# Safety Analysis of Parameterised Networks with Non-Blocking Rendez-Vous

**Lucie Guillou**
IRIF, CNRS, Université Paris Cité, France

**Arnaud Sangnier**
IRIF, CNRS, Université Paris Cité, France

**Nathalie Sznajder**
LIP6, CNRS, Sorbonne Université, France

**Abstract**

We consider networks of processes that all execute the same finite-state protocol and communicate via a rendez-vous mechanism. When a process requests a rendez-vous, another process can respond to it and they both change their control states accordingly. We focus here on a specific semantics, called non-blocking, where the process requesting a rendez-vous can change its state even if no process can respond to it. We study the parameterised coverability problem of a configuration in this context, which consists in determining whether there is an initial number of processes and an execution allowing to reach a configuration bigger than a given one. We show that this problem is EXPSPACE-complete and can be solved in polynomial time if the protocol is partitioned into two sets of states, the states from which a process can request a rendez-vous and the ones from which it can answer one. We also prove that the problem of the existence of an execution bringing all the processes in a final state is undecidable in our context. These two problems can be solved in polynomial time with the classical rendez-vous semantics.

**2012 ACM Subject Classification** Theory of computation → Formal languages and automata theory

**Keywords and phrases** Parameterised verification, Coverability, Counter machines

**Digital Object Identifier** 10.4230/LIPIcs..2023.

## 1 Introduction

*Verification of distributed/concurrent systems.* Because of their ubiquitous use in applications we rely on constantly, the development of formal methods to guarantee the correct behaviour of distributed/concurrent systems has become one of the most important research directions in the field of computer systems verification in the last two decades. Unfortunately, such systems are difficult to analyse for several reasons. Among others, we can highlight two aspects that make the verification process tedious. First, these systems often generate a large number of different executions due to the various interleavings generated by the concurrent behaviours of the entities involved. Understanding how these interleavings interact is a complex task and can often lead to errors at the design-level or make the model of these systems very complex. Second, in some cases, the number of participants in a distributed system may be unbounded and not known a priori. To fully guarantee the correctness of such systems, the analysis would have to be performed for all possible instances of the system, i.e., an infinite number of times. As a consequence, classical techniques to verify finite state systems, like testing or model-checking, cannot be easily adapted to distributed systems and it is often necessary to develop new techniques.

*Parameterised verification.* When designing systems with an unbounded number of participants, one often provides one schematic program (or protocol) intended to be implemented by multiple identical processes, parameterised by the number of participants. In general, even if the verification problem is decidable for a given instance of the parameter, verifying


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

all possible instances is undecidable ([3]). However, several parameters come into play that can be adjusted to allow automatic verification. One key aspect to obtain decidability is to assume that the processes do not manipulate identities and use simple communication mechanisms like pairwise synchronisation (or rendez-vous) [13], broadcast of a message to all the entities [10] (which can as well be lossy in order to simulate mobility [6]), shared register containing values of a finite set [11], and so on (see [9] for a survey). In all the aforementioned cases, all the entities execute the same protocol given by a finite state automaton. Note that parameterised verification, when decidable like in the above models, is also sometimes surprisingly easy, compared to the same problem with a fixed number of participants. For instance, liveness verification of parameterised systems with shared memory is PSPACE-complete for a fixed number of processes and in NP when parameterised [7].

*Considering rendez-vous communication.* In one of the seminal papers for the verification of parameterised networks [13], German and Sistla (and since then [4, 14]) assume that the entities communicate by “rendez-vous”, a synchronisation mechanism in which two processes (the *sender* and the *receiver*) agree on a common action by which they jointly change their local state. This mechanism is synchronous and symmetric, meaning that if no process is ready to receive a message, the sender cannot send it. However, in some applications, such as Java Thread programming, this is not exactly the primitive that is implemented. When a Thread is suspended in a waiting state, it is woken up by the reception of a message `notify` sent by another Thread. However, the sender is not blocked if there is no suspended Thread waiting for its message; in this case, the sender sends the `notify` anyway and the message is simply lost. This is the reason why Delzanno et. al. have introduced *non-blocking* rendez-vous in [5] a communication primitive in which the sender of a message is not blocked if no process receives it. One of the problems of interest in parameterised verification is the coverability problem: is it possible that, starting from an initial configuration, (at least) one process reaches a bad state? In [5], and later in [19], the authors introduce variants of Petri nets to handle this type of communication. In particular, the authors investigate in [19] the coverability problem for an extended class of Petri nets with non-blocking arcs, and show that for this model the coverability problem is decidable using the techniques of Well-Structured Transitions Systems [1, 2, 12]. However, since their model is an extension of Petri nets, the latter problem is EXPSPACE-hard [16] (no upper bound is given). Relying on Petri nets to obtain algorithms for parameterised networks is not always a good option. In fact, the coverability problem for parameterised networks with rendez-vous is in P[13], while it is EXPSPACE-complete for Petri nets [18, 16]. Hence, no upper bound or lower bound can be directly deduced for the verification of networks with non-blocking rendez-vous from [19].

