Safety Verification of Communicating One-Counter Machines

Alexander Heußner[1], Tristan Le Gall[2], and Grégoire Sutre[3]

1 **Université Libre de Bruxelles, Brussels, Belgium**
2 **CEA, LIST, DILS/LMeASI, Gif-sur-Yvette, France**
3 **Univ. Bordeaux & CNRS, LaBRI, UMR 5800, Talence, France**

Abstract

In order to verify protocols that tag messages with integer values, we investigate the decidability of the reachability problem for systems of communicating one-counter machines. These systems consist of local one-counter machines that asynchronously communicate by exchanging the value of their counters via, a priori unbounded, FIFO channels. This model extends communicating finite-state machines (CFSM) by infinite-state local processes and an infinite message alphabet. The main result of the paper is a complete characterization of the communication topologies that have a solvable reachability question. As already CFSM exclude the possibility of automatic verification in presence of mutual communication, we also consider an under-approximative approach to the reachability problem, based on rendezvous synchronization.

Keywords and phrases counter machines, FIFO channels, infinite message alphabet, reachability

1 Introduction

Today's ubiquity of systems that are distributed over a network and that are vital even in sensitive areas renders the formal verification of distributed systems to one of the most challenging and imperative problems in computer science. Such systems consist of several processes that asynchronously exchange data over a network topology. A well-established model for asynchronous distributed systems is the combination of local finite-state machines with point-to-point, unbounded FIFO queues that pass messages from a finite alphabet—known as *communicating finite-state machines* (CFSM). CFSM laid the foundation for a family of infinite-state models that is parametrized by the computational power of the local machines, such as communicating Petri nets [10] and communicating pushdown processes [16, 14].

However, basic safety verification questions, like reachability, are known to be undecidable for CFSM already on simple topologies [6, 19]. One important line of current research is the influence of the underlying communication topology to these verification questions, especially in combination with different restrictions on the interplay between communication and the local machine's power [16, 7, 14]. In this paper, we extend this research towards the verification of communicating machines that locally use counters and can exchange these via message passing, thus introducing two additional sources of infinity to CFSM's unbounded channels. Infinite message alphabets are demanded in practice to model protocols based on (*a priori* unbounded) sequence numbers.

Motivating Example. A simple sliding window protocol is depicted in Figure 1. A client sends a sequence number (ignoring the additional data part) to the server. The latter compares the received sequence number with the one it expects, and either advances to the next message on equality, demands the client to resend a message if it is greater, or fails if the sequence number was already received. Deciding whether we can check if such a protocol is "correct", i.e., whether the `error` state is reachable, is the main subject of this paper.

FSTTCS.
Editors: N.N; pp. 1–23

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

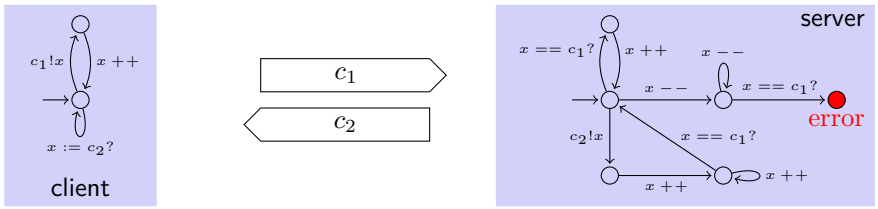


Figure 1 Motivating example: simple sliding window protocol.

Contributions. We present the formal model of *systems of communicating one-counter machines*. This model is parametrized by a communication topology, specifying point-to-point Fifo channels between processes. Processes are one-counter machines that can send or receive the contents of their local counter. We consider an extension of one-counter machines where tests are not limited to zero-tests $\mathtt{x} = 0$, but can be any unary Presburger predicate $\varphi(\mathtt{x})$. Channels are a priori unbounded, and messages are natural numbers. Different ways of relating these messages to the machine's local counters are investigated. As our main result, we establish a complete classification of the topologies over which the reachability problem for systems of communicating one-counter machines is decidable. The underlying proof relies, on the one hand, on a reduction from the well-known undecidability of the reachability problem for two-counter Minsky machines. On the other hand, we use a reduction technique that inductively combines one-counter machines along a hierarchical order, which is based on the topology. This way, the reachability problem is reduced to the case of two-processes that are connected by one channel. We show that the reachability problem is decidable in this setting. The proof is based on a characterization of one-counter reachability relations in terms of binary Presburger predicates. This characterization entails that one-counter reachability relations are effectively closed under intersection. As the undecidable topologies include cyclic architectures, that nevertheless are important in practice (see previous motivating example), we consider an under-approximative approach based on *eager* runs, i.e., runs where a send action is directly followed by its reception. We characterize the strongly-connected topologies that have a decidable eager-reachability problem. In particular, the topology of our motivating example, which is a cycle of length two, allows to decide the verification problem.

Related Works. The basic undecidability result for Cfsm [6] is the corner stone for most ongoing research on models based local machines that communicate over Fifo channels. Prominent approaches to regain decidability for reachability/safety are restrictions on the size of the channels or the message alphabets (already in [6, 19]), as well as the focus on lossy channel systems [9, 1]. Recent research dealt with the influence of the underlying topology on decidability questions. Systems mixing lossy and perfect channels are considered in [7], but these remain with finite local processes. Communicating pushdown machines are considered in [16, 14] while discussing a typing of channel ends that forces the decoupling of pushdown and channel actions. Our channels are also typed but introduce the aspect of decoupling differently by losing the value of a local counter when communicating. Note, nonetheless, that all these works only consider *finite* message alphabets.

Cfsm-style systems with *infinite* message alphabets were discussed in [17], but this work focused on the definition of a static analysis technique, and thus the practical implementation of verification algorithms. Also closely related are data words and their different underlying automata models that rely on an infinite input/output alphabet and local registers [3, 4]. However, these automata only allow to use an equality test on the infinite data alphabet and not to modify and test registers like counters do.

Counter machines are a classical formalism in computer science [18]. Besides the *two-counter* (Minsky) machines, which are Turing-complete, the verification of *one-counter* automata has gained a renewed interest recently [8, 13, 12, 2]. Using one-counter automata with Presburger tests also appears in [5], yet only as symbolic representation of reachability sets and not as operational model for the underlying programs.

Outline. We introduce systems of communicating one-counter machines in Section 2. Section 3 presents our main result: the characterization of communication topologies that have a solvable reachability question. The proof of the positive case is provided in Section 4. Section 5 presents preliminary results on the decidability of the reachability question when we only consider eager runs. Some conclusions and perspectives are given in Section 6.

2 Systems of Communicating One-Counter Machines

Given a (possibly infinite) alphabet M, let M^* denote the set of all finite *words* over M, $\varepsilon \in M^*$ the *empty* word, and $u \cdot v \in M^*$ the *concatenation* of two words $u, v \in M^*$. For a set of values X and a finite set of indices I, let X^I denote the set of all mappings from I to X. Such mappings may be interpreted as I-indexed X-valued *vectors*. Let x^i denote the i-th component of a vector $\boldsymbol{x} \in X^I$. Two constant vectors are introduced, for convenience: $\mathbf{0} \in \mathbb{N}^I$, which maps every index to 0, and $\boldsymbol{\varepsilon} \in (M^*)^I$, which maps every index to ε.

Communication Topologies. In our framework, channels are point-to-point. Each channel c has a source endpoint $\mathsf{src}(c)$, and a destination endpoint $\mathsf{dst}(c)$. These endpoints are pairs $(p, *)$ where p is the process communicating at this endpoint, and $* \in \{\bullet, \circ\}$ is the communication *type* of p on the channel c, written $\mathsf{typ}(p, c)$.

▶ **Definition 2.1.** A *topology* is a quadruple $\mathcal{T} = \langle P, C, \mathsf{src}, \mathsf{dst} \rangle$ where P is a finite, non-empty set of *processes*, C is a finite, possibly empty set of *channels*, $\mathsf{src} : C \to P \times \{\bullet, \circ\}$ is a *source* mapping, and $\mathsf{dst} : C \to P \times \{\bullet, \circ\}$ is a *destination* mapping.

For better readability, we slightly abuse notation by identifying an endpoint $(p, *)$ with its process p or its type $*$, depending on the context. For instance, we write $\mathsf{src}(c) = p$ instead of $\mathsf{src}(c) = (p, *)$ for some $* \in \{\bullet, \circ\}$. Given a process $p \in P$, we let $C(p)$ denote the set of all channels with source or target p. Formally, $C(p) = \{c \in C \mid \mathsf{src}(c) = p \vee \mathsf{dst}(c) = p\}$.

We use the communication type of a channel to express two types of interrelation between the value of the message and the local counter value of a machine using the channel's endpoint before and after the communication action. We assert that $\bullet$ demands a less restrictive policy than $\circ$. This difference is formalized in the semantics introduced subsequently.

For each channel $c \in C$, we let $\overset{c}{\rightarrowtail}$ denote the binary relation on the set of processes P defined by $p \overset{c}{\rightarrowtail} q$ if $p = \mathsf{src}(c)$ and $q = \mathsf{dst}(c)$. Naturally, any topology may be viewed as the labeled *directed* graph $(P, \{\overset{c}{\rightarrowtail}\}_{c \in C})$. We assume some familiarity with classical notions on directed graphs, such as weak connectedness, strong connectedness, leaf nodes, etc. We also introduce the undirected binary relation $\overset{c}{\leftrightarrow}$, defined by $p \overset{c}{\leftrightarrow} q$ if $p \overset{c}{\rightarrowtail} q$ or $p \overset{c}{\leftarrowtail} q$. An *undirected path* in $\mathcal{T}$ is an alternating sequence $(p_0, c_1, p_1, \ldots, c_n, p_n)$, of processes $p_i \in P$ and channels $c_i \in C$, such that $p_{i-1} \overset{c_i}{\leftrightarrow} p_i$ for all $i \in \{1, \ldots, n\}$. Moreover, the undirected path is called *simple* when $p_0, \ldots, p_n$ are distinct. A *simple undirected cycle* in $\mathcal{T}$ is an undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n)$, with $n \geq 1$, such that $p_1, \ldots, p_n$ are distinct, $c_1, \ldots, c_n$ are distinct, and $p_0 = p_n$. A *simple undirected shunt* in $\mathcal{T}$ is a simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n)$, with $n \geq 2$, such that $\mathsf{typ}(p_0, c_1) = \mathsf{typ}(p_n, c_n) = \bullet$.

▶ **Definition 2.2.** Let $\mathcal{T}$ be a topology. $\mathcal{T}$ is called *cycle-free* if it contains no simple undirected cycle. $\mathcal{T}$ is called *shunt-free* if it contains no simple undirected shunt.

▶ Remark. Our notion of shunt is close to the *confluence* criterion presented in [14] for communicating pushdown processes. Simply put, confluence allows to synchronize two pushdown stacks, and a shunt permits to synchronize two counters, as will be seen later. However, shunts require at least one additional, intermediary process whereas confluence can be established directly between two processes. In our case, the topology $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{q}$ with channel endpoints of type $\bullet$ is shunt-free, and will be shown to have a decidable reachability problem.

Systems of Communicating One-Counter Machines. Classically, one-counter machines are finite-state automata, equipped with a counter, represented by a variable $\mathtt{x}$, that holds a non-negative integer value. The counter is initially set to zero, and can be incremented, decremented (provided that it remains non-negative), and tested for zero. In this paper, we consider an extension of counter machines where tests are not limited to zero-tests $\mathtt{x} = 0$, but can be any unary Presburger predicate $\varphi(\mathtt{x})$. Such Presburger tests do not increase the expressive power of one-counter machines in terms of recognized languages [5]. We will show in the next section that the same property holds for their binary reachability relations. Presburger tests will be handy to merge several communicating one-counter machines in a single communicating one-counter machine.

Recall that *Presbuger arithmetic* is the first-order theory of the natural numbers with addition. A n-ary *Presburger predicate* is a Presburger formula φ with exactly n free variables. As usual, we write $\varphi(\mathtt{x}_1, \ldots, \mathtt{x}_n)$ to indicate that $\mathtt{x}_1, \ldots, \mathtt{x}_n$ are the free variables of φ. We let $\mathcal{P}_n$ denote the set of all n-ary Presburger predicates.

▶ **Definition 2.3.** A *system of communicating one-counter machines* is a pair $\mathcal{S} = \langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ where $\mathcal{T}$ is a topology and, for each process p in P, $\mathcal{M}^p$ is a quintuple $\mathcal{M}^p = \langle S^p, I^p, F^p, A^p, \Delta^p \rangle$, called a *communicating one-counter machine*, where

- S^p is a finite set of *states*,
- $I^p, F^p \subseteq S^p$ are subsets of *initial states* and *final states*,
- $A^p \subseteq A_{\text{cnt}} \cup A^p_{\text{com}}$ is a finite set of *actions*, where
$$A_{\text{cnt}} = \{\mathtt{add}(k) \mid k \in \mathbb{Z}\} \cup \{\mathtt{test}(\varphi) \mid \varphi \in \mathcal{P}_1\}$$
$$A^p_{\text{com}} = \{c\,! \mid c \in C \wedge \mathsf{src}(c) = p\} \cup \{c\,? \mid c \in C \wedge \mathsf{dst}(c) = p\}$$
- $\Delta^p \subseteq S^p \times A^p \times S^p$ is a finite set of transition *rules*.

We give the operational semantics $[\![\mathcal{S}]\!]$ of a system of communicating one-counter machines $\mathcal{S}$ as labeled transition system. A *configuration* of $[\![\mathcal{S}]\!]$ is triple $\sigma = (\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{w})$ where $\boldsymbol{s}$ maps each process p to a state in S^p, $\boldsymbol{x}$ maps each process p to a counter value in $\mathbb{N}$, and $\boldsymbol{w}$ maps each channel c to a word over the set of natural numbers. Formally, the set of configurations of $[\![\mathcal{S}]\!]$ is $(\prod_{p \in P} S^p) \times \mathbb{N}^P \times (\mathbb{N}^*)^C$. An *initial configuration* is a configuration $(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{w})$ such that $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{w} = \boldsymbol{\varepsilon}$, and $s^p \in I^p$ for all $p \in P$. Analogously, a *final configuration* is a configuration $(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{w})$ such that $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{w} = \boldsymbol{\varepsilon}$, and $s^p \in F^p$ for all $p \in P$. The *transition relation* of $[\![\mathcal{S}]\!]$, written $\rightarrow$, is the set of all triples (σ_1, a, σ_2), where $\sigma_1 = (\boldsymbol{s}_1, \boldsymbol{x}_1, \boldsymbol{w}_1)$ and $\sigma_2 = (\boldsymbol{s}_2, \boldsymbol{x}_2, \boldsymbol{w}_2)$ are configurations, and a is an action in A^p, for some $p \in P$, satisfying the following conditions:

- $(s^p_1, a, s^p_2) \in \Delta^p$ and $s^q_1 = s^q_2$ for all $q \in P$ with $q \neq p$,
- if $a = \mathtt{add}(k)$ then $x^p_2 = x^p_1 + k$, $x^q_1 = x^q_2$ for all $q \in P$ with $q \neq p$, and $\boldsymbol{w}_1 = \boldsymbol{w}_2$,
- if $a = \mathtt{test}(\varphi(\mathtt{x}))$ then the valuation $\{\mathtt{x} \mapsto x^p_1\}$ satisfies $\varphi(\mathtt{x})$, $\boldsymbol{x}_1 = \boldsymbol{x}_2$ and $\boldsymbol{w}_1 = \boldsymbol{w}_2$,
- if $a = c\,!$, then

- $w_2^c = w_1^c \cdot x_1^p$ and $w_1^d = w_2^d$ for all $d \in C$ with $d \neq c$, and
- if $\mathsf{src}(c) = \bullet$ then $\boldsymbol{x}_1 = \boldsymbol{x}_2$; otherwise $x_1^q = x_2^q$ for all $q \in P$ with $q \neq p$.

