

Benedikt Bollig  
Dietrich Kuske  
Ingmar Meinecke

Propositional Dynamic Logic for  
Message-Passing Systems

Research Report LSV-07-22

June 2007

Laboratoire  
Spécification  
et  
Vérification



CENTRE NATIONAL  
DE LA RECHERCHE  
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan  
61, avenue du Président Wilson  
94235 Cachan Cedex France



# Propositional Dynamic Logic for Message-Passing Systems

Benedikt Bollig<sup>1</sup>, Dietrich Kuske<sup>2</sup>, and Ingmar Meinecke<sup>2</sup>

<sup>1</sup> LSV, ENS Cachan, CNRS  
61, Av. du Président Wilson, F-94235 Cachan Cedex, France,  
bollig@lsv.ens-cachan.fr

<sup>2</sup> Institut für Informatik, Universität Leipzig  
PF 100920, D-04009 Leipzig, Germany,  
{kuske,meinecke}@informatik.uni-leipzig.de

**Abstract.** We examine a bidirectional Propositional Dynamic Logic (PDL) for finite and infinite message sequence charts (MSCs) extending LTL and  $\text{TLC}^-$ . By this kind of multi-modal logic we can express properties both in the entire future and in the past of an event. Path expressions strengthen the classical until operator of temporal logic. For every formula defining an MSC language, we construct a communicating finite-state machine (CFM) accepting the same language. The CFM obtained has size exponential in the size of the formula. This synthesis problem is solved in full generality, i.e., also for MSCs with unbounded channels. The model checking problem for CFMs and HMSCs turns out to be in PSPACE for existentially bounded MSCs. Finally, we show that, for PDL with intersection, the semantics of a formula cannot be captured by a CFM anymore.

## 1 Introduction

Message sequence charts (MSCs) are an important formalism describing the executions of distributed message-passing systems. They are a common notation in telecommunication and defined by an ITU standard [ITU96]. A significant task is to verify certain properties of message-passing systems. The model checking problem asks for an algorithm that decides whether, given a formula  $\varphi$  and a finite machine  $\mathcal{A}$ , every behavior of  $\mathcal{A}$  satisfies  $\varphi$ . Meenakshi and Ramanujam [MR04] showed the undecidability of model checking for the class of all MSCs and very restrictive temporal logics (their result transfers easily from Lamport diagrams to MSCs). In [Pel00,MM01,GMSZ02,GKM06], the model checking problem was tackled successfully for several logics by restricting to existentially  $B$ -bounded MSCs, i.e., to MSCs that can be scheduled such that the channel capacity respects a given size  $B$ . In [Pel00,MM01,GKM06] formulas of temporal logics or monadic second-order logic are translated into machine models such that the semantics of the formula and the behavior of the machine coincide for existentially  $B$ -bounded MSCs (or their linearizations). In this approach, the bound  $B$  is given *a priori* for the implementation of the formula.

In the early stages of system design it seems more natural not to fix a channel size  $B$  but to implement the entire semantics of  $\varphi$ . We therefore follow a different approach here. Given a formula  $\varphi$ , we will construct a communicating finite-state machine  $\mathcal{A}_\varphi$  (a CFM for short) such that  $L(\varphi) = L(\mathcal{A}_\varphi)$  wrt. the class of all (finite and infinite) MSCs. Note that no restriction on the channel capacity is imposed. Once a system is synthesized directly from its specification, it can be assumed to be correct *a priori*. The bound  $B$  is applied only when we come to the actual model checking. Given another CFM  $\mathcal{B}$ , we build a CFM  $\mathcal{A}$  with  $L(\mathcal{A}) = L(\varphi) \cap L(\mathcal{B})$ . Now,  $\mathcal{A}$  is transformed into a finite sequential transition system with multiple Büchi-acceptance condition that accepts exactly the  $B$ -bounded linearizations from  $L(\mathcal{A})$ . Only in this step, the bound  $B$  is used. Clearly, for this finite transition system, we can decide emptiness.

For MSCs, there exist a few attempts to define a suitable temporal logic. Meenakshi and Ramanujam obtained exponential-time decision procedures for several temporal logics over Lamport diagrams (which are similar to MSCs) [MR00,MR04]. Peled [Pel00] considered the fragment  $\text{TLC}^-$  of the temporal logic TLC introduced in [APP95]. Like the logics above, our logic is interpreted directly over MSCs, not over linearizations; it combines elements from [MR04] (global next operator,

past operators) and [Pel00] (global next operator, existential interpretation of the until-operator). But in particular, it is inspired by *dynamic* LTL as introduced by Henriksen and Thiagarajan first for words [HT99]. There, standard LTL is extended by indexing the until operator with a regular expression to make it more flexible. The same authors applied dynamic LTL also to Mazurkiewicz traces but reasoned only about the future of an event in the same process [HT97]. In contrast, we might argue about the whole future of an event rather than about one single process. Moreover, we provide past operators to judge about events that have already been executed. We call our logic PDL because it is essentially the original propositional dynamic logic as first defined by Fischer and Ladner [FL79] but here in the framework of MSCs.

Although PDL can be seen as an extension of Peled’s  $\text{TLC}^-$ , our decision procedure is rather different. Instead of translating a PDL formula  $\varphi$  into a CFM directly, we use an inductive method inspired by [GK03,GK05]: The events of an MSC are colored by additional bits, one for each subformula of  $\varphi$ . Then we construct, for each such subformula  $\gamma$ , a CFM  $\mathcal{A}_\gamma$  whose task it is to check that the bit corresponding to  $\gamma$  is set at precisely those nodes where  $\gamma$  holds. For this, the CFM  $\mathcal{A}_\gamma$  reads the bits corresponding to the top-level subformulas of  $\gamma$ . As an example, let  $\gamma = \alpha \vee \beta$ . In this case, we introduce three 0-1-colorings  $c_\alpha$ ,  $c_\beta$ , and  $c_\gamma$ , and we build a CFM checking  $c_\gamma(v) = c_\alpha(v) \vee c_\beta(v)$  for each event  $v$  of an MSC. Similarly,  $\gamma = \neg\beta$  requires two 0-1-colorings and a CFM that checks  $c_\gamma(v) \neq c_\beta(v)$  for each event  $v$ . The overall CFM is obtained by running synchronously all the CFMs arising from the subformulas.

A typical subformula in PDL is  $\gamma = \langle \pi \rangle \#$  expressing that there is a path starting in the current vertex that corresponds to the path description  $\pi$  (a regular expression). The construction of a CFM for such a forward-path formula turns out to be the most difficult part. The basic idea is to start, in the current node  $v$ , a finite automaton  $\mathcal{A}$  that accepts the language  $L_\pi$  of the regular expression  $\pi$  and to ensure that  $\mathcal{A}$  will eventually reach an accepting state in some event  $v'$ . For sequential systems, this problem can be solved by two verification phases [HT99]: divide the infinite word nondeterministically into infinitely many intervals and make sure that any claim for  $\gamma$  to hold in some position  $v$  is certified by some position  $v'$  that belongs to the current or the following interval. In adapting this idea to our setting, a severe problem arises. We do not deal with just one sequential process but with several ones communicating with each other via channels, i.e., the path described by  $\pi$  can change process arbitrary many times. Hence the nondeterministic division into infinitely many intervals has to be done in such a way that every path in the MSC, even those switching processes, the verification phase changes infinitely often. This is accomplished in Section 3.3.

Altogether, we construct, for every PDL formula  $\varphi$ , an equivalent CFM that is exponential in the size of  $\varphi$  and the number of processes. Recall that by [BL06,BK06], existential monadic second order logic is expressively equivalent to CFMs, and that the set of CFM-languages is not closed under complementation. Since, on the other hand, PDL does not impose any restriction on the use of negation, we obtain the PDL is a proper fragment of existential MSO although this is not obvious.

Furthermore, the model checking problem for PDL can be decided in polynomial space for existentially  $B$ -bounded MSCs, following the lines outlined above. We also show how to model check high-level MSCs (HMSCs) against PDL formulas without knowing the bound  $B$  explicitly. Since the logic  $\text{TLC}^-$  of Peled is a fragment of PDL, we generalize the model checking result from [Pel00] and give an algorithm different from that of Peled.

The final technical section considers an enriched logic iPDL (PDL with intersection) where a node might be described by the intersection of two different paths. This extension seems natural to strengthen the expressive power of the formulas. But adapting a proof technique from colored grids, we show that iPDL is too strong for CFMs, i.e., there is an iPDL formula  $\varphi$  such that no CFM accepts precisely the semantics of  $\varphi$ .

*Outline.* In Section 2 we define message sequence charts, the logic PDL, and communicating finite-state machines. We continue, in Section 3, with several useful constructions for CFMs. In particular, the color language used later on for the translation of forward-path formulas is shown to be acceptable by a CFM. Sections 4 and 5 deal with the translation of PDL formulas into CFMs in the manner described above. The model checking problem is tackled in Section 6 before we

conclude, in Section 7, with the result that PDL with intersection (iPDL) cannot be implemented in terms of CFMs.

## 2 Definitions

The communication framework used in our paper is based on sequential processes that exchange asynchronously messages over point-to-point, error-free FIFO channels. Let  $\mathcal{P}$  be a finite set of process identities which we fix throughout this paper. Furthermore, let  $\text{Ch} = \{(p, q) \in \mathcal{P}^2 \mid p \neq q\}$  denote the set of *channels*. Processes act by either sending a message, that is denoted by  $p!q$  meaning that process  $p$  sends to process  $q$ , or by receiving a message, that is denoted by  $p?q$ , meaning that process  $p$  receives from process  $q$ . For any process  $p \in \mathcal{P}$ , we define a local alphabet (set of event types on  $p$ )  $\Sigma_p = \{p!q, p?q \mid q \in \mathcal{P} \setminus \{p\}\}$ , and we set  $\Sigma = \bigcup_{p \in \mathcal{P}} \Sigma_p$ .

### 2.1 Message sequence charts

Message sequence charts are special labeled partial orders. To define them, we need the following definitions: A  $\Sigma$ -labeled partial order is a triple  $M = (V, \leq, \lambda)$  where  $(V, \leq)$  is a partially ordered set and  $\lambda : V \rightarrow \Sigma$  is a mapping. For  $v \in V$  with  $\lambda(v) = p\theta q$  where  $\theta \in \{!, ?\}$ , let  $P(v) = p$  denote the process that  $v$  is located at. We define two binary relations  $\text{proc}$  and  $\text{msg}$  on  $V$  setting

- $(v, v') \in \text{proc}$  iff  $P(v) = P(v')$ ,  $v < v'$ , and, for any  $u \in V$  with  $P(v) = P(u)$  and  $v \leq u < v'$ , we have  $v = u$ ,
- $(v, v') \in \text{msg}$  iff there is a channel  $(p, q)$  with  $\lambda(v) = p!q$ ,  $\lambda(v') = q?p$ , and

$$|\{u \mid \lambda(u) = p!q, u \leq v\}| = |\{u \mid \lambda(u) = q?p, u \leq v'\}|.$$

**Definition 2.1.** A message sequence chart or MSC for short is a  $\Sigma$ -labeled partial order  $(V, \leq, \lambda)$  such that

- $\leq = (\text{proc} \cup \text{msg})^*$ ,
- $\{u \in V \mid u \leq v\}$  is finite for any  $v \in V$ ,
- $P^{-1}(p) \subseteq V$  is linearly ordered for any  $p \in \mathcal{P}$ , and
- $|\lambda^{-1}(p!q)| = |\lambda^{-1}(q?p)|$  for any  $(p, q) \in \text{Ch}$ .

We refer to the elements of  $V$  as events or nodes.

If  $(V, \leq, \lambda)$  is an MSC, then  $\text{proc}$  and  $\text{msg}$  are even partial and injective functions, so  $v' = \text{proc}(v)$  as well as  $v = \text{proc}^{-1}(v')$  are equivalent notions for  $(v, v') \in \text{proc}$ ;  $\text{msg}(v)$  and  $\text{msg}^{-1}(v')$  are to be understood similarly.

### 2.2 Propositional dynamic logic

*Path expressions*  $\pi$  and *local formulas*  $\alpha$  are defined by simultaneous induction. This induction is described by the following rules

$$\begin{aligned} \pi &::= \text{proc} \mid \text{msg} \mid \{\alpha\} \mid \pi; \pi \mid \pi + \pi \mid \pi^* \\ \alpha &::= \# \mid \sigma \mid \alpha \vee \alpha \mid \neg \alpha \mid \langle \pi \rangle \alpha \mid \langle \pi \rangle^{-1} \alpha \end{aligned}$$

where  $\sigma$  ranges over the alphabet  $\Sigma$ .

Local formulas express properties of single nodes in MSCs. To define the semantics of local formulas, let therefore  $M = (V, \leq, \lambda)$  be an MSC and  $v$  a node from  $M$ . Then we define

$$\begin{aligned} M, v \models \sigma &\iff \lambda(v) = \sigma \quad \text{for } \sigma \in \Sigma \\ M, v \models \alpha_1 \vee \alpha_2 &\iff M, v \models \alpha_1 \text{ or } M, v \models \alpha_2 \\ M, v \models \neg \alpha &\iff M, v \not\models \alpha \end{aligned}$$

The semantics of *forward*-path expressions  $\langle \pi \rangle \alpha$  is given by

$$\begin{aligned}
M, v \models \langle \text{proc} \rangle \alpha &\iff \text{there exists } v' \in V \text{ with } (v, v') \in \text{proc} \text{ and } M, v' \models \alpha \\
M, v \models \langle \text{msg} \rangle \alpha &\iff \text{there exists } v' \in V \text{ with } (v, v') \in \text{msg} \text{ and } M, v' \models \alpha \\
M, v \models \langle \{\alpha\} \rangle \beta &\iff M, v \models \alpha \text{ and } M, v \models \beta \\
M, v \models \langle \pi_1; \pi_2 \rangle \alpha &\iff M, v \models \langle \pi_1 \rangle \langle \pi_2 \rangle \alpha \\
M, v \models \langle \pi_1 + \pi_2 \rangle \alpha &\iff M, v \models \langle \pi_1 \rangle \alpha \vee \langle \pi_2 \rangle \alpha \\
M, v \models \langle \pi^* \rangle \alpha &\iff \text{there exists } n \geq 0 \text{ with } M, v \models (\langle \pi \rangle)^n \alpha
\end{aligned}$$

The semantics of *backward*-path expressions  $\langle \pi \rangle^{-1} \alpha$  is defined similarly by

$$\begin{aligned}
M, v \models \langle \text{proc} \rangle^{-1} \alpha &\iff \text{there exists } v' \in V \text{ with } (v', v) \in \text{proc} \text{ and } M, v' \models \alpha \\
M, v \models \langle \text{msg} \rangle^{-1} \alpha &\iff \text{there exists } v' \in V \text{ with } (v', v) \in \text{msg} \text{ and } M, v' \models \alpha \\
M, v \models \langle \{\alpha\} \rangle^{-1} \beta &\iff M, v \models \alpha \text{ and } M, v \models \beta \\
M, v \models \langle \pi_1; \pi_2 \rangle^{-1} \alpha &\iff M, v \models \langle \pi_1 \rangle^{-1} \langle \pi_2 \rangle^{-1} \alpha \\
M, v \models \langle \pi_1 + \pi_2 \rangle^{-1} \alpha &\iff M, v \models \langle \pi_1 \rangle^{-1} \alpha \vee \langle \pi_2 \rangle^{-1} \alpha \\
M, v \models \langle \pi^* \rangle^{-1} \alpha &\iff \text{there exists } n \geq 0 \text{ with } M, v \models (\langle \pi \rangle^{-1})^n \alpha
\end{aligned}$$

Global properties of an MSC are Boolean combinations of properties of the form “there exists a node satisfying the local formula  $\alpha$ ”. These global properties are expressed by *global formulas*  $\varphi$  whose syntax is given by

$$\varphi ::= E\alpha \mid A\alpha \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

where  $\alpha$  ranges over the set of local formulas. The semantics is defined by

$$\begin{aligned}
M \models E\alpha &\iff \text{there exists a node } v \text{ with } M, v \models \alpha \\
M \models A\alpha &\iff M, v \models \alpha \text{ for all nodes } v \\
M \models \varphi_1 \vee \varphi_2 &\iff M \models \varphi_1 \text{ or } M \models \varphi_2 \\
M \models \varphi_1 \wedge \varphi_2 &\iff M \models \varphi_1 \text{ and } M \models \varphi_2
\end{aligned}$$

Note that our syntax of global formulas does not allow explicit negation. But since we allow existential and universal quantification as well as disjunction and conjunction, the expressible properties are closed under negation.