*Our contributions.* We show that the coverability problem for parameterised networks with *non-blocking rendez-vous communication* over a finite alphabet is EXPSPACE-complete. To obtain this result, we consider an extension of counter machines (without zero test) where we add non-blocking decrement actions and edges that can bring back the machine to its initial location at any moment. We show that the coverability problem for these extended counter machines is EXPSPACE-complete (Section 3) and that it is equivalent to our problem over parameterised networks (Section 4). We consider then a subclass of parameterised networks – *wait-only protocols* – in which no state can allow to both request a rendez-vous and wait for one. This restriction is very natural to model concurrent programs since when a thread is waiting, it cannot perform any other action. We show that coverability problem can then be solved in polynomial time (Section 5). Finally, we show that the synchronization problem, where we look for a reachable configuration with all the processes in a given state, is undecidable in our framework, even for wait-only protocols (Section 6).

Due to lack of space, some proofs are only given in the appendix.

# 2 Rendez-vous Networks with Non-Blocking Semantics

For a finite alphabet $\Sigma$, we let $\Sigma^*$ denote the set of finite sequences over $\Sigma$ (or words). Given $w \in \Sigma^*$, we let $|w|$ denote its length: if $w = w_0 \dots w_{n-1} \in \Sigma^*$, then $|w| = n$. We write $\mathbb{N}$ to denote the set of natural numbers and $[i,j]$ to represent the set $\{k \in \mathbb{N} \mid i \leq k \text{ and } k \leq j\}$ for $i,j \in \mathbb{N}$. For a finite set $E$, the set $\mathbb{N}^E$ represents the multisets over $E$. For two elements $m, m' \in \mathbb{N}^E$, we denote $m + m'$ the multiset such that $(m+m')(e) = m(e) + m'(e)$ for all $e \in E$. We say that $m \leq m'$ if and only if $m(e) \leq m'(e)$ for all $e \in E$. If $m \leq m'$, then $m' - m$ is the multiset such that $(m' - m)(e) = m'(e) - m(e)$ for all $e \in E$. Given a subset $E' \subseteq E$ and $m \in \mathbb{N}^E$, we denote by $||m||_{E'}$ the sum $\Sigma_{e \in E'} m(e)$ of elements of $E'$ present in $m$. The size of a multiset $m$ is given by $||m|| = ||m||_E$. For $e \in E$, we use sometimes the notation $e$ for the multiset $m$ verifying $m(e) = 1$ and $m(e') = 0$ for all $e' \in E \setminus \{e\}$ and, to represent for instance the multiset with four elements $a, b, b$ and $c$, we will also use the notations $\wr a, b, b, c \int$ or $\wr a, 2 \cdot b, c \int$.

## 2.1 Rendez-Vous Protocols

We can now define our model of networks. We assume that all processes in the network follow the same protocol. Communication in the network is pairwise and is performed by *rendez-vous* through a finite communication alphabet $\Sigma$. Each process can either perform an internal action using the primitive $\tau$, or request a rendez-vous by sending the message $m$ using the primitive $!m$ or answer to a rendez-vous by receiving the message $m$ using the primitive $?m$ (for $m \in \Sigma$). Thus, the set of primitives used by our protocols is $RV(\Sigma) = \{\tau\} \cup \{?m, !m \mid m \in \Sigma\}$.

▶ **Definition 2.1** (Rendez-vous protocol). *A* rendez-vous protocol *(shortly protocol) is a tuple $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$ where $Q$ is a finite set of states, $\Sigma$ is a finite alphabet, $q_{in} \in Q$ is the initial state, $q_f \in Q$ is the final state and $T \subseteq Q \times RV(\Sigma) \times Q$ is the finite set of transitions.*

For a message $m \in \Sigma$, we denote by $R(m)$ the set of states $q$ from which the message $m$ can be received, i.e states $q$ such that there is a transition $(q, ?m, q') \in T$ for some $q' \in Q$.