- if $a = c\,?$, then
  - $w_1^c = x_2^p \cdot w_2^c$ and $w_1^d = w_2^d$ for all $d \in C$ with $d \neq c$, and
  - if $\mathsf{src}(c) = \bullet$ then $\boldsymbol{x}_1 = \boldsymbol{x}_2$; otherwise $x_1^q = x_2^q$ for all $q \in P$ with $q \neq p$.

For readability, we write $\sigma_1 \xrightarrow{a} \sigma_2$ in place of $(\sigma_1, a, \sigma_2) \in \rightarrow$. Notice that we do not explicitly index actions by the process that fires them, but we assert that one implicitly knows which process moves on each transition.

The semantics of counter operations $\mathtt{add}(k)$ and $\mathtt{test}(\varphi)$ is the usual one. A send or receive action on a channel appends or removes a message in $\mathbb{N}$, as one would expect. However, there are additional restrictions on the interplay of the communicated message and the local counter. If the endpoint of the channel has type $\bullet$, the message must equal the value of the counter *before* and *after* the action. So the value of the counter is not modified by a communication on this endpoint. On the contrary, if the endpoint has type $\circ$, then the local counter value is "lost" by a communication on this endpoint:

- an emission transfers the value of the counter from the process to the channel; the counter is non-deterministically set to an arbitrary value after the emission.
- a reception transfers the message from the channel to the local counter; the behavior mirrors that of an emission.

A *run* of $[\![\mathcal{S}]\!]$ is a finite, alternating sequence $\rho = (\sigma_0, a_1, \sigma_1, \ldots, a_n, \sigma_n)$ of configurations σ_i and actions a_i, satisfying $\sigma_{i-1} \xrightarrow{a_i} \sigma_i$ for all i. We say that ρ is a run *from* σ_0 *to* σ_n, and, abusing notation, we shortly write $\rho = \sigma_0 \xrightarrow{*} \sigma_n$. The *length* of ρ is n, and is denoted by $|\rho|$. A run of length zero consists of a single configuration. A *full run* of $[\![\mathcal{S}]\!]$ is a run from an initial configuration to a final configuration.

Exchange of Messages from a Finite Alphabet. On the contrary to classical *communicating finite-state machines* (Cfsm), communicating one-counter machines cannot (directly) send or receive messages from an arbitrary finite alphabet M. However, they are able to perform these actions indirectly, as follows. Assume, without loss of generality, that M is a finite set of natural numbers. Sending a message $m \in M$ on a channel c, like a Cfsm would, simply amounts to setting the local counter to m, and performing an emission on c. Receiving a message $m \in M$ from a channel c, like a Cfsm would, is done by performing a reception from c, and checking that the received message is m. To realize this check, the machine

- simply sets its counter to m before the reception, for an endpoint with type $\bullet$,
- or checks that the counter equals m after the reception, for an endpoint with type $\circ$.

Note that in this simulation of Cfsm-style communications, the counter is forcibly set to the (bounded) value corresponding to the message being exchanged, even for endpoints with type $\bullet$. We show, in the next section, another simulation of Cfsm-style communications where one of the two peers is able to retain the value of its counter.

3 A Characterization of Topologies with Solvable Reachability

We investigate the power of systems of communicating one-counter machines with regard to their communication topology. Therefore, we introduce the reachability problem parametrized by a given topology. Recall that a full run of $[\![\mathcal{S}]\!]$ is a run from an initial configuration to a final configuration.

▶ **Definition 3.1.** Given a topology $\mathcal{T}$, the *reachability problem* for systems of communicating one-counter machines with topology $\mathcal{T}$, denoted by RP-SC1CM($\mathcal{T}$), is defined as follows:
Input: a system of communicating one-counter machines $\mathcal{S}$ with topology $\mathcal{T}$,
Output: whether there exists a full run in $[\![\mathcal{S}]\!]$.

The main result of the paper is a complete classification of the topologies that have a solvable reachability problem. We observe that, in absence of shunts, systems of communicating one-counter machines are still more expressive than CFSM, but their reachability problems are decidable for the same topologies, namely, cycle-free topologies [19].

▶ **Theorem 3.2.** *Given a topology $\mathcal{T}$,* RP-SC1CM($\mathcal{T}$) *is decidable if and only if $\mathcal{T}$ is cycle-free and shunt-free.*

The proof of the theorem is presented at the end of this section for the "only if" direction, and in Section 4 for the "if" direction. Before that, let us provide a decomposition, in terms of undirected trees, i.e., of topologies that are cycle-free and shunt-free. Observe that a weakly-connected topology is cycle-free if and only if there is a unique simple undirected path between every two processes. The proof of the following proposition is given in Appendix A.

▶ **Proposition 3.3.** *Let $\mathcal{T}$ be a weakly-connected topology with at least two processes. If $\mathcal{T}$ is cycle-free and shunt-free, then there are two distinct processes r, r', with $r \overset{c}{\leftrightarrow} r'$ for some channel c, such that, for every simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ with $p_0 \in \{r, r'\}$ and $q \notin \{r, r'\}$, the process q has type $\circ$ on the channel d.*

An example illustrating the proposition is provided in Figure 2(a). This weakly-connected topology is cycle-free and shunt-free. Therefore, its underlying undirected graph is a tree. The processes r and r' may be seen as two "roots", connected by a channel. All other processes are descendants of these two "roots", and have type $\circ$ on the channel (input or output) that leads to the root, as required by Proposition 3.3. Note, however, that r and r' are allowed to have type $\bullet$ on all channels. Recall that a process with type $\circ$ on a channel "loses" the value of its counter when it communicates over this channel. On the contrary, no loss of information occurs with type $\bullet$. But an endpoint with type $\bullet$ can simulate an endpoint with type $\circ$, by artificially "losing" the value of the local counter. We formalize this property by introducing the partial order $\sqsubseteq$ on $\{\circ, \bullet\}$ defined by $\circ \sqsubset \bullet$. This partial order is extended to endpoints in the natural way: $(p, *) \sqsubseteq (p', *')$ if $p = p'$ and $* \sqsubseteq *'$. Given two topologies $\mathcal{T} = \langle P_\mathcal{T}, C_\mathcal{T}, \mathsf{src}_\mathcal{T}, \mathsf{dst}_\mathcal{T} \rangle$ and $\mathcal{U} = \langle P_\mathcal{U}, C_\mathcal{U}, \mathsf{src}_\mathcal{U}, \mathsf{dst}_\mathcal{U} \rangle$, we say that $\mathcal{U}$ is a *sub-topology* of $\mathcal{T}$ if $P_\mathcal{U} \subseteq P_\mathcal{T}$, $C_\mathcal{U} \subseteq C_\mathcal{T}$, and, for every channel $c \in C_\mathcal{U}$, it holds that $\mathsf{src}_\mathcal{U}(c) \sqsubseteq \mathsf{src}_\mathcal{T}(c)$ and $\mathsf{dst}_\mathcal{U}(c) \sqsubseteq \mathsf{dst}_\mathcal{T}(c)$. As one would expect, sub-topologies have an easier reachability problem. The formal proof of the following result is provided in Appendix A.

▶ **Proposition 3.4.** *For every topology $\mathcal{T}$ and for every sub-topology $\mathcal{U}$ of $\mathcal{T}$,* RP-SC1CM($\mathcal{U}$) *is reducible to* RP-SC1CM($\mathcal{T}$).

Cycle-freeness and Shunt-freeness of Decidable Topologies. In the remainder of this section, we prove the "only if" direction of Theorem 3.2, namely that RP-SC1CM($\mathcal{T}$) is undecidable if $\mathcal{T}$ contains a simple undirected cycle or a simple undirected shunt. As seen in Section 2, systems of communicating one-counter machines can simulate CFSM, and the simulation preserves the topology. Moreover, the reachability problem for CFSM with topology $\mathcal{T}$ is known to be undecidable if $\mathcal{T}$ contains a simple undirected cycle [19, 16]. It follows that RP-SC1CM($\mathcal{T}$) is undecidable if $\mathcal{T}$ contains a simple undirected cycle. The following lemma completes the proof of the "only if" direction of Theorem 3.2.

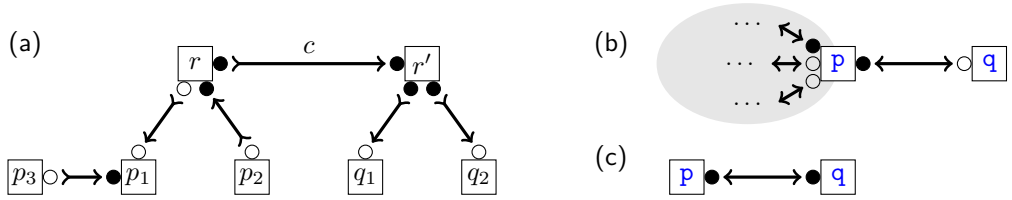


Figure 2 Topologies: (a) weakly connected cycle-free and shunt-free topology, (b) topology containing a leaf process q with type ∘ on its pendant channel, (c) decidable two-processes case.

▶ **Lemma 3.5.** *For every topology $\mathcal{T}$ containing a simple undirected shunt,* Rp-Sc1cm($\mathcal{T}$) *is undecidable.*

A detailed proof of the lemma is provided in Appendix A.1. We explain the main ideas of the proof on the tree topology $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{r} \overset{\mathtt{d}}{\leftarrowtail} \mathtt{q}$ where r has type ∘ on channels c and d, p has type • on c and q has type • on d. Let us call this topology $\mathcal{T}$. Notice that (p, c, r, d, q) is a simple undirected shunt. We show that the reachability problem for two-counter (Minsky) machines, which is known to be undecidable [18], is reducible to Rp-Sc1cm($\mathcal{T}$). Given a two-counter machine $\mathcal{M}$, one counter, say x, is maintained by p, and the other, say y, is maintained by q. Both processes p and q run a copy of $\mathcal{M}$, but they internalize (as add(0) actions) the counter actions of $\mathcal{M}$ that do not involve their counter. We only need to make sure that p and q take the same control path of $\mathcal{M}$. To this end, p and q send to r the transition rules that they traverse, and r checks that these rules are the same. However, p and q must not lose the value of their counter when communicating with r. So the simulation of Cfsm presented in Section 2 cannot be used. Instead, p and q encode the transition rules within the counter value itself, send it to r, and let r decode and check this information.

Assume that $\mathcal{M}$ contains $K > 0$ transition rules, encoded as $0, \ldots, K-1$. Instead of storing the values x and y of x and y in their local counters, p and q store $K \cdot x$ and $K \cdot y$, respectively. So, increments and decrements in $\mathcal{M}$ are multiplied by the constant K in p and q. On the sender side, when p or q takes a transition rule encoded by $\delta \in \{0, \ldots, K-1\}$, it increments its counter by δ, sends it to r, and decrements its counter by δ to restore its value. On the receiver side, when r performs a c? action, its counter is set to the message $m = \delta + (K \cdot x)$ sent by p, and it extracts the transition rule δ by computing (m mod K).

The simulation guarantees that the two-counter machine has a full run if and only if the constructed system of communicating one-counter machines, with topology $\mathcal{T}$, has a full run. It follows that Rp-Sc1cm($\mathcal{T}$) is undecidable. Note that, by Proposition 3.4, the reachability problem Rp-Sc1cm($\mathcal{T}$) would also be undecidable (and even more so) if r had type • instead of ∘ on its output channels.

▶ Remark. We need at least one intermediary process r between p and q, to decode and check their messages. Indeed, direct communications between p and q would synchronize their local counters, thus making it impossible to maintain two counters.

4 Decidability of Cycle-free and Shunt-free Topologies

This section is devoted to the proof of the "if" direction of Theorem 3.2, namely that Rp-Sc1cm($\mathcal{T}$) is decidable if $\mathcal{T}$ is cycle-free and shunt-free. Without loss generality, we only consider weakly-connected topologies. The proof comprises three independent parts. Firstly, we provide a characterization, in terms of Presburger predicates, of reachability relations of one-counter machines. Secondly, we show that any leaf process with type ∘ on its pendant

channel may be merged into its parent, thereby reducing the size of the topology. Iterating this reduction leads to a topology with only two processes and one channel. We show, in the third part, that Rp-Sc1cm($\mathcal{T}$) is decidable for such topologies.

Counter reachability relations of one-counter machines. A *one-counter machine* is a communicating one-counter machine $\mathcal{M} = \langle S, I, F, A, \Delta \rangle$ with no communication action, i.e., $A \subseteq A_{\text{cnt}}$. To fit our framework, we identify $\mathcal{M}$ with the system $\langle \mathcal{U}, (\mathcal{M}^p)_{p \in \{\mathtt{p}\}} \rangle$ of communicating one-counter machines, where $\mathcal{U} = \langle \{\mathtt{p}\}, \emptyset, \mathsf{src}, \mathsf{dst} \rangle$ is the topology with a single process $\mathtt{p}$ and no channel. We let Rp-1cm denote the reachability problem for one-counter machines, formally Rp-1cm = Rp-Sc1cm($\mathcal{U}$). It is well-known that Rp-1cm is decidable (see, e.g., [13]). In the next subsections, we show that, under certain conditions, two processes can be merged in a single "product" process (with only one counter). To do so, we summarize the behavior of a process between each communication action. This subsection is devoted to the characterization and computation of these summaries. Proofs rely on transformations of unary Presburger predicates, and can be found in Appendix B.1.

Let $\mathcal{M} = \langle S, I, F, A, \Delta \rangle$ be a one-counter machine. The *counter reachability relation* of $\mathcal{M}$ is the set of all pairs $(x, y) \in \mathbb{N} \times \mathbb{N}$ such that, for some $s \in I$ and $t \in F$, there exists a run from (s, x) to (t, y). To characterize counter reachability relations, we introduce the following class of binary Presburger predicates. We consider two distinguished Presburger variables $\mathtt{x}$ and $\mathtt{y}$. In short, one-counter Presburger predicates can express properties of $\mathtt{x}$, of $\mathtt{y}$, and of their differences $\mathtt{x} - \mathtt{y}$ and $\mathtt{y} - \mathtt{x}$. Formally, the class of *one-counter Presburger predicates* is generated by the grammar:

$$\psi \quad ::= \quad \varphi(\mathtt{x}) \mid \varphi(\mathtt{y}) \mid \exists z \cdot (\mathtt{x} = \mathtt{y} + z \wedge \varphi(z)) \mid \exists z \cdot (\mathtt{y} = \mathtt{x} + z \wedge \varphi(z)) \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{tt} \mid \mathtt{ff}$$

where φ ranges over the set $\mathcal{P}_1$ of unary Presburger predicates. The binary relation *defined* by a one-counter Presburger predicate ψ is the set of all pairs $(x, y) \in \mathbb{N} \times \mathbb{N}$ such that the valuation $\{\mathtt{x} \mapsto x, \mathtt{y} \mapsto y\}$ satisfies ψ.

We first show that counter reachability relations are definable by one-counter Presburger predicates, for the class of one-counter machines with zero-tests only. Formally, a one-counter machine $\mathcal{M} = \langle S, I, F, A, \Delta \rangle$ is called *basic* if $A \subseteq \{\mathtt{add}(k) \mid k \in \mathbb{Z}\} \cup \{\mathtt{test}(\mathtt{x} = 0)\}$.