**Definition 2.2.** *The set of subformulas  $\text{sub}(\alpha)$  of a local formula  $\alpha$  and the set of subformulas  $\text{sub}(\pi)$  of a path expression  $\pi$  are defined by synchronous induction as follows:*

$$\begin{aligned}
\text{sub}(\text{proc}) &= \text{sub}(\text{msg}) = \emptyset \\
\text{sub}(\{\alpha\}) &= \text{sub}(\alpha) \\
\text{sub}(\pi_1; \pi_2) &= \text{sub}(\pi_1 + \pi_2) = \text{sub}(\pi_1) \cup \text{sub}(\pi_2) \\
\text{sub}(\pi^*) &= \text{sub}(\pi)
\end{aligned}$$

and

$$\begin{aligned}
\text{sub}(\sigma) &= \{\sigma\} & \text{sub}(\alpha \vee \beta) &= \{\alpha \vee \beta\} \cup \text{sub}(\alpha) \cup \text{sub}(\beta) \\
\text{sub}(\neg\alpha) &= \{\neg\alpha\} \cup \text{sub}(\alpha) & \text{sub}(\langle \pi \rangle \alpha) &= \{\langle \pi \rangle \alpha\} \cup \text{sub}(\pi) \cup \text{sub}(\alpha) \\
&& \text{sub}(\langle \pi \rangle^{-1} \alpha) &= \{\langle \pi \rangle^{-1} \alpha\} \cup \text{sub}(\pi) \cup \text{sub}(\alpha)
\end{aligned}$$

Thus, in addition to the obvious definition, a subformula of a path expression is any of the local formulas occurring in the path expression as well as any subformula of these local formulas. In particular, contrary to what one might expect, a rather long local formula like  $\varphi = \langle \text{proc}; \{\sigma\}; \text{proc}; \{\sigma\}; \text{proc}; \{\sigma\} \dots \rangle \sigma$  has only two subformulas, namely  $\varphi$  itself and  $\sigma$ . The number of subformulas of  $\alpha$  is bounded by the length of  $\alpha$ , but the length of  $\alpha$  cannot be bounded in terms of the number of subformulas.

Note that a path expression  $\pi$  is a regular expression over  $\{\text{proc}, \text{msg}, \{\alpha_1\}, \dots, \{\alpha_n\}\}$  for some local formulas  $\alpha_i$ . The *size*  $s(\pi)$  of  $\pi$  is defined by  $s(\{\alpha\}) = s(\text{proc}) = s(\text{msg}) = 1$ ,  $s(\pi_1 + \pi_2) = s(\pi_1; \pi_2) = s(\pi_1) + s(\pi_2)$  and  $s(\pi^*) = s(\pi)$  (i.e., it is the number of occurrences of  $\{\alpha\}$ ,  $\text{msg}$ , and  $\text{proc}$  in the regular expression  $\pi$ ). Note that the size of the path expression  $\{\alpha\}$  is 1, independent from the concrete form of the local formula  $\alpha$ .

### 2.3 Communicating finite-state machines

The most natural formalism to describe (asynchronous) communication protocols are *communicating finite-state machines* (CFM for short) [BZ83]. CFMs are a basic model for distributed algorithms based on asynchronous message passing between concurrent processes:

**Definition 2.3.** A communicating finite-state machine (CFM) is a tuple  $\mathcal{A} = (C, n, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  with  $n \in \mathbb{N}$  where

- $C$  is a finite set of message contents or control messages,
- $\mathcal{A}_p = (S_p, \rightarrow_p, \iota_p)$  is a finite labeled transition system over the alphabet  $\Sigma_p \times \{0, 1\}^n \times C$  for any  $p \in \mathcal{P}$  (i.e.,  $\rightarrow_p \subseteq S_p \times (\Sigma_p \times \{0, 1\}^n \times C) \times S_p$ ) with initial state  $\iota_p \in S_p$ ,
- $F \subseteq \prod_{p \in \mathcal{P}} S_p$  is a set of global final states.

Now let  $\mathcal{A}$  be a CFM as above,  $M = (V, \leq, \lambda)$  be an MSC, and  $c : V \rightarrow \{0, 1\}^n$ . A *run of  $\mathcal{A}$  on  $(M, c)$*  is a pair of mappings  $\rho : V \rightarrow \bigcup_{p \in \mathcal{P}} S_p$  and  $\mu : V \rightarrow C$  such that, for any  $v \in V$ ,

1.  $\mu(v) = \mu(\text{msg}(v))$  if  $\text{msg}(v)$  is defined,
2.  $(\rho(\text{proc}^{-1}(v)), \lambda(v), c(v), \mu(v), \rho(v)) \in \rightarrow_{P(v)}$  if  $\text{proc}^{-1}(v)$  is defined, and  $(\iota_p, \lambda(v), c(v), \mu(v), \rho(v)) \in \rightarrow_{P(v)}$  otherwise.

In order to define when the run  $(\rho, \mu)$  is accepting, we will use Büchi-conditions on each process. For this, one is usually interested in the set of states that appear infinitely often. But since, even in an infinite MSC, some of the processes may execute only finitely many events, the set of states appearing infinitely often is here generalized to the set of states that appear *cofinally*: Let  $\text{cofin}_\rho(p) = \{s \in S_p \mid \forall v \in V_p \exists v' \in V_p : v \leq v' \wedge \rho(v') = s\}$  where  $V_p = P^{-1}(p)$ . Then the run  $(\rho, \mu)$  is *accepting* if there is some  $(s_p)_{p \in \mathcal{P}} \in F$  such that  $s_p \in \text{cofin}_\rho(p)$  for all  $p \in \mathcal{P}$ . The *language of  $\mathcal{A}$*  is the set  $L(\mathcal{A})$  of all pairs  $(M, c)$  that admit an accepting run.

We will also consider CFMs that run on MSCs without coloring. Then, we deal with structures  $(C, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  where  $\mathcal{A}_p$  is a finite labeled transition system over  $\Sigma_p \times C$  for any  $p \in \mathcal{P}$ .

## 3 Construction of CFMs

In this section, we present some particular CFMs and constructions of CFMs. These results will be used in later sections.

### 3.1 Intersection

Here, we show that the intersection of languages accepted by CFMs can again be accepted by a CFM. Since the acceptance by a CFM is defined in terms of a Büchi-condition, we can adopt the flag construction [Cho74] from the theory of word automata with Büchi-acceptance condition (cf. proof of Lemma 1.2 in [Tho90]). The additional problem we face here is the interplay between different processes. In order to keep the CFM for the intersection small, we introduce the following notion.

**Definition 3.1.** Let  $F \subseteq \prod_{p \in \mathcal{P}} S_p$ . The index of  $F$  is the least number  $n$  such that there are sets  $F_p^i \subseteq S_p$  for  $p \in \mathcal{P}$  and  $1 \leq i \leq n$  with  $F = \bigcup_{1 \leq i \leq n} \prod_{p \in \mathcal{P}} F_p^i$ .  
The index of a CFM is the index of its set of accepting states.

Clearly, the index of a CFM is bounded by  $s^{|\mathcal{P}|}$  where  $s$  is the maximal size of a set of local states  $S_p$ . To see that the index can be quite large, let  $n \in \mathbb{N}$ ,  $\mathcal{P} = S_p = [n]$  for all  $p \in \mathcal{P}$  (where we let  $[n] = \{1, \dots, n\}$ ). Furthermore, let  $F$  be the set of all tuples  $(s_p)_{p \in \mathcal{P}}$  such that  $\{s_p \mid p \in \mathcal{P}\} = [n]$ , i.e., the set of surjections from  $[n]$  onto  $[n]$ . Hence  $F$  contains  $n!$  many elements. Any two of them differ in at least two positions. Hence the index of  $F$  equals its size and is exponential in  $n$  and therefore in  $|\mathcal{P}|$ .

**Lemma 3.2.** For  $1 \leq i \leq m$ , let  $\mathcal{A}^i = (C^i, n, (S_p^i, \rightarrow_p^i, \iota_p^i)_{p \in \mathcal{P}}, F^i)$  be CFMs of index 1. Then there exists a CFM  $\mathcal{A}$  of index 1 that accepts  $(M, c)$  with  $M$  an MSC and  $c : V \rightarrow \{0, 1\}^n$  iff it is accepted by  $\mathcal{A}^i$  for all  $i \in [m]$ .

The set of messages of  $\mathcal{A}$  is  $\prod_{i \in [m]} C^i$  and the set of local states of process  $p$  is  $\{0, 1, \dots, m\} \times \prod_{i \in [m]} S_p^i$ .

*Proof.* Since  $F^i$  has index 1, there exist sets  $F_p^i \subseteq S_p^i$  with  $F^i = \prod_{p \in \mathcal{P}} F_p^i$ .

The idea of the proof is that  $\mathcal{A}$  will simulate all the machines  $\mathcal{A}^i$  in parallel. In addition, it checks that, for each  $p \in \mathcal{P}$  and  $i \in [m]$ , some state from  $F_p^i$  is assumed cofinally (i.e., infinitely often or, if process  $p$  executes only finitely many events, at the last event from  $p$ ). Formally, we set  $\iota_p = \begin{cases} (m, \iota_p^1, \dots, \iota_p^m) & \text{if } (\iota_p^1, \dots, \iota_p^m) \in \prod_{i \in [m]} F_p^i \text{ and } F = \prod_{p \in \mathcal{P}} (\{m\} \times \prod_{i \in [m]} S_p^i). \\ (0, \iota_p^1, \dots, \iota_p^m) & \text{otherwise} \end{cases}$ .

Furthermore,  $(a, (s_i)_{i \in [m]}) \xrightarrow{\sigma, c, (b_i)_{i \in [m]}}_p (a', (s'_i)_{i \in [m]})$  with  $b_i \in C^i$  is a transition of  $\mathcal{A}$  iff

1.  $s_i \xrightarrow{\sigma, c, b_i}_p s'_i$  is a transition of  $\mathcal{A}^i$  for all  $i \in [m]$
2.  $a' = \begin{cases} m & \text{if } s_p^i \in F_p^i \text{ for all } i \in [m] \\ 0 & \text{if } a = m \text{ and } s_p^i \notin F_p^i \text{ for some } i \in [m] \\ a + 1 & \text{if } a < m, s_p^{a+1} \in F_p^{a+1} \text{ and } s_p^i \notin F_p^i \text{ for some } i \in [m] \\ a & \text{otherwise.} \end{cases}$

Recall the classical flag construction for  $\omega$ -word automata. There, the value of the counter  $a$  indicates that the composite machine waits for an accepting state of the simulated machine  $a + 1$ ; a value  $m$  indicates that all simulated machines went through some accepting states. Here, we do the same. But, in addition, if all component states of the composite machine are accepting, then we set the counter value directly to  $m$ . This is useful when process  $p$  executes only finitely many events. Then, at its final event  $v$ , all the component machines have to be in some accepting state. For processes executing infinitely many events, this is of no importance.  $\square$

**Proposition 3.3.** For  $i \in [m]$ , let  $\mathcal{A}^i = (C^i, n, (S_p^i, \rightarrow_p^i, \iota_p^i)_{p \in \mathcal{P}}, F^i)$  be a CFM of index  $\ell_i$ . Then there exists a CFM  $\mathcal{A}$  that accepts  $(M, c)$  with  $M$  an MSC and  $c : V \rightarrow \{0, 1\}^n$  iff it is accepted by  $\mathcal{A}^i$  for all  $i \in [m]$ .

The set of messages of  $\mathcal{A}$  is  $\prod_{i \in [m]} C^i$ . Moreover, the set of local states of process  $p$  is  $\iota_p \cup (\{0, 1, 2, \dots, m\} \times \prod_{i \in [m]} S_p^i \times \prod_{i \in [m]} [\ell_i])$ .

*Proof.* Since the index of  $\mathcal{A}^i$  is  $\ell_i$ , its language is the union of languages  $L_1^i, \dots, L_{\ell_i}^i$  that can each be accepted by a CFM of index 1. The language in question is therefore  $\bigcup_{j \in \prod_{i \in [m]} [\ell_i]} \bigcap_{i \in [m]} L_{j_i}^i$ . By Lemma 3.2, the intersection  $\bigcap_{i \in [m]} L_{j_i}^i$  can be accepted by a CFM of index 1 with set of local states  $\{0, 1, 2, \dots, m\} \times \prod_{i \in [m]} S_p^i \times \{j\}$ . The disjoint union of all these CFMs (together with new local initial states) accepts the language in question; its set of states equals  $\iota_p \cup (\{0, 1, 2, \dots, m\} \times \prod_{i \in [m]} S_p^i \times \prod_{i \in [m]} [\ell_i])$  as claimed.  $\square$

### 3.2 Infinitely running processes

For an MSC  $M = (V, \leq, \lambda)$ , let  $\text{Inf}(M) \subseteq \text{Ch}$  denote the set of those channels  $(p, q)$  that are used infinitely often, i.e.,  $\text{Inf}(M) = \{(p, q) \in \text{Ch} \mid \lambda^{-1}(p!q) \text{ is infinite}\}$ . From a set  $I \subseteq \text{Ch}$ , we want to construct a CFM of index 1 that checks whether  $\text{Inf}(M) = I$ .

**Lemma 3.4.** *Let  $I \subseteq \text{Ch}$ . There exists a CFM  $\mathcal{A}_1$  of index 1 with three local states per process and one message that accepts an MSC  $M$  iff  $\text{Inf}(M) \subseteq I$ .*

*Proof.* The sets of local states are given by  $S_p = \{0, 1, 2\}$  for any  $p \in \mathcal{P}$ , the state 0 is locally initial. The only control message is 1. Then we set  $a \xrightarrow{\sigma, 1} b$  iff  $(a = b \text{ and } \sigma \text{ uses a channel from } I)$  or  $(a < b \text{ and } \sigma \text{ does not use a channel from } I)$  or  $a = b = 1$ . Then state 0 indicates that no channel of  $\text{Ch} \setminus I$  has been used, 1 indicates that some channel from  $\text{Ch} \setminus I$  has been used and that some channel will be used, and 2 denotes that some channel from  $\text{Ch} \setminus I$  has been used but none will ever be used in the future. Hence, process  $p$  uses the channels from  $\text{Ch} \setminus I$  only finitely often iff it can visit 0 or 2 cofinally. Setting  $F = \prod_{p \in \mathcal{P}} \{0, 2\}$  therefore finishes the construction of the desired CFM.  $\square$

**Lemma 3.5.** *Let  $I \subseteq \text{Ch}$ . There exists a CFM  $\mathcal{B}_1$  of index 1 with  $4^{|\mathcal{P}|}$  local states per process and one message that accepts an MSC  $M$  iff  $I \subseteq \text{Inf}(M)$ .*

*Proof.* For  $p \in \mathcal{P}$  let  $S_p = \{0, 1\}^{\Sigma_p}$  and set  $C = \{1\}$ . The locally initial state  $\iota_p \in S_p$  maps all  $\tau \in \Sigma_p$  to 0. Then we set  $g \xrightarrow{\sigma, 1} g'$  for  $g, g' \in S_p$  and  $\sigma \in \Sigma_p$  iff  $g'(\tau) = \begin{cases} g(\tau) & \text{if } \tau \neq \sigma \\ 1 - g(\tau) & \text{otherwise} \end{cases}$  for all  $\tau \in \Sigma_p$ . Thus, the local process  $p$  counts modulo 2 the number of occurrences of any local action. The channel  $(p, q)$  is used infinitely often iff the following two properties hold:

- Process  $p$  visits a state  $g_p$  with  $g_p(p!q) = 0$  cofinally.
- Process  $q$  visits a state  $g_q$  with  $g_q(q?p) = 1$  cofinally.