A *configuration* associated to the protocol $\mathcal{P}$ is a non-empty multiset $C$ over $Q$ for which $C(q)$ denotes the number of processes in the state $q$ and $||C||$ denotes the total number of processes in the configuration $C$. A configuration $C$ is said to be *initial* if and only if $C(q) = 0$ for all $q \in Q \setminus \{q_{in}\}$. We denote by $\mathcal{C}(\mathcal{P})$ the set of configurations and by $\mathcal{I}(\mathcal{P})$ the set of initial configurations. Finally for $n \in \mathbb{N} \setminus \{0\}$, we use the notation $\mathcal{C}_n(\mathcal{P})$ to represent the set of configurations of size $n$, i.e. $\mathcal{C}_n(\mathcal{P}) = \{C \in \mathcal{C} \mid ||C|| = n\}$. When the protocol is made clear from the context, we shall write $\mathcal{C}$, $\mathcal{I}$ and $\mathcal{C}_n$.

We explain now the semantics associated with a protocol. For this matter we define the relation $\rightarrow_{\mathcal{P}} \subseteq \bigcup_{n \geq 1} \mathcal{C}_n \times \left(\{\tau\} \cup \Sigma \cup \{\mathbf{nb}(m) \mid m \in \Sigma\}\right) \times \mathcal{C}_n$ as follows. Given $n \in \mathbb{N} \setminus \{0\}$ and $C, C' \in \mathcal{C}_n$ and $m \in \Sigma$, we have:

1. $C \xrightarrow{\tau}_{\mathcal{P}} C'$ iff there exists $(q, \tau, q') \in T$ such that $C(q) > 0$ and $C' = C - \wr q \int + \wr q' \int$ **(internal)**;
2. $C \xrightarrow{m}_{\mathcal{P}} C'$ iff there exists $(q_1, !m, q_1') \in T$ and $(q_2, ?m, q_2') \in T$ such that $C(q_1) > 0$ and $C(q_2) > 0$ and $C(q_1) + C(q_2) \geq 2$ and $C' = C - \wr q_1, q_2 \int + \wr q_1', q_2' \int$ **(rendez-vous)**;
3. $C \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C'$ iff there exists $(q_1, !m, q_1') \in T$, such that $C(q_1) > 0$ and $(C - \wr q_1 \int)(q_2) = 0$ for all $(q_2, ?m, q_2') \in T$ and $C' = C - \wr q_1 \int + \wr q_1' \int$ **(non-blocking request)**.

Intuitively, from a configuration $C$, we allow the following behaviours: either a process takes an internal transition (labeled by $\tau$), or two processes synchronize over a rendez-vous $m$, or a process requests a rendez-vous to which no process can answer (non-blocking sending).

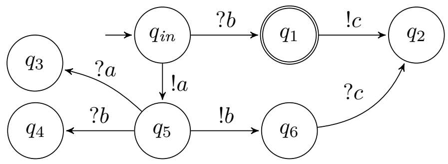


**Figure 1** Example of a rendez-vous protocol $\mathcal{P}$

This allows us to define $S_{\mathcal{P}}$ the transition system $(\mathcal{C}(\mathcal{P}), \rightarrow_{\mathcal{P}})$ associated to $\mathcal{P}$. We will write $C \rightarrow_{\mathcal{P}} C'$ when there exists $a \in \{\tau\} \cup \Sigma \cup \{\mathbf{nb}(m) \mid m \in \Sigma\}$ such that $C \xrightarrow{a}_{\mathcal{P}} C'$ and denote by $\rightarrow^*_{\mathcal{P}}$ the reflexive and transitive closure of $\rightarrow_{\mathcal{P}}$. Furthermore, when made clear from the context, we might simply write $\rightarrow$ instead of $\rightarrow_{\mathcal{P}}$. An *execution* is a finite sequence of configurations $\rho = C_0 C_1 \dots$ such that, for all $0 \leq i < |\rho|$, $C_i \rightarrow_{\mathcal{P}} C_{i+1}$, the execution is said to be initial if $C_0 \in \mathcal{I}(\mathcal{P})$.

▶ **Example 2.2.** Figure 1 provides an example of a rendez-vous protocol where $q_{in}$ is the initial state and $q_1$ the final state. A configuration associated to this protocol is for instance the multiset $\wr 2 \cdot q_1, 1 \cdot q_4, 1 \cdot q_5 \int$ and the following sequence represents an initial execution: $\wr 2 \cdot q_{in} \int \xrightarrow{\mathbf{nb}(a)} \wr q_{in}, q_5 \int \xrightarrow{b} \wr q_1, q_6 \int \xrightarrow{c} \wr 2 \cdot q_2 \int$.

▶ Remark 2.3. When we only allow behaviours of type **(internal)** and **(rendez-vous)**, this semantics corresponds to the classical rendez-vous semantics ([13, 4, 14]). In opposition, we will refer to the semantics defined here as the *non-blocking semantics* where a process is not *blocked* if it requests a rendez-vous and no process can answer to it. Note that all behaviours possible in the classical rendez-vous semantics are as well possible in the non-blocking semantics but the converse is false.