▶ **Lemma 4.1.** *For every basic one-counter machine $\mathcal{M}$, the counter reachability relation of $\mathcal{M}$ is defined by a one-counter Presburger predicate.*

However, the converse of the lemma does not hold. Consider, for instance, the one-counter Presburger predicate $\psi = \exists \mathtt{k} \cdot (\mathtt{x} = \mathtt{k} + \mathtt{k}) \wedge (\mathtt{x} = \mathtt{y})$. In a basic one-counter machine, it is not possible to check that a given, a priori unknown value $\mathtt{x}$ is even without "losing" this value. In fact, the proof of Lemma 4.1 shows that the binary relation defined by ψ cannot be the counter reachability of a basic one-counter machine. We need the additional expressive power stemming from Presburger tests.

We now show that counter reachability relations are precisely the relations definable by one-counter Presburger predicates. This entails, in particular, that counter reachability relations are closed under intersection. We will use this property in the proof of Lemma 4.4.

▶ **Theorem 4.2.** *For every binary relation $R \subseteq \mathbb{N} \times \mathbb{N}$, the two following assertions are equivalent:*

- *R is the counter reachability relation of a one-counter machine,*
- *R is defined by a one-counter Presburger predicate.*

▶ Remark. The proof of Theorem 4.2 is constructive, in the sense that a one-counter Presburger predicate is computable from a given one-counter machine, and vice versa.

Merging leaf processes. We show how to reduce the number of processes in a system of communicating one-counter machines, by merging a leaf process with type $\circ$ on its pendant channel into its parent. Let $\mathcal{U} = \langle P_{\mathcal{U}}, C_{\mathcal{U}}, \mathsf{src}_{\mathcal{U}}, \mathsf{dst}_{\mathcal{U}} \rangle$ be a topology, and select a distinguished process $\mathtt{p}$ in $P_{\mathcal{U}}$. We add to the topology a new process $\mathtt{q} \notin P_{\mathcal{U}}$ and a new channel $\mathtt{c} \notin C_{\mathcal{U}}$ between $\mathtt{p}$ and $\mathtt{q}$. Formally, we consider any topology $\mathcal{T} = \langle P, C, \mathsf{src}, \mathsf{dst} \rangle$ with set of processes $P = P_{\mathcal{U}} \cup \{\mathtt{q}\}$ and set of channels $C = C_{\mathcal{U}} \cup \{\mathtt{c}\}$, whose source and target mappings coincide with those of $\mathcal{U}$ on C, and such that $\mathtt{p} \overset{\mathtt{c}}{\leftrightarrow} \mathtt{q}$. Observe that $C(\mathtt{q}) = \{\mathtt{c}\}$, hence, $\mathtt{q}$ is a leaf process with pendant channel $\mathtt{c}$. The topology $\mathcal{T}$ is depicted on Figure 2(b).

▶ **Lemma 4.3.** *If* $\mathtt{p}$ *has type* $\bullet$ *on* $\mathtt{c}$ *and* $\mathtt{q}$ *has type* $\circ$ *on* $\mathtt{c}$ *then* Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-Sc1cm($\mathcal{U}$).

A formal proof of the lemma is provided in Appendix B.2. Let us explain the main ideas of the proof. Assume that $\mathtt{c}$ is directed as $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{q}$. Consider a system of communicating one-counter machines $\mathcal{S} = \langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$. To simulate $\mathcal{S}$ over the topology $\mathcal{U}$, we merge processes $\mathtt{p}$ and $\mathtt{q}$ in a single "product" process $\widehat{\mathtt{p}}$. So, the communicating one-counter machines $\mathcal{M}^p$ are kept unchanged for all processes in $p \in P \setminus \{\mathtt{p}, \mathtt{q}\}$. But the process $\widehat{\mathtt{p}}$ must simulate both processes $\mathtt{p}$ and $\mathtt{q}$, as well as the channel $\mathtt{c}$ in-between. We choose a specific interleaving of $\mathtt{p}$ and $\mathtt{q}$ where $\mathtt{c}$ is almost always empty, and such that $\widehat{\mathtt{p}}$, which has a single counter, is able to retain both $\mathtt{p}$'s counter and $\mathtt{q}$'s counter.

In essence, $\widehat{\mathtt{p}}$ behaves as $\mathtt{p}$, but also maintains, in its state, the local state of $\mathtt{q}$ as well as an abstraction of $\mathtt{q}$'s counter. We abstract $\mathtt{q}$'s counter by the set $\{0, \bot, =\}$, where 0 means zero, $\bot$ means some unknown value, and $=$ means that $\mathtt{q}$'s counter holds the same value as $\mathtt{p}$'s counter. Furthermore, the process $\mathtt{q}$ is always scheduled first. Since $\mathtt{c}$ is the only channel with source or target $\mathtt{q}$, this means, in particular, that every reception by $\mathtt{q}$ from $\mathtt{c}$ occurs immediately after the matching emission by $\mathtt{p}$ on $\mathtt{c}$. When $\widehat{\mathtt{p}}$ simulates an emission by $\mathtt{p}$ on $\mathtt{c}$ and the matching reception by $\mathtt{q}$, it internalizes this synchronization $\mathtt{c}! \cdot \mathtt{c}?$, and sets $\mathtt{q}$'s abstract counter to $=$. Indeed, since $\mathtt{q}$ has type $\circ$ on $\mathtt{c}$, the reception by $\mathtt{q}$ from $\mathtt{c}$ overwrites its counter with the value of $\mathtt{p}$'s counter. Then, $\widehat{\mathtt{p}}$ simulates, in one step, the behavior of $\mathtt{q}$ from this matching reception to the next reception. Observe that the next reception of $\mathtt{q}$ from $\mathtt{c}$ will, again, overwrite its counter. Therefore, thanks to Theorem 4.2, this behavior of $\mathtt{q}$ can be summarized in a single Presburger test, that accounts for the local state reached by $\mathtt{q}$. This way, $\widehat{\mathtt{p}}$ does not need to maintain the value held by $\mathtt{q}$'s counter. The construction guarantees that $\mathcal{S}$ has a full run if and only if the resulting system of communicating one-counter machines, with topology $\mathcal{U}$, has a full run.

The proof for the other direction $\mathtt{q} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{p}$ is similar. However, instead of scheduling $\mathtt{q}$ first, it is now scheduled last.

Two processes connected by one channel. We now consider the topology depicted on Figure 2(c), with two distinct processes $\mathtt{p}$ and $\mathtt{q}$ and a channel from $\mathtt{p}$ to $\mathtt{q}$ with type $\bullet$ on both end-points. Formally, $\mathcal{T} = \langle \{\mathtt{p}, \mathtt{q}\}, \{\mathtt{c}\}, \mathsf{src}, \mathsf{dst} \rangle$ with $\mathsf{src}(\mathtt{c}) = (\mathtt{p}, \bullet)$ and $\mathsf{dst}(\mathtt{c}) = (\mathtt{q}, \bullet)$.

▶ **Lemma 4.4.** Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-1cm.

A formal proof of the lemma is provided in Appendix B.3. Informally, given a system of communicating one-counter machines $\mathcal{S} = \langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$, we construct a one-counter machine $\mathcal{N}$ that simulates the "product" of $\mathtt{p}$ and $\mathtt{q}$. As in the proof of Lemma 4.3, we schedule the sender last (here, $\mathtt{p}$) and the receiver first (here, $\mathtt{q}$). Thus, emissions $\mathtt{c}!$ and receptions $\mathtt{c}?$ occur consecutively, with no other action in between. Since $\mathtt{p}$ and $\mathtt{q}$ have type $\bullet$ on $\mathtt{c}$, each sequence of actions $\mathtt{c}! \cdot \mathtt{c}?$ may occur only if $\mathtt{p}$'s counter and $\mathtt{q}$'s counter hold

the same value. So $\mathcal{N}$ internalizes each synchronization $\mathtt{c}!\cdot\mathtt{c}?$, and simulates, in one step, the behavior of $\mathtt{p}$ and $\mathtt{q}$ from one synchronization to the next. This is possible thanks to Theorem 4.2, which entails that counter reachability relations are (effectively) closed under intersection. The construction guarantees that $\mathcal{S}$ has a full run if and only if the constructed one-counter machine $\mathcal{N}$ has a full run.

Wrap up. We now have the necessary ingredients to prove the "if" direction of Theorem 3.2. Consider a weakly-connected topology $\mathcal{T}$ that is both cycle-free and shunt-free. We show that Rp-Sc1cm($\mathcal{T}$) is reducible to Rp-1cm. If $\mathcal{T}$ contains only one process, then $\mathcal{T}$ contains no channel as it is cycle-free, hence, Rp-Sc1cm($\mathcal{T}$) is obviously reducible to Rp-1cm. Assume that $\mathcal{T}$ contains at least two processes. By Proposition 3.3, there exists two distinct processes r, r' and a channel c, with $r \overset{c}{\leftrightarrow} r'$, such that, for every simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ with $p_0 \in \{r, r'\}$ and $q \notin \{r, r'\}$, the process q has type $\circ$ on the channel d. Moreover, according to Proposition 3.4, we may replace some end points $(p, \circ)$ by $(p, \bullet)$, as the reachability problem Rp-Sc1cm($\mathcal{T}$) is reducible to the reachability problem for the transformed topology. So we assume, without loss of generality, that for every simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, p, d, q)$ with $p_0 \in \{r, r'\}$, the process p has type $\bullet$ on the channel d. In particular, r and r' have type $\bullet$ on c.

Since $\mathcal{T}$ is cycle-free, its underlying undirected graph $(P, \{\overset{c}{\leftrightarrow}\}_{c\in C})$ is a tree. Pick a leaf process q that is distinct from r and r' (if any). Let $\mathcal{T} - q$ denote the topology obtained from $\mathcal{T}$ by removing the process q as well as its pendant channel. The simple undirected path from r to q ends with a channel $p \overset{d}{\leftrightarrow} q$ that satisfies $C(q) = \{d\}$, $\mathsf{typ}(p, d) = \bullet$ and $\mathsf{typ}(q, d) = \circ$. It follows from Lemma 4.3 that Rp-Sc1cm($\mathcal{T}$) is reducible to Rp-Sc1cm($\mathcal{T} - q$). By iterating this elimination technique in a bottom-up fashion, we obtain that Rp-Sc1cm($\mathcal{T}$) is reducible to Rp-Sc1cm($\mathcal{U}$) where $\mathcal{U}$ is the topology consisting of the two processes r, r' and the single channel c. According to Lemma 4.4, Rp-Sc1cm($\mathcal{U}$) is reducible to Rp-1cm. We conclude that Rp-Sc1cm($\mathcal{T}$) is reducible to Rp-1cm. Since the latter decidable, we get that the former is decidable, too.

5 Systems with Eager Communication

As seen in Example 1, cyclic topologies are the backbone of communication protocols. However, already for Cfsm, the reachability problem is undecidable in this case—this is also mirrored in Theorem 3.2. In the following, we will consider a restriction to so called *eager runs*. This restriction provides an under-approximative answer to the reachability problem Rp-Sc1cm($\mathcal{T}$) considered in the previous sections. Eager runs are close to globally 1-bounded runs, and have been successfully applied, in combination with other restrictions, for the reachability analysis of communicating pushdown processes [14].

▶ **Definition 5.1.** A run $\rho = (\sigma_0, a_1, \sigma_1, \ldots, a_n, \sigma_n)$ in $[\![\mathcal{S}]\!]$ is called *eager* if, for every channel c and for every $i \in \{1, \ldots, n-1\}$, $a_i = c!$ if and only if $a_{i+1} = c?$.

Thus, the restriction to eager runs renders message-passing synchronization via Fifo channels to a rendezvous. As all runs for a cycle-free topology can be reordered[1] into eager runs [14], the restriction to eager runs is only interesting in presence of cycles. Eager runs

[1] A run ρ can be *re-ordered* into a run ρ' if ρ can be transformed into ρ' by iteratively commuting adjacent transitions that (i) are from different processes, and (ii) do *not* form a matching send/receive pair.

restrict message passing in a cyclic topology by assuring that all other channels must be empty if one channel is currently transferring a message.

The *eager-reachability problem* Rp-Sc1cm-eager($\mathcal{T}$) is defined in the same way as Rp-Sc1cm($\mathcal{T}$) except that we search for a full run that has to be eager. This problem provides an under-approximative answer to Rp-Sc1cm($\mathcal{T}$), in the sense that we can deduce a positive answer to Rp-Sc1cm($\mathcal{T}$) from a positive answer to Rp-Sc1cm-eager($\mathcal{T}$). Moreover, it follows from Theorem 3.2 that, for every cycle-free topology $\mathcal{T}$, Rp-Sc1cm-eager($\mathcal{T}$) is decidable if and only if $\mathcal{T}$ is shunt-free. The following result allows to derive decidability for the eager-reachability problem while focussing cyclic communication, i.e., strongly-connected topologies. A detailed proof is in given Appendix C.

▶ **Proposition 5.2.** *Given a strongly-connected topology $\mathcal{T}$,* Rp-Sc1cm-eager($\mathcal{T}$) *is decidable if and only if $\mathcal{T}$ contains at most two processes.*

Strongly-connected topologies of *at most two* processes $p, q \in P$ contain (i) self-loops and (ii) a number of channels in-between p and q. By the restriction to eager runs, channels of type (i) become irrelevant. For simplicity, let us thus consider $\mathcal{T} : q \overset{c}{\rightarrowtail} r \overset{c'}{\rightarrowtail} q$ where all channels types are $\bullet$. Then eagerness allows us reverse the direction of a channel, leading to $\mathcal{U} : q \overset{c}{\rightarrowtail} r \overset{c'}{\leftarrowtail} q$. With the same encoding as in Lemma 3.5, we can tag each message in addition by the channel c or c' it is sent over. As eager message passing only uses one channel at a time, we can assert that all messages are now passed over one common channel. Hence we can apply the positive decidability result of Lemma 4.4 on two processes connected by one channel. The same construction can be extended to more than two channels between p and q, and by Proposition 3.4 (generalized to eager-reachability) to channel ends of type $\circ$.

For the converse, consider a strongly-connected component of *at least three* processes. There are two cases: (a) a directed cycle of length at least three, e.g., $\mathcal{T}_a : p \overset{c_1}{\rightarrowtail} q \overset{c_2}{\rightarrowtail} r \overset{c_3}{\rightarrowtail} p$ and (b) at least two directed cycles, each of length two, that are disjoint except for one common node, e.g., $\mathcal{T}_b : q \overset{c_1}{\rightarrowtail} p \overset{c_2}{\rightarrowtail} r \overset{c_3}{\rightarrowtail} p \overset{c_4}{\rightarrowtail} q$. We show a reduction from the reachability problem for two-counter machines. The underlying idea is to use the restriction to eager runs—hence the guarantee that in cyclic topology sending a message over one channel blocks the others—to "force" a certain communication behavior that allows to implement a protocol that gives one distinct process always access to two counters that are stored and passed inside the topology between the other processes without getting lost. For example in case of $\mathcal{T}_a$, process p simulates the two-counter machine $\mathcal{M}$, by storing one of the counters locally, the other at r. Before accessing the non-local counter, the protocol assures that we switch the counters by using q as buffer. In case of $\mathcal{T}_b$, $\mathcal{M}$ is simulated by p while q and r are used as registers for either one of the two counters.

The topology of Figure 1 falls in the scope of the previous proposition. If we additionally assume that the client's reception of the message on c_2 has precedence over sending a new message on c_1, then the channels are used in a half-duplex/mutex way. By [14], every run can then be re-ordered into an eager run. Hence, we can decide in this case whether the protocol is safe or not.

