Communicating Finite-State Machines, First-Order Logic, and Star-Free Propositional Dynamic Logic[☆]

Benedikt Bollig, Marie Fortin, Paul Gastin

LSV, CNRS & ENS Paris-Saclay, Université Paris-Saclay, Cachan, France

Abstract

Message sequence charts (MSCs) naturally arise as executions of communicating finite-state machines (CFMs), in which finite-state processes exchange messages through unbounded FIFO channels. We study the first-order logic of MSCs, featuring Lamport's happened-before relation. To this end, we introduce a star-free version of propositional dynamic logic (PDL) with loop and converse. Our main results state that (i) every first-order sentence can be transformed into an equivalent star-free PDL sentence (and conversely), and (ii) every star-free PDL sentence can be translated into an equivalent CFM. This answers an open question and settles the exact relation between CFMs and fragments of monadic second-order logic. As a byproduct, we show that first-order logic over MSCs has the three-variable property.

Keywords:
communicating finite-state machines, first-order logic, happened-before relation, propositional dynamic logic

1. Introduction

The study of logic-automata connections has ever played a key role in computer science, relating concepts that are a priori very different. Its motivation is at least twofold. First, automata may serve as a tool to decide logical theories. Beginning with the work of Büchi, Elgot, and Trakhtenbrot, who established in the early 60s the expressive equivalence of monadic second-order (MSO) logic and finite automata [Büc60, Elg61, Tra62], the "automata-theoretic" approach to logic has been successfully applied, for example, to MSO logic on trees [TW68], temporal logics [VW86], and first-order logic with two variables over words with an equivalence relation (aka *data words*) [BDM$^+$11]. Second, automata serve as models of various kind of state-based systems. Against this background, Büchi-like theorems lay the foundation of *synthesis*, i.e., the process of transforming high-level specifications (represented as logic formulas) into faithful system models. In this paper, we provide a Büchi theorem for *communicating finite-state machines (CFMs)*, which are a classical model of concurrent message-passing systems.

1.1. Context and Known Results

Let us give a brief account of what was already known on the relation between logic and automata (without claim of completeness).

Finite automata. As mentioned above, Büchi, Elgot, and Trakhtenbrot proved that *finite automata* over words are expressively equivalent to MSO logic [Büc60, Elg61, Tra62]. Finite automata can be considered as single finite-state processes and, therefore, serve as a model of *sequential* systems. Their executions are words, which, seen as a logical structure, consist of a set of positions (also referred to as *events*) that carry letters from a finite alphabet and are *linearly* ordered by some binary relation $\leq$. The simple MSO (even first-order) formula $\forall x.(a(x) \implies \exists y.(x \leq y \wedge b(y)))$ says that every "request" a is eventually followed by an "acknowledgment" b. In fact, Büchi's theorem allows one to turn

[☆]Partly supported by ANR FREDDA (ANR-17-CE40-0013) and ReLaX, UMI2000 (CNRS, ENS Paris-Saclay, Univ. Bordeaux, CMI, IMSc).
Email addresses: bollig@lsv.fr (Benedikt Bollig), fortin@lsv.fr (Marie Fortin), gastin@lsv.fr (Paul Gastin)

any logical MSO specification into a finite automaton. The latter can then be considered correct by construction. Though the situation quickly becomes more intricate when we turn to other automata models, Büchi theorems have been established for expressive generalizations of finite automata that also constitute natural system models. In the following, we will discuss some of them.

Data automata. *Data automata* accept (in the context of system models, we may also say *generate*) words that, in addition to the linear order $\leq$ and its direct-successor relation, are equipped with an equivalence relation $\sim$ [BDM$^+$11]. Positions (events) that belong to the same equivalence class may be considered as being executed by one and the same process, while $\leq$ reflects a sort of global control. It is, therefore, convenient to also include a predicate that connects *successive* events in an equivalence class. Bojańczyk et al. showed that data automata are expressively equivalent to existential MSO logic with two first-order variables [BDM$^+$11]. A typical formula is $\neg\exists x.\exists y.(x \neq y \wedge x \sim y)$, which says that every equivalence class is a singleton. It should be noted that data automata scan a word twice and, therefore, can hardly be seen as a system model. However, they are expressively equivalent to *class-memory automata*, which distinguish between a global control (modeling, e.g., a shared variable) and a local control for every process [BS10].

Asynchronous automata. Unlike finite automata and data automata, *asynchronous automata* are models of *concurrent* shared-memory systems, with a finite number of processes. In his influential paper [Lam78], Lamport postulated that events in an execution of a distributed system are partially ordered by what is commonly referred to as the happened-before or causal-precedence relation, a fundamental concept in distributed computing [AW04, Ray13, Lyn96, Tel01]. In fact, executions of asynchronous automata are Mazurkiewicz traces [DR95], where the relation $\leq$ is no longer a total, but a partial order. Thus, there may be *parallel* events x and y, for which neither $x \leq y$ nor $y \leq x$ holds. A typical logical specification is the mutual exclusion property, which can be expressed in MSO logic as $\neg\exists x.\exists y.(CS(x) \wedge CS(y) \wedge x \parallel y)$ where the parallel operator $x \parallel y$ is defined as $\neg(x \leq y) \wedge \neg(y \leq x)$. The formula says that there are no two events x and y that access a critical section simultaneously. Asynchronous automata are closed under complementation [Zie87] so that the inductive approach to translating formulas into automata can be applied to obtain a Büchi theorem [Tho90]. Note that complementability is also the key ingredient for MSO characterizations of **nested-word automata** [AM09] and **branching automata** running over series-parallel posets (aka N-free posets) [Kus00, Bed15].

Communicating finite-state machines. The situation is quite different in the realm of *communicating finite-state machines (CFMs)*, aka *communicating automata* or *message-passing automata*, where a fixed number of finite-state processes communicate by exchanging messages through *unbounded* FIFO channels [BZ83]. A CFM accepts/generates message-sequence charts (MSCs), which are similar to UML's sequence diagrams [Ara98] and standardised by the International Telecommunication Union [IT99]. MSCs are equipped with Lamport's happened-before relation $\leq$: an event e happens before an event f if, and only if, there is a "message flow" path from e to f [Lam78]. Additional binary predicates connect (i) the emission of a message with its reception, and (ii) successive events executed by one and the same process. Unfortunately, the class of MSC languages accepted by CFMs is not closed under complementation [BL06] so that an inductive translation of MSO logic into automata must fail (in fact, CFMs are strictly less expressive than MSO logic).

There have been several attempts to overcome this problem. When channels are bounded, closure under complementation is recovered so that CFMs are expressively equivalent to MSO logic [HMK$^+$05, Kus03, GKM06, GKM07]. Note that, however, the corresponding proofs are much more intricate than in the case of finite automata. In the unbounded case, since MSO logic is too expressive, first-order (FO) logic is moving into focus. Actually, FO logic can be considered, in many ways, a reference specification language. Apart from being a natural concept in itself, it plays a key role in automated theorem proving and is central in the verification of reactive systems. Over words, FO logic even enjoys manifold characterizations: It defines exactly the star-free languages and coincides with recognizability by aperiodic monoids or natural subclasses of finite (Büchi, respectively) automata (cf. [DG08, Tho97] for overviews). Moreover, linear-time temporal logics are usually measured against their expressive power with respect to FO logic. For example, LTL is considered the yardstick temporal logic not least due to Kamp's famous theorem, stating that LTL and FO logic are expressively equivalent [Kam68].

While FO logic on words is well understood, a lot remains to be said once message-passing concurrency enters into the picture. Actually, algebraic and automata-theoretic approaches that work for words, trees, or Mazurkiewicz

traces do not carry over. On the positive side, it was shown that CFMs with unbounded channels capture FO logic (and, therefore, are expressively equivalent to *existential* MSO logic) when dropping the happened-before relation $\leq$ [BL06] or when restricting to two first-order variables [BFG18]. Both results rely on normal forms of FO logic, due to Hanf [Han65] and Scott [GO99], respectively. Hanf's normal form is a boolean combination of statements of the form "neighborhood N of radius d occurs at least k times", where the neighborhood of an event e is an isomorphism type of the substructure induced by the elements that have distance at most d from e. Hanf's result requires structures of bounded degree so that the number of possible neighborhoods is actually finite. However, MSCs with the happened-before relation are structures of *unbounded* degree: Due to the happened-before relation $\leq$, all events on a given process have distance at most 1 from each other. To evaluate Scott's normal form, on the other hand, it is sufficient to determine the *type* of every event e (the type describing all events for every possible relationship with e), which can be accomplished by a CFM. However, the normal form only applies to two-variable logic, while we consider full FO logic, which, already over one process, is strictly more expressive.

It should be noted that distributed automata can also be used as acceptors of the underlying (graph) architecture. In that case, logical characterizations have been obtained in terms of MSO and modal logics [Kuu13, HJK$^+$15, Rei15, Rei17]. However, in our framework, the architecture is fixed and we rather reason about the set of *executions* of a CFM.

1.2. Contribution

Until now, the following central problem remained open:

> *Can every first-order sentence, with happened-before relation and arbitrarily many variables, be transformed into an equivalent communicating finite-state machine, without any channel bounds?*

In this paper, we answer the question positively. To do so, we make a detour through a variant of propositional dynamic logic (PDL) with loop and converse [FL79, Str81], which is another fundamental logic, with applications in artificial intelligence and verification [HM92, DL94, LL05, Lan06, GLL09]. Actually, we introduce *star-free* PDL, which serves as an interface between FO logic and CFMs. That is, there are two main tasks to accomplish:

(i) Translate every FO sentence into a star-free PDL sentence.

(ii) Translate every star-free PDL sentence into a CFM.

Both parts constitute results of own interest. In particular, step (i) implies that, over MSCs, FO logic has the three-variable property, i.e., every FO sentence over MSCs can be rewritten into one that uses only three different variable names. Note that this is already interesting in the special case of words, where it follows from Kamp's theorem [Kam68]. It is also noteworthy that star-free PDL is a *two-dimensional* temporal logic in the sense of Gabbay et al. [Gab81, GHR94]. Since every star-free PDL sentence is equivalent to some FO sentence, we actually provide a (higher-dimensional) temporal logic over MSCs that is expressively complete for FO logic.[1] While step (i) is based on purely logical considerations, step (ii) builds on new automata constructions that allow us to cope with the loop operator of PDL.

Combining (i) and (ii) yields the translation from FO logic to CFMs. It follows that CFMs are expressively equivalent to *existential* MSO logic. Moreover, we can derive self-contained proofs of several results on channel-bounded CFMs whose original proofs refer to involved constructions for Mazurkiewicz traces (cf. Section 5). In fact, we also extend these results to infinite MSCs.

1.3. Outline

In Section 2, we recall basic notions such as MSCs, FO logic, and CFMs. We also give a brief overview of what was already known on the relation between logic and CFMs. Section 3 presents star-free PDL and shows that it captures FO logic over MSCs. In Section 4, we establish the translation of star-free PDL into CFMs. As corollaries,

[1] It is open whether there is an equivalent one-dimensional one.

we obtain the translation of FO sentences into CFMs and the equivalence between CFMs and existential MSO logic. Several applications of our results are presented in Section 5. In particular, we obtain known results on CFMs with existentially bounded channels as a corollary. As a reference, an overview of previously known facts is presented in Section 2.4. We conclude in Section 6.

A preliminary version of this paper has been presented at the 29th International Conference on Concurrency Theory (CONCUR'18) and is accessible at `http://drops.dagstuhl.de/opus/frontdoor.php?source_opus=9545`. There, we considered *finite* MSCs. The present paper generalizes our results to infinite MSCs, which require several technical adjustments. Moreover, we provide full proofs as well as an application to channel-bounded CFMs.

2. Preliminaries

We consider message-passing systems in which processes communicate through unbounded FIFO channels. We fix a nonempty finite set of *processes* P and a nonempty finite set of *labels* Σ. For all $p, q \in P$ such that $p \neq q$, there is a channel (p, q) that allows p to send messages to q. The set of channels is denoted Ch.

In the following, we define message sequence charts, which represent executions of a message-passing system, and logics to reason about them. Then, we recall the definition of communicating finite-state machines and state one of our main results.

2.1. Message Sequence Charts

A *message sequence chart (MSC)* (over P and Σ) is a graph $M = (E, \rightarrow, \lhd, loc, \lambda)$ with nonempty, finite or countably infinite set E of nodes, also called events, edge relations $\rightarrow, \lhd \subseteq E \times E$, and node-labeling functions $loc\colon E \to P$ and $\lambda\colon E \to \Sigma$. An example MSC over $P = \{p_1, p_2, p_3\}$ and $\Sigma = \{\square, \bigcirc, \diamond\}$ is depicted in Figure 1. A node $e \in E$ is an *event* that is executed by process $loc(e) \in P$. In particular, $E_p := \{e \in E \mid loc(e) = p\}$ is the set of events located on p. Note that E_p can be finite or infinite. The label $\lambda(e) \in \Sigma$ may provide more information about e such as the message that is sent/received at e or "enter critical section" or "output some value".

Edges describe causal dependencies between events:

- The relation $\rightarrow$ contains *process edges*. They connect successive events executed by the same process, that is, we actually have $\rightarrow \subseteq \bigcup_{p \in P}(E_p \times E_p)$. Every process p is sequential so that $\rightarrow \cap (E_p \times E_p)$ must be the direct-successor relation of some total order on E_p. We let $\leq_{\mathsf{proc}} := \rightarrow^*$ and $<_{\mathsf{proc}} := \rightarrow^+$, and we require that every event $e \in E$ has a "finite past", i.e., $\{f \in E \mid f \leq_{\mathsf{proc}} e\}$ is finite.

- The relation $\lhd$ contains *message edges*. If $e \lhd f$, then e is a *send event* and f is the corresponding *receive event*. In particular, $(loc(e), loc(f)) \in Ch$. Each event is part of at most one message edge. An event that is neither a send nor a receive event is called *internal*. Moreover, for all $(p, q) \in Ch$ and $(e, f), (e', f') \in \lhd \cap (E_p \times E_q)$, we have $e \leq_{\mathsf{proc}} e'$ iff $f \leq_{\mathsf{proc}} f'$ (which guarantees a FIFO behavior).

We require that $\rightarrow \cup \lhd$ be acyclic (intuitively, messages cannot travel backwards in time). The associated partial order is denoted $\leq := (\rightarrow \cup \lhd)^*$ with strict part $< = (\rightarrow \cup \lhd)^+$. Actually, MSCs correspond to the space-time diagrams from Lamport's seminal paper [Lam78] when we assume a single FIFO channel between each pair of processes, and $\leq$ is commonly referred to as the *happened-before relation*.

We do not distinguish isomorphic MSCs. Let $\mathbb{MSC}(P, \Sigma)$ denote the set of MSCs over P and Σ. An MSC is *finite* if its set of events E is finite. We denote the set of finite MSCs by $\mathbb{MSC}^{\mathrm{fin}}(P, \Sigma)$.

It is worth noting that, when P is a singleton, an MSC with events $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots$ can be identified with the (finite or infinite) word $\lambda(e_1)\lambda(e_2)\lambda(e_3)\ldots$ over Σ.

Example 1. Consider the (infinite) MSC from Figure 1 over $P = \{p_1, p_2, p_3\}$ and $\Sigma = \{\square, \bigcirc, \diamond\}$. We have $E_{p_1} = \{e_i \mid i \in \mathbb{N}\}$, $E_{p_2} = \{f_0, \ldots, f_5\}$, $E_{p_3} = \{g_i \mid i \in \mathbb{N}\}$. The process relation is given by $e_i \rightarrow e_{i+1}$ and $g_i \rightarrow g_{i+1}$ for all $i \in \mathbb{N}$, as well as $f_i \rightarrow f_{i+1}$ for all $i \in \{0, \ldots, 4\}$. Concerning the message relation, we have $e_1 \lhd f_0$, $e_4 \lhd g_5$, etc. Moreover, $e_2 \leq f_3$, but neither $e_2 \leq f_1$ nor $f_1 \leq e_2$.

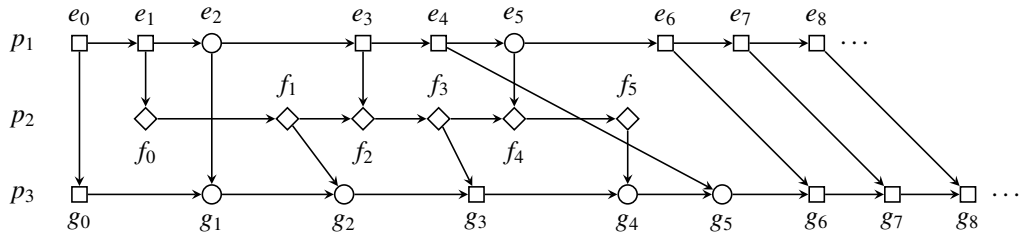


Figure 1: An (infinite) message sequence chart (MSC).

2.2. MSO Logic and Its Fragments

Next, we give an account of *monadic second-order* (MSO) logic and its fragments. Note that we restrict our attention to MSO logic interpreted *over MSCs*. We fix an infinite supply $\mathcal{V}_{\mathsf{event}} = \{x, y, \ldots\}$ of first-order variables, which range over events of an MSC, and an infinite supply $\mathcal{V}_{\mathsf{set}} = \{X, Y, \ldots\}$ of second-order variables, ranging over sets of events. The syntax of MSO (P and Σ are fixed) is given as follows:

$$\Phi \;::=\; p(x) \mid a(x) \mid x = y \mid x \to y \mid x \lhd y \mid x \leq y \mid x \in X \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists x.\Phi \mid \exists X.\Phi$$

where $p \in P$, $a \in \Sigma$, $x, y \in \mathcal{V}_{\mathsf{event}}$, and $X \in \mathcal{V}_{\mathsf{set}}$. We use the standard abbreviations to also include implication $\Longrightarrow$, conjunction $\wedge$, and universal quantification $\forall$. Moreover, the relation $x \leq_{\mathsf{proc}} y$ can be defined by $x \leq y \wedge \bigvee_{p \in P} p(x) \wedge p(y)$. We write $\mathsf{Free}(\Phi)$ for the set of free variables of Φ.

Let $M = (E, \to, \lhd, loc, \lambda)$ be an MSC. An *interpretation* (for M) is a mapping $\nu \colon \mathcal{V}_{\mathsf{event}} \cup \mathcal{V}_{\mathsf{set}} \to E \cup 2^E$ assigning to each $x \in \mathcal{V}_{\mathsf{event}}$ an event $\nu(x) \in E$, and to each $X \in \mathcal{V}_{\mathsf{set}}$ a set of events $\nu(X) \subseteq E$. We write $M, \nu \models \Phi$ if M satisfies Φ when the free variables of Φ are interpreted according to ν. Hereby, satisfaction is defined in the usual manner. In fact, whether $M, \nu \models \Phi$ holds or not only depends on the interpretation of variables that occur free in Φ. Thus, we may restrict ν to any set of variables that contains at least all free variables. For example, for $\Phi(x, y) = (x \lhd y)$, we have $M, [x \mapsto e, y \mapsto f] \models \Phi(x, y)$ iff $e \lhd f$. For a *sentence* $\Phi \in$ MSO (without free variables), we define $\mathbb{L}(\Phi) := \{M \in \mathbb{MSC}(P, \Sigma) \mid M \models \Phi\}$.

We say that two formulas Φ and Φ' are *equivalent*, written $\Phi \equiv \Phi'$, if, for all MSCs $M = (E, \to, \lhd, loc, \lambda)$ and interpretations $\nu \colon \mathcal{V}_{\mathsf{event}} \cup \mathcal{V}_{\mathsf{set}} \to E \cup 2^E$, we have $M, \nu \models \Phi$ iff $M, \nu \models \Phi'$.

Let us identify two important fragments of MSO logic: *First-order* (FO) formulas do not make use of second-order quantification (however, they may contain formulas $x \in X$). Moreover, *existential* MSO (EMSO) formulas are of the form $\exists X_1 \ldots \exists X_n.\Phi$ with $\Phi \in$ FO.

Let $\mathcal{F}$ be MSO or EMSO or FO and let $R \subseteq \{\to, \lhd, \leq\}$. We obtain the logic $\mathcal{F}[R]$ by restricting $\mathcal{F}$ to formulas that do not make use of $\{\to, \lhd, \leq\} \setminus R$. Note that $\mathcal{F} = \mathcal{F}[\to, \lhd, \leq]$. Moreover, we let $\mathcal{L}(\mathcal{F}[R]) := \{\mathbb{L}(\Phi) \mid \Phi \in \mathcal{F}[R] \text{ is a sentence}\}$.

As the reflexive transitive closure of an MSO-definable binary relation is MSO-definable, MSO and MSO$[\to, \lhd]$ have the same expressive power: $\mathcal{L}(\text{MSO}[\to, \lhd, \leq]) = \mathcal{L}(\text{MSO}[\to, \lhd])$. However, MSO$[\leq]$ (without the message relation) is strictly weaker than MSO [BL06]. In fact, over totally ordered MSCs, MSO$[\leq]$ only has the expressive power of MSO logic over ordinary words, and hence of finite automata, when restricting to valid linear extensions of MSCs. This allows one to apply a classical pumping argument to finite automata to show the result.

Example 2. Let us start with an easy formula saying that an MSC is infinite. This can be expressed in FO$[\to]$ by $\Phi = \bigvee_{p \in P} \exists x\, p(x) \wedge \forall x \exists y (p(x) \implies x \to y)$. Thus, $\mathbb{L}(\Phi) = \mathbb{MSC}(P, \Sigma) \setminus \mathbb{MSC}^{\text{fin}}(P, \Sigma)$.

Example 3. We now give an FO$[\leq]$ formula that allows us to recover, at some event f, the most recent event e that happened in the past on, say, process p. More precisely, we define the predicate $latest_p(x, y)$ as $x \leq y \wedge p(x) \wedge \forall z((z \leq y \wedge p(z)) \implies z \leq x)$. We are interested in the MSC language where process q always maintains the latest information that it can have about p. Thus, it is defined by

$$\Phi^{\mathsf{latest}}_{p,q} = \forall x \forall y. \left((latest_p(x, y) \wedge q(y)) \implies \bigvee_{a \in \Sigma} (a(x) \wedge a(y)) \right) \in \text{FO}[\leq]\,.$$

For example, for $P = \{p_1, p_2, p_3\}$ and $\Sigma = \{\Box, \bigcirc, \Diamond\}$, the MSC M from Figure 1 is contained in $\mathbb{L}(\Phi^{\mathsf{latest}}_{p_1,p_3})$. In particular, $M, [x \mapsto e_5, y \mapsto g_5] \models \mathit{latest}_{p_1}(x, y)$ and $\lambda(e_5) = \lambda(g_5) = \bigcirc$.

2.3. Communicating Finite-State Machines

In a communicating finite-state machine, each process $p \in P$ can perform internal actions of the form $\langle a \rangle$, where $a \in \Sigma$, or send/receive messages from a finite set of messages Msg. A send action $\langle a, !_q m \rangle$ of process p writes message $m \in \mathit{Msg}$ to channel (p, q), and performs $a \in \Sigma$. A receive action $\langle a, ?_q m \rangle$ reads message m from channel (q, p). Accordingly, we let $\mathit{Act}_p(\mathit{Msg}) := \{\langle a \rangle \mid a \in \Sigma\} \cup \{\langle a, !_q m \rangle \mid a \in \Sigma, m \in \mathit{Msg}, q \in P \setminus \{p\}\} \cup \{\langle a, ?_q m \rangle \mid a \in \Sigma, m \in \mathit{Msg}, q \in P \setminus \{p\}\}$ denote the set of possible actions of process p.

Definition 1. A *communicating finite-state machine (CFM)* over P and Σ is a tuple $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathit{Msg}, \mathit{Acc})$ consisting of a finite set of messages Msg and a finite-state transition system $\mathcal{A}_p = (S_p, \iota_p, \Delta_p)$ for each process p, with finite set of states S_p, initial state $\iota_p \in S_p$, and transition relation $\Delta_p \subseteq S_p \times \mathit{Act}_p(\mathit{Msg}) \times S_p$. Moreover, we have an acceptance condition Acc, which is a *positive* Boolean combination of atomic conditions $\langle p, s \rangle$ or $\langle p, s \rangle_\infty$, where $p \in P$ and $s \in S_p$.

Intuitively, $\langle p, s \rangle$ requires that process p terminates in state s (and, thus, executes only finitely many events), while $\langle p, s \rangle_\infty$ requires that process p enters state s infinitely often (which implies that p executes infinitely many events). This kind of "mixed" acceptance condition is quite convenient. Using positive Boolean combinations of acceptance conditions for infinite words was originally proposed in [EL87]. Other, syntactically different acceptance criteria have been adopted in the literature, like Büchi or Muller conditions [Kus03, BK08]. However, it is easily seen that they are all expressively equivalent.

Given a transition $t = (s, \alpha, s') \in \Delta_p$, we let $\mathit{source}(t) = s$ and $\mathit{target}(t) = s'$ denote the source and target states of t. In addition, if $\alpha = \langle a \rangle$, then t is an *internal transition* and we let $\mathit{label}(t) = a$. If $\alpha = \langle a, !_q m \rangle$, then t is a *send transition* and we let $\mathit{label}(t) = a$, $\mathit{msg}(t) = m$, and $\mathit{receiver}(t) = q$. Finally, if $\alpha = \langle a, ?_q m \rangle$, then t is a *receive transition* with $\mathit{label}(t) = a$, $\mathit{msg}(t) = m$, and $\mathit{sender}(t) = q$.

A *run* ρ of $\mathcal{A}$ on an MSC $M = (E, \rightarrow, \lhd, \mathit{loc}, \lambda) \in \mathbb{MSC}(P, \Sigma)$ is a mapping associating with each event $e \in E_p$ a transition $\rho(e) \in \Delta_p$, and satisfying the following conditions:

1. for all events $e \in E$, we have $\mathit{label}(\rho(e)) = \lambda(e)$,
2. for all $\rightarrow$-minimal events $e \in E$, we have $\mathit{source}(\rho(e)) = \iota_p$, where $p = \mathit{loc}(e)$,
3. for all process edges $(e, f) \in \rightarrow$, we have $\mathit{target}(\rho(e)) = \mathit{source}(\rho(f))$,
4. for all internal events $e \in E$, $\rho(e)$ is an internal transition, and
5. for all message edges $e \lhd f$, $\rho(e)$ and $\rho(f)$ are respectively send and receive transitions such that $\mathit{msg}(\rho(e)) = \mathit{msg}(\rho(f))$, $\mathit{receiver}(\rho(e)) = \mathit{loc}(f)$, and $\mathit{sender}(\rho(f)) = \mathit{loc}(e)$.

