weak bisimulation between the centralized and the decentralized monitors synthesized from a formula φ and a location environment σ. Whenever we write $M \xrightarrow{c} N$ for $M, N \in \mathsf{DMon}$, we assume that $c \in \{\ell : (!G, \gamma) \mid \ell \in \mathcal{L}, G \subseteq \mathcal{L}, \gamma \in \mathsf{Con}\}$, as per the labeling of the communication transitions of decentralized monitors. We write $[m]_\ell \in M$, for $M \in \mathsf{DMon}$, if $[m]_\ell$ is one of its constituents: formally, $[m]_\ell \in [m]_\ell$ and, if $[m]_\ell \in M$, then $[m]_\ell \in M \diamond N$ and $[m]_\ell \in N \diamond M$ (recall that $\diamond$ denotes either $\wedge$ or $\vee$). We start by defining when $M \in \mathsf{DMon}$ can(not) communicate:

Definition 4.7. Let $M \in \mathsf{DMon}$. We say that M can communicate if there exists $[m]_\ell \in M$ such that $m \xrightarrow{c} n$, for some $c \in \mathsf{Com}$ and $n \in \mathsf{LMon}$. Otherwise, we say that M cannot communicate.

Definition 4.8. We say that a monitor synthesis $\mathcal{M}_{-}(-)$ is *principled* when it satisfies the following conditions, for every formula φ and environment σ such that $\mathcal{M}_{\sigma}(\varphi)$ is defined:

Verdict Agreement: For every verdict v, $\mathsf{cm}_{\sigma}(\varphi) \Rrightarrow v$ if and only if $\mathcal{M}_{\sigma}(\varphi) \Rrightarrow v$;

Verdict Irrevocability: For every verdict v and $\mathcal{M}_{\sigma}(\varphi) \xrightarrow{A} M_1 \twoheadrightarrow M_2 \twoheadrightarrow M$, if $M_2 \Rrightarrow v$ then $M \Rrightarrow v$;

Reactivity: For every A, if φ has no free recursion variables, then there exists M such that $\mathcal{M}_{\sigma}(\varphi) \xrightarrow{A} M$;

Bounded Communication: For every $\mathcal{M}_{\sigma}(\varphi) \xrightarrow{A} M \twoheadrightarrow M'$, there exists M'' such that $M' \twoheadrightarrow M''$ and M'' cannot communicate;

Processing-Communication Alternation: For every $\mathcal{M}_{\sigma}(\varphi) \xrightarrow{A} M \twoheadrightarrow M_1$:

(1) $\mathcal{M}_{\sigma}(\varphi)$ cannot communicate, and

(2) $M_1 \xrightarrow{c} M_2$ implies $M_1 \overset{A'}{\not\rightarrow}$, for every A';

Formula Convergence: If $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M'$, with M' that cannot communicate, and $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$, for some formula φ' and environment σ', then $M' = \mathcal{M}_{\sigma'}(\varphi')$.

From the Formula Convergence property, we can immediately derive an auxiliary property:

Lemma 4.9 (Uniqueness). *If $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M_1$, with M_1 that cannot communicate, and $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M_2$, with M_2 that cannot communicate, then $M_1 = M_2$.*

Proof. Indeed, since $\mathsf{cm}_\sigma(\varphi)$ has a possible transition for every A, by Lemma 3.3 we know that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$. By applying Formula Convergence twice, we obtain $M_1 = \mathcal{M}_{\sigma'}(\varphi') = M_2$. □

Let $\mathcal{M}_{-}(-)$ be a decentralized synthesis function. We define relation $\mathcal{R}_{\mathcal{M}}$ as follows:

$$\mathcal{R}_{\mathcal{M}} \triangleq \mathcal{R}_1 \cup \mathcal{R}_2$$
$$\mathcal{R}_1 \triangleq \{(\mathcal{M}_\sigma(\varphi), \mathsf{cm}_\sigma(\varphi)) \mid \mathsf{fv}_\mathsf{l}(\varphi) \subseteq \mathsf{dom}(\sigma) \text{ and } \mathsf{fv}_\mathsf{r}(\varphi) = \emptyset\}$$
$$\mathcal{R}_2 \triangleq \left\{(M', \mathsf{cm}_{\sigma'}(\varphi')) \mid \mathsf{fv}_\mathsf{l}(\varphi) \subseteq \mathsf{dom}(\sigma),\ \mathsf{fv}_\mathsf{r}(\varphi) = \emptyset \text{ and } \mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M' \twoheadrightarrow \mathcal{M}_{\sigma'}(\varphi')\right\}.$$

The crucial property of any principled synthesis function is the following:

Theorem 4.10. *For every principled synthesis $\mathcal{M}_{-}(-)$, $\mathcal{R}_{\mathcal{M}}$ is a weak bisimulation.*

Proof. To prove that $\mathcal{R}_{\mathcal{M}}$ is a weak bisimulation, we first consider a pair $(\mathcal{M}_\sigma(\varphi), \mathsf{cm}_\sigma(\varphi)) \in \mathcal{R}_1$. The first condition of weak bisimulation holds because, from Processing-Communication Alternation and Reactivity, we conclude that $\mathcal{M}_\sigma(\varphi) \overset{c}{\not\rightarrow}$, and thus $M = M'$. Then the result follows from Verdict Agreement.

For the second condition of weak bisimulation, we assume that $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M$. Since $\mathsf{cm}_\sigma(\varphi)$ has a possible transition for every A, by Lemma 3.3 we know that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$ for some σ' and φ'. From Bounded Communication, we can use Formula Convergence to obtain that $M \twoheadrightarrow \mathcal{M}_{\sigma'}(\varphi')$. By definition, $M\mathcal{R}_2\mathsf{cm}_{\sigma'}(\varphi')$ and this suffices to conclude.

The third condition of weak bisimulation is not relevant, as $\mathcal{M}_\sigma(\varphi) \overset{c}{\not\rightarrow}$ via Processing-Communication Alternation and Reactivity. To check the fourth condition, we assume that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$ (because of Lemma 3.3, we know that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A}$ results in a monitor synthesized from a formula). Via Reactivity we conclude that $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M$ for some M. Then we can use Bounded Communication and Formula Convergence to derive that $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow \mathcal{M}_{\sigma'}(\varphi')$. As $\mathcal{M}_{\sigma'}(\varphi')\mathcal{R}_1\mathsf{cm}_{\sigma'}(\varphi')$, we can conclude.

Now consider a pair $(M', \mathsf{cm}_{\sigma'}(\varphi')) \in \mathcal{R}_2$, with $\mathcal{M}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M' \twoheadrightarrow \mathcal{M}_{\sigma'}(\varphi')$ for some σ, φ and M. To verify the first condition from left to right, we assume $M' \twoheadrightarrow M''$ and $M'' \Rrightarrow v$. Since $\mathsf{cm}_\sigma(\varphi)$ has a possible transition for every A and Lemma 3.3, we know that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma''}(\varphi'')$ for some σ'' and φ''. Hence, we can use Bounded Communication and Formula Convergence to conclude that $M'' \twoheadrightarrow \mathcal{M}_{\sigma''}(\varphi'')$. Furthermore, we obtain from Uniqueness and Processing-Communication Alternation that $\mathcal{M}_{\sigma'}(\varphi') = \mathcal{M}_{\sigma''}(\varphi'')$. Then we apply Verdict Irrevocability to conclude that $\mathcal{M}_{\sigma'}(\varphi') \Rrightarrow v$. Finally, we can conclude using Verdict Agreement. For the right to left direction, we use Verdict Agreement to obtain that $\mathcal{M}_{\sigma'}(\varphi') \Rrightarrow v$, which immediately satisfies the first condition of weak bisimulation. For the second condition of weak bisimulation, we assume that $M' \xrightarrow{A'} M''$. From Processing-Communication Alternation, we conclude that $M' \overset{c}{\not\to}$, and thus that $M' = \mathcal{M}_{\sigma'}(\varphi')$. This sends back to the second condition for pairs in $\mathcal{R}_1$.

For the third condition, we assume that $M' \xrightarrow{c} M''$. Since $\mathsf{cm}_\sigma(\varphi)$ has a possible transition for every A and Lemma 3.3, we know that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma''}(\varphi'')$ for some σ'' and φ''. Hence, we can use Bounded Communication and Formula Convergence to conclude that $M'' \twoheadrightarrow \mathcal{M}_{\sigma''}(\varphi'')$. Furthermore, we obtain from Uniqueness and Processing-Communication Alternation that $\mathcal{M}_{\sigma'}(\varphi') = \mathcal{M}_{\sigma''}(\varphi'')$. By definition, $(M'', \mathsf{cm}_{\sigma'}(\varphi')) \in \mathcal{R}_2$ and this suffices to conclude.

To check the fourth condition, we assume that $\mathsf{cm}_{\sigma'}(\varphi') \xrightarrow{A'} \mathsf{cm}_{\sigma''}(\varphi'')$. We note that if $N \to N'$ and N has no free recursion variables, then neither does N'—by straightforward induction on the derivation of $N \to N'$. Therefore, $\mathcal{M}_{\sigma'}(\varphi')$ has no free recursion variables, and neither does φ'. Via Reactivity we conclude that $\mathcal{M}_{\sigma'}(\varphi') \xrightarrow{A'} M_1$. Then we use Bounded Communication and Formula Convergence to derive that $M_1 \twoheadrightarrow \mathcal{M}_{\sigma''}(\varphi'')$, resulting in $M' \twoheadrightarrow \mathcal{M}_{\sigma'}(\varphi') \xrightarrow{A'} M_1 \twoheadrightarrow \mathcal{M}_{\sigma''}(\varphi'')$, which concludes the proof. □

4.3 From Formulas to Decentralized Monitors

We now describe how to synthesize decentralized monitors for a fragment of Hyper-maxHML, and show that this synthesis function satisfies Definition 4.8. This allows us to apply Theorem 4.10 and obtain soundness and violation-completeness of these synthesized monitors.

In what follows, we consider formulas from PHyper-recHML, the subset of Hyper-recHML given by the following grammar (see Section 5 for a discussion on the fragment chosen):

$$\varphi ::= \forall\pi.\varphi \mid \exists\pi.\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \psi$$
$$\psi ::= \mathsf{tt} \mid \mathsf{ff} \mid \psi\wedge\psi \mid \psi\vee\psi \mid \pi=\pi \mid \pi\neq\pi \mid \max x.\psi \mid \min x.\psi \mid x \mid [a_\pi]\psi \mid \langle a_\pi\rangle\psi.$$

We denote the class of formulas of type ψ with Qf (quantifier-free). PHyper-recHML is a subset of Hyper-recHML and thus its semantics over $\mathsf{HTrc}_{\mathcal{L}}$ is the one given in Table 1.

We synthesize decentralized monitors for the fragment of PHyper-recHML only containing formulas of type φ without least fixed-points, which we call PHyper-maxHML. The synthesis for decentralized monitors is given in Table 11. To derive monitors belonging to DMon for formulas of type φ (through function DM), we need to derive a monitor belonging to LMon for formulas of type $\psi \in \mathsf{Qf}$ (through function Dm). The synthesis functions are parametrized by a location $\ell \in \mathcal{L}$ and a partial function σ from Π to $\mathcal{L}$ that is defined for every free location variable in ψ and φ.

Note that, in the definition of $\mathrm{DM}_\sigma(\psi)$, $\mathsf{cm}_\sigma(\psi)$ is the monitor resulting from the centralized synthesis function defined in Table 5. Intuitively $\mathrm{DM}_\sigma(\psi)$ synthesizes a local monitor at each location relevant to ψ, which are the locations associated by σ to the free location variables in ψ. If $\sigma = \emptyset$, then ψ does not have any free location variable and it is a Boolean combination of tt and ff; thus, there is no need for communication between locations. In fact, a verdict can be obtained from ψ immediately and it coincides with the one reached in the centralized synthesis.

Thus, every closed formula φ on which we apply our synthesis is:

(1) either *trivial*, i.e., φ is logically equivalent to tt or ff,
(2) or is such that every subformula $\psi \in \mathsf{Qf}$ of φ is in the scope of a quantifier.

For non-trivial formulas, the $\sigma = \emptyset$ case for $\mathrm{DM}_\sigma(\psi)$ never applies, and we can ignore it. The decentralized monitor for a closed formula φ is $\mathrm{DM}_\emptyset(\varphi)$.

Remark 4.11. In the first clause of the definition of the synthesis function for box formulas, it might seem superfluous to send a message also when the monitor observes some $b \neq a$. However,

Table 11. Decentralized Monitor Synthesis, Where ℓ_0 is Any Fixed Element of $\mathcal{L}$

$$\textsc{dM}_\sigma(\forall\pi.\varphi) = \bigwedge_{\ell\in\mathcal{L}} \textsc{dM}_{\sigma[\pi\mapsto\ell]}(\varphi) \qquad \textsc{dM}_\sigma(\exists\pi.\varphi) = \bigvee_{\ell\in\mathcal{L}} \textsc{dM}_{\sigma[\pi\mapsto\ell]}(\varphi)$$

$$\textsc{dM}_\sigma(\varphi\wedge\varphi') = \textsc{dM}_\sigma(\varphi) \wedge \textsc{dM}_\sigma(\varphi') \qquad \textsc{dM}_\sigma(\varphi\vee\varphi') = \textsc{dM}_\sigma(\varphi) \vee \textsc{dM}_\sigma(\varphi')$$

$$\textsc{dM}_\sigma(\psi) = \begin{cases} \bigvee_{\ell\in\mathsf{rng}(\sigma)} [\textsc{dm}^\ell_\sigma(\psi)]_\ell & \text{if } \sigma \neq \emptyset \\ [v]_{\ell_0} & \text{if } \sigma = \emptyset \wedge \textsc{cm}_\sigma(\psi) \Rrightarrow v \end{cases}$$

$$\textsc{dm}^\ell_\sigma(\mathsf{tt}) = \mathsf{yes} \qquad \textsc{dm}^\ell_\sigma(\mathsf{ff}) = \mathsf{no} \qquad \textsc{dm}^\ell_\sigma(x) = x \qquad \textsc{dm}^\ell_\sigma(\max x.\psi) = \mathsf{rec}\ x.\textsc{dm}^\ell_\sigma(\psi)$$

$$\textsc{dm}^\ell_\sigma(\psi\wedge\psi') = \textsc{dm}^\ell_\sigma(\psi) \otimes \textsc{dm}^\ell_\sigma(\psi') \qquad \textsc{dm}^\ell_\sigma(\psi\vee\psi') = \textsc{dm}^\ell_\sigma(\psi) \oplus \textsc{dm}^\ell_\sigma(\psi')$$

$$\textsc{dm}^\ell_\sigma(\pi=\pi') = \begin{cases} \mathsf{yes} & \text{if } \sigma(\pi) = \sigma(\pi') \\ \mathsf{no} & \text{otherwise} \end{cases} \qquad \textsc{dm}^\ell_\sigma(\pi\neq\pi') = \begin{cases} \mathsf{yes} & \text{if } \sigma(\pi) \neq \sigma(\pi') \\ \mathsf{no} & \text{otherwise} \end{cases}$$

$$\textsc{dm}^\ell_\sigma([a_\pi]\psi) = \begin{cases} a.(!(\mathsf{rng}(\sigma)\backslash\{\ell\}),a).\textsc{dm}^\ell_\sigma(\psi) + \sum\limits_{b\neq a} b.(!(\mathsf{rng}(\sigma)\backslash\{\ell\}),b).\mathsf{yes} & \text{if } \sigma(\pi)=\ell \\ \sum\limits_{b\in\mathsf{Act}} b.\Big((?\{\sigma(\pi)\},a).\textsc{dm}^\ell_\sigma(\psi) + \sum\limits_{b\neq a}(?\{\sigma(\pi)\},b).\mathsf{yes}\Big) & \text{otherwise} \end{cases}$$

$$\textsc{dm}^\ell_\sigma(\langle a_\pi\rangle\psi) = \begin{cases} a.(!(\mathsf{rng}(\sigma)\backslash\{\ell\}),a).\textsc{dm}^\ell_\sigma(\psi) + \sum\limits_{b\neq a} b.(!(\mathsf{rng}(\sigma)\backslash\{\ell\}),b).\mathsf{no} & \text{if } \sigma(\pi)=\ell \\ \sum\limits_{b\in\mathsf{Act}} b.\Big((?\{\sigma(\pi)\},a).\textsc{dm}^\ell_\sigma(\psi) + \sum\limits_{b\neq a}(?\{\sigma(\pi)\},b).\mathsf{no}\Big) & \text{otherwise} \end{cases}$$

this is important to make sure monitors do not deadlock. To see this, consider a synthesis where that definition instead looks like:

$$\textsc{dm}^\ell_\sigma([a_\pi]\psi) = \begin{cases} a.(!(\mathsf{rng}(\sigma)\backslash\{\ell\}),a).\textsc{dm}^\ell_\sigma(\psi) + \sum\limits_{b\neq a} b.\mathsf{yes} & \text{if } \sigma(\pi)=\ell \\ \sum\limits_{b\in\mathsf{Act}} b.(?\{\sigma(\pi)\},a).\textsc{dm}^\ell_\sigma(\psi) & \text{otherwise.} \end{cases}$$

Consider $\mathsf{Act} = \{a,b\}$, $\mathcal{L} = \{\ell,\ell'\}$ and some hypertrace T such that $T(\ell) = b.t_1$ and $T(\ell') = b.t_2$ for some traces t_1 and t_2. Now consider $m \otimes n$, where $m = \textsc{dm}^\ell_\sigma([a_\pi]\psi)$, $n = \textsc{dm}^\ell_\sigma([a_{\pi'}]\psi')$, $\sigma(\pi) = \ell$ and $\sigma(\pi') = \ell'$. For $A(\ell) = A(\ell') = b \neq a$, we then get $m \xrightarrow{A(\ell)} \mathsf{yes}$ and $n \xrightarrow{A(\ell')} (?\{\sigma(\pi')\},a).\textsc{dm}^\ell_\sigma(\psi')$, and monitor $\mathsf{yes}\otimes(?\{\sigma(\pi')\},a).\textsc{dm}^\ell_\sigma(\psi')$ is stuck because the receive action of the monitor $(?\{\sigma(\pi')\},a).\textsc{dm}^\ell_\sigma(\psi')$ has no matching send. It is precisely to avoid these scenarios that we make sure that, for each sending transition, there is a corresponding receiving transition, and a monitor always sends the last action it read to all other locations in the range of the environment σ.

The same applies to the synthesis of diamonds, being the latter the same as the synthesis for box formulas in Table 11 with no verdicts in place of yes. ◀

Example 4.12. In order to highlight the inter-monitor communication, we consider the following formula:

$$\varphi = \exists\pi.\exists\pi'.([a_\pi]\mathsf{ff} \wedge [b_{\pi'}]\mathsf{ff}),$$

over $\mathcal{L} = \{1,2\}$ and $\mathsf{Act} = \{a,b\}$, which states that the two traces must start with different actions. Indeed, $[a_\pi]\mathsf{ff}$ requires that the trace at the location bound to π does not start with an a and the one at the location bound to π' does not start with a b. Hence, the only way to satisfy this formula is to associate different locations to π and π', in such a way that the trace at the location bound to π starts with a b and the trace at the location bound to π' starts with an a. By letting $\sigma = [\pi \mapsto \ell, \pi' \mapsto \ell']$, the synthesis function applied to this property gives:

$$\mathrm{DM}_{\emptyset}(\varphi) = \bigvee_{\ell,\ell' \in \mathcal{L}} \bigvee_{\ell'' \in \{\ell,\ell'\}} \left[\mathrm{Dm}^{\ell''}_{\sigma}([a_\pi]\mathsf{ff} \wedge [b_{\pi'}]\mathsf{ff})\right]_{\ell''},$$

where:

$$\mathrm{Dm}^{\ell''}_{\sigma}([a_\pi]\mathsf{ff} \wedge [b_{\pi'}]\mathsf{ff}) = \begin{cases} (a.(!\emptyset,a).\mathsf{no} + b.(!\emptyset,b).\mathsf{yes}) \otimes & \text{if } \ell = \ell' = \ell'' \\ \quad (b.(!\emptyset,b).\mathsf{no} + a.(!\emptyset,a).\mathsf{yes}) & \\ (a.(!\{\ell'\},a).\mathsf{no} + b.(!\{\ell'\},b).\mathsf{yes}) \otimes & \text{if } \ell \neq \ell' \text{ and } \ell'' = \ell \\ \quad (a.((?\{\ell'\},b).\mathsf{no} + (?\{\ell'\},a).\mathsf{yes}) + & \\ \quad\; b.((?\{\ell'\},b).\mathsf{no} + (?\{\ell'\},a).\mathsf{yes})) & \\ (a.((?\{\ell\},a).\mathsf{no} + (?\{\ell\},b).\mathsf{yes}) + & \text{if } \ell \neq \ell' \text{ and } \ell'' = \ell' \\ \; b.((?\{\ell\},a).\mathsf{no} + (?\{\ell\},b).\mathsf{yes})) & \\ \otimes\, (b.(!\{\ell\},b).\mathsf{no} + a.(!\{\ell\},a).\mathsf{yes}) & \end{cases}$$ ◀

Remark 4.13. We explicitly added in Table 11 a rule for formulas of the form $\langle a_\pi \rangle \psi$ even though such formulas are logically equivalent to:

$$[a_\pi]\psi \wedge \textstyle\bigwedge_{b \neq a}[b_\pi]\mathsf{ff}. \tag{3}$$

However, working with this formulation of the diamond has drawbacks in terms of the size of the resulting monitor. To showcase this, consider the decentralized synthesis applied on Wolper's property (φ^h_e of Equation (2) from Example 2.4), expressed here as $\exists\pi.\psi$, with:

$$\psi = \max x.(\psi_1 \wedge \psi_2) \qquad \psi_1 = [a_\pi]\langle a_\pi \rangle x \qquad \psi_2 = [b_\pi]\langle a_\pi \rangle x.$$

Let $\mathcal{L} = \{1,2\}$ and $\mathsf{Act} = \{a,b\}$. The synthesis is applied thus:

$$\mathrm{DM}_{\emptyset}(\varphi^h_e) = \textstyle\bigvee_{\ell \in \mathcal{L}} \left[\mathsf{rec}\ x.\left(\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\psi_1) \otimes \mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\psi_2)\right)\right]_{\ell},$$

with:

$$\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\psi_1) = a.(!\emptyset,a).\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\langle a_\pi \rangle x) + b.(!\emptyset,b).\mathsf{yes}$$
$$\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\psi_2) = b.(!\emptyset,b).\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\langle a_\pi \rangle x) + a.(!\emptyset,a).\mathsf{yes},$$

and, by using the decentralized monitor synthesis for diamond formulas based on Equation (3):

$$\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\langle a_\pi \rangle x) = (a.(!\emptyset,a).x + b.(!\emptyset,b).\mathsf{yes}) \ \otimes\ (b.(!\emptyset,b).\mathsf{no} + a.(!\emptyset,a).\mathsf{yes}). \tag{4}$$

This indicates that the synthesis for diamonds based on Equation (3) leads to monitors with a high degree of parallelism; for simplicity, we have just two parallel components in Equation (4) because we assumed to have just two actions. However, $|\mathsf{Act}| - 1$ parallel conjunctions are required in general for every diamond (and may lead to an exponential blow up with several diamonds in sequence). By contrast, using the rule for diamonds given in Table 11 reduces Equation (4) to:

$$\mathrm{Dm}^{\ell}_{[\pi\mapsto\ell]}(\langle a_\pi \rangle x) = a.(!\emptyset,a).x + b.(!\emptyset,b).\mathsf{no},$$

and the synthesized monitor now contains no occurrence of any parallel operator. ◀

Soundness and violation completeness for the synthesis defined in Table 11 follow from Corollaries 4.5 and 4.6 by using Theorem 4.10, once we prove the following key result:

THEOREM 4.14. *The synthesis function* DM *defined in Table 11 is principled.*

The proof is long and technical. It is carried out by showing that DM satisfies the six properties of Definition 4.8. The details are in Appendix B.

5 Conclusion

We provided two methods to synthesize monitors for hyperproperties expressed as fragments of Hyper-recHML. Our first synthesis procedure constructs monitors that analyze hypertraces in a centralized manner and are guaranteed to correctly detect all violations of the respective formula, as long as it does not have a least fixed-point operator. Our second synthesis algorithm constructs monitors that operate in a decentralized manner and communicate with one another using multicast to share relevant information between them. The decentralized-monitor synthesis provides the same correctness guarantees as the centralized one, but is only defined for formulas with trace quantifiers that do not appear inside any fixed-point operator. This additional restriction, which is natural and present in many monitoring set-ups for hyperlogics, e.g., [Agrawal and Bonakdarpour, 2016; Bonakdarpour and Finkbeiner, 2018; Brett et al., 2017; Clarkson et al., 2014; Finkbeiner et al., 2019; Gutsfeld et al., 2021], allows us to focus on examining the intricacies of monitoring in a decentralized setting with monitor communication. More precisely, it allows us to fix the σ in the synthesis function which, in turn, produces a *static* set of locations with which a monitor can communicate. Despite the restriction to PHyper-recHML, our synthesis algorithm still covers properties that were previously not even expressible, hence not monitorable, in state-of-the-art hyperlogics.

Future Work. Of course, the picture is still incomplete: we have a centralized-monitor synthesis procedure for an expressive fragment of Hyper-recHML, whereas our decentralized-monitor synthesis deals with a more restricted fragment of that logic. It is not clear if this restriction is necessary; for example, a different decentralized-monitor synthesis for a larger fragment might be obtained by utilizing a different communication paradigm other than multicast, which was adopted in this study. In fact, we conjecture that broadcast communications might allow us to synthesize decentralized monitors for a larger Hyper-recHML fragment, including formulae that mix greatest-fixed-points and quantifiers, like φ_a defined in Equation (1); currently, monitors only send messages to the locations in the range of the specified σ.

A second interesting direction is to allow monitors to infer information from communications they did not receive. A good starting point to explore such a synthesis algorithm (and prove its correctness) can be the synthesis properties in Definition 4.8. To fully delineate the power of decentralized monitoring, a maximality result in the spirit of those presented in Aceto et al. [2019a, 2021] is needed, which we intend to establish in the future.

A third avenue for future investigation is to synthesize monitors that detect all satisfying hypertraces for the respective dual fragments of Hyper-recHML, instead of focusing on monitors that detect violations (that is, we can focus on satisfaction completeness instead of on violation completeness). A fourth possible direction is to provide an estimation (or, at least, a lower/upper bound) of the size of synthesized centralized and decentralized monitors for a given formula; this will require non-trivial combinatorial arguments, like those in Aceto et al. [2019b]. Another possible direction is to extend the proposed approach to hyperlogics that adopt asynchronous semantics, like the one adopted in Chalupa and Henzinger [2023] and Gutsfeld et al. [2021]. Furthermore, it would also be challenging to understand how we can model a scenario where the number of localities varies over time, and how we can handle the dynamic creation/deletion of the associated local monitors.

Finally, another relevant direction we intend to pursue in future is the development of tools for monitoring Hyper-recHML specifications at runtime, based on the results of this article. We expect that our decentralized-monitor synthesis procedure can be implemented by generating a dedicated monitor for every location in a way that is very similar to the synthesis of μHML monitors presented in Aceto et al. [2022b, 2024b] and Attard et al. [2021] and implemented in the tool `detectEr` available at https://duncanatt.github.io/detecter/.

Related Work. To the best of our knowledge, Agrawal and Bonakdarpour were the first to study RV for hyperproperties expressed in HyperLTL in Agrawal and Bonakdarpour [2016], where they investigated monitorability for k-safety hyperproperties expressed in HyperLTL. They also gave a semantic characterization of monitorable k-safety hyperproperties, which is a natural extension to hyperproperties of the 'universal version' of the classic definition of monitorability presented by Aceto et al. [2021] and Pnueli and Zaks [2006]. In contrast to this work, we do not restrict ourselves to alternation-free formulas (see Equation (1)) and every monitorable formula considered by Agrawal and Bonakdarpour can be expressed in our monitorable fragment. Brett et al. [2017] improve on the work presented in Agrawal and Bonakdarpour [2016] by presenting an algorithm for monitoring the full alternation-free fragment of HyperLTL. They also highlight challenges that arise when monitoring arbitrary HyperLTL formulas, namely (i) quantifier alternations, (ii) inter-trace dependencies and (iii) relative ordering of events across traces. Our decentralized-monitor synthesis addresses (i) by using the number of locations as an upper bound on the number of traces, and (ii) and (iii) via synchronized multicasts.