6 Conclusion and Perspectives

The study of the reachability question for systems of communicating one-counter machines demands to tackle the additional sources of infinity with respect to Cfsm, namely, the infinite message alphabet and the local counters. Thanks to a characterization of one-counter reachability relations in terms of binary Presburger predicates, we have obtained

a complete classification of the topologies having a solvable reachability question. This shows, in particular, that decidable topologies are the same as for the weaker model of Cfsm. To address topologies allowing mutual communications, we have considered an under-approximative approach by restricting runs to eager ones. As a preliminary result, we provide a characterization of the strongly-connected topologies that have a solvable eager-reachability question. A complete characterization of decidable topologies for eager reachability is currently under investigation. Further, we plan to extend our results from message passing infinite counter values to systems of communicating pushdown machines that can pass their stacks, i.e., finite words whose length cannot be bounded a priori.

References

1 P. Abdulla, B. Jonsson. Verifying programs with unreliable channels. *Inform. & Comput.*, 127(2):91–101, 1996.
2 S. Böhm, S. Göller, P. Jancar. Bisimilarity of one-counter processes is PSPACE-complete. In *CONCUR'10*, *LNCS* 6269, pp. 177–191. Springer, 2010.
3 M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, L. Segoufin. Two-variable logic on data words. *ACM TOCL*, 12:27, 2011.
4 B. Bollig, A. Cyriac, P. Gastin, K. Narayan Kumar. Model checking languages of data words. In *FOSSACSS'12*, 7213 *LNCS*, pp. 391–405. Springer, 2012.
5 A. Bouajjani, P. Habermehl, R. Mayr. Automatic verification of recursive procedures with one integer parameter. *Theor. C*, 295:85–106, 2003.
6 D. Brand, P. Zafiropoulo. On communicating finite-state machines. Research Report 1053, IBM Zürich Research Laboratory, 1981.
7 P. Chambart, P. Schnoebelen. Mixing lossy and perfect fifo channels. In *CONCUR 2008*, 5201 *LNCS*, pp. 340–355, 2008.
8 S. Demri, R. Lazic, A. Sangnier. Model checking freeze LTL over one-counter automata. In *FOSSACS'08*, 4962 *LNCS*, pp. 490–504. Springer, 2008.
9 A. Finkel. Decidability of the termination problem for completely specified protocols. *Distributed Computing*, 7(3):129–135, 1994.
10 A. Finkel, G. Memmi. Fifo nets: a new model of parallel computation. In *TCS'83*, *LNCS* 145, pp. 111–121. Springer, 1983.
11 S. Ginsburg, E. H. Spanier. Semigroups, Presburger formulas and languages. *Pacific J. Math.*, 16(2):285–296, 1966.
12 S. Göller, C. Haase, J. Ouaknine, J. Worrell. Branching-time model checking of parametric one-counter automata. In *FOSSACS'12*, *LNCS* 7213, pp. 406–420. Springer, 2012.
13 C. Haase, S. Kreutzer, J. Ouaknine, J. Worrell. Reachability in succinct and parametric one-counter automata. In *CONCUR 2009*, *LNCS* 5710, pp. 369–383, 2009.
14 A. Heußner, J. Leroux, A. Muscholl, G. Sutre. Reachability analysis of communicating pushdown systems. In *FOSSACS 2010*, *LNCS* 6014, pp. 267–281. Springer, 2010.
15 J. E. Hopcroft, J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theor.CS*, 8(2):135–159, 1979.
16 S. La Torre, P. Madhusudan, G. Parlato. Context-bounded analysis of concurrent queue systems. In *TACAS'08*, *LNCS* 4963, pp. 299–314. Springer, 2008.
17 T. Le Gall, B. Jeannet. Lattice automata In *SAS'07*, *LNCS* 4634, pp. 52–68. Springer, 2007.
18 M. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.
19 J.K. Pachl. Reachability problems for communicating finite state machines. Research Report CS-82-11, Dept. of C.S. Univ. of Waterloo, 1982.

A Proofs of Section 3

▶ **Proposition 3.3.** *Let $\mathcal{T}$ be a weakly-connected topology with at least two processes. If $\mathcal{T}$ is cycle-free and shunt-free, then there are two distinct processes r, r', with $r \overset{c}{\leftrightarrow} r'$ for some channel c, such that, for every simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ with $p_0 \in \{r, r'\}$ and $q \notin \{r, r'\}$, the process q has type $\circ$ on the channel d.*

Proof. Let us write $\mathcal{T} = \langle P, C, \mathsf{src}, \mathsf{dst} \rangle$. We consider two cases. Assume that there exists a channel $c \in C$ with both endpoints of type $\bullet$. There are two processes r, r' in P such that $\mathsf{src}(c) = (r, \bullet)$ and $\mathsf{dst}(c) = (r', \bullet)$. Moreover, $r \neq r'$ since $\mathcal{T}$ is cycle-free. Consider a simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ with $p_0 \in \{r, r'\}$ and $q \in P \setminus \{r, r'\}$. By contradiction, assume that q has type $\bullet$ on d. Let $\overline{p_0}$ denote the process $\overline{p_0} = r'$ if $p_0 = r$ and $\overline{p_0} = r$ if $p_0 = r'$.

- If $\overline{p_0} = p_i$ for some $i \in \{1, \ldots, n\}$, then $(p_0, c_1, p_1, \ldots, c_i, \overline{p_0})$ and $(p_0, c, \overline{p_0})$ are both simple undirected paths from p_0 to $\overline{p_0}$. Since $\mathcal{T}$ is cycle-free, they are necessarily equal, hence $c_1 = c$ and $p_1 = \overline{p_0}$. We obtain that $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ is simple undirected shunt, since p_0 has type $\bullet$ on $c_1 = c$.
- Otherwise, $\overline{p_0} \neq p_i$ for all $i \in \{0, \ldots, n\}$. We obtain that $(\overline{p_0}, c, p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ is simple undirected shunt, since $\overline{p_0}$ has type $\bullet$ on c.

In both cases, we derive that $\mathcal{T}$ contains a simple undirected shunt, which is impossible. This shows that q has type $\circ$ on d.

Assume now that, on the contrary to the above case, it holds that $\mathsf{src}(c) = \circ$ or $\mathsf{dst}(c) = \circ$, for every channel $c \in C$. We introduce the relation $\preceq$ on P defined by $p \preceq q$ if $p = q$ or there exists a simple undirected path $(p, c, p_1, c_1, \ldots, p_n, c_n, q)$ such that p has type $\bullet$ on c. Let us show that $\preceq$ is a partial order on P. Reflexivity is obvious.

- Antisymmetry: If $p \preceq q$ and $q \preceq p$, then $p = q$. Otherwise, by unicity of simple undirected paths between p and q, $\mathcal{T}$ would contain a simple undirected path $(p, c, p_1, c_1, \ldots, p_n, c_n, q)$ such that p has type $\bullet$ on c and q has type $\bullet$ on c_n. This is impossible, since $\mathcal{T}$ contains no simple undirected shunt nor channel with both endpoints of type $\bullet$.
- Transitivity: If $p \preceq q$ and $q \preceq r$ then $p \preceq r$. Indeed, if $p \neq q$ and $q \neq r$ then q is necessarily on the simple undirected path from p to r. Otherwise, $\mathcal{T}$ would contain a simple undirected path $(p, c, p_1, c_1, \ldots, p_n, c_n, q)$ such that p has type $\bullet$ on c and q has type $\bullet$ on c_n. This is impossible for the same reasons as for antisymmetry.

Let r be a minimal element of the partially ordered set $(P, \preceq)$. Since $\mathcal{T}$ is weakly-connected and contains at least two processes, there exists a process r' such that $r \overset{c}{\leftrightarrow} r'$ for some channel c. Moreover, $r \neq r'$ since $\mathcal{T}$ is cycle-free. Consider a simple undirected path $(p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ with $p_0 \in \{r, r'\}$ and $q \in P \setminus \{r, r'\}$. By contradiction, assume that q has type $\bullet$ on d.

- If $r = p_i$ for some $i \in \{0, \ldots, n\}$, then $(r, c_{i+1}, p_{i+1}, \ldots, c_n, p_n, d, q)$ is a simple undirected path, hence, $q \preceq r$.
- Otherwise, $r \neq p_i$ for all $i \in \{0, \ldots, n\}$. It follows that $(r, c, p_0, c_1, p_1, \ldots, c_n, p_n, d, q)$ is a simple undirected path, hence, $q \preceq r$.

In both cases, we derive that $q \preceq r$. This entails that $q = r$, which is impossible. We get that q has type $\circ$ on d. This concludes the proof the Proposition. ◀

▶ **Proposition 3.4.** *For every topology $\mathcal{T}$ and for every sub-topology $\mathcal{U}$ of $\mathcal{T}$,* Rp-Sc1cm($\mathcal{U}$) *is reducible to* Rp-Sc1cm($\mathcal{T}$).

Proof. We consider the case where $\mathcal{T} = \langle P_\mathcal{T}, C_\mathcal{T}, \mathsf{src}_\mathcal{T}, \mathsf{dst}_\mathcal{T} \rangle$ and $\mathcal{U} = \langle P_\mathcal{U}, C_\mathcal{U}, \mathsf{src}_\mathcal{U}, \mathsf{dst}_\mathcal{U} \rangle$ have the same processes and channels, i.e., $P_\mathcal{T} = P_\mathcal{U} = P$ and $C_\mathcal{T} = C_\mathcal{U} = C$. The extension

to the general case is obvious. Consider a system of communicating one-counter machines $\mathcal{S} = \langle \mathcal{U}, (\mathcal{M}^p)_{p\in P}\rangle$. We need to take care of endpoints of type $\circ$ in $\mathcal{U}$ that are of type $\bullet$ in $\mathcal{T}$. For every channel $c \in C$ such that $\mathsf{src}_{\mathcal{U}}(c) = (p, \bullet)$ but $\mathsf{src}_{\mathcal{T}}(c) = (p, \circ)$, we insert in $\mathcal{M}^p$, before each receive from c, two loops $\mathtt{add}(-1)$ and $\mathtt{add}(1)$ that set the counter to an arbitrary value. We proceed analogously for sends, except that the two loops are inserted after the send. Let $\widehat{\mathcal{M}^p}$ denote the resulting communicating one-counter machine, and define $\widehat{\mathcal{S}} = \langle \mathcal{T}, (\widehat{\mathcal{M}^p})_{p\in P}\rangle$. By construction, $[\![\mathcal{S}]\!]$ has a full run if and only if $[\![\widehat{\mathcal{S}}]\!]$ does. ◀

A.1 Cycle-freeness and Shunt-freeness of Decidable Topologies

▶ **Lemma 1.1.** *For every topology $\mathcal{T}$ containing a simple undirected shunt,* Rp-Sc1cm($\mathcal{T}$) *is undecidable.*

Proof. We use a construction based on the synchronization over a finite alphabet of two communicating one-counter machines proposed in Section **??** to simulate a two counter (Minsky) machine on a system of communicating one-counter machines whose topology is an undirected shunt. This allows to polynomially reduce the undecidability of the reachability problem of two-counter machines to the undecidability of the reachability question of a system of communicating one-counter machines that is not shunt-free by Proposition 3.4.

Assert that $\mathcal{T}$ is given as $p_1 \rightarrowtail^{c_1} q \overset{c_2}{\longleftarrow} p_2$, with $\mathsf{src}(c_1) = \bullet = \mathsf{src}(c_2)$ and $\mathsf{dst}(c_1) = \circ = \mathsf{dst}(c_1)$. Let $\mathcal{M} = \langle S^{\mathcal{M}}, s_I^{\mathcal{M}}, s_F^{\mathcal{M}}, A^{\mathcal{M}}, \Delta^{\mathcal{M}}\rangle$ be a two-counter machine where

- $S^{\mathcal{M}}$ is a finite set of states,
- $s_i^{\mathcal{M}}, s_f^{\mathcal{M}} \in S^{\mathcal{M}}$ are the single initial and final state,
- $A^{\mathcal{M}} = \{\mathtt{add}_i(z), \mathtt{test}_i \,|i \in \{1, 2\} \wedge z \in \mathbb{Z}\}$ is the set of counter actions,
- $\Delta^{\mathcal{M}} \subseteq S^{\mathcal{M}} \times A^{\mathcal{M}} \times S^{\mathcal{M}}$ is the set of transition rules.

We assert the standard counter system semantics such that configurations of $\mathcal{M}$ are given as $(q, x_1, x_2) \in S^{\mathcal{M}} \times \mathbb{N} \times \mathbb{N}$.

We construct a system of communicating counter machines $\mathcal{S}$ by "splitting" a two-counter machine $\mathcal{M}$ into two one-counter machines that each simulate $\mathcal{M}$ by mapping a different one of $\mathcal{M}$'s counters to the local counter and doing the actions on the other "virtually". We use a third process to synchronize both machine's executions by passing messages over the finite action alphabet over the shunt. The latter is possible as $A^{\mathcal{M}}$ is finite.

Formally, $\mathcal{S}$ relies on the local machines $\mathcal{M}^{p_1}$, $\mathcal{M}^q$, and $\mathcal{M}^{p_2}$ as follows: $\mathcal{M}^{p_i}$ equals $\mathcal{M}$ where we additionally send or receive each fired action after its completion on channel c_i. Further, $\mathcal{M}^{p_i}$ replaces all actions of $\mathcal{M}$ on counter $1 - i$ by the action $\mathtt{add}(0)$. Thus, we have two copies of $\mathcal{M}$ that both synchronize their actions and that each simulate a different one the two counters locally. The process $\mathcal{M}^q$ assures that each reception of a message $m \in A^{\mathcal{M}}$ on channel c_1 is followed by a reception of m on c_2. Hence, we can assert $S_I^q = S_F^q = \{s^q\} \subset S^q$.

Given a run of $\mathcal{M}$ from $s_i^{\mathcal{M}}$ to $s_f^{\mathcal{M}}$, there exists, by the previous construction, a run of $\mathcal{S}$ from the configuration $((s_i^{\mathcal{M}}, s^q, s_i^{\mathcal{M}}), \mathbf{0}, \boldsymbol{\varepsilon})$ to $((s_f^{\mathcal{M}}, s^q, s_f^{\mathcal{M}}), \mathbf{0}, \boldsymbol{\varepsilon})$. Vice versa, if there exists a run in $[\![\mathcal{S}]\!]$ from $((s_i^{\mathcal{M}}, s^q, s_i^{\mathcal{M}}), \mathbf{0}, \boldsymbol{\varepsilon})$ to $((s_f^{\mathcal{M}}, s_q, s_f^{\mathcal{M}}), \mathbf{0}, \boldsymbol{\varepsilon})$ then by induction on the sequence of fired transitions and the fact that the channels must be empty at the end, there exists a run of $\mathcal{M}$ from $s_i^{\mathcal{M}}$ to $s_f^{\mathcal{M}}$.

The previous construction is independent of the direction of the channels in the shunt as for the following topology $p_1 \rightarrowtail^{c_1} q \overset{c_2}{\longrightarrow} p_2$ the process q "forwards" the actions from p_1 to p_2, and for $p_1 \overset{c_1}{\longleftarrow} q \overset{c_2}{\longrightarrow} p_2$ process q "generates" a sequence of actions that is executed on both p_1 and p_2. Similar to the discussion in Section **??**, the previous argumentation is also independent of the length of the shunt.