We say that ρ is *accepting* if it satisfies the acceptance condition Acc, written $\rho \models \mathit{Acc}$. The relation $\rho \models \mathit{Acc}$ is defined inductively. Disjunction and conjunction are interpreted as usual. Moreover, we let $\rho \models \langle p, s \rangle$ if either $E_p = \emptyset$ and $s = \iota_p$, or E_p is a nonempty finite set and $s = \mathit{target}(\rho(e))$, where e is the last event of E_p. Finally, $\rho \models \langle p, s \rangle_\infty$ if $s = \mathit{target}(\rho(e))$ for infinitely many events $e \in E_p$ (which implies that E_p is infinite).

The *language* $\mathbb{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of MSCs M such that there exists an accepting run of $\mathcal{A}$ on M. Moreover, $\mathcal{L}(\text{CFM}) := \{\mathbb{L}(\mathcal{A}) \mid \mathcal{A} \text{ is a CFM}\}$. Recall that, for these definitions, we have fixed P and Σ.

Following [HMK+05, GKM07, Kus03], we call a CFM $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathit{Msg}, \mathit{Acc})$ *deterministic* if, for all processes p and transitions $t_1 = (s_1, \alpha_1, s_1')$ and $t_2 = (s_2, \alpha_2, s_2')$ of $\mathcal{A}_p$ such that $s_1 = s_2$ and $\mathit{label}(t_1) = \mathit{label}(t_2)$, the following hold:

- If t_1 and t_2 are internal transitions, then $s_1' = s_2'$.
- If t_1 and t_2 are send transitions such that $\mathit{receiver}(t_1) = \mathit{receiver}(t_2)$, then $s_1' = s_2'$ and $\mathit{msg}(t_1) = \mathit{msg}(t_2)$.
- If t_1 and t_2 are receive transitions such that $\mathit{sender}(t_1) = \mathit{sender}(t_2)$ and $\mathit{msg}(t_1) = \mathit{msg}(t_2)$, then $s_1' = s_2'$.

Example 4. Consider the simple (deterministic) CFM $\mathcal{A}$ depicted in Figure 2. The set of processes is $P = \{p_1, p_2, p_3\}$. Moreover, we have $\Sigma = \{\square, \bigcirc, \diamond\}$ and $Msg = \{\square, \bigcirc\}$. Process p_1 sends messages to p_2 and p_3. Each message can be either $\square$ or $\bigcirc$, and the message sent is made "visible" in terms of Σ. Process p_2 simply forwards every message it receives to p_3. In any case, the action is $\diamond$. Finally, p_3 receives and "outputs" messages from p_1 and p_2 in any order. Note that, in this example, there are no local transitions, i.e., every transition is either sending or receiving. As acceptance condition, we take $Acc = \langle p_1, s_{p_1} \rangle_\infty$, which says that p_1 executes infinitely many events.

The CFM $\mathcal{A}$ can be seen as a first (naïve) attempt to solve the problem described in Example 3 by the formula $\Phi^{\mathsf{latest}}_{p_1,p_3}$ if we restrict to messages sent from p_i to p_j with $i < j$. Unfortunately, the protocol implemented by $\mathcal{A}$ is erroneous: For the MSC M in Figure 1, we have $M \in \mathbb{L}(\Phi^{\mathsf{latest}}_{p_1,p_3})$, but $M \notin \mathbb{L}(\mathcal{A})$. In $\mathcal{A}$, at g_2 and g_5, process p_3 should announce $\bigcirc$, but it outputs $\square$. It turns out that it is very difficult to come up with a CFM $\mathcal{A}^{\mathsf{latest}}_{p_1,p_3}$ such that $\mathbb{L}(\mathcal{A}^{\mathsf{latest}}_{p_1,p_3}) = \mathbb{L}(\Phi^{\mathsf{latest}}_{p_1,p_3})$ (even to show that such a CFM exists at all). This is already a challenging problem in the more specialized setting of Mazurkiewicz traces. However, we obtain $\mathcal{A}^{\mathsf{latest}}_{p_1,p_3}$ as a corollary of our logical characterization of CFMs, which we present in the following.

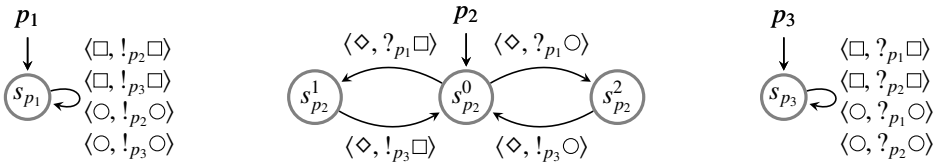


Figure 2: A communicating finite-state machine.

As we have demonstrated in the previous example, it is a worthwhile task to translate (simple) logical specifications like $\Phi^{\mathsf{latest}}_{p,q}$ into (complicated) machine models, preferably automatically. However, coming up with automata models directly can be very difficult. One of our main results (Theorem 3) states that *every* FO formula can be translated into a CFM. Our proof goes via an intermediate logic, namely star-free propositional dynamic logic ($\mathrm{PDL}_{\mathsf{sf}}$), which is introduced in the next section and shown to be expressively equivalent to $\mathrm{FO}[\rightarrow, \lhd, \leq]$. Then, in Section 4, we show how to translate $\mathrm{PDL}_{\mathsf{sf}}$ formulas into equivalent CFMs.

2.4. An Overview of Known Results

Let us give a brief account of what was already known on the relation between logic and CFMs. Note that we do not rely on any of these results.

Fact 1 [Büc60, Elg61, Tra62]**.** *Suppose $|P| = 1$ (i.e., CFMs are essentially finite automata). We have $\mathcal{L}(\mathrm{MSO}) = \mathcal{L}(\mathrm{CFM})$.*

This classical result is known as the Büchi-Elgot-Trakhtenbrot theorem. It was first generalized to CFMs with universally bounded channels (Fact 2). See Section 5.1 for the formal definition of existentially and universally bounded MSCs. Intuitively, a language L of MSCs is *universally B-bounded* if *all* linearizations of all MSCs in L can be executed with channel capacity B. We denote by $\mathbb{MSC}_{\forall B}(P, \Sigma)$ the set of MSCs in $\mathbb{MSC}(P, \Sigma)$ which are universally B-bounded. Moreover, $\mathbb{MSC}^{\mathrm{fin}}_{\forall B}(P, \Sigma) := \mathbb{MSC}_{\forall B}(P, \Sigma) \cap \mathbb{MSC}^{\mathrm{fin}}(P, \Sigma)$.

Fact 2 [HMK$^+$05]**.** *For all $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\forall B}(P, \Sigma)$, the following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$;*
2. *$L = \mathbb{L}(\mathcal{A})$ for some deterministic CFM $\mathcal{A}$;*
3. *$L = \mathbb{L}(\Phi)$ for some MSO formula Φ.*

Moreover, there is a deterministic CFM $\mathcal{A}$ such that $\mathbb{L}(\mathcal{A}) = \mathbb{MSC}^{\mathrm{fin}}_{\forall B}(P, \Sigma)$.

Kuske generalized the theorem to *infinite* universally bounded MSCs, while using a different proof technique.

Fact 3 [Kus03]**).** *For all $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}_{\forall B}(P, \Sigma)$, the following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$;*
2. *$L = \mathbb{L}(\mathcal{A})$ for some deterministic CFM $\mathcal{A}$;*
3. *$L = \mathbb{L}(\Phi)$ for some MSO formula Φ.*

In the case of finite MSCs, the logical characterization was lifted to existentially bounded MSCs by Genest et al. (cf. Fact 8). We denote by $\mathbb{MSC}_{\exists B}(P, \Sigma)$ the set of MSCs in $\mathbb{MSC}(P, \Sigma)$ which are *existentially B*-bounded, i.e., for which *some* linearization can be executed with channel capacity B. We also let $\mathbb{MSC}^{\text{fin}}_{\exists B}(P, \Sigma) := \mathbb{MSC}_{\exists B}(P, \Sigma) \cap \mathbb{MSC}^{\text{fin}}(P, \Sigma)$.

Fact 4 [GKM06]**).** *For all $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\text{fin}}_{\exists B}(P, \Sigma)$, the following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$;*
2. *$L = \mathbb{L}(\Phi)$ for some MSO formula Φ.*

Moreover, there is a CFM $\mathcal{A}$ such that $\mathbb{L}(\mathcal{A}) = \mathbb{MSC}^{\text{fin}}_{\exists B}(P, \Sigma)$.

On the other hand, it turns out that deterministic CFMs are now strictly weaker:

Fact 5 [GKM06]**).** *CFMs are inherently non-deterministic: There is a CFM $\mathcal{A}$ such that $\mathbb{L}(\mathcal{A}) \subseteq \mathbb{MSC}^{\text{fin}}_{\exists B}(P, \Sigma)$ and, for all deterministic CFMs $\mathcal{A}'$, we have $\mathbb{L}(\mathcal{A}) \neq \mathbb{L}(\mathcal{A}')$.*

The proofs of Facts 2, 3, and 4 reduce message-passing systems to finite-state shared-memory systems so that involved results from Mazurkiewicz trace theory [DR95] can be applied. This generic approach is no longer applicable when the restriction on the channel capacity is dropped. In fact, in general, CFMs do not capture MSO logic:

Fact 6 [BL06, BFG18]**).** *For all $L \subseteq \mathbb{MSC}^{\text{fin}}(P, \Sigma)$, the following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$;*
2. *$L = \mathbb{L}(\Phi)$ for some sentence $\Phi \in \text{EMSO}[\rightarrow, \lhd]$;*
3. *$L = \mathbb{L}(\Phi)$ for some sentence $\Phi \in \text{EMSO}^2[\rightarrow, \lhd, \leq]$.*

However, MSO is strictly more expressive than CFMs: There is an MSO sentence Φ such that $\mathbb{L}(\Phi) \subseteq \mathbb{MSC}^{\text{fin}}(P, \Sigma)$ and, for all CFMs $\mathcal{A}$, we have $\mathbb{L}(\Phi) \neq \mathbb{L}(\mathcal{A})$.

The characterizations from Fact 6 were given for finite MSCs. Over infinite MSCs, EMSO$[\rightarrow, \lhd]$ is strictly weaker than CFMs as it cannot express that *there are infinitely many events* to satisfy some property. Actually, this is already true for one process, i.e., finite automata and words. However, CFMs can be characterized by the logic $\text{EMSO}^{\infty}[\rightarrow, \lhd]$, extending EMSO$[\rightarrow, \lhd]$ by the quantifier $\exists^{\infty} x.\Phi$, which requires that there be *infinitely many events* x such that Φ holds.

Fact 7 [BK08]**).** *For all $L \subseteq \mathbb{MSC}(P, \Sigma)$, the following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$;*
2. *$L = \mathbb{L}(\Phi)$ for some sentence $\Phi \in \text{EMSO}^{\infty}[\rightarrow, \lhd]$.*

Note that one of our main results (Theorem 4) is the equivalence of CFMs and EMSO$[\rightarrow, \lhd, \leq]$, which properly generalizes Facts 6 and 7. Moreover, we will show in Section 5 how to obtain Fact 4 as a corollary, while generalizing it to infinite MSCs.

3. Star-Free Propositional Dynamic Logic

In this section, we introduce a star-free version of propositional dynamic logic and show that it is expressively equivalent to FO$[\rightarrow, \lhd, \leq]$. This is the second main result of the paper. Then, in Section 4, we show how to translate star-free PDL formulas into CFMs.

Table 1: The semantics of $\mathrm{PDL}_{\mathsf{sf}}$.

$M \models \mathsf{E}\,\varphi$ if $M, e \models \varphi$ for some event $e \in E$	
$M \models \neg\xi$ if $M \not\models \xi$	$M \models \xi_1 \vee \xi_2$ if $M \models \xi_1$ or $M \models \xi_2$
$M, e \models p$ if $loc(e) = p$	$M, e \models \langle\pi\rangle\,\varphi$ if $\exists f \in [\![\pi]\!]_M(e) : M, f \models \varphi$
$M, e \models a$ if $\lambda(e) = a$	$M, e \models \mathsf{Loop}(\pi)$ if $(e, e) \in [\![\pi]\!]_M$
$M, e \models \neg\varphi$ if $M, e \not\models \varphi$	$M, e \models \varphi_1 \vee \varphi_2$ if $M, e \models \varphi_1$ or $M, e \models \varphi_2$
$[\![\rightarrow]\!]_M := \{(e, f) \in E \times E \mid e \rightarrow f\}$	$[\![\lhd_{p,q}]\!]_M := \{(e, f) \in E_p \times E_q \mid e \lhd f\}$
$[\![\leftarrow]\!]_M := \{(f, e) \in E \times E \mid e \rightarrow f\}$	$[\![\lhd_{p,q}^{-1}]\!]_M := \{(f, e) \in E_q \times E_p \mid e \lhd f\}$
$[\![\mathsf{jump}_{p,r}]\!]_M := E_p \times E_r$	$[\![\{\varphi\}?]\!]_M := \{(e, e) \mid e \in E : M, e \models \varphi\}$
$[\![\overset{\varphi}{\rightarrow}]\!]_M := \{(e, f) \in E \times E \mid e <_{\mathsf{proc}} f$ and $\forall g \in E$: $e <_{\mathsf{proc}} g <_{\mathsf{proc}} f \implies M, g \models \varphi\}$	
$[\![\overset{\varphi}{\leftarrow}]\!]_M := \{(e, f) \in E \times E \mid f <_{\mathsf{proc}} e$ and $\forall g \in E$: $f <_{\mathsf{proc}} g <_{\mathsf{proc}} e \implies M, g \models \varphi\}$	
$[\![\pi_1 \cdot \pi_2]\!]_M := \{(e, g) \in E \times E \mid \exists f \in E : (e, f) \in [\![\pi_1]\!]_M \wedge (f, g) \in [\![\pi_2]\!]_M\}$	
$[\![\pi_1 \cup \pi_2]\!]_M := [\![\pi_1]\!]_M \cup [\![\pi_2]\!]_M$	$[\![\pi^{\mathsf{c}}]\!]_M := (E \times E) \setminus [\![\pi]\!]_M$
$[\![\pi_1 \cap \pi_2]\!]_M := [\![\pi_1]\!]_M \cap [\![\pi_2]\!]_M$	

3.1. Syntax and Semantics

Originally, propositional dynamic logic (PDL) has been used to reason about program schemas and transition systems [FL79]. Since then, PDL and its extension with intersection and converse have developed a rich theory with applications in artificial intelligence and verification [HM92, DL94, LL05, Lan06, GLL09]. It has also been applied in the context of MSCs [BKM10, Men13].

Here, we introduce a *star-free* version of PDL, denoted $\mathrm{PDL}_{\mathsf{sf}}$. It will serve as an "interface" between FO logic and CFMs. The syntax of $\mathrm{PDL}_{\mathsf{sf}}$ and its fragment $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ is given by the following grammar:

$\mathrm{PDL}_{\mathsf{sf}} = \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup, \cap, \mathsf{c}]$

$\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$		
Sentence	$\xi ::= \mathsf{E}\,\varphi \mid \xi \vee \xi \mid \neg\xi$	
Event formula	$\varphi ::= p \mid a \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$	
Path formula	$\pi ::= \rightarrow \mid \leftarrow \mid \lhd_{p,q} \mid \lhd_{p,q}^{-1} \mid \overset{\varphi}{\rightarrow} \mid \overset{\varphi}{\leftarrow} \mid \mathsf{jump}_{p,r} \mid \{\varphi\}? \mid \pi \cdot \pi$	$\pi \cup \pi \mid \pi \cap \pi \mid \pi^{\mathsf{c}}$

where $p, r \in P$, $q \in P \setminus \{p\}$, and $a \in \Sigma$. We refer to ξ as a *sentence*, to φ as an *event formula*, and to π as a *path formula*. We name the logic star-free because we use the operators $(\cup, \cap, \mathsf{c}, \cdot)$ of star-free regular expressions instead of the regular-expression operators $(\cup, \cdot, *)$ of classical PDL. However, the formula $\overset{\varphi}{\rightarrow}$, whose semantics is explained below, can be seen as a restricted use of the construct π^*.

A sentence ξ is evaluated with respect to an MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$. An event formula φ is evaluated with respect to M and an event $e \in E$. Finally, a path formula π is evaluated over *two* events. In other words, it defines a binary relation $[\![\pi]\!]_M \subseteq E \times E$. We often write $M, e, f \models \pi$ to denote $(e, f) \in [\![\pi]\!]_M$. Moreover, for $e \in E$, we let $[\![\pi]\!]_M(e) := \{f \in E \mid (e, f) \in [\![\pi]\!]_M\}$. When M is clear from the context, we may write $[\![\pi]\!]$ instead of $[\![\pi]\!]_M$. The semantics of sentences, event formulas, and path formulas is given in Table 1. For a sentence ξ, we let $\mathbb{L}(\xi) := \{M \in \mathbb{MSC}(P, \Sigma) \mid M \models \xi\}$.

We use the standard abbreviations for sentences and event formulas such as implication and conjunction. Moreover, we let $true := p \vee \neg p$ (for some arbitrary process $p \in P$) and $false := \neg true$. Finally, we define the event formula

$\langle\pi\rangle := \langle\pi\rangle\, true$, and the path formulas $\xrightarrow{+} := \xrightarrow{true}$ and $\xrightarrow{*} := \xrightarrow{+} \cup \{true\}?$.

The *size* of a $\mathrm{PDL}_{\mathsf{sf}}$ formula is defined by mutual induction. We let $|\mathsf{E}\,\varphi| = |\varphi| + 1$, $|\neg\xi| = |\xi| + 1$, and $|\xi_1 \vee \xi_2| = |\xi_1| + |\xi_2| + 1$. For $p \in P$ and $a \in \Sigma$, $|p| = |a| = 1$. Moreover, $|\neg\varphi| = |\varphi| + 1$, $|\langle\pi\rangle\,\varphi| = |\pi| + |\varphi| + 1$, $|\mathsf{Loop}(\pi)| = |\pi| + 1$, and $|\varphi_1 \vee \varphi_2| = |\varphi_1| + |\varphi_2| + 1$. If $\pi \in \{\rightarrow, \leftarrow, \lhd_{p,q}, \lhd_{p,q}^{-1} \mid (p,q) \in Ch\} \cup \{\mathsf{jump}_{p,q} \mid p, q \in P\}$, we let $|\pi| = 1$. Moreover, $|\{\varphi\}?| = |\xrightarrow{\varphi}| = |\xleftarrow{\varphi}| = |\varphi| + 1$ and $|\pi^{\mathsf{c}}| = |\pi| + 1$. Finally, $|\pi_1\ op\ \pi_2| = |\pi_1| + |\pi_2| + 1$ for all $op \in \{\cup, \cap, \cdot\}$.

The usual temporal logic modalities can be expressed easily. For instance, $\langle\rightarrow\rangle\,\varphi$ means that the *next* event on the same process satisfies φ, and $\langle\xrightarrow{\varphi}\rangle\,\psi$ corresponds to the *strict until* $\mathsf{X}(\varphi\,\mathsf{U}\,\psi)$. The corresponding past modalities can be written similarly. See Section 5.2 for more modalities.

Example 5. Consider again the MSC M from Figure 1. For the path formula $\pi = \lhd_{p_1,p_3}^{-1}\rightarrow\lhd_{p_1,p_2}\rightarrow\lhd_{p_2,p_3}\rightarrow$, we have $M, g_5 \models \mathsf{Loop}(\pi)$. Moreover, $(e_2, e_5) \in [\![\xrightarrow{\square}]\!]_M$ but $(e_2, e_6) \notin [\![\xrightarrow{\square}]\!]_M$. To give an example of an event formula, note that we have $M, e_0 \models \langle\xrightarrow{+}\rangle\neg\langle\xrightarrow{+}\rangle\neg\square$ (as we eventually see only $\square$). Finally, since E_{p_1} is infinite, we also have $M \models \neg\,\mathsf{E}(p_1 \wedge \neg\langle\rightarrow\rangle)$.

Note that there are some redundancies in the logic. For example (letting $\equiv$ denote logical equivalence), $\rightarrow \equiv \xrightarrow{false}$, $\pi_1 \cap \pi_2 \equiv (\pi_1^{\mathsf{c}} \cup \pi_2^{\mathsf{c}})^{\mathsf{c}}$, and $\mathsf{Loop}(\pi) \equiv \langle\{true\}? \cap \pi\rangle$. Some of them are necessary to define certain subclasses of $\mathrm{PDL}_{\mathsf{sf}}$. For every $R \subseteq \{\mathsf{Loop}, \cup, \cap, \mathsf{c}\}$, we let $\mathrm{PDL}_{\mathsf{sf}}[R]$ denote the fragment of $\mathrm{PDL}_{\mathsf{sf}}$ that does not make use of $\{\mathsf{Loop}, \cup, \cap, \mathsf{c}\} \setminus R$. In particular, $\mathrm{PDL}_{\mathsf{sf}} = \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup, \cap, \mathsf{c}]$. Syntactically, $\xrightarrow{*}$ is not contained in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ since union is not permitted.

Given a $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formula π, we denote by $\mathsf{Comp}(\pi)$ the set of pairs $(p,q) \in P \times P$ such that there may be a π-path from some event on process p to some event on process q. Formally, we let $\mathsf{Comp}(\rightarrow) = \mathsf{Comp}(\leftarrow) = \mathsf{Comp}(\xrightarrow{\varphi}) = \mathsf{Comp}(\xleftarrow{\varphi}) = \mathsf{Comp}(\{\varphi\}?) = \mathsf{id}$, where $\mathsf{id} = \{(p,p) \mid p \in P\}$; $\mathsf{Comp}(\lhd_{p,q}) = \mathsf{Comp}(\lhd_{q,p}^{-1}) = \{(p,q)\}$; $\mathsf{Comp}(\mathsf{jump}_{p,r}) = \{(p,r)\}$; and $\mathsf{Comp}(\pi_1 \cdot \pi_2) = \mathsf{Comp}(\pi_2) \circ \mathsf{Comp}(\pi_1) = \{(p,r) \mid \exists q : (p,q) \in \mathsf{Comp}(\pi_1), (q,r) \in \mathsf{Comp}(\pi_2)\}$.

Notice that, for all path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$, the relation $\mathsf{Comp}(\pi)$ is either empty or a singleton $\{(p,q)\}$ or the identity id. Moreover, $M, e, f \models \pi$ implies $(loc(e), loc(f)) \in \mathsf{Comp}(\pi)$. Therefore, all events in $[\![\pi]\!](e)$ are on the same process, and if this set is nonempty (i.e., if $M, e \models \langle\pi\rangle$), then $\min[\![\pi]\!](e)$ is well-defined. We also define $\max[\![\pi]\!](e) \in [\![\pi]\!](e) \cup \{\infty\}$, with the convention $\max[\![\pi]\!](e) = \infty$ if $[\![\pi]\!](e)$ is infinite. We extend $\leq$ and $\leq_{\mathsf{proc}}$ to $E \cup \{\infty\}$ by setting $e \leq \infty$ and $e \leq_{\mathsf{proc}} \infty$ for all $e \in E \cup \{\infty\}$.

Example 6. Consider $\pi = \xrightarrow{+}\lhd_{p_1,p_2}\rightarrow\lhd_{p_2,p_3}\rightarrow$. We have $\mathsf{Comp}(\pi) = \{(p_1, p_3)\}$. Moreover, given the MSC from Figure 1, $\min[\![\pi]\!](e_2) = g_4$ and $\max[\![\pi]\!](e_2) = g_5$. On the other hand, $\max[\![\xrightarrow{\square} \cdot \lhd_{p_1,p_3}]\!](e_5) = \infty$.

Remark 1. The logic $\mathrm{PDL}_{\mathsf{sf}}[\cup]$ over MSCs is analogous to Conditional XPath [Mar05].[2] Formulas from Conditional XPath are interpreted over ordered unranked trees. Therefore, rather than atomic formulas $\rightarrow$ and $\lhd_{p,q}$ as well as their inverse operators, there are tailored formulas allowing one to move to a child or the parent of a given node, or to go to its immediate left/right sibling. However, while Marx showed that Conditional XPath is expressively complete for FO logic over ordered unranked trees, our expressive completeness result over MSCs crucially relies on the Loop modality, which is not contained in $\mathrm{PDL}_{\mathsf{sf}}[\cup]$ and not provided by Conditional XPath.

3.2. From $\mathrm{PDL}_{\mathsf{sf}}$ *to* FO^3

Let $\mathrm{FO}^3[\rightarrow, \lhd, \leq]$ be the set of formulas from $\mathrm{FO}[\rightarrow, \lhd, \leq]$ that use at most three different first-order variables (however, a variable can be quantified and reused several times in a formula). The main result of this section is that, for formulas with zero or one free variable, the logics $\mathrm{FO}[\rightarrow, \lhd, \leq]$, $\mathrm{FO}^3[\rightarrow, \lhd, \leq]$, $\mathrm{PDL}_{\mathsf{sf}}$, and $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ are expressively equivalent.

[2] Thanks to Sylvain Schmitz for pointing this out.

Consider FO[$\rightarrow, \lhd, \leq$] formulas Φ_0, $\Phi_1(x)$, and $\Phi_2(x,y)$ with respectively zero, one, and two free variables (hence, Φ_0 is a sentence). Consider also some $\mathrm{PDL}_{\mathsf{sf}}$ sentence ξ, event formula φ, and path formula π. The respective formulas are equivalent, written $\Phi_0 \equiv \xi$, $\Phi_1(x) \equiv \varphi$, and $\Phi_2(x,y) \equiv \pi$, if, for all MSCs M and all events e, f in M, we have

$$\begin{array}{rcr} M \models \Phi_0 & \text{iff} & M \models \xi \\ M, [x \mapsto e] \models \Phi_1(x) & \text{iff} & M, e \models \varphi \\ M, [x \mapsto e, y \mapsto f] \models \Phi_2(x,y) & \text{iff} & M, e, f \models \pi \end{array}$$

We start with a simple observation, which can be shown easily by induction:

Proposition 1. *Every* $\mathrm{PDL}_{\mathsf{sf}}$ *formula is equivalent to some* $\mathrm{FO}^3[\rightarrow, \lhd, \leq]$ *formula. More precisely, for every* $\mathrm{PDL}_{\mathsf{sf}}$ *sentence* ξ*, event formula* φ*, and path formula* π*, there exist some* $\mathrm{FO}^3[\rightarrow, \lhd, \leq]$ *sentence* $\widetilde{\xi}$*, formula* $\widetilde{\varphi}(x)$ *with one free variable, and formula* $\widetilde{\pi}(x,y)$ *with two free variables, respectively, such that,* $\xi \equiv \widetilde{\xi}$*,* $\varphi \equiv \widetilde{\varphi}(x)$*, and* $\pi \equiv \widetilde{\pi}(x,y)$*.*

The main result of this section is a *strong* converse of Proposition 1: Every FO[$\rightarrow, \lhd, \leq$] formula with at most two free variables is equivalent to some $\mathrm{PDL}_{\mathsf{sf}}$ formula. This is formally stated and proved in Section 3.5. We first investigate in the next section some basic properties of $\mathrm{PDL}_{\mathsf{sf}}$. Then, we show in Section 3.4 that the complement of a $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formula is equivalent to a finite union of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formulas. This is crucial to deal with negation in the translation from FO to $\mathrm{PDL}_{\mathsf{sf}}$. The other main difficulty is existential quantification, which is dealt with in Section 3.5.