In Finkbeiner et al. [2019] investigate RV for HyperLTL [Clarkson et al., 2014] formulas w.r.t. three different input classes, namely the bounded sequential, the unbounded sequential and the parallel classes. They also develop the monitoring tool RVHyper [Finkbeiner et al., 2018] based on the sequential algorithms developed for those input classes. The parallel class is closest to our set-up, since it consists in a *fixed* number of system executions that are processed synchronously.

Beutner et al. [2024] study runtime monitoring for HYPER^{2}LTL$_{\text{fp}}$, a temporal logic that is interpreted over sets of *finite* traces of *equal length*. Unlike HYPER^{2}LTL [Beutner et al., 2023], HYPER^{2}LTL$_{\text{fp}}$ permits quantification under temporal operators, which is also allowed in our logic Hyper-recHML. In contrast to HyperLTL, HYPER^{2}LTL$_{\text{fp}}$ features second-order quantification over sets of finite traces and can express properties like common knowledge.

In Gutsfeld et al. [2021] study automated analysis techniques for asynchronous hyperproperties and propose a novel automata-theoretic framework, the so-called alternating asynchronous parity automata, together with the fixed-point logic H_μ for expressing asynchronous hyperproperties. The logic H_μ has commonalities with PHyper-recHML, but it only allows for prenex formulas; moreover, its semantics progresses asynchronously on each trace. Properties such as "an atomic proposition does not occur at a certain level in the tree (of traces)" are not expressible in their logic H_μ, but can be described in Hyper-recHML.

Chalupa and Henzinger [2023] explore the potential of monitoring for hyperproperties using prefix transducers. They develop a transducer language, called prefix expressions, give it an operational semantics over a hypertrace (reminiscent of the semantics in Section 4) and then implement it to assess the induced overheads. They show how transducers can use the writing capabilities as a method for monitor synchronization across traces, akin to the monitor communication and verdict aggregation of Section 4. Since transducers are, in principle, more powerful that passive monitors, additional guarantees are required to ensure that they do not interfere unnecessarily with system executions.

References

Luca Aceto, Antonis Achilleos, Elli Anastasiadi, and Adrian Francalanza. 2022a. Monitoring hyperproperties with circuits. In *42nd IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE '22)*. Mohammad Reza Mousavi and Anna Philippou (Eds.), LNCS, Vol. 13273, Springer, 1–10. DOI: https://doi.org/10.1007/978-3-031-08679-3_1

Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, Daniele Gorla, and Jana Wagemaker. 2024a. Centralized vs decentralized monitors for hyperproperties. In *35th International Conference on Concurrency Theory (CONCUR)*. LIPiCS, Vol. 311, Article 34, 1–19.

Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, Daniele Gorla, and Jana Wagemaker. 2025. Centralized vs decentralized monitors for hyperproperties. arXiv:2405.12882. Retrieved from https://arxiv.org/abs/2405.12882

Luca Aceto, Antonis Achilleos, Duncan Paul Attard, Léo Exibard, Adrian Francalanza, and Anna Ingólfsdóttir. 2022b. A monitoring tool for linear-time μHML. In *24th IFIP WG 6.1 International Conference on Coordination Models and Languages (COORDINATION)*. LNCS, Vol. 13271, Springer, 200–219.

Luca Aceto, Antonis Achilleos, Duncan Paul Attard, Léo Exibard, Adrian Francalanza, and Anna Ingólfsdóttir. 2024b. A monitoring tool for linear-time μHML. *Science of Computer Programming* 232 (2024), 103031. DOI: https://doi.org/10.1016/j.scico.2023.103031

Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. 2019a. Adventures in monitorability: From branching to linear time and back again. *Proceedings of the ACM on Programming Languages* 3, 52 (2019), 1–29. DOI: https://doi.org/10.1145/3290365

Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. 2019b. The cost of monitoring alone. In *From Reactive Systems to Cyber-Physical Systems—Essays Dedicated to Scott A. Smolka on the Occasion of His 65th Birthday*. LNCS, Vol. 11500, Springer, 259–275.

Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. 2019c. Testing equivalence vs. Runtime monitoring. In *Models, Languages, and Tools for Concurrent and Distributed Programming*. LNCS, Vol. 11665, Springer, 28–44.

Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. 2021. An operational guide to monitorability with applications to regular properties. *Software and Systems Modeling* 20, 2 (2021), 335–361.

Luca Aceto, Duncan Paul Attard, Adrian Francalanza, and Anna Ingólfsdóttir. 2024c. Runtime instrumentation for reactive components. In *38th European Conference on Object-Oriented Programming (ECOOP)*. LIPIcs, Vol. 313, Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Article 16, 1–33.

Luca Aceto, Ian Cassar, Adrian Francalanza, and Anna Ingólfsdóttir. 2023. On first-order runtime enforcement of branching-time properties. *Acta Informatica* 60, 4 (2023), 385–451.

Shreya Agrawal and Borzoo Bonakdarpour. 2016. Runtime verification of k-safety hyperproperties in HyperLTL. In *IEEE 29th Computer Security Foundations Symposium*. IEEE Computer Society, 239–252.

Bowen Alpern and Fred B. Schneider. 1985. Defining liveness. *Information Processing Letters* 21, 4 (1985), 181–185. DOI: https://doi.org/10.1016/0020-0190(85)90056-0

Duncan Paul Attard, Luca Aceto, Antonis Achilleos, Adrian Francalanza, Anna Ingólfsdóttir, and Karoliina Lehtinen. 2021. Better late than never or: Verifying asynchronous components at runtime. In *41st IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE)*. LNCS, Vol. 12719, Springer, 207–225.

Ezio Bartocci, Yliès Falcone, Adrian Francalanza, and Giles Reger. 2018. Introduction to runtime verification. In *Lectures on Runtime Verification—Introductory and Advanced Topics*. Ezio Bartocci and Yliès Falcone (Eds.), LNCS, Vol. 10457. Springer, 1–33. DOI: https://doi.org/10.1007/978-3-319-75632-5_1

Raven Beutner and Bernd Finkbeiner. 2022. Software verification of hyperproperties beyond k-safety. *In 34th International Conference on Computer Aided Verification (CAV '22)*. Sharon Shoham and Yakir Vizel (Eds.), LNCS, Vol. 13371, Springer, 341–362. DOI: https://doi.org/10.1007/978-3-031-13185-1_17

Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. 2023. Second-order hyperproperties. In *35th International Conference on Computer Aided Verification (CAV '23)*. LNCS, Vol. 13965, Springer, 309–332.

Raven Beutner, Bernd Finkbeiner, Hadar Frenkel, and Niklas Metzger. 2024. Monitoring second-order hyperproperties. In *23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.), ACM, 180–188. Retrieved from https://dl.acm.org/doi/10.5555/3635637.3662865

Laura Bocchi, Tzu-Chun Chen, Romain Demangeon, Kohei Honda, and Nobuko Yoshida. 2017. Monitoring networks through multiparty session types. *Theoretical Computer Science* 669 (2017), 33–58.

Laura Bocchi, Kohei Honda, Emilio Tuosto, and Nobuko Yoshida. 2010. A theory of design-by-contract for distributed multiparty interactions. In *CONCUR 2010—Concurrency Theory*. Paul Gastin and François Laroussinie (Eds.), Springer, Berlin, 162–176.

Borzoo Bonakdarpour and Bernd Finkbeiner. 2016. Runtime verification for HyperLTL. In *16th International Conference on Runtime Verification (RV '16)*. Yliès Falcone and César Sánchez (Eds.), LNCS, Vol. 10012, Springer, 41–45. DOI: https://doi.org/10.1007/978-3-319-46982-9_4

Borzoo Bonakdarpour and Bernd Finkbeiner. 2018. The complexity of monitoring hyperproperties. In *31st IEEE Computer Security Foundations Symposium (CSF '18)*. IEEE Computer Society, 162–174. DOI: https://doi.org/10.1109/CSF.2018.00019

Borzoo Bonakdarpour, Pierre Fraigniaud, Sergio Rajsbaum, David A. Rosenblueth, and Corentin Travers. 2022. Decentralized asynchronous crash-resilient runtime verification. *Journal of the ACM* 69, 5, Article 34 (2022), 1–31. DOI: https://doi.org/10.1145/3550483

Borzoo Bonakdarpour, Pierre Fraigniaud, Sergio Rajsbaum, and Corentin Travers. 2016. Challenges in fault-tolerant distributed runtime verification. In *7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA '16)*. Tiziana Margaria and Bernhard Steffen (Eds.), LNCS, Vol. 9953, 363–370. DOI: https://doi.org/10.1007/978-3-319-47169-3_27

Borzoo Bonakdarpour, César Sánchez, and Gerardo Schneider. 2018. Monitoring hyperproperties by combining static analysis and runtime verification. In *8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation. Verification (ISoLA '18)*. Tiziana Margaria and Bernhard Steffen (Eds.), LNCS, Vol. 11245, Springer, 8–27. DOI: https://doi.org/10.1007/978-3-030-03421-4_2

Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. 2015. Unifying hyper and epistemic temporal logics. In *18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '15)*. Andrew M. Pitts (Ed.), LNCS, Vol. 9034, Springer, 167–182. DOI: https://doi.org/10.1007/978-3-662-46678-0_11

Noel Brett, Umair Siddique, and Borzoo Bonakdarpour. 2017. Rewriting-based runtime verification for alternation-free HyperLTL. In *23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Axel Legay and Tiziana Margaria (Eds.), Springer, Berlin, 77–93.

Ian Cassar, Adrian Francalanza, Claudio Antares Mezzina, and Emilio Tuosto. 2017. Reliability and fault-tolerance by choreographic design. In *2nd International Workshop on Pre- and Post-Deployment Verification Techniques (PrePost@iFM '17)*. Adrian Francalanza and Gordon J. Pace (Eds.), EPTCS, Vol. 254, 69–80. DOI: https://doi.org/10.4204/EPTCS.254.6

Marek Chalupa and Thomas A. Henzinger. 2023. Monitoring hyperproperties with prefix transducers. In *23rd International Conference on Runtime Verification (RV)*. LNCS, Vol. 14245, Springer, 168–190.

Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. 2014. Temporal logics for hyperproperties. In *3rd International Conference on Principles of Security and Trust (POST '14)*. Martín Abadi and Steve Kremer (Eds.), LNCS, Vol. 8414, Springer, 265–284. DOI: https://doi.org/10.1007/978-3-642-54792-8_15

Michael R. Clarkson and Fred B. Schneider. 2010. Hyperproperties. *Journal of Computer Security* 18, 6 (2010), 1157–1210. DOI: https://doi.org/10.3233/JCS-2009-0393

E. Allen Emerson and Joseph Y. Halpern. 1986. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM* 33, 1, 151–178. DOI: https://doi.org/10.1145/4904.4999

Bernd Finkbeiner, Christopher Hahn, Marvin Stenger, and Leander Tentrup. 2018. RVHyper: A runtime verification tool for temporal hyperproperties. In *24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '18)*. LNCS, Vol. 10806, Springer, 194–200.

Bernd Finkbeiner, Christopher Hahn, Marvin Stenger, and Leander Tentrup. 2019. Monitoring hyperproperties. *Formal Methods in System Design* 54, 3 (2019), 336–363. DOI: https://doi.org/10.1007/s10703-019-00334-z

Bernd Finkbeiner and Martin Zimmermann. 2017. The first-order logic of hyperproperties. In *34th Symposium on Theoretical Aspects of Computer Science (STACS '17)*. Heribert Vollmer and Brigitte Vallée (Eds.), LIPIcs, Vol. 66, Schloss Dagstuhl—Leibniz-Zentrum für Informatik. Article 30, 1–14. DOI: https://doi.org/10.4230/LIPIcs.STACS.2017.30

Pierre Fraigniaud, Sergio Rajsbaum, and Corentin Travers. 2020. A lower bound on the number of opinions needed for fault-tolerant decentralized run-time monitoring. *Journal of Applied and Computational Topology* 4, 1 (2020), 141–179. DOI: https://doi.org/10.1007/s41468-019-00047-6

Adrian Francalanza. 2017. Consistently-detecting monitors. In *28th International Conference on Concurrency Theory (CONCUR)*. LIPIcs, Vol. 85, Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Article 8, 1–19.

Adrian Francalanza. 2021. A theory of monitors. *Information and Computation* 281 (2021), 104704. DOI: https://doi.org/10.1016/j.ic.2021.104704

Adrian Francalanza, Luca Aceto, and Anna Ingólfsdóttir. 2017. Monitorability for the Hennessy-Milner logic with recursion. *Formal Methods in System Design* 51, 1 (2017), 87–116. DOI: https://doi.org/10.1007/S10703-017-0273-Z

Adrian Francalanza, Andrew Gauci, and Gordon J. Pace. 2013. Distributed system contract monitoring. *The Journal of Logic and Algebraic Programming* 82, 5–7 (2013), 186–215.

Jens Oliver Gutsfeld, Markus Müller-Olm, and Christoph Ohrem. 2021. Automata and fixpoints for asynchronous hyperproperties. *Proceedings of the ACM on Programming Languages* 5, POPL, Article 38 (Jan. 2021), 29 pages. DOI: https://doi.org/10.1145/3434319

Christopher Hahn, Marvin Stenger, and Leander Tentrup. 2019. Constraint-based monitoring of hyperproperties. In *25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Tomáš Vojnar and Lijun Zhang (Eds.), Springer International Publishing, Cham, 115–131.

Jun Inoue and Yoriyuki Yamagata. 2017. Operational semantics of process monitors. In *17th International Conference on Runtime Verification (RV)*. LNCS, Vol. 10548, Springer, 403–409.

Limin Jia, Hannah Gommerstadt, and Frank Pfenning. 2016. Monitors and blame assignment for higher-order session types. In *43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*. ACM, 582–594.

Dexter Kozen. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science* 27, 3 (1983), 333–354. DOI: https://doi.org/10.1016/0304-3975(82)90125-6

Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. 2000. An automata-theoretic approach to branching-time model checking. *Journal of the ACM* 47, 2 (2000), 312–360. DOI: https://doi.org/10.1145/333979.333987

Ruggero Lanotte, Massimo Merro, and Andrei Munteanu. 2021. A process calculus approach to detection and mitigation of PLC malware. *Theoretical Computer Science* 890 (2021), 125–146.

Ruggero Lanotte, Massimo Merro, and Andrei Munteanu. 2023. Industrial control systems security via runtime enforcement. *ACM Transactions on Privacy and Security* 26, 1 (2023), 4:1–4:41.

Kim G. Larsen. 1990. Proof systems for satisfiability in Hennessy-Milner logic with recursion. *Theoretical Computer Science* 72, 2 (1990), 265–288. DOI: https://doi.org/10.1016/0304-3975(90)90038-J

Claudio Antares Mezzina and Jorge A. Pérez. 2017. Causally consistent reversible choreographies: A monitors-as-memories approach. In *19th International Symposium on Principles and Practice of Declarative Programming (PPDP '17)*. ACM, New York, NY, USA, 127–138. DOI: https://doi.org/10.1145/3131851.3131864

Iain Phillips. 1987. Refusal testing. *Theoretical Computer Science* 50 (1987), 241–284. DOI: https://doi.org/10.1016/0304-3975(87)90117-4

Amir Pnueli. 1977. The temporal logic of programs. In *18th IEEE Annual Symposium on Foundations of Computer Science (FOCS '77)*. IEEE, 46–57. DOI: https://doi.org/10.1109/SFCS.1977.32

Amir Pnueli and Aleksandr Zaks. 2006. PSL model checking and run-time verification via testers. In *International Symposium on Formal Methods (FM)*. LNCS, Vol. 4085, Springer, 573–586.

George M. Reed and A. W. Roscoe. 1999. The timed failures-stability model for CSP. *Theoretical Computer Science* 211, 1–2 (1999), 85–127. DOI: https://doi.org/10.1016/S0304-3975(98)00214-X

A. W. Roscoe. 1997. *The Theory and Practice of Concurrency*. Prentice Hall PTR, Upper Saddle River.

Moshe Y. Vardi. 1988. A temporal fixpoint calculus. In *Conference Record of the 15th Annual ACM Symposium on Principles of Programming Languages*. Jeanne Ferrante and Peter Mager (Eds.), ACM Press, 250–259. DOI: https://doi.org/10.1145/73560.73582

Pierre Wolper. 1983. Temporal logic can be more expressive. *Information and Control* 56, 1/2 (1983), 72–99. DOI: https://doi.org/10.1016/S0019-9958(83)80051-5

Appendices

A Proofs for Soundness of Centralized Monitors

We need a few preliminary lemmata to prove Theorem 3.5. We first show that centralized monitors never output the end verdict if they are end-free themselves:

LEMMA A.1. *If m is an* end*-free monitor and $m \Rrightarrow v$, then $v = \mathsf{no}$ or $v = \mathsf{yes}$.*

We link the semantics of a formula to the special case where the synthesized centralized monitor is a verdict, which we use in various subsequent proofs:

LEMMA A.2. *If $\mathsf{cm}_\sigma(\varphi) = \mathsf{yes}$, then $[[\varphi]]^\rho_\sigma = \mathsf{HTrc}_\mathcal{L}$, for all ρ; if $\mathsf{cm}_\sigma(\varphi) = \mathsf{no}$, then $[[\varphi]]^\rho_\sigma = \emptyset$, for all ρ.*

We show that if synthesized monitors of two formulas are equal, then these two formulas must have the same semantics, which is used to prove Corollary A.6.

LEMMA A.3. *Let φ and ψ be (possibly open) formulas. If $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma'}(\psi)$, then $[[\varphi]]^\rho_\sigma = [[\psi]]^\rho_{\sigma'}$, for all ρ.*

PROOF. We proceed by induction on φ. As a first base case we consider $\varphi = \mathsf{tt}$. We then have $\mathsf{cm}_\sigma(\varphi) = \mathsf{yes} = \mathsf{cm}_{\sigma'}(\psi)$. We can immediately conclude that, for all ρ, we have $[[\varphi]]^\rho_\sigma = [[\psi]]^\rho_{\sigma'}$ using Lemma A.2. The base cases for $\varphi = \mathsf{ff}$, $\varphi = (\pi = \pi')$ and $\varphi = (\pi \neq \pi')$ are similar.

The last base case is for $\varphi = x$ for x a recursion variable. Then $\mathsf{cm}_\sigma(\varphi) = x = \mathsf{cm}_{\sigma'}(\psi)$, from which we can conclude that $\psi = x$ or $\psi = Q_1 \ldots Q_n.\psi'$ for Q_i a universal or existential quantifier, $n \geq 1$, $\mathcal{L} = \{\ell\}$ and $\psi' = x$. In the first case it follows immediately that, for all ρ, we have $[[\varphi]]^\rho_\sigma = [[\psi]]^\rho_{\sigma'}$. In the second case we obtain that $[[\psi]]^\rho_\sigma = [[\psi']]^\rho_{\sigma[\pi\mapsto\ell]} = \rho(x) = [[\varphi]]$.

We now proceed with a selection of the inductive cases. We first consider $\varphi = [a_\pi]\varphi'$. Thus $\mathsf{cm}_\sigma(\varphi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\varphi') + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{yes} = \mathsf{cm}_{\sigma'}(\psi)$. We distinguish two cases: either $\psi = [a_{\pi'}]\psi'$, $\sigma(\pi) = \sigma'(\pi')$ and $\mathsf{cm}_\sigma(\varphi') = \mathsf{cm}_{\sigma'}(\psi')$, or $\mathsf{Act} = \{a, b\}$, $\psi = \langle b_{\pi'}\rangle\psi'$, $\sigma(\pi) = \sigma'(\pi')$, $\mathsf{cm}_\sigma(\varphi') = \mathsf{no}$ and $\mathsf{cm}_{\sigma'}(\psi') = \mathsf{yes}$. In the first case we use the induction hypothesis to obtain that, for all ρ, we have $[[\varphi']]^\rho_\sigma = [[\psi']]^\rho_{\sigma'}$. The definition of the semantics gives the desired result immediately. In the second case, we use Lemma A.2 to obtain that, for all ρ, we have $[[\varphi]]^\rho_\sigma = \{T \mid hd(T)(\sigma(\pi)) = a \text{ implies } tl(T) \in \emptyset\} = \{T \mid hd(T)(\sigma'(\pi')) = b \ \wedge\ tl(T) \in \mathsf{HTrc}_{\mathcal{L}}\} = [[\psi]]^\rho_{\sigma'}$.

For the case where $\varphi = \varphi_1 \wedge \varphi_2$ and the case where $\varphi = \forall\pi.\varphi'$, we need an intermediary result:

$$\text{If } \mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma_1}(\varphi_1) \otimes \cdots \otimes \mathsf{cm}_{\sigma_k}(\varphi_k) \text{ for k}\geq 2\text{, then } [[\varphi]]^\rho_\sigma = \bigcap_{i=1}^{k} [[\varphi_i]]^\rho_{\sigma_i}, \text{ for all } \rho. \tag{5}$$

We will prove this claim at the end, but first finish the main proof using this result. Consider $\varphi = \varphi_1 \wedge \varphi_2$. Hence, $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_\sigma(\varphi_1) \otimes \mathsf{cm}_\sigma(\varphi_2) = \mathsf{cm}_{\sigma'}(\psi)$. We use intermediary result (Equation (5)) to conclude. Now consider the case where $\varphi = \forall\pi.\varphi'$ and $|\mathcal{L}| = 1$: we get $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi') = \mathsf{cm}_{\sigma'}(\psi)$. From the induction hypothesis we obtain that, for all ρ, we have that $[[\varphi']]^\rho_{\sigma[\pi\mapsto\ell]} = [[\psi]]^\rho_{\sigma'}$. Since $\mathcal{L} = \{\ell\}$, we have, for all ρ, that $[[\varphi]]^\rho_\sigma = [[\varphi']]^\rho_{\sigma[\pi\mapsto\ell]}$, and we are done. If $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ for $k \geq 2$, we have that $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma[\pi\mapsto\ell_1]}(\varphi_1) \otimes \cdots \otimes \mathsf{cm}_{\sigma[\pi\mapsto\ell_k]}(\varphi_k) = \mathsf{cm}_{\sigma'}(\psi)$ and we use intermediary result (Equation (5)) to conclude.

Now we prove claim (Equation (5)) by induction on φ. Without loss of generality we can assume that for all $i \in \{1, \dots, k\}$ we have that $\mathsf{cm}_{\sigma_i}(\varphi_i) \neq m \otimes n$ for all m, n (we refer to this as maximality).

We distinguish the case where $k = 2$ or $k > 2$. If $k = 2$, we get from the synthesis function that φ is a conjunction or a universal formula. In case $\varphi = \varphi' \wedge \varphi''$, we obtain that $\mathsf{cm}_\sigma(\varphi') \otimes \mathsf{cm}_\sigma(\varphi'') = \mathsf{cm}_{\sigma_1}(\varphi_1) \otimes \mathsf{cm}_{\sigma_2}(\varphi_2)$. We use the induction hypothesis of the main proof to conclude that, for all ρ, we have $[[\varphi']]^\rho_\sigma = [[\varphi_1]]^\rho_{\sigma_1}$ and, for all ρ, we have $[[\varphi'']]^\rho_\sigma = [[\varphi_2]]^\rho_{\sigma_2}$. The desired result now follows immediately from the semantics:

$$[[\varphi]]^\rho_\sigma = [[\varphi']]^\rho_\sigma \cap [[\varphi'']]^\rho_\sigma \overset{IH}{=} [[\varphi_1]]^\rho_{\sigma_1} \cap [[\varphi_2]]^\rho_{\sigma_2}.$$

The case for $\varphi = \forall\pi.\varphi'$ and $k = 2$ implies that $\mathcal{L} = \{\ell_1, \ell_2\}$ and it is handled similarly. Then we treat the case where $k > 2$. From the synthesis function, we can again conclude that φ is a conjunction or a universal formula. The universal case is a complicated version of the conjunctive case, so we only threat the former. Let $\varphi = \forall\pi.\varphi'$ and $K = \{1, \dots, k\}$. Then, $\mathsf{cm}_\sigma(\varphi) = \otimes_{\ell\in\mathcal{L}}\mathsf{cm}_{\sigma[\psi\mapsto\ell]}(\varphi')$, and from our assumption of maximality of $\otimes_{i\in K}\mathsf{cm}_{\sigma_i}(\varphi_i)$, there is a partition $(K_\ell)_{\ell\in\mathcal{L}}$, such that for every $\ell \in \mathcal{L}$, $\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi') = \otimes_{i\in K_\ell}\mathsf{cm}_{\sigma_i}(\varphi_i)$. By the induction hypothesis, for all ρ we have that $[[\varphi']]^\rho_{\sigma[\pi\mapsto\ell]} = \bigcap_{i\in K_\ell}[[\varphi_i]]^\rho_{\sigma_i}$ and we can conclude that $[[\varphi]]^\rho_\sigma = \bigcap_{\ell\in\mathcal{L}}[[\varphi']]^\rho_{\sigma[\pi\mapsto\ell]} = \bigcap_{\ell\in\mathcal{L}}\bigcap_{i\in K_\ell}[[\varphi_i]]^\rho_{\sigma_i} = \bigcap_{i\in K}[[\varphi_i]]^\rho_{\sigma_i}$. □

The following immediate fact is used to prove in Lemma 3.2, which shows soundness of synthesized monitors when they have not taken any transition yet, and Lemma 3.4:

Corollary A.4. $[[\max x.\psi]]_\sigma = [[\psi\{^{\max x.\psi}/_x\}]]_\sigma$.

The next lemma shows that, for any combination of synthesized parallel monitors, we can construct an equal synthesized monitor, and link their semantics. It is used in various lemmas below.

Lemma A.5. *If $n = \mathsf{cm}_{\sigma_1}(\varphi_1) \otimes \cdots \otimes \mathsf{cm}_{\sigma_k}(\varphi_k)$ with $k \geq 1$, then there exist σ and ψ such that $n = \mathsf{cm}_\sigma(\psi)$ and $[[\psi]]_\sigma = \bigcap_{i\in I}[[\varphi_i]]_{\sigma_i}$. Similarly, if $n = \mathsf{cm}_{\sigma_1}(\varphi_1) \oplus \cdots \oplus \mathsf{cm}_{\sigma_k}(\varphi_k)$, there exist σ and ψ such that $n = \mathsf{cm}_\sigma(\psi)$ and $[[\psi]]_\sigma = \bigcup_{i\in I}[[\varphi_i]]_{\sigma_i}$.*

Proof. We only prove the case for $\otimes$, the case for $\oplus$ is similar. We proceed by induction on k. The base case is trivial. For the inductive case we consider $n = \mathsf{cm}_{\sigma_1}(\varphi_1) \otimes \mathsf{cm}_{\sigma_2}(\varphi_2) \otimes \cdots \otimes \mathsf{cm}_{\sigma_k}(\varphi_k)$, where we know from the induction hypothesis that there exist σ', ψ' such that $\mathsf{cm}_{\sigma_2}(\varphi_2) \otimes \cdots \otimes \mathsf{cm}_{\sigma_k}(\varphi_k) = \mathsf{cm}_{\sigma'}(\psi')$ and $[[\psi']]_{\sigma'} = \bigcap_{i\in\{2,\ldots,k\}} [[\varphi_i]]_{\sigma_i}$. Let $\pi_1, \ldots, \pi_j$ be all the variables occuring in φ_1. Let $\pi'_1, \ldots \pi'_j$ all be fresh variables not occuring in φ_1 or in ψ'. Lastly, let $\varphi'_1 = \varphi_1\{^{\pi'_1}/_{\pi_1}\} \ldots \{^{\pi'_j}/_{\pi_j}\}$.