Hence a global state  $(g_p)_{p \in \mathcal{P}}$  is final (i.e., belongs to  $F$ ) iff, for any  $(p, q) \in I$ , we have  $g_p(p!q) = 0$  and  $g_q(q?p) = 1$ .  $\square$

**Proposition 3.6.** *Let  $I \subseteq \text{Ch}$ . There exists a CFM  $\mathcal{B}$  of index 1 with  $3 \cdot 3 \cdot 4^{|\mathcal{P}|}$  local states per process and one message that accepts an MSC  $M$  iff  $I = \text{Inf}(M)$ .*

*Proof.* Follows immediately from Lemmas 3.4, 3.5, and 3.2.  $\square$

### 3.3 The color language

For later use, and in order to become acquainted with the computational power of CFMs, in this section we build a CFM that accepts “black/white colored” MSCs. The aim is that such a coloring is acceptable if any infinite path in the MSC has infinitely many color changes (cf. Cor. 3.9).

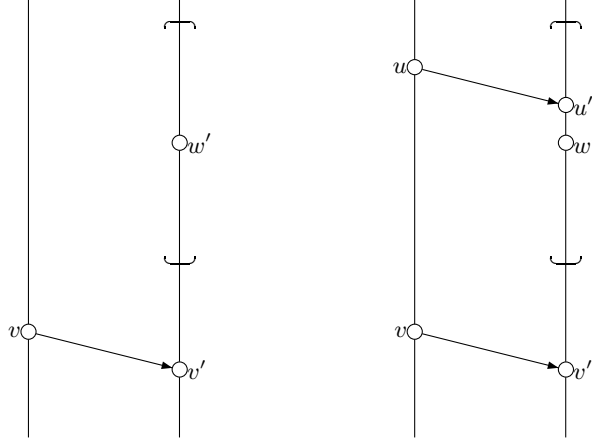
Let  $M$  be an MSC and  $c : V \rightarrow \{0, 1\}$ . On  $V$ , we define an equivalence relation  $\sim$  setting  $u \sim w$  iff  $P(u) = P(w)$  and, for all  $v \in V$  with  $(u \leq v \leq w \text{ or } w \leq v \leq u)$  and  $P(u) = P(v)$ , we have  $c(u) = c(v) = c(w)$  (i.e., a  $\sim$ -equivalence class is a maximal monochromatic interval on a process line).

Let  $\text{Col}$  be the set of all pairs  $(M, c)$  with  $c : V \rightarrow \{0, 1\}$  such that the following hold

- (1) if  $v$  is minimal on its process, then  $c(v) = 1$
- (2) if  $(v, v') \in \text{msg}$  and  $w' \leq v'$  with  $P(w') = P(v')$ , then there exists  $(u, u') \in \text{msg}$  with  $\lambda(u') = \lambda(v')$ ,  $c(u) = c(u')$ , and  $u' \sim w'$
- (3) any equivalence class of  $\sim$  is finite.

Figure 1 visualizes the second condition, on the left, we have the precondition while the right diagram indicates the conclusion. More precisely, in the precondition, we have a message  $(v, v') \in \text{msg}$  from process  $p$  to process  $q$  and some node  $w'$  preceding  $v'$  on the same process. Note that equivalence classes of  $\sim$  are intervals on process lines. The borders of the equivalence class containing  $w'$  are indicated. Then, by the conclusion, there is a message  $(u, u') \in \text{msg}$  from  $p$  to  $q$  such that  $u'$  belongs to the indicated equivalence class of  $\sim$  (that also contains  $w'$ ) and the colors of  $u$  and  $u'$  are the same (which is not indicated).

In general, there can be messages  $(u, u') \in \text{msg}$  such that the colors of  $u$  and  $u'$  are different, i.e.,  $c(u) \neq c(u')$  (Corollary 3.9 will show that there are many such messages). The second condition ensures that there are also “many” messages where the send and the receive event carry the same color.



**Fig. 1.** The second condition

**Proposition 3.7.** *There exists a CFM  $\mathcal{A}_{\text{Col}}$  that accepts the set Col. The CFM  $\mathcal{A}_{\text{Col}}$  has two messages and its number of local states is in  $2^{O(|\mathcal{P}|)}$ .*

*Proof.* Since the language Col consists of pairs  $(M, c)$ , any process  $p$  of an accepting MPA executes a sequence of events from  $((\Sigma_p \times \{0, 1\}) \times \{0, 1\})^\infty$  (with  $\{0, 1\}^\infty$  the set of finite and infinite words over  $\{0, 1\}$ ) where  $((\sigma, a), b)$  stands for an  $(\sigma, a)$ -labeled event that sends or receives  $b$ . Our automaton  $\mathcal{A}_{\text{Col}}$  will always send the current value of the mapping  $c$ , i.e., the set of control messages is  $\{0, 1\}$  and we will only execute events from

$$\Gamma_p = \{((p!q, a), a) \mid q \in \mathcal{P}, a \in \{0, 1\}\} \cup \{((p?q, a), b) \mid q \in \mathcal{P}, a, b \in \{0, 1\}\}.$$

Having this in mind, consider for  $p \in \mathcal{P}$ ,  $B \subseteq \mathcal{P}$ , and  $i \in \{0, 1\}$  the language  $L_{B,i}^p \subseteq \Gamma_p^*$  with  $w \in L_{B,i}^p$  iff

- if  $((p!q, a), a)$  occurs in  $w$ , then  $a = i$ ,
- if  $((p?q, a), b)$  occurs in  $w$ , then  $a = i$  and  $q \in B$ ,
- for all  $q \in B$ , the letter  $((p?q, i), i)$  occurs in  $w$ .

Then  $L_{B,i}^p$  is regular and can be accepted by a finite deterministic automaton  $\mathcal{B}_{B,i}^p$  with  $2^{|B|}$  many states. We build the  $p$ -component  $\mathcal{A}_p$  of  $\mathcal{A}_{\text{Col}}$  from the disjoint union of all these automata  $\mathcal{B}_{B,i}^p$  – it therefore has

$$\sum_{B \subseteq \mathcal{P}} 2 \cdot 2^{|B|} \leq 2 \cdot 4^{|\mathcal{P}|}$$

many states. More precisely,  $\mathcal{A}_p$  is obtained from this disjoint union by adding  $\varepsilon$ -transition from any accepting state of  $\mathcal{A}_{B,i}^p$  to any initial state of  $\mathcal{A}_{C,j}^p$  iff  $B \supseteq C$  and  $i \neq j$ . The initial states of  $\mathcal{A}_p$  are the initial states of  $\mathcal{B}_{B,1}^p$ . A finite run is accepting if it ends in some final state of one of the automata  $\mathcal{B}_{B,i}^p$ , an infinite run is accepting if it takes infinitely many  $\varepsilon$ -transitions.

Note that a word  $((p\theta_n q_n, a_n), b_n)_{0 \leq n < N} \in \Gamma_p^\infty$  is accepted by  $\mathcal{A}_p$  iff

- $a_0 = 1$
- if  $\theta_n = !$ , then  $a_n = b_n$
- if  $\theta_n = ?$  and  $m \leq n$ , then there exists  $k \in \mathbb{N}$  with  $p\theta_k q_k = p?q_n$ ,  $a_k = b_k$ , and, for all  $\ell$  in between  $m$  and  $k$ , we have  $a_m = a_\ell = a_k$ .

Hence the MPA consisting of these components accepts the language Col. □

The *index*  $\text{ind}(v)$  of a node  $v$  is the maximal number of mutually non-equivalent nodes from  $V_p$  below  $v$ . Note that  $c(v) = \text{ind}(v) \bmod 2$  for all nodes  $v$  if the pair  $(M, c)$  satisfies (1) in the definition of the language Col.

**Lemma 3.8.** *Let  $(M, c) \in \text{Col}$ . Then, for any  $(v, v') \in \text{msg}$  with  $\text{ind}(v) < \text{ind}(v')$ , we have  $c(v) \neq c(v')$ .*

*Proof.* Suppose there is  $(v, v') \in \text{msg}$  with  $\text{ind}(v) < \text{ind}(v')$  but  $c(v) = c(v')$ . Since any element of  $M$  dominates a finite set, we can assume  $v'$  to be minimal with this problem. If  $\text{ind}(v) + 1 = \text{ind}(v')$ , we are done since  $c(v) = \text{ind}(v) \bmod 2 \neq (\text{ind}(v) + 1) \bmod 2 = c(v')$ . So let  $\text{ind}(v) + 1 < \text{ind}(v')$ . Since  $(M, c) \in \text{Col}$  and  $\text{ind}(v') - 1 > \text{ind}(v) \geq 1$ , there exists  $(u, u') \in \text{msg}$  with  $\lambda(u') = \lambda(v')$ ,  $c(u) = c(u')$ , and  $\text{ind}(u') = \text{ind}(v') - 1$ . In particular,  $u' < v'$  and therefore  $u < v$ . But then  $\text{ind}(u) \leq \text{ind}(v)$ . Now we have  $\text{ind}(u) \leq \text{ind}(v) < \text{ind}(v') - 1 = \text{ind}(u')$ , i.e.,  $u' < v'$  is another counterexample to the statement of the lemma. But this contradicts the choice of  $v'$ . □

**Corollary 3.9.** *Let  $(M, c) \in \text{Col}$  and let  $(v_1, v_2, \dots)$  be some infinite path in  $M$ . Then there exist infinitely many  $i \in \mathbb{N}$  with  $c(v_i) \neq c(v_{i+1})$ .*

*Proof.* Since  $\text{ind}^{-1}(n)$  is finite for any  $n \in \mathbb{N}$ , there are infinitely many  $i \in \mathbb{N}$  with  $\text{ind}(v_i) < \text{ind}(v_{i+1})$ . If  $(v_i, v_{i+1}) \in \text{proc}$ , then  $\text{ind}(v_{i+1}) = \text{ind}(v_i) + 1$  and therefore  $c(v_i) \neq c(v_{i+1})$ . If, in the other case,  $(v_i, v_{i+1}) \in \text{msg}$ , then by Lemma 3.8, we get  $c(v_i) \neq c(v_{i+1})$ . □

## 4 Translation of local formulas

Let  $\alpha$  be a local formula of PDL. We will construct a “small” CFM that accepts a pair  $(M, c)$  with  $M$  an MSC and  $c : V \rightarrow \{0, 1\}$  iff  $c$  is the characteristic function of the set of positions satisfying  $\alpha$ , i.e.,

$$c(v) = \begin{cases} 1 & \text{if } M, v \models \alpha \\ 0 & \text{otherwise.} \end{cases}$$

To obtain this CFM, we will first construct another CFM that accepts  $(M, (c_\beta)_{\beta \in \text{sub}(\alpha)})$  iff, for all positions  $v \in V$  and all subformulas  $\beta$  of  $\alpha$ , we have  $M, v \models \beta$  iff  $c_\beta(v) = 1$ . This CFM will consist of several CFMs running in conjunction, one for each subformula. For instance, if  $\sigma \in \Sigma$  and  $\delta = \beta \vee \gamma$  are subformulas of  $\alpha$ , then we will have sub-CFMs that check whether, for any position  $v$ , we have  $c_\sigma(v) = 1$  iff  $\lambda(v) = \sigma$  and  $c_\delta(v) = c_\beta(v) \vee c_\gamma(v)$ , resp. We first define these sub-CFMs for subformulas of the form  $\sigma$ ,  $\beta \vee \gamma$ , and  $\neg\beta$ .

*Example 4.1.* For  $\sigma \in \Sigma$ , we define the CFM  $\mathcal{A}_\sigma = (\{m\}, 1, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  as follows: For  $p \in \mathcal{P}$ , let  $S_p = \{\iota_p\}$  and  $(\iota_p, \tau, b, m, \iota_p) \in \rightarrow_p$  iff

- $\tau = \sigma$  and  $b = 1$  or
- $\tau \neq \sigma$  and  $b = 0$ .

Furthermore,  $F = \{(\iota_p)_{p \in \mathcal{P}}\}$ . Then it is easily checked that  $(M, c)$  is accepted by  $\mathcal{A}_\sigma$  iff

$$\forall v \in V : \lambda(v) = \sigma \iff c(v) = 1.$$

*Example 4.2.* Next we define a CFM  $\mathcal{A}_\vee = (\{m\}, 3, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$ : For  $p \in \mathcal{P}$ , let  $S_p = \{\iota_p\}$  and  $(\iota_p, \tau, (b_1, b_2, b_3), m, \iota_p) \in \rightarrow_p$  iff  $b_3 = b_1 \vee b_2$ . Furthermore,  $F = \{(\iota_p)_{p \in \mathcal{P}}\}$ . Then it is easily checked that  $(M, c)$  is accepted by  $\mathcal{A}_\vee$  iff

$$\forall v \in V : c_3(v) = c_1(v) \vee c_2(v).$$

The CFM  $\mathcal{A}_\wedge$  is defined similarly.

*Example 4.3.* Next we define a CFM  $\mathcal{A}_E = (\{m\}, 1, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$ : For  $p \in \mathcal{P}$ , let  $S_p = \{\iota_p, s_p\}$ ,  $(\iota_p, \tau, 0, m, \iota_p) \in \rightarrow_p$ ,  $(\iota_p, \tau, 1, m, s_p) \in \rightarrow_p$ , and  $(s_p, \tau, b, m, s_p) \in \rightarrow_p$  for all  $\tau \in \Sigma_p$  and  $b \in \{0, 1\}$ . Furthermore,  $F$  is the set of tuples  $(f_p)_{p \in \mathcal{P}}$  that contain at least one occurrence of  $s_p$ . Hence the index of this CFM is the number of processes  $|\mathcal{P}|$ .

Then it is easily checked that  $(M, c)$  is accepted by  $\mathcal{A}_E$  iff there exists a node  $v$  with  $c(v) = 1$ .

The CFM  $\mathcal{A}_A = (\{m\}, 1, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  has again just one local state per process (and is therefore of index 1): For  $p \in \mathcal{P}$ , let  $S_p = \{\iota_p\}$ ,  $(\iota_p, \tau, b, m, \iota_p) \in \rightarrow_p$  iff  $b = 1$  for any  $\tau \in \Sigma_p$ , and  $F = \{(\iota_p)_{p \in \mathcal{P}}\}$ .

Then it is easily checked that  $(M, c)$  admits a run and is therefore accepted by  $\mathcal{A}_A$  iff  $c(v) = 1$  for all nodes  $v$ .

#### 4.1 The backward-path automaton

Let  $\pi$  be a path expression, i.e., a regular expression over the alphabet  $\{\text{proc}, \text{msg}, \{\alpha_1\}, \dots, \{\alpha_n\}\}$ . Replacing  $\{\alpha_i\}$  by  $i$ , we obtain a regular expression over the alphabet  $\Gamma = \{\text{proc}, \text{msg}, 1, 2, \dots, n\}$ . Let  $L_\pi \subseteq \Gamma^*$  be the language of this regular expression.

A word over  $\Gamma$  together with a node from an MSC describes a path starting in that node that walks *backwards*. The letters *proc* and *msg* denote the direction of the path, the letters  $i$  denote requirements about the node currently visited (namely, that  $\alpha_i$  shall hold). This idea motivates the following definition.

**Definition 4.4.** For an MSC  $M$ , functions  $c_1, \dots, c_n : V \rightarrow \{0, 1\}$ , a node  $v \in V$  and a word  $W \in \Gamma^*$ , we define inductively  $(M, c_1, \dots, c_n), v \models^{-1} W$ :

$$\begin{aligned} (M, c_1, \dots, c_n), v &\models^{-1} \varepsilon \\ (M, c_1, \dots, c_n), v &\models^{-1} \text{proc } W \iff \text{there exists } v' = \text{proc}^{-1}(v) \text{ with } (M, c_1, \dots, c_n), v' \models^{-1} W \\ (M, c_1, \dots, c_n), v &\models^{-1} \text{msg } W \iff \text{there exists } v' = \text{msg}^{-1}(v) \text{ with } (M, c_1, \dots, c_n), v' \models^{-1} W \\ (M, c_1, \dots, c_n), v &\models^{-1} i W \iff c_i(v) = 1 \text{ and } (M, c_1, \dots, c_n), v \models^{-1} W \end{aligned}$$

We easily verify that  $M, v \models \langle \pi \rangle^{-1} \#$  iff there exists  $W \in L_\pi$  such that  $M, v \models^{-1} W$ .

Let  $\mathcal{C} = (Q, \iota, T, G)$  be a finite automaton over  $\Gamma$  recognizing  $L_\pi^{rev}$ , the reversal of the language  $L_\pi$ . Note that we can assume  $|Q| \in O(s(\pi))$ . For  $q \in Q$  and  $W \in \Gamma^*$ , we write  $q.W \subseteq Q$  for the set of states that can be reached from  $q$  reading the word  $W$ . Furthermore,  $P.L = \bigcup_{p \in P, W \in L} p.W$  for  $P \subseteq Q$  and  $L \subseteq \Gamma^*$ .