3.3. Basic Properties of $\mathrm{PDL}_{\mathsf{sf}}$

First, the converse of a $\mathrm{PDL}_{\mathsf{sf}}$ formula is definable in $\mathrm{PDL}_{\mathsf{sf}}$ (easy induction on π).

Lemma 1. *Let* $R \subseteq \{\mathsf{Loop}, \cup, \cap, \mathsf{c}\}$ *and* $\pi \in \mathrm{PDL}_{\mathsf{sf}}[R]$ *be a path formula. There exists* $\pi^{-1} \in \mathrm{PDL}_{\mathsf{sf}}[R]$ *such that, for all MSCs* M*,* $[\![\pi^{-1}]\!]_M = [\![\pi]\!]_M^{-1} = \{(f,e) \mid (e,f) \in [\![\pi]\!]_M\}$*.*

A second observation is that unions in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ path formulas can always be pulled to the front of the formula.

Lemma 2. *Every* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ *path formula is equivalent to a finite union of* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *path formulas, and every* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ *event formula is equivalent to some* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *event formula.*

Proof. This is easy to prove by induction on $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ formulas, using the following identities: if $(\pi_i)_{1 \leq i \leq n}$ and $(\pi'_j)_{1 \leq j \leq m}$ are $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formulas, then

$$(\bigcup_i \pi_i) \cdot (\bigcup_j \pi'_j) \equiv \bigcup_{i,j} \pi_i \cdot \pi_j\,, \quad \mathsf{Loop}(\bigcup_i \pi_i) \equiv \bigvee_i \mathsf{Loop}(\pi_i)\,, \quad \langle \bigcup_i \pi_i \rangle \varphi \equiv \bigvee_i \langle \pi_i \rangle \varphi\,. \qquad \square$$

The next lemma shows that all $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formulas are, in some sense, monotone.

Lemma 3 (monotonicity)**.** *Let* $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *be a path formula,* M *be an MSC, and* e, f, e' *be events of* M *such that* $M, e, e' \models \pi$*, and* $M, f \models \langle \pi \rangle$ *(ie.,* $M, f, g \models \pi$ *for some event* g *in* M*).*

(a) If $e \leq_{\mathsf{proc}} f$*, then there exists* f' *such that* $e' \leq_{\mathsf{proc}} f'$ *and* $M, f, f' \models \pi$*.*

(b) If $f \leq_{\mathsf{proc}} e$*, then there exists* f' *such that* $f' \leq_{\mathsf{proc}} e'$ *and* $M, f, f' \models \pi$*.*

Proof. We only show (a). Part (b) is similar.

We prove by induction on π that, for all event formulas $\psi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$, the property holds for $\pi \cdot \{\psi\}?$.

- If $\pi = \{\varphi\}?$, then $e' = e$, and we can take $f' = f$.
- If $\pi = \mathsf{jump}_{p,q}$, we take $f' = e'$.
- If $\pi = \lhd_{p,q}$, then $e \lhd e'$ and there exists f' such that $M, f, f' \models \lhd_{p,q} \cdot \{\psi\}?$. In particular, $f \lhd f'$. Since the channels are FIFO, we have $e' \leq_{\mathsf{proc}} f'$.

- The cases $\pi = \lhd^{-1}_{p,q}$, $\pi = \rightarrow$, and $\pi = \leftarrow$ are similar.

- Suppose $\pi = \xrightarrow{\varphi}$. If $f <_{\mathsf{proc}} e'$, we take $f' = e'$. Otherwise, we let f' be any event such that $M, f, f' \models \xrightarrow{\varphi}$. We then have $e' \leq_{\mathsf{proc}} f <_{\mathsf{proc}} f'$. Similarly, if $\pi = \xleftarrow{\varphi}$, then either there exists $e \leq_{\mathsf{proc}} f' <_{\mathsf{proc}} f$ such that $M, f, f' \models \xleftarrow{\varphi} \cdot \{\psi\}?$, or $M, f, e' \models \xleftarrow{\varphi}$.

- Suppose $\pi = \pi_1 \cdot \pi_2$. There exists e_1 such that $M, e, e_1 \models \pi_1$ and $M, e_1, e' \models \pi_2 \cdot \{\psi\}?$. In particular, $M, e, e_1 \models \pi_1 \cdot \{\langle \pi_2 \rangle \psi\}?$. By induction hypothesis on π_1, there exists f_1 such that $e_1 \leq_{\mathsf{proc}} f_1$ and $M, f, f_1 \models \pi_1 \cdot \{\langle \pi_2 \rangle \psi\}?$. By induction hypothesis on π_2, there exists f' such that $M, f_1, f' \models \pi_2 \cdot \{\psi\}?$ and $e' \leq_{\mathsf{proc}} f'$. □

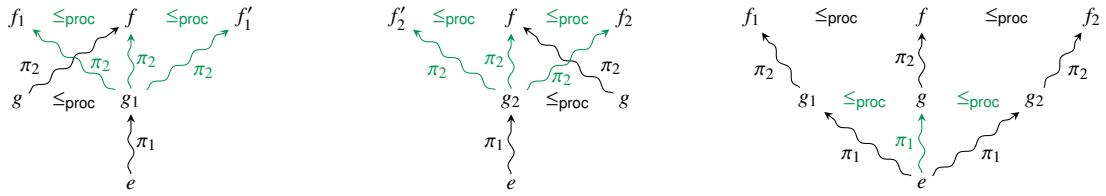
A crucial consequence of Lemma 3 is that, for all path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ and events e in some MSC, $[\![\pi]\!](e)$ contains precisely the events that lie in the interval between $\min[\![\pi]\!](e)$ and $\max[\![\pi]\!](e)$ and that satisfy $\langle \pi^{-1} \rangle$.

Lemma 4. *Let π be a $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formula. For all MSCs M and events e such that $M, e \models \langle \pi \rangle$, we have*

$$[\![\pi]\!](e) = \{f \in E \mid \min[\![\pi]\!](e) \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} \max[\![\pi]\!](e) \wedge M, f \models \langle \pi^{-1} \rangle\}\,.$$

Proof. The left-to-right inclusion is trivial. For the right-to-left inclusion, we show by induction on π that, for all events e and $f_1 \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} f_2$ such that $M, e, f_1 \models \pi$, $M, e, f_2 \models \pi$, and $M, f \models \langle \pi^{-1} \rangle$, we have $M, e, f \models \pi$.

All cases apart from concatenation are immediate. So assume $\pi = \pi_1 \cdot \pi_2$. There exist g_1, g_2, g such that $M, e, g_i \models \pi_1$, $M, g_i, f_i \models \pi_2$, $M, g, f \models \pi_2$, and $M, g \models \langle \pi_1^{-1} \rangle$. We distinguish three cases, illustrated in Figure 3.



Case 1. Case 2. Case 3.

Figure 3: Proof of Lemma 4.

1. If $g \leq_{\mathsf{proc}} g_1$, then by Lemma 3(a) applied to π_2, g, f, g_1, there exists $f_1' \geq_{\mathsf{proc}} f$ such that $M, g_1, f_1' \models \pi_2$. By induction hypothesis on π_2, we then have $M, g_1, f \models \pi_2$, hence $M, e, f \models \pi_1 \cdot \pi_2$.

2. Similarly, if $g_2 \leq_{\mathsf{proc}} g$, then by applying Lemma 3(b) to π_2, g_2, g, f, we find $f_2' \leq_{\mathsf{proc}} f$ such that $M, g_2, f_2' \models \pi_2$. Using the induction hypothesis on π_2, we obtain $M, g_2, f \models \pi_2$, hence $M, e, f \models \pi_1 \cdot \pi_2$.

3. Otherwise, we have $g_1 \leq_{\mathsf{proc}} g \leq_{\mathsf{proc}} g_2$. By induction hypothesis on π_1, we get $M, e, g \models \pi_1$, hence $M, e, f \models \pi_1 \cdot \pi_2$. □

3.4. Characterizing the Complement of a Path Formula

Using Lemma 4, we can give a characterization of $[\![\pi^{\mathsf{c}}]\!](e)$ (when $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$) that also relies on intervals delimited by $\min[\![\pi]\!](e)$ and $\max[\![\pi]\!](e)$. More precisely, $[\![\pi^{\mathsf{c}}]\!](e)$ is the union of the following sets (see Figure 4):

(i) the interval of all events to the left of $\min[\![\pi]\!](e)$,

(ii) the interval of all events to the right of $\max[\![\pi]\!](e)$ (assuming $\max[\![\pi]\!](e) \neq \infty$),

(iii) the set of events located between $\min[\![\pi]\!](e)$ and $\max[\![\pi]\!](e)$ and satisfying $\neg \langle \pi^{-1} \rangle$,

(iv) all events located on other processes than $\min[\![\pi]\!](e)$.

This description of $[\![\pi^{\mathsf{c}}]\!](e)$ can be used to rewrite π^{c} as a union of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formulas. In a first step, we show that, if π is a $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formula, then the relation $\{(e, \min[\![\pi]\!](e)) \mid e \in E\}$ can also be expressed in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$.

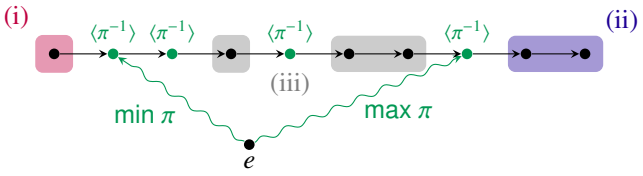


Figure 4: Characterization of $[\![\pi^c]\!](e)$ for $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$.

Lemma 5. *Let $R = \emptyset$ or $R = \{\mathsf{Loop}\}$. For every path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[R]$, there exists a $\mathrm{PDL}_{\mathsf{sf}}[R]$ path formula $\mathsf{min}\,\pi$ of size $O(|\pi|^2)$ such that, for all MSCs M and events e, f, we have $M, e, f \models \mathsf{min}\,\pi$ iff $f = \min[\![\pi]\!](e)$.*

Proof. Let us first observe that, for all $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formulas π_1 and π_2, for all MSCs M and events e of M such that $[\![\pi_1 \cdot \pi_2]\!](e) \neq \emptyset$, we have

$$\min[\![\pi_1 \cdot \pi_2]\!](e) = \min[\![\pi_2]\!](\min[\![\pi_1 \cdot \{\langle\pi_2\rangle\}?]\!](e)) \,. \tag{1}$$

Indeed, if $[\![\pi_1 \cdot \pi_2]\!](e) \neq \emptyset$, then $f = \min[\![\pi_1 \cdot \{\langle\pi_2\rangle\}?]\!](e)$ and $g = \min[\![\pi_2]\!](f)$ are well-defined (and reciprocally). Clearly, $M, e, g \models \pi_1 \cdot \pi_2$. To prove minimality, let f', g' such that $M, e, f' \models \pi_1$ and $M, f', g' \models \pi_2$ (cf. Figure 5). We have $f \leq_{\mathsf{proc}} f'$, hence, by Lemma 3(b), there exists $g'' \leq_{\mathsf{proc}} g'$ such that $M, f, g'' \models \pi_2$. Then $g \leq_{\mathsf{proc}} g'' \leq_{\mathsf{proc}} g'$.

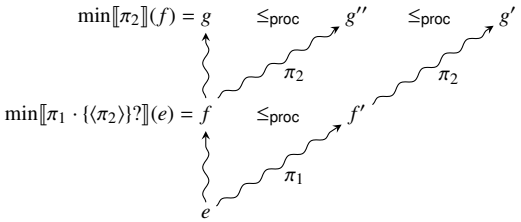


Figure 5: Proof of Lemma 5.

We can now give the definition of $\mathsf{min}\,\pi$. Since concatenation of paths is associative, we view π as a nonempty sequence of atomic steps and we construct $\mathsf{min}\,\pi$ by induction on the length of π. Without loss of generality, we assume that the last atomic step of the path formula is $\{true\}?$. Hence, the basis of the induction is when $\pi = \{true\}?$, in which case we let $\mathsf{min}\,\pi = \{true\}?$.

For the inductive case, assume that $\pi = r \cdot \pi'$ with r an atomic path formula. Inspired by Equation (1), we define inductively $\mathsf{min}\,\pi = \hat{r} \cdot \mathsf{min}\,\pi'$, where $\hat{r}$ is a path formula such that, for all events e and f with $M, e, f \models r \cdot \{\langle\pi'\rangle\}?$, we have $M, e, f \models \hat{r}$ if and only if $f = \min[\![r \cdot \{\langle\pi'\rangle\}?]\!](e)$. For an atomic path formula r, we define

$$\hat{r} = \begin{cases} r & \text{if } r \in \{\{\varphi\}?, \rightarrow, \leftarrow, \lhd_{p,q}, \lhd_{p,q}^{-1} \mid (p,q) \in Ch\} \\ \mathsf{jump}_{p,q} \cdot \{\neg \langle \overset{+}{\leftarrow} \cdot \pi' \rangle\}? & \text{if } r = \mathsf{jump}_{p,q} \\ \overset{\varphi}{\leftarrow} \cdot \{\neg\varphi \vee \neg \langle \overset{\varphi}{\leftarrow} \cdot \pi' \rangle\}? & \text{if } r = \overset{\varphi}{\leftarrow} \\ \xrightarrow{\varphi \wedge \neg \langle \pi' \rangle} & \text{if } r = \overset{\varphi}{\rightarrow} . \end{cases}$$

Notice that $M, e, f \models \hat{r}$ does not imply $M, f \models \langle\pi'\rangle$. We could enforce this by appending a test $\{\langle\pi'\rangle\}?$ at the end of $\hat{r}$, but this would be redundant due to the right context of $\hat{r}$ in $\mathsf{min}\,\pi = \hat{r} \cdot \mathsf{min}\,\pi'$. Finally, note that $\hat{r}$ is of size $O(|\pi|)$. We deduce immediately that $\mathsf{min}\,\pi$ is of size $O(|\pi|^2)$. □

Moreover, we associate with every path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ a formula $\mathsf{max}\,\pi$ in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$. Note that, for *finite* MSCs, we could define $\mathsf{max}\,\pi$ similarly to $\mathsf{min}\,\pi$. However, for *infinite* MSCs, we cannot use the same definition since we may have $\max[\![\pi_1 \cdot \pi_2]\!](e) = f$ but $\max[\![\pi_1 \cdot \{\langle\pi_2\rangle\}?]\!](e) = \infty$.

Lemma 6. *Let $R = \emptyset$ or $R = \{\mathsf{Loop}\}$. For every path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[R]$, there exists a $\mathrm{PDL}_{\mathsf{sf}}[R]$ path formula $\mathsf{max}\,\pi$ of size $O(|\pi|^2)$ such that, for all MSCs M and events e, f, we have $M, e, f \models \mathsf{max}\,\pi$ iff $f = \max[\![\pi]\!](e)$.*

In particular, if $\max[\![\pi]\!](e) = \infty$*, then no event in* M *will satisfy* $\mathsf{max}\,\pi$*. So we have* $\max[\![\pi]\!](e) = \infty$ *iff* $M, e \models \langle\pi\rangle \wedge \neg\langle \mathsf{max}\,\pi\rangle$.

Proof. As in Lemma 5, we view π as a nonempty sequence of atomic steps. If $\pi = r \cdot \pi'$ with r an atomic path formula, we will define inductively $\mathsf{max}\,\pi = \hat{r} \cdot \mathsf{max}\,\pi'$, where $\hat{r}$ is a path formula of size $O(|\pi|)$.

- Without loss of generality, we assume that the last atomic step of the path formula is $\{true\}?$. Hence, the basis of the induction is when $\pi = \{true\}?$, in which case we let $\mathsf{max}\,\pi = \{true\}?$.
- If $\pi = r \cdot \pi_1$, where $r \in \{\{\varphi\}?, \rightarrow, \leftarrow, \lhd_{p,q}, \lhd_{p,q}^{-1} \mid (p,q) \in Ch\}$, we let

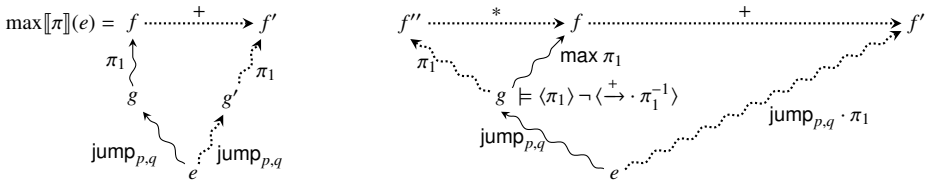
$$\mathsf{max}\,\pi = r \cdot \mathsf{max}\,\pi_1\,.$$

- If $\pi = \mathsf{jump}_{p,q} \cdot \pi_1$, we let

$$\mathsf{max}\,\pi = \mathsf{jump}_{p,q} \cdot \{\langle\pi_1\rangle \neg \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle\}? \cdot \mathsf{max}\,\pi_1\,.$$

  To prove the correctness, let M be an MSC, and e, f events of M.

  First, assume that $f = \max[\![\mathsf{jump}_{p,q} \cdot \pi_1]\!](e)$. Let g such that $M, e, g \models \mathsf{jump}_{p,q}$ and $M, g, f \models \pi_1$ (see Figure 6a). We must have $f = \max[\![\pi_1]\!](g)$, hence $M, g, f \models \mathsf{max}\,\pi_1$. Suppose that $M, g \not\models \langle\pi_1\rangle \neg \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle$. Then, in particular, $M, f \models \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle$, i.e., there exist $f' >_{\mathsf{proc}} f$ and g' such that $M, g', f' \models \pi_1$. Since $loc(f') = loc(f)$, we also have $loc(g') = loc(g) = q$. Hence $M, e, g' \models \mathsf{jump}_{p,q}$ and $M, e, f' \models \pi$, which contradicts the maximality of f.



(a) $f = \max[\![\pi]\!](e)$. (b) $M, e, f \models \mathsf{max}\,\pi$.

Figure 6: Proof of Lemma 6 where $\pi = \mathsf{jump}_{p,q} \cdot \pi_1$.

  Conversely, assume that $M, e, f \models \mathsf{max}\,\pi$. Let g such that $M, e, g \models \mathsf{jump}_{p,q}$, $M, g \models \langle\pi_1\rangle \neg \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle$, and $M, g, f \models \mathsf{max}\,\pi_1$ (see Figure 6b). Clearly, $M, e, f \models \mathsf{jump}_{p,q} \cdot \pi_1$. Suppose that f is not maximal, i.e., that there exists $f' >_{\mathsf{proc}} f$ such that $M, e, f' \models \pi$. Then, for all f'' such that $M, g, f'' \models \pi_1$, we have $f'' \leq_{\mathsf{proc}} f <_{\mathsf{proc}} f'$ (by induction hypothesis), hence $M, f'' \models \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle$. This contradicts the fact that $M, g \models \langle\pi_1\rangle \neg \langle \overset{+}{\rightarrow} \cdot \pi_1^{-1}\rangle$.

- If $\pi = \overset{\varphi}{\leftarrow} \cdot \pi_1$, we let

$$\mathsf{max}\,\pi = \overset{\varphi \wedge \neg\langle\pi_1\rangle}{\longleftarrow} \cdot \mathsf{max}\,\pi_1\,.$$

  Let M be an MSC, and e, f events of M. Assume that $f = \max[\![\pi]\!](e)$ and let g be such that $M, e, g \models \overset{\varphi}{\leftarrow}$ and $M, g, f \models \pi_1$. We must have $f = \max[\![\pi_1]\!](g)$. Let g' be maximal with $M, e, g' \models \overset{\varphi}{\leftarrow} \cdot \{\langle\pi_1\rangle\}?$. We have $M, e, g' \models \overset{\varphi \wedge \neg\langle\pi_1\rangle}{\longleftarrow}$ and $g \leq_{\mathsf{proc}} g'$. We deduce from Lemma 3(a) that $f = \max[\![\pi_1]\!](g')$.

  Conversely, assume that $M, e, f \models \mathsf{max}\,\pi$ and let g be such that $M, e, g \models \overset{\varphi \wedge \neg\langle\pi_1\rangle}{\longleftarrow}$ and $M, g, f \models \mathsf{max}\,\pi_1$. We have $M, e, f \models \pi$. Let f', g' be such that $M, e, f' \models \pi$, $M, e, g' \models \overset{\varphi}{\leftarrow}$, and $M, g', f' \models \pi_1$. We have $g' \leq_{\mathsf{proc}} g$ and using Lemma 3(a) we deduce that $f' \leq_{\mathsf{proc}} f$. Therefore, $f = \max[\![\pi]\!](e)$.

(a) $f = \max[\![\pi]\!](e)$ and $[\![\xrightarrow{\varphi}]\!](e)$ infinite.

(b) $M, e, f \models \mathsf{max}\,\pi$.

Figure 7: Proof of Lemma 6 where $\pi = \xrightarrow{\varphi} \cdot \pi_1$.

- If $\pi = \xrightarrow{\varphi} \cdot \pi_1$, we let

$$\mathsf{max}\,\pi = \xrightarrow{\varphi} \cdot \{\psi_1 \vee \psi_2\}? \cdot \mathsf{max}\,\pi_1\,, \qquad \text{where}$$
$$\psi_1 = \neg\varphi \vee \neg\langle \xrightarrow{\varphi} \cdot \pi_1 \rangle$$
$$\psi_2 = \langle \pi_1 \rangle \neg \langle \xrightarrow{+} \cdot \pi_1^{-1} \rangle\,.$$

  Let M be an MSC, and e, f events of M.

  Assume that $f = \max[\![\pi]\!](e)$, and that $[\![\xrightarrow{\varphi}]\!](e)$ is finite. Then, $[\![\xrightarrow{\varphi} \cdot \{\langle\pi_1\rangle\}?]\!](e)$ is also finite and non-empty. Let $g = \max[\![\xrightarrow{\varphi} \cdot \{\langle\pi_1\rangle\}?]\!](e)$. We have $M, g \models \psi_1$. In addition, since $[\![\pi]\!](e)$ is finite, $[\![\pi_1]\!](g)$ must be finite, and by Lemma 3(a), we get $\max[\![\pi_1]\!](g) = f$. Hence $M, e, f \models \mathsf{max}\,\pi$. Now, assume that $[\![\xrightarrow{\varphi}]\!](e)$ is infinite. Let g be any event such that $M, e, g \models \xrightarrow{\varphi}$ and $M, g, f \models \pi_1$. By maximality of f, we have $M, g, f \models \mathsf{max}\,\pi_1$ (see Figure 7a). Suppose towards a contradiction that $M, f \models \langle \xrightarrow{+} \cdot \pi_1^{-1} \rangle$. Then, there exist $f' >_{\mathsf{proc}} f$ and g' such that $M, g', f' \models \pi_1$. By Lemma 3(a), $g' >_{\mathsf{proc}} g >_{\mathsf{proc}} e$. Since $[\![\xrightarrow{\varphi}]\!](e)$ is infinite, we have $M, e, g' \models \xrightarrow{\varphi}$, and thus $M, e, f' \models \pi$, which contradicts the maximality of f. Hence, $M, f \models \neg\langle \xrightarrow{+} \cdot \pi_1^{-1} \rangle$, $M, g \models \psi_2$, and $M, e, f \models \mathsf{max}\,\pi$.

  Conversely, assume that $M, e, f \models \mathsf{max}\,\pi$. Let g such that $M, e, g \models \xrightarrow{\varphi}$, $M, g \models \psi_1 \vee \psi_2$, and $M, g, f \models \mathsf{max}\,\pi_1$. If $g \models \psi_1$, we have $g = \max[\![\xrightarrow{\varphi} \cdot \{\langle\pi_1\rangle\}?]\!](e)$. By Lemma 3(a), we conclude that $f = \max[\![\pi]\!](e)$. Now, suppose that $M, g \models \psi_2$, and that there exists $f' >_{\mathsf{proc}} f$ such that $M, e, f' \models \pi$ (see Figure 7b). For all f'' such that $M, g, f'' \models \pi_1$, we have $f'' \leq_{\mathsf{proc}} f <_{\mathsf{proc}} f'$, hence $M, f'' \models \langle \xrightarrow{+} \cdot \pi_1^{-1} \rangle$. This contradicts the fact that $M, g \models \langle\pi_1\rangle \neg \langle \xrightarrow{+} \cdot \pi_1^{-1} \rangle$. □

We are now ready to prove that any Boolean combination of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formulas is equivalent to a positive one, i.e., one that does not use complement.

Lemma 7. *For all path formulas* $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$*, there exist* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *path formulas* $(\pi_i)_{1 \leq i \leq |P|^2+3}$ *such that* $\pi^{\mathsf{c}} \equiv \bigcup_{1 \leq i \leq |P|^2+3} \pi_i$.

Proof. We show $\pi^{\mathsf{c}} \equiv \sigma$, where

$$\sigma = (\mathsf{min}\,\pi \cdot \xleftarrow{+}) \cup (\mathsf{max}\,\pi \cdot \xrightarrow{+}) \cup (\pi \cdot \xrightarrow{+} \cdot \{\neg\langle\pi^{-1}\rangle\}?) \cup \bigcup_{(p,q) \in P^2} \{\neg\langle\pi\rangle\, q\}? \cdot \mathsf{jump}_{p,q}\,.$$

Let $M = (E, \rightarrow, \lhd, loc, \lambda)$ be an MSC and $e, f \in E$. We write $p = loc(e)$, $q = loc(f)$. Let us show that $M, e, f \models \pi^c$ iff $M, e, f \models \sigma$. If $M, e \models \neg\langle\pi\rangle\, q$, then both $M, e, f \models \pi^{\mathsf{c}}$ and $M, e, f \models \sigma$ hold. In the following, we assume that $M, e \models \langle\pi\rangle\, q$, and thus that $\min[\![\pi]\!](e) \in E_q$ and $\max[\![\pi]\!](e) \in E_q \cup \{\infty\}$. Again, if $f <_{\mathsf{proc}} \min[\![\pi]\!](e)$ or $\max[\![\pi]\!](e) <_{\mathsf{proc}} f$, then both $M, e, f \models \pi^{\mathsf{c}}$ and $M, e, f \models \sigma$ hold. And if $\min[\![\pi]\!](e) \leq_{\mathsf{proc}} f \leq_{\mathsf{proc}} \max[\![\pi]\!](e)$, then, by Lemma 4, we have $M, e, f \models \pi^c$ iff $M, f \models \neg\langle\pi^{-1}\rangle$, iff $M, e, f \models \sigma$. □

3.5. From FO *to* $\mathrm{PDL}_{\mathsf{sf}}$

We will now show that every $\mathrm{FO}[\rightarrow, \lhd, \leq]$ formula with at most two free variables can be translated into an equivalent $\mathrm{PDL}_{\mathsf{sf}}$ formula, as stated in Theorem 1 below. As we proceed by induction, we actually need a more general statement, which takes into account arbitrarily many free variables. In the following proposition, $\widetilde{\pi}(x, y)$ refers to the FO formula obtained from π due to Proposition 1. To obtain a formula $\widetilde{\pi}(x, x)$ with one free variable, we first construct $\widetilde{\pi}(x, y)$ according to Proposition 1 and then replace y by x.