Then $\mathsf{cm}_{\sigma_1}(\varphi_1) = \mathsf{cm}_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}(\varphi'_1)$ and $[[\varphi_1]]_{\sigma_1} = [[\varphi'_1]]_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}$. Furthermore $\mathsf{cm}_{\sigma'}(\psi') = \mathsf{cm}_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}(\psi')$ and $[[\psi']]_{\sigma'} = [[\psi']]_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}$. Thus we can conclude that $n = \mathsf{cm}_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}(\varphi'_1 \wedge \psi')$. We also have that:

$$\begin{aligned}
&[[\varphi'_1 \wedge \psi']]_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}\\
&= [[\varphi'_1]]_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]} \cap [[\psi']]_{\sigma'[\pi'_1\mapsto\sigma_1(\pi_1)]\ldots[\pi'_j\mapsto\sigma_1(\pi_j)]}\\
&= [[\varphi_1]]_{\sigma_1} \cap [[\psi']]_{\sigma'}\\
&= [[\varphi_1]]_{\sigma_1} \cap \bigcap_{i\in\{2,\ldots,k\}} [[\varphi_i]]_{\sigma_i} = \bigcap_{i\in I} [[\varphi_i]]_{\sigma_i}. \quad \square
\end{aligned}$$

Using the definition of the synthesis function, Lemmas A.3 and A.5, we obtain that, for any synthesized monitor that is a combination of synthesized parallel monitors, we can relate the semantics of the corresponding formulas:

Corollary A.6. *For all φ and σ, if $\mathsf{cm}_\sigma(\varphi) = m \odot n$, and there exist ψ_1, ψ_2 and σ_1, σ_2 such that $m = \mathsf{cm}_{\sigma_1}(\psi_1)$ and $n = \mathsf{cm}_{\sigma_2}(\psi_2)$, then $[[\varphi]]_\sigma = [[\psi_1]]_{\sigma_1} \cap [[\psi_2]]_{\sigma_2}$ if $\odot = \otimes$, and $[[\varphi]]_\sigma = [[\psi_1]]_{\sigma_1} \cup [[\psi_2]]_{\sigma_2}$ if $\odot = \oplus$.*

Now we can prove soundness of synthesized monitors when they have not taken any transition yet:

Lemma 3.2. *If $\mathsf{cm}_\sigma(\varphi) \Rrightarrow v$, then $[[\varphi]]_\sigma = \mathsf{HTrc}_{\mathcal{L}}$, if $v = \mathsf{yes}$, and $[[\varphi]]_\sigma = \emptyset$, if $v = \mathsf{no}$.*

Proof. We proceed by induction on $\Rrightarrow$, focusing on the case $v = \mathsf{yes}$. The proof for the case $v = \mathsf{no}$ follows similar lines.

The base is for $\mathsf{cm}_\sigma(\varphi) = v$: here the result immediately follows from Lemmas A.1 and A.2.

For the first inductive case, we consider $\mathsf{cm}_\sigma(\varphi) = m \oplus n \Rrightarrow \mathsf{yes}$ and $m \Rrightarrow \mathsf{yes}$. From the synthesis function and possibly Lemma A.5 we conclude that there exist φ_1, φ_2 and σ_1, σ_2 such that $m = \mathsf{cm}_{\sigma_1}(\varphi_1)$ and $n = \mathsf{cm}_{\sigma_2}(\varphi_2)$. We apply the induction hypothesis to obtain that $[[\varphi_1]]_{\sigma_1} = \mathsf{HTrc}_{\mathcal{L}}$. The result then follows from Corollary A.6.

The case for end can be skipped, as no formula is synthesized into monitors with end.

Let $\mathsf{cm}_\sigma(\varphi) = m \otimes n \Rrightarrow \mathsf{yes}$ because $m \Rrightarrow \mathsf{yes}$ and $n \Rrightarrow \mathsf{yes}$. From the synthesis function and possibly Lemma A.5 we conclude that there exist φ_1, φ_2 and σ_1, σ_2 such that $m = \mathsf{cm}_{\sigma_1}(\varphi_1)$ and $n = \mathsf{cm}_{\sigma_2}(\varphi_2)$. We can apply the induction hypothesis to obtain that $[[\varphi_1]]_{\sigma_1} = \mathsf{HTrc}_{\mathcal{L}}$ and $[[\varphi_1]]_{\sigma_1} = \mathsf{HTrc}_{\mathcal{L}}$. The result then follows from Corollary A.6.

The case for $\mathsf{cm}_\sigma(\varphi) = m \oplus n \Rrightarrow \mathsf{yes}$ because $m \Rrightarrow \mathsf{no}$ and $n \Rrightarrow v$ is similar to the previous case.

Let $\mathsf{cm}_\sigma(\varphi) = m + n \Rrightarrow \mathsf{yes}$ because $m \Rrightarrow \mathsf{yes}$. This case can be skipped, because, by the definition of the synthesis function, we know that neither $m \Rrightarrow \mathsf{yes}$ nor $n \Rrightarrow \mathsf{yes}$ can hold whenever $\mathsf{cm}_\sigma(\varphi) = m + n$.

For the last inductive case, we consider $\mathsf{cm}_\sigma(\varphi) = \mathsf{rec}\ x.m \Rrightarrow \mathsf{yes}$ because $m\{^{\mathsf{rec}\ x.m}/_x\} \Rrightarrow \mathsf{yes}$. By definition of the synthesis, we know that $\varphi = \max x.\psi$ and $m = \mathsf{cm}_\sigma(\psi)$. Note that

$\mathsf{cm}_\sigma(\psi)\{^{\mathsf{rec}\ x.\mathsf{cm}_\sigma(\psi)}/_x\} = \mathsf{cm}_\sigma(\psi\{^{\max x.\psi}/_x\})$. From the induction hypothesis and Corollary A.4, we obtain that $[[\varphi]]_\sigma = [[\psi\{^{\max x.\psi}/_x\}]]_\sigma = \mathsf{HTrc}_\mathcal{L}$. □

The following lemma is a basic fact regarding substitution, and we use it in various places throughout the article.

Lemma A.7. *If* $\mathsf{cm}_\sigma(\varphi) = m$, $\mathsf{cm}_\sigma(\psi) = n$, *and the free location variables of* ψ *are not bound in* φ, *then* $\mathsf{cm}_\sigma(\varphi\{^\psi/_x\}) = m\{^n/_x\}$.

Proof. The proof is by induction on φ and the only interesting case is the one where $\varphi = \exists\pi.\varphi'$ or $\varphi = \forall\pi.\varphi'$. We describe the case for $\varphi = \exists\pi.\varphi'$, as the other one is dual.

In this case, $m = \mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_\sigma(\exists\pi.\varphi') = \bigoplus_{\ell\in\mathcal{L}} \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi')$. By the induction hypothesis, $\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi'\{^\psi/_x\}) = \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi')\{^{\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\psi)}/_x\}$. Since π does not appear free in ψ, $\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\psi) = \mathsf{cm}_\sigma(\psi) = n$. Therefore, $\mathsf{cm}_\sigma(\varphi\{^\psi/_x\}) = \bigoplus_{\ell\in\mathcal{L}} \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi'\{^\psi/_x\}) = \bigoplus_{\ell\in\mathcal{L}} \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi')\{^n/_x\} = m\{^n/_x\}$. □

The next lemma shows that synthesized monitors always transition to other synthesized monitors:

Lemma 3.3. *If* $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} m'$, *then* $m' = \mathsf{cm}'_\sigma(\varphi')$ *for some* $'_\sigma$ *and* φ'.

Proof. We proceed by induction on $\xrightarrow{A}$. The interesting cases are the inductive cases. We first treat both inductive cases of the sum. If $\mathsf{cm}_\sigma(\varphi) = m + n \xrightarrow{A} m'$, we have two cases: either $\varphi = [a_\pi]\psi$ or $\varphi = \langle a_\pi\rangle\psi$. We treat the case for $\varphi = [a_\pi]\psi$, the other case is similar. Thus we are in the case where $\mathsf{cm}_\sigma(\varphi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\psi) + \sum_{b\neq a} b_{\sigma(\pi)}.\mathsf{yes}$. This means that m' can be $\mathsf{cm}_\sigma(\psi)$ or yes. In both cases we have obtained the required result.

If $\mathsf{cm}_\sigma(\varphi) = \mathsf{rec}\ x.m \xrightarrow{A} m'$ because $m\{^{\mathsf{rec}\ x.m}/_x\} \xrightarrow{A} m'$, we know that $m = \mathsf{cm}_\sigma(\psi)$ and $\varphi = \max x.\psi$. We use Lemma A.7 to conclude that $m\{^{\mathsf{rec}\ x.m}/_x\} = \mathsf{cm}_\sigma(\psi\{^\varphi/_x\})$ (note that the free location variables of φ are not bound in ψ). Thus we can use the induction hypothesis to know that $m' = \mathsf{cm}_{\sigma'}(\varphi')$ for some σ' and formula φ'.

If $\mathsf{cm}_\sigma(\varphi) = m \odot n \xrightarrow{A} m' \odot n'$, from the synthesis function and possibly Lemma A.5 we obtain σ_1, σ_2 and ψ_1, ψ_2 such that $m = \mathsf{cm}_{\sigma_1}(\psi_1)$ and $n = \mathsf{cm}_{\sigma_2}(\psi_2)$. We then use the induction hypothesis to obtain σ'_1, σ'_2 and ψ'_1, ψ'_2 such that $m' = \mathsf{cm}_{\sigma'_1}(\psi'_1)$ and $n' = \mathsf{cm}_{\sigma'_2}(\psi'_2)$. The required result now follows from Lemma A.5. □

The last result we need to prove soundness is the following, which essentially shows that the relation between the hypertrace and the semantics of the formula are carried over through the instrumentation steps:

Lemma 3.4 *Let* $\mathsf{cm}_\sigma(\varphi) \triangleright T \rightarrowtail \mathsf{cm}_{\sigma'}(\varphi') \triangleright T'$. *If* $T' \notin [[\varphi']]_{\sigma'}$ *then* $T \notin [[\varphi]]_\sigma$; *if* $T' \in [[\varphi']]_{\sigma'}$ *then* $T \in [[\varphi]]_\sigma$.

Proof. From $\mathsf{cm}_\sigma(\varphi) \triangleright T \rightarrowtail \mathsf{cm}_{\sigma'}(\varphi') \triangleright T'$ we deduce that $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$ and $T \xrightarrow{A} T'$. We proceed by induction on $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$.

In the first base case we have $\mathsf{cm}_\sigma(\varphi) = v = \mathsf{cm}_{\sigma'}(\varphi')$. The result follows immediately from Lemma A.2.

The cases where $\mathsf{cm}_\sigma(\varphi) = a_\ell.m$ are not applicable, as the synthesis function cannot produce $a_\ell.m$ (being $|\mathsf{Act}| > 1$).

The first inductive case is for $\mathsf{cm}_\sigma(\varphi) = m + n$ and $\mathsf{cm}_{\sigma'}(\varphi') = m'$ or $\mathsf{cm}_{\sigma'}(\varphi') = n'$. We distinguish two cases.

(1) $\varphi = [a_\pi]\psi$. Here we have two more cases:
 (a) $m' = \mathsf{cm}_\sigma(\psi)$, $m = a_{\sigma(\pi)}.m'$, and $A(\sigma(\pi)) = a$. Assume that $T' \notin [\![\psi]\!]_\sigma$. Then $T \notin [\![[a_\pi]\psi]\!]_\sigma$ follows immediately from the definitions. If we assume instead that $T' \in [\![\psi]\!]_\sigma$, we also know immediately that $T \in [\![[a_\pi]\psi]\!]_\sigma$.
 (b) $m' = \mathsf{yes}$, $m = b_{\sigma(\pi)}.m'$ for $b \neq a$, and $A(\sigma(\pi)) \neq a$. It follows immediately that $T \in [\![[a_\pi]\psi]\!]_\sigma$ as $A(\sigma(\pi)) \neq a$, and that $T' \notin [\![\varphi']\!]_{\sigma'}$ never holds.

(2) $\varphi = \langle a_\pi\rangle\psi$. Also here we have two more cases:
 (a) $m' = \mathsf{cm}_\sigma(\psi)$, $m = a_{\sigma(\pi)}.m'$, and $A(\sigma(\pi)) = a$. Assume that $T' \notin [\![\psi]\!]_\sigma$. Then $T \notin [\![\langle a_\pi\rangle\psi]\!]_\sigma$ follows immediately from the definitions. If we assume instead that $T' \in [\![\psi]\!]_\sigma$, we also know immediately that $T \in [\![\langle a_\pi\rangle\psi]\!]_\sigma$.
 (b) $m' = \mathsf{no}$, $m = b_{\sigma(\pi)}.m'$ for $b \neq a$, and $A(\sigma(\pi)) \neq a$. It follows immediately that $T \notin [\![\langle a_\pi\rangle\psi]\!]_\sigma$ as $A(\sigma(\pi)) \neq a$, and that $T' \in [\![\varphi']\!]_{\sigma'}$ never holds.

Consider $\mathsf{cm}_\sigma(\varphi) = \mathsf{rec}\ x.m \xrightarrow{A} m'$. Then we have $\varphi = \max x.\psi$ and $m = \mathsf{cm}_\sigma(\psi)$. We know $\mathsf{cm}_{\sigma'}(\varphi') = m'$, and $\mathsf{cm}_\sigma(\psi)\{^{\mathsf{rec}\ x.\mathsf{cm}_\sigma(\psi)}/_x\} \xrightarrow{A} m'$. Assume that $T' \in [\![\varphi']\!]_{\sigma'}$. Note that $\mathsf{cm}_\sigma(\psi)\{^{\mathsf{rec}\ x.\mathsf{cm}_\sigma(\psi)}/_x\} = \mathsf{cm}_\sigma(\psi\{^{\max x.\psi}/_x\})$. From the induction hypothesis and Corollary A.4, we obtain that $T \in [\![\psi\{^{\max x.\psi}/_x\}]\!]_\sigma = [\![\varphi]\!]_\sigma$. The case for $T' \notin [\![\varphi']\!]_{\sigma'}$ is similar.

Last we consider $\mathsf{cm}_\sigma(\varphi) = m \odot n \xrightarrow{A} m' \odot n'$, because $m \xrightarrow{A} m'$, $n \xrightarrow{A} n'$, and $m' \odot n' = \mathsf{cm}_{\sigma'}(\varphi')$. From Lemma A.5, we know that there exist ψ_1,ψ_2 and τ_1, τ_2 such that $m' = \mathsf{cm}_{\tau_1}(\psi_1)$ and $n' = \mathsf{cm}_{\tau_2}(\psi_2)$. Similarly, we obtain also that $m = \mathsf{cm}_{\sigma_1}(\varphi_1)$ and $n = \mathsf{cm}_{\sigma_2}(\varphi_2)$ for some φ_1, φ_2 and σ_1, σ_2. We distinguish two cases, depending on whether $\odot = \otimes$ or $\odot = \oplus$. We limit ourselves to presenting the proof for the former case since the case $\odot = \oplus$ is similar. We assume $T' \in [\![\varphi']\!]_{\sigma'}$. Then by Corollary A.6 we know that $T' \in [\![\psi_1]\!]_{\tau_1}$ and $T' \in [\![\psi_2]\!]_{\tau_2}$. From the induction hypothesis we get that $T \in [\![\varphi_1]\!]_{\sigma_1}$ and $T \in [\![\varphi_2]\!]_{\sigma_2}$. By Corollary A.6 we know that $T \in [\![\varphi]\!]_\sigma$.

If, on the other hand, we assume $T' \notin [\![\varphi']\!]_{\sigma'}$, the argument is dual. □

B Proofs for Correct Synthesis of Decentralized Monitors

We now prove that the synthesis defined in Table 11 satisfies the properties in Definition 4.8 by proving one property in each of the following subsections. We start with some preliminary definitions and results that will be used for proving different properties.

We start with a lemma that is the equivalent of Lemma A.7 for $\mathsf{dm}^\ell_\sigma(-)$; the proof is identical. Then, we show that $\mathsf{dM}_\sigma(\varphi)$ cannot communicate.

Lemma B.1. *If $\mathsf{dm}^\ell_\sigma(\psi_1) = m$, $\mathsf{dm}^\ell_\sigma(\psi_2) = n$, and the free location variables of ψ_2 are not bound in ψ_1, then $\mathsf{dm}^\ell_\sigma(\psi_1\{^{\psi_2}/_x\}) = m\{^n/_x\}$.*

Lemma B.2. *Let $\mathsf{dM}_\sigma(\varphi)$ be defined; then, for all $[m]_\ell \in \mathsf{dM}_\sigma(\varphi)$ and $c \in \{(!G, \gamma), (?\ell, \gamma) \mid G \subseteq \mathcal{L}, \ell \in \mathcal{L}, \gamma \in \mathsf{Con}\}$, we have that $m \overset{c}{\not\to}$.*

Proof. The statement follows from a straightforward induction on φ and the following claim: for all ℓ, ψ, σ and $c \in \{(!G, \gamma), (?\ell, \gamma) \mid G \subseteq \mathcal{L}, \ell \in \mathcal{L}, \gamma \in \mathsf{Con}\}$, it holds that $\mathsf{dm}^\ell_\sigma(\psi) \overset{c}{\not\to}$. Such claim can be proved by induction on $\xrightarrow{c}$ to show that, if $m \xrightarrow{c} n$, then $m \neq \mathsf{dm}^\ell_\sigma(\psi)$ for all σ, ψ, ℓ. □

As most results in this work concern monitors that are reached in a run from a synthesized decentralized monitor, it is convenient to capture this in a definition.

Definition B.3. We say that a monitor M is *A-derived* if, for all $[m]_\ell \in M$, there exist ψ, σ, m' such that $\mathsf{dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m' \twoheadrightarrow m$. A monitor M is *action-derived* if there exists an A such that M is

A-derived. We call a local monitor m action-derived/A-derived for ℓ if $[m]_\ell$ is action-derived/A-derived.

We define a syntactic characterization of the kinds of monitors that are action-derived. This provides us with a convenient inductive proof principle.

Definition B.4. The class of *relevant local monitors* is defined for $\psi, \psi_a \in \mathsf{Qf}$ by the grammar:

$$m ::= \textsc{dm}^\ell_\sigma(\psi) \quad \mid (!G, a).\textsc{dm}^\ell_\sigma(\psi) \quad \mid \sum_{a \in \mathsf{Act}} (?\{\ell'\}, a).\textsc{dm}^\ell_\sigma(\psi_a) \mid \quad m \otimes m \mid \quad m \oplus m.$$

The class of relevant communicating monitors is then defined with the following grammar:

$$M ::= [m]_\ell \mid M \wedge M \mid M \vee M,$$

where m is a relevant local monitor and $\ell \in \mathcal{L}$. When it is clear from the context if a monitor is a relevant local monitor or a relevant communicating monitor, we will simply call it a relevant monitor, or say that the monitor is relevant.

The classes of relevant communicating/local monitors are closed under transitions:

Lemma B.5. *Let M and m be a relevant communicating monitor and a relevant local monitor, respectively. If $M \xrightarrow{\lambda} M'$ and $m \xrightarrow{\lambda} m'$, then M' and m' are a relevant communicating monitor and a relevant local monitor, respectively.*

Proof. The proof for $M \xrightarrow{\lambda} M'$ follows directly from the result for $m \xrightarrow{\lambda} m'$; this can be proved by induction on $\xrightarrow{\lambda}$ and the only interesting case is for $m = \textsc{dm}^\ell_\sigma(\psi)$. By Lemma B.2, it must be that $\lambda = a \in \mathsf{Act}$. The only base cases are $\textsc{dm}^\ell_\sigma(\psi) = a.m'$, but this cannot occur because of definition of the synthesis function and our assumption that $|\mathsf{Act}| \geq 2$, or $\textsc{dm}^\ell_\sigma(\psi) = m' = v$, that is trivial.

For the inductive case, when $\textsc{dm}^\ell_\sigma(\psi) = n_1 + n_2$ and $n_1 \xrightarrow{a} m'$, ψ must be of the form $[a'_\pi]\psi'$ or $\langle a'_\pi \rangle \psi'$; so:

$$\textsc{dm}^\ell_\sigma(\psi) = a'.(!\{\sigma(\pi') \mid \sigma(\pi') \neq \ell\}, a').\textsc{dm}^\ell_\sigma(\psi') + \sum_{b \neq a'} b.(!\{\sigma(\pi') \mid \sigma(\pi') \neq \ell\}, b).v,$$

or:

$$\textsc{dm}^\ell_\sigma(\psi) = \sum_{b \in \mathsf{Act}} b.\big((?\{\sigma(\pi)\}, a').\textsc{dm}^\ell_\sigma(\psi') + \sum_{b \neq a'} (?\{\sigma(\pi)\}, b).v\big),$$

where v is yes, if ψ is a box, and is no, if ψ is a diamond. In all cases, it is immediate that m' is a relevant monitor, regardless of whether $a = a'$ or not.

Next we consider the case for $\textsc{dm}^\ell_\sigma(\psi) = m = m_1 \odot m_2$ and $m' = m'_1 \odot m'_2$. Hence, $\psi = \psi_1 \diamond \psi_2$ (for $\diamond = \wedge$ if $\odot = \otimes$ and $\diamond = \vee$ if $\odot = \oplus$), $m_1 = \textsc{dm}^\ell_\sigma(\psi_1) \xrightarrow{a} m'_1$ and $m_2 = \textsc{dm}^\ell_\sigma(\psi_2) \xrightarrow{a} m'_2$. From the induction hypothesis it follows that m'_1 and m'_2 are relevant monitors, thus making m' relevant as well.

Last we consider $\textsc{dm}^\ell_\sigma(\psi) = \mathsf{rec}\ x.n \xrightarrow{a} m'$ because $n\{^{\mathsf{rec}\ x.n}/_x\} \xrightarrow{a} m'$. Thus we obtain that $\psi = \max x.\psi'$ and $n = \textsc{dm}^\ell_\sigma(\psi')$. As $\textsc{dm}^\ell_\sigma(\psi')\{^{\mathsf{rec}\ x.\textsc{dm}^\ell_\sigma(\psi')}/_x\} = \textsc{dm}^\ell_\sigma(\psi'\{^{\max x.\psi'}/_x\})$ (Lemma B.1), we use the induction hypothesis to obtain that m' is relevant. □

Relevant monitors can be used to prove properties of action-derived monitors, because of the following:

Lemma B.6. *Every action-derived monitor M is relevant.*

B.1 Proofs for Verdict Agreement of $\text{DM}_{\_}(\cdot)$

To relate verdicts of centralized and decentralized monitors, we need the following intermediary result:

Lemma B.7. *Let $\psi \in \mathsf{Qf}$ and $\varphi \in$ PHyper-*rec*HML.*

(1) *If there exists $\ell \in \mathcal{L}$ for which $\text{Dm}^{\ell}_{\sigma}(\psi) = v$, then $\text{Cm}_{\sigma}(\psi) = v$.*
(2) *If $\text{Cm}_{\sigma}(\psi) = v$, then for all $\ell \in \mathcal{L}$ we have $\text{Dm}^{\ell}_{\sigma}(\psi) = v$.*

We first relate the verdicts of centralized and decentralized synthesized monitors for ψ-formulas.

Lemma B.8. *Let σ be such that $\mathsf{fv}_{\mathsf{I}}(\psi) \subseteq \mathsf{dom}(\sigma)$. If there exists $\ell \in \mathcal{L}$ such that $\text{Dm}^{\ell}_{\sigma}(\psi) \Rrightarrow v$, then $\text{Cm}_{\sigma}(\psi) \Rrightarrow v$. Conversely, if $\text{Cm}_{\sigma}(\psi) \Rrightarrow v$, then $\text{Dm}^{\ell}_{\sigma}(\psi) \Rrightarrow v$, for all $\ell \in \mathcal{L}$.*

Proof. First, observe that ψ is quantifier-free (by definition) and that the definition of $\text{cm}_{\sigma}(\psi)$ is the same as the definition of $\text{Dm}^{\ell}_{\sigma}(\psi)$ except for the box and diamond modalities. Hence, we only prove the first statement by induction on $\Rrightarrow$, as the other proof is identical.

For the base case, we can skip the case for end (as a synthesized monitor never takes that shape) and so the only possibility is $\text{Dm}^{\ell}_{\sigma}(\psi) = v \Rrightarrow v$: the result follows immediately from Lemma B.7.

For the inductive case, we can skip the case for +: if $\text{Dm}^{\ell}_{\sigma}(\psi) = m + n$, then either $\psi = [a_{\pi}]\psi'$ or $\psi = \langle a_{\pi}\rangle\psi'$; from the synthesis function, it then follows that $\text{Dm}^{\ell}_{\sigma}(\psi)$ cannot evaluate to a verdict.

If $\text{Dm}^{\ell}_{\sigma}(\psi) = m \otimes n \Rrightarrow \mathsf{no}$ because $m \Rrightarrow \mathsf{no}$, then $\psi = \psi_1 \wedge \psi_2$ and $\text{Dm}^{\ell}_{\sigma}(\psi_1) = m$. We use the induction hypothesis to derive that $\text{cm}_{\sigma}(\psi_1) \Rrightarrow \mathsf{no}$ and thus $\text{cm}_{\sigma}(\psi) \Rrightarrow v$. The case for $\text{Dm}^{\ell}_{\sigma}(\psi) = m \oplus n \Rrightarrow \mathsf{yes}$ because $m \Rrightarrow \mathsf{yes}$ is dual.

Next we consider $\text{Dm}^{\ell}_{\sigma}(\psi) = m \otimes n \Rrightarrow v$ because $m \Rrightarrow \mathsf{yes}$ and $n \Rrightarrow v$. Thus $\psi = \psi_1 \wedge \psi_2$, $\text{Dm}^{\ell}_{\sigma}(\psi_1) = m$ and $\text{Dm}^{\ell}_{\sigma}(\psi_2) = n$. We use the induction hypothesis to derive that $\text{cm}_{\sigma}(\psi_1) \Rrightarrow \mathsf{yes}$ and $\text{cm}_{\sigma}(\psi_2) \Rrightarrow v$, which concludes the proof. The case for $\text{Dm}^{\ell}_{\sigma}(\psi) = m \oplus n \Rrightarrow v$ because $m \Rrightarrow \mathsf{no}$ and $n \Rrightarrow v$ is dual.

Finally, if $\text{Dm}^{\ell}_{\sigma}(\psi) = \mathsf{rec}\ x.m \Rrightarrow v$ because $m\{^{\mathsf{rec}\ x.m}/_{x}\} \Rrightarrow v$, then $\psi = \max x.\psi'$ and $m = \text{Dm}^{\ell}_{\sigma}(\psi')$. As $\text{Dm}^{\ell}_{\sigma}(\psi')\{^{\mathsf{rec}\ x.\text{Dm}^{\ell}_{\sigma}(\psi')}/_{x}\} = \text{Dm}^{\ell}_{\sigma}(\psi'\{^{\max x.\psi'}/_{x}\})$, from Lemma B.1, we get that $\text{Dm}^{\ell}_{\sigma}(\psi'\{^{\max x.\psi'}/_{x}\}) \Rrightarrow v$. By the induction hypothesis, we obtain that $\text{cm}_{\sigma}(\psi'\{^{\max x.\psi'}/_{x}\}) \Rrightarrow v$. This, together with $\text{cm}_{\sigma}(\psi')\{^{\mathsf{rec}\ x.\text{cm}_{\sigma}(\psi')}/_{x}\} = \text{cm}_{\sigma}(\psi'\{^{\max x.\psi'}/_{x}\})$ (that holds by Lemma A.7), implies that $\text{cm}_{\sigma}(\psi')\{^{\mathsf{rec}\ x.\text{cm}_{\sigma}(\psi')}/_{x}\} \Rrightarrow v$. The latter gives that $\text{cm}_{\sigma}(\psi) = \mathsf{rec}\ x.\text{cm}_{\sigma}(\psi') \Rrightarrow v$. □

The next lemma states a few basic facts about verdict evaluation in a centralized and decentralized setting; the proof is immediate. Notationally, when $I = \emptyset$, we let $\bigvee_{i \in I} m_i = \mathsf{no} = \bigoplus_{i \in I} m_i$ and $\bigwedge_{i \in I} m_i = \mathsf{yes} = \bigotimes_{i \in I} m_i$.