Note that this proof relies on reduction to systems of basic one-counter machines, whose only Presburger test are $\mathtt{x} = 0$. ◀

B Proofs of Section 4

B.1 Counter reachability relations of one-counter machines

▶ **Lemma 2.1.** *Every unary Presburger predicate $\varphi(\mathtt{x})$ is logically equivalent to a disjunction of unary Presburger predicates of the form $\exists \mathtt{k} \cdot (\mathtt{x} = \beta + \mathtt{k} \cdot \pi)$ where $\beta, \pi \in \mathbb{N}$ are constants.*

Proof. Recall that, for every dimension n, Presburger-definable subsets of $\mathbb{N}^n$ are *semilinear sets*, i.e., finite unions of *linear* sets $\boldsymbol{b} + \mathbb{N}\boldsymbol{p}_1 + \cdots + \mathbb{N}\boldsymbol{p}_k$, where $\boldsymbol{b} \in \mathbb{N}^n$ is a *basis* and $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k \in \mathbb{N}^n$ are *periods* [11]. In dimension 1, it is readily seen that one period is sufficient. Indeed, every set of the form $b + \mathbb{N}p_1 + \cdots + \mathbb{N}p_k$ may be written as a semilinear set $B + \mathbb{N}p$ where B is a finite subset of $\mathbb{N}$ and p is either zero if all p_i's are zero, or the greatest common divisor of the p_i's otherwise. The lemma follows. ◀

▶ **Lemma 2.2.** *For every basic one-counter machine $\mathcal{M}$, the counter reachability relation of $\mathcal{M}$ is defined by a one-counter Presburger predicate.*

Proof. Consider a basic one-counter machine $\mathcal{M} = \langle S, I, F, A, \Delta \rangle$. Let Z denote the set of all pairs $(s,t) \in S \times S$ such that there is a run from $(s,0)$ to $(t,0)$ in $\mathcal{M}$. For every $s, t \in S$, we introduce the subsets $\overrightarrow{R}_{s,t}$ and $\overleftarrow{R}_{s,t}$ of $\mathbb{N}$ defined by:

- $y \in \overrightarrow{R}_{s,t}$ if there exists a run with no (zero-)`test` from $(s,0)$ to (t,y) in $\mathcal{M}$, and
- $x \in \overleftarrow{R}_{s,t}$ if there exists a run with no (zero-)`test` from (s,x) to $(t,0)$ in $\mathcal{M}$.

It is well-known that $\overrightarrow{R}$ and $\overleftarrow{R}$ are definable in Presburger arithmetic (see, e.g., [15]). Abusing notation, let $\overrightarrow{R}_{s,t}(\mathtt{y})$ and $\overleftarrow{R}_{s,t}(\mathtt{x})$ denote unary Presburger predicates defining $\overrightarrow{R}_{s,t}$ and $\overleftarrow{R}_{s,t}$, respectively. Each run of $\mathcal{M}$ either contains at least one zero-test action, or contains no zero-test action. In the former case, the run is of the form $\rho_1 \cdot (r,0) \xrightarrow{*} (u,0) \cdot \rho_2$ where ρ_1 and ρ_2 contain no zero-test. In the latter case, the run may be decomposed as $\rho_1 \cdot (r,d) \cdot \rho_2$ where d is the minimal value of the counter in ρ. We derive that the counter reachability relation of $\mathcal{M}$ is defined by the binary Presburger predicate[2]

$$\bigvee_{s \in I, t \in F} \left(\bigvee_{(r,u) \in Z} \overleftarrow{R}_{s,r}(\mathtt{x}) \wedge \overrightarrow{R}_{u,t}(\mathtt{y}) \right) \vee \left(\bigvee_{r \in S} \underbrace{\exists \mathtt{d} \cdot (\overleftarrow{R}_{s,r}(\mathtt{x} - \mathtt{d}) \wedge \overrightarrow{R}_{r,t}(\mathtt{y} - \mathtt{d}))}_{\psi_{s,r,t}} \right)$$

It remains to show that each formula $\psi_{s,r,t}$ can be replaced by a one-counter Presburger predicate. According to Lemma 2.1, each $\psi_{s,r,t}$ is logically equivalent to a disjunction of formulas of the form

$$\chi \quad = \quad \exists \mathtt{d} \cdot \exists \mathtt{h} \cdot \exists \mathtt{k} \cdot (\mathtt{x} = \mathtt{d} + \beta + \mathtt{h} \cdot \pi \ \wedge \ \mathtt{y} = \mathtt{d} + \gamma + \mathtt{k} \cdot \sigma)$$

where $\beta, \gamma, \pi, \sigma \in \mathbb{N}$ are constants. We consider three cases.

- If $\pi = 0$ then χ is logically equivalent to the formula

$$\mathtt{x} \geq \beta \ \wedge \ \exists \mathtt{k} \cdot (\mathtt{y} + \beta = \mathtt{x} + \gamma + \mathtt{k} \cdot \sigma)$$

[2] For a unary Presburger predicate φ, the notation $\varphi(x - y)$, where x and y are variables, is a shortcut for the formula $\exists z \cdot (x = y + z \wedge \varphi(z))$ where z is a variable distinct from x and y.

By splitting the right conjunct with respect to the relative order of $\mathtt{x}$ and $\mathtt{y}$, we obtain that χ is logically equivalent to the one-counter Presburger predicate $(\mathtt{x} \geq \beta) \wedge (\chi^{\mathtt{xy}} \vee \chi^{\mathtt{yx}})$ where

$$\begin{array}{rcl} \chi^{\mathtt{xy}} & = & \exists \mathtt{z} \cdot (\mathtt{x} = \mathtt{y} + \mathtt{z} \ \wedge \ \exists \mathtt{k} \cdot (\beta = \mathtt{z} + \gamma + \mathtt{k} \cdot \sigma)) \\ \chi^{\mathtt{yx}} & = & \exists \mathtt{z} \cdot (\mathtt{y} = \mathtt{x} + \mathtt{z} \ \wedge \ \exists \mathtt{k} \cdot (\mathtt{z} + \beta = \gamma + \mathtt{k} \cdot \sigma)) \end{array}$$

- If $\sigma = 0$ then, by proceeding as above, we obtain that χ is logically equivalent to a one-counter Presburger predicate.
- If $\pi > 0$ and $\sigma > 0$ then we split χ by considering two cases: either $\mathtt{x} < \pi \cdot \sigma$ or $\mathtt{x} \geq \pi \cdot \sigma$. The first case leads to the one-counter Presburger predicate

$$\chi_1 \ = \ \bigvee_{n=0}^{\beta+\pi\cdot\sigma-1} \mathtt{x} = n \ \wedge \ \exists \mathtt{d} \cdot \exists \mathtt{h} \cdot \exists \mathtt{k} \cdot (n = \mathtt{d} + \beta + \mathtt{h} \cdot \pi \ \wedge \ \mathtt{y} = \mathtt{d} + \gamma + \mathtt{k} \cdot \sigma)$$

The second case is expressed by the one-counter Presburger predicate

$$\chi_2 \ = \ \bigvee_{m=0}^{\pi-1} (\exists \mathtt{k} \cdot (\mathtt{k} \geq \sigma \ \wedge \ \mathtt{x} = m + \beta + \mathtt{k} \cdot \pi)) \ \wedge \ (\exists \mathtt{h} \cdot \exists \mathtt{k} \cdot (\mathtt{y} = m + \mathtt{h} \cdot \pi + \gamma + \mathtt{k} \cdot \sigma))$$

It is routinely checked that χ is logically equivalent to the one-counter Presburger predicate $\chi_1 \vee \chi_2$.

Thus, we have shown that each $\psi_{s,r,t}$ is logically equivalent to a one-counter Presburger predicate. We conclude that the counter reachability relation of $\mathcal{M}$ is defined by a one-counter Presburger predicate. ◀

▶ **Theorem 4.2.** *For every binary relation $R \subseteq \mathbb{N} \times \mathbb{N}$, the two following assertions are equivalent:*

- *R is the counter reachability relation of a one-counter machine,*
- *R is defined by a one-counter Presburger predicate.*

Proof. Assume that R is defined by a one-counter Presburger predicate ψ. Note that unary predicates with the same free variable are closed under conjunction. Therefore, we can put ψ into disjunctive normal form $\bigvee_{i \in I} (\chi_i^{\mathtt{x}} \wedge \chi_i^{\mathtt{y}} \wedge \chi_i^{\mathtt{xy}} \wedge \chi_i^{\mathtt{yx}})$ where:

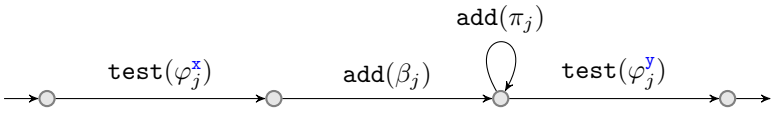
- Each χ_i^x, with $x \in \{\mathtt{x}, \mathtt{y}\}$, is either $\mathtt{tt}$, or a unary Presburger predicate $\varphi_i^x(x)$. If $\chi_i^x = \mathtt{tt}$ then we may equivalently replace it by the unary Presburger predicate $x = x$.
- Each χ_i^{xy}, with $(x, y) \in \{(\mathtt{x}, \mathtt{y}), (\mathtt{y}, \mathtt{x})\}$, is either $\mathtt{tt}$, or a formula $\exists z \cdot (x = y + z \wedge \varphi_i^{xy}(z))$, where φ_i^{xy} is a unary Presburger predicate and $z \notin \{x, y\}$. Furthermore, we may suppose, without loss of generality, that $\chi_i^{\mathtt{xy}} = \mathtt{tt}$ or $\chi_i^{\mathtt{yx}} = \mathtt{tt}$. Indeed, if that is not the case, then $\chi_i^{\mathtt{xy}} \wedge \chi_i^{\mathtt{yx}}$ entails $\mathtt{x} = \mathtt{y}$. Hence, $\chi_i^{\mathtt{xy}} \wedge \chi_i^{\mathtt{yx}}$ is logically equivalent to $\exists z \cdot (x = y + z \wedge \varphi_i^{=}(z))$, where $\varphi_i^{=}$ is the formula $z = 0 \wedge \varphi_i^{\mathtt{xy}}(z) \wedge \varphi_i^{\mathtt{yx}}(z)$. So we may equivalently replace $\chi_i^{\mathtt{xy}}$ and $\chi_i^{\mathtt{yx}}$ by $\exists z \cdot (x = y + z \ \wedge \ \varphi_i^{=}(z))$ and $\mathtt{tt}$, respectively.

We derive that ψ is logically equivalent to $\bigvee_{i \in I} (\varphi_i^{\mathtt{x}}(\mathtt{x}) \wedge \varphi_i^{\mathtt{y}}(\mathtt{y}) \wedge \xi_i^{\mathtt{xy}})$ where each $\xi_i^{\mathtt{xy}}$ is either $\mathtt{tt}$, or $\exists z \cdot (\mathtt{x} = \mathtt{y} + z \wedge \varphi_i^{\mathtt{xy}}(z))$, or $\exists z \cdot (\mathtt{y} = \mathtt{x} + z \wedge \varphi_i^{\mathtt{yx}}(z))$. It follows from Lemma 2.1 that ψ can be transformed into a logically equivalent Presburger predicate

$$\bigvee_{j \in J} \underbrace{\varphi_j^{\mathtt{x}}(\mathtt{x}) \ \wedge \ \varphi_j^{\mathtt{y}}(\mathtt{y}) \ \wedge \ \exists \mathtt{k} \cdot (\mathtt{y} = \mathtt{x} + \beta_j + \mathtt{k} \cdot \pi_j)}_{\psi_j}$$

where $\varphi_j^{\mathtt{x}}, \varphi_j^{\mathtt{y}} \in \mathcal{P}_1$ and $\beta_j, \pi_j \in \mathbb{Z}$ are constants[3] satisfying $\beta_j \cdot \pi_j \geq 0$. It is routinely checked that, for each $j \in J$, the binary relation defined by the disjunct ψ_j is the counter reachability relation of the one counter machine $\mathcal{M}_j$ depicted below:

[3] Here, we use constants in $\mathbb{Z}$ instead of $\mathbb{N}$, but this is purely a notational convenience.



We obtain that R is the counter reachability relation of the disjoint union of the one-counter machines $\mathcal{M}_j$.

Let us now prove the converse. Assume that R is the counter reachability relation of a one-counter machine $\mathcal{M} = \langle S, I, F, A, \Delta \rangle$. According to Lemma 2.1, we may suppose, without loss of generality, that every `test` action in $\mathcal{M}$ is of the form $\mathtt{test}(\exists \mathtt{k} \cdot (\mathtt{x} = \beta + \mathtt{k} \cdot \pi)$ where $\beta, \pi \in \mathbb{N}$ are constants. Moreover, the action $\mathtt{test}(\exists \mathtt{k} \cdot (\mathtt{x} = \beta + \mathtt{k} \cdot \pi))$ is semantically equivalent to the sequence of actions $\mathtt{add}(-\beta) \cdot \mathtt{test}(\pi \,|\, \mathtt{x}) \cdot \mathtt{add}(\beta)$, where $\pi \,|\, \mathtt{x}$ is a shortcut for $\exists \mathtt{k} \cdot (\mathtt{x} = \mathtt{k} \cdot \pi)$. So we may suppose, without loss of generality, that every `test` action in $\mathcal{M}$ is either $\mathtt{test}(\mathtt{x} = 0)$, or of the form $\mathtt{test}(\pi \,|\, \mathtt{x})$ where $\pi > 0$ is a constant. By a construction inspired from [5], we derive from $\mathcal{M}$ an "equivalent" one-counter machine $\mathcal{N}$ with zero-tests only. Let K denote the least common multiple of all π such that $\mathtt{test}(\pi \,|\, \mathtt{x})$ is an action of $\mathcal{M}$. Intuitively, $\mathcal{N}$ behaves as $\mathcal{M}$ but also maintains, in its state, the value of the counter modulo K. Formally, the set of states of $\mathcal{N}$ is $S \times \{0, \ldots, K-1\}$, and its set of transition rules is defined below. We do not fix the initial states of $\mathcal{N}$ nor its final states, yet. For simplicity, we write $(s,m) \overset{a}{\hookrightarrow} (t,n)$ for a transition rule $((s,m), a, (t,n))$ of $\mathcal{N}$. For every $s, t \in S$ and $m, n \in \{0, \ldots, K-1\}$,

$$
\begin{array}{lcll}
(s,m) & \xhookrightarrow{\mathtt{add}(k)} & (t,n) & \text{if} \quad (s, \mathtt{add}(k), t) \in \Delta \ \wedge \ n = (m+k) \bmod K \\
(s,m) & \xhookrightarrow{\mathtt{test}(\mathtt{x}=0)} & (t,n) & \text{if} \quad (s, \mathtt{test}(\mathtt{x} = 0), t) \in \Delta \ \wedge \ m = n = 0 \\
(s,m) & \xhookrightarrow{\mathtt{add}(0)} & (t,n) & \text{if} \quad (s, \mathtt{test}(\pi \,|\, \mathtt{x}), t) \in \Delta \ \wedge \ m = n \ \wedge \ m \bmod \pi = 0
\end{array}
$$

A routine induction on the length of runs shows that, for every $s, t \in S$ and $x, y \in \mathbb{N}$, there exists a run from (s,x) to (t,y) in $\mathcal{M}$ if and only if there exists a run from $((s, x \bmod K), x)$ to $((t, y \bmod K), y)$ in $\mathcal{N}$. According to Lemma 4.1, for any chosen initial and final states, the counter reachability relation of $\mathcal{N}$ is defined by a one-counter Presburger predicate. Therefore, R is defined by the following one-counter Presburger predicate:

$$
\bigvee_{\substack{s \in I, t \in F \\ 0 \leq m, n < K}} \exists \mathtt{h} \cdot (\mathtt{x} = m + \mathtt{h} \cdot K) \ \wedge \ \exists \mathtt{k} \cdot (\mathtt{y} = n + \mathtt{k} \cdot K) \ \wedge \ ((s,m), \mathtt{x}) \rightsquigarrow_{\mathcal{N}} ((t,n), \mathtt{y})
$$

where $((s,m), \mathtt{x}) \rightsquigarrow_{\mathcal{N}} ((t,n), \mathtt{y})$ is a one-counter Presburger predicate defining the counter reachability relation of $\mathcal{N}$ with initial state (s,m) and final state (t,n). ◀

B.2 Merging leaf processes

Let us recall the context of Subsection **??**. We selected a distinguished process $\mathtt{p}$ in the topology $\mathcal{U} = \langle P_{\mathcal{U}}, C_{\mathcal{U}}, \mathsf{src}_{\mathcal{U}}, \mathsf{dst}_{\mathcal{U}} \rangle$. The topology $\mathcal{T} = \langle P, C, \mathsf{src}, \mathsf{dst} \rangle$ has set of processes $P = P_{\mathcal{U}} \cup \{\mathtt{q}\}$, where $\mathtt{q} \notin P_{\mathcal{U}}$, and set of channels $C = C_{\mathcal{U}} \cup \{\mathtt{c}\}$, where $\mathtt{c} \notin C_{\mathcal{U}}$. The source and target mappings of $\mathcal{T}$ coincide with those of $\mathcal{U}$ on C, and it is assumed that $\mathtt{p} \overset{\mathtt{c}}{\leftrightarrow} \mathtt{q}$. Hence, $C(\mathtt{q}) = \{\mathtt{c}\}$.