Proposition 2. *Every formula $\Phi \in \mathrm{FO}[\rightarrow, \lhd, \leq]$ with at least one free variable is equivalent to a positive Boolean combination of formulas of the form $\widetilde{\pi}(x, y)$, where $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ and $x, y \in \mathsf{Free}(\Phi)$.*

Proof. In the following, we will simply write $\pi(x, y)$ for $\widetilde{\pi}(x, y)$.

The proof is by induction. For convenience, we assume that Φ is in prenex normal form. If Φ is quantifier free, then it is a Boolean combination of atomic formulas. For $x, y \in \mathcal{V}_{\mathsf{event}}$, atomic formulas are translated as follows:

$$p(x) \equiv \{p\}?(x, x) \qquad x \rightarrow y \equiv \rightarrow(x, y) \qquad x = y \equiv \{true\}?(x, y)$$

$$a(x) \equiv \{a\}?(x, x) \qquad x \lhd y \equiv \bigvee_{(p,q)\in Ch} \lhd_{p,q}(x, y)$$

Moreover, $x \leq y$ is equivalent to the disjunction of the formulas $(\pi \cdot \lhd_{p_1,p_2} \cdot \overset{+}{\rightarrow} \cdot \lhd_{p_2,p_3} \cdots \overset{+}{\rightarrow} \cdot \lhd_{p_{m-1},p_m} \cdot \pi')(x, y)$, where $1 \leq m \leq |P|$, $p_1, \ldots, p_m \in P$ are such that $p_i \neq p_j$ for all $1 \leq i < j \leq m$, and $\pi, \pi' \in \{\overset{+}{\rightarrow}, \{true\}?\}$.

Universal quantification. We have $\forall x.\Psi \equiv \neg\exists x.\neg\Psi$. Negation can be eliminated thanks to Lemma 7. Hence, this case reduces to existential quantification.

Existential quantification. Suppose that $\Phi = \exists x.\Psi$. If x is not free in Ψ, then $\Phi \equiv \Psi$ and we are done by induction. Otherwise, assume that $\mathsf{Free}(\Psi) = \{x_1, \ldots, x_n\}$ with $n > 1$, and that $x = x_n$. By induction, Ψ is equivalent to a positive Boolean combination of formulas of the form $\pi(y, z)$ with $y, z \in \mathsf{Free}(\Psi)$. Bringing Ψ into disjunctive normal form, we obtain a finite disjunction of formulas of the form $\bigwedge_j \pi_j(y_j, z_j)$, where $y_j = x_{i_1}$ and $z_j = x_{i_2}$ for some $i_1 \leq i_2$. This step may cause an exponential blow-up so that the overall construction is nonelementary (which is unavoidable [Sto74]). Note that the variable ordering can be guaranteed by replacing π_j with π_j^{-1} whenever needed.

Now, $\Phi = \exists x_n.\Psi$ is equivalent to a finite disjunction of formulas of the form

$$\bigwedge_{j\in I} \pi_j(y_j, z_j) \;\wedge\; \underbrace{\exists x_n.\Big(\bigwedge_{j\in J} \pi_j(y_j, x_n) \wedge \bigwedge_{j\in J'} \pi_j(x_n, x_n)\Big)}_{=:\,\Upsilon}$$

for three finite, pairwise disjoint index sets I, J, J' such that $y_j, z_j \in \{x_1, \ldots, x_{n-1}\}$ for all $j \in I$, and $y_j \in \{x_1, \ldots, x_{n-1}\}$ for all $j \in J$. Notice that $\mathsf{Free}(\Upsilon) \subseteq \{x_1, \ldots, x_{n-1}\}$. If $J = \emptyset$, then[3]

$$\Upsilon \equiv \bigvee_{p,q\in P} \Big(\mathsf{jump}_{p,q} \cdot \{\bigwedge_{j\in J'} \mathsf{Loop}(\pi_j)\}? \cdot \mathsf{jump}_{q,p}\Big)(x_1, x_1)\,.$$

So assume $J \neq \emptyset$. We define below a formula Υ' and prove that it is equivalent to Υ. Intuitively, by Lemma 4, we know that Υ holds iff the intersection of the intervals $[\min[\![\pi_j]\!](y_j), \max[\![\pi_j]\!](y_j)]$ contains some event satisfying $\psi = \bigwedge_{j\in J} \langle \pi_j^{-1} \rangle \wedge \bigwedge_{j\in J'} \mathsf{Loop}(\pi_j)$. The formula Υ' identifies some π_k such that $\min[\![\pi_k]\!](y_k)$ is maximal (first line), some π_ℓ such that $\max[\![\pi_\ell]\!](y_\ell)$ is minimal (second line), and tests that there exists an event x_n satisfying ψ between

[3] In this case, Υ is a sentence whereas x_1 is free in the right hand side. Notice that $\equiv$ does not require the two formulas to have the same free variables.

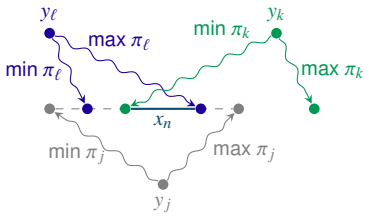


Figure 8: Proof of Claim 1.

the two (third line). This is illustrated in Figure 8.

$$\Upsilon' := \bigvee_{k,\ell \in J} \left(\begin{array}{ll} & \bigwedge_{j\in J}((\mathsf{min}\,\pi_j)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{min}\,\pi_k)^{-1})(y_j,y_k) \\ \wedge & \bigwedge_{j\in J}\Big(\{\langle \pi_j\rangle \wedge \neg\langle \mathsf{max}\,\pi_j\rangle\}?(y_j,y_j) \vee ((\mathsf{max}\,\pi_\ell)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{max}\,\pi_j)^{-1})(y_\ell,y_j)\Big) \\ \wedge & (\pi_k \cdot \{\psi\}? \cdot \pi_\ell^{-1})(y_k,y_\ell) \end{array} \right)$$

Claim 1. *We have* $\mathsf{Free}(\Upsilon') = \mathsf{Free}(\Upsilon) \subseteq \{x_1,\ldots,x_{n-1}\}$ *and* $\Upsilon \equiv \Upsilon'$.

Proof. Assume $M, \nu \models \Upsilon$. There exists $e \in E$ such that $M, \nu(y_j), e \models \pi_j$ for all $j \in J$, and $M, e \models \mathsf{Loop}(\pi_j)$ for all $j \in J'$. In particular, all $\min[\![\pi_j]\!](\nu(y_j))$ and $\max[\![\pi_j]\!](\nu(y_j)) \in E \cup \{\infty\}$ for $j \in J$ are well-defined and on process $loc(e)$ or equal to ∞. Let $k \in J$ such that $\min[\![\pi_k]\!](\nu(y_k))$ is maximal, i.e., $\min[\![\pi_j]\!](\nu(y_j)) \leq_{\mathsf{proc}} \min[\![\pi_k]\!](\nu(y_k))$ for all $j \in J$. Then, for all $j \in J$, we have $M, \nu(y_j), \nu(y_k) \models (\mathsf{min}\,\pi_j)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{min}\,\pi_k)^{-1}$. Similarly, let $\ell \in J$ such that $\max[\![\pi_\ell]\!](\nu(y_\ell)) \in E \cup \{\infty\}$ is minimal. Then, for all $j \in J$, either $M, \nu(y_j), \nu(y_j) \models \{\langle \pi_j\rangle \wedge \neg\langle \mathsf{max}\,\pi_j\rangle\}?$ (i.e., $\max[\![\pi_j]\!](\nu(y_j)) = \infty$), or else $M, \nu(y_\ell), \nu(y_j) \models (\mathsf{max}\,\pi_\ell)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{max}\,\pi_j)^{-1}$. In addition, we have $M, e \models \psi$, $M, \nu(y_k), e \models \pi_k$, and $M, \nu(y_\ell), e \models \pi_\ell$. Hence, $M, \nu(y_k), \nu(y_\ell) \models \pi_k \cdot \{\psi\}? \cdot \pi_\ell^{-1}$. So we have $M, \nu \models \Upsilon'$.

Conversely, assume $M, \nu \models \Upsilon'$. Let $k, \ell \in J$ such that the corresponding sub-formula is satisfied. There exists $e \in E$ such that $M, \nu(y_k), e \models \pi_k$, $M, e \models \psi$, and $M, e, \nu(y_\ell) \models \pi_\ell^{-1}$. Note that we have $\min[\![\pi_k]\!](\nu(y_k)) \leq_{\mathsf{proc}} e \leq_{\mathsf{proc}} \max[\![\pi_\ell]\!](\nu(y_\ell))$. For all $j \in J'$, we have $M, e \models \mathsf{Loop}(\pi_j)$, i.e., $M, \nu[x_n \mapsto e] \models \pi_j(x_n, x_n)$. Now, let $j \in J$. We have $M, \nu(y_j), \nu(y_k) \models (\mathsf{min}\,\pi_j)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{min}\,\pi_k)^{-1}$, hence $\min[\![\pi_j]\!](\nu(y_j)) \leq_{\mathsf{proc}} \min[\![\pi_k]\!](\nu(y_k)) \leq_{\mathsf{proc}} e$. Similarly, $e \leq_{\mathsf{proc}} \max[\![\pi_\ell]\!](\nu(y_\ell)) \leq_{\mathsf{proc}} \max[\![\pi_j]\!](\nu(y_j)) \in E \cup \{\infty\}$. In addition, since $M, e \models \psi$, we have $M, e \models \langle \pi_j^{-1}\rangle$. Applying Lemma 4, we get $M, \nu(y_j), e \models \pi_j$, i.e., $M, \nu[x_n \mapsto e] \models \pi_j(y_j, x_n)$. Hence, $M, \nu \models \Upsilon$. □(Claim 1)

We conclude that Υ is equivalent to some positive Boolean combination of formulas $\pi(x,y)$, with $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ and $x, y \in \{x_1,\ldots,x_{n-1}\} = \mathsf{Free}(\Phi)$. Therefore, so is Φ. Note that, due to $\overset{*}{\rightarrow}$, the formulas $(\mathsf{min}\,\pi_j)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{min}\,\pi_k)^{-1}$ and $(\mathsf{max}\,\pi_\ell)\cdot \overset{*}{\rightarrow} \cdot (\mathsf{max}\,\pi_j)^{-1}$ are in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ instead of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$. By Lemma 2, these can be transformed into finite unions of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ path formulas. □

We are now able to prove the main result relating $\mathrm{FO}[\rightarrow, \lhd, \leq]$ and $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$.

Theorem 1. *Every* $\mathrm{FO}[\rightarrow, \lhd, \leq]$ *formula with at most two free variables is equivalent to some* $\mathrm{PDL}_{\mathsf{sf}}$ *formula. More precisely, for every* $\mathrm{FO}[\rightarrow, \lhd, \leq]$ *sentence* Φ_0*, formula* $\Phi_1(x)$ *with one free variable, and formula* $\Phi_2(x,y)$ *with two free variables, there exist some* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *sentence* ξ*,* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *event formula* φ*, and* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *path formulas* π_{ij}*, respectively, such that,* $\Phi_0 \equiv \xi$*,* $\Phi_1(x) \equiv \varphi$*, and* $\Phi_2(x,y) \equiv \bigcup_i \bigcap_j \pi_{ij}$.

Proof. Let $\Phi_2(x_1,x_2)$ be an $\mathrm{FO}[\rightarrow, \lhd, \leq]$ formula with two free variables. We apply Proposition 2 to $\Phi_2(x_1,x_2)$ and obtain a positive Boolean combination of path formulas $\pi(y,z)$ with $y, z \in \{x_1, x_2\}$. Next, we replace formula $\pi(x_1,x_1)$ by $\bigvee_{p,q}(\{\mathsf{Loop}(\pi)\}? \cdot \mathsf{jump}_{p,q})(x_1,x_2)$. Similarly, $\pi(x_2,x_2)$ is replaced by $\bigvee_{p,q}(\mathsf{jump}_{p,q} \cdot \{\mathsf{Loop}(\pi)\}?)(x_1,x_2)$. Also, $\pi(x_2,x_1)$ is replaced by $\pi^{-1}(x_1,x_2)$. Finally, we transform it into disjunctive normal form: we obtain $\Phi_1(x_1,x_2) \equiv \bigvee_i \bigwedge_j \pi_{ij}(x_1,x_2)$, which concludes the proof in the case of two free variables.

Next, let $\Phi_1(x)$ be an FO$[\rightarrow, \lhd, \leq]$ formula with one free variable. As above, applying Proposition 2 to $\Phi_1(x)$, we obtain PDL$_{\mathsf{sf}}$[Loop] path formulas π_{ij} such that $\Phi_1(x) \equiv \bigvee_i \bigwedge_j \pi_{ij}(x,x)$. Now, $M, [x \mapsto e] \models \pi_{ij}(x,x)$ iff $M, e \models \mathsf{Loop}(\pi_{ij})$. Hence, $\Phi(x) \equiv \bigvee_i \bigwedge_j \mathsf{Loop}(\pi_{ij})$.

Finally, an FO$[\rightarrow, \lhd, \leq]$ sentence Φ_0 is a Boolean combination of formulas of the form $\exists x.\Phi_1(x)$. Applying the theorem to $\Phi_1(x)$, we obtain an equivalent PDL$_{\mathsf{sf}}$[Loop] event formula φ. Then, we take $\xi = \mathsf{E}\,\varphi$, which is trivially equivalent to $\exists x.\Phi_1(x)$. □

From Theorem 1 and Proposition 1, we deduce that FO has the three variable property:

Corollary 1. $\mathcal{L}(\text{FO}[\rightarrow, \lhd, \leq]) = \mathcal{L}(\text{FO}^3[\rightarrow, \lhd, \leq])$.

4. From PDL$_{\mathsf{sf}}$[Loop] to CFMs

In this section, we show that, from a PDL$_{\mathsf{sf}}$[Loop] sentence, we can effectively construct an equivalent CFM of exponential size (Theorem 2). Together with Theorem 1, this implies that every FO sentence can be translated to an equivalent CFM (Theorem 3).

In the inductive translation of PDL$_{\mathsf{sf}}$[Loop] formulas into CFMs, *event* formulas will be evaluated by *MSC transducers*. An MSC transducer for an event formula φ produces a truth value at every event on the given MSC. More precisely, it outputs 1 when φ holds at the current event, and 0 otherwise. We introduce MSC transducers formally in the next section. Then, we present the actual translation of PDL$_{\mathsf{sf}}$[Loop] event formulas into MSC transducers in Section 4.2. We concluce in Section 4.3 with the translation of sentences, PDL$_{\mathsf{sf}}$[Loop] or FO, into CFMs.

4.1. Letter-to-letter MSC Transducers

Let Γ be a nonempty finite output alphabet. A *(nondeterministic) letter-to-letter MSC transducer* (or simply, *transducer*) $\mathcal{A}$ over P and from Σ to Γ is a CFM over P and $\Sigma \times \Gamma$. The transducer $\mathcal{A}$ accepts the relation

$$[\![\mathcal{A}]\!] = \{((E, \rightarrow, \lhd, loc, \lambda), (E, \rightarrow, \lhd, loc, \gamma)) \mid (E, \rightarrow, \lhd, loc, \lambda \times \gamma) \in \mathbb{L}(\mathcal{A})\}\,.$$

Transducers are closed under product and composition, using standard constructions:

Lemma 8. *Let $\mathcal{A}$ be a transducer from Σ to Γ, and $\mathcal{A}'$ a transducer from Σ to Γ'. There exists a transducer $\mathcal{A} \times \mathcal{A}'$ from Σ to $\Gamma \times \Gamma'$ such that*

$$\begin{aligned}[\![\mathcal{A} \times \mathcal{A}']\!] = \{((E, \rightarrow, \lhd, loc, \lambda), (E, \rightarrow, \lhd, loc, \gamma \times \gamma')) \mid \\ ((E, \rightarrow, \lhd, loc, \lambda), (E, \rightarrow, \lhd, loc, \gamma)) \in [\![\mathcal{A}]\!], \\ ((E, \rightarrow, \lhd, loc, \lambda), (E, \rightarrow, \lhd, loc, \gamma')) \in [\![\mathcal{A}']\!]\}\,.\end{aligned}$$

Lemma 9. *Let $\mathcal{A}$ be a transducer from Σ to Γ, and $\mathcal{A}'$ a transducer from Γ to Γ'. There exists a transducer $\mathcal{A}' \circ \mathcal{A}$ from Σ to Γ' such that*

$$[\![\mathcal{A}' \circ \mathcal{A}]\!] = [\![\mathcal{A}']\!] \circ [\![\mathcal{A}]\!] = \{(M, M'') \mid \exists M' \in \mathbb{MSC}(P, \Gamma) : (M, M') \in [\![\mathcal{A}]\!], (M', M'') \in [\![\mathcal{A}']\!]\}\,.$$

4.2. Translation of PDL$_{\mathsf{sf}}$[Loop] *Event Formulas into MSC Transducers*

For a PDL$_{\mathsf{sf}}$[Loop] event formula φ and an MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$ over P and Σ, we define an MSC $M_\varphi = (E, \rightarrow, \lhd, loc, \gamma)$ over P and $\{0,1\}$, by setting $\gamma(e) = 1$ if $M, e \models \varphi$, and $\gamma(e) = 0$ otherwise.

The goal of this section is to show that (Proposition 3), from any PDL$_{\mathsf{sf}}$[Loop] event formula φ, we can construct an MSC transducer $\mathcal{A}_\varphi$ of exponential size which is equivalent to φ, that is, $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.

We start with the case of formulas from PDL$_{\mathsf{sf}}[\emptyset]$, i.e., without Loop. Lemma 10 actually follows from [BKM10, Theorem 4.16] since PDL$_{\mathsf{sf}}[\emptyset]$ is a restricted fragment of the (loop-free) logic studied in [BKM10]. For completeness, we provide a proof of the following simpler lemma.

Lemma 10. *Let φ be a* PDL$_{\mathsf{sf}}[\emptyset]$ *event formula. There exists a transducer $\mathcal{A}_\varphi$ with $2^{O(|\varphi|)}$ states per process such that $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.*

Proof. Any $\mathrm{PDL}_{\mathsf{sf}}[\emptyset]$ event formula is equivalent to some linear-size formula φ over the syntax

$$\varphi ::= p \mid a \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \lhd_{p,q} \rangle \varphi \mid \langle \lhd_{p,q}^{-1} \rangle \varphi \mid \langle \xrightarrow{\varphi} \rangle \varphi \mid \langle \xleftarrow{\varphi} \rangle \varphi \mid \langle \mathsf{jump}_{p,q} \rangle \varphi$$

Indeed, we have $\langle \pi_1 \cdot \pi_2 \rangle \varphi \equiv \langle \pi_1 \rangle (\langle \pi_2 \rangle \varphi)$, and $\langle \{\varphi\}? \rangle \psi \equiv \varphi \wedge \psi$. Notice that $\rightarrow \equiv \xrightarrow{\mathit{false}}$ and $\leftarrow \equiv \xleftarrow{\mathit{false}}$.

We define $\mathcal{A}_\varphi$ by induction on φ, by composition of the transducers for the atomic formulas $\varphi = p$ with $p \in P$, or $\varphi = a$ with $a \in \Sigma$, and of transducers $\mathcal{B}_\vee$, $B_\neg$, $\mathcal{B}_{\lhd_{p,q}}$, $\mathcal{B}_{\lhd_{p,q}^{-1}}$, $\mathcal{B}_{\mathsf{jump}_{p,q}}$, $\mathcal{B}_{\mathsf{XU}}$, and $\mathcal{B}_{\mathsf{YS}}$ corresponding to each construct of the logic. These transducers are defined in Figure 9. For instance, the transducer $\mathcal{B}_\neg$ from $\{0,1\}$ to $\{0,1\}$ outputs the negation of the bit read and $\mathcal{B}_\vee$ from $\{0,1\}^2$ to $\{0,1\}$ outputs the disjunction of the two bits read. The transducer $\mathcal{B}_{\lhd_{p,q}}$ from $\{0,1\}$ to $\{0,1\}$ outputs 1 at an event e iff e is a send event from p to q and the corresponding receive event f is labeled 1. The transducers $\mathcal{B}_{\lhd_{p,q}^{-1}}$ and $\mathcal{B}_{\mathsf{jump}_{p,q}}$ are defined similarly. The deterministic transducer $\mathcal{B}_{\mathsf{YS}}$ from $\{0,1\}^2$ to $\{0,1\}$ corresponds to the *strict since* modality. On each process, it outputs 1 at some event e if there is $g <_{\mathsf{proc}} e$, where the second bit is 1 and for all $g <_{\mathsf{proc}} f <_{\mathsf{proc}} e$ the first bit at f is 1. The transducer $\mathcal{B}_{\mathsf{XU}}$ corresponds to the reverse *strict until* modality. We then let

$$\begin{aligned}
\mathcal{A}_{\varphi_1 \vee \varphi_2} &= \mathcal{B}_\vee \circ (\mathcal{A}_{\varphi_1} \times \mathcal{A}_{\varphi_2}) & \mathcal{A}_{\neg\varphi} &= \mathcal{B}_\neg \circ \mathcal{A}_\varphi \\
\mathcal{A}_{\langle \lhd_{p,q} \rangle \varphi} &= \mathcal{B}_{\lhd_{p,q}} \circ \mathcal{A}_\varphi & \mathcal{A}_{\langle \lhd_{p,q}^{-1} \rangle \varphi} &= \mathcal{B}_{\lhd_{p,q}^{-1}} \circ \mathcal{A}_\varphi \\
\mathcal{A}_{\langle \xrightarrow{\varphi_1} \rangle \varphi_2} &= \mathcal{B}_{\mathsf{XU}} \circ (\mathcal{A}_{\varphi_1} \times \mathcal{A}_{\varphi_2}) & \mathcal{A}_{\langle \mathsf{jump}_{p,q} \rangle \varphi} &= \mathcal{B}_{\mathsf{jump}_{p,q}} \circ \mathcal{A}_\varphi \\
\mathcal{A}_{\langle \xleftarrow{\varphi_1} \rangle \varphi_2} &= \mathcal{B}_{\mathsf{YS}} \circ (\mathcal{A}_{\varphi_1} \times \mathcal{A}_{\varphi_2}) .
\end{aligned}$$

This concludes the proof of Lemma 10. □

Next, we look at a single loop where the path $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\emptyset]$ is *functional*. We call a path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}$ functional if, for all MSCs M and events e in M, $[\![\pi]\!](e)$ is either empty or a singleton. Abusing notation, when $[\![\pi]\!](e) \neq \emptyset$, we simply write $[\![\pi]\!](e) = e'$ instead of $[\![\pi]\!](e) = \{e'\}$.

We say that a functional path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}$ is *monotone* if, for all MSCs M and events e, f such that $[\![\pi]\!](e) \neq \emptyset$, $[\![\pi]\!](f) \neq \emptyset$, and $e \leq_{\mathsf{proc}} f$, we have $[\![\pi]\!](e) \leq_{\mathsf{proc}} [\![\pi]\!](f)$.

Notice that, for all path formulas $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$, the path formulas $\mathsf{min}\ \pi$ and $\mathsf{max}\ \pi$ are functional. Moreover, as a direct consequence of Lemma 3(a), we obtain:

Lemma 11. *All* functional $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *path formulas are monotone.*

Lemma 12. *Let π be a* $\mathrm{PDL}_{\mathsf{sf}}[\emptyset]$ *functional path formula, and $\varphi = \mathsf{Loop}(\pi)$. There exists a transducer $\mathcal{A}_\varphi$ with $2^{O(|\varphi|)}$ states per process such that $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.*

Proof. We can assume that $\mathsf{Comp}(\pi) \subseteq \mathsf{id}$. We define $\mathcal{A}_\varphi$ as the composition of three transducers that will guess and check the evaluation of φ. More precisely, $\mathcal{A}_\varphi$ will be obtained as an inverse projection α^{-1}, followed by the intersection with some MSC language K, followed by a projection β.

We first enrich the labeling of the MSC with a color from $\Theta = \{\square, \blacksquare, \bigcirc, \bullet\}$. Intuitively, colors $\square$ and $\blacksquare$ will correspond to a guess that the formula φ is satisfied, and colors $\bigcirc$ and $\bullet$ to a guess that the formula is not satisfied. We will construct a CFM that enforces a coloring that, at every event, correctly reflects the truth value of φ. We require that labels from $\{\square, \blacksquare\}$ alternate on a process (Condition 1. below) and that, moreover, for every event e with a color from $\{\square, \blacksquare\}$, there exist a π-successor and a π^{-1}-successor that both have the same color (Condition 2.). This will then ensure that an event with color from $\{\square, \blacksquare\}$ satisfies φ. Moreover, for every $\{\bigcirc, \bullet\}$-colored event e that has both a π-successor f and a π^{-1}-successor f', the colors of e, f, and f' should not coincide (again, Condition 2.). This, in turn, ensures that e does not satisfy φ. Let us formalize these ideas.

Consider the projection $\alpha \colon \mathbb{MSC}(P, \Sigma \times \Theta) \to \mathbb{MSC}(P, \Sigma)$ which erases the color from the labeling. The inverse projection α^{-1} can be realized with a transducer $\mathcal{A}$, i.e., $[\![\mathcal{A}]\!] = \{(\alpha(M'), M') \mid M' \in \mathbb{MSC}(P, \Sigma \times \Theta)\}$.

Define the projection $\beta \colon \mathbb{MSC}(P, \Sigma \times \Theta) \to \mathbb{MSC}(P, \{0,1\})$ by $\beta((E, \rightarrow, \lhd, loc, \lambda \times \theta)) = (E, \rightarrow, \lhd, loc, \gamma)$, where $\gamma(e) = 1$ if $\theta(e) \in \{\square, \blacksquare\}$, and $\gamma(e) = 0$ otherwise. The projection β can be realized with a transducer $\mathcal{A}''$, i.e., $[\![\mathcal{A}'']\!] = \{(M', \beta(M')) \mid M' \in \mathbb{MSC}(P, \Sigma \times \Theta)\}$.