Lemma B.9.

(1) $\bigoplus_{i \in I} m_i \Rrightarrow \mathsf{yes} \quad \Leftrightarrow \quad \exists i \in I.m_i \Rrightarrow \mathsf{yes} \quad \Leftrightarrow \quad \bigvee_{i \in I} m_i \Rrightarrow \mathsf{yes}$;
(2) $\bigoplus_{i \in I} m_i \Rrightarrow \mathsf{no} \quad \Leftrightarrow \quad \forall i \in I.m_i \Rrightarrow \mathsf{no} \quad \Leftrightarrow \quad \bigvee_{i \in I} m_i \Rrightarrow \mathsf{no}$;
(3) $\bigotimes_{i \in I} m_i \Rrightarrow \mathsf{yes} \quad \Leftrightarrow \quad \forall i \in I.m_i \Rrightarrow \mathsf{yes} \quad \Leftrightarrow \quad \bigwedge_{i \in I} m_i \Rrightarrow \mathsf{yes}$;
(4) $\bigotimes_{i \in I} m_i \Rrightarrow \mathsf{no} \quad \Leftrightarrow \quad \exists i \in I.m_i \Rrightarrow \mathsf{no} \quad \Leftrightarrow \quad \bigwedge_{i \in I} m_i \Rrightarrow \mathsf{no}$.

Now we can relate verdicts for monitors synthesized from any formula in the centralized and decentralized setting, thus proving Verdict Agreement.

Lemma B.10. *Let $\varphi \in$PHyper-maxHML and σ be such that $\mathsf{fv}_{\mathsf{I}}(\varphi) \subseteq \mathsf{dom}(\sigma)$. Then, $\text{DM}_{\sigma}(\varphi) \Rrightarrow v \Leftrightarrow \text{CM}_{\sigma}(\varphi) \Rrightarrow v$.*

Proof. The proof proceeds by induction on φ. From the synthesis functions we know that $v \neq \mathsf{end}$. For the base case we consider $\varphi = \psi$. We distinguish when $\sigma = \emptyset$ or not. If $\sigma = \emptyset$, by assumption we know that $\mathsf{fv}_\mathsf{l}(\psi) = \emptyset$ and so ψ is trivial; therefore, $\mathsf{cm}_\sigma(\psi) = v \Rrightarrow v$ and $\mathrm{DM}_\sigma(\psi) = [v]_{\ell_0} \Rrightarrow v$. Let us now consider the case for $\sigma \neq \emptyset$. If $v = \mathsf{no}$, we derive:

$$\begin{aligned} \mathrm{DM}_\sigma(\psi) &= \bigvee_{\ell \in \mathsf{rng}(\sigma)} [\mathrm{Dm}^\ell_\sigma(\psi)]_\ell \Rrightarrow \mathsf{no} \\ &\Leftrightarrow \forall \ell \in \mathsf{rng}(\sigma).[\mathrm{Dm}^\ell_\sigma(\psi)]_\ell \Rrightarrow \mathsf{no} && \text{(Lemma B.9)} \\ &\Leftrightarrow \forall \ell \in \mathsf{rng}(\sigma).\mathrm{Dm}^\ell_\sigma(\psi) \Rrightarrow \mathsf{no} \\ &\Leftrightarrow \mathsf{cm}_\sigma(\psi) \Rrightarrow \mathsf{no}. && \text{(Lemma B.8)} \end{aligned}$$

If $v = \mathsf{yes}$, we have:

$$\begin{aligned} \mathrm{DM}_\sigma(\psi) &= \bigvee_{\ell \in \mathsf{rng}(\sigma)} [\mathrm{Dm}^\ell_\sigma(\psi)]_\ell \Rrightarrow \mathsf{yes} \\ &\Leftrightarrow \exists \ell \in \mathsf{rng}(\sigma).[\mathrm{Dm}^\ell_\sigma(\psi)]_\ell \Rrightarrow \mathsf{yes} && \text{(Lemma B.9)} \\ &\Leftrightarrow \exists \ell \in \mathsf{rng}(\sigma).\mathrm{Dm}^\ell_\sigma(\psi) \Rrightarrow \mathsf{yes} \\ &\Leftrightarrow \mathsf{cm}_\sigma(\psi) \Rrightarrow \mathsf{yes}. && \text{(Lemma B.8)} \end{aligned}$$

For the inductive step, we first consider the case for $\exists \pi.\varphi$. If $v = \mathsf{yes}$, we derive:

$$\begin{aligned} \mathrm{DM}_\sigma(\exists \pi.\varphi) &= \bigvee_{\ell \in \mathcal{L}} \mathrm{DM}_{\sigma[\pi \mapsto \ell]}(\varphi) \Rrightarrow \mathsf{yes} \\ &\Leftrightarrow \exists \ell \in \mathcal{L}.\mathrm{DM}_{\sigma[\pi \mapsto \ell]}(\varphi) \Rrightarrow \mathsf{yes} && \text{(Lemma B.9)} \\ &\overset{IH}{\Leftrightarrow} \exists \ell \in \mathcal{L}.\mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\varphi) \Rrightarrow \mathsf{yes} \\ &\Leftrightarrow \bigoplus_{\ell \in \mathcal{L}} \mathsf{cm}_{\sigma[\pi \mapsto \ell]}(\varphi) = \mathsf{cm}_\sigma(\exists \pi.\varphi) \Rrightarrow \mathsf{yes}. && \text{(Lemma B.9)} \end{aligned}$$

The case for $v = \mathsf{no}$ is dual. The cases for $\forall \pi.\varphi$, $\varphi_1 \wedge \varphi_2$ and $\varphi_1 \vee \varphi_2$ are similar. □

B.2 Proofs for Verdict Irrevocability of $\mathrm{DM}_{-}(\cdot)$

We start by proving Verdict Irrevocability at the level of local monitors.

Lemma B.11. *Let m be a relevant local monitor. If $m \xrightarrow{c} n$ and $m \Rrightarrow v$, then $n \Rrightarrow v$.*

Proof. We proceed by induction on m. The base case for $m = \mathrm{Dm}^\ell_\sigma(\psi)$ can be concluded from Lemma B.2; in the base cases for $m = (!G, a).\mathrm{Dm}^\ell_\sigma(\psi)$ and $m = \sum_{a \in \mathsf{Act}}(?\{\ell'\}, a).\mathrm{Dm}^\ell_\sigma(\psi_a)$, the premise $m \Rrightarrow v$ is not satisfied.

For the inductive step, if $m = n_1 \otimes n_2$ with n_1 and n_2 relevant monitors, then $m \Rrightarrow v$ yields two cases (by Lemma A.1, $v \neq \mathsf{end}$ and we can thus exclude that case):

(1) $n_1 \Rrightarrow \mathsf{no}$ and $v = \mathsf{no}$. From $m \xrightarrow{c} n$ we derive another three cases, where we let $n = o_1 \otimes o_2$:
  (a) $n_1 \xrightarrow{c} o_1$ and $n_2 = o_2$. We use the induction hypothesis to obtain that $o_1 \Rrightarrow \mathsf{no}$. Then, $n \Rrightarrow \mathsf{no}$ follows immediately.
  (b) $n_2 \xrightarrow{c} o_2$ and $n_1 = o_1$. Like before, $o_2 \Rrightarrow \mathsf{no}$ and $n \Rrightarrow \mathsf{no}$.
  (c) $c = (?\ell', \gamma)$, $n_1 \xrightarrow{c} o_1$ and $n_2 \xrightarrow{c} o_2$. By induction, $o_1 \Rrightarrow \mathsf{no}$, $o_2 \Rrightarrow \mathsf{no}$ and $n \Rrightarrow \mathsf{no}$.
(2) $n_1 \Rrightarrow \mathsf{yes}$ and $n_2 \Rrightarrow v$. This case is symmetric.

The case for $\oplus$ is similar to the one for $\otimes$ and, therefore, we omit it. □

Next we show that the verdict is not end and that verdicts carry over receiving transitions:

Lemma B.12. *Let $M \in \mathsf{DMon}$ be action-derived. If $M \Rrightarrow v$, then $v \neq \mathsf{end}$.*

Lemma B.13. *Let $M \in \mathsf{DMon}$ be action-derived. If $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$ and $M \Rrightarrow v$, then $N \Rrightarrow v$.*

Proof. We proceed by induction on $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$. For the first base case, we consider $M = [m]_{\ell'}$, $N = [n]_{\ell'}$ and $m \xrightarrow{(?\ell,\gamma)} n$: then, $M \Rrightarrow v$ implies that $m \Rrightarrow v$, and (via Lemma B.11, that can be used thanks to Lemma B.6) we can conclude $n \Rrightarrow v$, thus $N \Rrightarrow v$. In the other two base cases, we have $M = N$ and the result follows immediately.

For the inductive step we consider $M = M_1 \otimes M_2$ (the case for $M = M_1 \oplus M_2$ is similar), $N = N_1 \otimes N_2$, $M_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_1$ and $M_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$. From $M_1 \otimes M_2 \Rrightarrow v$ we distinguish two cases, where we use Lemma B.12 to exclude the case for $v = \mathsf{end}$:

(1) $M_1 \Rrightarrow \mathsf{yes}$ and $M_2 \Rrightarrow v$, From the induction hypothesis we get $N_1 \Rrightarrow \mathsf{yes}$ and $N_2 \Rrightarrow v$. We immediately obtain that $N_1 \otimes N_2 \Rrightarrow v$ according to the transition rules for $\Rrightarrow$.
(2) $M_1 \Rrightarrow \mathsf{no}$ and $v = \mathsf{no}$. We use the induction hypothesis to obtain that $N_1 \Rrightarrow \mathsf{no}$, and the result follows immediately. □

We first prove Verdict Irrevocability for just one communication step; Verdict Irrevocability then easily follows via straightforward induction.

Lemma B.14. *Let $M \in \mathsf{DMon}$ be action-derived. If $M \xrightarrow{c} N$ and $M \Rrightarrow v$, then $N \Rrightarrow v$.*

Proof. Let $c = \ell : (!G, \gamma)$; we proceed by induction on $M \xrightarrow{c} N$. For the base case, we consider $M = [m]_\ell$, $N = [n]_\ell$ and $m \xrightarrow{(!G,\gamma)} n$: then, $M \Rrightarrow v$ implies that $m \Rrightarrow v$; we conclude by Lemma B.6 and B.11.

For the inductive step, we consider $M = M_1 \otimes M_2$ (the case for $M = M_1 \oplus M_2$ is similar), $N = N_1 \otimes N_2$, $M_1 \xrightarrow{c} N_1$ and $M_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$. From $M_1 \otimes M_2 \Rrightarrow v$ we distinguish four cases, where we use Lemma B.12 to exclude the case for $v = \mathsf{end}$:

(1) $M_1 \Rrightarrow \mathsf{no}$, $v = \mathsf{no}$. We use the induction hypothesis to obtain that $N_1 \Rrightarrow \mathsf{no}$, and the result follows.
(2) $M_2 \Rrightarrow \mathsf{no}$, $v = \mathsf{no}$. We use Lemma B.13 to derive that $N_2 \Rrightarrow \mathsf{no}$, which concludes the proof.
(3) $M_1 \Rrightarrow \mathsf{yes}$, $M_2 \Rrightarrow v$. We use the induction hypothesis to obtain that $N_1 \Rrightarrow \mathsf{yes}$ and Lemma B.13 to derive that $N_2 \Rrightarrow v$, and the result follows.
(4) $M_2 \Rrightarrow \mathsf{yes}$, $M_1 \Rrightarrow v$. Same as the previous case. □

Corollary B.15. *If $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M_1 \twoheadrightarrow M_2 \twoheadrightarrow M$ and $M_2 \Rrightarrow v$, then $M \Rrightarrow v$.*

B.3 Proofs for Reactivity of $\mathrm{DM}_{-}(\cdot)$

We start by defining an ordering on the formulae of kind ψ, based on the number of top-level fixed-point operators in them, where by "top-level" we mean "not under the scope of a modality." For example, the number of top-level fixed-points in:

$$\max x.\max y.([a_\pi]x \wedge \langle b_{\pi'}\rangle y) \quad \vee \quad \max z.[a_\pi]\max t.(\langle b_{\pi'}\rangle t \wedge z), \tag{6}$$

is 3, max t being the only non-top-level fixed-point operator.

Definition B.16. We let $\psi \prec \psi'$ iff:

(1) the number of top-level fixed-point operators in ψ is smaller than the number of top-level fixed-point operators in ψ', or
(2) the number is the same and ψ is a strict subformula of ψ'.

Notice that $\prec$ is well-founded and that the bottom elements are tt, ff, $\pi = \pi$, $\pi \neq \pi$ and x. Coming back to (Equation (6)), we have that:

$$\begin{array}{rcl} \max z.[a_\pi] \max t.(\langle b_{\pi'}\rangle t \wedge z) & \prec & (\max x.[b_\pi]x \wedge \max y.\langle a_{\pi'}\rangle y) \\ \max t.(\langle b_{\pi'}\rangle t \wedge z) & \prec & \max z.[a_\pi] \max t.(\langle b_{\pi'}\rangle t \wedge z), \end{array}$$

respectively by point 1 and 2 of Definition B.16.

Lemma B.17. *For all $\varphi \in$ PHyper-maxHML without free recursion variables, σ such that* $\mathsf{fv_l}(\varphi) \subseteq \mathsf{dom}(\sigma)$*, and $A : \mathcal{L} \to \mathsf{Act}$, there exists $M \in \mathsf{DMon}$ such that $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M$.*

Proof. We proceed by induction on φ; the proof is trivial for all the inductive cases. So we only treat the base case, where $\varphi = \psi \in \mathsf{Qf}$ and has no free recursion variables. We prove that $\mathrm{Dm}^\ell_\sigma(\psi)$ is reactive by induction on $\prec$. The base cases are the bottom elements of $\prec$ and they all entail that $\mathrm{Dm}^\ell_\sigma(\psi)$ is a verdict; then, the claim follows immediately because $v \xrightarrow{a} v$ holds for each v and a. For the inductive case:

— If $\psi = [a_\pi]\psi'$ or $\psi = \langle a_\pi\rangle\psi'$, for some a, π and ψ', then $\mathrm{Dm}^\ell_\sigma(\psi)$ is reactive by construction.
— If $\psi = \psi_1 \diamond \psi_2$, then $\mathrm{Dm}^\ell_\sigma(\psi) = \mathrm{Dm}^\ell_\sigma(\psi_1) \odot \mathrm{Dm}^\ell_\sigma(\psi_2)$ (with $\odot = \otimes$ if $\diamond = \wedge$ and $\odot = \oplus$ if $\diamond = \vee$). By Definition B.16, $\psi_1 \prec \psi$ and $\psi_2 \prec \psi$; hence, the inductive hypothesis yields that $\mathrm{Dm}^\ell_\sigma(\psi_1)$ and $\mathrm{Dm}^\ell_\sigma(\psi_2)$ are reactive. Reactivity of $\mathrm{Dm}^\ell_\sigma(\psi)$ immediately follows from the operational rules for $\odot$.
— If $\psi = \max x.\psi'$, we show that, for every $a \in \mathsf{Act}$, there exists $m \in \mathsf{LMon}$ such that $\mathrm{Dm}^\ell_\sigma(\max x.\psi') = \mathsf{rec}\ x.\mathrm{Dm}^\ell_\sigma(\psi') \xrightarrow{a} m$. Since formulas are guarded, $\psi'\{^{\max x.\psi'}/_x\} \prec \max x.\psi'$; therefore, the induction hypothesis yields that $\mathrm{Dm}^\ell_\sigma(\psi'\{^{\max x.\psi'}/_x\}) \xrightarrow{a} m$ for some m. By Lemma B.1, we have that $\mathrm{Dm}^\ell_\sigma(\psi')\{^{\mathsf{rec}\ x.\mathrm{Dm}^\ell_\sigma(\psi')}/_x\} = \mathrm{Dm}^\ell_\sigma(\psi'\{^{\max x.\psi'}/_x\})$, which concludes the proof, by using the operational rule for recursive monitors. □

B.4 Proofs for Bounded Communication of $\mathrm{DM}_{-}(\cdot)$

We first show the existence of a finite communication path to a monitor that cannot communicate, and then argue that this monitor is unique. To this aim, we start with a few lemmas that are all stating basic facts regarding communication, and are used in many of the lemmas that follow. We start with three basic facts regarding receiving and sending of messages for action-derived local monitors: the first item states that local monitors can never receive the same message from the same sender in consecutive communications; the second one states that, if a monitor cannot receive a certain message from some sender, then it cannot receive that message in the future after taking only communication transitions; the last one states that monitors can only send the action they read last.

Lemma B.18. *Let $m \in \mathsf{LMon}$ be A-derived for ℓ. Then*:

(1) *If $m \xrightarrow{(?\ell',\gamma)} n$, then $n \overset{(?\ell',\gamma)}{\not\longrightarrow}$.*
(2) *If $m \overset{(?\ell',\gamma)}{\not\longrightarrow}$ and $m \twoheadrightarrow n$, then $n \overset{(?\ell',\gamma)}{\not\longrightarrow}$.*
(3) *If $m \xrightarrow{(!G,\gamma)} n$, then $\gamma = A(\ell)$.*

Proof. We prove the first two items for relevant local monitors, which is sufficient because m is a relevant local monitor via Lemma B.6. We proceed by induction on the structure of m.

In the base case for $m = \mathrm{DM}^{\ell}_{\sigma}(\psi)$, by Lemma B.2 m cannot communicate; hence, the premise of the first item is not satisfied and in the second item we get $n = m$, making the statement trivially true. For the base case with $m = (!G, a).\mathrm{DM}^{\ell}_{\sigma}(\psi)$, the premise of the first item is not satisfied; for the second item, $n = m$ or $n = \mathrm{DM}^{\ell}_{\sigma}(\psi)$, and the result follows from the premise of the lemma or Lemma B.2. If $m = \sum_{a\in\mathsf{Act}}(?\{\ell''\}, a).\mathrm{DM}^{\ell}_{\sigma}(\psi_a)$, both items follow immediately from Lemma B.2.

For the inductive step, let $m = m_1 \odot m_2$, with m_1 and m_2 relevant monitors. For the first statement, we derive two cases from $m_1 \odot m_2 \xrightarrow{(?\ell',\gamma)} n$.

(1) $n = n_1 \odot n_2$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_1$ and $m_2 \xrightarrow{(?\ell',\gamma)} n_2$. The induction hypothesis gives that $n_1 \not\xrightarrow{(?\ell',\gamma)}$ and $n_2 \not\xrightarrow{(?\ell',\gamma)}$. Hence, $n \not\xrightarrow{(?\ell',\gamma)}$.
(2) $n = n_1 \odot m_2$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_1$ and $m_2 \not\xrightarrow{(?\ell',\gamma)}$. From the induction hypothesis, we get that $n_1 \not\xrightarrow{(?\ell',\gamma)}$. Hence, $n = n_1 \odot m_2 \not\xrightarrow{(?\ell',\gamma)}$.

For the second statement we derive from $m_1 \odot m_2 \not\xrightarrow{(?\ell',\gamma)}$ that $m_1 \not\xrightarrow{(?\ell',\gamma)}$ and $m_2 \not\xrightarrow{(?\ell',\gamma)}$. Furthermore, we know that $n = n_1 \odot n_2$ such that $m_1 \twoheadrightarrow n_1$ and $m_2 \twoheadrightarrow n_2$. From the induction hypothesis, we obtain that $n_1 \not\xrightarrow{(?\ell',\gamma)}$ and $n_2 \not\xrightarrow{(?\ell',\gamma)}$. Hence, $n \not\xrightarrow{(?\ell',\gamma)}$.

Let us now prove the third item. Because m is A-derived for ℓ, we know that there exist σ, ψ and m' such that $\mathrm{DM}^{\ell}_{\sigma}(\psi) \xrightarrow{A(\ell)} m' \twoheadrightarrow m$; we proceed by induction on $\xrightarrow{A(\ell)}$. In the base case $\mathrm{DM}^{\ell}_{\sigma}(\psi) = a.m'$; from the synthesis function and our assumption that $|\mathsf{Act}| \geq 2$, we observe that this never occurs.

For the inductive case, if $\mathrm{DM}^{\ell}_{\sigma}(\psi) = m_1 + m_2$ and $m_1 \xrightarrow{A(\ell)} m'$, ψ must be of the form $[a_\pi]\psi'$ or $\langle a_\pi\rangle\psi'$; so:

$$\mathrm{DM}^{\ell}_{\sigma}(\psi) = a.(!\{\sigma(\pi') \mid \sigma(\pi') \neq \ell\}, a).\mathrm{DM}^{\ell}_{\sigma}(\psi') + \sum_{b\neq a} b.(!\{\sigma(\pi') \mid \sigma(\pi') \neq \ell\}, b).v,$$

or:

$$\mathrm{DM}^{\ell}_{\sigma}(\psi) = \sum_{b\in\mathsf{Act}} b.\big((?\{\sigma(\pi)\}, a).\mathrm{DM}^{\ell}_{\sigma}(\psi') + \sum_{b\neq a}(?\{\sigma(\pi)\}, b).v\big),$$

where $v = \mathsf{yes}/\mathsf{no}$ according to the modality (resp., box/diamond). It is immediate that $\gamma = A(\ell)$ in $m \xrightarrow{(!G,\gamma)} n$.

Next we consider the case for $\mathrm{DM}^{\ell}_{\sigma}(\psi) = m''_1 \odot m''_2$ and $m' = m'_1 \odot m'_2$. Hence, $\psi = \psi_1 \diamond \psi_2$ (for $\diamond = \wedge$ if $\odot = \otimes$ and $\diamond = \vee$ if $\odot = \oplus$), $m''_1 = \mathrm{DM}^{\ell}_{\sigma}(\psi_1) \xrightarrow{A(\ell)} m'_1$ and $m''_2 = \mathrm{DM}^{\ell}_{\sigma}(\psi_2) \xrightarrow{A(\ell)} m'_2$. Let $m = m_1 \odot m_2$, where $m'_1 \twoheadrightarrow m_1$ and $m'_2 \twoheadrightarrow m_2$. We assume without loss of generality that $n = n_1 \odot m_2$ and $m_1 \xrightarrow{(!G,\gamma)} n_1$. From the induction hypothesis, it follows that $\gamma = A(\ell)$.

Last we consider $\mathrm{DM}^{\ell}_{\sigma}(\psi) = \mathsf{rec}\ x.m'' \xrightarrow{A(\ell)} m'$ because $m''\{^{\mathsf{rec}\ x.m''}/_{x}\} \xrightarrow{A(\ell)} m'$. Thus, $\psi = \max x.\psi'$ and $m'' = \mathrm{DM}^{\ell}_{\sigma}(\psi')$. By Lemma B.1, $\mathrm{DM}^{\ell}_{\sigma}(\psi')\{^{\mathsf{rec}\ x.\mathrm{DM}^{\ell}_{\sigma}(\psi')}/_{x}\} = \mathrm{DM}^{\ell}_{\sigma}(\psi'\{^{\max x.\psi'}/_{x}\})$ and we use the induction hypothesis to obtain the desired result. □

We generalize the third property from Lemma B.18 to communicating monitors: if an action-derived monitor M can take a communicating transition, the message it sends is always the last action taken by the location of the sender:

Lemma B.19. *Let $M \in$ DMon be A-derived. If $M \xrightarrow{\ell:(!G,\gamma)} N$, then $\gamma = A(\ell)$.*

We now state a few other basic properties: Lemma B.32 uses a 'commutativity' property for communicating monitors (Lemma B.30), and this is based on five similar results for local monitors (Lemma B.20-Lemma B.24), that are in addition used in a variety of other lemmas.

The following lemma shows that, if action-derived local monitors can both send and receive a message, the order in which this happens does not matter.

Lemma B.20. *Let m be a relevant local monitor such that $m \xrightarrow{(!G,\gamma)} m_1$ and $m \xrightarrow{(?\ell',\gamma')} m_2$, for some $G \subseteq \mathcal{L}$ and $\ell' \in \mathcal{L}$. Then, $m_1 \xrightarrow{(?\ell',\gamma')} m_3$ and $m_2 \xrightarrow{(!G,\gamma)} m_3$, for some m_3.*

Proof. We proceed by induction on m. In the base case for $m = \text{DM}^{\ell}_{\sigma}(\psi)$, we conclude by Lemma B.2; in the base cases for $m = (!G, a).\text{DM}^{\ell}_{\sigma}(\psi)$ and $m = \sum_{a\in\text{Act}}(?\{\ell''\}, a).\text{DM}^{\ell}_{\sigma}(\psi_a)$, the premise of the lemma is not satisfied.

For the inductive step, let $m = n_1 \odot n_2$, with n_1 and n_2 relevant monitors and $\odot \in \{\otimes, \oplus\}$. From $m \xrightarrow{(!G,\gamma)} m_1$ we derive w.l.o.g. that $m_1 = n_1' \odot n_2$ and $n_1 \xrightarrow{(!G,\gamma)} n_1'$. From $m \xrightarrow{(?\ell',\gamma')} m_2$ we derive three cases.

(1) $m_2 = m_2' \odot m_2''$ such that $n_1 \xrightarrow{(?\ell',\gamma')} m_2'$ and $n_2 \xrightarrow{(?\ell',\gamma')} m_2''$. From the induction hypothesis, we derive that $n_1' \xrightarrow{(?\ell',\gamma')} m_3'$ and $m_2' \xrightarrow{(!G,\gamma)} m_3'$, for some m_3'. Thus, $m_1 = n_1' \odot n_2 \xrightarrow{(?\ell',\gamma')} m_3' \odot m_2''$ and $m_2 = m_2' \odot m_2'' \xrightarrow{(!G,\gamma)} m_3' \odot m_2''$, as desired.
(2) $m_2 = m_2' \odot n_2$ such that $n_1 \xrightarrow{(?\ell',\gamma')} m_2'$ and $n_2 \not\xrightarrow{(?\ell',\gamma')}$. From the induction hypothesis, we derive $n_1' \xrightarrow{(?\ell',\gamma')} m_3'$ and $m_2' \xrightarrow{(!G,\gamma)} m_3'$, for some m_3'. Thus, $m_1 = n_1' \odot n_2 \xrightarrow{(?\ell',\gamma')} m_3' \odot n_2$ and $m_2 = m_2' \odot n_2 \xrightarrow{(!G,\gamma)} m_3' \odot n_2$.
(3) $m_2 = n_1 \odot m_2'$ such that $n_1 \not\xrightarrow{(?\ell',\gamma')}$ and $n_2 \xrightarrow{(?\ell',\gamma')} m_2'$. We obtain that $m_2 = n_1 \odot m_2' \xrightarrow{(!G,\gamma)} n_1' \odot m_2'$ and $m_1 = n_1' \odot n_2 \xrightarrow{(?\ell',\gamma')} n_1' \odot m_2'$, where we use Lemma B.18 to conclude that $n_1' \not\xrightarrow{(?\ell',\gamma')}$ from $n_1 \not\xrightarrow{(?\ell',\gamma')}$. □

The next lemma proves uniqueness of reached states after a receiving transition.

Lemma B.21. *Let m be a relevant local monitor such that $m \xrightarrow{(?\ell',\gamma)} n_1$ and $m \xrightarrow{(?\ell',\gamma)} n_2$. Then, $n_1 = n_2$.*

Proof. We proceed by induction on m. In the base case for $m = \text{DM}^{\ell}_{\sigma}(\psi)$, we conclude by Lemma B.2. For $m = (!G, a).\text{DM}^{\ell}_{\sigma}(\psi)$, the premise of the lemma is not satisfied; for $m = \sum_{a\in\text{Act}}(?\{\ell''\}, a).\text{DM}^{\ell}_{\sigma}(\psi_a)$, we can conclude because there is only one entry in the sum for each combination of ℓ'' and a.