▶ **Lemma 2.3.** *If* $\mathtt{p}$ *has type* $\bullet$ *on* $\mathtt{c}$ *and* $\mathtt{q}$ *has type* $\circ$ *on* $\mathtt{c}$ *then* Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-Sc1cm($\mathcal{U}$).

We prove this lemma by considering two cases, depending on the direction of the channel $\mathtt{c}$ between $\mathtt{p}$ and $\mathtt{q}$. These two cases are stated, and proved, in Lemma 2.4 and Lemma 2.5. To help readability, all objects obtained by reduction will be written with a hat. So, in particular, we write $\widehat{\mathcal{T}} = \mathcal{U}$, $\widehat{P} = P_{\mathcal{U}} = P \setminus \{\mathtt{q}\}$, and $\widehat{C} = C_{\mathcal{U}} = C \setminus \{\mathtt{c}\}$.

▶ **Lemma 2.4.** *Assume that* $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{q}$. *If* $\mathtt{p}$ *has type* $\bullet$ *on* $\mathtt{c}$ *and* $\mathtt{q}$ *has type* $\circ$ *on* $\mathtt{c}$ *then* Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-Sc1cm($\mathcal{U}$).

Proof. Suppose that $\mathsf{typ}(\mathtt{p},\mathtt{c}) = \bullet$ and $\mathsf{typ}(\mathtt{q},\mathtt{c}) = \circ$. Let $\mathcal{S} = \langle \mathcal{T}, (\mathcal{M}^p)_{p\in P}\rangle$ be a system of communicating one-counter machines, with topology $\mathcal{T}$. We construct, from $\mathcal{S}$, a system of communicating one-counter machines $\widehat{\mathcal{S}} = \langle \widehat{\mathcal{T}}, (\widehat{\mathcal{M}}^p)_{p\in\widehat{P}}\rangle$, with topology $\widehat{\mathcal{T}}$, such that $[\![\mathcal{S}]\!]$ has a full run if and only if $[\![\widehat{\mathcal{S}}]\!]$ has a full run. In order to define $\widehat{\mathcal{S}}$, we need to provide $\widehat{\mathcal{M}}^p$ for each process $p \in \widehat{P}$. Intuitively, we keep in $\widehat{\mathcal{S}}$ the same one-counter machines as in $\mathcal{S}$, except for $\mathtt{p}$ and $\mathtt{q}$, that are merged in the "product" machine $\widehat{\mathcal{M}}^{\mathtt{p}}$. Formally, $\widehat{\mathcal{M}}^p = \mathcal{M}^p$ for every $p \in P \setminus \{\mathtt{p},\mathtt{q}\}$, and $\widehat{\mathcal{M}}^{\mathtt{p}} = \langle \widehat{S}^{\mathtt{p}}, \widehat{I}^{\mathtt{p}}, \widehat{F}^{\mathtt{p}}, A^{\mathtt{p}}, \widehat{\Delta}^{\mathtt{p}}\rangle$ is defined as follows.

The formal definition of $\widehat{\mathcal{M}}^{\mathtt{p}}$ is given in the next paragraph. Before that, we provide some intuitions for the simulation of $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{q}$ in $\widehat{\mathcal{S}}$. Let $\widehat{\mathtt{p}}$ denote the process $\mathtt{p}$ in $\widehat{\mathcal{S}}$. Recall that our intention is to simulate, in $\widehat{\mathtt{p}}$, both processes $\mathtt{p}$ and $\mathtt{q}$, as well as the channel $\mathtt{c}$ in-between. We choose a specific interleaving of $\mathtt{p}$ and $\mathtt{q}$ where $\mathtt{c}$ is almost always empty, and such that $\widehat{\mathtt{p}}$, which has a single counter, is able to retain both $\mathtt{p}$'s counter and $\mathtt{q}$'s counter. In essence, $\widehat{\mathtt{p}}$ behaves as $\mathtt{p}$, but also maintains, in its state, the local state of $\mathtt{q}$ as well as an abstraction of $\mathtt{q}$'s counter. We abstract $\mathtt{q}$'s counter by the set $\{0,\bot,=\}$, where 0 means zero, $\bot$ means some unknown value, and $=$ means that $\mathtt{q}$'s counter holds the same value as $\mathtt{p}$'s counter. Furthermore, the process $\mathtt{q}$ is always scheduled first. Since $\mathtt{c}$ is the only channel with source or target $\mathtt{q}$, this means, in particular, that every reception by $\mathtt{q}$ from $\mathtt{c}$ occurs immediately after the matching emission by $\mathtt{p}$ on $\mathtt{c}$. When $\widehat{\mathtt{p}}$ simulates an emission by $\mathtt{p}$ on $\mathtt{c}$ and the matching reception by $\mathtt{q}$, it internalizes this synchronization $\mathtt{c}!\cdot\mathtt{c}?$, and sets $\mathtt{q}$'s abstract counter to $=$. Indeed, since $\mathtt{q}$ has type $\circ$ on $\mathtt{c}$, the reception by $\mathtt{q}$ from $\mathtt{c}$ overwrites its counter with the value of $\mathtt{p}$'s counter. Then, $\widehat{\mathtt{p}}$ simulates, in one step, the behavior of $\mathtt{q}$ from this matching reception to the next reception. Observe that the next reception of $\mathtt{q}$ from $\mathtt{c}$ will, again, overwrite its counter. Therefore, thanks to Theorem 4.2, this behavior of $\mathtt{q}$ can be summarized in a single Presburger test, that accounts for the local state reached by $\mathtt{q}$. This way, $\widehat{\mathtt{p}}$ does not need to maintain the value held by $\mathtt{q}$'s counter.

We now provide the formal definition of $\widehat{\mathcal{M}}^{\mathtt{p}}$. Recall that $\mathtt{q}$'s counter is abstracted by the set $\{0,\bot,=\}$, where 0 means zero, $\bot$ means some unknown value, and $=$ means that $\mathtt{q}$'s counter holds the same value as $\mathtt{p}$'s counter. As expected, the sets of states, initial states and final states of $\widehat{\mathcal{M}}^{\mathtt{p}}$ are:

$$\widehat{S}^{\mathtt{p}} \;=\; S^{\mathtt{p}} \times S^{\mathtt{q}} \times \{0,\bot,=\} \qquad \text{and} \qquad \widehat{I}^{\mathtt{p}} \;=\; I^{\mathtt{p}} \times I^{\mathtt{q}} \times \{=\} \qquad \text{and} \qquad \widehat{F}^{\mathtt{p}} \;=\; F^{\mathtt{p}} \times F^{\mathtt{q}} \times \{0\}$$

The set of transition rules $\widehat{\Delta}^{\mathtt{p}}$ is defined below. For simplicity, we write $s \overset{a}{\hookrightarrow} t$ in place of $(s,a,t) \in \widehat{\Delta}^{\mathtt{p}}$. Let $\mathcal{M}^{\mathtt{q}}_{\mathrm{cnt}}$ denote the one-counter machine obtained from $\mathcal{M}^{\mathtt{q}}$ by removing all transition rules labeled with communication actions. According to Theorem 4.2, for every pair $(s^{\mathtt{q}}_1, s^{\mathtt{q}}_2)$ of states in $S^{\mathtt{q}}$, there exists a computable one-counter Presburger predicate, written $(s^{\mathtt{q}}_1, \mathtt{x}) \rightsquigarrow_{\mathtt{q}} (s^{\mathtt{q}}_2, \mathtt{y})$, defining the counter reachability relation of $\mathcal{M}^{\mathtt{q}}_{\mathrm{cnt}}$ with initial state

$s_1^{\mathtt{q}}$ and final state $s_2^{\mathtt{q}}$. For every $s_1^{\mathtt{p}}, s_2^{\mathtt{p}} \in S^{\mathtt{p}}$, $s_1^{\mathtt{q}}, s_2^{\mathtt{q}} \in S^{\mathtt{q}}$ and $\alpha \in \{0, \bot\}$,

$$\begin{array}{lcll}
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, \alpha) & \xhookrightarrow{a} & (s_2^{\mathtt{p}}, s_1^{\mathtt{q}}, \alpha) & \text{if} \quad (s_1^{\mathtt{p}}, a, s_2^{\mathtt{p}}) \in \Delta^{\mathtt{p}} \ \wedge \ a \neq \mathtt{c}\,! \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, \bot) & \xrightarrow{\mathtt{add}(0)} & (s_2^{\mathtt{p}}, s_2^{\mathtt{q}}, =) & \text{if} \quad (s_1^{\mathtt{p}}, \mathtt{c}\,!, s_2^{\mathtt{p}}) \in \Delta^{\mathtt{p}} \ \wedge \ (s_1^{\mathtt{q}}, \mathtt{c}\,?, s_2^{\mathtt{q}}) \in \Delta^{\mathtt{q}} \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, =) & \xrightarrow{\varphi(\mathtt{x})} & (s_1^{\mathtt{p}}, s_2^{\mathtt{q}}, 0) & \text{if} \quad \varphi(\mathtt{x}) \ = \ (s_1^{\mathtt{q}}, \mathtt{x}) \rightsquigarrow_{\mathtt{q}} (s_2^{\mathtt{q}}, 0) \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, =) & \xrightarrow{\varphi(\mathtt{x})} & (s_1^{\mathtt{p}}, s_2^{\mathtt{q}}, \bot) & \text{if} \quad \varphi(\mathtt{x}) \ = \ \exists \mathtt{y} \cdot (s_1^{\mathtt{q}}, \mathtt{x}) \rightsquigarrow_{\mathtt{q}} (s_2^{\mathtt{q}}, \mathtt{y})
\end{array}$$

This completes the definition of $\widehat{\mathcal{S}} = \langle \widehat{\mathcal{T}}, (\widehat{\mathcal{M}^p})_{p \in \widehat{P}} \rangle$.

It remains to show that $[\![\mathcal{S}]\!]$ has a full run if and only if $[\![\widehat{\mathcal{S}}]\!]$ has a full run. Assume that $[\![\mathcal{S}]\!]$ has a full run ρ. Without loss of generality, we may assume that ρ schedules $\mathtt{q}$ first. Since $C(\mathtt{q}) = \{\mathtt{c}\}$, this means, by standard partial-order techniques, that ρ may be decomposed as $\rho = \pi_0 \cdot \chi_0 \cdot \xi_1 \cdot \pi_1 \cdot \chi_1 \cdots \xi_n \cdot \pi_n \cdot \chi_n$ where

- each $\pi_i = (\boldsymbol{s}_i, \boldsymbol{x}_i, \boldsymbol{w}_i) \xrightarrow{*} (\boldsymbol{t}_i, \boldsymbol{y}_i, \boldsymbol{w}_i)$ contains only moves of $\mathtt{q}$ and no reception from $\mathtt{c}$,
- each $\chi_i = (\boldsymbol{t}_i, \boldsymbol{y}_i, \boldsymbol{w}_i) \xrightarrow{*} (\boldsymbol{u}_i, \boldsymbol{z}_i, \boldsymbol{w}_{i+1})$ contains no move of $\mathtt{q}$ nor emission on $\mathtt{c}$, and
- each $\xi_i = (\boldsymbol{u}_{i-1}, \boldsymbol{z}_{i-1}, \boldsymbol{w}_i) \xrightarrow{\mathtt{c}\,!} \cdot \xrightarrow{\mathtt{c}\,?} (\boldsymbol{s}_i, \boldsymbol{x}_i, \boldsymbol{w}_i)$.

Observe that the channel $\mathtt{c}$ remains empty in ρ except in the intermediate configuration of each ξ_i. Since $\mathsf{typ}(\mathtt{p}, \mathtt{c}) = \bullet$, we derive that $x_i^{\mathtt{q}} = x_i^{\mathtt{p}}$ for all $i \in \{0, \ldots, n\}$. We construct a full run $\widehat{\rho} = \widehat{\pi}_0 \cdot \widehat{\chi}_0 \cdot \widehat{\xi}_1 \cdot \widehat{\pi}_1 \cdot \widehat{\chi}_1 \cdots \widehat{\xi}_n \cdot \widehat{\pi}_n \cdot \widehat{\chi}_n$ of $[\![\widehat{\mathcal{S}}]\!]$ as follows. We let $\widehat{\boldsymbol{x}}_i$, $\widehat{\boldsymbol{y}}_i$ and $\widehat{\boldsymbol{z}}_i$ denote the projection on $\widehat{P}$ of $\boldsymbol{x}_i$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$, respectively. Note that $\widehat{\boldsymbol{x}}_i = \widehat{\boldsymbol{y}}_i = \widehat{\boldsymbol{z}}_{i-1}$. Similarly, $\widehat{\boldsymbol{w}}_i = (w_i^d)_{d \in \widehat{C}}$ is the projection on $\widehat{C}$ of $\boldsymbol{w}_i$. The runs $\widehat{\pi}_i$, $\widehat{\chi}_i$ and $\widehat{\xi}_i$ of $[\![\widehat{\mathcal{S}}]\!]$ are defined by:

- $\widehat{\pi}_i$ is obtained from π_i by compressing it into a single transition of $\widehat{\mathcal{M}^{\mathtt{p}}}$ that simulates π_i. This is possible because $x_i^{\mathtt{q}} = x_i^{\mathtt{p}}$. Formally,

$$\widehat{\pi}_i \quad = \quad \begin{cases} (\boldsymbol{s}_i, =, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{\exists \mathtt{y} \cdot (s_i^{\mathtt{q}}, \mathtt{x}) \rightsquigarrow_{\mathtt{q}} (t_i^{\mathtt{q}}, \mathtt{y}))} (\boldsymbol{t}_i, \bot, \widehat{\boldsymbol{y}}_i, \widehat{\boldsymbol{w}}_i) & \text{if } i < n \\ (\boldsymbol{s}_i, =, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{(s_i^{\mathtt{q}}, \mathtt{x}) \rightsquigarrow_{\mathtt{q}} (t_i^{\mathtt{q}}, 0)} (\boldsymbol{t}_i, 0, \widehat{\boldsymbol{y}}_i, \widehat{\boldsymbol{w}}_i) & \text{if } i = n \end{cases}$$

- $\widehat{\chi}_i$ is obtained from χ_i by keeping the same actions, and lifting the configurations to $[\![\widehat{\mathcal{S}}]\!]$. Formally,

$$\widehat{\chi}_i \quad = \quad \begin{cases} (\boldsymbol{t}_i, \bot, \widehat{\boldsymbol{y}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{*} (\boldsymbol{u}_i, \bot, \widehat{\boldsymbol{z}}_i, \widehat{\boldsymbol{w}}_{i+1}) & \text{if } i < n \\ (\boldsymbol{t}_i, 0, \widehat{\boldsymbol{y}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{*} (\boldsymbol{u}_i, 0, \widehat{\boldsymbol{z}}_i, \widehat{\boldsymbol{w}}_{i+1}) & \text{if } i = n \end{cases}$$

- $\widehat{\xi}_i$ is obtained from ξ_i by internalizing it into a single transition of $\widehat{\mathcal{M}^{\mathtt{p}}}$. Formally,

$$\widehat{\xi}_i \quad = \quad (\boldsymbol{u}_{i-1}, \bot, \widehat{\boldsymbol{z}}_{i-1}, \widehat{\boldsymbol{w}}_i) \xrightarrow{\mathtt{add}(0)} (\boldsymbol{s}_i, =, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i)$$

By construction, $\widehat{\rho} = \widehat{\pi}_0 \cdot \widehat{\chi}_0 \cdot \widehat{\xi}_1 \cdot \widehat{\pi}_1 \cdot \widehat{\chi}_1 \cdots \widehat{\xi}_n \cdot \widehat{\pi}_n \cdot \widehat{\chi}_n$ is a full run of $[\![\widehat{\mathcal{S}}]\!]$.