**Lemma 4.5.** *There exists a CFM  $\mathcal{A}$  with sets of local states  $2^Q$  and set of messages  $2^Q$  such that, for any run on  $(M, c_1, \dots, c_n)$  and any  $v \in V$ , we have  $\rho(v) = \{q \in Q \mid \exists W \in \Gamma^* : q \in \iota.W \text{ and } M, v \models^{-1} W^{rev}\}$ .*

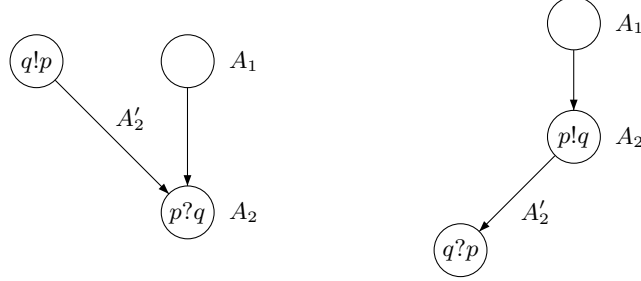
*Proof.* To define the set of transitions, let  $A_1, A_2, A'_2 \subseteq 2^Q$ . Furthermore, let  $a = (\sigma, b_1, \dots, b_n, b) \in \Sigma_p \times \{0, 1\}^{n+1}$  and  $N = \{i \in [n] \mid b_i = 1\}$ . Then we set

$$A_1 \xrightarrow{a, A'_2}_p A_2$$

iff the following conditions hold

- (1) If  $\sigma$  is a send action, then  $A_2 = A'_2 = \iota.N^* \cup A_1.\text{proc}.N^*$
- (2) If  $\sigma$  is a receive action, then  $A_2 = \iota.N^* \cup A_1.\text{proc}.N^* \cup A'_2.\text{msg}.N^*$ .

Here,  $A_1$  is the local state assumed before the execution of the (labeled) action  $a$ ,  $A_2$  is the local state assumed afterwards, and  $A'_2$  is the message involved in this transition. Depending on whether  $a$  is a receive or a send action, the message is consumed by  $a$  or emitted by  $a$ . These two situations are visualized in Figure 2.



**Fig. 2.** Transitions of the CFM

The local initial state is  $\emptyset$  for any  $p \in \mathcal{P}$  and any tuple of local states is accepting. Now let  $(\rho, \mu)$  be a run of this CFM on  $(M, c_1, \dots, c_n)$ .

For  $v \in V$  set  $N_v = \{i \in [n] \mid c_i(v) = 1\}$  and  $M(v) = \{W \in \Gamma^* \mid (M, c_1, \dots, c_n), v \models^{-1} W\}$ . Then it is easily verified that

$$M(v) = N_v^* \cup \underbrace{N_v^* \text{proc } M(\text{proc}^{-1}(v))}_{\text{if } \text{proc}^{-1}(v) \text{ is defined}} \cup \underbrace{N_v^* \text{msg } M(\text{msg}^{-1}(v))}_{\text{if } \text{msg}^{-1}(v) \text{ is defined}} .$$

On the other hand,

$$\rho(v) = \iota.N_v^* \cup \underbrace{\rho(\text{proc}^{-1}(v)).\text{proc}.N_v^*}_{\text{if } \text{proc}^{-1}(v) \text{ is defined}} \cup \underbrace{\rho(\text{msg}^{-1}(v)).\text{msg}.N_v^*}_{\text{if } \text{msg}^{-1}(v) \text{ is defined}} .$$

Hence, by induction on the partial order  $(V, \leq)$ , we have  $q \in \rho(v)$  iff there exists  $W^{rev} \in M(v)$  with  $q \in \iota.W$ .  $\square$

**Theorem 4.6.** *Let  $\langle \pi \rangle^{-1} \alpha$  be a local formula such that  $\pi$  is a regular expression over the alphabet  $\{\text{proc}, \text{msg}, \{\alpha_1\}, \dots, \{\alpha_n\}\}$ . Then there exists a CFM  $\mathcal{A}_{\langle \pi \rangle^{-1} \alpha}$  with the following property: Let  $M$  be an MSC and let  $c_i : V \rightarrow \{0, 1\}$  be the characteristic function of the set of positions satisfying  $\alpha_i$  (for all  $i \in [n+1]$ ) where  $\alpha_{n+1} = \alpha$ . Then  $(M, c_1, \dots, c_n, c_{n+1}, c)$  is accepted iff  $c$  is the characteristic function of the set of positions satisfying  $\langle \pi \rangle^{-1} \alpha$ .*

*The CFM we construct has  $2^{O(s(\pi))}$  local states per process,  $2^{O(s(\pi))}$  many control messages, and any tuple of local states is accepting (in particular, the CFM has index 1).*

*Proof.* Again, since  $M, v \models \langle \pi \rangle^{-1} \alpha$  iff  $M, v \models \langle \pi; \{\alpha\} \rangle^{-1} \#$ , we will assume  $\alpha = \#$ . The CFM  $\mathcal{A}_{\langle \pi \rangle^{-1} \alpha}$  simulates the run  $(\rho, \mu)$  of the CFM  $\mathcal{A}$  from Lemma 4.5 and verifies that  $c(v) = 1$  iff  $\rho(v) \cap G \neq \emptyset$  for all nodes  $v \in V$ . Then we have

$$\begin{aligned} c(v) = 1 &\iff \rho(v) \cap G \neq \emptyset \\ &\iff \exists W \in \Gamma^* : \iota.W \cap G \neq \emptyset \text{ and } M, v \models^{-1} W^{rev} \\ &\iff \exists W \in L(\mathcal{C}) = L_\pi^{rev} : M, v \models^{-1} W^{rev} \\ &\iff \exists W \in L_\pi : M, v \models^{-1} W \\ &\iff M, v \models \langle \pi \rangle^{-1} \alpha \end{aligned}$$

$\square$

## 4.2 The forward-path automaton

We now turn to a similar CFM corresponding to subformulas of the form  $\langle \pi \rangle \#$ . We will prove the following analog to Theorem 4.6. This proof will, however, be substantially more difficult.

**Theorem 4.7.** *Let  $I \subseteq \text{Ch}$  and let  $\langle \pi \rangle \alpha$  be a local formula such that  $\pi$  is a regular expression over the alphabet  $\{\text{proc}, \text{msg}, \{\alpha_1\}, \dots, \{\alpha_n\}\}$ . Then there exists a CFM  $\mathcal{A}_{\langle \pi \rangle \alpha}$  of index 1 with the following property: Let  $M$  be an MSC with  $\text{Inf}(M) = I$  and let  $c_i : V \rightarrow \{0, 1\}$  be the characteristic function of the set of positions satisfying  $\alpha_i$  (for all  $i \in [n+1]$ ) where  $\alpha_{n+1} = \alpha$ . Then  $(M, c_1, \dots, c_n, c_{n+1}, c)$  is accepted iff  $c$  is the characteristic function of the set of positions satisfying  $\langle \pi \rangle \alpha$ . The CFM we construct has  $2^{O(s(\pi)+|\mathcal{P}|)}$  local states per process and  $2^{O(s(\pi))}$  many control messages.*

The rest of this section is devoted to the proof of this theorem. Since  $M, v \models \langle \pi \rangle \alpha$  iff  $M, v \models \langle \pi; \{\alpha\} \rangle \#$ , we will assume  $\alpha = \#$ , i.e.,  $\alpha$  holds true for any node of any MSC.

Let  $\Gamma = \{\text{proc}, \text{msg}, 1, 2, \dots, n\}$ . If, in the regular expression  $\pi$ , we replace any occurrence of  $\{\alpha_i\}$  by  $i$ , we obtain a regular expression over the alphabet  $\Gamma$ . Let  $L_\pi \subseteq \Gamma^*$  be the language denoted by this regular expression. Then there is a finite automaton  $\mathcal{C} = (Q, \iota, T, G)$  over  $\Gamma$  with set of states  $Q$ , initial state  $\iota$ , set of transitions  $T$ , and set of final states  $G$ . Note that  $|Q| \in O(s(\pi))$ .

A word over  $\Gamma$  together with a node from an MSC describe a path starting in that node walking forwards. The following is therefore the forward-version of Def. 4.4.

**Definition 4.8.** *For an MSC  $M$ , functions  $c_1, \dots, c_n : V \rightarrow \{0, 1\}$ , a node  $v \in V$  and a word  $W \in \Gamma^*$ , we define inductively  $(M, c_1, \dots, c_n), v \models W$ :*

$$\begin{aligned} (M, c_1, \dots, c_n), v &\models \varepsilon \\ (M, c_1, \dots, c_n), v &\models \text{proc } W \iff \text{there exists } v' = \text{proc}(v) \text{ with } (M, c_1, \dots, c_n), v' \models W \\ (M, c_1, \dots, c_n), v &\models \text{msg } W \iff \text{there exists } v' = \text{msg}(v) \text{ with } (M, c_1, \dots, c_n), v' \models W \\ (M, c_1, \dots, c_n), v &\models iW \iff c_i(v) = 1 \text{ and } (M, c_1, \dots, c_n), v \models W \end{aligned}$$

Now the following is immediate

**Lemma 4.9.** *Let  $M$  be an MSC and, for  $i \in [n]$ , let  $c_i : V \rightarrow \{0, 1\}$  be the characteristic function of the set of positions satisfying  $\alpha_i$ . Then  $M, v \models \langle \pi \rangle \#$  iff there exists  $W \in L_\pi$  such that  $M, v \models W$ .*

Thus, in order to prove Theorem 4.7, it suffices to construct a CFM that accepts  $(M, c_1, \dots, c_n, c)$  iff

$$\begin{aligned} \forall v \in V : c(v) = 0 &\implies \forall W \in L_\pi : (M, c_1, \dots, c_n), v \not\models W \\ \wedge \forall v \in V : c(v) = 1 &\implies \exists W \in L_\pi : (M, c_1, \dots, c_n), v \models W \end{aligned}$$

Since the class of languages accepted by CFMs is closed under intersection, we can handle the two implications separately in the following two subsections.

**Any 0 is justified** We construct a CFM that accepts  $(M, c_1, \dots, c_n, c)$  iff, for any  $v \in V$  with  $c(v) = 0$ , there does not exist  $W \in L_\pi$  with  $(M, c_1, \dots, c_n, c), v \models W$ . The basic idea is rather simple: whenever the CFM encounters a node  $v$  with  $c(v) = 0$ , it will start the automaton  $\mathcal{C}$  (that accepts  $L_\pi$ ) and check that it cannot reach an accepting state whatever path we choose starting in  $v$ . Since the CFM has to verify more than one 0, the set of local states  $S_p$  equals  $2^{Q \setminus G}$  with initial state  $\iota_p = \emptyset$  for any  $p \in \mathcal{P}$ . The set of control messages  $C$  equals  $2^{Q \setminus G}$ , too. Furthermore, any tuple of local states is accepting.

To define the set of transitions, let  $A_1, A_2 \in S_p$  and  $A'_2 \in C$ . Moreover, let  $a = (\sigma, b_1, \dots, b_n, b) \in \Sigma_p \times \{0, 1\}^{n+1}$  and  $N = \{i \in [n] \mid b_i = 1\}$ . Then we have a transition

$$A_1 \xrightarrow{a, A'_2}_p A_2$$

iff the following conditions hold

- (1) if  $b = 0$ , then  $\iota.N^* \subseteq A_2$
- (2)  $A_1.\text{proc}.N^* \subseteq A_2$
- (3) if  $\sigma$  is a receive action, then  $A'_2.\text{msg}.N^* \subseteq A_2$
- (4) if  $\sigma$  is a send action, then  $A'_2 = A_2$ .

**Lemma 4.10.** *Let  $(\rho, \mu)$  be a run of the above CFM on  $(M, c_1, \dots, c_n, c)$  and let  $v_0 \in V$  with  $c(v_0) = 0$ . Then there does not exist  $W \in L_\pi$  with  $(M, c_1, \dots, c_n), v_0 \models W$ .*

*Proof.* Suppose there is  $W \in L_\pi$  with  $(M, c_1, \dots, c_n), v_0 \models W$ . Write  $W = w_0 a_1 w_1 \dots a_n w_n$  with  $a_k \in \{\text{proc}, \text{msg}\}$  and  $w_k \in [n]^*$  for all appropriate  $k$ . Since  $(M, c_1, \dots, c_n), v_0 \models W$ , there exist nodes  $v_k \in V$  with  $v_{k+1} = a_{k+1}(v_k)$  and  $w_k \subseteq N_{v_k}^*$  where  $N_{v_k} = \{i \in [n] \mid c_i(v_k) = 1\}$ . Since  $W \in L_\pi$ , there are states  $q_i \in Q$  with  $q_0 \in \iota.w_0$ ,  $q_{i+1} \in q_i.a_{i+1}w_{i+1}$ , and  $q_n \in G$ .

Since  $c(v_0) = 0$ , we have  $\iota.N_{v_0}^* \subseteq \rho(v_0)$  by (1) and therefore  $q_0 \in \iota.w_0 \subseteq \rho(v_0)$  by  $w_0 \in N_{v_0}^*$ . By induction, assume  $k < n$  and  $q_k \in \rho(v_k)$ . If  $a_{k+1} = \text{proc}$ , then by (2)  $q_{k+1} \in q_k.\text{proc}.N_{v_{k+1}}^* \subseteq \rho(v_{k+1})$ . If  $a_{k+1} = \text{msg}$ , then  $v_k$  is a send event. Hence, by (4),  $\mu(v_k) = \rho(v_k)$ . Since  $(v_k, v_{k+1}) \in \text{msg}$ , this implies  $\mu(v_{k+1}) = \rho(v_k)$ . Hence, by (3),  $q_{k+1} \in \rho(v_k).\text{msg}.N_{v_{k+1}}^* \subseteq \rho(v_{k+1})$ . This finishes the inductive argument. Hence  $q_n \in \rho(v_n) \cap G$ , contradicting our definition  $S_p = 2^{Q \setminus G}$ .  $\square$

**Lemma 4.11.** *Suppose  $(M, c_1, \dots, c_n, c)$  satisfies*

$$\forall v \in V : c(v) = 0 \implies \forall W \in L_\pi : (M, c_1, \dots, c_n), v \not\models W.$$

*Then  $(M, c_1, \dots, c_n, c)$  admits a run of the above CFM.*

*Proof.* For  $v \in V$ , let  $N_v = \{i \in [n] \mid c_i(v) = 1\}$ . Then define  $\rho(v)$  to be the union of the following sets

- (a)  $\iota.N_v^*$  if  $c(v) = 0$
- (b)  $\rho(\text{proc}^{-1}(v)).\text{proc}.N_v^*$  if  $\text{proc}^{-1}(v)$  is defined (i.e., if  $v$  is not minimal on its process)
- (c)  $\rho(\text{msg}^{-1}(v)).\text{msg}.N_v^*$  if  $\text{msg}^{-1}(v)$  is defined (i.e., if  $\lambda(v)$  is a receive action).

Furthermore, let

$$\mu(v) = \begin{cases} \rho(v) & \text{if } \lambda(v) \text{ is a send action} \\ \rho(\text{msg}^{-1}(v)) & \text{otherwise.} \end{cases}$$

Then, for any  $v \in V$ , the transition conditions (1-4) are satisfied by the mappings  $\rho$  and  $\mu$  (recall that the local initial states are  $\emptyset$ ).

Now, by contradiction, assume  $(\rho, \mu)$  is no run, i.e., there is some  $v_0 \in V$  with  $\rho(v_0) \notin 2^{Q \setminus G}$ . Hence there exists  $q_0 \in \rho(v_0) \cap G$ . Setting  $W_0 = \varepsilon$ , we therefore have

$$(M, c_1, \dots, c_n, c), v_k \models W_k, q_k \in \rho(v_k), \text{ and } q_k.W_k \cap G \neq \emptyset \quad (*)$$

for  $k = 0$ . Now assume that  $(*)$  holds for some  $k \geq 0$ .