For the inductive step, let $m = m_1 \odot m_2$, with m_1 and m_2 relevant monitors. From $m \xrightarrow{(?\ell',\gamma)} n_1$ we derive two cases.

(1) $n_1 = n_1' \odot n_1''$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_1'$ and $m_2 \xrightarrow{(?\ell',\gamma)} n_1''$. From $m \xrightarrow{(?\ell',\gamma)} n_2$ we derive two more cases.
  (a) $n_2 = n_2' \otimes n_2''$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_2'$ and $m_2 \xrightarrow{(?\ell',\gamma)} n_2''$. From the induction hypothesis, we conclude that $n_1' = n_2'$ and $n_1'' = n_2''$, and thus that $n_1 = n_2$.

(b) $n_2 = n_2' \odot m_2$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_2'$ and $m_2 \not\xrightarrow{(?\ell',\gamma)}$. This is a contradiction with $m_2 \xrightarrow{(?\ell',\gamma)} n_1''$.

(2) $n_1 = n_1' \odot m_2$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_1'$ and $m_2 \not\xrightarrow{(?\ell',\gamma)}$. We then conclude from $m \xrightarrow{(?\ell',\gamma)} n_2$ that $n_2 = n_2' \odot m_2$ such that $m_1 \xrightarrow{(?\ell',\gamma)} n_2'$. From the induction hypothesis, we conclude that $n_1' = n_2'$ and thus that $n_1 = n_2$. □

The next lemma states that, if a monitor can receive a message from two different senders, the order in which this happens does not matter.

Lemma B.22. *Let m be a relevant local monitor such that $m \xrightarrow{(?\ell_1,\gamma_1)} m_1$ and $m \xrightarrow{(?\ell_2,\gamma_2)} m_2$, where $\ell_1 \neq \ell_2$. Then, $m_1 \xrightarrow{(?\ell_2,\gamma_2)} m_3$ and $m_2 \xrightarrow{(?\ell_1,\gamma_1)} m_3$, for some m_3.*

Proof. We proceed by induction on m. In the base case, for $m = \text{DM}^{\ell}_{\sigma}(\psi)$, we conclude by Lemma B.2; for $m = (!G, a).\text{DM}^{\ell}_{\sigma}(\psi)$ and $m = \sum_{a\in\text{Act}}(?\{\ell'\}, a).\text{DM}^{\ell}_{\sigma}(\psi_a)$, the premise of the lemma is not satisfied. For the inductive step, let $m = n_1 \odot n_2$, with n_1 and n_2 relevant monitors. From $m \xrightarrow{(?\ell_1,\gamma_1)} m_1$ and $m \xrightarrow{(?\ell_2,\gamma_2)} m_2$, w.l.o.g. we derive the following four cases.

(1) $m_1 = n_1' \odot n_2'$ and $m_2 = n_1'' \odot n_2''$, where:

$$n_1 \xrightarrow{(?\ell_1,\gamma_1)} n_1', \tag{7}$$

$$n_2 \xrightarrow{(?\ell_1,\gamma_1)} n_2', \tag{8}$$

$$n_1 \xrightarrow{(?\ell_2,\gamma_2)} n_1'', \text{ and} \tag{9}$$

$$n_2 \xrightarrow{(?\ell_2,\gamma_2)} n_2''. \tag{10}$$

From the induction hypothesis on Equations (7) and (9) and on Equations (8) and (10), $n_1' \xrightarrow{(?\ell_2,\gamma_2)} r_1$ and $n_1'' \xrightarrow{(?\ell_1,\gamma_1)} r_1$, for some r_1, and $n_2' \xrightarrow{(?\ell_2,\gamma_2)} r_2$ and $n_2'' \xrightarrow{(?\ell_1,\gamma_1)} r_2$, for some r_2. We conclude that $m_1 \xrightarrow{(?\ell_2,\gamma_2)} r_1 \odot r_2$ and $m_2 \xrightarrow{(?\ell_1,\gamma_1)} r_1 \odot r_2$.

(2) $m_1 = n_1' \odot n_2'$ and $m_2 = n_1'' \odot n_2$, where Equations (7)–(9) hold, and:

$$n_2 \not\xrightarrow{(?\ell_2,\gamma_2)}. \tag{11}$$

From the induction hypothesis on Equations (7) and (9), we get that:

$$n_1' \xrightarrow{(?\ell_2,\gamma_2)} r_1 \text{ and} \tag{12}$$

$$n_1'' \xrightarrow{(?\ell_1,\gamma_1)} r_1, \tag{13}$$

for some r_1. Lemma B.18 and Equation (11) yield that $n_2' \not\xrightarrow{(?\ell_2,\gamma_2)}$; therefore, from Equation (12), we get that $m_1 = n_1' \odot n_2' \xrightarrow{(?\ell_2,\gamma_2)} r_1 \odot z_2$; finally, Equations (8) and (13) yield $m_2 = n_1'' \odot n_2 \xrightarrow{(?\ell_1,\gamma_1)} r_1 \odot n_2$.

(3) $m_1 = n_1' \odot n_2$ and $m_2 = n_1'' \odot n_2$, where Equations (7), (9), and (11) hold, and:

$$n_2 \not\xrightarrow{(?\ell_1,\gamma_1)}. \tag{14}$$

From the induction hypothesis on Equations (7) and (9), we get Equations (12) and (13), for some r_1; this, together with Equations (11) and (14), yields $m_1 = n'_1 \odot n_2 \xrightarrow{(?\ell_2,\gamma_2)} r_1 \odot n_2$ and $m_2 = n''_1 \odot n_2 \xrightarrow{(?\ell_1,\gamma_1)} r_1 \odot n_2$.

(4) $m_1 = n'_1 \odot n_2$ and $m_2 = n_1 \odot n''_2$, where Equations (7), (10), and (14) hold, and:

$$n_1 \overset{(?\ell_2,\gamma_2)}{\not\longrightarrow} . \tag{15}$$

Lemma B.18, with Equations (7) and (15) and with Equations (10) and (14), yields that $n'_1 \overset{(?\ell_2,\gamma_2)}{\not\longrightarrow}$ and $n''_2 \overset{(?\ell_1,\gamma_1)}{\not\longrightarrow}$, respectively; therefore, from Equations (7) and (10), we get that $m_1 = n'_1 \odot n_2 \xrightarrow{(?\ell_2,\gamma_2)} n'_1 \odot n''_2$ and $m_2 = n_1 \odot n''_2 \xrightarrow{(?\ell_1,\gamma_1)} n'_1 \odot n''_2$. □

The following lemma shows that if action-derived local monitors can send two different messages, the order in which this happens does not matter.

Lemma B.23. *Let m be a relevant local monitor such that $m \xrightarrow{(!G,\gamma)} m_1$ and $m \xrightarrow{(!G',\gamma')} m_2$, for some $G, G' \subseteq \mathcal{L}$ and $\gamma, \gamma' \in \mathsf{Con}$ such that either $G \neq G'$ or $\gamma \neq \gamma'$. Then, $m_1 \xrightarrow{(!G',\gamma')} m_3$ and $m_2 \xrightarrow{(!G,\gamma)} m_3$, for some m_3.*

Proof. We proceed by induction on m. The base case for $m = \mathrm{DM}^{\ell}_{\sigma}(\psi)$ follows from Lemma B.2; for $m = (!G, a).\mathrm{DM}^{\ell}_{\sigma}(\psi)$ and $m = \sum_{a \in \mathsf{Act}}(?\{\ell'\}, a).\mathrm{DM}^{\ell}_{\sigma}(\psi_a)$, the premise of the lemma is not satisfied.

For the inductive step, let $m = n_1 \odot n_2$, with n_1 and n_2 relevant monitors. From $m \xrightarrow{(!G,\gamma)} m_1$ we derive w.l.o.g. that $m_1 = n'_1 \odot n_2$ and $n_1 \xrightarrow{(!G,\gamma)} n'_1$. From $m \xrightarrow{(!G',\gamma')} m_2$ we derive two cases.

(1) $m_2 = n''_1 \odot n_2$ such that $n_1 \xrightarrow{(!G',\gamma')} n''_1$. From the induction hypothesis, we derive that $n'_1 \xrightarrow{(!G',\gamma')} n'$ and $n''_1 \xrightarrow{(!G,\gamma)} n'$, for some n'. Thus, $n'_1 \odot n_2 \xrightarrow{(!G',\gamma')} n' \odot n_2$ and $n''_1 \odot n_2 \xrightarrow{(!G,\gamma)} n' \odot n_2$, as desired.

(2) $m_2 = n_1 \odot n''_2$ such that $n_2 \xrightarrow{(!G',\gamma')} n''_2$. Then, $n_1 \odot n''_2 \xrightarrow{(!G,\gamma)} n'_1 \odot n''_2$ and $n'_1 \odot n_2 \xrightarrow{(!G',\gamma')} n'_1 \odot n''_2$. □

The following lemma is the version of Lemma B.21 for sending.

Lemma B.24. *Let m be a relevant local monitor such that $m \xrightarrow{(!G,\gamma)} n_1$ and $m \xrightarrow{(!G,\gamma)} n_2$. Then, either $n_1 = n_2$, or $n_1 \xrightarrow{(!G,\gamma)} m'$ and $n_2 \xrightarrow{(!G,\gamma)} m'$, for some m'.*

Before we continue on the path to proving Lemma B.32, we show how to formally capture the state of a decentralized monitor after taking a receiving transition, for which we only need Lemma B.21. To this aim, we use $M[[n]_\ell/[m]_\ell]$ as notation for replacing $[m]_\ell$ in M with $[n]_\ell$ (the operation does not change M in case $[m]_\ell$ does not occur in M).

Lemma B.25. *Let M and N be action-derived monitors such that $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$; let $\{[m_i]_{\ell_i} \mid 1 \leq i \leq k\}$ be the set of all the $[m]_\ell \in M$. For each $1 \leq i \leq k$, let n_i be m_i, when $\ell_i \notin G$ or $m_i \overset{(?\ell,\gamma)}{\not\longrightarrow}$, and be such that $m_i \xrightarrow{(?\ell,\gamma)} n_i$, otherwise. Then, $M[[n_1]_{\ell_1}/[m_1]_{\ell_1}] \cdots [[n_k]_{\ell_k}/[m_k]_{\ell_k}] = N$.*

Lemmas B.26–B.29 lift the previous results from local to decentralized monitors and are used to prove Lemma B.30. This result, together with Lemma B.31, leads to Lemma B.32, which captures that the order of communication does not matter: regardless of the order in which the communication steps are executed, the 'final' state (i.e., the state where no more communication can take place) is unique.

Lemma B.26. *Let M be action-derived. If $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$, then $N \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$.*

Proof. We proceed by induction on $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N$. In the first base case we consider $M = [m]_{\ell'}$, $N = [n]_{\ell'}$, $\ell' \in G$, and $m \xrightarrow{(?\ell,\gamma)} n$: by Lemma B.18, which we can apply because M is action-derived, we obtain that $n \overset{(?\ell,\gamma)}{\not\longrightarrow}$, and so $[n]_{\ell'} \overset{G:(?\ell,\gamma)}{\rightsquigarrow} [n]_{\ell'}$. In the second and third base case we consider $M = [m]_{\ell'} = N$, and thus we are done immediately. In the inductive case we consider $M = M_1 \diamond M_2$, $N = N_1 \diamond N_2$, $M_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_1$ and $M_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$. From the induction hypothesis we get that $N_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_1$ and $N_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$, and thus that $N = N_1 \diamond N_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_1 \diamond N_2$. □

Lemma B.27. *Let M be A-derived and such that $M \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} N_1$ and $M \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_2$. Then, either $N_1 = N_2$, or $N_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_3$ and $N_2 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} N_3$, for some N_3.*

Proof. We proceed by induction on $M \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} N_1$. In the first base case we consider $M = [m]_\ell$, $N_1 = [n_1]_\ell$, $\ell \in G_1$, and $m \xrightarrow{(?\ell_1,A(\ell_1))} n_1$. We distinguish two cases:

(1) $N_2 = [n_2]_\ell$, $m \xrightarrow{(?\ell_2,A(\ell_2))} n_2$, $\ell \in G_2$. Since M is action-derived, we can use Lemma B.22 to obtain two further cases:
   (a) $\ell_1 \neq \ell_2$ and $n_1 \xrightarrow{(?\ell_2,A(\ell_2))} q$ and $n_2 \xrightarrow{(?\ell_1,A(\ell_1))} q$. Hence, $N_1 = [n_1]_\ell \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} [q]_\ell$ and $N_2 = [n_2]_\ell \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} [q]_\ell$.
   (b) $\ell_1 = \ell_2$. Via Lemma B.21 we conclude that $n_1 = n_2$ and thus that $N_1 = N_2$.

(2) $N_2 = [m]_\ell$ and either $\ell \notin G_2$ or $m \overset{(?\ell_2,A(\ell_2))}{\not\longrightarrow}$. In the latter case, we know by Lemma B.18(2) that $n_1 \overset{(?\ell_2,A(\ell_2))}{\not\longrightarrow}$. Hence, in both cases, $N_1 = [n_1]_\ell \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} [n_1]_\ell = N_1$. Since $N_2 = M$, we know that $N_2 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} N_1$, as desired.

For the next two base cases, we consider $N_1 = [m]_\ell$ and either $\ell \notin G_1$ or $m \overset{(?\ell_1,A(\ell_1))}{\not\longrightarrow}$. We distinguish two cases:

(1) $N_2 = [n_2]_\ell$, $m \xrightarrow{(?\ell_2,A(\ell_2))} n_2$, $\ell \in G_2$. If $m \overset{(?\ell_1,A(\ell_1))}{\not\longrightarrow}$, Lemma B.18(2) entails that $n_2 \overset{(?\ell_1,A(\ell_1))}{\not\longrightarrow}$. Hence, in all cases, $N_2 = [n_2]_\ell \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} [n_2]_\ell = N_2$. Since $N_1 = M$, we know that $N_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_2$, as desired.

(2) $N_2 = [m]_\ell$ and either $\ell \notin G_2$ or $m \overset{(?\ell_2,A(\ell_2))}{\not\longrightarrow}$. Hence, $M = N_2 = N_1$, and we are done immediately.

For the inductive step, we let $M = M_1 \diamond M_2$, $N_1 = O_1 \diamond O_2$, $M_1 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_1$ and $M_2 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_2$. This implies that $N_2 = O_3 \diamond O_4$, with $M_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_3$ and $M_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_4$. We apply the induction hypothesis to $M_1 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_1$ to obtain two cases:

(1) $O_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q_1$ and $O_3 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} Q_1$. We apply the induction hypothesis to $M_2 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_2$ to obtain two more cases:
   (a) $O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q_2$ and $O_4 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} Q_2$. Hence, $O_1 \diamond O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q_1 \diamond Q_2$ and $O_3 \diamond O_4 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} Q_1 \diamond Q_2$.
   (b) $O_2 = O_4$. We apply Lemma B.26 to $M_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_4$ to conclude that $O_4 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_4$. Being $O_2 = O_4$, we obtain that $O_1 \diamond O_2 = O_1 \diamond O_4 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q_1 \diamond O_4$ and $O_3 \diamond O_4 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} Q_1 \diamond O_4$.

(2) $O_1 = O_3$. We apply the induction hypothesis to $M_2 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_2$ to obtain two more cases:
   (a) $O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q_2$ and $O_4 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} Q_2$. We apply Lemma B.26 to $M_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_3$ to conclude that $O_3 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_3$. Being $O_1 = O_3$, we have that $O_1 \diamond O_2 = O_3 \diamond O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_3 \diamond Q_2$ and $O_3 \diamond O_4 \overset{G_1:(?\ell_1,A(\ell_1))}{\rightsquigarrow} O_3 \diamond Q_2$.
   (b) $O_2 = O_4$. Then we immediately have $N_1 = N_2$. □

Lemma B.28. *Let M be action-derived and such that $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_1$ and $M \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$. Then, $N_1 = N_2$.*

Lemma B.29. *Let M be action-derived and such that $M \xrightarrow{\ell_1:(!G_1,\gamma_1)} N_1$ and $M \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_2$. Then, $N_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_3$ and $N_2 \xrightarrow{\ell_1:(!G_1,\gamma_1)} N_3$, for some N_3.*

Proof. We proceed by induction on $M \xrightarrow{\ell_1:(!G_1,\gamma_1)} N_1$. In the base case we consider $M = [m]_{\ell_1}$, $N_1 = [n_1]_{\ell_1}$ and $m \xrightarrow{(!G_1,\gamma_1)} n_1$. We distinguish two cases based on $M \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_2$:

(1) $N_2 = [n_2]_{\ell_1}$, $m \xrightarrow{(?\ell_2,A(\ell_2))} n_2$, and $\ell_1 \in G_2$. Hence, we can apply Lemma B.20 to obtain that there exists m_1 such that $n_1 \xrightarrow{(?\ell_2,A(\ell_2))} m_1$ and $n_2 \xrightarrow{(!G_1,\gamma_1)} m_1$. Hence, $N_2 \xrightarrow{\ell_1:(!G_1,\gamma_1)} [m_1]_{\ell_1}$ and $N_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} [m_1]_{\ell_1}$.

(2) $N_2 = [m]_{\ell_1}$ and either $\ell_1 \notin G_2$ or $m \not\xrightarrow{(?\ell_2,A(\ell_2))}$. The latter case implies that $n_1 \not\xrightarrow{(?\ell_2,A(\ell_2))}$, by applying Lemma B.18 to $m \xrightarrow{(!G_1,\gamma_1)} n_1$. As $M = N_2$, in both cases we have $N_2 \xrightarrow{\ell_1:(!G_1,\gamma_1)} N_1$ and $N_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} N_1$, as desired.

For the inductive case, we consider $M = M_1 \diamond M_2$, $N_1 = O_1 \diamond O_2$, $M_1 \xrightarrow{\ell_1:(!G_1,\gamma_1)} O_1$ and $M_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} O_2$. We deduce that $N_2 = O_3 \diamond O_4$, with $M_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_3$ and $M_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_4$. We use the induction hypothesis to conclude that there exists Q such that $O_1 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q$ and $O_3 \xrightarrow{\ell_1:(!G_1,\gamma_1)} Q$. Since $M \xrightarrow{\ell_1:(!G_1,\gamma_1)} N_1$, we know that $\gamma_1 = A(\ell_1)$ (Lemma B.19). Hence, we can apply Lemma B.27 on $M_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} O_2$ and $M_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} O_4$ and have two cases:

(1) $O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q'$ and $O_4 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} Q'$. Hence, $N_1 = O_1 \diamond O_2 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q \diamond Q'$ and $N_2 = O_3 \diamond O_4 \xrightarrow{\ell_1:(!G_1,\gamma_1)} Q \diamond Q'$.

(2) $O_2 = O_4$. We apply Lemma B.26 to $M_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} O_2$ to conclude that $O_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} O_2$. Being $O_2 = O_4$, we have $N_1 = O_1 \diamond O_2 = O_1 \diamond O_4 \overset{G_2:(?\ell_2,A(\ell_2))}{\rightsquigarrow} Q \diamond O_4$ and $N_2 = O_3 \diamond O_4 \xrightarrow{\ell_1:(!G_1,\gamma_1)} Q \diamond O_4$.
□

Lemma B.30. *Let M be action-derived and such that $M \xrightarrow{c_1} M_1$ and $M \xrightarrow{c_2} M_2$. Then, either $c_1 = c_2$ and $M_1 = M_2$, or $M_1 \xrightarrow{c_2} N$ and $M_2 \xrightarrow{c_1} N$, for some N.*

Proof. We proceed by induction on $M \xrightarrow{c_1} M_1$. Let $c_1 = \ell_1 : (!G_1, \gamma_1)$ and $c_2 = \ell_2 : (!G_2, \gamma_2)$. In the base case we consider $M = [m]_{\ell_1}$, $M_1 = [m']_{\ell_1}$ and $m \xrightarrow{(!G_1,\gamma_1)} m'$. Thus we know that $M_2 = [m'']_{\ell_1}$, $\ell_1 = \ell_2$ and $m \xrightarrow{(!G_2,\gamma_2)} m''$. We distinguish two cases:

(1) $G_1 \neq G_2$ or $\gamma_1 \neq \gamma_2$. Then we apply Lemma B.23 to obtain that $m' \xrightarrow{(!G_2,\gamma_2)} n$ and $m'' \xrightarrow{(!G_1,\gamma_1)} n$. Hence, $M_1 \xrightarrow{c_2} [n]_{\ell_1}$ and $M_2 \xrightarrow{c_1} [n]_{\ell_1}$.
(2) $G_1 = G_2$ and $\gamma_1 = \gamma_2$. We apply Lemma B.24 to obtain that $m' \xrightarrow{(!G_2,\gamma_2)} n$ and $m'' \xrightarrow{(!G_1,\gamma_1)} n$, or that $m' = m''$. In the former case, we conclude like in the first item; in the latter case, we get immediately that $M_1 = M_2$ and $c_1 = c_2$.

For the inductive case, let $M = N_1 \diamond N_2$, $M_1 = N_1' \diamond N_2'$, $N_1 \xrightarrow{c_1} N_1'$ and $N_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} N_2'$. We distinguish two cases.

(1) $M_2 = O_1 \diamond O_2$, with $N_1 \xrightarrow{c_2} O_1$ and $N_2 \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} O_2$. We obtain two further cases from the induction hypothesis applied to N_1:
   (a) $N_1' \xrightarrow{c_2} Q$ and $O_1 \xrightarrow{c_1} Q$. Since $M \xrightarrow{c_1} M_1$ and $M \xrightarrow{c_2} M_2$, we obtain via Lemma B.19 that $\gamma_1 = A(\ell_1)$ and $\gamma_2 = A(\ell_2)$. Thus we can apply Lemma B.27 to $N_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} N_2'$ and $N_2 \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} O_2$ to obtain two further cases:
      (i) $N_2' \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} Q'$ and $O_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} Q'$. Hence $N_1' \diamond N_2' \xrightarrow{c_2} Q \diamond Q'$ and $O_1 \diamond O_2 \xrightarrow{c_1} Q \diamond Q'$, as desired.
      (ii) $N_2' = O_2$. By Lemma B.26, $O_2 \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} O_2$; hence, $N_1' \diamond N_2' = N_1' \diamond O_2 \xrightarrow{c_2} Q \diamond O_2$ and $O_1 \diamond O_2 \xrightarrow{c_1} Q \diamond O_2$.
   (b) $N_1' = O_1$ and $c_1 = c_2$. Since $c_1 = c_2$, we obtain from Lemma B.28 that $N_2' = O_2$. Hence, $M_1 = M_2$ and we can conclude.
(2) $M_2 = O_1 \diamond O_2$, with $N_1 \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} O_1$ and $N_2 \xrightarrow{c_2} O_2$. By Lemma B.29 (applied twice, to N_1 and N_2, which we can do because, like before, $\gamma_1 = A(\ell_1)$ and $\gamma_2 = A(\ell_2)$), we obtain $N_1' \overset{G_2:(?\ell_2,\gamma_2)}{\rightsquigarrow} Q$ and $O_1 \xrightarrow{c_1} Q$, and $N_2' \xrightarrow{c_2} Q'$ and $O_2 \overset{G_1:(?\ell_1,\gamma_1)}{\rightsquigarrow} Q'$. Hence, $N_1' \diamond N_2' \xrightarrow{c_2} Q \diamond Q'$ and $O_1 \diamond O_2 \xrightarrow{c_1} Q \diamond Q'$, which concludes the proof. □

Lemma B.31. *If $\mathrm{dM}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M' \twoheadrightarrow N \not\xrightarrow{c}$ for all c and $M' \xrightarrow{c} M''$ for some c, then $M'' \twoheadrightarrow N$.*

Proof. We proceed by induction on the length of $M' \twoheadrightarrow N$. The base case is for $M' = N$: as N cannot send messages, we then also have $M' \not\xrightarrow{c}$, and the desired result trivially holds. In the inductive step, we consider $M' \xrightarrow{c'} M_1 \twoheadrightarrow N$. Let $c = \ell : (!G, \gamma)$ and $c' = \ell' : (!G', \gamma')$; by Lemma B.30, either there exists N' such that $M_1 \xrightarrow{c} N'$ and $M'' \xrightarrow{c'} N'$, or $M'' = M_1$. In the latter case we can conclude immediately, because $M_1 \twoheadrightarrow N$. In the former case we use the induction hypothesis on $\mathrm{dM}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M_1 \twoheadrightarrow N$ and $M_1 \xrightarrow{c} N'$ to obtain that $N' \twoheadrightarrow N$ from which we can conclude because $M' \xrightarrow{c} M'' \xrightarrow{c'} N'$. □

Lemma B.32. *If* $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M' \twoheadrightarrow N \overset{c}{\not\rightarrow}$ *for all* c *and* $M' \twoheadrightarrow M''$*, then* $M'' \twoheadrightarrow N$.

Proof. We proceed by induction on the length of $M' \twoheadrightarrow M''$. If the length is 0, then $M' = M''$ and we are done immediately. Otherwise we consider $M' \xrightarrow{c} M_1 \twoheadrightarrow M''$ with $c = \ell : (!G, \gamma)$; by Lemma B.31, $M_1 \twoheadrightarrow N$. From the induction hypothesis on $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M_1 \twoheadrightarrow N$ and $M_1 \twoheadrightarrow M''$, we obtain that $M'' \twoheadrightarrow N$ and conclude. □

The next three lemmas are used to prove Lemmas B.44, B.45, B.56, and B.60. They describe 'correct' communication paths of local monitors, where 'correct' means that the local monitors only take receiving transitions that correspond to messages containing the last action of the sender. Lemma B.34 shows that in correct communication paths, the order of communication does not matter for the final destination (when no sending and receiving can take place). Lemma B.35 is a direct generalization of Lemma B.34, and is the equivalent of Lemma B.32 on the level of local monitors. Lemma B.36 is used to show that those correct communication paths can be combined when multiple local monitors are put in parallel, which is used in Lemmas B.56 and B.60.

We first define formally what is such a correct communication path.

Definition B.33. Let $m \Rightarrow_A m'$ denote the existence of an integer $h > 0$, of h monitors $m_1, \ldots, m_h$ and of $h-1$ communication actions $c_1, \ldots, c_{h-1}$ such that $m_1 = m$, $m_h = m'$, $m_i \xrightarrow{c_i} m_{i+1}$ for every $i = 1, \ldots, h-1$ and, if $c_i = (?\ell, \gamma)$, then $\gamma = A(\ell)$.

Hence, $\Rightarrow_A$ is the same as $\twoheadrightarrow$ but with the extra constraint on actions of kind $(?\ell, \gamma)$.

Lemma B.34. *Let* m *be a relevant local monitor such that* $m \Rightarrow_A m' \overset{c}{\not\rightarrow}$*, for all* c*, and either* $m \xrightarrow{(?\ell', A(\ell'))} n$*, for some* $\ell' \in \mathcal{L}$*, or* $m \xrightarrow{(!G,\gamma)} n$*, for some* $G \subseteq \mathcal{L}$*. Then* $n \Rightarrow_A m'$.

Proof. Let $m \Rightarrow_A m'$ in h steps; we proceed by induction on h. In the base case, $m = m'$ and, as m' cannot communicate, the premise of the lemma is not satisfied. Now suppose that $m \xrightarrow{c_1} m_1 \Rightarrow_A m'$, with $c_1 = (!G, \gamma)$ or $c_1 = (?\ell'', A(\ell''))$ for some $\ell'' \in \mathcal{L}$. First, let us consider the case in which $m \xrightarrow{(?\ell',A(\ell'))} n$, for some $\ell' \in \mathcal{L}$; depending on the form of c_1, we have two subcases:

(1) $c_1 = (!G, \gamma)$. We use Lemma B.20 to obtain that $m_1 \xrightarrow{(?\ell',A(\ell'))} n'$ and $n \xrightarrow{c_1} n'$. Hence, $m_1 \Rightarrow_A m'$ in $h-1$ steps, and $m_1 \xrightarrow{(?\ell',A(\ell'))} n'$. We apply the induction hypothesis to obtain that $n' \Rightarrow_A m'$, from which we can conclude because $n \xrightarrow{c_1} n'$.