## 2.2 Verification Problems

We now present the problems studied in this work. For this matter, given a protocol $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$, we define two sets of final configurations. The first one $\mathcal{F}_{\exists}(\mathcal{P}) := \{C \in \mathcal{C}(\mathcal{P}) \mid C(q_f) > 0\}$ characterises the configurations where one of the processes is in the final state. The second one $\mathcal{F}_{\forall}(\mathcal{P}) := \{C \in \mathcal{C}(\mathcal{P}) \mid C(Q \setminus \{q_f\}) = 0\}$ represents the configurations where all the processes are in the final state. Here again, when the protocol is clear from the context, we might use the notations $\mathcal{F}_{\exists}$ and $\mathcal{F}_{\forall}$. We study three problems: the *state coverability problem* (SCover), the *configuration coverability* problem (CCover) and the *synchronization problem* (Synchro), which all take as input a protocol $\mathcal{P}$ and can be stated as follows:

| **Problem name** | **Question** |
|---|---|
| SCover | Are there $C_0 \in \mathcal{I}$ and $C_f \in \mathcal{F}_{\exists}$, such that $C_0 \rightarrow^* C_f$? |
| CCover | Given $C \in \mathcal{C}$, are there $C_0 \in \mathcal{I}$ and $C' \geq C$, such that $C_0 \rightarrow^* C'$? |
| Synchro | Are there $C_0 \in \mathcal{I}$ and $C_f \in \mathcal{F}_{\forall}$, such that $C_0 \rightarrow^* C_f$? |

▶ Remark 2.4. The difficulty in solving these problems lies in the fact that we are seeking for an initial configuration allowing a specific execution but the set of initial configurations is infinite. The difference between SCover and Synchro is that in the first one we ask for at least one process to end up in the final state whereas the second one requires all the processes to end in this state. Note that SCover is an instance of CCover but Synchro is not.

▶ **Example 2.5.** The rendez-vous protocol of Figure 1 is a positive instance of SCover, as shown in Example 2.2. However, this is not the case for the Synchro: if an execution brings a process in $q_2$, this process cannot be brought afterwards to $q_1$. If $q_2$ is the final state, $\mathcal{P}$ is now a positive instance of Synchro (see Example 2.2). Note that if the final state is $q_4$, $\mathcal{P}$ is not a positive instance of SCover anymore. In fact, the only way to reach a configuration with a process in $q_4$ is to put (at least) two processes in state $q_5$ as this is the only state from which one process can send the message $b$. However, this cannot happen, since from an initial configuration, the only available action consists in sending the message $a$ as a non-blocking request. Once there is one process in state $q_5$, any other attempt to put another process in this state will induce a reception of message $a$ by the process already in $q_5$, which will hence leave $q_5$. Finally, note that for any $n \in \mathbb{N}$, the configuration $\wr n \cdot q_3 \int$ is coverable, even if $\mathcal{P}$ with $q_3$ as final state is not a positive instance of Synchro.

## 3 Coverability for Non-Blocking Counter Machines

We first detour into new classes of counter machines, which we call *non-blocking counter machines* and *non-blocking counter machines with restore*, in which a new way of decrementing the counters is added to the classical one: a non-blocking decrement, which is an action that can always be performed. If the counter is strictly positive, it is decremented; otherwise it is let to 0. We show that the coverability of a control state in this model is Expspace-complete, and use this result to solve coverability problems in rendez-vous protocols.

To define counter machines, given a set of integer variables (also called counters) $X$, we use the notation $\mathsf{CAct}(X)$ to represent the set of associated actions given by $\{\mathtt{x+}, \mathtt{x-}, \mathtt{x=0} \mid \mathtt{x} \in X\} \cup \{\bot\}$. Intuitively, $\mathtt{x+}$ increments the value of the counter $\mathtt{x}$, while $\mathtt{x-}$ decrements it and $\mathtt{x=0}$ checks if it is equal to 0. We are now ready to state the syntax of this model.

▶ **Definition 3.1.** *A* counter machine *(shortly CM) is a tuple $M = (Loc, X, \Delta, \ell_{in})$ such that $Loc$ is a finite set of locations, $\ell_{in} \in Loc$ is an initial location, $X$ is a finite set of counters, and $\Delta \subseteq Loc \times \mathsf{CAct}(X) \times Loc$ is finite set of transitions.*

We will say that a CM is test-free (shortly test-free CM) whenever $\Delta \cap \{\mathtt{x=0} \mid \mathtt{x} \in X\} = \emptyset$. A configuration of a CM $M = (\mathrm{Loc}, X, \Delta, \ell_{in})$ is a pair $(\ell, v)$ where $\ell \in \mathrm{Loc}$ specifies the current location of the CM and $v : X \to \mathbb{N}$ associates to each counter a natural value. Given two configurations $(\ell, v)$ and $(\ell', v')$ and a transition $\delta \in \Delta$, we define $(\ell, v) \overset{\delta}{\rightsquigarrow}_M (\ell', v')$ if and only if $\delta = (\ell, op, \ell')$ and one of the following holds:

- $op = \bot$ and $v = v'$;
- $op = \mathtt{x+}$ and $v'(\mathtt{x}) = v(\mathtt{x}) + 1$ and $v'(\mathtt{x}') = v(\mathtt{x}')$ for all $\mathtt{x}' \in X \setminus \{\mathtt{x}\}$;
- $op = \mathtt{x-}$ and $v'(\mathtt{x}) = v(\mathtt{x}) - 1$ and $v'(\mathtt{x}') = v(\mathtt{x}')$ for all $\mathtt{x}' \in X \setminus \{\mathtt{x}\}$;
- $op = \mathtt{x=0}$ and $v(\mathtt{x}) = 0$ and $v'(\mathtt{x}') = v(\mathtt{x}')$ for all $\mathtt{x}' \in X$.

In order to simulate the non-blocking semantics of our rendez-vous protocols with counter machines, we extend the class of test-free CM with non-blocking decrement actions.

▶ **Definition 3.2.** *A* non-blocking test-free counter machine *(shortly NB-CM) is a tuple $M = (Loc, X, \Delta_b, \Delta_{nb}, \ell_{in})$ such that $(Loc, X, \Delta_b, \ell_{in})$ is a test-free CM and $\Delta_{nb} \subseteq Loc \times \{\mathtt{nb(x-)} \mid \mathtt{x} \in X\} \times Loc$ is a finite set of* non-blocking *transitions.*

Again, a configuration is given by a pair $(\ell, v) \in \mathrm{Loc} \times \mathbb{N}^X$. Given two configurations $(\ell, v)$ and $(\ell, v')$ and $\delta \in \Delta_b \cup \Delta_{nb}$, we extend the transition relation $(\ell, v) \overset{\delta}{\rightsquigarrow}_M (\ell, v')$ over the set $\Delta_{nb}$ in the following way: for $\delta = (\ell, \mathtt{nb(x-)}, \ell') \in \Delta_{nb}$, we have $(\ell, v) \overset{\delta}{\rightsquigarrow}_M (\ell', v')$ if and only if $v'(\mathtt{x}) = \max(0, v(\mathtt{x}) - 1)$, and $v'(\mathtt{x}') = v(\mathtt{x}')$ for all $\mathtt{x}' \in X \setminus \{\mathtt{x}\}$.

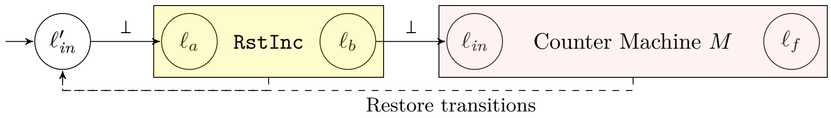


**Figure 2** The NB+R-CM $N$

We say that $M$ is an NB-CM *with restore* (shortly NB+R-CM) when $(\ell, \perp, \ell_{in}) \in \Delta$ for all $\ell \in \mathrm{Loc}$, i.e. from each location, there is a transition leading to the initial location with no effect on the counters values.

For a CM $M$ with set of transitions $\Delta$ (resp. an NB-CM with sets of transitions $\Delta_b$ and $\Delta_{nb}$), we will write $(\ell, v) \rightsquigarrow_M (\ell', v')$ whenever there exists $\delta \in \Delta$ (resp. $\delta \in \Delta_b \cup \Delta_{nb}$) such that $(\ell, v) \overset{\delta}{\rightsquigarrow}_M (\ell', v')$ and use $\rightsquigarrow_M^*$ to represent the reflexive and transitive closure of $\rightsquigarrow_M$. When the context is clear we shall write $\rightsquigarrow$ instead of $\rightsquigarrow_M$. We let $\mathbf{0}_X$ be the valuation such that $\mathbf{0}_X(\mathtt{x}) = 0$ for all $\mathtt{x} \in X$. An execution is a finite sequence of configurations $(\ell_0, v_0) \rightsquigarrow (\ell_1, v_1) \rightsquigarrow \ldots \rightsquigarrow (\ell_k, v_k)$. It is said to be initial if $(\ell_0, v_0) = (\ell_{in}, \mathbf{0}_X)$. A configuration $(\ell, v)$ is called reachable if $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell, v)$.