If  $q_k \in \rho(v_k)$  because of (a), we have  $c(v_k) = 0$  and there exists  $w_k \in N_{v_0}^*$  with  $q_k \in \iota.w_k$ . But then  $(M, c_1, \dots, c_n, c), v_k \models w_k W_k$  and  $w_k W_k \in L_\pi$ , a contradiction. Hence we have  $q_k \in \rho(v_k)$  because of (b) or (c). If  $q_k \in \rho(\text{proc}^{-1}(v_k)).\text{proc}.N_{v_k}^*$ , then set  $v_{k+1} = \text{proc}^{-1}(v_k)$  and choose  $q_{k+1} \in \rho(v_{k+1})$  and  $w_k \in N_{v_k}^*$  with  $q_k \in q_{k+1}.\text{proc}.w_k$ . Setting  $W_{k+1} = \text{proc}.w_k.W_k$  yields  $(*)$  for  $k + 1$ . If  $q_k \in \rho(\text{msg}^{-1}(v_k)).\text{msg}.N_{v_k}^*$ , we can argue similarly.

Hence we find an infinite sequence of nodes  $v_0 > v_1 > v_2 \dots$  which is impossible since  $v_0$  dominates only a finite set. Thus,  $(\rho, \mu)$  is a run.  $\square$

**Proposition 4.12.** *There exists a CFM  $\mathcal{A}_0$  of index 1 that accepts  $(M, c_1, \dots, c_n, c)$  iff*

$$\forall v \in V : c(v) = 0 \implies \forall W \in L_\pi : (M, c_1, \dots, c_n), v \not\models W.$$

*The number of local states per process as well as the number of messages are in  $2^{O(s(\pi))}$ . Furthermore, any run of the CFM is accepting.*

*Proof.* The proof is immediate by the above two lemmas.  $\square$

**Any 1 is justified** We next construct a CFM that accepts  $(M, c_1, \dots, c_n, c)$  iff, for any  $v \in V$  with  $c(v) = 1$ , there exists  $W \in L_\pi$  with  $(M, c_1, \dots, c_n, c), v \models W$ . Again, the basic idea is simple: whenever the CFM encounters a node  $v$  with  $c(v) = 1$ , it will start the automaton  $\mathcal{C}$  (that accepts  $L_\pi$ ) and check that it can reach an accepting state along one of the possible paths. Thus, before, we had to prevent  $\mathcal{C}$  from reaching an accepting state. This time, we have to ensure that any verification of a  $c(v) = 1$  will eventually result in an accepting state being reached. For sequential Büchi-automata, solutions to this problem are known: collect some claims to be verified in one set and, only when all of them are verified, start verifying those claims that have been encountered during the previous verification phase. The resulting Büchi-automaton accepts iff the verification phase is changed infinitely often. We will adapt precisely this idea here. But then, the CFM would have to accept if, along *each and every* path, the verification phase changes infinitely often. This is the point where the CFM  $\mathcal{A}_{\text{Col}}$  comes into play since, by Corollary 3.9, it verifies that any path runs through infinitely many color changes. Thus, we will first construct a CFM that runs on tuples  $(M, c_0, c_1, \dots, c_n, c)$  where we assume that  $(M, c_0) \in \text{Col}$ . The actual CFM that verifies all claims  $c(v) = 1$  will run this newly constructed CFM in conjunction with  $\mathcal{A}_{\text{Col}}$  (that verifies  $(M, c_0) \in \text{Col}$ ) and project away the labeling  $c_0$ .

For any  $p \in \mathcal{P}$ , the set of local states  $S_p$  equals  $2^Q \times 2^Q \times \{0, 1\}$  with initial state  $\iota_p = (\emptyset, \emptyset, 1)$ , the set of control messages  $C$  equals  $2^Q \times 2^Q$ .

To define the set of transitions, let  $(A_1, B_1, d_1), (A_2, B_2, d_2) \in S_p$  and  $(A'_2, B'_2) \in C$ . Furthermore, let  $a = (\sigma, b_0, b_1, \dots, b_n, b) \in \Sigma_p \times \{0, 1\}^{n+2}$ . We use the following abbreviations

$$\begin{aligned} \overline{A_2} &= \begin{cases} A_2 & \text{if } \sigma \text{ is a receive action} \\ A_2 \cup A'_2 & \text{otherwise} \end{cases} \\ \overline{B_2} &= \begin{cases} B_2 & \text{if } \sigma \text{ is a receive action} \\ B_2 \cup B'_2 & \text{otherwise} \end{cases} \\ D &= \begin{cases} \overline{A_2} & \text{if } d_1 \neq d_2 \\ \overline{B_2} & \text{otherwise} \end{cases} \\ N &= \{i \in [n] \mid b_i = 1\} \end{aligned}$$

Then we set

$$(A_1, B_1, d_1) \xrightarrow{a, (A'_2, B'_2)}_p (A_2, B_2, d_2)$$

iff the following seven conditions hold

- (1)  $d_2 = b_0$
- (2) if  $b = 1$ , then  $\iota.N^* \cap (G \cup \overline{B_2}) \neq \emptyset$
- (3) if  $d_1 \neq d_2$ , then  $A_1 = \emptyset$
- (4)  $\forall q \in A_1 : q.\text{proc}.N^* \cap (G \cup \overline{A_2}) \neq \emptyset$
- (5)  $\forall q \in B_1 : q.\text{proc}.N^* \cap (G \cup D) \neq \emptyset$
- (6) if  $\sigma$  is a receive action, then  $\forall q \in A'_2 : q.\text{msg}.N^* \cap (G \cup \overline{A_2}) \neq \emptyset$
- (7) if  $\sigma$  is a receive action, then  $\forall q \in B'_2 : q.\text{msg}.N^* \cap (G \cup D) \neq \emptyset$ .

Recall that  $I$  is a set of channels and that we are only interested in MSCs that use precisely these channels infinitely often. Let  $(f_p)_{p \in \mathcal{P}} \in \prod_{p \in \mathcal{P}} S_p$  be accepting in  $\mathcal{A}$  iff  $f_p \in \{(\emptyset, \emptyset, 0), (\emptyset, \emptyset, 1)\}$  for all  $p \in \mathcal{P}$  that are not involved in any of the channels from  $I$ , i.e., that satisfy  $I \cap (\{p\} \times \mathcal{P} \cup \mathcal{P} \times \{p\}) = \emptyset$ . This finishes the construction of the CFM  $\mathcal{A}$  of index 1.

**Lemma 4.13.** *Let  $(\rho, \mu)$  be an accepting run of the above CFM  $\mathcal{A}$  on  $(M, c_0, c_1, \dots, c_n, c)$  and suppose  $(M, c_0) \in \text{Col}$  and  $I \subseteq \text{Inf}(M)$ . Then, for any  $v_0 \in V$  with  $c(v_0) = 1$ , there exists  $W \in L_\pi$  with  $(M, c_1, \dots, c_n), v_0 \models W$ .*

*Proof.* For  $v \in V$ , let

$$\begin{aligned}\rho(v) &= (A_v, B_v, d_v) \\ \mu(v) &= (A'_v, B'_v) \\ \overline{A}_v &= \begin{cases} A_v & \text{if } \lambda(v) \text{ is a receive action} \\ A_v \cup A'_v & \text{otherwise} \end{cases} \\ \overline{B}_v &= \begin{cases} B_v & \text{if } \lambda(v) \text{ is a receive action} \\ B_v \cup B'_v & \text{otherwise} \end{cases} \\ N_v &= \{i \in [n] \mid c_i(v) = 1\}\end{aligned}$$

Since  $c(v_0) = 1$ , (2) implies the existence of  $w_0 \in N_{v_0}^*$  and  $q_0 \in \iota.w_0 \cap (G \cup \overline{A}_{v_0} \cup \overline{B}_{v_0})$ . We now define a finite or infinite sequence  $(v_i, w_i, q_i)_{1 \leq i < N}$  (where  $N \in \mathbb{N} \cup \{\omega\}$ ) such that the following hold for all  $0 \leq i < N$ :

- (a)  $(v_i, v_{i+1}) \in \text{proc} \cup \text{msg}$
- (b)  $q_{i+1} \in q_i.a.N_{v_{i+1}}^*$  with  $a = \begin{cases} \text{proc} & \text{if } (v_i, v_{i+1}) \in \text{proc} \\ \text{msg} & \text{otherwise} \end{cases}$
- (c)  $q_{i+1} \in G \cup \overline{A}_{v_{i+1}} \cup \overline{B}_{v_{i+1}}$
- (d) if  $q_i \in \overline{A}_{v_i}$ , then  $q_{i+1} \in G \cup \overline{A}_{v_{i+1}}$
- (e) if  $q_{i+1} \notin G \cup \overline{A}_{v_{i+1}}$ , then  $q_i.a.N_{v_{i+1}}^* \cap (G \cup \overline{A}_{v_{i+1}}) = \emptyset$  with  $a = \begin{cases} \text{proc} & \text{if } (v_i, v_{i+1}) \in \text{proc} \\ \text{msg} & \text{otherwise} \end{cases}$
- (f)  $q_i \in G \iff N = i + 1$

Recall that  $q_0 \in G \cup \overline{A}_{v_0} \cup \overline{B}_{v_0}$ . Hence we can assume that the sequence has been constructed up to index  $i$  with  $q_i \in G \cup \overline{A}_{v_i} \cup \overline{B}_{v_i}$ . We distinguish five cases

- (i) If  $q_i \in G$ , then set  $N = i + 1$  which finishes the construction of the sequence (and ensures (f) *a posteriori* for all  $i < N$ ).
- (ii) Suppose  $q_i \in A_{v_i} \setminus G$ . Since  $A_{v_i} \neq \emptyset$  and since the run is accepting, the node  $v_i$  is not maximal on its process, so we can set  $v_{i+1} = \text{proc}(v_i)$ . Then, by (4), there exist  $w_{i+1} \in \text{proc} N_{v_{i+1}}^*$  and  $q_{i+1} \in q_i.w_{i+1} \cap (G \cup \overline{A}_{v_{i+1}})$ . With these choices, (a-e) hold for  $i$ .
- (iii) Suppose  $q_i \in \overline{A}_{v_i} \setminus (G \cup A_{v_i}) \subseteq A'_{v_i}$ . Then  $v_i$  is a send event, i.e.,  $v_{i+1} = \text{msg}(v_i)$  is a well-defined receive event. Hence, by (7), there exist  $w_{i+1} \in \text{msg} N_{v_{i+1}}^*$  and  $q_{i+1} \in q_i.w_{i+1} \cap (G \cup \overline{A}_{v_{i+1}})$ . With these choices, (a-e) hold for  $i$ .
- (iv) Suppose  $q_i \in B_{v_i} \setminus (G \cup \overline{A}_{v_{i+1}})$ . Since  $B_{v_i} \neq \emptyset$  and since the run is accepting, the node  $v_i$  cannot be maximal on its process, i.e.,  $v_{i+1} = \text{proc}(v_i)$  is well-defined. But then, by (5),  $\emptyset \neq q_i.\text{proc}.N_{v_{i+1}}^* \cap (G \cup \overline{A}_{v_{i+1}} \cup \overline{B}_{v_{i+1}})$ . Hence we can choose  $w_{i+1}$  and  $q_{i+1}$  such that (a-e) hold.
- (v) Finally, suppose  $q_i \in \overline{B}_{v_i} \setminus (G \cup \overline{A}_{v_i} \cup B_{v_i}) \subseteq B'_{v_i}$ . Then  $v_i$  is a send event, i.e.,  $v_{i+1} = \text{msg}(v_i)$  is a well-defined receive event. Hence, by (8),  $w_{i+1} \in \text{msg} N_{v_{i+1}}^*$  and  $\emptyset \neq q_i.N_{v_{i+1}} \cap (G \cup \overline{A}_{v_{i+1}} \cup \overline{B}_{v_{i+1}})$ . Hence we can choose  $w_{i+1}$  and  $q_{i+1}$  such that (a-e) hold.

If the construction can be carried out *ad infinitum*, then set  $N = \omega$  which, again, ensures (e) *a posteriori* for all  $i < N$ .

Now suppose  $N = \omega$ . Then, by Cor. 3.9, there exist  $0 < i < k$  with  $c_0(v_i) \neq c_0(v_{i+1})$  and  $c_0(v_k) \neq c_0(v_{k+1})$ . By (1), this implies  $d_{v_i} \neq d_{v_{i+1}}$  and  $d_{v_k} \neq d_{v_{k+1}}$ . Hence, by (3), we get  $\overline{A}_{v_i} = \overline{A}_{v_k} = \emptyset$ . Since, by (e),  $q_i \notin G$ , (c) implies  $q_i \in \overline{B}_{v_i}$ . Depending on whether  $(v_i, v_{i+1}) \in \text{proc}$  or not, (5) or (7) imply  $\emptyset \neq q_i.a.N_{v_{i+1}} \cap (G \cup \overline{A}_{v_{i+1}})$  and therefore  $q_{i+1} \in \overline{A}_{v_{i+1}}$  by (e) and (f). Hence, an iterative application of (d) and (f) imply  $q_k \in \overline{A}_{v_k}$ . But above, we showed that this set is empty. Hence  $N$  is finite.

Clearly,  $(M, c_1, \dots, c_n), v_{N-1} \models \varepsilon$ . From (b), we obtain  $(M, c_1, \dots, c_n), v_{N-2} \models w_{N-1}$  and, by induction,  $(M, c_1, \dots, c_n), v_0 \models W$  with  $W = w_0 w_1 \dots w_{N-1}$ . Since  $q_0 \in \iota.w_0$ , (d) implies  $q_{N-1} \in \iota.W$  and therefore  $W \in L_\pi$  follows from (f).  $\square$

**Lemma 4.14.** *Suppose  $(M, c_1, \dots, c_n, c)$  satisfies*

$$\forall v \in V : c(v) = 1 \implies \exists W \in L_\pi : (M, c_1, \dots, c_n), v \models W$$

and  $\text{Inf}(M) \subseteq I$ . Then there exists a mapping  $c_0 : V \rightarrow \{0, 1\}$  such that  $(M, c_0) \in \text{Col}$  and the above CFM  $\mathcal{A}$  accepts  $(M, c_0, c_1, \dots, c_n, c)$ .

*Proof.* For any  $v \in V$  with  $c(v) = 1$ , there exist  $0 \leq k^v \in \mathbb{N}$ ,  $w_0^v \in [n]^*$ ,  $w_i^v \in \{\text{proc}, \text{msg}\}[n]^*$  for  $1 \leq i \leq k^v$ , and  $q_i^v \in Q$  for  $0 \leq i \leq k^v$  such that

- (a)  $q_0^v \in \iota.w_0^v$ ,  $q_{i+1}^v \in q_i^v.w_{i+1}^v$  for  $1 \leq i < k^v$ , and  $q_{k^v}^v \in G$
- (b)  $v_0^v = v$  and, for  $0 \leq i < k^v$ ,  $v_{i+1}^v = \begin{cases} \text{proc}(w_i^v) & \text{if } w_{i+1}^v \in \text{proc}[n]^* \\ \text{msg}(w_i^v) & \text{if } w_{i+1}^v \in \text{msg}[n]^* \end{cases}$

We define inductively a sequence of subsets of  $V$ : Let  $H_0 = \emptyset$ . Inductively, let  $H_{n+1} \subseteq V \setminus \bigcup_{0 \leq i \leq n} H_i$  be nonempty and finite such that

- (A)  $\bigcup_{0 \leq i \leq n+1} H_i$  is downwards closed in  $M$
- (B) for any  $v \in V \setminus \bigcup_{0 \leq i \leq n+1} H_i$  with  $\lambda(v) = p?q$ ,
  - (B1) there exist infinitely many  $v' \in V \setminus \bigcup_{0 \leq i \leq n+1} H_i$  with  $\lambda(v) = \lambda(v')$
  - (B2) there exist  $u, u' \in H_{n+1}$  with  $(u, u') \in \text{msg}$  and  $\lambda(u') = \lambda(v)$
- (C) for any  $v \in H_n$  with  $c(v) = 1$ , we have  $v_{k^v}^v \in H_{n+1}$ .

Then  $V = \bigcup_{n \geq 0} H_n$ .