Let us prove the converse, and assume that $[\![\widehat{\mathcal{S}}]\!]$ has a full run $\widehat{\rho}$. Without loss of generality, we may suppose that $[\![\widehat{\mathcal{S}}]\!]$ schedules $\mathtt{p}$ first. This means that $\widehat{\rho}$ may be decomposed as $\widehat{\rho} = \widehat{\pi}_0 \cdot \widehat{\chi}_0 \cdot \widehat{\xi}_1 \cdot \widehat{\pi}_1 \cdot \widehat{\chi}_1 \cdots \widehat{\xi}_n \cdot \widehat{\pi}_n \cdot \widehat{\chi}_n$ where

- each $\widehat{\pi}_i = (\boldsymbol{s}_i, =, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{\varphi_i(\mathtt{x})} (\boldsymbol{t}_i, \alpha_i, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i)$ is a move of $\mathtt{p}$,
- each $\widehat{\chi}_i = (\boldsymbol{t}_i, \alpha_i, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{*} (\boldsymbol{u}_i, \alpha_i, \widehat{\boldsymbol{x}}_{i+1}, \widehat{\boldsymbol{w}}_{i+1})$ contains no configuration with $=$, and
- each $\widehat{\xi}_i = (\boldsymbol{u}_{i-1}, \alpha_{i-1}, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i) \xrightarrow{\mathtt{add}(0)} (\boldsymbol{s}_i, =, \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{w}}_i)$.

Note that, by construction of $\widehat{\mathcal{M}^{\mathtt{p}}}$, $\alpha_n = 0$ and $\alpha_i = \bot$ for all $i \in \{0, \ldots, n-1\}$. We construct a full run $\rho = \pi_0 \cdot \chi_0 \cdot \xi_1 \cdot \pi_1 \cdot \chi_1 \cdots \xi_n \cdot \pi_n \cdot \chi_n$ of $[\![\mathcal{S}]\!]$ as follows. We let $\boldsymbol{x}_i$ denote the extension of $\widehat{\boldsymbol{x}}_i$ to P, defined by $x_i^{\mathtt{q}} = x_i^{\mathtt{p}}$. Similarly, $\boldsymbol{w}_i$ is the extension of $\widehat{\boldsymbol{w}}_i$ to C, defined by $w_i^c = \varepsilon$. The runs π_i, χ_i and ξ_i of $[\![\mathcal{S}]\!]$ are defined by:

- Since $(s_i^{\mathtt{p}}, s_i^{\mathtt{q}}, =) \xrightarrow{\varphi_i(\mathtt{x})} (t_i^{\mathtt{p}}, t_i^{\mathtt{q}}, \alpha_i)$, there exists $y_i^{\mathtt{q}} \in \mathbb{N}$ and a local run μ_i in $\mathcal{M}^{\mathtt{q}}$, with only actions in A_{cnt}, from $(s_i^{\mathtt{q}}, x_i^{\mathtt{q}})$ to $(t_i^{\mathtt{q}}, y_i^{\mathtt{q}})$. Furthermore, $y_i^{\mathtt{q}} = 0$ if $i = n$, since $\alpha_n = 0$. Define $y_i^p = x_i^p$ for all $p \in \widehat{P}$. Note, also, that $s_i^p = t_i^p$ for all $p \in \widehat{P}$. Therefore, the local run μ_i of $\mathcal{M}^{\mathtt{q}}$ is lifted to a run $\pi_i = (\boldsymbol{s}_i, \boldsymbol{x}_i, \boldsymbol{w}_i) \xrightarrow{*} (\boldsymbol{t}_i, \boldsymbol{y}_i, \boldsymbol{w}_i)$ of $[\![\mathcal{S}]\!]$.
- Since $\alpha_i \in \{0, \bot\}$, $\widehat{\chi}_i$ contains only transitions that originate from rules of the one-counter machines $\mathcal{M}^p$, with $p \in \widehat{P}$. Therefore, $\widehat{\chi}_i$ is lifted to a run $\chi_i = (\boldsymbol{t}_i, \boldsymbol{y}_i, \boldsymbol{w}_i) \xrightarrow{*} (\boldsymbol{u}_i, \boldsymbol{z}_i, \boldsymbol{w}_{i+1})$ of $[\![\mathcal{S}]\!]$, where $z_i^{\mathtt{q}} = y_i^{\mathtt{q}}$ and $z_i^p = x_{i+1}^p$ for all $p \in \widehat{P}$.
- Recall that $\mathsf{typ}(\mathtt{q}, \mathtt{c}) = \circ$ and $w_i^c = \varepsilon$. Since $(u_{i-1}^{\mathtt{p}}, \mathtt{c}\,!, s_i^{\mathtt{p}}) \in \Delta^{\mathtt{p}}$ and $(u_{i-1}^{\mathtt{q}}, \mathtt{c}\,?, s_i^{\mathtt{q}}) \in \Delta^{\mathtt{q}}$, we obtain that $\xi_i = (\boldsymbol{u}_{i-1}, \boldsymbol{z}_{i-1}, \boldsymbol{w}_i) \xrightarrow{\mathtt{c}\,!} \cdot \xrightarrow{\mathtt{c}\,?} (\boldsymbol{s}_i, \boldsymbol{x}_i, \boldsymbol{w}_i)$ is a run of $[\![\mathcal{S}]\!]$.

By construction, $\rho = \pi_0 \cdot \chi_0 \cdot \xi_1 \cdot \pi_1 \cdot \chi_1 \cdots \xi_n \cdot \pi_n \cdot \chi_n$ is a full run of $[\![\mathcal{S}]\!]$. ◀

▶ **Lemma 2.5.** *Assume that* $\mathtt{q} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{p}$. *If* $\mathtt{p}$ *has type* $\bullet$ *on* $\mathtt{c}$ *and* $\mathtt{q}$ *has type* $\circ$ *on* $\mathtt{c}$ *then* Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-Sc1cm($\mathcal{U}$).

Proof. We proceed as for Lemma 2.4. However, instead of scheduling $\mathtt{q}$ first, we now schedule it last. Naturally, this impacts the definition of the "product" machine $\widehat{\mathcal{M}}^{\mathtt{p}}$. The set of states remains $\widehat{S}^{\mathtt{p}} = S^{\mathtt{p}} \times S^{\mathtt{q}} \times \{0, \bot, =\}$. But the sets of initial states and final states become:

$$\widehat{I}^{\mathtt{p}} \;=\; I^{\mathtt{p}} \times I^{\mathtt{q}} \times \{0\} \qquad \text{and} \qquad \widehat{F}^{\mathtt{p}} \;=\; F^{\mathtt{p}} \times F^{\mathtt{q}} \times \{=\}$$

Furthermore, the set of transition rules $\widehat{\Delta}^{\mathtt{p}}$ accounts for the reversed roles of $\mathtt{p}$ and $\mathtt{q}$ regarding communication over the channel $\mathtt{c}$. For every $s_1^{\mathtt{p}}, s_2^{\mathtt{p}} \in S^{\mathtt{p}}$, $s_1^{\mathtt{q}}, s_2^{\mathtt{q}} \in S^{\mathtt{q}}$ and $\alpha \in \{0, \bot\}$,

$$\begin{array}{lcll}
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, \alpha) & \overset{a}{\hookrightarrow} & (s_2^{\mathtt{p}}, s_1^{\mathtt{q}}, \alpha) & \text{if} \quad (s_1^{\mathtt{p}}, a, s_2^{\mathtt{p}}) \in \Delta^{\mathtt{p}} \;\wedge\; a \neq \mathtt{c}\,? \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, =) & \xhookrightarrow{\mathsf{add}(0)} & (s_2^{\mathtt{p}}, s_2^{\mathtt{q}}, \bot) & \text{if} \quad (s_1^{\mathtt{p}}, \mathtt{c}\,?, s_2^{\mathtt{p}}) \in \Delta^{\mathtt{p}} \;\wedge\; (s_1^{\mathtt{q}}, \mathtt{c}\,!, s_2^{\mathtt{q}}) \in \Delta^{\mathtt{q}} \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, 0) & \xhookrightarrow{\varphi(\mathtt{x})} & (s_1^{\mathtt{p}}, s_2^{\mathtt{q}}, =) & \text{if} \quad \varphi(\mathtt{x}) \;=\; (s_1^{\mathtt{q}}, 0) \rightsquigarrow_{\mathtt{q}} (s_2^{\mathtt{q}}, \mathtt{x}) \\
(s_1^{\mathtt{p}}, s_1^{\mathtt{q}}, \bot) & \xhookrightarrow{\varphi(\mathtt{x})} & (s_1^{\mathtt{p}}, s_2^{\mathtt{q}}, =) & \text{if} \quad \varphi(\mathtt{x}) \;=\; \exists \mathtt{y} \cdot (s_1^{\mathtt{q}}, \mathtt{y}) \rightsquigarrow_{\mathtt{q}} (s_2^{\mathtt{q}}, \mathtt{x})
\end{array}$$

The proof that $[\![\mathcal{S}]\!]$ has a full run if and only if $[\![\widehat{\mathcal{S}}]\!]$ has a full run is similar to the proof of Lemma 2.4, except that, as mentioned above, $\mathtt{q}$ is now scheduled last. ◀

B.3 Two processes connected by one channel

Let us recall the context of Subsection **??**. We consider the topology $\mathcal{T} = \langle \{\mathtt{p}, \mathtt{q}\}, \{\mathtt{c}\}, \mathsf{src}, \mathsf{dst} \rangle$ with $\mathsf{src}(\mathtt{c}) = (\mathtt{p}, \bullet)$ and $\mathsf{dst}(\mathtt{c}) = (\mathtt{q}, \bullet)$. It is assumed that $\mathtt{p} \neq \mathtt{q}$.

▶ **Lemma 2.6.** Rp-Sc1cm($\mathcal{T}$) *is reducible to* Rp-1cm.

Proof. Let $\mathcal{S} = \langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ be a system of communicating one-counter machines, with topology $\mathcal{T}$. Recall that $\mathcal{T}$ has a exactly two processes, $\mathtt{p}$ and $\mathtt{q}$, and a unique channel $\mathtt{p} \overset{\mathtt{c}}{\rightarrowtail} \mathtt{q}$. We construct, from $\mathcal{S}$, a one-counter machine $\mathcal{N}$ that simulates the "product" of $\mathtt{p}$ and $\mathtt{q}$. Let us start with an informal description of this construction. As in the proof of Lemma 4.3, we schedule the sender last (here, $\mathtt{p}$) and the receiver first (here, $\mathtt{q}$). Thus, emissions $\mathtt{c}\,!$ and receptions $\mathtt{c}\,?$ occur consecutively, with no other action in between. Since $\mathtt{p}$ and $\mathtt{q}$ have type $\bullet$ on $\mathtt{c}$, each sequence of actions $\mathtt{c}\,! \cdot \mathtt{c}\,?$ may occur only if $\mathtt{p}$'s counter and $\mathtt{q}$'s counter hold the same value. So $\mathcal{N}$ internalizes each synchronization $\mathtt{c}\,! \cdot \mathtt{c}\,?$, and simulates, in one step, the behavior of $\mathtt{p}$ and $\mathtt{q}$ from one synchronization to the next. This is possible thanks to Theorem 4.2, which entails that counter reachability relations are (effectively) closed under intersection.

We now provide the formal definition of $\mathcal{N}$. Let $\mathcal{M}^{\mathsf{p}}_{\mathrm{cnt}}$ and $\mathcal{M}^{\mathsf{q}}_{\mathrm{cnt}}$ denote the one-counter machines obtained, from $\mathcal{M}^{\mathsf{p}}$ and $\mathcal{M}^{\mathsf{q}}$, respectively, by removing all transition rules labeled with communication actions. We introduce the set $\boldsymbol{S} = S^{\mathsf{p}} \times S^{\mathsf{q}}$ of global states. Consider a pair $(\boldsymbol{s}, \boldsymbol{t})$ of global states. According to Theorem 4.2, the counter reachability relation of $\mathcal{M}^{\mathsf{p}}_{\mathrm{cnt}}$ with initial state s^{p} and final state t^{p} is defined by a one-counter Presburger predicate ψ^{p}. Likewise, the counter reachability relation of $\mathcal{M}^{\mathsf{q}}_{\mathrm{cnt}}$ with initial state s^{q} and final state t^{q} is defined by a one-counter Presburger predicate ψ^{q}. Therefore, by Theorem 4.2, the binary relation defined by $\psi^{\mathsf{p}} \wedge \psi^{\mathsf{q}}$ is the counter reachability relation of a one-counter machine $\mathcal{N}_{\boldsymbol{s},\boldsymbol{t}} = \langle S_{\boldsymbol{s},\boldsymbol{t}}, I_{\boldsymbol{s},\boldsymbol{t}}, F_{\boldsymbol{s},\boldsymbol{t}}, A_{\boldsymbol{s},\boldsymbol{t}}, \Delta_{\boldsymbol{s},\boldsymbol{t}} \rangle$. Moreover, $\mathcal{N}_{\boldsymbol{s},\boldsymbol{t}}$ is computable from $\mathcal{M}^{\mathsf{p}}$ and $\mathcal{M}^{\mathsf{q}}$. By construction, for every $\boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{S}$ and $x, y \in \mathbb{N}$, the two following assertions are equivalent:

- there is a run from (s^{p}, x) to (t^{p}, y) in $[\![\mathcal{M}^{\mathsf{p}}_{\mathrm{cnt}}]\!]$ and a run from (s^{q}, x) to (t^{q}, y) in $[\![\mathcal{M}^{\mathsf{q}}_{\mathrm{cnt}}]\!]$,
- for some $r \in I_{\boldsymbol{s},\boldsymbol{t}}$ and $u \in F_{\boldsymbol{s},\boldsymbol{t}}$, there is a run from (r, x) to (u, y) in $[\![\mathcal{N}_{\boldsymbol{s},\boldsymbol{t}}]\!]$.