(2) $c_1 = (?\ell'', A(\ell''))$ for some $\ell'' \in \mathcal{L}$. Via Lemma B.22, we get that either $m_1 \xrightarrow{(?\ell',A(\ell'))} n'$ and $n \xrightarrow{c_1} n'$ (in which case we proceed as before), or $\ell' = \ell''$. In the latter case, we use Lemma B.21 to derive that $m_1 = n$ and conclude, because $m_1 \Rightarrow_A m'$.

Then, we consider the case in which $m \xrightarrow{(!G,\gamma)} n$, for some $G \subseteq \mathcal{L}$, and we again distinguish two subcases:

(1) $c_1 = (!G', \gamma')$. If $G \neq G'$ or $\gamma \neq \gamma'$, then we use Lemma B.23 to obtain that $m_1 \xrightarrow{(!G,\gamma)} n'$ and $n \xrightarrow{c_1} n'$ and proceed like in the first subcase above. If $G = G'$ and $\gamma = \gamma'$, we obtain from Lemma B.24 that $m_1 = n$ and proceed like the second subcase above.

(2) $c_1 = (?\ell'', A(\ell''))$ for some $\ell'' \in \mathcal{L}$. Via Lemma B.20, we get that $m_1 \xrightarrow{(!G,\gamma)} n'$ and $n \xrightarrow{c_1} n'$; we then proceed as in the first subcase above. □

We can extend the previous lemma to a more general statement:

Lemma B.35. *Let m be a relevant local monitor such that $m \Rightarrow_A m' \not\xrightarrow{c}$, for all c, and $m \Rightarrow_A n$. Then $n \Rightarrow_A m'$.*

Lemma B.36. *Let m_1^1 and m_1^2 be relevant local monitors such that $m_1^1 \Rightarrow_A n_1 \not\xrightarrow{c}$ and $m_1^2 \Rightarrow_A n_2 \not\xrightarrow{c}$, for all c. Then, $m_1^1 \odot m_1^2 \Rightarrow_A n_1 \odot n_2$, for $\odot \in \{\otimes, \oplus\}$.*

The following lemma shows that, for every receiving transition in a monitor m that is reached from a monitor synthesized from some $\psi \in \mathsf{Qf}$, there exists a corresponding sending transition in another local monitor reached from a synthesized local monitor for ψ located at a different location.

Lemma B.37. *Let $\mathrm{DM}_\sigma^\ell(\psi) \xrightarrow{A(\ell)} m \rightarrow m_1$ and $m_1 \xrightarrow{(?\ell',\gamma)} m'$. Then $\ell' \in \mathsf{rng}(\sigma)$, $\ell' \neq \ell$, and there are n, n' and G' such that $\mathrm{DM}_\sigma^{\ell'}(\psi) \xrightarrow{A(\ell')} n \xrightarrow{(!G',A(\ell'))} n'$ and $\ell \in G'$.*

Proof. We proceed by induction on $\xrightarrow{A(\ell)}$. If $\mathrm{DM}_\sigma^\ell(\psi) = v$, the premise of the lemma is not satisfied; moreover, we do not consider the case where $\mathrm{DM}_\sigma^\ell(\psi) = a.m$, as this cannot occur in the synthesis function (because of our assumption that $|\mathsf{Act}| \geq 2$).

For the first inductive case, we consider $\mathrm{DM}_\sigma^\ell(\psi) = n_1 + n_2 \xrightarrow{A(\ell)} m$, because $n_1 \xrightarrow{A(\ell)} m$ or $n_2 \xrightarrow{A(\ell)} m$. Hence, $\psi = [a_\pi]\psi'$ or $\psi = \langle a_\pi \rangle \psi'$. As $m_1 \xrightarrow{(?\ell',\gamma)} m'$, we know that $\sigma(\pi) = \ell' \neq \ell$. Thus, from the synthesis function, we immediately have that $\ell' \in \mathsf{rng}(\sigma)$ and $\mathrm{DM}_\sigma^{\ell'}(\psi) \xrightarrow{A(\ell')} n$ such that $n \xrightarrow{(!G',A(\ell'))} n'$ with $\ell \in G'$ (being $\ell \in \mathsf{rng}(\sigma)$).

Next we consider $\mathrm{DM}_\sigma^\ell(\psi) = n_1 \odot n_2 \xrightarrow{A(\ell)} n_1' \odot n_2' (= m) \rightarrow n_1'' \odot n_2'' (= m_1)$, with $n_1 \xrightarrow{A(\ell)} n_1'$, $n_2 \xrightarrow{A(\ell)} n_2'$, $n_1' \rightarrow n_1''$, and $n_2' \rightarrow n_2''$. From the synthesis function, we know that $\psi = \psi_1 \diamond \psi_2$ (with $\diamond = \wedge$ if $\odot = \otimes$ and $\diamond = \vee$ if $\odot = \oplus$), $n_1 = \mathrm{DM}_\sigma^\ell(\psi_1)$ and $n_2 = \mathrm{DM}_\sigma^\ell(\psi_2)$. The transition $m_1 = n_1'' \odot n_2'' \xrightarrow{(?\ell',\gamma)} m'$ can happen in two ways:

(1) $m' = m_1' \odot m_2'$, for $n_1'' \xrightarrow{(?\ell',\gamma)} m_1'$ and $n_2'' \xrightarrow{(?\ell',\gamma)} m_2'$. From the induction hypothesis, we get that $\ell' \in \mathsf{rng}(\sigma)$, $\mathrm{DM}_\sigma^{\ell'}(\psi_1) \xrightarrow{A(\ell')} o_1 \xrightarrow{(!G_1',A(\ell'))} o_1'$ and $\ell \in G_1'$; similarly, $\mathrm{DM}_\sigma^{\ell'}(\psi_2) \xrightarrow{A(\ell')} o_2 \xrightarrow{(!G_2',A(\ell'))} o_2'$ and $\ell \in G_2'$. Hence, $\mathrm{DM}_\sigma^{\ell'}(\psi_1 \diamond \psi_2) = \mathrm{DM}_\sigma^{\ell'}(\psi_1) \odot \mathrm{DM}_\sigma^{\ell'}(\psi_2) \xrightarrow{A(\ell')} o_1 \odot o_2 \xrightarrow{(!G_1',A(\ell'))} o_1' \odot o_2$ with $\ell \in G_1'$, as desired.

(2) $m' = m_1' \odot n_2''$, for $n_1'' \xrightarrow{(?\ell',\gamma)} m_1'$ and $n_2'' \not\xrightarrow{(?\ell',\gamma)}$ (the symmetric case is identical). From the induction hypothesis, we get that $\ell' \in \mathsf{rng}(\sigma)$, $\mathrm{DM}_\sigma^{\ell'}(\psi_1) \xrightarrow{A(\ell')} o \xrightarrow{(!G',A(\ell'))} o'$ and $\ell \in G'$. It is immediate from the synthesis function that $\mathrm{DM}_\sigma^{\ell'}(\psi_2) \xrightarrow{A(\ell')} o_2$, for some monitor o_2. Thus, $\mathrm{DM}_\sigma^{\ell'}(\psi_1 \diamond \psi_2) = \mathrm{DM}_\sigma^{\ell'}(\psi_1) \odot \mathrm{DM}_\sigma^{\ell'}(\psi_2) \xrightarrow{A(\ell')} o \odot o_2 \xrightarrow{(!G',A(\ell'))} o' \odot o_2$ with $\ell \in G'$, as desired.

Last, we consider $\mathrm{DM}_\sigma^\ell(\psi) = \mathsf{rec}\, x.o \xrightarrow{A(\ell)} m$ because $o\{^{\mathsf{rec}\, x.o}/_x\} \xrightarrow{A(\ell)} m$. Thus, $\psi = \max x.\psi'$ and $o = \mathrm{DM}_\sigma^\ell(\psi')$. By Lemma B.1, $\mathrm{DM}_\sigma^\ell(\psi')\{^{\mathsf{rec}\, x.\mathrm{DM}_\sigma^\ell(\psi')}/_x\} = \mathrm{DM}_\sigma^\ell(\psi'\{^{\max x.\psi'}/_x\})$; from induction hypothesis, we obtain that $\ell' \in \mathsf{rng}(\sigma)$, $\mathrm{DM}_\sigma^{\ell'}(\psi'\{^{\max x.\psi'}/_x\}) \xrightarrow{A(\ell')} n \xrightarrow{(!G',A(\ell'))} n'$ and $\ell \in G'$. Since $\mathrm{DM}_\sigma^{\ell'}(\psi)\{^{\mathsf{rec}\, x.\mathrm{DM}_\sigma^{\ell'}(\psi)}/_x\} = \mathrm{DM}_\sigma^{\ell'}(\psi'\{^{\max x.\psi'}/_x\})$, we get that $\mathrm{DM}_\sigma^{\ell'}(\max x.\psi') = \mathsf{rec}\, x.\mathrm{DM}_\sigma^{\ell'}(\psi) \xrightarrow{A(\ell')} n$, which concludes the proof. □

Lemma B.40 is used to prove Lemma B.41 and it states that a communicating monitor never makes a choice between a send and a receive, so if M_1 can send (with some transition labeled with c) and reach M_2, it cannot have reached M_2 by receiving some message (intuitively because the send from c is still present if M_1 takes a receiving transition, and it is not present in M_2). Instead, if a monitor can both send and receive, this must be because multiple monitors are connected in parallel, either locally with $\oplus/\otimes$ or as communicating monitors via $\vee/\wedge$.

To this aim, we first define a function cc from relevant (local) monitors to natural numbers that counts the number of top-level send prefixes and prove that it decreases after every send action and remains the same after every receive action.

Definition B.38. For arbitrary σ, ℓ and ψ, we define:

- $cc(\textsc{dm}^{\ell}_{\sigma}(\psi)) = cc(\sum_{a\in\mathsf{Act}}(?\{\ell'\}, a).\textsc{dm}^{\ell}_{\sigma}(\psi_a)) = 0$;
- $cc((!G, a).\textsc{dm}^{\ell}_{\sigma}(\psi)) = 1$;
- $cc(m_1 \odot m_2) = cc(m_1) + cc(m_2)$, where $\odot$ denotes either $\oplus$ or $\otimes$;
- $cc([m]_{\ell}) = cc(m)$; and
- $cc(M \diamond N) = cc(M) + cc(N)$, where $\diamond$ denotes either $\vee$ or $\wedge$.

Lemma B.39. *Let m_1 be a relevant local monitor.*

(1) *If $m_1 \xrightarrow{(!G,\gamma)} m_2$, then $cc(m_1) > cc(m_2)$.*
(2) *If $m_1 \xrightarrow{(?G,\gamma)} m_3$, then $cc(m_1) = cc(m_3)$.*

Proof. For the first claim, we proceed by induction on m_1. In the base cases for $m_1 = \textsc{dm}^{\ell}_{\sigma}(\psi)$ and $m_1 = \sum_{a\in\mathsf{Act}}(?\{\ell'\}, a).\textsc{dm}^{\ell}_{\sigma}(\psi_a)$, we have that $m_1 \xrightarrow{(!G,\gamma)} m_2$ does not hold (by using Lemma B.2 for $m_1 = \textsc{dm}^{\ell}_{\sigma}(\psi)$); for $m_1 = (!G, \gamma).\textsc{dm}^{\ell}_{\sigma}(\psi)$, we have that $m_2 = \textsc{dm}^{\ell}_{\sigma}(\psi)$ and, thus, $cc(m_2) = 0 < 1 = cc((!G, \gamma).\textsc{dm}^{\ell}_{\sigma}(\psi))$. For the inductive case, let $m_1 = m \odot m'$, with m and m' relevant monitors; the result follows immediately from the definition of cc and the inductive hypothesis.

For the second claim, we again proceed by induction on m_1. The base cases for $m_1 = \textsc{dm}^{\ell}_{\sigma}(\psi)$ and $m_1 = (!G, \gamma).\textsc{dm}^{\ell}_{\sigma}(\psi)$ entail that $m_1 \xrightarrow{(?G,\gamma)} m_3$ does not hold; for $m_1 = \sum_{a\in\mathsf{Act}}(?\{\ell'\}, a).\textsc{dm}^{\ell}_{\sigma}(\psi_a)$, then $m_3 = \textsc{dm}^{\ell}_{\sigma}(\psi_a)$, for some ψ_a, and therefore $cc(m_1) = cc(m_3) = 0$. The case for $m_1 = m \odot m'$ is straightforward from the inductive hypothesis. □

Lemma B.40. *Let m_1 and M_1 be action-derived monitors.*

(1) *If $m_1 \xrightarrow{(!G,\gamma)} m_2$ and $m_1 \xrightarrow{(?G,\gamma)} m_3$, then $m_2 \neq m_3$.*
(2) *If $M_1 \xrightarrow{\ell:(!G,\gamma)} M_2$ and $M_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} M_3$, then $M_2 \neq M_3$.*

Proof. Notice that m_1 is a relevant local monitor by Lemma B.6. For the first claim, we use Lemma B.39 to obtain that $cc(m_3) = cc(m_1) < cc(m_2)$ and so $m_2 \neq m_3$. The second statement is proved similarly, by observing that $M_1 \xrightarrow{\ell:(!G,\gamma)} M_2$ implies $cc(M_1) > cc(M_2)$ and that $M_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} M_3$ implies $cc(M_1) = cc(M_3)$ (both claims can be proved by a straightforward induction, by using Lemma B.39). □

From the previous lemma we can then conclude that, if M_1 reaches M_2 by a sending transition, and we combine M_1 with another monitor N_1 that transitions to M_2 composed with N_2, then N_1 must have taken a receiving step. This is useful for Lemmas B.44 and B.50, which are in turn used to prove Lemmas B.55, B.56, and B.61.

Lemma B.41. *Let M_1 be action-derived, $M_1 \xrightarrow{\ell:(!G,\gamma)} M_2$ and $M_1 \diamond N_1 \xrightarrow{\ell:(!G,\gamma)} M_2 \diamond N_2$. Then, $N_1 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$.*

The next lemma states that action transitions are deterministic for synthesized monitors, and is used in the proof of Lemma B.44.

Lemma B.42. *If $\mathrm{DM}^{\ell}_{\sigma}(\psi) \xrightarrow{A(\ell)} m$ and $\mathrm{DM}^{\ell}_{\sigma}(\psi) \xrightarrow{A(\ell)} n$, then $m = n$.*

Proof. We proceed by induction on $\xrightarrow{A(\ell)}$. The interesting cases are the inductive ones. If $\mathrm{DM}^{\ell}_{\sigma}(\psi) = m_1 + m_2 \xrightarrow{A(\ell)} m$, then either $\psi = [a_\pi]\psi'$ or $\psi = \langle a_\pi \rangle \psi'$, for some a, π and ψ'. As there is only one possible transition for each $a \in \mathsf{Act}$, we have obtained the required result.

If $\mathrm{DM}^{\ell}_{\sigma}(\psi) = \mathsf{rec}\ x.m_1 \xrightarrow{A(\ell)} m$ because $m_1\{^{\mathsf{rec}\ x.m_1}/_x\} \xrightarrow{A(\ell)} m$, we know that $\psi = \max x.\psi'$ and $m_1 = \mathrm{DM}^{\ell}_{\sigma}(\psi')$. By Lemma B.1, $m_1\{^{\mathsf{rec}\ x.m_1}/_x\} = \mathrm{DM}^{\ell}_{\sigma}(\psi'\{^{\psi}/_x\})$ (note that the free location variables of ψ are not bound in ψ'); thus, we can use the induction hypothesis to get $m = n$.

If $\mathrm{DM}^{\ell}_{\sigma}(\psi) = m_1 \odot m_2 \xrightarrow{A(\ell)} m_3 \odot m_4 = m$, we know from the synthesis definition that $\psi = \psi_1 \diamond \psi_2$ (for $\diamond = \vee$ if $\odot = \oplus$ and $\diamond = \wedge$ if $\odot = \otimes$), for some ψ_1 and ψ_2 such that $m_1 = \mathrm{DM}^{\ell}_{\sigma}(\psi_1)$ and $m_2 = \mathrm{DM}^{\ell}_{\sigma}(\psi_2)$. Hence, $\mathrm{DM}^{\ell}_{\sigma}(\psi_1) \xrightarrow{A(\ell)} m_3$ and $\mathrm{DM}^{\ell}_{\sigma}(\psi_2) \xrightarrow{A(\ell)} m_4$. Similarly, we know that $n = n_1 \odot n_2$ for some n_1 and n_2 such that $\mathrm{DM}^{\ell}_{\sigma}(\psi_1) \xrightarrow{A(\ell)} n_1$ and $\mathrm{DM}^{\ell}_{\sigma}(\psi_2) \xrightarrow{A(\ell)} n_2$. We use the induction hypothesis to conclude that $m_3 = n_1$ and $m_4 = n_2$, from which we derive that $m = n$. □

In what follows we need to specify a condition on a group of local monitors who all take some 'correct' set of communication transitions (correct as in following the restrictions in $\Rightarrow_A$). This condition states that the reached states in the group of local monitors will no longer take receiving transitions that match a sending transition that occurred on one of the local communication paths.

Definition B.43. Let $\ell_1, \ldots, \ell_k \in \mathcal{L}$ and $m_1, \ldots, m_k, n_1, \ldots, n_k$ be monitors. We write $m_i \underset{A}{\overset{\ell_i}{\Rightarrow}}{}^{k}_{i=1} n_i$ if, for every $i \in \{1, \ldots, k\}$, there exist an integer $h_i > 0$, h_i monitors $m_i^1, \ldots, m_i^{h_i}$, and $h_i - 1$ actions $c_i^1, \ldots, c_i^{h_i-1}$ such that $m_i = m_i^1$, $m_i^{h_i} = n_i$, $m_i^j \xrightarrow{c_i^j} m_i^{j+1}$ for every $j = 1, \ldots, h_i - 1$ and:

(1) if $c_i^j = (?\ell, \gamma)$ for some γ and ℓ, then $\gamma = A(\ell)$;

(2) if $c_i^j = (!G, \gamma)$ for some G and γ, then, for every $i' \neq i$, $\ell_{i'} \in G$ implies that $n_{i'} \overset{(?\ell_i,\gamma)}{\not\longrightarrow}$.

Essentially, $m_i \underset{A}{\overset{\ell_i}{\Rightarrow}}{}^{k}_{i=1} n_i$ amounts to saying that, for every $i \leq k$, $m_i \Rightarrow_A n_i$ plus condition 2.

The next lemma is used in the proof of Lemma B.45. It proves that local monitors that 'accidentally' receive messages from another local monitor's communication transitions are such that they cannot take any more receiving transitions matching those accidental messages. This captures the intuition that a local monitor always takes all receiving transitions that match a message at once, and no new receiving transitions appear for a local monitor when it is taking a chain of consecutive communication transitions.

Important in the proof is Lemma B.37: in $\Rightarrow_A$ local monitors can send and receive freely, but in $\rightarrow$ between decentralized monitors, communication is always initiated by a sender. Therefore, all receiving transitions that need to happen need to be initiated by some sending transition.

Lemma B.44. *Let* $\mathsf{rng}(\sigma) = \{\ell_1, \cdots, \ell_k\} \neq \emptyset$ *and, for every* $i \in \{1, \ldots, k\}$, $\textsc{dm}^{\ell_i}_{\sigma}(\psi) \xrightarrow{A(\ell_i)} m_i \Rightarrow_A n_i \Rightarrow_A o_i$, *with* o_i *that cannot communicate. If* $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$, *then for every* $j \in \{1, \cdots, k\}$ *there exist some* $q_1, \ldots, q_k$ *such that* $\bigvee_{i=1}^{k} [n_i]_{\ell_i} \rightarrow \bigvee_{i=1}^{k} [q_i]_{\ell_i}$, $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$, *and* $q_j = o_j$.

Proof. Fix any $j \in \{1, \ldots, k\}$ and let $n_j \Rightarrow_A o_j$ in h_j steps; we proceed by induction on h_j. In the base case, we have that $n_j = o_j$ and so it suffices to consider $q_i = n_i$, for every i.

For the inductive step, let $n_j \xrightarrow{c} n'_j \Rightarrow_A o_j$ with $c = (!G, \gamma)$ or $c = (?\ell, A(\ell))$ for some $\ell \in \mathcal{L}$:

(1) Case $c = (!G, \gamma)$: By Lemma B.18, we know that $\gamma = A(\ell_j)$; moreover, $\bigvee_{i=1}^{k} [n_i]_{\ell_i} \xrightarrow{\ell_j : (!G,\gamma)} \bigvee_{i=1}^{k} [n'_i]_{\ell_i}$, where $\bigvee_{i \in \{1..k\} \setminus \{j\}} [n_i]_{\ell_i} \overset{G:(?\ell_j,\gamma)}{\rightsquigarrow} \bigvee_{i \in \{1..k\} \setminus \{j\}} [n'_i]_{\ell_i}$ by Lemma B.41.
We now argue that $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$. For each $\ell_i \in \mathsf{rng}(\sigma)$, we have that $m_i \Rightarrow_A n_i$, say in g_i steps. By Lemma B.25, we obtain that either $n_i \xrightarrow{(?\ell_j, A(\ell_j))} n'_i$ or $n_i = n'_i$, for all $i \neq j$; thus, $m_i \Rightarrow_A n'_i$ for each $\ell_i \in \mathsf{rng}(\sigma)$ in $f_i = g_i + 1$ or $f_i = g_i$ steps, respectively. Furthermore, if $f_i = g_i + 1$, then $c_i^{g_i+1} = (?\ell_j, A(\ell_j))$, for all $i \neq j$, and $c_j^{g_j+1} = c$. For the second condition of $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$, let $i \neq j$ and $m_i \Rightarrow_A n'_i$ in f_i steps; let $f \leq f_i$ and $c_i^f = (!G', \gamma')$. Now take some $\ell_h \in G' \cap \mathsf{rng}(\sigma)$ with $h \neq i$, and we have to show that $n'_h \not\xrightarrow{(?\ell_i,\gamma')}$. If $f \leq g_i$, then $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$ implies that $n_h \not\xrightarrow{(?\ell_i,\gamma')}$ and, by Lemma B.18, we can conclude that $n'_h \not\xrightarrow{(?\ell_i,\gamma')}$. If $f = g_i + 1$, then we have reasoned above that $i = j$, $G' = G$ and $\gamma' = \gamma$. Furthermore, we know that either $n_h \xrightarrow{(?\ell_i,\gamma)} n'_h$ or $n'_h = n_h \not\xrightarrow{(?\ell_i,\gamma)}$; in both cases, Lemma B.18 entails that $n'_h \not\xrightarrow{(?\ell_i,\gamma)}$ and so $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$.
By Lemma B.34, for all $\ell_i \in \mathsf{rng}(\sigma)$, we have that $\textsc{dm}^{\ell_i}_{\sigma}(\psi) \xrightarrow{A(\ell_i)} m_i \Rightarrow_A n'_i \Rightarrow_A o_i$. Since $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$, we can apply the induction hypothesis to obtain that $\bigvee_{i=1}^{k} [n'_i]_{\ell_i} \rightarrow \bigvee_{i=1}^{k} [q_i]_{\ell_i}$, for some q_is such that $n'_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}_{k_{i=1} q_i}$ and $q_j = o_j$. It remains to show that $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$: $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$ and $n'_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}_{k_{i=1} q_i}$ immediately yield the first condition for $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$ and they yield the second condition by using Lemma B.18.

(2) Case $c = (?\ell, A(\ell))$: By Lemma B.37, $\ell \in \mathsf{rng}(\sigma)$ (say, $\ell = \ell_h$), $j \neq h$, and $\textsc{dm}^{\ell_h}_{\sigma}(\psi) \xrightarrow{A(\ell_h)} p_h \xrightarrow{(!G', A(\ell_h))} p'_h$, for some $G' \subseteq \mathcal{L}$ such that $\ell_j \in G'$. From Lemma B.42, $p_h = m_h$. Since $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$, then in particular $m_h \Rightarrow_A n_h$ with the same transitions; by Lemmas B.20 and B.23, $n_h \xrightarrow{(!G', A(\ell_h))} n'_h$, if $m_h \Rightarrow_A n_h$ does not include any transition labeled by $(!G', A(\ell_h))$. If such a transition does exist, from $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$, we get that $n_i \not\xrightarrow{(?\ell_h, A(\ell_h))}$, for all $i \neq h$ and $\ell_i \in \mathsf{rng}(\sigma) \cap G'$; but this would contradict $n_j \xrightarrow{c} n'_j$, since $\ell_j \in G'$. Hence, we have $n_h \xrightarrow{(!G', A(\ell_h))} n'_h$ and so $\bigvee_{i=1}^{k} [n_i]_{\ell_i} \xrightarrow{\ell_h : (!G', A(\ell_h))} \bigvee_{i=1}^{k} [n'_i]_{\ell_i}$, where, by Lemma B.41, $\bigvee_{i \in \{1..k\} \setminus \{h\}} [n_i]_{\ell_i} \overset{G':(?\ell_h, A(\ell_h))}{\rightsquigarrow} \bigvee_{i \in \{1..k\} \setminus \{h\}} [n'_i]_{\ell_i}$. We can now argue that $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n'_i$ as in case 1 above. By Lemma B.35, $n'_i \Rightarrow_A q_i$ for all $i \in \{1, \ldots, k\}$ and, by the induction hypothesis,

$\bigvee_{i=1}^{k}[n'_i]_{\ell_i} \to \bigvee_{i=1}^{k}[q_i]_{\ell_i}$ with $n'_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}_{k_{i=1}q_i}$ and $q_j = o_j$. We can then conclude, by arguing that $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$ as in case 1. □

The next lemma is important for Lemmas B.56 and B.61 (part of the lemmas for Formula Convergence). It is actually stronger than what is needed for Lemma B.56, as therein we shall just need that $\bigvee_{i=1}^{k}[n_i]_{\ell_i}$ transitions to some monitor that cannot comunicate, and not exactly to $\bigvee_{i=1}^{k}[o_i]_{\ell_i}$. However, the latter is needed for Lemma B.61, so we present the stronger statement already here. The lemma shows that local monitors can be combined and, even if their communications 'accidentally' influence one another, the same final state (where no communication can take place) is reached. It also shows another important property: in $\Rightarrow_A$ local monitors can send and receive freely, but in $\to$ between decentralized monitors communication is always initiated by a sender. This lemma captures that all the receiving transitions a local monitor can do before it reaches a state that cannot communicate are triggered by sending actions present in other local monitors located at locations found in the range of σ.

Lemma B.45. *Let* $\mathsf{rng}(\sigma) = \{\ell_1, \dots, \ell_k\}$ *and, for all* $i \in \{1, \dots, k\}$, $\mathrm{DM}_{\sigma}^{\ell_i}(\psi) \xrightarrow{A(\ell_i)} m_i \Rightarrow_A n_i \Rightarrow_A o_i$, *with* o_i *that cannot communicate. If* $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$, *then* $\bigvee_{i=1}^{k}[n_i]_{\ell_i} \to \bigvee_{i=1}^{k}[o_i]_{\ell_i}$.

Proof. We proceed by induction on k. If $k = 0$, the result is trivial. In the inductive step, by Lemma B.44, there exist $q_1, \dots, q_k$ and $j \in \{1, \dots, k\}$ such that $\bigvee_{i=1}^{k}[n_i]_{\ell_i} \to \bigvee_{i=1}^{k}[q_i]_{\ell_i}$, $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$, and $q_j = o_j$. For all $i \neq j$, we have $\mathrm{DM}_{\sigma}^{\ell_i}(\psi) \xrightarrow{A(\ell_i)} m_i \Rightarrow_A n_i \Rightarrow_A o_i$ and $n_i \Rightarrow_A q_i$; by Lemma B.35, $q_i \Rightarrow_A o_i$. From this, we derive that $\mathrm{DM}_{\sigma}^{\ell_i}(\psi) \xrightarrow{A(\ell_i)} m_i \Rightarrow_A q_i \Rightarrow_A o_i$, for all $i \in \{1, \dots, k\}$.