Now set, for  $v \in H_n$ ,

- $c_0(v) = n \bmod 2$  and  $d_v = c_0(v)$
- if  $v$  is a send event, then

$$\begin{aligned} A_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_{n-1}, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}} \text{ with } w_{i+1}^{\bar{v}} \in \text{proc}[n]^*\} \\ B_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_n, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}} \text{ with } w_{i+1}^{\bar{v}} \in \text{proc}[n]^*\} \\ A'_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_{n-1}, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}} \text{ with } w_{i+1}^{\bar{v}} \in \text{msg}[n]^*\} \\ B'_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_n, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}} \text{ with } w_{i+1}^{\bar{v}} \in \text{msg}[n]^*\} \end{aligned}$$

- if  $v$  is a receive event, then

$$\begin{aligned} A_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_{n-1}, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}}\} \\ B_v &= \{q_i^{\bar{v}} \mid \bar{v} \in H_n, v = v_i^{\bar{v}} \text{ for some } 1 \leq i < k^{\bar{v}}\} \\ A'_v &= A'_{\text{msg}^{-1}(v)} \\ B'_v &= B'_{\text{msg}^{-1}(v)} \end{aligned}$$

Then the pair of mappings  $(\rho, \mu)$  with  $\rho(v) = (A_v, B_v, d_v)$  and  $\mu(v) = (A'_v, B'_v)$  is a run of the CFM on  $(M, c_0, c_1, \dots, c_n, c)$  and  $(M, c_0) \in \text{Col}$ .  $\square$

**Proposition 4.15.** *Let  $I \subseteq \text{Ch}$ . There is a CFM  $\mathcal{A}$  that accepts  $(M, c_1, \dots, c_n, c)$  with  $\text{Inf}(M) = I$  iff*

$$\forall v \in V : c(v) = 1 \implies \exists W \in L_\pi : (M, c_1, \dots, c_n), v \models W.$$

*The number of local states per process is in  $2^{O(|\mathcal{P}|+s(\pi))}$  and the number of messages is in  $2^{O(s(\pi))}$ .*

*Proof.* By Lemma 3.2, there exists a CFM  $\mathcal{B}$  with the given number of states and messages that accepts  $(M, c_0, c_1, \dots, c_n, c)$  iff it is accepted by  $\mathcal{A}_{\text{Col}}$  from Prop. 3.7 and by the above CFM  $\mathcal{A}$ , i.e., iff  $(M, c_0) \in \text{Col}$ , and  $(M, c_0, c_1, \dots, c_n, c)$  is accepted by  $\mathcal{A}$ . Projecting away the function  $c_0$  gives the CFM  $\mathcal{A}_1$  by the above two lemmas.  $\square$

*Proof (of Theorem 4.7).* The result follows immediately from Prop. 4.12, 4.15, and Lemma 3.2.  $\square$

### 4.3 The overall construction

**Theorem 4.16.** *Let  $I \subseteq \text{Ch}$  and let  $\alpha$  be a local formula of PDL. Then one can construct a CFM  $\mathcal{B}$  of index 1 such that  $(M, (c_\beta)_{\beta \in \text{sub}(\alpha)})$  with  $\text{Inf}(M) = I$  is accepted by  $\mathcal{B}$  iff  $c_\beta : V \rightarrow \{0, 1\}$  is the characteristic function of the set of positions that satisfy  $\beta$  for all  $\beta \in \text{sub}(\alpha)$ .*

*With  $m$  the number of subformulas of the form  $\langle \pi \rangle \gamma$  and  $\langle \pi \rangle^{-1} \gamma$  and  $n \in \mathbb{N}$  such that  $s(\pi; \gamma) \leq n$  for all such subformulas, the number of local states per process is in  $2^{O(m(n+|\mathcal{P}|))}$  and the number of control messages is in  $2^{O(mn)}$ .*

*Proof.* The CFM  $\mathcal{B}$  has to accept  $(M, (c_\beta)_{\beta \in \text{sub}(\alpha)})$  iff

- (1)  $\mathcal{A}_\sigma$  accepts  $(M, c_\sigma)$  for all  $\sigma \in \text{sub}(\alpha) \cap \Sigma$  (cf. Example 4.1)
- (2)  $\mathcal{A}_\vee$  accepts  $(M, c_\gamma, c_\delta, c_{\gamma \vee \delta})$  for all  $\gamma \vee \delta \in \text{sub}(\alpha)$  (cf. Example 4.2)
- (3)  $\mathcal{A}_\neg$  accepts  $(M, c_\gamma, c_{\neg \gamma})$  for all  $\neg \gamma \in \text{sub}(\alpha)$  (cf. Example 4.2)
- (4)  $\mathcal{A}_{\langle \pi \rangle \gamma}$  accepts  $(M, c_{\alpha_1}, \dots, c_{\alpha_n}, c_\gamma, c_{\langle \pi \rangle \gamma})$  for all  $\langle \pi \rangle \gamma \in \text{sub}(\alpha)$  where  $\alpha_1, \dots, \alpha_n$  are those local formulas for which  $\{\alpha_i\}$  appears in the path expression  $\pi$  (cf. Theorem 4.7)
- (5)  $\mathcal{A}_{\langle \pi \rangle^{-1} \gamma}$  accepts  $(M, c_{\alpha_1}, \dots, c_{\alpha_n}, c_\gamma, c_{\langle \pi \rangle^{-1} \gamma})$  for all  $\langle \pi \rangle^{-1} \gamma \in \text{sub}(\alpha)$  where  $\alpha_1, \dots, \alpha_n$  are those local formulas for which  $\{\alpha_i\}$  appears in the path expression  $\pi$  (cf. Theorem 4.6).

Recall that the CFMs from (4) all have index 1, their number of local states per process is bounded by  $2^{O(n+|\mathcal{P}|)}$ , and their number of messages is bounded by  $2^{O(n)}$ . Hence, by Lemma 3.2, there exists a CFM of index 1 that checks all the requirements in (4). Its number of states is in

$$(m+1) \cdot \prod_{\langle \pi \rangle \gamma \in \text{sub}(\alpha)} 2^{O(n+|\mathcal{P}|)} \subseteq (m+1) \cdot 2^{O(m(n+|\mathcal{P}|))} \\ \subseteq 2^{O(m(n+|\mathcal{P}|))}.$$

and the number of control messages belongs to

$$\prod_{\langle \pi \rangle \gamma \in \text{sub}(\alpha)} 2^{O(s(\pi))} \subseteq 2^{O(mn)}.$$

Any tuple of local states in any of the CFMs from (5) is accepting. Furthermore, any of them has  $2^{O(n)}$  local states per process and equally many messages. Hence there is a CFM with  $2^{O(mn)}$  local states per process and equally many messages that checks all the requirements in (5). Furthermore, all tuples of states of this machine are accepting.

Recall that the CFMs  $\mathcal{A}_\sigma$ ,  $\mathcal{A}_\vee$ , and  $\mathcal{A}_\neg$  have just one local state per process, i.e., they only restrict the labels  $(\sigma, (b_\beta)_{\beta \in \text{sub}(\alpha)})$  allowed in  $M$ . Hence, without additional states or messages, one can change the above two CFMs into a CFM  $\mathcal{B}$  of index 1 that checks (1-5). Its number of local states per process is in  $2^{O(m(n+|\mathcal{P}|))}$  and its number of messages in  $2^{O(mn)}$ .  $\square$

## 5 Translation of global formulas

A *basic global formula* is a formula of the form  $A\alpha$  or  $E\alpha$  for  $\alpha$  a local formula. Then global formulas are positive Boolean combinations of basic global formulas.

**Proposition 5.1.** *Let  $\varphi$  be a global formula and  $I \subseteq \text{Ch}$ . Then one can construct a CFM  $\mathcal{A}$  that accepts  $M$  with  $\text{Inf}(M) = I$  iff  $M \models \varphi$ .*

*With  $\ell$  the number of basic global subformulas of  $\varphi$ ,  $m$  the number of subformulas of the form  $\langle \pi \rangle \beta$  and  $\langle \pi \rangle^{-1} \beta$ , and  $n \in \mathbb{N}$  such that  $s(\pi; \beta) \leq n$  for all such subformulas, the number of local states per process is in  $2^{O(m(n+|\mathcal{P}|))+|\mathcal{P}|\ell}$  and the number of control messages is in  $2^{O(\ell+mn)}$ .*

*Proof.* Let  $H$  be the set of basic global subformulas of  $\varphi$ . Let  $\beta = \bigwedge \{\alpha \mid E\alpha \in H \text{ or } A\alpha \in H\}$ . Using Proposition 3.3, one can construct a CFM that accepts  $(M, (c_\gamma)_{\gamma \in \text{sub}(\beta)})$  with  $\text{Inf}(M) = I$  iff

- $c_\gamma$  is the characteristic function of the set of positions satisfying  $\gamma$  for all  $\gamma \in \text{sub}(\beta)$  (Thm. 4.16)
- $M \models A\alpha$  for all  $A\alpha \in H$  (Example 4.3)
- $M \models E\alpha$  for all  $E\alpha \in H$  (Example 4.3).

Recall that the CFM checking  $c_\gamma$  as well as those checking  $A\alpha$  all have index 1 while the CFM for  $E\alpha$  have index  $|\mathcal{P}|$ . Hence the number of local states per process of the resulting CFM belongs to  $(|H| + 2) \cdot 2^{O(m(n+|\mathcal{P}|))} \cdot 2^{|H|} \cdot |\mathcal{P}|^{|H|} \subseteq 2^{O(m(n+|\mathcal{P}|)+|\mathcal{P}|^\ell)}$  and its number of messages is in  $2^{O(mn)}$ . Let  $\mathcal{A}_H$  denote the projection of this CFM to the set of MSCs (i.e., we project away the labelings  $c_\gamma$ ). Then  $\mathcal{A}_H$  accepts an MSC  $M$  with  $\text{Inf}(M) = I$  iff  $M \models \psi$  for all  $\psi \in H$ .

Now the CFM  $\mathcal{A}$  is the disjoint union of at most  $2^\ell$  many CFMs of the form  $\mathcal{A}_H$ .  $\square$

**Theorem 5.2.** *Let  $\varphi$  be a global formula of PDL. Then one can construct a CFM  $\mathcal{A}$  that accepts  $M$  iff  $M \models \varphi$ .*

*With  $\ell$  the number of basic global subformulas of  $\varphi$ ,  $m$  the number of subformulas of the form  $\langle \pi \rangle \beta$ , and  $n \in \mathbb{N}$  such that  $s(\pi; \beta) \leq n$  for all such subformulas, the number of local states per process is in  $2^{O(\ell+m(n+|\mathcal{P}|)+|\mathcal{P}|^2)}$  and the number of control messages is in  $2^{O(\ell+mn)}$ .*

*Proof.* Let, for  $I \subseteq \text{Ch}$ ,  $\mathcal{A}_I$  denote the CFM from Proposition 5.1 and let  $\mathcal{A}$  be the disjoint union of these CFMs.  $\square$

## 6 Model checking

### 6.1 CFMs vs. PDL specifications

We aim at an algorithm that decides whether, given a global formula  $\varphi$  and a CFM  $\mathcal{A}$ , every MSC  $M \in L(\mathcal{A})$  satisfies  $\varphi$ . The undecidability of this problem can be shown following, e.g., the proof in [MR04] (that paper deals with Lamport diagrams and a fragment  $\text{LD}_0$  of PDL, but the proof ideas can be easily transferred to our setting). To gain decidability, we follow the successful approach of, e.g., [MM01,GMSZ02,GKM06], and restrict attention to existentially  $B$ -bounded MSCs from  $L(\mathcal{A})$ .

For a finite or infinite word  $w \in \Sigma^\infty$  and  $a \in \Sigma$ , let  $|w|_a$  denote the number of occurrences of  $a$  in  $w$ . For  $0 \leq i \leq j < |w|$ , the infix  $w[i, j]$  is the factor of  $w$  starting in position  $i$  and ending in position  $j$ , i.e.,  $w = u w[i, j] v$  with  $|u| = i$  and  $|w[i, j]| = j - i + 1$ . If  $|w| > i$ , then we write  $w(i)$  for  $w[i, i]$ , the letter no.  $i + 1$  in  $w$  (note that  $w(0)$  is the first letter of  $w$ ).

Let  $M = (V, \leq, \lambda)$  be an MSC. A *linearization* of  $M$  is a linear order  $\preceq \supseteq \leq$  on  $V$  of order type at most  $\omega$  (i.e., also with respect to  $\preceq$ , any node  $v \in V$  dominates a finite set). Since equally-labeled nodes of  $M$  are comparable, we can safely identify a linearization of  $M$  with a word from  $\Sigma^\infty$ .

A word  $w \in \Sigma^\infty$  is  *$B$ -bounded* (where  $B \in \mathbb{N}$ ) if, for any  $(p, q) \in \text{Ch}$  and any finite prefix  $u$  of  $w$ ,  $0 \leq |u|_{p!q} - |u|_{q?p} \leq B$ . An MSC  $M$  is *existentially  $B$ -bounded* if it admits a  $B$ -bounded linearization. Intuitively, this means that the MSC  $M$  can be scheduled in such a way that none of the channels  $(p, q)$  ever contains more than  $B$  pending messages.

**Lemma 6.1.** *A  $B$ -bounded word  $w \in \Sigma^\infty$  is a linearization of some MSC  $M$  iff, for any  $(p, q) \in \text{Ch}$ , any finite prefix of  $w$  can be extended to a finite prefix  $u$  of  $w$  such that*

- (1)  $|u|_{p!q} = |u|_{q?p}$  or
- (2) the last letter of  $u$  is  $p!q$  or  $q?p$ .

*Proof.* First suppose that  $w$  is a linearization of some MSC. Then  $|w|_{p!q} = |w|_{q?p}$ . If this number is finite, we can extend any finite prefix to some finite prefix satisfying (1). Otherwise, any suffix contains at least one occurrence of  $p!q$ , so any prefix can be extended to some larger prefix ending with  $p!q$ .

Conversely suppose that any finite prefix can be extended to a finite prefix satisfying (1) or (2). We construct from  $w$  an MSC as follows:

- the set of nodes equals  $V = \{v \in \mathbb{N} \mid v < |w|\}$ ,

- for  $v \in V$  let  $\lambda(v) = w(v)$ ,
- let  $(i, j) \in \text{proc}'$  iff  $0 \leq i < j < |w|$  and there exists a process  $p \in \mathcal{P}$  with  $\lambda(i), \lambda(j) \in \Sigma_p$  and, for all  $k$  with  $i \leq k < j$  and  $\lambda(k) \in \Sigma_p$ , we have  $i = k$ ,
- let  $(i, j) \in \text{msg}'$  iff  $i, j \in V$  and there exists a channel  $(p, q) \in \text{Ch}$  such that  $w(i) = p!q$ ,  $w(j) = q?p$ , and  $|w[0, i]|_{p!q} = |w[0, j]|_{q?p}$ ,
- then set  $\preceq = (\text{msg}' \cup \text{proc}')^* \subseteq V^2$ .