Finally, consider the language $K \subseteq \mathbb{MSC}(P, \Sigma \times \Theta)$ of MSCs $M' = (E, \rightarrow, \lhd, loc, \lambda \times \theta)$ satisfying the following two conditions:

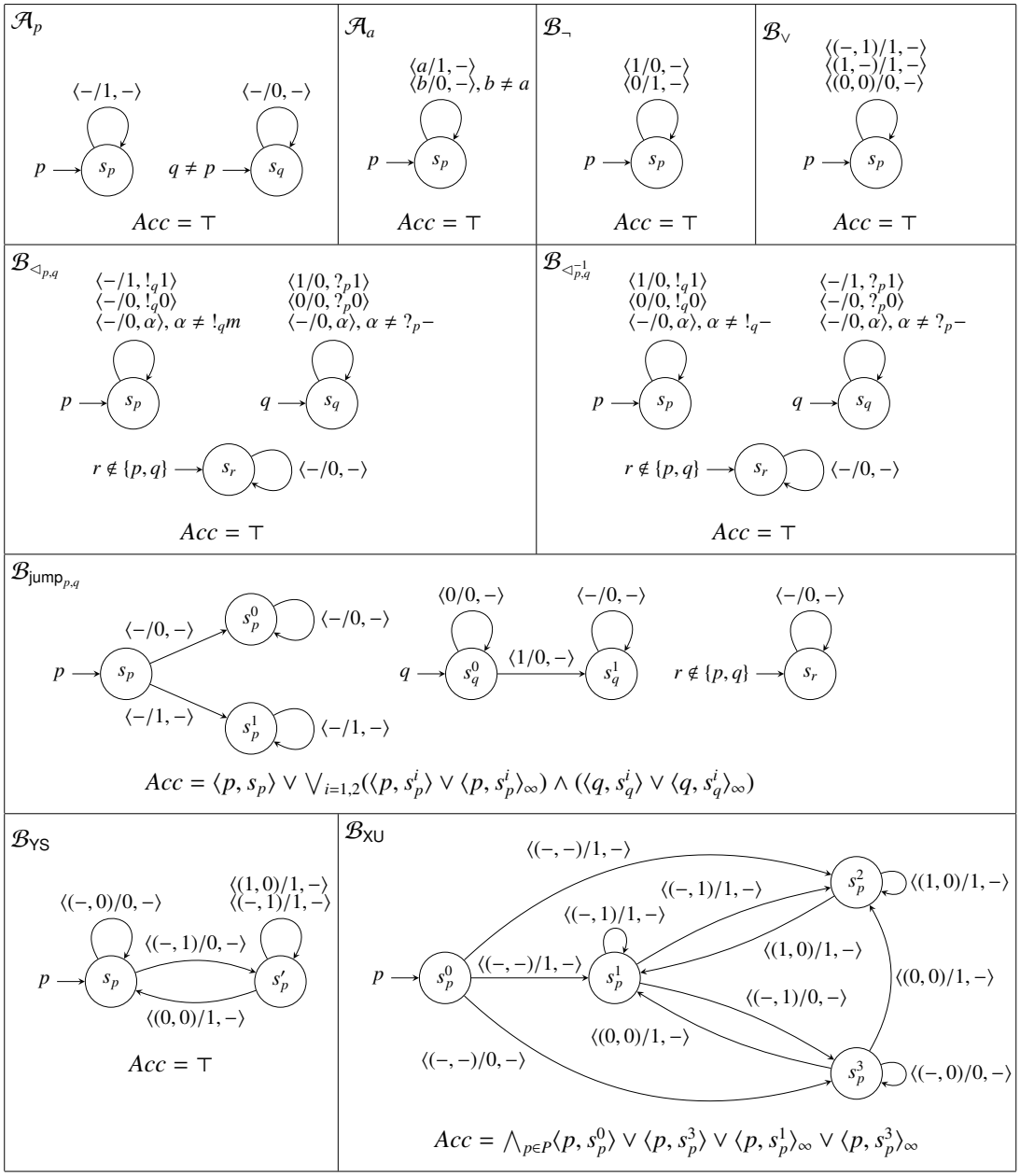


Figure 9: Transducers used to define $\mathcal{A}_\varphi$. In a transition labeled $\langle a/b, \alpha \rangle$, a is the input letter, b is the output letter, and α is either empty or a read or write action. Notice that in $\mathcal{B}_{\lhd_{p,q}}$, the automaton for process q has no transitions reading $\langle c/0, ?_p d \rangle$, with $c \neq d$, hence a wrong guess by process p cannot lead to an accepting run. The acceptance condition $\top$ means *true* and is always satisfied.

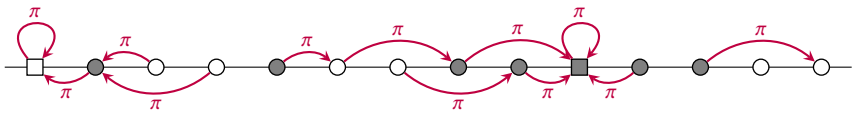


Figure 10: Proof of Claim 2: 2-coloring of E_0 in Graph G.

1. Colors □ and ■ alternate on each process $p \in P$: if $e_1 < e_2 < e_3 < \cdots$ are the events in $E_p \cap \theta^{-1}(\{\square, \blacksquare\})$, then $\theta(e_i) = \square$ if i is odd, and $\theta(e_i) = \blacksquare$ if i is even.
2. For all $e \in E$, $\theta(e) \in \{\square, \blacksquare\}$ iff there exist $f, f' \in E$ such that $M, e, f \models \pi$, $M, e, f' \models \pi^{-1}$, and $\theta(e) = \theta(f) = \theta(f')$.

The first property is trivial to check with a CFM. Using Lemma 10, we show that the second property can also be checked with a CFM. First, from π we construct a $\mathrm{PDL}_{\mathsf{sf}}[\emptyset]$ event formula ψ over P and $\Sigma \times \Theta$ such that, for all $M' = (E, \to, \lhd, loc, \lambda \times \theta) \in \mathbb{MSC}(P, \Sigma \times \Theta)$ and events $e \in E$, we have $M', e \models \psi$ iff the following holds: $\theta(e) \in \{\square, \blacksquare\}$ iff there are $f, f' \in E$ such that $\theta(e) = \theta(f) = \theta(f')$, $\alpha(M'), e, f \models \pi$, and $\alpha(M'), e, f' \models \pi^{-1}$. Namely, we define

$$\psi = (\square \vee \blacksquare) \iff \bigvee_{col \in \{\square, \blacksquare, \bigcirc, \bullet\}} col \wedge \langle \hat{\pi} \rangle col \wedge \langle \hat{\pi}^{-1} \rangle col$$

where the state formula col from $\{\square, \blacksquare, \bigcirc, \bullet\}$ is an abbreviation for $\bigvee_{a \in \Sigma}(a, col)$ and $\hat{\pi}$ is obtained from π by replacing state formulas a by $\bigvee_{col \in \Theta}(a, col)$. Now, the language for the second condition is $\{M' \in \mathbb{MSC}(P, \Sigma \times \Theta) \mid \text{every event of } M'_\psi \text{ is labeled with 1}\}$, for which we can easily give a CFM using the transducer $\mathcal{A}_\psi$ from $\Sigma \times \Theta$ to $\{0, 1\}$ given by Lemma 10.

We deduce that there is a transducer $\mathcal{A}'$ such that $[\![\mathcal{A}']\!] = \{(M', M') \mid M' \in K\}$. We let $\mathcal{A}_\varphi = \mathcal{A}'' \circ \mathcal{A}' \circ \mathcal{A}$. Notice that $[\![\mathcal{A}_\varphi]\!] = \{(\alpha(M'), \beta(M')) \mid M' \in K\}$. From the following two claims, we deduce immediately that $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.

Claim 2. *For all $M \in \mathbb{MSC}(P, \Sigma)$, there exists $M' \in K$ with $\alpha(M') = M$.*

Proof of Claim 2. Let $M = (E, \to, \lhd, loc, \lambda) \in \mathbb{MSC}(P, \Sigma)$. Let $E_1 = \{e \in E \mid M, e \models \varphi\}$ and $E_0 = E \setminus E_1$. Consider the graph $G = (E, \{(e, f) \mid M, e, f \models \pi\})$. Since π is functional, every vertex has outdegree at most 1, and, by Lemma 11, there are no cycles except for self-loops. So the restriction of G to E_0 is a forest, and there exists a 2-coloring $\chi \colon E_0 \to \{\bigcirc, \bullet\}$ such that, for all $e, f \in E_0$ with $M, e, f \models \pi$, we have $\chi(e) \neq \chi(f)$. This is illustrated in Figure 10. Moreover, there exists $\theta \colon E \to \Theta$ such that $\theta(e) = \chi(e)$ for $e \in E_0$, and $\theta(e) \in \{\square, \blacksquare\}$ for $e \in E_1$ is such that Condition 1 of the definition of K is satisfied. It is easy to see that Condition 2 is also satisfied. Indeed, if $\theta(e) \in \{\square, \blacksquare\}$, then $e \in E_1$, $M, e, e \models \pi$, and $M, e, e \models \pi^{-1}$. Now, if $\theta(e) \notin \{\square, \blacksquare\}$, then $e \in E_0$ and either $M, e \not\models \langle \pi \rangle$ or, by definition of θ, we have $\theta(e) \neq \theta(f)$ for the unique f such that $M, e, f \models \pi$. □(Claim 2)

Claim 3. *For all $M' \in K$, we have $\beta(M') = M_\varphi$, where $M = \alpha(M')$.*

Proof of Claim 3. Let $M' = (E, \to, \lhd, loc, \lambda \times \theta) \in K$ and $M = \alpha(M')$. Suppose towards a contradiction that $M_\varphi \neq \beta(M') = (E, \to, \lhd, loc, \gamma)$.

First, we show that, for all $e \in E$, $\gamma(e) = 0$ implies $M, e \not\models \varphi$. So assume $\gamma(e) = 0$. Then, we have $\theta(e) \in \{\bigcirc, \bullet\}$. Take any $f, f' \in E$ such that $M, e, f \models \pi$ and $M, e, f' \models \pi^{-1}$ (if there are no such events, we have $M, e \not\models \varphi$). Due to Condition 2., $\theta(e) = \theta(f) = \theta(f')$ does not hold, which implies $M, e \not\models \varphi$.

So there exists $f \in E$ such that $\gamma(f) = 1$ and $M, f \not\models \varphi$. Notice that $\theta(f) \in \{\square, \blacksquare\}$. Let f' be the unique event such that $M, f, f' \models \pi$. Such an event exists by Condition 2., and is unique since π is functional.

Suppose $f' <_{\mathsf{proc}} f$. Let $f_0 = f$, $f_1 = f'$, and for all $i \in \mathbb{N}$, let f_{i+1} be the unique event such that $M, f_i, f_{i+1} \models \pi$. Note that, for all i, $\theta(f_{i+1}) = \theta(f_i) \in \{\square, \blacksquare\}$. By Condition 1., there exists g_0 such that $f_0 >_{\mathsf{proc}} g_0 >_{\mathsf{proc}} f_1$ and $\theta(f_0) \neq \theta(g_0) \in \{\square, \blacksquare\}$. For an illustration, see Figure 11. Again, for all $i \in \mathbb{N}$, let g_{i+1} be the unique event such that $M, g_i, g_{i+1} \models \pi$. Note that all $f_0, f_1, \ldots$ have the same color, in $\{\square, \blacksquare\}$, and all $g_0, g_1, \ldots$ carry the complementary color. Thus, $f_i \neq g_j$ for all $i, j \in \mathbb{N}$. But, by Lemma 11, this implies $f_0 >_{\mathsf{proc}} g_0 >_{\mathsf{proc}} f_1 >_{\mathsf{proc}} g_1 >_{\mathsf{proc}} \cdots$, which contradicts the fact that the past of f_0 is finite.

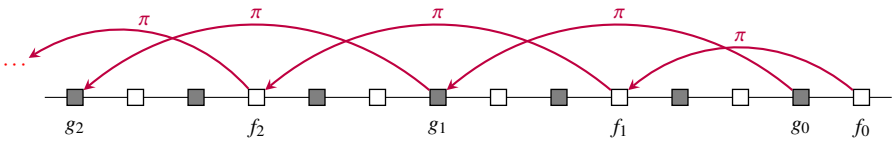


Figure 11: Proof of Claim 3.

Similarly, suppose $f <_{\mathsf{proc}} f'$. Let $f_0 = f'$, $f_1 = f$, and for all $i \in \mathbb{N}$, let f_{i+1} be *some* event such that $M, f_i, f_{i+1} \models \pi^{-1}$ and $\theta(f_{i+1}) = \theta(f_i) \in \{\square, \blacksquare\}$. Let us show that $f_0 >_{\mathsf{proc}} f_1 >_{\mathsf{proc}} f_2 >_{\mathsf{proc}} \cdots$, by contradiction. Assume that $f_i \leq_{\mathsf{proc}} f_{i+1}$ for some minimal $i \geq 1$. Since π is functional, we have $[\![\pi]\!](f_i) = \{f_{i-1}\}$, and $[\![\pi]\!](f_{i+1}) = \{f_i\}$. Then, by Lemma 11, $f_{i-1} \leq_{\mathsf{proc}} f_i$, which contradicts the minimality of i. □(Claim 3)

This concludes the proof of Lemma 12. □

The general case is more complicated. We first show how to rewrite an arbitrary loop formula using loops on paths of the form π or $\pi \cdot \xrightarrow{+}$ where π is functional. Intuitively, this means that loop formulas will only be used to perform the following test. Given an event e such that there exists a (unique) e' with $M, e, e' \models \pi$, and e' is on the same process as e, which one of the following is true: $e' <_{\mathsf{proc}} e$, $e' = e$, or $e <_{\mathsf{proc}} e'$? Indeed, we have $M, e \models \mathsf{Loop}(\pi \cdot \xrightarrow{+})$ iff $e' <_{\mathsf{proc}} e$.

Lemma 13. *For all* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *path formulas* π*,*

$$\mathsf{Loop}(\pi) \equiv \mathsf{Loop}(\min \pi) \vee \left(\langle \pi^{-1} \rangle \wedge \mathsf{Loop}((\min \pi) \cdot \xrightarrow{+}) \wedge \neg \mathsf{Loop}((\max \pi) \cdot \xrightarrow{+})\right).$$

Proof. The result essentially follows from Lemma 4, saying that $[\![\pi]\!](e)$ is exactly the set of events in the interval from $\min[\![\pi]\!](e)$ to $\max[\![\pi]\!](e)$ that satisfy $\langle \pi^{-1} \rangle$.

First, if we have $M, e \models \mathsf{Loop}(\pi)$ and $M, e \not\models \mathsf{Loop}(\min \pi)$, then $\min[\![\pi]\!](e) <_{\mathsf{proc}} e \leq_{\mathsf{proc}} \max[\![\pi]\!](e)$ and $M, e \models \langle \pi^{-1} \rangle$, hence $M, e \models \langle \pi^{-1} \rangle \wedge \mathsf{Loop}((\min \pi) \cdot \xrightarrow{+}) \wedge \neg \mathsf{Loop}((\max \pi) \cdot \xrightarrow{+})$.

Conversely, if we have $M, e \models \mathsf{Loop}(\min \pi)$, then $M, e \models \mathsf{Loop}(\pi)$, and if $M, e \models \langle \pi^{-1} \rangle \wedge \mathsf{Loop}((\min \pi) \cdot \xrightarrow{+}) \wedge \neg \mathsf{Loop}((\max \pi) \cdot \xrightarrow{+})$, then $M, e \models \langle \pi^{-1} \rangle$ and $\min[\![\pi]\!](e) <_{\mathsf{proc}} e \leq_{\mathsf{proc}} \max[\![\pi]\!](e)$. By Lemma 4, this implies $M, e, e \models \pi$. Hence, $M, e \models \mathsf{Loop}(\pi)$. □

Finally, we are ready to prove the general case, translating $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ event formulas to MSC transducers:

Proposition 3. *For every* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *event formula* φ*, there exists a transducer* $\mathcal{A}_\varphi$ *with* $2^{O(|\varphi|^2)}$ *states per process such that* $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.

Proof. We proceed by induction on the number of loop subformulas in φ. The base case is stated in Lemma 10. Let $\psi = \mathsf{Loop}(\pi')$ be a subformula of φ such that π' contains no loop subformulas and $\mathsf{Comp}(\pi') \subseteq \mathsf{id}$. We will show below that there exists a transducer $\mathcal{A}_\psi$ with $2^{O(|\psi|^2)}$ states per process such that $[\![\mathcal{A}_\psi]\!] = \{(M, M_\psi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$. Suppose that we have constructed $\mathcal{A}_\psi$. Consider the formula φ' over $\Sigma \times \{0, 1\}$ obtained from φ by replacing ψ by $\bigvee_{a \in \Sigma}(a, 1)$, and all event formulas a, with $a \in \Sigma$, by $(a, 0) \vee (a, 1)$. It contains fewer Loop operators than φ, so by induction hypothesis, we have a transducer $\mathcal{A}_{\varphi'}$ for φ'. We then let $\mathcal{A}_\varphi = \mathcal{A}_{\varphi'} \circ (\mathcal{A}_{Id} \times \mathcal{A}_\psi)$, where $\mathcal{A}_{Id}$ is the transducer for the identity relation.

Thus, all we need to prove is that we can construct such a transducer $\mathcal{A}_\psi$. We first apply Lemma 13. We construct transducers of size $2^{O(|\pi'|^2)}$ for the formulas $\mathsf{Loop}(\min \pi')$, $\langle \pi'^{-1} \rangle$, $\mathsf{Loop}((\min \pi') \cdot \xrightarrow{+})$ and $\mathsf{Loop}((\max \pi') \cdot \xrightarrow{+})$, and define $\mathcal{A}_\psi$ as the expected composition of these transducers. Recall that both $\min \pi'$ and $\max \pi'$ are functional, and of size $O(|\pi'|^2)$. Using Lemmas 12 and 10, we already have transducers for $\mathsf{Loop}(\min \pi')$ and $\langle \pi'^{-1} \rangle$.

So it suffices to show that for any functional path formula $\pi \in \mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$, there exists a transducer of size $2^{O(|\pi|)}$ for the formula $\psi = \mathsf{Loop}(\pi \cdot \xrightarrow{+})$. We assume that $\mathsf{Comp}(\pi) \subseteq \mathsf{id}$.

We start with some easy remarks. Let $p \in P$ be some process and $e \in E_p$. A necessary condition for $M, e \models \psi$ is that $M, e \models \langle \pi \rangle$, and since π is functional, that $M, e \not\models \mathsf{Loop}(\pi)$.

We let E_p^π be the set of events $e \in E_p$ satisfying $\langle\pi\rangle$. For all $e \in E_p^\pi$, we let $e' \in E_p$ be the unique event such that $M, e, e' \models \pi$. The transducer $\mathcal{A}_\psi$ will establish, for each $e \in E_p^\pi$, whether $e' <_{\mathsf{proc}} e$, $e' = e$, or $e <_{\mathsf{proc}} e'$, and it will output 1 if $e' <_{\mathsf{proc}} e$, and 0 otherwise. The case $e' = e$ means $M, e \models \mathsf{Loop}(\pi)$ and can be checked with the help of Lemma 12. So the difficulty is to distinguish between $e' <_{\mathsf{proc}} e$ and $e <_{\mathsf{proc}} e'$ when $M, e \models \langle\pi\rangle \wedge \neg\mathsf{Loop}(\pi)$.

Claim 4. *Let $\psi = \mathsf{Loop}(\pi \cdot \xrightarrow{+})$ and let f be the minimal event in E_p^π (assuming this set is nonempty). Then, $M, f \models \psi$ iff $M, f \models \mathsf{Loop}(\mathsf{min}\ (\xleftarrow{+} \cdot \pi^{-1}))$.*

Proof of Claim 4. The right to left implication holds without any hypothesis. Conversely, let $f' = [\![\pi]\!](f)$ and assume that $f' \xrightarrow{+} f$. Then, $M, f, f \models \xleftarrow{+} \cdot \pi^{-1}$, and $g = [\![\mathsf{min}\ (\xleftarrow{+} \cdot \pi^{-1})]\!](f)$ is well-defined and $g \leq_{\mathsf{proc}} f$. This is illustrated in Figure 12. Moreover, $M, g \models \langle\pi\rangle$ and by minimality of f in E_p^π, we conclude that $g = f$. □(Claim 4)

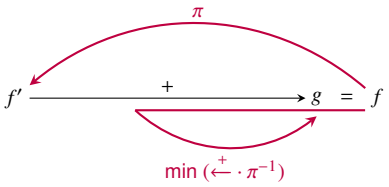


Figure 12: Proof of Claim 4.

Claim 5. *Let e, f be consecutive events in E_p^π, i.e., $e, f \in E_p^\pi$ and $M, e, f \models \xrightarrow{\neg\langle\pi\rangle}$.*

1. *If $M, e \models \psi$, then $[M, f \models \psi$ iff $M, f \not\models \mathsf{Loop}(\pi) \vee \mathsf{Loop}(\mathsf{min}\ (\xrightarrow{+} \cdot \pi^{-1}))]$.*
2. *If $M, e \not\models \psi$, then $[M, f \models \psi$ iff $M, f \models \mathsf{Loop}(\mathsf{max}\ (\pi \cdot \xrightarrow{\neg\langle\pi\rangle}))]$.*

Proof of Claim 5. We show the two statements.

1. Assume that $M, e \models \psi$. The left-to-right implication holds without any hypothesis. Conversely, assume that $M, f \not\models \psi$. If $M, f \models \mathsf{Loop}(\pi)$, we are done. Otherwise, let $e' = [\![\pi]\!](e)$ and $f' = [\![\pi]\!](f)$. We have $e' <_{\mathsf{proc}} e$ and $f <_{\mathsf{proc}} f'$. Moreover, $M, f \models \langle\xrightarrow{+} \cdot \pi^{-1}\rangle$, hence $g = [\![\mathsf{min}\ (\xrightarrow{+} \cdot \pi^{-1})]\!](f)$ is well-defined and $g \leq_{\mathsf{proc}} f$. Notice that $g \in E_p^\pi$, and $f <_{\mathsf{proc}} g' = [\![\pi]\!](g)$. If $g <_{\mathsf{proc}} f$ (see Figure 13), we get $g \leq_{\mathsf{proc}} e$, and using Lemma 11, we obtain $g' \leq_{\mathsf{proc}} e' <_{\mathsf{proc}} e <_{\mathsf{proc}} f$, a contradiction. Therefore, $g = f$ and $M, f \models \mathsf{Loop}(\mathsf{min}\ (\xrightarrow{+} \cdot \pi^{-1}))$.

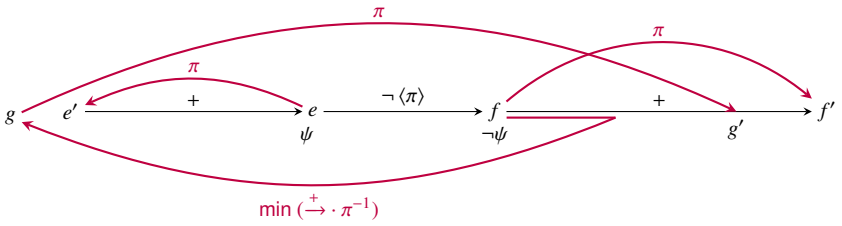


Figure 13: Proof of Claim 5(1.).

2. Assume that $M, e \not\models \psi$. The right-to-left implication holds easily since $[\![\mathsf{max}\ (\pi \cdot \xrightarrow{\neg\langle\pi\rangle})]\!] \subseteq [\![\pi \cdot \xrightarrow{+}]\!]$. Conversely, assume that $M, f \models \psi$. Let $e' = [\![\pi]\!](e)$ and $f' = [\![\pi]\!](f)$. We have $e \leq_{\mathsf{proc}} e'$ and $f' <_{\mathsf{proc}} f$ (see Figure 14). From Lemma 11 we get $e' \leq_{\mathsf{proc}} f'$ and since e, f are consecutive in E_p^π, we obtain $M, f', f \models \xrightarrow{\neg\langle\pi\rangle}$. Therefore, $M, f \models \mathsf{Loop}(\pi \cdot \xrightarrow{\neg\langle\pi\rangle}) \equiv \mathsf{Loop}(\mathsf{max}\ (\pi \cdot \xrightarrow{\neg\langle\pi\rangle}))$.

This concludes the proof of the claim. □(Claim 5).

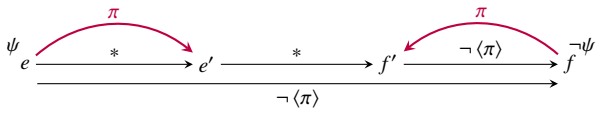


Figure 14: Proof of Claim 5(2.).

To conclude the proof of Proposition 3, let us consider the five formulas $\varphi_1 = \langle\pi\rangle$, $\varphi_2 = \mathsf{Loop}(\pi)$, $\varphi_3 = \mathsf{Loop}(\mathsf{min}\,(\overset{+}{\leftarrow} \cdot \pi^{-1}))$, $\varphi_4 = \mathsf{Loop}(\mathsf{min}\,(\overset{+}{\rightarrow} \cdot \pi^{-1}))$, and $\varphi_5 = \mathsf{Loop}(\mathsf{max}\,(\pi \cdot \overset{\neg\langle\pi\rangle}{\longrightarrow}))$. We can easily see that $\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_4$ is a necessary condition for $\psi = \mathsf{Loop}(\pi \cdot \overset{+}{\rightarrow})$. Also, both φ_3 and φ_5 are sufficient conditions for ψ. The only case which is not covered is when $M, f \models \varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3 \wedge \neg\varphi_4 \wedge \neg\varphi_5$. In this case, from Claims 4 and 5, we see that $M, f \models \psi$ iff f is not minimal in E_p^π and $M, e \models \psi$, where e is the predecessor of f in E_p^π.