We shall now define the coverability problem for (non-blocking test-free) counter machines, which asks whether a given location can be reached from the initial configuration. We denote this problem $\textsc{Cover}[\mathcal{M}]$, for $\mathcal{M} \in \{\text{CM}, \text{test-free CM}, \text{NB-CM}, \text{NB+R-CM}\}$. It takes as input a machine $M$ in $\mathcal{M}$ (with initial location $\ell_{in}$ and working over a set $X$ of counters) and a location $\ell_f$ and it checks whether there is a valuation $v \in \mathbb{N}^X$ such that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_f, v)$.

In the rest of this section, we will prove that $\textsc{Cover}$[NB+R-CM] is $\textsc{Expspace}$-complete. To this end, we first establish that $\textsc{Cover}$[NB-CM] is in $\textsc{Expspace}$, by an adaptation of Rackoff's proof which shows that coverability in Vector Addition Systems is in $\textsc{Expspace}$ [18]. This gives also the upper bound for $NB + R - CM$, since any NB+R-CM is a NB-CM.

▶ **Theorem 3.3.** *$\textsc{Cover}[NB - CM]$ and $\textsc{Cover}[NB + R - CM]$ are in $\textsc{Expspace}$.*

To obtain the lower bound, inspired by Lipton's proof showing that coverability in Vector Addition Systems is $\textsc{Expspace}$-hard [8, 16], we rely on 2$\textsc{Exp}$-bounded-test-free CM. We say that a CM $M = (\mathrm{Loc}, X, \Delta, \ell_{in})$ is 2EXP-*bounded* if there exists $n \in O(|\mathrm{Loc}| + |X| + |\Delta|)$ such that any reachable configuration $(\ell, v)$ satisfies $v(\mathtt{x}) \leq 2^{2^n}$ for all $\mathtt{x} \in X$. We use then the following result.

▶ **Theorem 3.4** ([8, 16]). *$\textsc{Cover}$[2$\textsc{Exp}$-bounded-test-free CM] is $\textsc{Expspace}$-hard.*

We now show how to simulate a 2$\textsc{Exp}$-bounded-test free-CM by a NB+R-CM, by carefully handling restore transitions that may occur at any point in the execution. We will ensure that each restore transition is followed by a reset of the counters, so that we can always extract from an execution of the NB+R-CM a correct initial execution of the original test free-CM. The way we enforce resetting of the counters is inspired by the way Lipton simulates 0-tests of a CM in a test-free CM. As in [16, 8], we will describe the final NB+R-CM by means of several submachines. To this end, we define *procedural non-blocking counter machines* that are NB-CM with several identified *output states*: formally, a procedural-NB-CM is a tuple $N = (\mathrm{Loc}, X, \Delta_b, \Delta_{nb}, \ell_{in}, L_{out})$ such that $(\mathrm{Loc}, X, \Delta_b, \Delta_{nb}, \ell_{in})$ is a NB-CM and $L_{out} \subseteq \mathrm{Loc}$.

Now fix a 2EXP-bounded-test-free CM $M = (\mathrm{Loc}, X, \Delta, \ell_{in})$, $\ell_f \in \mathrm{Loc}$ the location to be covered, and $n \in O(|M|)$ such that any reachable configuration $(\ell, v)$ satisfies $v(\mathtt{x}) \leq 2^{2^n}$ for all $\mathtt{x} \in X$. We build a NB+R-CM $N$ as pictured in Figure 2. The goal of the procedural NB-CM `RstInc` is to ensure that all counters in $X$ are reset. Hence, after each restore transition, we are sure that we start over a fresh execution of the test-free CM $M$. We will