We now show that $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$. For each $i \in \{1, \dots, k\}$, we have $m_i \Rightarrow_A n_i$ (say, in f_i steps) and $n_i \Rightarrow_A q_i$ (say, in g_i steps); thus, $m_i \Rightarrow_A q_i$ in $h_i = f_i + g_i$ steps. For the second condition of $m_i \overset{A}{\underset{i}{\Rightarrow}}{}_{k_{i=1}q_i}$, take any i, any $h \leq h_i$ (with $c_i^h = (!G, \gamma)$) and any $\ell_{i'} \in G \cap \mathsf{rng}(\sigma)$ with $i' \neq i$, and we have to show that $q_{i'} \overset{(?\ell_i,\gamma)}{\not\longrightarrow}$. If $h \leq f_i$, then $m_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} n_i$ entails that $n_{i'} \overset{(?\ell_i,\gamma)}{\not\longrightarrow}$; by Lemma B.18, $q_{i'} \overset{(?i\ell,\gamma)}{\not\longrightarrow}$. On the other hand, if $h > f_i$, the result follows from $n_i \overset{\ell_i}{\underset{A}{\Rightarrow}}{}^{k}_{i=1} q_i$.

Finally, the induction hypothesis entails that $\bigvee_{i \in \{1..k\} \setminus \{j\}}[q_i]_{\ell_i} \to \bigvee_{i \in \{1..k\} \setminus \{j\}}[o_i]_{\ell_i}$. Hence, $\bigvee_{i \in \{1..k\}}[n_i]_{\ell_i} \to \bigvee_{i \in \{1..k\} \setminus \{j\}}[q_i]_{\ell_i} \vee [o_j]_{\ell_j} \to \bigvee_{i \in \{1..k\}}[o_i]_{\ell_i}$. In the last step we use that o_j cannot communicate and therefore does not change anymore. □

Another important feature that we need for Bounded Communication and Formula Convergence is the following: if different monitors influence each other via communication when combined (so a decentralized monitor is combined with another monitor and all of a sudden a local monitor in the decentralized system receives messages that it did not receive before), this makes no difference on the states the monitors reach when no more communication can take place.

We first set-up a definition that helps us with two things. First, it allows us to describe what a decentralized monitor M looks like if it is combined via $\vee$ or $\wedge$ with another decentralized monitor, and then transitions according to a communication path of that other monitor. In other words, M will change state when the other monitor sends messages that local monitors in M can receive, and we formally capture what M looks like after these changes. Second, this definition will allow us to prove that influences from another monitor captured via Definition B.46 do not affect the final outcome (when no more communication takes place). These two elements then lead to Lemma B.55, where the different monitors in a join will send messages that 'accidentally' reach the others and it is shown this is not a problem. Lemma B.55 is used to prove Lemma B.56 (Bounded Communication) and Lemma B.65 (Formula Convergence).

To this end, we define a set L of monitors that can all take the same receiving transition. Definition B.46 defines this set properly, and Lemma B.47 proves three general facts about this set.

Definition B.46. Let $H \subseteq \mathcal{L}$ and $M \in \mathsf{DMon}$. Let $L = \{[m_1]_{\ell_1}, \dots, [m_k]_{\ell_k}\}$. We say that L is a *set of recipients for* $H : (?\ell', \gamma)$ if, for all $j \in \{1, \dots, k\}$:

(1) $\ell_j \in H$, and
(2) $m_j \xrightarrow{(?\ell',\gamma)} n_j$, for some $n_j \in \mathsf{LMon}$.

We call a set L an *M-cover* for $H : (?\ell', \gamma)$ if:

(1) L is a set of recipients for $H : (?\ell', \gamma)$,
(2) every $[m]_\ell \in L$ is action-derived, and
(3) if $[m]_\ell \in M$, $\ell \in H$ and $m \rightarrow m' \xrightarrow{(?\ell',\gamma)} n$, then $[m']_\ell \in L$.

Lemma B.47. *Let $H \subseteq \mathcal{L}$, M be action-derived, and $L = \{[m_1]_{\ell_1}, \dots, [m_k]_{\ell_k}\}$ be an M-cover for $H : (?\ell, \gamma)$.*

(1) *If $M = [m]_{\ell'} \notin L$, and $M \xrightarrow{\ell':(!G,\gamma)} N$ or $M \overset{G:(?\ell'',\gamma)}{\rightsquigarrow} N$, then $N = [n]_{\ell'} \notin L$.*
(2) *If $M \overset{G:(?\ell',\gamma)}{\rightsquigarrow} N$, then L is also an N-cover for $H : (?\ell, \gamma)$.*
(3) *If $M \xrightarrow{\ell':(!G,\gamma)} N$, then L is also an N-cover for $H : (?\ell, \gamma)$.*

Proof. For the first item, we observe that $N = [n]_{\ell'}$, where:

— $m = n$; or
— $m \xrightarrow{(?\ell'',\gamma)} n$, or;
— $m \xrightarrow{(!G,\gamma)} n$.

In the first case, we have $[n]_{\ell'} = [m]_{\ell'} \notin L$. In the other two cases, because L is an M-cover, either $\ell' \notin H$, and so $[n]_{\ell'} \notin L$, or $m \overset{(?\ell,\gamma)}{\not\longrightarrow}$, and by Lemma B.18 this implies that $n \overset{(?\ell,\gamma)}{\not\longrightarrow}$ and again $[n]_{\ell'} \notin L$.

For the second item, we proceed by induction on $\overset{G:(?\ell',\gamma)}{\rightsquigarrow}$. For the base case, $M = [m]_{\ell''}$, $N = [n]_{\ell''}$, $\ell'' \in G$, and $m = n$ or $m \xrightarrow{(?\ell',\gamma)} n$. Suppose that $n \rightarrow n_1 \xrightarrow{(?\ell,\gamma)} n_2$ and $\ell'' \in H$. Then, we also know that $m \rightarrow n_1 \xrightarrow{(?\ell,\gamma)} n_2$; as $[m]_{\ell''} = M$, because L is an M-cover, we obtain that $[n_1]_{\ell''} \in L$, yielding that L is an N-cover. For the inductive case, $M = M_1 \diamond M_2$, $N = N_1 \diamond N_2$, $M_1 \overset{G:(?\ell',\gamma)}{\rightsquigarrow} N_1$ and $M_2 \overset{G:(?\ell',\gamma)}{\rightsquigarrow} N_2$. Since L is an M-cover, it is also an M_1-cover and an M_2-cover. From the induction hypothesis, L is an N_1-cover and an N_2-cover, and thus an N-cover.

For the third item, we proceed by induction on $\xrightarrow{\ell':(!G,\gamma)}$. For the base case, $M = [m]_{\ell'}$, $N = [n]_{\ell'}$, and $m \xrightarrow{(!G,\gamma)} n$. Suppose that $n \rightarrow n_1 \xrightarrow{(?\ell,\gamma)} n_2$. Then, we also know that $m \rightarrow n_1 \xrightarrow{(?\ell,\gamma)} n_2$; as $[m]_{\ell'} = M$, because L is an M-cover, we obtain that $[n_1]_{\ell'} \in L$, yielding that L is an N-cover. For the inductive case, $M = M_1 \diamond M_2$, $N = N_1 \diamond N_2$, $M_1 \xrightarrow{\ell':(!G,\gamma)} N_1$ and $M_2 \overset{G:(?\ell',\gamma)}{\rightsquigarrow} N_2$. Since L is an M-cover, it is also an M_1-cover and an M_2-cover. From the induction hypothesis and the previous item, L is an N_1-cover and an N_2-cover, and thus an N-cover. □

In what follows it is useful to describe the state of a monitor M after receiving the message corresponding to L. To do this formally, we observe the following:

Lemma B.48. *Let L be an M-cover for $H : (?\ell, \gamma)$. For every $[m]_{\ell'} \in L$, if $m \xrightarrow{(?\ell,\gamma)} m_1$ and $m \xrightarrow{(?\ell,\gamma)} m_2$, then $m_1 = m_2$ and $[m_1]_\ell \notin L$.*

With this lemma, we can then define what M after L is:

Definition B.49. Let $L = \{[m_1]_{\ell_1}, \dots, [m_k]_{\ell_k}\}$ be an M-cover for $H : c$ and, for each $i \leq k$, let n_i be the unique monitor (by Lemma B.48) such that $m_i \xrightarrow{c} n_i$. We define $M[L] = M[[n_1]_{\ell_1}/[m_1]_{\ell_1}] \dots [[n_k]_{\ell_k}/[m_k]_{\ell_k}]$.

We can now use Definitions B.46 and B.49 and Lemma B.47 to describe a monitor that 'accidentally' receives messages from another monitor's communication transitions.

Lemma B.50. *Let M_1 and M_2 be A-derived and such that $M_1^1 \xrightarrow{c_1} M_1^2 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} M_1^n$ and $M_1^1 \diamond M_2^1 \xrightarrow{c_1} M_1^2 \diamond M_2^2 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} M_1^n \diamond M_2^n$, where $M_1^1 = M_1, M_2^1 = M_2, n > 0$, and $c_i = \ell_i : (!G_i, A(\ell_i))$, for every $i \in \{1, \dots, n-1\}$. Then, for every $i \in \{1, \dots, n-1\}$, there exists an M_2-cover L_i for $G_i : (?\ell_i, A(\ell_i))$, such that $M_2[L_1] \dots [L_i] = M_2^{i+1}$.*

Proof. We proceed by induction on n. If $n = 1$, then the lemma holds vacuously. Otherwise, we have:

$$M_1^1 \diamond M_2^1 \xrightarrow{c_1} M_1^2 \diamond M_2^2 \xrightarrow{c_2} \dots \xrightarrow{c_{n-2}} M_1^{n-1} \diamond M_2^{n-1} \xrightarrow{c_{n-1}} M_1^n \diamond M_2^n.$$

The induction hypothesis tells us that there exist $L_1, \dots, L_{n-2}$ such that, for each $i \in \{1, \dots, n-2\}$, L_i is an M_2-cover for $G_i : (?\ell_i, A(\ell_i))$, and $M_2[L_1] \cdots [L_i] = M_2^{i+1}$. Let:

$$L_{n-1} = \{[m']_\ell \mid \ell \in G_{n-1} \text{ and } \exists\, [m]_\ell \in M_2.\ m \twoheadrightarrow m' \xrightarrow{(?\ell_{n-1}, A(\ell_{n-1}))} m''\}.$$

Then, L_{n-1} is an M_2-cover for $G_{n-1} : (?\ell_{n-1}, A(\ell_{n-1}))$ by Definition B.46, and also an M_2^i-cover for $G_{n-1} : (?\ell_{n-1}, A(\ell_{n-1}))$ by Lemmas B.41 and B.47.

What is left to show is that $M_2[L_1] \cdots [L_{n-2}][L_{n-1}] = M_2^n$. We can conclude this from the induction hypothesis, if we show that $M_2^{n-1}[L_{n-1}] = M_2^n$. By Lemma B.41, $M_2^{n-1} \overset{G_{n-1}:(\ell_{n-1},\gamma_{n-1})}{\rightsquigarrow} M_2^n$. Let $\{[m_i]_{\ell_i} \mid 1 \leq i \leq k\} = \{[m]_\ell \in M_2^{n-1}\}$. For each $1 \leq i \leq k$, let m'_i be m_i, when $\ell_i \notin G_{n-1}$ or $m_i \not\xrightarrow{(?\ell_{n-1},\gamma_{n-1})}$, or be such that $m_i \xrightarrow{(?\ell_{n-1},\gamma_{n-1})} m'_i$, otherwise. Then, $M[L_{n-1}] = M_2^{n-1}[[m'_1]_{\ell_1}/[m_1]_{\ell_1}] \cdots [[m'_k]_{\ell_k}/[m_k]_{\ell_k}] = M_2^n$, by Lemma B.25. □

Lemmas B.51 and B.52 are used to prove Lemma B.53, and they state some basic facts of behavior of $M[L]$ w.r.t. the behavior of M.

Lemma B.51. *Let $H \subseteq \mathcal{L}$, M be A-derived, and L be an M-cover for $H : (?\ell', A(\ell'))$. If $M \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N$, for some $\ell'' \in \mathcal{L}$, then $M[L] \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N[L]$.*

Proof. We proceed by induction on $\overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow}$. In the first base case, $M = [m]_\ell$, $N = [n]_\ell$, $\ell \in G$ and $m \xrightarrow{(?\ell'',A(\ell''))} n$. If $[m]_\ell \notin L$, then $[n]_\ell \notin L$ via Lemma B.47. Hence, $M = M[L]$ and

$N = N[L]$, which concludes the proof. Otherwise, $M[L] = [m_1]_\ell$ with $m \xrightarrow{(?\ell',A(\ell'))} m_1$ and $\ell \in H$. We distinguish two cases:

(1) $\ell'' \neq \ell'$. We then apply Lemma B.22 on m: $n \xrightarrow{(?\ell',A(\ell'))} n_1$ and $m_1 \xrightarrow{(?\ell'',A(\ell''))} n_1$. From this, we derive that $M[L] = [m_1]_\ell \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} [n_1]_\ell$. Since L is an M-cover, it is also an N-cover via Lemma B.47. Thus, from $n \xrightarrow{(?\ell',A(\ell'))} n_1$ we obtain that $N = [n]_\ell \in L$. Hence, from $n \xrightarrow{(?\ell',A(\ell'))} n_1$ and via Lemma B.48, we get $N[L] = [n_1]_\ell$ and $M[L] = [m_1]_\ell \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} [n_1]_\ell = N[L]$.

(2) $\ell'' = \ell'$. Then, $A(\ell'') = A(\ell')$, and, via Lemma B.21, we obtain that $m_1 = n$; so, $M[L] = [m_1]_\ell = [n]_\ell = N$. From $m \xrightarrow{(?\ell',A(\ell'))} m_1$ and Lemma B.18, we conclude that $m_1 \not\xrightarrow{(?\ell',A(\ell'))}$. Since $\ell'' = \ell'$ and $m_1 = n$, it holds that $n \not\xrightarrow{(?\ell'',A(\ell''))}$. and $n \not\xrightarrow{(?\ell',A(\ell'))}$, which implies that $[n]_\ell \notin L$, and therefore $N = N[L]$. From $n \not\xrightarrow{(?\ell'',A(\ell''))}$ we derive that $M[L] = [n]_\ell \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} [n]_\ell = N = N[L]$, which is what we wanted to show.

In the second base case, we consider $M = [m]_\ell$, $N = [m]_\ell$, and $m \not\xrightarrow{(?\ell'',A(\ell''))}$. If $M = M[L]$, then $[m]_\ell \notin L$ (by Lemma B.48), and thus also $N = N[L]$, which concludes the proof. Otherwise, we have that $M[L] = [m_1]_\ell$ with $m \xrightarrow{(?\ell',A(\ell'))} m_1$ and $\ell \in H$. We know that $M = N$, so $M[L] = N[L]$. We know that $m \not\xrightarrow{(?\ell'',A(\ell''))}$ and, via Lemma B.18, that $m_1 \not\xrightarrow{(?\ell'',A(\ell''))}$. We can conclude that $M[L] = [m_1]_\ell \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} [m_1]_\ell = N[L]$.

In the third base case, we consider $M = [m]_\ell$, $N = [m]_\ell$ and $\ell \notin G$. If $M = M[L]$, then $[m]_\ell \notin L$, and thus also $N = N[L]$, which concludes the proof. Otherwise, we have that $M[L] = [m_1]_\ell$ with $m \xrightarrow{(?\ell',A(\ell'))} m_1$ and $\ell \in H$. We know that $M = N$, so $M[L] = N[L]$. Since $\ell \notin G$, we can conclude that $M[L] = [m_1]_\ell \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} [m_1]_\ell = N[L]$.

For the inductive step, $M = M_1 \diamond M_2$, $N = N_1 \diamond N_2$, $M_1 \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N_1$, and $M_2 \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N_2$. Clearly, M_1 and M_2 are A-derived and L is an M_1-cover and an M_2-cover; thus, we can apply the induction hypothesis to obtain that $M_1[L] \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N_1[L]$, and $M_2[L] \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N_2[L]$. From this, we derive $M[L] = M_1[L] \diamond M_2[L] \overset{G:(?\ell'',A(\ell''))}{\rightsquigarrow} N_1[L] \diamond N_2[L] = N[L]$. □

Lemma B.52. *Let $H \subseteq \mathcal{L}$, M be A-derived and L be an M-cover for $H : (?\ell', A(\ell'))$. If $M \xrightarrow{\ell:(!G,\gamma)} N$, then $M[L] \xrightarrow{\ell:(!G,\gamma)} N[L]$.*

Proof. The proof proceeds by induction on $\xrightarrow{\ell:(!G,\gamma)}$. For the base case, $M = [m]_\ell$, $N = [n]_\ell$ and $m \xrightarrow{(!G,\gamma)} n$. If $M = M[L]$, then also $N = N[L]$ (by Lemma B.47) and the proof is done. Otherwise, we must have that $M[L] = [m]_\ell[[m_1]_\ell/[m]_\ell] = [m_1]_\ell$ for $m \xrightarrow{(?\ell',A(\ell'))} m_1$ and $\ell \in H$. By Lemma B.20, $n \xrightarrow{(?\ell',A(\ell'))} n_1$ and $m_1 \xrightarrow{(!G,\gamma)} n_1$. Since L is an M-cover, we know it is also an N-cover (Lemma B.47) and therefore $[n]_\ell \in L$. Then we obtain from $n \xrightarrow{(?\ell',A(\ell'))} n_1$ and Lemma B.48 that $N[L] = [n_1]_\ell$; hence, $M[L] = [m_1]_\ell \xrightarrow{\ell:(!G,\gamma)} [n_1]_\ell = N[L]$.

For the inductive step, $M = M_1 \diamond M_2$, $N = N_1 \diamond N_2$, $M_1 \xrightarrow{\ell:(!G,\gamma)} N_1$, and $M_2 \overset{G:(?\ell,\gamma)}{\rightsquigarrow} N_2$. Clearly, M_1 and M_2 are A-derived, and L is an M_1-cover and an M_2-cover; thus, we can apply the induction hypothesis to obtain that $M_1[L] \xrightarrow{\ell:(!G,\gamma)} N_1[L]$. By Lemma B.19, $\gamma = A(\ell)$. Since $M_2 \overset{G:(?\ell,A(\ell))}{\rightsquigarrow} N_2$, we can apply Lemma B.51 to get $M_2[L] \overset{G:(?\ell,A(\ell))}{\rightsquigarrow} N_2[L]$. Thus $M[L] = M_1[L] \diamond M_2[L] \xrightarrow{\ell:(!G,\gamma)} N_1[L] \diamond N_2[L] = N[L]$. □

The next lemma shows that $M_i[L]$ is the state of M_i after receiving the message corresponding to L. This can be understood as M_i receiving messages from some other decentralized monitor (as what happens in Lemma B.55). So, even if M_i receives at any point some messages from another decentralized monitor, the same state is reached when no more communication can take place.

Lemma B.53. *Let $H \subseteq \mathcal{L}$, M_1 be action-derived, and $M_1 \xrightarrow{c_1} M_2 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} M_n \xrightarrow{c_n} N$, where $n \geq 1$ and N cannot communicate. If $s \leq n$ and L is an M_s-cover for $H : (?\ell, A(\ell))$, then, for all $i \in \{s, \dots, n\}$, it holds that $M_i[L] \xrightarrow{c_i} M_{i+1}[L] \xrightarrow{c_{i+1}} \dots \xrightarrow{c_{n-1}} M_n[L] \xrightarrow{c_n} N$.*

We can generalize the previous result to include a union of L's. Notationally, if $L = L_1 \cup \dots \cup L_k$, we denote $M[L_1]\dots[L_k]$ as $M[L]$. This lemma is a cornerstone for proving Lemma B.55 (together with Lemma B.50).

Lemma B.54. *Let $H \subseteq \mathcal{L}$, M_1 be action-derived, and $M_1 \xrightarrow{c_1} M_2 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} M_n \xrightarrow{c_n} N$, where $n \geq 1$ and N cannot communicate. Let $s \leq n$ and $L = \bigcup_{j=1}^{k} L_j$ such that L_j is an M_s-cover for $H_j : (?\ell_j, A(\ell_i))$, for all j. Then, for all $i \in \{s, \dots, n\}$, it holds that $M_i[L] \xrightarrow{c_i} M_{i+1}[L] \xrightarrow{c_{i+1}} \dots \xrightarrow{c_{n-1}} M_n[L] \xrightarrow{c_n} N$.*

Proof. We proceed by induction on k. If $k = 0$, the result is trivial. In the inductive step, we obtain from the induction hypothesis that:

$$M_i[L_1]\dots[L_{k-1}] \xrightarrow{c_i} M_{i+1}[L_1]\dots[L_{k-1}] \xrightarrow{c_{i+1}} \dots \xrightarrow{c_{n-1}} M_n[L_1]\dots[L_{k-1}] \xrightarrow{c_n} N.$$

If we show that L_k is a cover for $M_i[L_1]\dots[L_{k-1}]$, the result follows immediately from Lemma B.53. Take $[m]_\ell \in M_i[L_1]\dots[L_{k-1}]$ with $\ell \in H_k$ such that $m \longrightarrow m' \xrightarrow{(?\ell^k, A(\ell^k))} n$. Hence, we know that there exists $[m_1]_\ell \in M_i$ such that $m_1 \longrightarrow m$, and thus also $m_1 \longrightarrow m'$. As L_k is an M_s-cover, it is also an M_i-cover via Lemma B.47. Hence, we obtain that $[m']_\ell \in L_k$, as desired. □

The next lemma is actually stronger than what is needed for Bounded Communication (for Bounded Communication it is sufficient that $\Diamond_{\ell \in \mathcal{L}} M_\ell$ in the lemma below can transition to some decentralized monitor that cannot communicate, and it does not exactly need to be $\Diamond_{\ell \in \mathcal{L}} N_\ell$). However, for Formula Convergence we need the lemma in its strong version, so we prove only that one.

Lemma B.55. *Let $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ and, for all $i \in \{1, \dots, k\}$, we have $\mathrm{DM}_{\sigma_i}(\varphi_i) \xrightarrow{A} M_i \longrightarrow N_i$, where N_i cannot communicate. Then, $\Diamond_{i=1}^{k} M_i \longrightarrow \Diamond_{i=1}^{k} N_i$.*

Proof. By induction on k. If $k = 1$, we are done immediately. If $\mathcal{L} = \{\ell_1, \dots \ell_{k+1}\}$, let $M_1 \xrightarrow{c_1} M_1^2 \xrightarrow{c_2} \dots \xrightarrow{c_n} N_1$ for $n \geq 0$; by Lemma B.19, $c_j = \ell_j : (!G_j, A(\ell_j))$, for all $j \in \{1, \dots, n\}$. By Lemma B.50, there exist $L_1, \dots, L_n$ such that, for each $j \in \{1, \dots, n\}$, L_j is a $(\Diamond_{i=2}^{k} M_i)$-cover for $G_j : (?\ell_j, A(\ell_j))$. Let:

$$N = \bigDiamond_{\ell=2}^{k} M_i[L_1]\dots[L_n];$$

then, $\Diamond_{i=1}^{k} M_i = M_1 \diamond (\Diamond_{i=2}^{k} M_i) \xrightarrow{c_1} \dots \xrightarrow{c_n} N_1 \diamond N$. From the induction hypothesis, we know that $\Diamond_{i=2}^{k} M_i \longrightarrow \Diamond_{i=2}^{k} N_i$. Thus, we can apply Lemma B.54 to conclude that $N \longrightarrow \Diamond_{i=2}^{k} N_i$. From $\Diamond_{i=1}^{k} M_i \longrightarrow N_1 \diamond_N$ we can conclude, since N_1 cannot take any communication steps and therefore does not change anymore. □

We can now finally prove Bounded Communication:

Lemma B.56. *For every* $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M \twoheadrightarrow M'$, *there exists* M'' *such that* $M' \twoheadrightarrow M''$ *and* M'' *cannot communicate.*

Proof. We proceed by induction on φ. The base case is for $\varphi \in \mathsf{Qf}$ and we have two possible subcases. If $\sigma = \emptyset$, then $\mathrm{DM}_\sigma(\varphi) = [v]_{\ell_0} \xrightarrow{A} [v]_{\ell_0} = M = M'$ and $[v]_{\ell_0}$ cannot communicate, since $v \overset{c}{\not\rightarrow}$, for all c. If $\sigma \neq \emptyset$, then $\mathrm{DM}_\sigma(\varphi) = \bigvee_{\ell \in \mathsf{rng}(\sigma)} [\mathrm{Dm}^\ell_\sigma(\varphi)]_\ell \xrightarrow{A} \bigvee_{\ell \in \mathsf{rng}(\sigma)} [m_\ell]_\ell = M$ and $M \twoheadrightarrow \bigvee_{\ell \in \mathsf{rng}(\sigma)} [n_\ell]_\ell = M'$, where $m_\ell \twoheadrightarrow n_\ell$. Actually, we have that $m_\ell \Rightarrow_A n_\ell$ because we know from the semantics that, for any receive $\xrightarrow{(?\ell',\gamma)}$ in the path $m_\ell \twoheadrightarrow n_\ell$, there exists ℓ' such that $m_{\ell'} \twoheadrightarrow m'_{\ell'} \xrightarrow{(!G,\gamma)}$ (with $\ell \in G$) and, by Lemma B.18, that $\gamma = A(\ell')$.

As n_ℓ is a relevant local monitor, we prove by induction on the structure of relevant local monitors that there exists n'_ℓ such that $n_\ell \Rightarrow_A n'_\ell \overset{c}{\not\rightarrow}$, for all c. The first two base cases ($n_\ell = \mathrm{Dm}^\ell_\sigma(\psi)$ and $n_\ell = (!G, a).\mathrm{Dm}^\ell_\sigma(\psi)$) are trivial because of Lemma B.2. For the third base case, $n_\ell = \sum_{a \in \mathsf{Act}} (?\{\ell'\}, a).\mathrm{Dm}^\ell_\sigma(\psi_a)$, we observe that there is some $a \in \mathsf{Act}$ such that $a = A(\ell')$; so, we take that transition and conclude again by Lemma B.2. For the inductive case, $n_\ell = n_1 \odot n_2$ and, from the inductive hypothesis, $n_1 \Rightarrow_A m_1$ and $n_2 \Rightarrow_A m_2$ such that $m_1 \overset{c}{\not\rightarrow}$ and $m_2 \overset{c}{\not\rightarrow}$, for all c. By Lemma B.36, $n_1 \odot n_2 \Rightarrow_A m_1 \odot m_2 \overset{c}{\not\rightarrow}$, for all c.

Hence, $n_\ell \Rightarrow_A n'_\ell$ such that $n'_\ell \overset{c}{\not\rightarrow}$, for all c. From this, we derive that $\mathrm{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \Rightarrow_A n'_\ell$, for all $\ell \in \mathsf{rng}(\sigma)$. We can then apply Lemma B.45 (with $m_\ell = n_\ell$ in the statement of the lemma) to conclude that $\bigvee_{\ell \in \mathsf{rng}(\sigma)} [m_\ell]_\ell \twoheadrightarrow \bigvee_{\ell \in \mathsf{rng}(\sigma)} [n'_\ell]_\ell$. It is immediate that $\bigvee_{\ell \in \mathsf{rng}(\sigma)} [n'_\ell]_\ell \overset{c}{\not\rightarrow}$, for all c. We apply Lemma B.32 to derive that $M' \twoheadrightarrow \bigvee_{\ell \in \mathsf{rng}(\sigma)} [n'_\ell]_\ell$, as desired.