Suppose  $(i, j) \in \text{msg}'$  and  $j < i$ . Then  $|w[0, j]|_{p!q} - |w[0, j]|_{q?p} < |w[0, i]|_{p!q} - |w[0, j]|_{q?p} = 0$ , contradicting the  $B$ -boundedness of  $w$ . Hence  $\text{msg}'$  and  $\text{proc}'$  are contained in  $\leq$  proving that  $\preceq$  is a partial order on  $V$ . Since  $\preceq$  is contained in the natural order  $\leq$  on the set of natural numbers  $V$ , the word  $w$  is a linearization of  $M = (V, \preceq, \lambda)$ . It therefore remains to be shown that  $M$  is an MSC:

- It is easily verified that  $\text{msg} = \text{msg}'$  and  $\text{proc} = \text{proc}'$  implying  $\preceq = (\text{msg} \cup \text{proc})^*$ .
- By the definition of  $\text{proc}'$ , any two nodes  $i$  and  $j$  with  $P(i) = P(j)$  are ordered by  $\preceq$ .
- Let  $(p, q) \in \text{Ch}$  be some channel. Since  $w$  is  $B$ -bounded, we have  $|w|_{p!q} \geq |w|_{q?p}$  (since this holds for any prefix of  $w$ ). Now suppose  $|w|_{p!q} > |w|_{q?p}$ . Then there are only finitely many occurrences of  $q?p$ ; let  $u_1$  with  $|u_1|_{p!q} - |u_1|_{q?p} > 0$  be a finite prefix of  $w$  that contains all occurrences of  $q?p$ . Then by our assumption on  $w$ , we can extend  $u_1$  to a finite prefix  $u_2$  of  $w$  whose last letter is  $p!q$  (since there are no occurrences of  $q?p$  beyond  $u_1$ ). Hence  $|u_2|_{p!q} - |u_2|_{q?p} > |u_1|_{p!q} - |u_1|_{q?p}$ . Inductively, we find a finite prefix  $u$  with  $|u|_{p!q} - |u|_{q?p} > B$ , contradicting the  $B$ -boundedness of  $w$ . Hence  $|\lambda^{-1}(p!q)| = |w|_{p!q} = |w|_{q?p} = |\lambda^{-1}(q?p)|$  which finishes the proof that  $(V, \preceq, \lambda)$  is an MSC.  $\square$

We next construct, from a CFM  $\mathcal{A} = (C, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  and a bound  $B \in \mathbb{N}$ , a finite transition system over  $\Sigma$  with multiple Büchi-acceptance conditions that accepts the set of  $B$ -bounded linearizations of MSCs from  $L(\mathcal{A})$ :

- A configuration is a tuple  $((s_p)_{p \in \mathcal{P}}, \chi, (p, q))$  with  $s_p \in S_p$  for all  $p \in \mathcal{P}$ ,  $\chi : \text{Ch} \rightarrow C^*$  with  $|\chi(p', q')| \leq B$  for all  $(p', q') \in \text{Ch}$ , and  $(p, q) \in \text{Ch}$ .
- We have a transition

$$((s_p^1)_{p \in \mathcal{P}}, \chi^1, (p^1, q^1)) \xrightarrow{a} ((s_p^2)_{p \in \mathcal{P}}, \chi^2, (p^2, q^2))$$

for an action  $a \in \Sigma_p$  iff there exists a control message  $c \in C$  such that

- (T1)  $s_p^1 \xrightarrow{a, c} s_p^2$  is a transition of the local machine  $\mathcal{A}_p$  and  $s_q^1 = s_q^2$  for  $q \neq p$ .
- (T2) Send events: if  $a = p!q$ , then  $\chi^2(p, q) = \chi^1(p, q) c$  (i.e., message  $c$  is inserted into the channel from  $p$  to  $q$ ) and  $\chi^1(p', q') = \chi^2(p', q')$  for  $(p', q') \neq (p, q)$  (i.e., all other channels are unchanged)
- (T3) Receive events: if  $a = p?q$ , then  $\chi^1(q, p) = c \chi^2(q, p)$  (i.e., message  $c$  is deleted from the channel from  $q$  to  $p$ ) and  $\chi^1(q', p') = \chi^2(q', p')$  for  $(q', p') \neq (q, p)$  (i.e., all other channels are unchanged)
- (T4)  $(p^2, q^2)$  is the channel that  $a$  writes to or reads from.
- The initial configuration is the tuple  $((\iota_p)_{p \in \mathcal{P}}, \chi, (p, q))$  with  $\chi(p', q') = \varepsilon$  for all  $(p', q') \in \text{Ch}$ , and where  $(p, q) \in \text{Ch}$  is an arbitrary but fixed channel, i.e., the local machines are in their initial state and all channels are empty.

A finite or infinite path  $((s_p^i)_{p \in \mathcal{P}}, \chi^i, (p^i, q^i))_{0 \leq i < \varkappa}$  (for some  $\varkappa \in \mathbb{N} \cup \{\omega\}$ ) in this transition system is *successful* if

- (S1) there exists a tuple  $(f_p)_{p \in \mathcal{P}} \in F$  such that, for all  $p \in \mathcal{P}$  and  $0 \leq i < \varkappa$ , there exists  $i \leq j < \varkappa$  with  $s_p^j = f_p$  and
- (S2) for all  $(p, q) \in \text{Ch}$  and  $0 \leq i < \varkappa$ , there exists  $i \leq j < \varkappa$  such that  $\chi^i(p, q) = \varepsilon$  or  $(p^j, q^j) = (p, q)$ .

**Lemma 6.2.** *Let  $w \in \Sigma^\infty$ . Then the following are equivalent:*

- (i)  $w$  is the label of some successful path in the above transition system
- (ii)  $w$  is a  $B$ -bounded linearization of some MSC from  $L(\mathcal{A})$ .

*Proof.* To prove the implication (ii) $\Rightarrow$ (i), let  $M = (V, \preceq, \lambda) \in L(\mathcal{A})$  be an MSC accepted by  $\mathcal{A}$ , let  $w \in \Sigma^\infty$  be a  $B$ -bounded linearization of  $M$ , and let  $(\mu, \rho)$  be a successful run of  $\mathcal{A}$  on  $M$ . Without loss of generality, we can assume  $V = \{v \in \mathbb{N} \mid 0 \leq v < |w|\}$  and  $\preceq \subseteq \leq$  such that  $w$  is the sequence of labels of  $(V, \preceq, \lambda)$ . For  $i = 0$ , let  $((s_p^i), \chi^i, (p^i, q^i))$  be the initial configuration of the transition system. Now let  $i > 0$ . For  $p \in \mathcal{P}$ , let  $s_p^i = \iota_p$  if there is no  $0 \leq j < i$  with  $w(j) \in \Sigma_p$ ; otherwise set  $s_p^i = \rho(j)$  for  $j$  the maximal natural number with  $j < i$  and  $w(j) \in \Sigma_p$ . For  $(p, q) \in \text{Ch}$ , set  $\chi^i(p, q) = \mu(j_1) \mu(j_2) \dots \mu(j_k)$  where  $0 \leq j_1 < j_2 < \dots < j_k < i$  is the sequence of those nodes from  $V$  with  $\lambda(j_\ell) = p!q$  and  $\text{msg}(j_\ell) \geq i$  (since  $w$  is  $B$ -bounded, we have  $0 \leq k \leq B$ ). Finally,  $(p^i, q^i)$  is the channel that the action  $w(i-1)$  writes to or reads from. Then it can be checked that the sequence of these configurations  $((s_p^i), \chi^i, (p^i, q^i))_{0 \leq i < |w|}$  forms a  $w$ -labeled path in the transition system. We show that it is successful:

- (S1) Since  $(\rho, \mu)$  is successful, there exists  $(f_p)_{p \in \mathcal{P}} \in F$  such that for all  $p \in \mathcal{P}$  and all  $v \in V$  with  $\lambda(v) \in \Sigma_p$ , there exists  $v' \in V$  with  $\lambda(v') \in \Sigma_p$ ,  $v \preceq v'$ , and  $\rho(v') = f_p$  (or  $f_p = \iota_p$  if no such node  $v$  exists). Now let  $0 \leq i < |w|$  and let  $v < i$  denote the maximal natural number with  $w(v) \in \Sigma_p$  (the case that no such number exists is left to the reader). Then there exists  $v' \in V$  with  $\lambda(v') \in \Sigma_p$ ,  $v \preceq v'$ , and  $\rho(v') = f_p$ . Because of the maximality of  $v$ , we obtain  $i < v'$ . Furthermore,  $s_p^{v'+1} = \rho(v') = f_p$ .
- (S2) Let  $0 \leq i < |w|$ . Since  $w$  is  $B$ -bounded, the previous lemma implies the existence of  $i \leq j < |w|$  such that  $|w[0, j]|_{p!q} = |w[0, j]|_{q?p}$  or the last letter of  $w[0, j]$  is  $p!q$  or  $q?p$ . Hence  $\chi^{j+1}(p, q) = \varepsilon$  or  $(p^{j+1}, q^{j+1}) = (p, q)$ .

Conversely assume (i). Since all the channels in the transition system contain at most  $B$  messages, the word  $w$  is  $B$ -bounded. Since the  $w$ -labeled path satisfies (S2), the word  $w$  satisfies (1) and (2) in the previous lemma. Hence  $w$  is the linearization of some MSC. Now, using (S1) it can be verified similarly to the above that this MSC is accepted by  $\mathcal{A}$ .  $\square$

**Theorem 6.3.** *The following can be decided in polynomial space:*

*Input:*  $B \in \mathbb{N}$  (given in unary), CFM  $\mathcal{B}$ , and a global formula  $\varphi \in \text{PDL}$ .

*Question:* Is there an existentially  $B$ -bounded MSC  $M \in L(\mathcal{B})$  with  $M \models \varphi$ ?

*Proof.* Theorem 5.2 allows to build a CFM  $\mathcal{A}_\varphi$  that accepts  $M$  iff  $M \models \varphi$ . From Proposition 3.3, we then obtain a CFM  $\mathcal{A}$  with  $L(\mathcal{A}) = L(\mathcal{B}) \cap L(\mathcal{A}_\varphi)$ , i.e.,  $M \in L(\mathcal{A})$  iff  $M \in L(\mathcal{B})$  and  $M \models \varphi$ . To decide the existence of some existentially  $B$ -bounded MSC in  $L(\mathcal{A})$ , it suffices to decide whether the above transition system has some successful path. Recall that such a path has to simultaneously satisfy  $b = |\mathcal{P}| + |\text{Ch}|$  many Büchi-conditions. Extending the configurations of the transition system by a counter that counts up to  $b+1$  allows to have just one Büchi-condition [Cho74]. Note that any configuration of the resulting transition system can be stored in space

$$\log(b) + |\mathcal{P}| \log n + |\text{Ch}| B \log |C| + \log |\text{Ch}|$$

where  $C$  is the set of message contents of  $\mathcal{A}$  and  $n$  is the maximal number of local states a process of  $\mathcal{A}$  has. But due to Theorem 5.2 the size of the CFM  $\mathcal{A}_\varphi$  is exponential in the size of  $\varphi$ . By Proposition 3.3,  $\mathcal{A}$  stays exponential in the size of the input. Hence, the model checking problem can be decided in polynomial space.  $\square$

## 6.2 HMSCs vs. PDL specifications

In [Pel00], Peled provides a PSPACE model checking algorithm for high-level message sequence charts (HMSCs) against formulas of the logic  $\text{TLC}^-$ . The logic  $\text{TLC}^-$  is a fragment of our logic PDL as can be shown easily. Now, we aim to model check an HMSC against a global formula of PDL, and, thereby, to generalize Peled's result.

**Definition 6.4.** An HMSC  $\mathcal{H} = (S, \rightarrow, s_0, c, \mathcal{M})$  is given as a finite, directed graph  $(S, \rightarrow)$  with initial node  $s_0 \in S$ ,  $\mathcal{M}$  a finite set of finite MSCs, and a labeling function  $c : S \rightarrow \mathcal{M}$ .

For defining the semantics of HMSCs we need a concatenation operation. Let  $M_1 = (V_1, \leq_1, \lambda_1)$  and  $M_2 = (V_2, \leq_2, \lambda_2)$  be two finite MSCs over the same process set  $\mathcal{P}$  with disjoint node sets. Then  $M_1 M_2 = (V, \leq, \lambda)$  is given by  $V = V_1 \cup V_2$ ,  $\lambda = \lambda_1 \cup \lambda_2$ , and  $\leq$  is the least partial order containing  $\leq_1 \cup \leq_2$  and  $\{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2, P(v_1) = P(v_2)\}$ . Informally, the events of  $M_2$  succeed the events of  $M_1$  for each process, respectively.

Let  $\mathcal{H} = (S, \rightarrow, s_0, c, \mathcal{M})$  be an HMSC. Let  $\eta$  be a maximal path of  $(S, \rightarrow)$  starting in  $s_0$ , i.e., either a path  $\eta = s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n$  that ends in an  $s_n \in S$  such that there is no  $s \in S$  with  $s_n \rightarrow s$  or an infinite path  $\eta = s_0 \rightarrow s_1 \rightarrow \dots$ . The labeling function  $c$  can now be extended to paths by  $c(\eta) = c(s_0)c(s_1)\dots$ . The MSC language of the HMSC  $\mathcal{H}$  is now  $L(\mathcal{H}) = \{c(\eta) \mid \eta \text{ is a maximal path starting in } s_0\}$ . Note that for any HMSC  $\mathcal{H}$  the language  $L(\mathcal{H})$  is existentially  $B$ -bounded for some  $B \in \mathbb{N}$ . Indeed, since any finite MSC  $M$  is existentially  $B_M$ -bounded for some  $B_M \in \mathbb{N}$ , there is a  $B$ -bounded linearization for every  $c(\eta)$  when  $B = \max\{B_M \mid M \in \mathcal{M}\}$ .

**Theorem 6.5.** *The following problem is decidable in polynomial space:*

*Input: An HMSC  $\mathcal{H}$  and a global formula  $\varphi \in \text{PDL}$ .*

*Question: Is there an MSC  $M \in L(\mathcal{H})$  with  $M \models \varphi$ ?*

*Proof.* Let  $\mathcal{H} = (S, \rightarrow, s_0, c, \mathcal{M})$  be an HMSC. For every  $s \in S$  we can find a linearization  $w_s = a_1 \dots a_m \in \Sigma^*$  of  $c(s)$ . For this linearization, it is easy to construct a finite automaton accepting  $w_s$  only. In detail  $\mathcal{S}_s = (Q_s, T_s, r_s^0, \{r_s^m\})$  with  $Q_s = \{r_s^0, r_s^1, \dots, r_s^m\}$ ,  $T_s = \{(r_s^{i-1}, a_i, r_s^i) \mid i = 1, \dots, m\}$ ,  $r_s^0$  initial, and  $r_s^m$  final. Now, we put  $\mathcal{S}_{\mathcal{H}} = (Q, T, r_{s_0}^0)$  with  $Q = \bigcup_{s \in S} Q_s$  the disjoint union of the different state sets. We have a transition  $(r_s^i, a, r_{s'}^j) \in T$  iff

- $s = s'$  and  $i + 1 = j < m$  where  $m + 1$  is the number of states for  $\mathcal{S}_s$ , or
- $s = s'$ ,  $i + 1 = j = m$ , and there is no  $s'' \in S$  with  $s \rightarrow s''$ , or
- $s \rightarrow s'$ ,  $i = m - 1$ , and  $j = 0$ .

A run in  $\mathcal{S}_{\mathcal{H}}$  is successful if

- it is finite and ends in a state  $r_s^m$  for some  $s \in S$ , or
- it is infinite and one of the states from  $\{r_s^0 \mid s \in S\}$  is traversed infinitely often.

Then  $\mathcal{S}_{\mathcal{H}}$  accepts a linearization for each and every MSC  $M \in L(\mathcal{H})$ , and, vice versa, each (finite or infinite) word accepted by  $\mathcal{S}_{\mathcal{H}}$  is a linearization of an  $M \in L(\mathcal{H})$ . Note that the size of  $\mathcal{S}_{\mathcal{H}}$  is linear in the size of  $\mathcal{H}$ .

By Theorem 5.2, we can build a CFM  $\mathcal{A}_{\varphi}$  with  $M \in L(\mathcal{A}_{\varphi})$  iff  $M \models \varphi$ . From  $\mathcal{A}_{\varphi}$  we construct in the same way as for Lemma 6.2 a sequential transition system  $\mathcal{S}_{\varphi}$  out of  $\mathcal{A}_{\varphi}$  but this time without any channel bound, i.e.,  $\mathcal{S}_{\varphi}$  may have infinitely many states. But since we are interested in running the systems  $\mathcal{S}_{\mathcal{H}}$  and  $\mathcal{S}_{\varphi}$  synchronously and  $\mathcal{S}_{\mathcal{H}}$  is implicitly existentially  $B$ -bounded for some  $B \in \mathbb{N}$ , the resulting system  $\mathcal{S}$  is finite again. In detail, we construct  $\mathcal{S}$  as follows (cf. the construction before Lemma 6.2): Let  $(r_{s_0}^0, (\iota_p)_{p \in \mathcal{P}}, \chi, (p, q))$  with  $\chi(p', q') = \varepsilon$  for all  $(p', q') \in \text{Ch}$ , and where  $(p, q) \in \text{Ch}$  is an arbitrary but fixed channel, be the initial state of  $\mathcal{S}$ . We construct the state set and the transition relation of  $\mathcal{S}$  stepwise. Let  $(r_s^i, (s_p^1)_{p \in \mathcal{P}}, \chi^1, (p^1, q^1))$  be already constructed. Then we add state  $(r_{s'}^j, (s_p^2)_{p \in \mathcal{P}}, \chi^2, (p^2, q^2))$  and a transition with label  $a \in \Sigma$  between the two states iff

- $(r_s^i, a, r_{s'}^j)$  is a transition in  $\mathcal{S}_{\mathcal{H}}$  and
- $((s_p^1)_{p \in \mathcal{P}}, \chi^1, (p^1, q^1)), a, ((s_p^2)_{p \in \mathcal{P}}, \chi^2, (p^2, q^2))$  is a transition in  $\mathcal{S}_{\varphi}$  according to conditions (T1) to (T4) defined above.