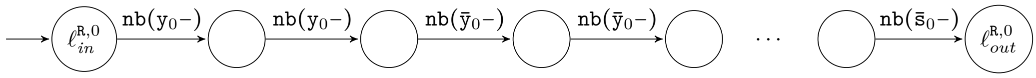


**Figure 3** Description of $\mathtt{Rst}_0$

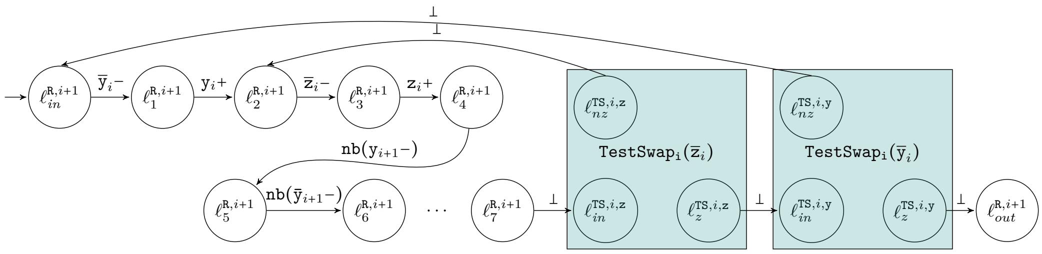


**Figure 4** Description of $\mathtt{Rst}_{\mathtt{i+1}}$

need the mechanism designed by Lipton to test whether a counter is equal to 0. So, we define two families of counters $(Y_i)_{0\le i\le n}$ and $(\overline{Y_i})_{0\le i\le n}$ as follows. Let $Y_i = \{\mathtt{y}_i, \mathtt{z}_i, \mathtt{s}_i\}$ and $\overline{Y}_i = \{\overline{\mathtt{y}}_i, \overline{\mathtt{z}}_i, \overline{\mathtt{s}}_i\}$ for all $0 \le i < n$ and $Y_n = X$ and $\overline{Y}_n = \emptyset$ and $X' = \bigcup_{0\le i\le n} Y_i \cup \overline{Y}_i$. All the machines we will describe from now on will work over the set of counters $X'$.

**Procedural-NB-CM $\mathtt{TestSwap}_i(\mathtt{x})$.** We use a family of procedural-NB-CM defined in [16, 8]: for all $0 \le i < n$, for all $\overline{\mathtt{x}} \in \overline{Y}_i$, $\mathtt{TestSwap}_i(\overline{\mathtt{x}})$ is a procedural-NB-CM with initial location $\ell_{in}^{\mathtt{TS},i,\mathtt{x}}$, and two output locations $\ell_z^{\mathtt{TS},i,\mathtt{x}}$ and $\ell_{nz}^{\mathtt{TS},i,\mathtt{x}}$. It tests if the value of $\overline{\mathtt{x}}$ is equal to 0, using the fact that the sum of the values of $\mathtt{x}$ and $\overline{\mathtt{x}}$ is equal to $2^{2^i}$. If $\overline{\mathtt{x}} = 0$, it swaps the values of $\mathtt{x}$ and $\overline{\mathtt{x}}$, and the execution ends in the output location $\ell_z^{\mathtt{TS},i,\mathtt{x}}$. Otherwise, counters values are left unchanged and the execution ends in $\ell_{nz}^{\mathtt{TS},i,\mathtt{x}}$. In any case, other counters are not modified by the execution. Note that $\mathtt{TestSwap}_i(\mathtt{x})$ makes use of variables in $\bigcup_{1\le j<i} Y_i \cup \overline{Y}_i$.

**Procedural NB-CM $\mathtt{Rst}_i$.** We use these machines to define a family of procedural-NB-CM $(\mathtt{Rst}_i)_{0\le i\le n}$ that reset the counters in $Y_i \cup \overline{Y_i}$, assuming that their values are less or equal than $2^{2^i}$. Let $0 \le i \le n$, we let $\mathtt{Rst}_i = (\mathrm{Loc}^{\mathtt{R},i}, X', \Delta_b^{\mathtt{R},i}, \Delta_{nb}^{\mathtt{R},i}, \ell_{in}^{\mathtt{R},i}, \{\ell_{out}^{\mathtt{R},i}\})$. The machine $\mathtt{Rst}_0$ is pictured Figure 3. For all $0 \le i < n$, the machine $\mathtt{Rst}_{i+1}$ uses counters from $Y_i \cup \overline{Y_i}$ and procedural-NB-CM $\mathtt{Testswap}_i(\overline{\mathtt{z}}_i)$ and $\mathtt{Testswap}_i(\overline{\mathtt{y}}_i)$ to control the number of times variables from $Y_{i+1}$ and $\overline{Y}_{i+1}$ are decremented. It is pictured Figure 4. Observe that since $Y_n = X$, and $\overline{Y_n} = \emptyset$, the machine $\mathtt{Rst}_n$ will be a bit different from the picture : there will only be non-blocking decrements over counters from $Y_n$, that is over counters $X$ from the initial test-free CM $M$. If $\overline{\mathtt{y}}_i$, $\overline{\mathtt{z}}_i$ (and $\overline{\mathtt{s}}_i$) are set to $2^{2^i}$ and $\mathtt{y}_i$, $\mathtt{z}_i$ (and $\mathtt{s}_i$) are set to 0, then each time this procedural-NB-CM takes an outer loop, the variables of $Y_{i+1} \cup \overline{Y}_{i+1}$ are decremented (in a non-blocking fashion) $2^{2^i}$ times. This is ensured by the properties of $\mathtt{TestSwap}_i(\mathtt{x})$. Moreover, the location $\ell_z^{\mathtt{TS},i,\mathtt{y}}$ will be reached only when the counter $\overline{\mathtt{y}}_i$ will be set to 0, and this will happen after $2^{2^i}$ taking of the outer loop, again thanks to the properties of $\mathtt{TestSwap}_i(\mathtt{x})$. So, all in all, variables from $Y_i$ and $\overline{Y}_{i+1}$ will take a non-blocking decrement $2^{2^i}.2^{2^i}$ times, that is $2^{2^{i+1}}$.