We may assume, without loss of generality, that the sets of states $S_{\boldsymbol{s},\boldsymbol{t}}$, for $\boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{S}$, are pairwise disjoint, as well as disjoint from $\boldsymbol{S}$. The one-counter machine $\mathcal{N} = \langle S, I, F, A, \Delta \rangle$ is obtained by linking the one-counter machines $\mathcal{N}_{\boldsymbol{s},\boldsymbol{t}}$ together. Its sets of states, initial states and final states are:

$$S = (S^{\mathsf{p}} \times S^{\mathsf{q}}) \cup \bigcup_{\boldsymbol{s},\boldsymbol{t} \in \boldsymbol{S}} S_{\boldsymbol{s},\boldsymbol{t}} \qquad \text{and} \qquad I = I^{\mathsf{p}} \times I^{\mathsf{q}} \qquad \text{and} \qquad F = F^{\mathsf{p}} \times F^{\mathsf{q}}$$

The set of actions of $\mathcal{N}$ and its set of transition rules are:

$$A = \{\mathtt{add}(0)\} \cup \bigcup_{\boldsymbol{s},\boldsymbol{t} \in \boldsymbol{S}} A_{\boldsymbol{s},\boldsymbol{t}} \qquad \text{and} \qquad \Delta = \Delta_{\mathrm{link}} \cup \Delta_{\mathrm{com}} \cup \bigcup_{\boldsymbol{s},\boldsymbol{t} \in \boldsymbol{S}} \Delta_{\boldsymbol{s},\boldsymbol{t}}$$

where Δ_{link} and Δ_{com} are defined by

$$\begin{aligned} \Delta_{\mathrm{link}} &= \bigcup_{\boldsymbol{s},\boldsymbol{t} \in \boldsymbol{S}} \{\boldsymbol{s}\} \times \{\mathtt{add}(0)\} \times I_{\boldsymbol{s},\boldsymbol{t}} \cup F_{\boldsymbol{s},\boldsymbol{t}} \times \{\mathtt{add}(0)\} \times \{\boldsymbol{t}\} \\ \Delta_{\mathrm{com}} &= \{(\boldsymbol{s}, \mathtt{add}(0), \boldsymbol{t}) \mid \boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{S} \wedge (s^{\mathsf{p}}, \mathsf{c}\,!, s^{\mathsf{p}}) \in \Delta^{\mathsf{p}} \wedge (t^{\mathsf{q}}, \mathsf{c}\,?, t^{\mathsf{q}}) \in \Delta^{\mathsf{q}}\} \end{aligned}$$

It remains to show that $[\![\mathcal{S}]\!]$ has a full run if and only if $[\![\mathcal{N}]\!]$ has a full run. Assume that $[\![\mathcal{S}]\!]$ has a full run ρ. Without loss of generality, we may assume that ρ schedules each reception $\mathsf{c}\,?$ immediately after the matching emission $\mathsf{c}\,!$. This means, by standard partial-order techniques, that ρ may be decomposed as $\rho = \pi_0 \cdot \xi_1 \cdot \pi_1 \cdots \xi_n \cdot \pi_n$ where

- each $\pi_i = (\boldsymbol{s}_i, \boldsymbol{x}_i, \varepsilon) \xrightarrow{*} (\boldsymbol{t}_i, \boldsymbol{y}_i, \varepsilon)$ contains no communication action,
- each $\xi_i = (\boldsymbol{t}_{i-1}, \boldsymbol{y}_{i-1}, \varepsilon) \xrightarrow{\mathsf{c}\,!} \cdot \xrightarrow{\mathsf{c}\,?} (\boldsymbol{s}_i, \boldsymbol{x}_i, \varepsilon)$.

The definition of ξ_i entails that $x^{\mathsf{q}}_i = y^{\mathsf{p}}_{i-1}$ for all $i \in \{1, \ldots, n\}$, by definition of ξ_i. Since $\mathsf{typ}(\mathsf{p}, \mathsf{c}) = \mathsf{typ}(\mathsf{q}, \mathsf{c}) = \bullet$, it also holds that $\boldsymbol{x}_i = \boldsymbol{y}_{i-1}$. We get that $x^{\mathsf{p}}_i = x^{\mathsf{q}}_i$ and $y^{\mathsf{p}}_i = y^{\mathsf{q}}_i$ for all $i \in \{0, \ldots, n\}$. We let x_i and y_i denote the common values $x^{\mathsf{p}}_i = x^{\mathsf{q}}_i$ and $y^{\mathsf{p}}_i = y^{\mathsf{q}}_i$, respectively. By construction, $[\![\mathcal{N}_{\boldsymbol{s},\boldsymbol{t}}]\!]$ contains a run $(r_i, x_i) \xrightarrow{*} (u_i, y_i)$, for some $r_i \in I_{\boldsymbol{s}_i,\boldsymbol{t}_i}$ and $u_i \in F_{\boldsymbol{s}_i,\boldsymbol{t}_i}$. It follows that $[\![\mathcal{N}]\!]$ contains a run $(\boldsymbol{s}_i, x_i) \xrightarrow{\mathtt{add}(0)} (r_i, x_i) \xrightarrow{*} (u_i, y_i) \xrightarrow{\mathtt{add}(0)} (\boldsymbol{t}_i, y_i)$. Moreover, by definition of Δ_{com}, there is also a run from $(\boldsymbol{t}_{i-1}, y_{i-1})$ to $(\boldsymbol{s}_i, x_i)$ in $[\![\mathcal{N}]\!]$ for all $i \in \{1, \ldots, n\}$. Thus, we have shown that $\mathcal{N}$ contains a run χ from $(\boldsymbol{s}_0, x_0)$ to $(\boldsymbol{t}_n, y_n)$. Since $\boldsymbol{s}_0 \in I$, $\boldsymbol{t}_n \in F$ and $x_0 = y_n = 0$, χ is a full run.

Let us prove the converse, and assume that $[\![\mathcal{N}]\!]$ has a full run χ. Obviously, χ may be decomposed as $\chi = \chi_0 \cdots \chi_n$ where each χ_i is a run $\chi_i = (\boldsymbol{s}_i, x_i) \xrightarrow{*} (\boldsymbol{s}_{i+1}, x_{i+1})$ with $\boldsymbol{s}_i, \boldsymbol{s}_{i+1}$ in $\boldsymbol{S}$ and no intermediate state in $\boldsymbol{S}$. Let $\boldsymbol{x}_i \in \mathbb{N}^P$ be defined by $x^{\mathsf{p}}_i = x^{\mathsf{q}}_i = x_i$. By construction of $\mathcal{N}$, for every $i \in \{0, \ldots, n\}$, the run χ_i is

- either reduced to a single configuration $(\boldsymbol{s}_i, x_i) = (\boldsymbol{s}_{i+1}, x_{i+1})$. In that case, $[\![\mathcal{S}]\!]$ contains the run reduced to the single configuration $(\boldsymbol{s}_i, \boldsymbol{x}_i, \varepsilon) = (\boldsymbol{s}_{i+1}, \boldsymbol{x}_{i+1}, \varepsilon)$.

- or single transition $(\boldsymbol{s}_i, x_i) \xrightarrow{\mathtt{add}(0)} (\boldsymbol{s}_{i+1}, x_{i+1})$. In that case, $(\boldsymbol{s}_i, \mathtt{add}(0), \boldsymbol{s}_{i+1})$ must be a transition rule in Δ_{com}. Therefore, $(\boldsymbol{s}_i, \boldsymbol{x}_i, \varepsilon) \xrightarrow{\mathtt{c}\,!} \cdot \xrightarrow{\mathtt{c}\,?} (\boldsymbol{s}_{i+1}, \boldsymbol{x}_{i+1}, \varepsilon)$ is a run in $[\![\mathcal{S}]\!]$.
- or a run $(\boldsymbol{s}_i, x_i) \xrightarrow{\mathtt{add}(0)} (r_i, x_i) \xrightarrow{*} (u_i, x_{i+1}) \xrightarrow{\mathtt{add}(0)} (\boldsymbol{s}_{i+1}, x_{i+1})$. In that case, $r_i \in I_{\boldsymbol{s}_i, \boldsymbol{s}_{i+1}}$, $u_i \in F_{\boldsymbol{s}_i, \boldsymbol{s}_{i+1}}$ and $(r_i, x_i) \xrightarrow{*} (u_i, x_{i+1})$ is a run in $[\![\mathcal{N}_{\boldsymbol{s}_i, \boldsymbol{s}_{i+1}}]\!]$. Therefore, there is a run from $(s_i^{\mathtt{p}}, x_i)$ to $(s_{i+1}^{\mathtt{p}}, x_{i+1})$ in $[\![\mathcal{M}_{\mathrm{cnt}}^{\mathtt{p}}]\!]$ and a run from $(s_i^{\mathtt{q}}, x_i)$ to $(s_{i+1}^{\mathtt{q}}, x_{i+1})$ in $[\![\mathcal{M}_{\mathrm{cnt}}^{\mathtt{q}}]\!]$. It follows that $[\![\mathcal{S}]\!]$ contains a run from $(\boldsymbol{s}_i, \boldsymbol{x}_i, \varepsilon)$ to $(\boldsymbol{s}_{i+1}, \boldsymbol{x}_{i+1}, \varepsilon)$.

Thus, we have shown that $[\![\mathcal{S}]\!]$ contains a run ρ from $(\boldsymbol{s}_0, \boldsymbol{x}_0, \varepsilon)$ to $(\boldsymbol{s}_{n+1}, \boldsymbol{x}_{n+1}, \varepsilon)$. Since $\boldsymbol{s}_0 \in I$, $\boldsymbol{s}_{n+1} \in F$ and $\boldsymbol{x}_0 = \boldsymbol{x}_{n+1} = \mathbf{0}$, ρ is a full run. ◀

C Proofs of Section 5

▶ **Proposition 5.2.** *Given a strongly-connected topology $\mathcal{T}$,* Rp-Sc1cm-eager($\mathcal{T}$) *is decidable if and only if $\mathcal{T}$ contains at most two processes.*

Proof. We provide the "only if"-direction of the proposition in the following in more detail than before by polynomially reducing the reachability problem of two-counter machines to the reachability problem of systems of communicating one-counter machines. The underlying idea is to use the restriction to eager runs to "force" a certain communication behavior that allows to implement a protocol that gives one distinct process always access to two counters that are stored and passed inside the topology between the other processes without getting lost.

We first remark that any graph that contains a strongly connected component of size ≥ 3 contains at least:

- a directed simple cycle of size ≥ 3, or
- two directed simple cycles of size 2 with a common process.

Now, we prove that the reachability problem is undecidable in both cases. Because of Proposition 3.4, we only have to consider channels with $\circ$-typed endpoints.

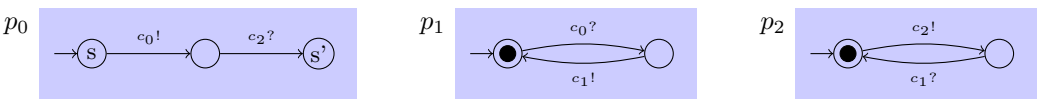
Case 1: Directed Cycles of length ≥ 3

As any two-counter machine is directly equivalent to a one counter machine enhanced with an additional integer value register and a operation that switches between counter and register, we will reduce the reachability problem of this machine to our setting. Let $\mathcal{M}$ be a one-register one-counter machine with an additional action for switching counter and register.

Let $\mathcal{T}$ be $p_0 \xrightarrow{c_0} p_1 \xrightarrow{c_1} p_2 \xrightarrow{c_2} p_0$ for $p_0, p_1, p_2 \in P$ and $c_0, c_1, c_2 \in C$. We introduce a system of communicating one-counter machines $\mathcal{S}$ that simulates $\mathcal{M}$ as follows

- p_0 locally simulates the one-register one-counter machine $\mathcal{M}$,
- p_1 acts as a buffer,
- p_2 simulates the register.

As before, p_0 locally simulates all counter actions of $\mathcal{M}$. We need to show how to implement a way to switch the counters of p_0 and p_2 that can be only be provoked by p_0, i.e., p_0 "remote controls" p_1 and p_2. This is possible due to the restriction to eager runs. Let us p_1 and p_2 be given as follows, and let us consider a sequence of two transitions of process p_0 from local state s to s':



Assume that when all processes above are in it's initial state, the counter values are $\langle a, -, b\rangle$, where $-$ denotes a random value. Due to our assumption of eagerness, the only firable sequence of transition when p_0 goes from s to s' is as follows:

1. p_0 sends its counter to p_1, and the counters' values are $\langle -, a, b\rangle$;
2. p_2 sends its counter to p_0, and the counters' values are $\langle b, a, -\rangle$;
3. p_1 sends its counter to p_2, and the counters' values are $\langle b, -, a\rangle$.

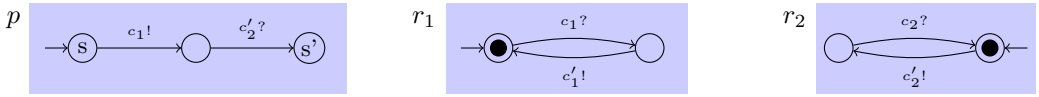
Note that p_0 can continue its calculation before the last operation of the switching sequence is done. Thus, we replace a switching transition of $\mathcal{M}$ by the above sequence of transitions from s to s'.

A routine induction shows that $\mathcal{M}$ can be simulated by $\mathcal{S}$ which is only polynomially larger. If $\mathcal{T}$ contains a directed cycle of more than three processes, we can use the previous reduction analogously while adding more "buffer" processes.

Case 2: Two cycles of size 2 with a common process

We will show how to simulate a two-counter machine $\mathcal{M}$ on a system of communicating one-counter machines $\mathcal{S}_{\mathcal{M}}$ on the topology $\mathcal{T}$ given by $p \xrightarrow{c_1} r_1 \xrightarrow{c_1'} p$ and $p \xrightarrow{c_2} r_2 \xrightarrow{c_2'} p$ (two direct cycles sharing a common node). Processes r_1 and r_2 act as two separate registers for $\mathcal{M}$'s counters 1 and 2, while p simulates the actions of $\mathcal{M}$ locally on its counter and stores in its control structure which of the two counters it currently holds. When an action on the other counter is to be done, it "stores" the current counter in its appropriate register and "loads" the other counter's value from the other register. As before, the realization of this load-store protocol relies on the underlying eagerness restriction.

Consider r_1, r_2 given as follows and the transition of p from s to s'.



We write the values of local counters of p, r_1, r_2 as $\langle x_1, -, x_2\rangle$ and assert that p initially works on counter 1. Due to our eagerness assumption, the only firable global sequence of transition when p goes from s to s' is:

1. p sends its counter to r_1, and the counters' values are $\langle -, x_1, x_2\rangle$;
2. p_2 sends its counter to r_2, and the counters' values are $\langle x_2, x_1, -\rangle$;

After the switching sequence, p works with the with x_2 (thus counter 2), r_1's counter is an arbitrary value that waits to be overwritten by the next received message, while p_1 stores x_1 and is ready to send it on demand to p. In addition, we assure that p initially and finally stores counter 1 locally, and counter 2 in r_2. Thus, process p replaces $\mathcal{M}$'s switching between the register and the counter by the previous protocol.

So $\mathcal{M}$ can be simulated by $\mathcal{S}$ and, thus, we can reduce the reachability problem from two-counter a given two-counter machine $\mathcal{M}$ to the reachability problem of $\mathcal{S}_{\mathcal{M}}$. Note that $\mathcal{S}_{\mathcal{M}}$ is only polynomially larger than $\mathcal{M}$. ◀