By Lemmas 10 and 12, we already have transducers $\mathcal{A}_{\varphi_i}$ for $i \in \{1, 2, 3, 4, 5\}$. We let $\mathcal{A}_\psi = \mathcal{A} \circ (\mathcal{A}_{\varphi_1} \times \mathcal{A}_{\varphi_2} \times \mathcal{A}_{\varphi_3} \times \mathcal{A}_{\varphi_4} \times \mathcal{A}_{\varphi_5})$, where, at an event f labeled $(b_1, b_2, b_3, b_4, b_5)$, the transducer $\mathcal{A}$ outputs 1 if $b_3 = 1$ or $b_5 = 1$ or if $(b_1, b_2, b_3, b_4, b_5) = (1, 0, 0, 0, 0)$ and the output was 1 at the last event e on the same process satisfying φ_1 (to do so, each process keeps in its state the output at the last event where b_1 was 1), and 0 otherwise. □

4.3. Translation of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *and* FO *Sentences into CFMs.*

Wrapping-up, we obtain our main results as corollaries. First the translation of $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ sentences to CFMs:

Theorem 2. *For every* $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ *sentence* ξ*, there exists a CFM* $\mathcal{A}_\xi$ *with* $2^{O(|\xi|^2)}$ *states per process such that* $\mathbb{L}(\mathcal{A}_\xi) = \mathbb{L}(\xi)$.

Proof. Given an event formula φ, we can construct $\mathcal{A}_\varphi$ according to Proposition 3. From $\mathcal{A}_\varphi$, it is easy to build CFMs for the sentences $\mathsf{E}\,\varphi$ and $\neg\,\mathsf{E}\,\varphi$. Closure of $\mathcal{L}$(CFM) under union and intersection takes care of disjunction and conjunction. □

By Theorem 1, every FO$[\rightarrow, \lhd, \leq]$ sentence Φ is equivalent to some $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ sentence ξ, for which there is an equivalent CFM $\mathcal{A}_\xi$ by Theorem 2. Therefore, we obtain:

Theorem 3. $\mathcal{L}(\mathrm{FO}[\rightarrow, \lhd, \leq]) \subseteq \mathcal{L}(\mathrm{CFM})$.

The translation is effective, but inherently non-elementary, already when $|P| = 1$ [Sto74].

It is standard to prove $\mathcal{L}(\mathrm{CFM}) \subseteq \mathcal{L}(\mathrm{EMSO}[\rightarrow, \lhd])$: The formula guesses an assignment of transitions to events in terms of existentially quantified second-order variables (one for each transition) and then checks, in its first-order kernel, that the assignment is indeed an (accepting) run. As, moreover, the class $\mathcal{L}$(CFM) is closed under projection, we obtain the following logical characterization of CFMs as a corollary:

Theorem 4. $\mathcal{L}(\mathrm{EMSO}[\rightarrow, \lhd, \leq]) = \mathcal{L}(\mathrm{CFM})$.

5. Applications

5.1. Existentially bounded MSCs

Though the translation of EMSO/FO formulas into CFMs is interesting on its own, it allows us to obtain some difficult results for bounded CFMs as corollaries. In fact, we even extend known results to infinite MSCs.

Bounded MSCs. The first logical characterizations of communicating finite-state machines were obtained for classes of *bounded* MSCs. Intuitively, this corresponds to restricting the channel capacity. Bounded MSCs are defined in terms of linearizations. A *linearization* of a given MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$ is a total order $\preceq \subseteq E \times E$ extending $\leq$ and of order type at most ω, i.e., $\leq \subseteq \preceq$ and $\{e \mid e \preceq f\}$ is finite for all $f \in E$. For $B \in \mathbb{N}$, we call $\preceq$ B-bounded if, for all $g \in E$ and $(p, q) \in Ch$, $|\{(e, f) \in \lhd \cap (E_p \times E_q) \mid e \preceq g \prec f\}| \leq B$. In other words, the number of pending messages in (p, q) never exceeds B if we follow the linearization defined by $\preceq$. There are (at least) two natural definitions of bounded MSCs: We call M $\exists B$-bounded if M has *some* B-bounded linearization. Accordingly,

it is $\forall B$-bounded if *all* its linearizations are B-bounded. The set of $\exists B$-bounded MSCs is denoted by $\mathbb{MSC}_{\exists B}(P,\Sigma)$, the set of $\forall B$-bounded MSCs by $\mathbb{MSC}_{\forall B}(P,\Sigma)$. Moreover, we let $\mathbb{MSC}^{\text{fin}}_{\exists B}(P,\Sigma) := \mathbb{MSC}_{\exists B}(P,\Sigma) \cap \mathbb{MSC}^{\text{fin}}(P,\Sigma)$ and $\mathbb{MSC}^{\text{fin}}_{\forall B}(P,\Sigma) := \mathbb{MSC}_{\forall B}(P,\Sigma) \cap \mathbb{MSC}^{\text{fin}}(P,\Sigma)$.

Example 7. The MSC from Figure 1 is $\exists 1$-bounded, but it is not $\forall B$-bounded, no matter what B is.

In this subsection, we will consider only $\exists B$-bounded MSCs. We show the following results. First, for a given channel bound B, the set $\mathbb{MSC}_{\exists B}(P,\Sigma)$ is FO$[\rightarrow, \lhd, \leq]$-definable (essentially due to [LM04]). By Theorem 4, we obtain [GKM06, Proposition 5.14] stating that this set is recognized by some CFM. Second, we obtain [GKM06, Proposition 5.3], a Büchi-Elgot-Trakhtenbrot theorem for existentially bounded MSCs, as a corollary of Theorem 4 in combination with a linearization normal form from [TW02].

Known results. Let $M = (E, \rightarrow, \lhd, loc, \lambda)$ be some finite or infinite MSC. Given $e \in E$, we write $\mathsf{type}(e) = p$ if e is an internal event on process p, $\mathsf{type}(e) = p!q$ if e is a write on channel (p,q), and $\mathsf{type}(e) = q?p$ if e is a read from channel (p,q). We associate with the linearization $\leq$ a word $M_{\leq}$ over the alphabet $\Sigma_{lin} = \Sigma \times (P \cup \{p!q, q?p \mid (p,q) \in Ch\})$. Namely, if the linearization is $e_1 \prec e_2 \prec e_3 \prec \cdots$, we let $M_{\leq} = a_1 a_2 a_3 \cdots$ where $a_i = (\lambda \times \mathsf{type})(e_i)$. Note that M can be retrieved from $M_{\leq}$. We let $Lin^B(M) = \{M_{\leq} \mid \leq \text{ is a } B\text{-bounded linearization of } M\} \subseteq \Sigma^*_{lin} \cup \Sigma^{\omega}_{lin}$.

Fact 8 [GKM06, Theorem 4.1]**).** *Let $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\text{fin}}_{\exists B}(P,\Sigma)$. The following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.*
2. *$L = \mathbb{L}(\Phi)$ for some MSO formula Φ.*
3. *$Lin^B(L)$ is a regular language (of finite words).*

The proof given in [GKM06] relies on the theory of Mazurkiewicz traces. Another major part of the proof is the construction of a CFM recognizing the set $\mathbb{MSC}^{\text{fin}}_{\exists B}(P,\Sigma)$ of finite $\exists B$-bounded MSCs [GKM06, Proposition 5.14]. We show below that this CFM, or more generally, a CFM for the set $\mathbb{MSC}_{\exists B}(P,\Sigma)$ of finite or infinite $\exists B$-bounded MSCs, can in fact be obtained as a simple application of Theorem 4. Moreover, we give an alternative proof of (3) $\Longrightarrow$ (1) (Section 5 in [GKM06]), and again extend the result to infinite MSCs.

As mentioned before, the implication (1) $\Longrightarrow$ (2) follows from a standard translation of CFMs into EMSO. Finally, (2) $\Longrightarrow$ (3) is also easy to prove: the channel bound can be used to translate the MSO sentence Φ into an MSO sentence over Σ_{lin}-labeled words, defining $Lin^B(L)$.

A CFM for Existentially Bounded MSCs. The set $\mathbb{MSC}_{\exists B}(P,\Sigma)$ of $\exists B$-bounded MSCs is in fact FO$[\lhd, \rightarrow, \leq]$-definable, and thus, we can apply Theorem 4 to construct a CFM $\mathcal{A}_{\exists B}$ recognizing $\mathbb{MSC}_{\exists B}(P,\Sigma)$. We describe below a formula defining $\mathbb{MSC}_{\exists B}(P,\Sigma)$.

Let us first recall a characterization of $\exists B$-bounded MSCs. Let $M = (E, \rightarrow, \lhd, loc, \lambda)$ be an MSC. We define a relation $rev_B \subseteq E \times E$ which consists of the set of pairs (f,g) such that f is a receive event from some channel (p,q) with corresponding send event $e \lhd f$, and g is the B-th send on channel (p,q) after event e. The relation rev_B is illustrated in Figure 15 (represented by the dashed edges) for $B = 1$ and an $\exists 1$-bounded MSC. It can be defined by the PDL$_{\mathsf{sf}}[\cup]$ path formula

$$\mathsf{rev}_B = \bigcup_{p \neq q} \lhd^{-1}_{p,q} \cdot \Big(\xrightarrow{\neg\langle \lhd_{p,q} \rangle} \cdot \{\langle \lhd_{p,q} \rangle\}? \Big)^B .$$

For completeness, let us also give a corresponding FO$[\rightarrow, \lhd, \leq]$ formula:

$$rev_B(x,y) := \exists z_0, z_1, \ldots, z_B.\ z_0 \lhd x \wedge z_B = y \wedge \bigwedge_{1 \leq i \leq B} \exists x_i.\ z_i \lhd x_i \wedge x \leq_{\mathsf{proc}} x_i$$
$$\wedge \bigwedge_{0 \leq i < B} z_i <_{\mathsf{proc}} z_{i+1} \wedge \neg(\exists z', x'.\ z_i <_{\mathsf{proc}} z' <_{\mathsf{proc}} z_{i+1} \wedge z' \lhd x' \wedge x \leq_{\mathsf{proc}} x') .$$

Fact 9 [LM04]**).** *M is $\exists B$-bounded if and only if the relation $(< \cup rev_B)$ is acyclic.*

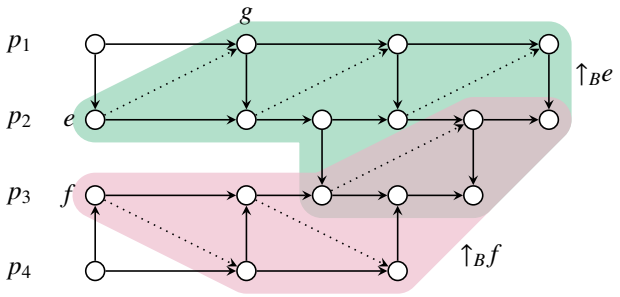


Figure 15: The relation rev_B for $B = 1$, and the sets $\uparrow_B e$ and $\uparrow_B f$.

In fact, a linearization $\preceq$ of M is B-bounded iff it contains rev_B. Indeed, assume that $rev_B \subseteq \preceq$ and that $\preceq$ is not B-bounded. Then, we find $e_0 \prec e_1 \prec \cdots \prec e_B \preceq g \prec f_0 \prec f_1 \prec \cdots \prec f_B$ with $(e_i, f_i) \in \lhd_{p,q} \cap E_p \times E_q$. Without loss of generality, we can assume that there are no other writes on channel (p, q) between e_i and e_{i+1}. This implies $(f_0, e_B) \in rev_B$, a contradiction. Conversely, if $rev_B \not\subseteq \preceq$ then we find $(f_0, e_B) \in rev_B$ with $e_B \prec f_0$. We deduce that $e_0 \prec e_1 \prec \cdots \prec e_B \prec f_0 \prec f_1 \prec \cdots \prec f_B$ with $(e_i, f_i) \in \lhd \cap E_p \times E_q$ and the linearization is not B-bounded.

Note that, if $(< \cup rev_B)$ contains a cycle, then it contains one of size at most $2|P|$. More precisely, M is $\exists B$-bounded if and only if it satisfies the $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}, \cup]$ formula $\xi_{\exists B} = \neg\, \mathsf{E}\, \mathsf{Loop}(\mathsf{lt}_B)$, where

$$\mathsf{lt}_B = \bigcup_{2 \le n \le |P|} ((\lhd \cup \mathsf{rev}_B) \cdot \xrightarrow{+})^n \qquad \lhd = \bigcup_{p \neq q} \lhd_{p,q}\,.$$

Again, let us determine an equivalent $\mathrm{FO}[\rightarrow, \lhd, \le]$ formula:

$$\Phi_{\exists B} = \bigwedge_{2 \le n \le 2|P|} \neg\Big(\exists x_0, \ldots, x_n.\ x_0 = x_n \wedge \bigwedge_{0 \le i < n} x_i < x_{i+1} \vee rev_B(x_i, x_{i+1})\Big)\,.$$

Applying Theorem 4 we obtain

Corollary 2. *Given $B > 0$, we can construct a CFM $\mathcal{A}_{\exists B}$ recognizing the set $\mathbb{MSC}_{\exists B}(P, \Sigma)$ of $\exists B$-bounded finite or infinite MSCs.*

FO-definable Linearizations for Existentially Bounded MSCs. We will make use of canonical linearizations of certain MSCs, which we adapt from [TW02, Definition 13] where the definition was given for traces. It is based on the following lemma. Though it is stated for a special case in [TW02], the proof can be taken almost verbatim. We only provide the proof for completeness.

Lemma 14 [TW02, Lemma 14]**.** *Let $(E, \le)$ be a partially ordered set, and $\sqsubset \subseteq E \times E$ a strict well-founded total order. For $e, f \in E$, we write $e \parallel f$ when $e \not\le f$ and $f \not\le e$, and we let $\uparrow e = \{f \in E \mid e \le f\}$. Then the relation $\prec \subseteq E \times E$ defined by*

$$e \prec f \iff \left(\begin{array}{ll} & e < f \\ \vee & e \parallel f \ \wedge\ \min_{\sqsubset}(\uparrow e \setminus \uparrow f) \sqsubset \min_{\sqsubset}(\uparrow f \setminus \uparrow e) \end{array} \right)$$

is a strict linear order extending $<$.

Proof. Notice that, for all $e \neq f$, we have either $e \prec f$ or $f \prec e$, but not both. Indeed, for all $e \neq f$, we have $\uparrow e \setminus \uparrow f \cap \uparrow f \setminus \uparrow e = \emptyset$, and if $e \parallel f$, then the two sets are nonempty as $e \in \uparrow e \setminus \uparrow f$ and $f \in \uparrow f \setminus \uparrow e$.

It remains to show that $\prec$ is transitive. Let $e_1, e_2, e_3 \in E$ such that $e_1 \prec e_2 \prec e_3$. Note that e_1, e_2, e_3 are pairwise distinct. For distinct $i, j \in \{1, 2, 3\}$, if $\uparrow e_i \setminus \uparrow e_j \neq \emptyset$, we let $e_{ij} = \min_{\sqsubset} \uparrow e_i \setminus \uparrow e_j$.

To prove $e_1 \prec e_3$, we distinguish several cases.

Case $e_1 < e_2 < e_3$**:** As $<$ is transitive, we get $e_1 < e_3$.

Case $e_1 < e_2 \parallel e_3$**:** This implies $e_3 \not\leq e_1$. If $e_1 < e_3$, we are done. So suppose $e_1 \parallel e_3$. Since $e_2 \prec e_3$, we have $e_{23} \sqsubset e_{32}$. From $\uparrow e_2 \subseteq \uparrow e_1$, we deduce $\uparrow e_2 \setminus \uparrow e_3 \subseteq \uparrow e_1 \setminus \uparrow e_3$. Thus, $e_{13} \sqsubseteq e_{23}$. Similarly, $e_{32} \sqsubseteq e_{31}$. We obtain $e_{13} \sqsubseteq e_{23} \sqsubset e_{32} \sqsubseteq e_{31}$.

Case $e_1 \parallel e_2 < e_3$**:** This case is very similar to the previous one.

Case $e_1 \parallel e_2 \parallel e_3$**:** Since $e_1 \prec e_2 \prec e_3$, we have $e_{12} \sqsubset e_{21}$ and $e_{23} \sqsubset e_{32}$. Suppose $e_1 \not\leq e_3$ (otherwise, we are done). We have $e_3 \not\leq e_1$, since $e_3 < e_1$ implies $e_{32} \sqsubseteq e_{12} \sqsubset e_{21} \sqsubseteq e_{23}$, a contradiction.

So we can assume $e_1 \parallel e_3$. It remains to show $e_{13} \sqsubset e_{31}$. First, one shows that

$$e_{13} \sqsubseteq e_{12}\,. \tag{2}$$

If $e_{12} \notin \uparrow e_3$, then (2) is immediate. So suppose $e_{12} \in \uparrow e_3$, i.e., $e_{12} \in \uparrow e_3 \setminus \uparrow e_2$. Then,

$$e_{23} \sqsubset e_{32} \sqsubseteq e_{12}\,. \tag{3}$$

Let us consider two cases. If $e_{23} \notin \uparrow e_1$ then $e_{21} \sqsubseteq e_{23}$. By (3), we obtain $e_{21} \sqsubset e_{12}$, which contradicts $e_1 \prec e_2$. Hence $e_{23} \in \uparrow e_1$ and we get $e_{13} \sqsubseteq e_{23}$. We deduce that (2) holds.

To conclude the proof, we distinguish once more two cases:

Case $e_{31} \in \uparrow e_2$**:** Then, $e_{12} \sqsubset e_{21} \sqsubseteq e_{31}$. Applying (2), we obtain $e_{13} \sqsubset e_{31}$.

Case $e_{31} \notin \uparrow e_2$**:** Then, $e_{23} \sqsubset e_{32} \sqsubseteq e_{31}$. If $e_{23} \in \uparrow e_1$, then $e_{13} \sqsubseteq e_{23} \sqsubset e_{31}$ and we are done. If $e_{23} \notin \uparrow e_1$, then $e_{21} \sqsubseteq e_{23}$, which implies $e_{12} \sqsubset e_{31}$. By (2), we obtain $e_{13} \sqsubset e_{31}$.

This concludes the proof of Lemma 14. □

We now define a canonical linear order on the events of an $\exists B$-bounded MSC $M = (E, \rightarrow, \lhd, loc, \lambda)$. We fix some strict total order $\sqsubset$ on P, and extend it to E as follows: $e \sqsubset f$ if $e <_{\mathsf{proc}} f$ or $loc(e) \sqsubset loc(f)$. Clearly, $\sqsubset$ is well-founded and a strict linear order on E. We apply Lemma 14 with $\sqsubset$ and $\leq_B = (\leq \cup rev_B)^*$ which is a partial order when the MSC M is $\exists B$-bounded. We obtain a linear order $\preceq_B$ of M extending both $<$ and rev_B.

Example 8. Consider the MSC M in Figure 15 and suppose $p_1 \sqsubset p_2 \sqsubset p_3 \sqsubset p_4$. We have $\min_{\sqsubset}(\uparrow e \setminus \uparrow f) = g$ and $\min_{\sqsubset}(\uparrow f \setminus \uparrow e) = f$. Since $g \sqsubset f$, we obtain $e \prec_B f$.

In general, $\preceq_B$ need not be of order type at most ω, i.e., it is not necessarily a linearization. For instance, with two processes $p \sqsubset q$, no communication events, and infinitely many local events on both p and q, the order type of $\preceq_B$ is $\omega + \omega$. When the order type of $\preceq_B$ is at most ω, it is a B-bounded linearization of M.

Finally, the relation $\prec_B$ is FO$[\rightarrow, \lhd, \leq]$-definable. Indeed, the strict partial order $\prec_B$ is FO$[\rightarrow, \lhd, \leq]$-definable since it can be expressed with the path formula lt_B given above. From its definition, we deduce that the relation $\prec_B$ is also FO$[\rightarrow, \lhd, \leq]$-definable.

We are now ready to give our alternative proof of the direction (3) $\Longrightarrow$ (1) in Fact 8.

Proof of Fact 8 (3) $\Longrightarrow$ (1). Let L be a set of $\exists B$-bounded *finite* MSCs such that $Lin^B(L)$ is regular. There exists an EMSO sentence Φ_{lin} over Σ_{lin}-labeled words such that $Lin^B(L) = \mathbb{L}(\Phi_{lin})$. Since $\preceq_B$ is FO$[\rightarrow, \lhd, \leq]$-definable, it is easy to translate Φ_{lin} into an EMSO$[\rightarrow, \lhd, \leq]$ formula Φ such that $\mathbb{L}(\Phi) \subseteq \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(P, \Sigma)$ and, for all $M \in \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(P, \Sigma)$, we have $M \models \Phi$ if and only if $M_{\preceq_B} \models \Phi_{lin}$. Let $\mathcal{A}$ be a CFM such that $\mathbb{L}(\mathcal{A}) = \mathbb{L}(\Phi \wedge \Phi_{\exists B})$. Then, for all $M \in L$, M is $\exists B$-bounded and $M_{\preceq_B} \models \Phi_{lin}$, hence $M \models \Phi \wedge \Phi_{\exists B}$, i.e., $M \in \mathbb{L}(\mathcal{A})$. Conversely, if $M \in \mathbb{L}(\mathcal{A})$, then $M \in \mathbb{MSC}^{\mathrm{fin}}_{\exists B}(P, \Sigma)$ and $\preceq_B$ is a linearization of M. Moreover, $M_{\preceq_B} \in Lin^B(L)$, hence $M \in L$. □

The extension of Fact 8 to infinite MSCs requires some extra work since the order type of $\preceq_B$ need not be at most ω. We denote by $\mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$ the set of infinite $\exists B$-bounded MSCs.

Theorem 5. *Let $B \in \mathbb{N}$ and $L \subseteq \mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$. The following are equivalent:*

1. *$L = \mathbb{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.*
2. *$L = \mathbb{L}(\Phi)$ for some MSO formula Φ.*
3. *$Lin^B(L)$ is an ω-regular language.*

The rest of Section 5.1 is devoted to the proof of (3) $\Longrightarrow$ (1), while (1) $\Longrightarrow$ (2) and (2) $\Longrightarrow$ (3) are again standard. Recall that we cannot exactly proceed as in the case of finite MSCs, as the canonical linear order associated with an infinite existentially bounded MSC is not necessarily of order type ω. We therefore adopt decomposition techniques and results from [TW02] and [Kus03], where Mazurkiewicz traces and universally bounded MSCs are considered, respectively. We first define a decomposition of an existentially bounded MSC into a finite part and boundedly many disconnected infinite parts M^j such that the canonical linear order of each M^j is in fact of order type ω, i.e., a canonical linearization (Lemma 16). Similarly, the given ω-regular word language $Lin^B(L)$ can be described as the composition of regular (and therefore EMSO-definable) finite and shuffled infinite parts such that the infinite parts correspond to linearizations of the MSCs M^j (Lemmas 18–20). The corresponding EMSO formulas over words can then be transformed into EMSO formulas over MSCs using FO-definability of canonical linearizations (Lemma 21). In this step, it is crucial that the separate infinite parts M^j actually have canonical linearizations. This guarantees that word formulas for linearizations are faithfully simulated by the associated formulas over MSCs. Using our main result, Theorem 4, we can now conclude that there is a CFM for the target language L.

Let us be precise. We start by defining a decomposition of infinite MSCs such that in each component of the decomposition, $\preceq_B$ is of order type at most ω.

Let $M = (E, \rightarrow, \lhd, loc, \lambda) \in \mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$. We denote by $\mathsf{Types}^{\mathrm{fin}}(M)$ (resp. $\mathsf{Types}^{\mathrm{inf}}(M)$) the set of types that occur finitely many times (resp. infinitely often) in M:

$$\mathsf{Types}^{\mathrm{fin}}(M) = \{\mathsf{type}(e) \mid e \in E \wedge \{f \in E \mid \mathsf{type}(e) = \mathsf{type}(f)\} \text{ is finite}\}$$
$$\mathsf{Types}^{\mathrm{inf}}(M) = \{\mathsf{type}(e) \mid e \in E \wedge \{f \in E \mid \mathsf{type}(e) = \mathsf{type}(f)\} \text{ is infinite}\}\,.$$

We then let

$$E^{\mathrm{fin}} = \{e \in E \mid \exists f \in E : e \preceq_B f \wedge \mathsf{type}(f) \in \mathsf{Types}^{\mathrm{fin}}(M)\} \qquad \text{and} \qquad E^{\mathrm{inf}} = E \setminus E^{\mathrm{fin}}\,.$$

Recall that $\preceq_B = (\leq \cup\, rev_B)^*$. The definition of E^{fin} and E^{inf} depends not only on M, but also on the given bound B. Note that E^{fin} and E^{inf} are not necessarily $\lhd$-closed: there may be some send events in E^{fin} whose matching receive events are in E^{inf} (the converse is not possible, since E^{fin} is downward closed). However, since M is B-bounded, there are at most B unmatched sends in E^{fin} for every channel (p, q).

Lemma 15. *There exists a B-bounded linearization $\preceq$ of M such that for all $e \in E^{\mathrm{fin}}$ and $f \in E^{\mathrm{inf}}$, $e \preceq f$.*

Proof. Recall that a linearization of M is B-bounded if and only if it contains $\preceq_B$.

Let $\preceq$ be any B-bounded linearization of M, and $\preceq'$ the concatenation of its restrictions to E^{fin} and E^{inf}. That is, $e \preceq' f$ if and only if $e \in E^{\mathrm{fin}}$ and $f \in E^{\mathrm{inf}}$, or $e \preceq f$ and $e, f \in E^{\mathrm{fin}}$, or $e \preceq f$ and $e, f \in E^{\mathrm{inf}}$. Clearly, $\preceq'$ is a total order on E. Let us show that for all $e \preceq_B f$, we have $e \preceq' f$. Since $\preceq$ is a B-bounded linearization of M, we have $e \preceq f$, and thus, if $e, f \in E^{\mathrm{fin}}$ or $e, f \in E^{\mathrm{inf}}$, we get $e \preceq' f$. If $e \in E^{\mathrm{fin}}$ and $f \in E^{\mathrm{inf}}$, then $e \preceq' f$ by definition. Finally, we cannot have $e \in E^{\mathrm{inf}}$ and $f \in E^{\mathrm{fin}}$ since E^{fin} is downward-closed with respect to $\preceq_B$. □

We further decompose E^{inf} into its connected components, as follows. We denote by $P_1, \ldots, P_m$ the maximal connected components of the undirected communication graph at infinity $(P, \{\{p, q\} \mid \mathsf{Types}^{\mathrm{inf}}(M) \cap \{p!q, q!p\} \neq \emptyset\})$. For all $1 \leq j \leq m$, we then let

$$E^j = E^{\mathrm{inf}} \cap \bigcup_{p \in P_j} E_p\,.$$

Note that, by definition, every E^j is infinite.