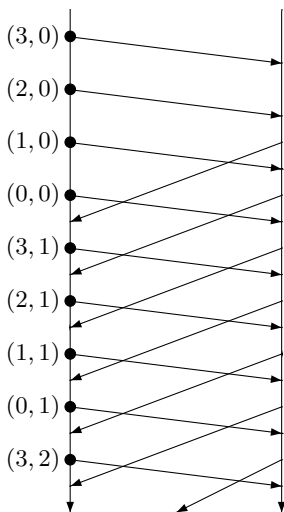
Note that for the both cases  $s \rightarrow s'$ ,  $i = m - 1$ ,  $j = 0$  and  $s = s'$ ,  $i + 1 = j = m$  we have  $\chi^2(p, q) = \varepsilon$  for each  $(p, q) \in \text{Ch}$ , i.e., all channels are empty. Indeed, for these states the MSC  $c(s)$  is executed completely, and, hence, the channels are empty. Therefore,  $\chi(p, q)$  is bounded for all channels  $(p, q)$  and all states constructed, i.e., the above construction terminates. A run in  $\mathcal{S}$  is successful if both

projections of the run on  $\mathcal{S}_{\mathcal{H}}$  and on  $\mathcal{S}_{\varphi}$  are successful. Now,  $\mathcal{S}$  admits a successful run iff there is a existentially  $B$ -bounded linearization  $w_M$  of some  $M \in L(\mathcal{H})$  (where  $B$  is implicitly given by  $\mathcal{H}$ ) which is accepted by  $\mathcal{S}_{\varphi}$  where the state set is restricted to those states with channel capacity at most  $B$ . In light of Lemma 6.2 this is the case iff  $M$  is accepted by  $\mathcal{A}_{\varphi}$ , i.e.,  $M \models \varphi$ . An analysis similar to this in the proof of Theorem 6.3 shows that the existence of a successful path of  $\mathcal{S}$  can be decided in polynomial space.  $\square$

## 7 PDL with intersection

We define a richer logic, called PDL with intersection or iPDL for short. In addition to the local formulas of PDL, we allow local formulas  $\langle \pi_1 \cap \pi_2 \rangle \alpha$  where  $\pi_1$  and  $\pi_2$  are path expressions and  $\alpha$  is a local formula. The intended meaning is that there exist two paths described by  $\pi_1$  and  $\pi_2$  respectively that both lead to the same node  $w$  where  $\alpha$  holds. It is the aim of this section to prove that CFMs are too weak to check all properties expressed in iPDL. To show this result more easily, we also allow atomic propositions of the form  $(a, b)$  with  $a, b \in \{0, 1\}$ ; they are evaluated over an MSC  $M = (V, \leq, \lambda)$  together with a mapping  $c : V \rightarrow \{0, 1\}^2$ . Then  $(M, c), v \models (a, b)$  iff  $c(v) = (a, b)$ . Let  $\mathcal{P} = \{0, 1\}$  be the set of processes. For  $m \geq 1$ , we first fix an MSC  $M_m = (V_m, \leq, \lambda)$  for the remaining arguments: On process 0, it executes the sequence  $(0!1)^m((0?1)(0!1))^\omega$ . The sequence of events on process 1 is  $(1?0)((1?0)(1!0))^\omega$ . In other words, process 0 sends  $m$  messages to process 1 and then acknowledges any message received from 1 immediately. Differently, process 1 has a buffer for two messages. After receiving message number  $k + 1$ , it acknowledges message number  $k$ .

Let  $E_{0!1}$  denote the set of send-events of process 0. For the  $k^{\text{th}}$  send-event  $v$  on process 0, let  $f(v) = ((m - k) \bmod m, (k - 1) \text{div } m)$ . Then  $f$  maps the set  $E_{0!1}$  bijectively onto the grid  $G_m = \{0, 1, \dots, m - 1\} \times \omega$ ; we denote the inverse of  $f$  by  $g$ . Figure 3 shows MSC  $M_4$  together with the mapping  $f$ .



**Fig. 3.** MSC  $M_4$  and the mapping  $f$ .

**Lemma 7.1.** *There exists a local formula  $\alpha$  of iPDL such that, for any  $m \geq 1$  and any  $c : V_m \rightarrow \{0, 1\}^2$  satisfying  $c(g(i, j)) = (0, 0)$  iff  $i = 0$ , we have  $(M_m, c) \models A\alpha$  iff  $c(g(i, j)) = c(g(i, j + i))$  for all  $(i, j) \in G_m$ .*

*Proof.* Let  $(i, j) \in G_m$ . Then observe the following:

- With  $\pi_1$  denoting the path description  $(\text{proc}; \{(0?1)\})^*; \text{proc}; \{(0!1)\}$ , we have  $M_m, g(i, j) \models \langle \pi_1 \rangle \beta$  iff  $i > 0$  and  $M_m, g(i-1, j) \models \beta$ , or  $i = 0$  and  $M_m, g(m-1, j+1) \models \beta$ .
- With  $\pi_2$  denoting the path description  $\text{msg}; \text{proc}; \text{msg}; \text{proc}$ , we have  $M_m, g(i, j) \models \langle \pi_2 \rangle \beta$  iff  $M_m, g(i+1, j+1) \models \beta$  whenever  $i < m-1$ .

As a consequence, we obtain

- if  $i > 0$ , then  $M_m, g(i, j) \models \langle \pi_1; \pi_2 \rangle \beta$  iff  $M_m, g(i, j+1) \models \beta$ .

Now let  $c : V_m \rightarrow \{0, 1\}^2$  be a function with  $c(g(i, j)) = (0, 0)$  iff  $i = 0$ . Then we have

1.  $(M_m, c), g(i, j) \models \langle \{\neg(0, 0)\}; (\pi_1; \pi_2)^* \rangle \beta$  iff  $i > 0$  and there exists  $k \geq 0$  with  $(M_m, c), g(i, j+k) \models \beta$ ,
2.  $(M_m, c), g(i, j) \models \langle \{\neg(0, 0)\}; \pi_1^*; \{(0, 0)\} \rangle \beta$  iff  $(M_m, c), g(0, j) \models \beta$ ,
3.  $(M_m, c), g(0, j) \models \langle \{\pi_2; \{\neg(0, 0)\}\}^* \rangle \beta$  iff there exists  $0 \leq k \leq m-1$  with  $(M_m, c), g(k, j+k) \models \beta$ .

Now let  $\pi_3 = (\{\neg(0, 0)\}; \pi_1)^*; \{(0, 0)\}; (\pi_2; \{\neg(0, 0)\})^*$  and  $\pi_4 = \{\neg(0, 0)\}; (\pi_1; \pi_2)^*$ . Then, we have  $(M_m, c), g(i, j) \models \langle \pi_3 \cap \pi_4 \rangle \beta$  iff  $i > 0$  and  $(M_m, c), g(i, j+i) \models \beta$ . Now let

$$\alpha = ((0!1) \wedge \neg(0, 0)) \rightarrow \bigwedge_{x \in \{0, 1\}^2} x \leftrightarrow \langle \pi_3 \cap \pi_4 \rangle x .$$

Then, for all  $(i, j) \in G_m$ , we have  $(M_m, c), g(i, j) \models \alpha$  iff  $c(g(i, j)) = c(g(i, j+i))$ .  $\square$

**Lemma 7.2.** *Let  $\mathcal{A} = (C, 2, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$  be a CFM that accepts all labeled MSCs  $(M_m, c)$  with  $m \geq 1$  such that*

- (1)  $c(g(i, j)) = (0, 0)$  iff  $i = 0$
- (2)  $c(g(i, j)) = c(g(i, j+i))$  for all  $(i, j) \in G_m$ .

*Then there exist  $m \geq 1$  and a labeled MSC  $(M_m, c)$  accepted by  $\mathcal{A}$ , satisfying (1), and violating (2).*

*Proof.* Let  $\mathcal{A}_p = (S_p, \rightarrow_p, \iota_p)$  for  $p = 0, 1$ , let  $m \geq 1$  be such that  $|S_0| \cdot |S_1| \cdot |C|^{m-1} < 3^{\frac{(m-1)(m-2)}{2}}$ , and let  $M_m = (V_m, \leq, \lambda)$ . Let furthermore  $H$  denote the set of mappings  $c : V_m \rightarrow \{0, 1\}^2$  satisfying (1), (2), and  $c(v) = (1, 1)$  for all  $v \notin E_{0!1}$  (i.e.,  $\lambda(v) \neq (0!1)$ ). Then, for any  $c \in H$ , the pair  $(M_m, c)$  is accepted by  $\mathcal{A}$  – let  $(\rho_c, \mu_c)$  be an accepting run of  $\mathcal{A}$  on  $(M_m, c)$ .

Let  $v = g(m-1, m-2)$  and  $W = \{w \in V \mid w \leq v\}$ . Then, for any event  $w \in W$  with  $\lambda(w) = !10$ , we have  $\text{msg}(w) \in W$ . On the other hand, there are precisely  $m-1$  events  $w_1, \dots, w_{m-1} \in W$  with  $\lambda(w_i) = 0!1$  and  $\text{msg}(w_i) \notin W$ . Let furthermore  $u \in W$  be the maximal event from process 1.

Consider  $(M_m, c_1)$  and  $(M_m, c_2)$  with  $c_1, c_2 \in H$  and  $c_1(g(i, j)) = c_2(g(i, j))$  for all  $0 \leq j < i < m$ . Then  $c_1 = c_2$  by (2). Hence  $|H|$  is the number of mappings from  $\{(i, j) \mid 0 \leq j < i < m\}$  to  $\{0, 1\}^2 \setminus \{(0, 0)\}$ , i.e.,  $3^{\frac{(m-1)(m-2)}{2}}$ .

Since this number exceeds  $|S_0| \cdot |S_1| \cdot |C|^{m-1}$ , there exist  $c_1$  and  $c_2$  in  $H$  with  $\rho_{c_1}(v) = \rho_{c_2}(v)$ ,  $\rho_{c_1}(u) = \rho_{c_2}(u)$ , and  $\mu_{c_1}(w_i) = \mu_{c_2}(w_i)$  for all  $1 \leq i \leq m-1$ .

Now define a mapping  $c : V \rightarrow \{0, 1\}^2$  by  $c(x) = c_1(x)$  for  $x \in W$  and  $c(x) = c_2(x)$  for  $x \notin W$ . Then,  $c$  satisfies (1) and violates (2). But  $(M_m, c)$  is accepted by  $\mathcal{A}$ : An accepting run  $(\rho, \mu)$  is defined (similarly to  $c$ ) by

$$\rho(x) = \begin{cases} \rho_{c_1}(x) & \text{for } x \in W \\ \rho_{c_2}(x) & \text{otherwise} \end{cases} \quad \text{and} \quad \mu(x) = \begin{cases} \mu_{c_1}(x) & \text{for } x \in W \\ \mu_{c_2}(x) & \text{otherwise.} \end{cases}$$

$\square$

**Theorem 7.3.** *There exists a local formula  $\alpha$  of iPDL such that the set of MSCs  $M$  satisfying  $\mathcal{A}\alpha$  cannot be accepted by a CFM.*

*Proof.* Let  $\alpha$  be the local formula from Lemma 7.1. Towards a contradiction, assume  $\mathcal{A}$  is a CFM such that, for any pair  $(M, c)$ , we have  $(M, c) \models A\alpha$  iff  $(M, c)$  is accepted by  $\mathcal{A}$ . In particular,  $\mathcal{A}$  accepts all pairs  $(M_m, c)$  satisfying (1) and (2) from Lemma 7.2. Hence there exists some pair  $(M_m, c)$  that is accepted by  $\mathcal{A}$ , satisfies (1), and violates (2). But now, by Lemma 7.1 again,  $(M_m, c) \models \neg A\alpha$ , contradicting our assumption on  $\mathcal{A}$ .

Using a new process 2, one can encode the mapping  $c$  by additional messages from processes 0 and 1 to process 2.  $\square$

## 8 Open questions

The semantics of every PDL formula  $\varphi$  is the behavior of a CFM  $\mathcal{A}$ . Hence any PDL formula is equivalent to some formula from existential monadic second order, but a precise description of the expressive power of PDL is not known. Because of quantification over paths, it cannot be captured by first-order logic. On the other hand, PDL is closed under negation, hence PDL is a proper fragment of existential monadic second order logic.

The decidability of the model checking problem for CFMs against MSO-formulas was shown in [GKM06] for existentially  $B$ -bounded MSCs. For compositional MSCs (a mechanism for the description of sets of MSCs that is similar but more general than HMSCs) and MSO, the decidability of the model checking problem was established in [MM01]. Since the logic iPDL, i.e., PDL with intersection, can be translated effectively into an MSO-formula, the model checking problem is decidable for iPDL. However, the complexity of MSO model checking is non-elementary. Therefore, we would like to know if we can do any better for iPDL.

In PDL, we can express properties of the past and of the future of an event by taking either a backward- or a forward-path in the graph of the MSC. We are not allowed to speak about a zig-zag-path where e.g. a mixed use of  $\text{proc}$  and  $\text{proc}^{-1}$  would be possible. It is an open question whether formulas of such a “mixed PDL” could be transformed to CFMs.

## References

- [APP95] R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS 1995)*, pages 90–100, Washington, DC, USA, 1995. IEEE Computer Society.
- [BK06] B. Bollig and D. Kuske. Distributed Muller automata and logics. Research Report LSV-06-11, Laboratoire Spécification et Vérification, ENS Cachan, France, 2006.
- [BL06] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006.
- [BZ83] D. Brand and P. Zafropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2), 1983.
- [Cho74] Y. Choueka. Theories of automata on  $\omega$ -tapes: a simplified approach. *Journal of Computer and System Sciences*, 8:117–141, 1974.
- [FL79] M.J. Fischer and R.E. Ladner. Propositional Dynamic Logic of regular programs. *J. Comput. System Sci.*, 18(2):194–211, 1979.
- [GK03] P. Gastin and D. Kuske. Satisfiability and model checking for MSO-definable temporal logics are in PSPACE. In *CONCUR’03*, Lecture Notes in Comp. Science vol. 2761, pages 222–236. Springer, 2003.
- [GK05] P. Gastin and D. Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. In *CONCUR’05*, Lecture Notes in Comp. Science vol. 3653, pages 533–547. Springer, 2005.
- [GKM06] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204:920–956, 2006.
- [GMSZ02] B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state high-level MSCs: model-checking and realizability. In *ICALP’02*, Lecture Notes in Comp. Science vol. 2380, pages 657–668. Springer, 2002.

- [HT97] J. G. Henriksen and P. S. Thiagarajan. A product version of dynamic linear time temporal logic. In Antoni Mazurkiewicz and Józef Winkowski, editors, *8th International International Conference on Concurrency Theory (CONCUR 1997)*, volume 1243, pages 45–58, Warsaw, Poland, 1–4 1997. Springer-Verlag.
- [HT99] J. G. Henriksen and P. S. Thiagarajan. Dynamic linear time temporal logic. *Ann. Pure Appl. Logic*, 96(1-3):187–207, 1999.
- [ITU96] ITU-TS Recommendation Z.120: Message Sequence Chart 1996 (MSC96), 1996.
- [MM01] P. Madhusudan and B. Meenakshi. Beyond message sequence graphs. In *FSTTCS 2001*, Lecture Notes in Comp. Science vol. 2245, pages 256–267. Springer, 2001.
- [MR00] B. Meenakshi and R. Ramanujam. Reasoning about message passing in finite state environments. In *Proc. of ICALP 2000*, volume 1853 of *Lecture Notes in Computer Science*, pages 487–498. Springer, 2000.
- [MR04] B. Meenakshi and R. Ramanujam. Reasoning about layered message passing systems. *Computer Languages, Systems, and Structures*, 30(3-4):529–554, 2004.
- [Pel00] D. Peled. Specification and verification of message sequence charts. In *Formal Techniques for Distributed System Development, FORTE/PSTV 2000*, volume 183 of *IFIP Conference Proceedings*, pages 139–154. Kluwer, 2000.
- [Tho90] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 133–191. Elsevier Science Publ. B.V., 1990.