We only treat one of the inductive cases, the others are similar. Let $\varphi = \exists \pi.\varphi'$. Then $\mathrm{DM}_\sigma(\varphi) = \bigvee_{\ell \in \mathcal{L}} \mathrm{DM}_{\sigma[\pi \mapsto \ell]}(\varphi') \xrightarrow{A} \bigvee_{\ell \in \mathcal{L}} M_\ell = M$. From the induction hypothesis we obtain that $M_\ell \twoheadrightarrow N_\ell$ such that N_ℓ cannot communicate. We apply Lemma B.55 and conclude that $M \twoheadrightarrow \bigvee_{\ell \in \mathcal{L}} N_\ell$. Then we can use Lemma B.32 to derive that $M' \twoheadrightarrow \bigvee_{\ell \in \mathcal{L}} N_\ell$, which concludes the proof. □

B.5 Proofs for Processing-Communication Alternation of $\mathrm{DM}_{\_}(\cdot)$

We showed the first item of Processing-Communication Alternation in Lemma B.2. For the second item, we actually prove the contrapositive, because this simplifies the inductive proofs; we first show the result for local monitors and then we conclude by an induction based on Lemma B.57.

Lemma B.57. *Let* $\mathrm{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m' \twoheadrightarrow m \xrightarrow{A'(\ell)} n$. *Then* $m \overset{c}{\not\rightarrow}$, *for all* c.

Proof. We focus on relevant local monitors, because m is a relevant local monitor via Lemma B.6. We proceed by induction on m. In the base case, for $m = \mathrm{Dm}^\ell_\sigma(\psi)$ the result follows from Lemma B.2; for $m = (!G, a).\mathrm{Dm}^\ell_\sigma(\psi)$ and $m = \sum_{a \in \mathsf{Act}} (?\{\ell''\}, a).\mathrm{Dm}^\ell_\sigma(\psi_a)$, the premise of the lemma is not satisfied.

For the inductive step, let $m = m_1 \odot m_2$ with m_1 and m_2 relevant monitors. We derive from $m_1 \odot m_2 \xrightarrow{A'(\ell)} n$ that $n = n_1 \odot n_2$, $m_1 \xrightarrow{A'(\ell)} n_1$ and $m_2 \xrightarrow{A'(\ell)} n_2$. Hence, we can apply the induction hypothesis to obtain that $m_1 \overset{c}{\not\rightarrow}$ and $m_2 \overset{c}{\not\rightarrow}$, from which we conclude. □

Lemma B.58. *Let* $M \in \mathsf{DMon}$ *be action-derived. If* $M \xrightarrow{A} N$, *then* $M \overset{c}{\not\rightarrow}$, *for all* c.

B.6 Proofs for Formula Convergence of $\textsc{dM}_{-}(\cdot)$

We start with a lemma that captures a basic equality between synthesized centralized and decentralized local monitors, used in the proofs of Lemmas B.60 and B.63.

Lemma B.59. *If $\mathsf{cm}_\sigma(\psi) = \mathsf{cm}_{\sigma'}(\psi')$, for either $\sigma = \sigma'$ or $|\mathcal{L}| = 1$, then $\textsc{dm}^\ell_\sigma(\psi) = \textsc{dm}^\ell_{\sigma'}(\psi')$, for all $\ell \in \mathcal{L}$.*

Proof. We proceed by induction on ψ. As a first base case, we consider $\psi = \mathsf{tt}$: we have $\mathsf{cm}_\sigma(\psi) = \mathsf{yes} = \mathsf{cm}_{\sigma'}(\psi')$ and, by Lemma B.7, we conclude that, for all $\ell \in \mathcal{L}$, we have $\textsc{dm}^\ell_\sigma(\psi) = \mathsf{yes} = \textsc{dm}^\ell_{\sigma'}(\psi')$. The base cases for $\psi = \mathsf{ff}$, $\psi = (\pi = \pi')$ and $\psi = (\pi \neq \pi')$ are similar. The last base case is for $\psi = x$: then, $\mathsf{cm}_\sigma(\psi) = x = \mathsf{cm}_{\sigma'}(\psi')$, from which we can conclude that $\psi' = x$. Immediately it follows that, for all $\ell \in \mathcal{L}$, $\textsc{dm}^\ell_\sigma(\psi) = x = \textsc{dm}^\ell_{\sigma'}(\psi')$.

We now proceed with the inductive cases. We first consider $\psi = [a_\pi]\psi_1$ (the case for $\psi = \langle a_\pi \rangle \psi_1$ is the same, with no in place of yes); thus $\mathsf{cm}_\sigma(\psi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\psi_1) + \sum_{b \neq a} b_{\sigma(\pi)}.\mathsf{yes} = \mathsf{cm}_{\sigma'}(\psi')$ and so $\psi' = [a_{\pi'}]\psi_2$, $\sigma(\pi) = \sigma'(\pi')$ and $\mathsf{cm}_\sigma(\psi_1) = \mathsf{cm}_{\sigma'}(\psi_2)$. We use the induction hypothesis to obtain that, for all $\ell \in \mathcal{L}$, we have $\textsc{dm}^\ell_\sigma(\psi_1) = \textsc{dm}^\ell_{\sigma'}(\psi_2)$. We have two possibilities:

(1) $\sigma(\pi)\ (= \sigma'(\pi')) = \ell$. In this case, $\textsc{dm}^\ell_\sigma(\psi) = a.(!(\mathsf{rng}(\sigma) \setminus \{\ell\}), a).\textsc{dm}^\ell_\sigma(\psi_1) + \sum_{b \neq a} b.(!(\mathsf{rng}(\sigma) \setminus \{\ell\}), b).\mathsf{yes}$ and $\textsc{dm}^\ell_{\sigma'}(\psi') = a.(!(\mathsf{rng}(\sigma') \setminus \{\ell\}), a).\textsc{dm}^\ell_{\sigma'}(\psi_2) + \sum_{b \neq a} b.(!(\mathsf{rng}(\sigma') \setminus \{\ell\}), b).\mathsf{yes}$. Using the induction hypothesis, we obtain the required result, since $\mathsf{rng}(\sigma) \setminus \{\ell\} = \mathsf{rng}(\sigma') \setminus \{\ell\}$, both if $\sigma = \sigma'$ and if $|\mathcal{L}| = 1$ (and so $\mathcal{L} = \{\ell\}$).

(2) $\sigma(\pi)\ (= \sigma'(\pi')) \neq \ell$ (notice that this case is not possible if $|\mathcal{L}| = 1$). Then, $\textsc{dm}^\ell_\sigma(\psi) = \sum_{b \in \mathsf{Act}} b.\big(?(\{\sigma(\pi)\}, a).\textsc{dm}^\ell_\sigma(\psi_1) + \sum_{b \neq a}(?(\{\sigma(\pi)\}), b).\mathsf{yes}\big)$ and $\textsc{dm}^\ell_{\sigma'}(\psi') = \sum_{b \in \mathsf{Act}} b.\big(?(\{\sigma'(\pi)\}, a).\textsc{dm}^\ell_{\sigma'}(\psi_2) + \sum_{b \neq a}(?(\{\sigma'(\pi')\}), b).\mathsf{yes}\big)$. Using the induction hypothesis, we obtain the required result.

Next we consider $\psi = \max x.\psi_1$. Hence, $\mathsf{cm}_\sigma(\psi) = \mathsf{rec}\ x.\mathsf{cm}_\sigma(\psi_1) = \mathsf{cm}_{\sigma'}(\psi')$. From this we conclude that $\psi' = \max x.\psi_2$ and $\mathsf{cm}_\sigma(\psi_1) = \mathsf{cm}_{\sigma'}(\psi_2)$. We use the induction hypothesis to obtain that, for all $\ell \in \mathcal{L}$, we have $\textsc{dm}^\ell_\sigma(\psi_1) = \textsc{dm}^\ell_{\sigma'}(\psi_2)$; thus, for all $\ell \in \mathcal{L}$, $\textsc{dm}^\ell_\sigma(\max x.\psi_1) = \mathsf{rec}\ x.\textsc{dm}^\ell_\sigma(\psi_1) = \mathsf{rec}\ x.\textsc{dm}^\ell_{\sigma'}(\psi_2) = \textsc{dm}^\ell_{\sigma'}(\max x.\psi_2) = \textsc{dm}^\ell_{\sigma'}(\psi')$.

Finally, let $\psi = \psi_1 \diamond \psi_2$. Hence, $\mathsf{cm}_\sigma(\psi) = \mathsf{cm}_\sigma(\psi_1) \odot \mathsf{cm}_\sigma(\psi_2) = \mathsf{cm}_{\sigma'}(\psi')$ (for $\odot = \otimes$ if $\diamond = \wedge$ and $\odot = \oplus$ if $\diamond = \vee$)), with $\psi' = \psi_1' \diamond \psi_2'$, $\mathsf{cm}_\sigma(\psi_1) = \mathsf{cm}_{\sigma'}(\psi_1')$ and $\mathsf{cm}_\sigma(\psi_2) = \mathsf{cm}_{\sigma'}(\psi_2')$. The result immediately follows from the induction hypothesis. □

To prove Formula Convergence for quantifier-free formulas, we need the following result. Informally, it captures part of the weak bisimulation at the level of local monitors.

Lemma B.60. *Let $\psi \in \mathsf{Qf}$. If $\mathsf{cm}_\sigma(\psi) \xrightarrow{A} \mathsf{cm}_\sigma(\psi')$, then $\textsc{dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \Rightarrow_A \textsc{dm}^\ell_\sigma(\psi')$, for all $\ell \in \mathcal{L}$.*

Proof. We proceed by induction on $\xrightarrow{A}$ for central monitors. The base case with $\mathsf{cm}_\sigma(\psi) = a_\ell.m$ never applies, because of the assumption that $|\mathsf{Act}| \geq 2$. Hence, the only base case is for $\mathsf{cm}_\sigma(\psi) = v = \mathsf{cm}_\sigma(\psi')$; then, Lemma B.7 entails that $\textsc{dm}^\ell_\sigma(\psi) = v = \textsc{dm}^\ell_\sigma(\psi')$, for all $\ell \in \mathcal{L}$, and we can conclude, as $v \xrightarrow{A(\ell)} v \Rightarrow_A v$ for any $A(\ell)$.

For the inductive case, let $\mathsf{cm}_\sigma(\psi) = m + n \xrightarrow{A} \mathsf{cm}_\sigma(\psi')$; hence, either $\psi = [a_\pi]\psi''$ or $\psi = \langle a_\pi \rangle \psi''$, and $\mathsf{cm}_\sigma(\psi) = a_{\sigma(\pi)}.\mathsf{cm}_\sigma(\psi'') + \sum_{b \neq a} b_{\sigma(\pi)}.v$, where $v = \mathsf{yes}$, in case of box, and $v = \mathsf{no}$, in case of diamond. We distinguish two cases.

(1) $\sigma(\pi) = \ell'$ and $A(\ell') = a$; this entails that $\mathsf{cm}_\sigma(\psi') = \mathsf{cm}_\sigma(\psi'')$. Fix $\ell \in \mathcal{L}$. If $\ell = \ell'$, we have $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} (!(\mathsf{rng}(\sigma) \setminus \{\ell\}, a).\mathsf{Dm}^\ell_\sigma(\psi'')$; if $\ell \neq \ell'$, we have $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} (?\{\ell'\}, a).\mathsf{Dm}^\ell_\sigma(\psi'') + \sum_{b \neq a}(?\{\ell'\}, b).v$. Thus, in both situations we can have that $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \xrightarrow{c} \mathsf{Dm}^\ell_\sigma(\psi'')$, where $\gamma = A(\ell')$ $(= a)$ whenever $c = (?\{\ell'\}, \gamma)$. From Lemma B.59 and $\mathsf{cm}_\sigma(\psi') = \mathsf{cm}_\sigma(\psi'')$, we conclude $\mathsf{Dm}^\ell_\sigma(\psi') = \mathsf{Dm}^\ell_\sigma(\psi'')$, for all $\ell \in \mathcal{L}$.

(2) $\sigma(\pi) = \ell'$, $A(\ell') = b$, and $a \neq b$; this entails that $\mathsf{cm}_\sigma(\psi') = v$. We use Lemma B.7 to conclude that $\mathsf{Dm}^\ell_\sigma(\psi') = v$, for all ℓ. Fix $\ell \in \mathcal{L}$. If $\ell = \ell'$, we have $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} (!(\mathsf{rng}(\sigma) \setminus \{\ell\}), b).v = (!(\mathsf{rng}(\sigma) \setminus \{\ell\}, b).\mathsf{Dm}^\ell_\sigma(\psi')$; if $\ell \neq \ell'$, we have $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} (?\{\ell'\}, a).\mathsf{Dm}^\ell_\sigma(\psi'') + \sum_{b \neq a}(?\{\ell'\}, b).v = (?\{\ell'\}, a).\mathsf{Dm}^\ell_\sigma(\psi'') + \sum_{b \neq a}(?\{\ell'\}, b).\mathsf{Dm}^\ell_\sigma(\psi')$. Thus in both situations we can conclude that $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \xrightarrow{c} \mathsf{Dm}^\ell_\sigma(\psi')$, where $\gamma = A(\ell')$ whenever $c = (?\{\ell'\}, \gamma)$.

Next we consider the recursive case: $\mathsf{cm}_\sigma(\psi) = \mathsf{rec}\ x.m \xrightarrow{A} \mathsf{cm}_\sigma(\psi')$ because $m\{^{\mathsf{rec}\ x.m}/_x\} \xrightarrow{A} \mathsf{cm}_\sigma(\psi')$. Thus we obtain that $\psi = \max x.\psi_1$ and $m = \mathsf{cm}_\sigma(\psi_1)$. As $\mathsf{cm}_\sigma(\psi_1)\{^{\mathsf{rec}\ x.\mathsf{cm}_\sigma(\psi_1)}/_x\} = \mathsf{cm}_\sigma(\psi_1\{^{\max x.\psi_1}/_x\})$ (Lemma A.7), we use the induction hypothesis to obtain that, for all ℓ, we have $\mathsf{Dm}^\ell_\sigma(\psi_1\{^{\max x.\psi_1}/_x\}) \xrightarrow{A(\ell)} m_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi')$. Using that $\mathsf{Dm}^\ell_\sigma(\psi_1)\{^{\mathsf{rec}\ x.\mathsf{Dm}^\ell_\sigma(\psi_1)}/_x\} = \mathsf{Dm}^\ell_\sigma(\psi_1\{^{\max x.\psi_1}/_x\})$ (Lemma B.1), we get that $\mathsf{Dm}^\ell_\sigma(\psi_1)\{^{\mathsf{rec}\ x.\mathsf{Dm}^\ell_\sigma(\psi_1)}/_x\} \xrightarrow{A(\ell)} m_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi')$. This implies that $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi')$.

Last, let $\mathsf{cm}_\sigma(\psi) = m \odot n \xrightarrow{A} m' \odot n'$ because $m \xrightarrow{A} m'$ and $n \xrightarrow{A} n'$. As ψ is quantifier-free, we get that $\psi = \psi_1 \diamond \psi_2$ (for $\diamond = \wedge$ if $\odot = \otimes$ and $\diamond = \vee$ if $\odot = \oplus$), $m = \mathsf{cm}_\sigma(\psi_1)$ and $n = \mathsf{cm}_\sigma(\psi_2)$. Also, $\psi' = \psi_1' \diamond \psi_2'$, $m' = \mathsf{cm}_\sigma(\psi_1')$ and $n' = \mathsf{cm}_\sigma(\psi_2')$. From the induction hypothesis we obtain that, for all ℓ, we have $\mathsf{Dm}^\ell_\sigma(\psi_1) \xrightarrow{A(\ell)} m^1_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi_1')$ and $\mathsf{Dm}^\ell_\sigma(\psi_2) \xrightarrow{A(\ell)} m^2_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi_2')$. Thus, for all ℓ, we have that $\mathsf{Dm}^\ell_\sigma(\psi_1 \diamond \psi_2) \xrightarrow{A(\ell)} m^1_\ell \odot m^2_\ell$ and, via Lemma B.36 (that can be used thanks to Lemma B.2), we conclude the desired $m^1_\ell \odot m^2_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi_1') \odot \mathsf{Dm}^\ell_\sigma(\psi_2')$. □

We can now prove Formula Convergence for $\psi \in \mathsf{Qf}$:

Lemma B.61. *Let $\psi \in \mathsf{Qf}$. If $\mathsf{DM}_\sigma(\psi) \xrightarrow{A} M \twoheadrightarrow M'$, with M' that cannot communicate, and $\mathsf{cm}_\sigma(\psi) \xrightarrow{A} \mathsf{cm}_\sigma(\psi')$, for some formula ψ', then $M' = \mathsf{DM}_\sigma(\psi')$.*

Proof. First, since $\mathsf{DM}_\sigma(\psi) \xrightarrow{A} M$, we have that $\mathsf{fv}_{\mathsf{I}}(\psi) \neq \emptyset$ and so $\sigma \neq \emptyset$. Then, Lemma B.60 tells us that $\mathsf{Dm}^\ell_\sigma(\psi) \xrightarrow{A(\ell)} m_\ell \Rightarrow_A \mathsf{Dm}^\ell_\sigma(\psi')$, for all $\ell \in \mathcal{L}$. Thus, we know that $\mathsf{DM}_\sigma(\psi) = \bigvee_{\ell \in \mathsf{rng}(\sigma)}[\mathsf{Dm}^\ell_\sigma(\psi)]_\ell \xrightarrow{A} \bigvee_{\ell \in \mathsf{rng}(\sigma)}[m_\ell]_\ell$; because $\mathsf{Dm}^\ell_\sigma(\psi') \stackrel{c}{\not\to}$ for all c (Lemma B.2), we can use Lemma B.45 (by taking $n_\ell = m_\ell$ in the statement of that result) to obtain that $\bigvee_{\ell \in \mathsf{rng}(\sigma)}[m_\ell]_\ell \twoheadrightarrow \bigvee_{\ell \in \mathsf{rng}(\sigma)}[\mathsf{Dm}^\ell_\sigma(\psi')]_\ell = \mathsf{DM}_\sigma(\psi')$ and $\mathsf{DM}_\sigma(\psi')$ cannot communicate. Hence, we apply Lemma B.32 to derive that $M' \twoheadrightarrow \mathsf{DM}_\sigma(\psi')$. Since M' cannot communicate by assumption, we get that $M' = \mathsf{DM}_\sigma(\psi)$. □

Then we show a result for the synthesis function in the centralized setting specifically for ψ-formulas, which differs from Lemma 3.3 because σ remains constant. This allows us to prove the base case in the proof for Formula Convergence (Lemma B.65) via Lemma B.61.

Lemma B.62. *If $\mathsf{cm}_\sigma(\psi) \xrightarrow{A} m'$, then $m' = \mathsf{cm}_\sigma(\psi')$ for some ψ'.*

The next lemma is used to prove Lemma B.64, which in turn is used in the final steps of the proof of Formula Convergence (Lemma B.65).

Lemma B.63. *If* $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma'}(\varphi')$ *and* $\mathcal{L} = \{\ell\}$*, then* $\mathrm{DM}_\sigma(\varphi) = \mathrm{DM}_{\sigma'}(\varphi')$*.*

Proof. We proceed by induction on φ. In the base case we consider $\varphi \in \mathsf{Qf}$: by Lemma B.59 and $\mathcal{L} = \{\ell\}$, we obtain $\mathrm{DM}_\sigma(\varphi) = [\mathrm{Dm}^\ell_\sigma(\varphi)]_\ell = [\mathrm{Dm}^\ell_{\sigma'}(\varphi')]_\ell = \mathrm{DM}_{\sigma'}(\varphi')$.

In the inductive case for $\varphi = \exists\pi.\varphi''$, we have $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi'') = \mathsf{cm}_{\sigma'}(\varphi')$. From the induction hypothesis we derive that $\mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi'') = \mathrm{DM}_{\sigma'}(\varphi')$, which proves the claim, since $\mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi'') = \mathrm{DM}_\sigma(\varphi)$ because $\mathcal{L} = \{\ell\}$. The case for $\varphi = \forall\pi.\varphi''$ is similar.

Finally, let $\varphi = \varphi_1 \diamond \varphi_2$. Hence, $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_\sigma(\varphi_1) \odot \mathsf{cm}_\sigma(\varphi_2) = \mathsf{cm}_{\sigma'}(\varphi')$ (for $\odot = \otimes$ if $\diamond = \wedge$ and $\odot = \oplus$ if $\diamond = \vee$). As $\mathcal{L} = \{\ell\}$, we know that $\varphi' = \varphi'_1 \diamond \varphi'_2$ with $\mathsf{cm}_\sigma(\varphi_1) = \mathsf{cm}_{\sigma'}(\varphi'_1)$ and $\mathsf{cm}_\sigma(\varphi_2) = \mathsf{cm}_{\sigma'}(\varphi'_2)$. From the induction hypothesis we conclude that $\mathrm{DM}_\sigma(\varphi_1) = \mathrm{DM}_{\sigma'}(\varphi'_1)$ and $\mathrm{DM}_\sigma(\varphi_2) = \mathrm{DM}_{\sigma'}(\varphi'_2)$. The result now follows: $\mathrm{DM}_\sigma(\varphi) = \mathrm{DM}_\sigma(\varphi_1) \diamond \mathrm{DM}_\sigma(\varphi_2) = \mathrm{DM}_{\sigma'}(\varphi'_1) \diamond \mathrm{DM}_{\sigma'}(\varphi'_2) = \mathrm{DM}_{\sigma'}(\varphi')$. □

Lemma B.64. *If* $\mathsf{cm}_\sigma(\varphi) = \oplus_{i\in I}\mathsf{cm}_{\sigma_i}(\varphi_i)$*, then* $\mathrm{DM}_\sigma(\varphi) = \bigvee_{i\in I} \mathrm{DM}_{\sigma_i}(\varphi_i)$*. If* $\mathsf{cm}_\sigma(\varphi) = \otimes_{i\in I}\mathsf{cm}_{\sigma_i}(\varphi_i)$*, then* $\mathrm{DM}_\sigma(\varphi) = \bigwedge_{i\in I} \mathrm{DM}_{\sigma_i}(\varphi_i)$*.*

Proof. We prove only the first claim (the second one is similar) and assume that $\oplus_{i\in I}\mathsf{cm}_{\sigma_i}(\varphi_i)$ is maximal, i.e., no $\mathsf{cm}_{\sigma_i}(\varphi_i)$ is of the form $m_1 \oplus m_2$. We proceed by induction on φ. If $|I| = 1$ (this covers the base cases—i.e., when φ is an atom—and somee inductive cases—e.g., an existential but with $|\mathcal{L}| = 1$), then we have $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_{\sigma_i}(\varphi_i)$ and the result follows from Lemma B.63.

If $\varphi = \varphi_1 \vee \varphi_2$, then $\mathsf{cm}_\sigma(\varphi) = \mathsf{cm}_\sigma(\varphi_1) \oplus \mathsf{cm}_\sigma(\varphi_2) = \oplus_{i\in I}\mathsf{cm}_{\sigma_i}(\varphi_i)$. Then, there is some $J \subsetneq I$ such that $\mathsf{cm}_\sigma(\varphi_1) = \oplus_{i\in J}\mathsf{cm}_{\sigma_i}(\varphi_i)$ and $\mathsf{cm}_\sigma(\varphi_2) = \oplus_{i\in I\setminus J}\mathsf{cm}_{\sigma_i}(\varphi_i)$. By the induction hypothesis, $\mathrm{DM}_\sigma(\varphi_1) = \bigvee_{i\in J} \mathrm{DM}_{\sigma_i}(\varphi_i)$ and $\mathrm{DM}_\sigma(\varphi_2) = \bigvee_{i\in I\setminus J} \mathrm{DM}_{\sigma_i}(\varphi_i)$, and therefore $\mathrm{DM}_\sigma(\varphi) = \mathrm{DM}_\sigma(\varphi_1) \vee \mathrm{DM}_\sigma(\varphi_2) = \bigvee_{i\in I} \mathrm{DM}_{\sigma_i}(\varphi_i)$.

Now consider $\varphi = \exists\pi.\varphi'$, with $|\mathcal{L}| > 1$; then, $\mathsf{cm}_\sigma(\varphi) = \oplus_{\ell\in\mathcal{L}}\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi')$. From our assumption on the maximality of $\oplus_{i\in I}\mathsf{cm}_{\sigma_i}(\varphi_i)$, there is a partition $(I_\ell)_{\ell\in\mathcal{L}}$ of I such that $\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi') = \oplus_{i\in I_\ell}\mathsf{cm}_{\sigma_i}(\varphi_i)$, for every $\ell \in \mathcal{L}$. By the induction hypothesis, $\mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi') = \bigvee_{i\in I_\ell} \mathrm{DM}_{\sigma_i}(\varphi_i)$, for every $\ell \in \mathcal{L}$, and we can conclude that $\mathrm{DM}_\sigma(\varphi) = \bigvee_{\ell\in\mathcal{L}} \mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi') = \bigvee_{\ell\in\mathcal{L}} \bigvee_{i\in I_\ell} \mathrm{DM}_{\sigma_i}(\varphi_i) = \bigvee_{i\in I} \mathrm{DM}_{\sigma_i}(\varphi_i)$. □

Lemma B.65. *If* $\mathrm{DM}_\sigma(\varphi) \xrightarrow{A} M \longrightarrow M'$*, with* M' *that cannot communicate, and* $\mathsf{cm}_\sigma(\varphi) \xrightarrow{A} \mathsf{cm}_{\sigma'}(\varphi')$*, for some formula* φ' *and environment* σ'*, then* $M' = \mathrm{DM}_{\sigma'}(\varphi')$*.*

Proof. We proceed by induction on φ. The case for ψ follows from Lemma B.61, where we know that $\sigma' = \sigma$ because of Lemma B.62.

We start with the inductive case for $\varphi = \exists\pi.\varphi''$. Since $\mathrm{DM}_\sigma(\exists\pi.\varphi'') = \bigvee_{\ell\in\mathcal{L}} \mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi'')$, we have $M = \bigvee_{\ell\in\mathcal{L}} M_\ell$, where $\mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi'') \xrightarrow{A} M_\ell$, for each ℓ. By Bounded Communication, we obtain that, for all ℓ, there exists N_ℓ such that $M_\ell \longrightarrow N_\ell$ and N_ℓ cannot communicate. We apply Lemma B.55 and conclude that $M \longrightarrow \bigvee_{\ell\in\mathcal{L}} N_\ell$. Then we can use Lemma B.32 to derive that $\bigvee_{\ell\in\mathcal{L}} N_\ell \longrightarrow M'$, which implies that $\bigvee_{\ell\in\mathcal{L}} N_\ell = M'$ because $\bigvee_{\ell\in\mathcal{L}} N_\ell$ cannot communicate. Moreover, $\mathsf{cm}_\sigma(\varphi) = \bigoplus_{\ell\in\mathcal{L}} \mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi'')$ and $\mathsf{cm}_{\sigma'}(\varphi') = \bigoplus_{\ell\in\mathcal{L}} \mathsf{cm}_{\sigma_\ell}(\varphi_\ell)$, where $\mathsf{cm}_{\sigma[\pi\mapsto\ell]}(\varphi'') \xrightarrow{A} \mathsf{cm}_{\sigma_\ell}(\varphi_\ell)$, for each $\ell \in \mathcal{L}$. From $\mathrm{DM}_{\sigma[\pi\mapsto\ell]}(\varphi'') \xrightarrow{A} M_\ell \longrightarrow N_\ell$ and the induction hypothesis, we can obtain that $N_\ell = \mathrm{DM}_{\sigma_\ell}(\varphi_\ell)$; hence, $M' = \bigvee_{\ell\in\mathcal{L}} \mathrm{DM}_{\sigma_\ell}(\varphi_\ell)$ and we conclude via Lemma B.64, that implies that $\bigvee_{\ell\in\mathcal{L}} \mathrm{DM}_{\sigma_\ell}(\varphi_\ell) = \mathrm{DM}_{\sigma'}(\varphi')$.

The case for $\varphi_1 \vee \varphi_2$ is similar; the cases for $\forall\pi.\varphi''$ and $\varphi_1 \wedge \varphi_2$ are dual. □

Received 22 October 2024; revised 1 May 2025; accepted 31 August 2025