Finally, we associate with E^{fin}, E^{inf}, and each E^j the following MSCs over Σ_{lin}:

$$M^{\text{fin}} = (E^{\text{fin}}, \rightarrow \cap\, E^{\text{fin}} \times E^{\text{fin}}, \lhd \cap\, E^{\text{fin}} \times E^{\text{fin}}, loc|_{E^{\text{fin}}}, (\lambda \times \mathsf{type})|_{E^{\text{fin}}})$$
$$M^{\text{inf}} = (E^{\text{inf}}, \rightarrow \cap\, E^{\text{inf}} \times E^{\text{inf}}, \lhd \cap\, E^{\text{inf}} \times E^{\text{inf}}, loc|_{E^{\text{inf}}}, (\lambda \times \mathsf{type})|_{E^{\text{inf}}})$$
$$M^j = (E^j, \rightarrow \cap\, E^j \times E^j, \lhd \cap\, E^j \times E^j, loc|_{E^j}, (\lambda \times \mathsf{type})|_{E^j})\,.$$

Note that send events of M which are located in E^{fin} and whose matching receive event is in E^{inf} become internal events in M^{fin}, and similarly for unmatched receive events in the infinite part. Adding the types of all events to the labeling allows us to maintain any information on M in its decomposition.

Lemma 16. *For all M^j with $1 \leq j \leq m$, $\leq_B$ is of order type ω.*

Proof. This is in fact true of any linear extension of $\leq_B$, and not just the canonical $\leq_B$. We want to prove that for every $e \in E^j$, the set $\{f \in E^j \mid f \leq_B e\}$ is finite. To do so, it suffices to prove that for all $p \in P_j$, there exists $f \in E^j_p$ such that $e \leq_B f$ (and thus $e \leq_B f$). By definition of P_j, there exists a path $p_1, \ldots, p_\ell$ such that $p_1 = loc(e)$, $p_\ell = p$, and for all $1 \leq i < \ell$, M^j contains infinitely many events of type $p_i!p_{i+1}$ or $p_i?p_{i+1}$. If M^j contains infinitely many events of type $p_i!p_{i+1}$, then for all e' on process p_i, there exist f' on process p_i and g' on process p_{i+1} such that $e' \leq_{\mathsf{proc}} f'$ and $f' \lhd g'$, hence $e' \leq_B g'$. If M^j contains infinitely many events of type $p_i?p_{i+1}$, then for all e' on process p_i, there exist f' on process p_i and g' on process p_{i+1} such that $e' \leq_{\mathsf{proc}} f'$ and f' rev_B g', hence $e' \leq_B g'$. Therefore, we obtain events $e_2, \ldots, e_\ell$ on processes $p_2, \ldots, p_\ell = p$ such that $e \leq_B e_2 \leq_B \cdots \leq_B e_\ell$. □

To avoid any ambiguity, in this proof, we write e.g. $\text{FO}[\Sigma, P, \leq, \lhd]$ or $\text{FO}[\Sigma_{lin}, P, \leq, \lhd]$, rather than $\text{FO}[\leq, \lhd]$, so as to distinguish between formulas over MSCs over Σ and P, and MSCs over Σ_{lin} and P coming from decompositions. We also denote by $\text{FO}[\Sigma_{lin}, \leq]$, $\text{EMSO}[\Sigma_{lin}, \leq]$, and $\text{MSO}[\Sigma_{lin}, \leq]$ formulas over word linearizations, Σ_{lin} being the alphabet, and $\leq$ the underlying total order.

In the rest of the proof, we restrict to MSCs M having a fixed set $T = \mathsf{Types}^{\text{inf}}(M)$ of types at infinity, with maximal connected components of the induced communication graph at infinity being $P_1, \ldots, P_m$. We decompose Σ_{lin} into disjoint subalphabets $\Sigma^{\text{fin}}_{lin}, \Sigma^1_{lin}, \ldots, \Sigma^m_{lin}$ as follows. First, we denote by $\Sigma^{\text{inf}}_{lin}$ the set of all $(a, t) \in \Sigma_{lin}$ such that $t \in T$ and $\Sigma^{\text{fin}}_{lin} = \Sigma_{lin} \setminus \Sigma^{\text{inf}}_{lin}$. We further decompose $\Sigma^{\text{inf}}_{lin}$ into $\Sigma^1_{lin}, \ldots, \Sigma^m_{lin}$, where Σ^j_{lin} denotes the set of pairs of the form (a, p), $(a, p!q)$ or $(a, p?q)$ with $p \in P_j$ (by definition of the decomposition, this is also equivalent to $q \in P_j$). Note that, while every event in M^j is labeled with a letter in Σ^j_{lin}, events in M^{fin} may have labels in any of the alphabets $\Sigma^{\text{fin}}_{lin}, \Sigma^1_{lin}, \ldots, \Sigma^m_{lin}$. However, the labels of all $\leq_B$-maximal events of M^j are in $\Sigma^{\text{fin}}_{lin}$.

For words $u, v \in \Sigma^*_{lin} \cup \Sigma^\omega_{lin}$, we denote by $u \sqcup\!\sqcup v$ the *shuffle* of u and v, i.e., the set of words $w \in \Sigma^*_{lin} \cup \Sigma^\omega_{lin}$ that are a possible interleaving of u and v.

Now, let $L \subseteq \mathbb{MSC}^\omega_{\exists B}(P, \Sigma)$ be a set of MSCs with types at infinity T. We decompose words in $Lin^B(L)$ according to the decomposition of their corresponding MSCs. More precisely, for $w \in Lin^B(L)$, we denote by $w^{\text{fin}}, w^{\text{inf}}, w^1, \ldots, w^m$ the restrictions of w to positions denoting events located respectively in the parts $M^{\text{fin}}, M^{\text{inf}}, M^1, \ldots, M^m$ of the corresponding MSC $M \in L$. That is, $w^{\text{fin}} \in \Sigma^*_{lin}\Sigma^{\text{fin}}_{lin} \cup \{\varepsilon\}$ denotes a linearization of M^{fin}, $w^{\text{inf}} \in \left(\Sigma^{\text{inf}}_{lin}\right)^\omega$ denotes a linearization of M^{inf}, and $w^j \in \left(\Sigma^j_{lin}\right)^\omega$ denotes a linearization of M^j. Moreover, w is a shuffle of w^{fin} and w^{inf}, and w^{inf} is itself a shuffle of $w^1, \ldots, w^m$.

Lemma 17. *For all $w \in Lin^B(L)$, we have $w^{\text{fin}}w^{\text{inf}} \in Lin^B(L)$ and $w^{\text{fin}} \cdot \bigsqcup_{j=1}^m w^j \subseteq Lin^B(L)$.*

Proof. Let $M \in L$ be the MSC corresponding to w. As in Lemma 15, the word $w^{\text{fin}}w^{\text{inf}}$ is also a B-bounded linearization of M, and thus in $Lin^B(L)$. In addition, any $v \in \bigsqcup_{j=1}^m w^j$ corresponds to a possible B-bounded linearization of M^{inf}. Indeed, since there are no causal dependencies between events in distinct M^j, any interleaving of the B-bounded linearizations $w^1, \ldots, w^m$ of the different components yields a B-bounded linearization of M^{inf}. This means that we also have $w^{\text{fin}}v \in Lin^B(L)$. □

Lemma 18. *If $Lin^B(L)$ is an ω-regular language then there is a finite sequence of pairs of regular languages $(K_1, L_1), \ldots, (K_k, L_k)$, with $K_i \subseteq \Sigma^*_{lin}$ and $L_i \subseteq \Sigma^\omega_{lin}$, such that*

$$\left\{(w^{\text{fin}}, w^{\text{inf}}) \mid w \in Lin^B(L)\right\} = \bigcup_{1 \leq i \leq k} K_i \times L_i\,.$$

Moreover, for all $1 \leq i \leq k$ *and* $(v_1, \ldots, v_m) \in \left(\Sigma_{lin}^1\right)^\omega \times \cdots \times \left(\Sigma_{lin}^m\right)^\omega$*, if* $L_i \cap \coprod_{j=1}^m v_j \neq \emptyset$ *then* $\coprod_{j=1}^m v_j \subseteq L_i$*.*

Proof. In order to distinguish, in a linearization, between positions that correspond to the finite or infinite part of the MSC, we define a formula $x \leq_B y$ such that, for all prefixes u of B-bounded linearizations $w \in \Sigma_{lin}^\omega$ of some MSC $M \in \mathbb{MSC}_{\exists B}^\omega(P, \Sigma)$, and for all positions i and j in u, we have $u, [x \mapsto i, y \mapsto j] \models x \leq_B y$ if and only if the pair of events in M associated to positions (i, j) is in the relation $\leq_B$. We first introduce an MSO$[\Sigma_{lin}, \leq]$ formula $x \lhd y$ with an analogous semantics. The formula simply says that (i) x is a write to channel (p, q) and y a read from channel (p, q), for some $(p, q) \in Ch$, (ii) between x and y, there are less than B messages sent, and read, on channel (p, q), (iii) the count modulo B of messages sent on channel (p, q) up until x, and of messages read from channel (p, q) up until y, are identical. We also let $x \leq_{\mathsf{proc}} y := x \leq y \wedge \mathsf{proc}(x) = \mathsf{proc}(y)$, where $\mathsf{proc}(x) = \mathsf{proc}(y)$ is a simple disjunction on the possible labels of x and y. It is then easy to define $x \leq_B y$ from the formulas $x \lhd y$ and $x \leq_{\mathsf{proc}} y$.

We can now define the regular language $K \subseteq \Sigma_{lin}^*$ by the MSO$[\Sigma_{lin}, \leq]$ formula $\forall x. \exists y. \left(x \leq_B y \wedge \bigvee_{(a,t) \in \Sigma_{lin}^{\mathrm{fin}}} (a, t)(y)\right)$. Notice that K contains all finite parts w^{fin} of words $w \in Lin^B(L)$.

Let $\sim \subseteq K \times K$ be the equivalence relation defined by $u \sim v$ if $u^{-1}Lin^B(L) = v^{-1}Lin^B(L)$. Since $Lin^B(L)$ is regular, $\sim$ is of finite index. We let $K_1, \ldots, K_k$ be the elements of $K/\sim$, and for all $1 \leq i \leq k$, $L_i = \left(K_i^{-1}Lin^B(L)\right) \cap \left(\Sigma_{lin}^{\mathrm{inf}}\right)^\omega$.

Let $w \in Lin^B(L)$. By Lemma 17, we have $w^{\mathrm{fin}}w^{\mathrm{inf}} \in Lin^B(L)$. Moreover, $w^{\mathrm{fin}} \in K$, so there exists i such that $w^{\mathrm{fin}} \in K_i$. Then $w^{\mathrm{inf}} \in K_i^{-1}Lin^B(L) \cap \left(\Sigma_{lin}^{\mathrm{inf}}\right)^\omega$, thus $(w^{\mathrm{fin}}, w^{\mathrm{inf}}) \in K_i \times L_i$. Conversely, if $(u, v) \in K_i \times L_i$, then $u'v \in Lin^B(L)$ for some $u' \in K_i$. Hence, $u \sim u'$ and $w = uv \in Lin^B(L)$. In addition, since v contains only letters from $\Sigma_{lin}^{\mathrm{inf}}$, u contains all positions from w^{fin}. Since u is in K, it also contains only positions from w^{fin}, that is, $u = w^{\mathrm{fin}}$. It follows that $v = w^{\mathrm{inf}}$.

Finally, we prove that each L_i is closed under commutation of letters in distinct subalphabets Σ_{lin}^j. Let $(v_1, \ldots, v_m) \in \left(\Sigma_{lin}^1\right)^\omega \times \cdots \times \left(\Sigma_{lin}^m\right)^\omega$ such that there exists some $v \in L_i \cap \coprod_{j=1}^m v_j$. Since $v \in L_i$, there exist $u \in K_i$ and $w \in Lin^B(L)$ such that $u = w^{\mathrm{fin}}$ and $v = w^{\mathrm{inf}}$. Since the alphabets Σ_{lin}^j are disjoint, this implies $v_j = w^j$ for all j. By Lemma 17, this means $u \cdot \coprod_{j=1}^m v_j \subseteq Lin^B(L)$, hence $\coprod_{j=1}^m v_j \subseteq u^{-1}Lin^B(L) \cap \left(\Sigma_{lin}^{\mathrm{inf}}\right)^\omega = L_i$. □

To further decompose each L_i according to the partition of w^{inf} into $w^1, \ldots, w^m$, we apply the lemma below, proven e.g. in [Kus03, Theorem 4.11].

Lemma 19. *Let* R *be an* ω*-regular language over a finite alphabet* $\Sigma = \Sigma_1 \uplus \cdots \uplus \Sigma_m$*. Suppose that every word from* R *contains every letter from* Σ *infinitely often and that, for all* $(u_1, \ldots, u_m) \in \Sigma_1^\omega \times \cdots \times \Sigma_m^\omega$*,* $u_1 \sqcup\!\sqcup \cdots \sqcup\!\sqcup u_m \cap R \neq \emptyset$ *implies* $u_1 \sqcup\!\sqcup \cdots \sqcup\!\sqcup u_m \subseteq R$*. Then* R *is a finite union of languages of the form* $R_1 \sqcup\!\sqcup \cdots \sqcup\!\sqcup R_m$*, where* R_j *is an* ω*-regular language over* Σ_j*.*

Lemma 20. *If* $Lin^B(L)$ *is an* ω*-regular language then there is a finite sequence of tuples of regular languages* $(K_1', L_i^1, \ldots, L_i^m)_{1 \leq i \leq \ell}$ *such that*

$$\left\{(w^{\mathrm{fin}}, w^1, \ldots, w^m) \mid w \in Lin^B(L)\right\} = \bigcup_{1 \leq i \leq \ell} K_i' \times L_i^1 \times \cdots \times L_i^m .$$

Proof. We apply Lemma 18 and then Lemma 19 to each L_i. We obtain a finite family $(K_i', L_i^1, \ldots, L_i^m)_{1 \leq i \leq \ell}$ such that all $K_i' \subseteq \Sigma_{lin}^*$ are regular languages and all $L_i^j \subseteq (\Sigma_{lin}^j)^\omega$ are ω-regular languages, and

$$\left\{(w^{\mathrm{fin}}, w^{\mathrm{inf}}) \mid w \in Lin^B(L)\right\} = \bigcup_{1 \leq i \leq k} K_i' \times \coprod_{j=1}^m L_i^j .$$

Moreover, for all $w \in Lin^B(L)$, $(w^1, \ldots, w^m)$ is the unique decomposition of w^{inf} as a shuffle of words in the subalphabets $\Sigma_{lin}^1, \ldots, \Sigma_{lin}^m$, which means that $w^{\mathrm{inf}} \in \coprod_{j=1}^m L_i^j$ if and only if $(w^1, \ldots, w^m) \in L_i^1 \times \cdots \times L_i^m$. We obtain

$$\left\{(w^{\mathrm{fin}}, w^1, \ldots, w^m) \mid w \in Lin^B(L)\right\} = \bigcup_{1 \leq i \leq \ell} K_i' \times L_i^1 \times \cdots \times L_i^m . \qquad \square$$

Since all languages K_i' and L_i^j from Lemma 20 are regular or ω-regular, they can be defined in EMSO$[\Sigma_{lin}, \leq]$. This translates into a set of tuples of EMSO$[\Sigma_{lin}, \lhd, \leq]$ formulas over MSC decompositions (using the canonical decomposition similarly to the proof for finite MSCs):

Lemma 21. *If $Lin^B(L)$ is an ω-regular language then there exists a finite set of tuples of* $\mathrm{EMSO}[\Sigma_{lin}, P, \lhd, \leq]$ *sentences* $(\Phi_i, \Psi_i^1, \ldots, \Psi_i^m)_{1 \leq i \leq \ell}$ *such that for all* $M \in \mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$ *with* $\mathsf{Types}^{\inf}(M) = T$*, we have* $M \in L$ *if and only if there exists* i *such that* $M^{\mathrm{fin}} \models \Phi_i$ *and* $M^j \models \Psi_i^j$ *for all* $1 \leq j \leq m$.

Proof. Let $(K_1', L_i^1, \ldots, L_i^m)_{1 \leq i \leq \ell}$ be as in Lemma 20. There are $\mathrm{EMSO}[\Sigma_{lin}, \leq]$ formulas $\widehat{\Phi}_i, \widehat{\Psi}_i^1, \ldots, \widehat{\Psi}_i^m$ such that $K_i' = \mathbb{L}(\Phi_i)$ and $L_i^j = \mathbb{L}(\Psi_i^j)$ for all i, j. Let $\Phi_i, \Psi_i^1, \ldots, \Psi_i^m$ be the $\mathrm{EMSO}[\Sigma_{lin}, P, \leq, \lhd]$ formulas obtained by replacing, in these $\mathrm{EMSO}[\Sigma_{lin}, \leq]$ formulas, every predicate $x \leq y$ with the $\mathrm{FO}[\Sigma_{lin}, P, \leq, \lhd]$ formula defining the canonical linearization $\leq_B$.

Let $M \in \mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$ with types at infinity T, and let $u \in \Sigma_{lin}^*, v_1 \in (\Sigma_{lin}^1)^{\omega}, \ldots, v_m \in (\Sigma_{lin}^m)^{\omega}$ be the canonical linearizations of the MSCs $M^{\inf}, M^1, \ldots, M^m$ (by Lemma 16, this is well-defined). Let $w \in u \cdot \coprod_{j=1}^m v_j$. Then w is a B-bounded linearization of M.

We have $M \in L$ if and only if $w \in Lin^B(L)$, which means, by Lemma 20, if and only if there exists i such that $u \models \widehat{\Phi}_i$ and $v_j \models \widehat{\Psi}_i^j$ for all j, that is, if and only if there exists i such that $M^{\inf} \models \Phi_i$ and $M^j \models \Psi_i^j$ for all j. □

Finally, we conclude the proof of Theorem 5 (3) $\Longrightarrow$ (1). First, we can assume that all MSCs $M \in L$ have the same set $T = \mathsf{Types}^{\inf}(M)$ of types at infinity. Indeed, properties (1), (2), and (3) hold for L if and only if they hold for every restriction of L to a particular set T of types at infinity. Now, by Theorem 4, definability of $\mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$ and Lemma 21, it is now enough to construct an $\mathrm{EMSO}[\Sigma, P, \lhd, \leq]$ formula Ξ such that, for all $M \in \mathbb{MSC}^{\omega}_{\exists B}(P, \Sigma)$, we have $M \models \Xi$ if and only if there exists i such that $M^{\mathrm{fin}} \models \Phi_i$ and $M^j \models \Psi_i^j$ for all $1 \leq j \leq m$. First, we can define $\mathrm{FO}[\Sigma, P, \lhd, \leq]$ formulas $\mathit{fin}(x), \mathit{inf}^1(x), \ldots, \mathit{inf}^m(x)$ that hold precisely at events in $M^{\mathrm{fin}}, M^1, \ldots, M^m$, respectively. Let $\widetilde{\Phi}_i$ be the $\mathrm{FO}[\Sigma, P, \lhd, \leq]$ formula obtained from Φ_i by (i) restricting every quantification to events in M^{fin}, for instance, replacing $\exists x.\xi$ with $\exists x.\mathit{fin}(x) \wedge \xi$, and (ii) replacing Σ_{lin} predicates in a straightforward way, for instance, the formula $(a, p?q)(x)$ is replaced with $a(x) \wedge p(x) \wedge \exists y.q(y) \wedge y \lhd x$. We define similarly formulas $\widetilde{\Psi}_i^j$ relativized to M^j. We then let

$$\Xi = \bigvee_{1 \leq i \leq \ell} \left(\widetilde{\Phi}_i \wedge \bigwedge_{1 \leq j \leq m} \widetilde{\Psi}_i^j \right).$$

5.2. Temporal Logic

The transformation of temporal-logic formulas into automata has many applications, ranging from synthesis to verification. Temporal logics are well understood in the realm of sequential systems where formulas can reason about linearly ordered sequences of events. As we have seen, executions of concurrent systems are actually partially ordered. Over partial orders, however, there is no longer a canonical temporal logic like LTL over words. Several natural temporal logics have been studied over Mazurkiewicz traces (see [GK07] for an overview). Starting from a formula in all these logics, we can always construct an equivalent asynchronous automaton [Zie87], a standard model of shared-memory systems. We will show below that this is still true when formulas are interpreted over MSCs and the system model is given in terms of CFMs.

Many temporal logics over partial orders are captured by the following generic language, which we will call $\mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}})$. Its formulas are defined as follows:

$$\varphi ::= a \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathsf{Co}\,\varphi \mid \langle \lhd \rangle \varphi \mid \langle \lhd^{-1} \rangle \varphi \mid \varphi\, \tilde{\mathsf{U}}\, \varphi \mid \varphi\, \tilde{\mathsf{S}}\, \varphi \qquad \text{where } a \in \Sigma,\ p \in P\,.$$

A formula $\varphi \in \mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}})$ is interpreted over events of MSCs. We say that $M, e \models a$ if $\lambda(e) = a$; similarly, $M, e \models p$ if $\mathit{loc}(e) = p$. The Co modality jumps to a parallel event: $M, e \models \mathsf{Co}\,\varphi$ if there exists $f \in E$ such that $e \not\leq f$, $f \not\leq e$, and $M, f \models \varphi$. The message modality goes to the matching receive: $M, e \models \langle \lhd \rangle \varphi$ if $e \lhd f$ for some $f \in E$ such that $M, f \models \varphi$. The definition is symmetric for $\langle \lhd^{-1} \rangle \varphi$. We use strict versions of until and since:

$$\begin{aligned}
M, e \models \varphi_1\, \tilde{\mathsf{U}}\, \varphi_2 \quad &\text{if} \quad \text{there exists } f \in E \text{ such that } e < f \text{ and } M, f \models \varphi_2 \\
&\qquad \text{and, for all } e < g < f,\ M, g \models \varphi_1 \\
M, e \models \varphi_1\, \tilde{\mathsf{S}}\, \varphi_2 \quad &\text{if} \quad \text{there exists } f \in E \text{ such that } f < e \text{ and } M, f \models \varphi_2 \\
&\qquad \text{and, for all } f < g < e,\ M, g \models \varphi_1\,.
\end{aligned}$$

We define derived modalities X_p, Y_p and U_p, with the following meaning: X_p moves to the first event on process p that is in the strict future of the current event, while Y_p moves to the last event on process p that is in the strict past of the current event; finally, U_p is the usual LTL (non-strict) until for a single process p, evaluated at the current event if it is on process p, or the first event of its future that is on process p otherwise:

$$\begin{aligned}\mathsf{X}_p\,\varphi &:= \neg p\;\tilde{\mathsf{U}}\;(p \wedge \varphi)\\ \mathsf{Y}_p\,\varphi &:= \neg p\;\tilde{\mathsf{S}}\;(p \wedge \varphi)\\ \varphi_1\,\mathsf{U}_p\,\varphi_2 &:= p \wedge \Big(\varphi_2 \vee (\varphi_1 \wedge (\neg p \vee \varphi_1)\;\tilde{\mathsf{U}}\;(p \wedge \varphi_2))\Big).\end{aligned}$$

This temporal logic and others have been studied in the context of Mazurkiewicz traces [GK07, GK10, Thi94, DG06]. The logic introduced by Thiagarajan in [Thi94] uses an until modality similar to U_p, except that if the current event is not on process p, the evaluation starts at the latest event on process p in the past of the current event (or the first event of process p if none exists). The second temporal modality of this logic is a unary modality O_p interpreted as follows: $O_p\,\varphi$ holds at e if the first event on process p that is not in the past of e satisfies φ. Both can similarly be expressed in $\mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}})$.

All these modalities can be easily translated into $\mathrm{FO}[\rightarrow, \lhd, \leq]$, and thus we can apply Theorem 1 and Proposition 3 to translate any $\mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}})$ formula into a transducer which determines the set of events where the formula holds. However, this approach does not give any guarantee on the size of the resulting transducer. Instead, we present a direct translation from $\mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}})$ to $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$, leading to a transducer of size exponential in the size of the formula and doubly exponential in the number of processes.

Theorem 6. *For all $\varphi \in \mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}}, \mathsf{X}_p, \mathsf{Y}_p, \mathsf{U}_p)$, there exists a transducer $\mathcal{A}_\varphi$ with $2^{|\varphi| \cdot 2^{O(|P| \log |P|)}}$ states per process such that $[\![\mathcal{A}_\varphi]\!] = \{(M, M_\varphi) \mid M \in \mathbb{MSC}(P, \Sigma)\}$.*

Proof. Similarly to the proof of Lemma 10, we will first translate every modality Mod into a transduceer $\mathcal{B}_{\mathsf{Mod}}$. In particular, if Mod is a unary modality, then $\mathcal{B}_{\mathsf{Mod}}$ is a transducer from $\{0,1\}$ to $\{0,1\}$, and if it is binary, then $\mathcal{B}_{\mathsf{Mod}}$ is a transducer from $\{0,1\}^2$ to $\{0,1\}$. Consider, for example, $\mathsf{Mod} = \mathsf{Co}$. For $M = (E, \rightarrow, \lhd, loc, \lambda \times \gamma)$ with $\lambda, \gamma : E \to \{0,1\}$, we will have $M \in \mathbb{L}(\mathcal{B}_{\mathsf{Co}})$ iff, for all events $e \in E$,

$$M, e \models \mathit{out} \iff M, e \models \mathsf{Co}\,\mathit{in}$$

with temporal-logic formulas $\mathit{out} = \bigvee_{a \in \{0,1\}} (a, 1)$ and $\mathit{in} = \bigvee_{b \in \{0,1\}} (1, b)$. Now suppose $\mathsf{Mod} = \tilde{\mathsf{U}}$. Then, for $M = (E, \rightarrow, \lhd, loc, \lambda \times \gamma)$ with $\lambda : E \to \{0,1\}^2$ and $\gamma : E \to \{0,1\}$, we will get $M \in \mathbb{L}(\mathcal{B}_{\tilde{\mathsf{U}}})$ iff, for all events $e \in E$,

$$M, e \models \mathit{out} \iff M, e \models \mathit{in}_1\;\tilde{\mathsf{U}}\;\mathit{in}_2\,.$$

where $\mathit{out} = \bigvee_{a,b \in \{0,1\}} ((a,b), 1)$, $\mathit{in}_1 = \bigvee_{a,b \in \{0,1\}} ((1,a), b)$, and $\mathit{in}_2 = \bigvee_{a,b \in \{0,1\}} ((a,1), b)$. In the following, we will exploit that, in the unary and the binary case, the formulas out, in, in_1, and in_2 can also be considered as event formulas from $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$.

The number of states per process of $\mathcal{B}_{\mathsf{Mod}}$ will be bounded by $2^{2^{O(|P| \log |P|)}}$. We then compose these transducers for a given formula $\varphi \in \mathrm{TL}(\mathsf{Co}, \lhd, \lhd^{-1}, \tilde{\mathsf{U}}, \tilde{\mathsf{S}}, \mathsf{X}_p, \mathsf{Y}_p, \mathsf{U}_p)$, just like in the proof of Lemma 10, so that we finally obtain the desired transducer $\mathcal{A}_\varphi$ with $2^{|\varphi| \cdot 2^{O(|P| \log |P|)}}$ states per process. Note that this procedure is in the spirit of [GK07, GK10], where formulas from temporal logics with MSO-definable modalities over Mazurkiewicz traces are translated into small Büchi automata.

We obtain $\mathcal{B}_{\mathsf{Mod}}$ as the transducer $\mathcal{A}_{\xi_{\mathsf{Mod}}}$ according to Theorem 2, where

$$\xi_{\mathsf{Mod}} \;=\; \mathsf{A}\,(\mathit{out} \iff \psi_{\mathsf{Mod}})$$

is a $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ sentence of size $2^{O(|P| \log |P|)}$. It remains to specify the event formulas ψ_{Mod}, which we address in the following.

Let Π denote the set of $\mathrm{PDL}_{\mathsf{sf}}[\emptyset]$ path formulas

$$\pi = \pi_1 \cdot \lhd_{p_1,p_2} \cdot \xrightarrow{+} \cdot \lhd_{p_2,p_3} \cdots \xrightarrow{+} \cdot \lhd_{p_{m-1},p_m} \cdot \pi_2$$

with $1 \leq m \leq |P|$, $p_1, \ldots, p_m \in P$ and $p_i \neq p_j$ for $1 \leq i < j \leq m$, $\pi_1, \pi_2 \in \{\xrightarrow{+}, \{true\}?\}$ and $\pi \not\equiv \{true\}?$. Note that for all events g, h, we have $g < h$ if and only if $M, g, h \models \pi$ for some $\pi \in \Pi$. Given $\pi \in \Pi$, we then define a state formula $\mathsf{is\text{-}next}(\pi)$ in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ such that for all events e, we have $M, e \models \mathsf{is\text{-}next}(\pi)$ if and only if $\min[\![\pi]\!](e)$ is well-defined, and is the minimal event on its process which is in the future of e:

$$\mathsf{is\text{-}next}(\pi) = \langle \pi \rangle \wedge \bigwedge_{\pi' \in \Pi} \neg \mathsf{Loop}(\min \pi \cdot \xleftarrow{+} \cdot (\pi')^{-1}) \,.$$

Symmetrically, we let

$$\mathsf{is\text{-}latest}(\pi) = \langle \pi^{-1} \rangle \wedge \bigwedge_{\pi' \in \Pi} \neg \mathsf{Loop}(\max (\pi^{-1}) \cdot \xrightarrow{+} \cdot \pi')$$

so that $M, e \models \mathsf{is\text{-}latest}(\pi)$ if and only if there exists $p \in P$ such that $[\![\max \pi^{-1}]\!](e) = \max\{g \in E_p \mid g < e\}$. Note that $\emptyset \neq [\![\pi^{-1}]\!](e) \subseteq \{g \in E_p \mid g < e\}$ which is finite, hence its maximum is well-defined. Given the special shape of π, we can simplify the definitions of $\min \pi$ and $\max \pi^{-1}$ to obtain formulas of size $O(|P|)$, so that both $\mathsf{is\text{-}next}(\pi)$ and $\mathsf{is\text{-}latest}(\pi)$ are of size $2^{O(|P| \log |P|)}$. For instance, if $\pi = \lhd_{p_1,p_2} \cdot \xrightarrow{+} \cdot \lhd_{p_2,p_3} \cdots \xrightarrow{+} \cdot \lhd_{p_{m-1},p_m} \cdot \xrightarrow{+}$, we have

$$\max \pi^{-1} \equiv \xleftarrow{\{\neg \langle \lhd^{-1}_{p_{m-1},p_m} \rangle\}?} \cdot \lhd^{-1}_{p_{m-1},p_m} \cdot \xleftarrow{\{\neg \langle \lhd^{-1}_{p_{m-2},p_{m-1}} \rangle\}?} \cdot \lhd^{-1}_{p_{m-2},p_{m-1}} \cdots \lhd^{-1}_{p_1,p_2} \,.$$

We now determine the formulas ψ_{Mod} for each modality in the temporal logic. The base cases and message modalities are trivial.

- Suppose that $\mathsf{Mod} = \mathsf{X}_p$ (the case $\mathsf{Mod} = \mathsf{Y}_p$ is similar). We let

$$\psi_{\mathsf{X}_p} = \bigvee_{\pi \in \Pi} \langle \pi \rangle\, p \wedge \mathsf{is\text{-}next}(\pi) \wedge \langle \min \pi \rangle\, in \,,$$

  where, as above, $in = \bigvee_{b \in \{0,1\}} (1, b)$. The formula ψ_{X_p} is of size $2^{O(|P| \log |P|)}$. Notice that if we are already on process p then we get a simpler, constant size, formula $\psi_{p \wedge \mathsf{X}_p} = \langle \rightarrow \rangle\, in$.

- Suppose that $\mathsf{Mod} = \mathsf{U}_p$. We use the constant size formula

$$\psi_{\mathsf{U}_p} = p \wedge \Big(in_2 \vee (in_1 \wedge \langle \xrightarrow{in_1} \rangle\, in_2) \Big) \,.$$

- Suppose that $\mathsf{Mod} = \tilde{\mathsf{U}}$ (the case $\mathsf{Mod} = \tilde{\mathsf{S}}$ is similar). Notice that in order to determine if $\psi_{\tilde{\mathsf{U}}}$ is true at a given event e, it suffices to consider all potential "minimal" events $f > e$ such that $M, f \models in_2$ and check whether $M, g \models in_1$ for all $e < g < f$. More precisely, we have $M, e \models in_1\ \tilde{\mathsf{U}}\ in_2$ if and only if there exists $\pi \in \Pi$ such that $f = [\![\min (\pi \cdot \{in_2\}?)]\!](e)$ is well-defined and for all $e < g < f$, $M, g \models in_2$, that is, if and only if $M, e \models \psi_{\tilde{\mathsf{U}}}$, where

$$\psi_{\tilde{\mathsf{U}}} = \bigvee_{\pi \in \Pi} \langle \pi \rangle\, in_2 \wedge \bigwedge_{\sigma, \tau \in \Pi} \neg \mathsf{Loop}(\sigma \cdot \{\neg in_1\}? \cdot \tau \cdot (\min (\pi \cdot \{in_2\}?))^{-1}) \,.$$

  The formula $\psi_{\tilde{\mathsf{U}}}$ is of size $2^{O(|P| \log |P|)}$.

- Suppose that $\mathsf{Mod} = \mathsf{Co}$. Given $e, f \in E$ with $loc(f) = q \neq loc(e)$, we have $f \parallel e$ if and only if one of the following holds: (a) all events on process q are parallel with e, (b) no event on q is in the past of e, and $f <_{\mathsf{proc}} \min\{g \in E_q \mid e < g\}$, (c) no event on q is in the future of e, and $\max\{g \in E_q \mid g < e\} <_{\mathsf{proc}} f$,

(d) $\max\{g \in E_q \mid g < e\} <_{\mathsf{proc}} f <_{\mathsf{proc}} \min\{g \in E_q \mid e < g\}$. This leads to the following definition:

$$\psi_{\mathsf{Co}} = \bigvee_{\substack{p \neq q \\ \sigma,\tau \in \Pi \setminus \{\xrightarrow{+}\}}} \Bigg(\langle \mathsf{jump}_{p,q} \rangle\, in \wedge \bigwedge_{\pi \in \Pi} \neg \langle \pi \rangle\, q \wedge \neg \langle \pi^{-1} \rangle\, q \Bigg) \vee$$
$$\Bigg(\mathsf{is\text{-}next}(\tau) \wedge \langle \min\, \tau \cdot \xleftarrow{+} \rangle\, (q \wedge in) \wedge \bigwedge_{\pi \in \Pi} \neg \langle \pi^{-1} \rangle\, q \Bigg) \vee$$
$$\Bigg(\mathsf{is\text{-}latest}(\sigma) \wedge \langle \max\, (\sigma^{-1}) \cdot \xrightarrow{+} \rangle\, (q \wedge in) \wedge \bigwedge_{\pi \in \Pi} \neg \langle \pi \rangle\, q \Bigg) \vee$$
$$\Big(\mathsf{is\text{-}latest}(\sigma) \wedge \mathsf{is\text{-}next}(\tau) \wedge \mathsf{Loop}(\max\, (\sigma^{-1}) \cdot \xrightarrow{+} \cdot \{in\}? \cdot \xrightarrow{+} \cdot (\min\, \tau)^{-1}) \Big)\,.$$

The formula ψ_{Co} is of size $2^{O(|P| \log |P|)}$.

This concludes the proof of Theorem 6. □

Note that the transducer constructed in Theorem 6 is non-deterministic. Informally, a transducer from Σ to Γ is *deterministic* if it can be obtained from a deterministic CFM over P and Σ by adding one output to each transition. So here a first source of non-determinism is that all the transducers we constructed essentially "guess" their output. However, even if we fix the output as part of the MSC and only want to construct a CFM which checks that the output is correct, it is not possible to avoid non-determinism. We cannot even avoid it for simple past formulas, which is in contrast to what happens for words or Mazurkiewicz traces.

Proposition 4. *Assume that $|\Sigma| \geq 2$ and $|P| \geq 3$. For $p \in P$ and $a \in \Sigma$, there is no deterministic CFM $\mathcal{A}$ over $\Sigma \times \{0,1\}$ such that $L(\mathcal{A}) = \{(E, \rightarrow, \lhd, loc, \lambda \times \gamma) \mid \gamma(e) = 1$ iff $(E, \rightarrow, \lhd, loc, \lambda), e \models \mathsf{Y}_p\, a\}$.*

Proof. Let $P = \{p, q, r\}$ and $\Sigma = \{0, 1\}$. We show that there exists no deterministic CFM recognizing the set L of MSCs $M = (E, \rightarrow, \lhd, loc, \lambda \times \gamma)$ such that for all $e \in E_q$, $\gamma(e) = 1$ if and only if $(E, \rightarrow, \lhd, loc, \lambda), e \models \mathsf{Y}_p\, 1$. Assume that there exists a deterministic CFM $\mathcal{A} = (\mathcal{A}_p, \mathcal{A}_q, \mathcal{A}_r, Msg, Acc)$ such that $L(\mathcal{A}) = L$. Fix $n > |S_q|^2$, where S_q is the set of states of $\mathcal{A}_q$. For all $k \in \{0, \ldots, n-1\}$, we define an MSC $M^k = (E, \rightarrow, \lhd^k, loc, \lambda \times \gamma^k)$, as depicted in Figure 16 (for $n = 5$ and $k = 2$):

- $E_p = \{e_i \mid 0 \leq i < 2n\}$, $E_q = \{f_i \mid 0 \leq i < 2n\}$, and $E_r = \{g_i \mid 0 \leq i < 2n\}$, with $e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_{2n-1}$, $f_0 \rightarrow f_1 \rightarrow \cdots \rightarrow f_{2n-1}$, and $g_0 \rightarrow g_1 \rightarrow \cdots \rightarrow g_{2n-1}$.
- For all $0 \leq i < k$, $e_{2i} \lhd^k f_i$, and for all $k \leq i < n$, $e_{2i} \lhd^k f_{n+i}$.
  For all $0 \leq i < n$, $e_{2i+1} \lhd^k g_{2i}$, and $g_{2i+1} \lhd^k f_{k+i}$.
- For all $0 \leq i < n$, $\lambda(e_{2i}) = 0$ and $\lambda(e_{2i+1}) = 1$.
  For all $h \in E_r \cup E_q$, $\lambda(h) = 0$.
- For all $0 \leq i < 2k-1$, $\gamma^k(f_i) = 0$, and for all $2k-1 \leq i < 2n$, $\gamma^k(f_i) = 1$.
  For all $h \in E_p \cup E_r$, $\gamma^k(h) = 0$.

Clearly, $M^k \in L(\mathcal{A})$. Let s_k and t_k be the states before reading respectively f_k and f_{k+n} in the unique run ρ^k of $\mathcal{A}$ on M^k: $s_k = source(\rho^k(f_k))$ and $t_k = source(\rho^k(f_{k+n}))$.

Note that for all k, the sequence of send and receive actions performed by process p or process r in M^k are the same, so the runs of $\mathcal{A}$ on MSCs M^k only differ on process q. In particular, the sequence of n messages sent by process r to process q is the same for all k. Moreover, since $n > |S_q|^2$, there exist $0 \leq k < k' < n$ such that $s_k = s_{k'}$ and $t_k = t_{k'}$. We can then combine the runs of $\mathcal{A}$ on M^k and $M^{k'}$ to define a run where process q receives the messages from process p and r in the same order as in M^k, but behaves as in $M^{k'}$ in the middle part where it receives the n messages from process r. More precisely, let $M = (E, \rightarrow, \lhd^k, loc, \lambda \times \gamma)$, where $(E, \rightarrow, \lhd^k, loc, \lambda)$ is as in M^k, and γ is defined as follows: $\gamma(h) = 0$ for all $h \in E_p \cup E_r$, $\gamma(f_i) = 0$ for all $0 \leq i < k + k' - 1$, and $\gamma(f_i) = 1$ for all $k + k' - 1 \leq i < n$. Then, $M \in L(\mathcal{A})$, but $M \notin L$. □

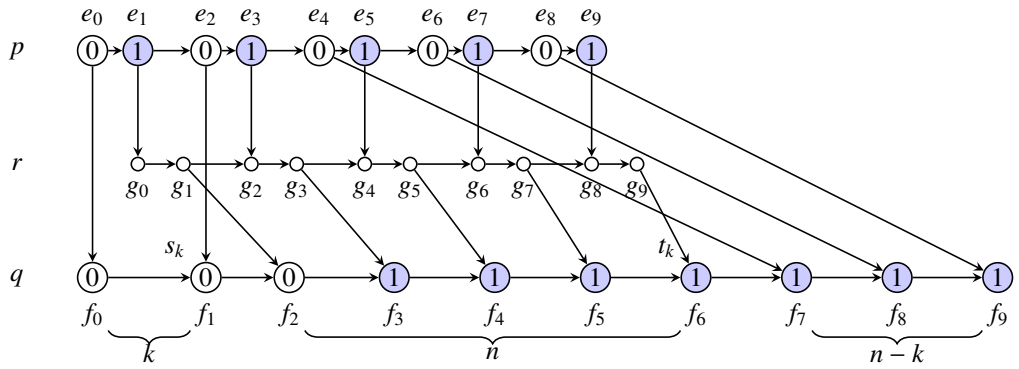


Figure 16: Definition of M^k. We only indicate the value of λ on process p, and of γ^k on process q.

5.3. The Gossip Problem

Gossiping is a technique used to maintain a consistent view of the global system state in a distributed system. The problem can be stated as follows: whenever process q receives a message from process r, q has to decide, for all processes p, whether it has more recent information on p than r. This problem is at the heart of many distributed algorithms. Interestingly, gossip protocols and related techniques, such as asynchronous mappings, have also been exploited in formal methods, in particular when it comes to establishing the expressive power of automata models [MS97, CMZ93, MKS03, DS97]. In particular, gossip protocols are the key to simulating high-level specifications, which include message sequence graphs and monadic second-order logic [HMK$^+$05, GKM06, Kus03, Zie87, Tho90]. All known techniques and algorithms, however, require that communication be synchronous or accomplished through FIFO channels with *limited* capacity.

We show that we can apply our results to construct a CFM that solves the gossip problem. This is defined more precisely below. Let $M = (E, \rightarrow, \lhd, loc, \lambda)$ be an MSC and $e \in E$. For all processes p such that $\{f \in E_p \mid f < e\} \neq \emptyset$, we let $\mathsf{latest}_p(e) = \max\{f \in E_p \mid f < e\}$. The *gossip transducer* should determine, for all processes p and all receive events e with $f \lhd e$, whether $\mathsf{latest}_p(e) < \mathsf{latest}_p(f)$. We show that this property can be expressed in $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ so that we can obtain the gossip transducer as a corollary of Proposition 3.

Let Π and $\mathsf{is\text{-}latest}(\pi)$ be defined as in the proof of Theorem 6. The property described above is expressed by the $\mathrm{PDL}_{\mathsf{sf}}[\mathsf{Loop}]$ formula below. It states that the event $\mathsf{latest}_p(e)$ is obtained from e with $\mathsf{max}\,(\sigma^{-1})$ and the event $\mathsf{latest}_p(f)$ is obtained from f with $\mathsf{max}\,(\tau^{-1})$. Then, it compares the two events using the loop modality.

$$\bigvee_{\substack{\sigma,\tau\in\Pi\\ q,r\in P}} \langle\sigma^{-1}\rangle\, p \wedge \mathsf{is\text{-}latest}(\sigma) \wedge \langle \lhd_{q,r}^{-1}\rangle\, (\langle\tau^{-1}\rangle\, p \wedge \mathsf{is\text{-}latest}(\tau)) \\ \wedge\, \mathsf{Loop}(\mathsf{max}\,(\sigma^{-1}) \cdot \xrightarrow{+} \cdot\, (\mathsf{max}\,(\tau^{-1}))^{-1} \cdot \lhd_{q,r})\,.$$

Note that the gossip CFM is unavoidably nondeterministic (this follows from Proposition 4). This is in contrast to the deterministic protocols for synchronous communication or message-passing environments with bounded channels [MS97, CMZ93, MKS03, DS97].

6. Conclusion

In this paper, we showed that every $\mathrm{FO}[\rightarrow, \lhd, \leq]$ formula over MSCs is effectively equivalent to a CFM. As an intermediate step, we used a purely logical transformation of own interest, relating FO logic with a star-free fragment of PDL. While star-free PDL constitutes a two-dimensional temporal logic over MSCs, we leave open whether there is a one-dimensional one, with a finite set of FO-definable modalities, that is expressively complete for $\mathrm{FO}[\rightarrow, \lhd, \leq]$.

Though our result closes an important gap concerning the expressive power of CFMs, there remain interesting open questions addressing both CFMs and automata on graphs in general:

First, it is still open whether every formula from the full PDL logic over MSCs can be translated into CFMs. In [BKM10], only a unidirectional fragment was considered. The difficulty comes with unrestricted usage of the

star operator, which allows one to go forth and back in an MSC unboundedly many times. While such two-way mechanisms do not add expressive power in the setting of words, the situation is unclear in the realm of MSCs. It would already be interesting to solve this question for more specialized structures such as pictures, which also come with natural notions of recognizability in terms of graph acceptors and two-way automata [KM01].

Second, it is worthwhile to also study architectures with one unbounded FIFO channel per process (as considered, e.g., in [BEJQ18]), or including pushdown processes. The latter give rise to multiply nested words, and it is an open question whether every first-order formula over multiply nested words (including the total order and the push-pop relation) can be translated into an equivalent multi-pushdown automaton (aka nested-word automaton). Unlike in MSCs, the matching relation of a nested word is not monotone so that the techniques presented in this paper do not apply. Note that, when dropping the total order and restricting to two nesting relations, one can still take advantage of Hanf's theorem [Bol08].

Finally, it will be interesting to see whether the technique from Section 5 can be applied to other meaningful classes of MSCs so as to obtain logical characterizations of restricted CMFs in terms of full MSO logic.

References

[AM09] R. Alur and P. Madhusudan. Adding nesting structure to words. *Journal of the ACM*, 56(3):1–43, 2009.

[Ara98] João Araújo. Formalizing sequence diagrams. In *Proceedings of the OOPSLA'98 Workshop on Formalizing UML. Why? How?*, volume 33, 10 of *ACM SIGPLAN Notices*, New York, 1998. ACM Press.

[AW04] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, 2004.

[BDM+11] M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Transactions on Computational Logic*, 12(4):27, 2011.

[Bed15] N. Bedon. Logic and branching automata. *Logical Methods in Computer Science*, 11(4), 2015.

[BEJQ18] A. Bouajjani, C. Enea, K. Ji, and S. Qadeer. On the completeness of verifying message passing programs under bounded asynchrony. In *Proceedings of CAV'18, Part II*, volume 10982, pages 372–391. Springer, 2018.

[BFG18] B. Bollig, M. Fortin, and P. Gastin. Communicating finite-state machines and two-variable logic. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics*, pages 17:1–17:14. Leibniz-Zentrum für Informatik, 2018.

[BK08] B. Bollig and D. Kuske. Muller message-passing automata and logics. *Information and Computation*, 206(9-10):1084–1094, 2008.

[BKM10] B. Bollig, D. Kuske, and I. Meinecke. Propositional dynamic logic for message-passing systems. *Logical Methods in Computer Science*, 6(3:16), 2010.

[BL06] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006.

[Bol08] B. Bollig. On the expressive power of 2-stack visibly pushdown automata. *Logical Methods in Computer Science*, 4(4:16), 2008.

[BS10] H. Björklund and T. Schwentick. On notions of regularity for data languages. *Theoretical Computer Science*, 411(4-5):702–715, 2010.

[Büc60] J. Büchi. Weak second order logic and finite automata. *Z. Math. Logik, Grundlag. Math.*, 5:66–62, 1960.

[BZ83] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2), 1983.

[CMZ93] R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Information and Computation*, 106:159–202, 1993.

[DG06] V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for mazurkiewicz traces. *Information and Computation*, 204(11):1597–1619, 2006.

[DG08] V. Diekert and P. Gastin. First-order definable languages. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.

[DL94] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1.*, pages 205–212. AAAI Press / The MIT Press, 1994.

[DR95] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[DS97] D. Dolev and N. Shavit. Bounded concurrent time-stamping. *SIAM J. Comput.*, 26(2):418–455, 1997.

[EL87] E. A. Emerson and C.-L. Lei. Modalities for model checking: branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306, Jun 1987.

[Elg61] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–52, 1961.

[FL79] M. J. Fischer and R. E. Ladner. Propositional Dynamic Logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[Gab81] D. M. Gabbay. Expressive functional completeness in tense logic. In Uwe Mönnich, editor, *Aspects of Philosophical Logic: Some Logical Forays into Central Notions of Linguistics and Philosophy*, pages 91–117. Springer Netherlands, Dordrecht, 1981.

[GHR94] D. M. Gabbay, I. Hodkinson, and M. A. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, vol. 1*. Oxford University Press, 1994.

[GK07] P. Gastin and D. Kuske. Uniform satisfiability in PSPACE for local temporal logics over Mazurkiewicz traces. *Fundamenta Informaticae*, 80(1-3):169–197, 2007.

[GK10] P. Gastin and D. Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. *Information and Computation*, 208(7):797–816, 2010.

[GKM06] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204(6):920–956, 2006.

[GKM07] B. Genest, D. Kuske, and A. Muscholl. On communicating automata with bounded channels. *Fundamenta Informaticae*, 80(1-3):147–167, 2007.

[GLL09] S. Göller, M. Lohrey, and C. Lutz. PDL with intersection and converse: satisfiability and infinite-state model checking. *Journal of Symbolic Logic*, 74(1):279–314, 2009.

[GO99] E. Grädel and M. Otto. On logics with two variables. *Theoretical Computer Science*, 224(1-2):73–113, 1999.

[Han65] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*. North-Holland, Amsterdam, 1965.

[HJK+15] L. Hella, M. Järvisalo, A. Kuusisto, J. Laurinharju, T. Lempiäinen, K. Luosto, J. Suomela, and J. Virtema. Weak models of distributed computing, with connections to modal logic. *Distributed Computing*, 28(1):31–53, 2015.

[HM92] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artif. Intell.*, 54(2):319–379, 1992.

[HMK+05] J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 202(1):1–38, 2005.

[IT99] ITU-TS. ITU-TS Recommendation Z.120: Message Sequence Chart 1999 (MSC99). Technical report, ITU-TS, Geneva, 1999.

[Kam68] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.

[KM01] J. Kari and C. Moore. New results on alternating and non-deterministic two-dimensional finite-state automata. In *Proceedings of STACS'01*, pages 396–406. Springer, 2001.

[Kus00] D. Kuske. Infinite series-parallel posets: Logic and languages. In *Proceedings of ICALP'00*, volume 1853 of *LNCS*, pages 648–662. Springer, 2000.

[Kus03] D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187:80–109, 2003.

[Kuu13] A. Kuusisto. Modal logic and distributed message passing automata. In *Proceedings of CSL'13*, volume 23 of *LIPIcs*, pages 452–468. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[Lam78] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.

[Lan06] M. Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2006.

[LL05] M. Lange and C. Lutz. 2-ExpTime lower bounds for Propositional Dynamic Logics with intersection. *Journal of Symbolic Logic*, 70(5):1072–1086, 2005.

[LM04] M. Lohrey and A. Muscholl. Bounded MSC Communication. *Information and Computation*, 189(2):160–181, 2004.

[Lyn96] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., 1996.

[Mar05] M. Marx. Conditional XPath. *ACM Trans. Database Syst.*, 30(4):929–959, 2005.

[Men13] R. Mennicke. Propositional dynamic logic with converse and repeat for message-passing systems. *Logical Methods in Computer Science*, 9(2:12):1–35, 2013.

[MKS03] M. Mukund, K. Narayan Kumar, and M. A. Sohoni. Bounded time-stamping in message-passing systems. *Theoretical Computer Science*, 290(1):221–239, 2003.

[MS97] M. Mukund and M. A. Sohoni. Keeping track of the latest gossip in a distributed system. *Distributed Computing*, 10(3):137–148, 1997.

[Ray13] M. Raynal. *Distributed Algorithms for Message-Passing Systems*. Springer, 2013.

[Rei15] F. Reiter. Distributed graph automata. In *Proceedings of LICS'15*, pages 192–201. IEEE Computer Society, 2015.

[Rei17] F. Reiter. Asynchronous distributed automata: A characterization of the modal mu-fragment. In *Proceedings of ICALP'17*, volume 80 of *LIPIcs*, pages 100:1–100:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Sto74] L. J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, MIT, 1974.

[Str81] R. S. Streett. Propositional dynamic logic of looping and converse. In *Proceedings of STOC'81*, pages 375–383. ACM, 1981.

[Tel01] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 2nd edition, 2001.

[Thi94] P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *LICS'94*, pages 438–447. IEEE Computer Society, 1994.

[Tho90] W. Thomas. On logical definability of trace languages. In *Proceedings of Algebraic and Syntactic Methods in Computer Science (ASMICS)*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990.

[Tho97] W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.

[Tra62] B. A. Trakhtenbrot. Finite automata and monadic second order logic. *Siberian Math. J*, 3:103–131, 1962. In Russian; English translation in *Amer. Math. Soc. Transl.* 59, 1966, 23–55.

[TW68] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.

[TW02] P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. *Inf. Comput.*, 179(2):230–249, 2002.

[VW86] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of LICS'86*, pages 332–344. IEEE Computer Society, 1986.

[Zie87] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.