For all $\mathtt{x} \in X'$, we say that $\mathtt{x}$ is *initialized* in a valuation $v$ if $\mathtt{x} \in Y_i$ for some $0 \le i \le n$ and $v(\mathtt{x}) = 0$, or $\mathtt{x} \in \overline{Y}_i$ for some $0 \le i \le n$ and $v(\mathtt{x}) = 2^{2^i}$. For $0 \le i \le n$, we say that a valuation $v \in \mathbb{N}^{X'}$ is *$i$-bounded* if for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v(\mathtt{x}) \le 2^{2^i}$.

The construction ensures that when one enters $\mathtt{Rst}_i$ with a valuation $v$ that is $i$-bounded, and in which all variables in $\bigcup_{0\le j<i} Y_j \cup \overline{Y}_j$ are initialized, the location $\ell_{out}^{\mathtt{R},i}$ is reached with a valuation $v'$ such that : $v'(\mathtt{x}) = 0$ for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$ and $v'(\mathtt{x}) = v(\mathtt{x})$ for all $\mathtt{x} \notin Y_i \cup \overline{Y}_i$.

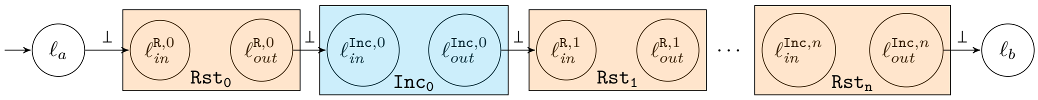


**Figure 5** `RstInc`

Moreover, if $v$ is $j$-bounded for all $0 \le j \le n$, then any valuation reached during the execution remains $j$-bounded for all $0 \le j \le n$.

**Procedural NB-CM $\mathtt{Inc}_i$.** The properties we seek for $\mathtt{Rst}_i$ are ensured whenever the variables in $\bigcup_{0 \le j < i} Y_j \cup \overline{Y}_j$ are initialized. This is taken care of by a family of procedural-NB-CM introduced in [16, 8]. For all $0 \le i < n$, $\mathtt{Inc}_i$ is a procedural-NB-CM with initial location $\ell_{in}^{\mathtt{Inc},i}$, and unique output location $\ell_{out}^{\mathtt{Inc},i}$. They enjoy the following property: for $0 \le i < n$, when one enters $\mathtt{Inc}_i$ with a valuation $v$ in which all the variables in $\bigcup_{0 \le j < i} Y_j \cup \overline{Y}_j$ are initialized and $v(\mathtt{x}) = 0$ for all $\mathtt{x} \in \overline{Y}_i$, then the location $\ell_{out}^{\mathtt{Inc}_i}$ is reached with a valuation $v'$ such that $v'(\mathtt{x}) = 2^{2^i}$ for all $\mathtt{x} \in \overline{Y}_i$, and $v'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$. Moreover, if $v$ is $j$-bounded for all $0 \le j \le n$, then any valuation reached during the execution remains $j$-bounded for all $0 \le j \le n$.

**Procedural NB-CM `RstInc`.** Finally, let `RstInc` be a procedural-NB-CM with initial location $\ell_a$ and output location $\ell_b$, over the set of counters $X'$ and built as an alternation of $\mathtt{Rst}_i$ and $\mathtt{Inc}_i$ for $0 \le i < n$, finished by $\mathtt{Rst}_n$. It is described Figure 5. Thanks to the properties of the machines $\mathtt{Rst}_i$ and $\mathtt{Inc}_i$, in the output location of each $\mathtt{Inc}_i$ machine, the counters in $\overline{Y}_i$ are set to $2^{2^i}$, which allow counters in $Y_{i+1} \cup \overline{Y}_{i+1}$ to be set to 0 in the output location of $\mathtt{Rst}_{i+1}$. Hence, in location $\ell_{out}^{\mathtt{Inc},n}$, counters in $Y_n = X$ are set to 0.

From [16, 8], each procedural machine $\mathtt{TestSwap}_i(\mathtt{x})$ and $\mathtt{Inc}_i$ has size at most $C \times n^2$ for some constant $C$. Hence, observe that $N$ is of size at most $B$ for some $B \in O(|M|^3)$. One can show that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow_M^* (\ell_f, v)$ for some $v \in \mathbb{N}^X$, if and only if $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell_f, v')$ for some $v' \in \mathbb{N}^{X'}$. Using Theorem 3.4, we obtain:

▶ **Theorem 3.5.** *Cover[NB+R-CM] is Expspace-hard.*

## 4 Coverability for Rendez-Vous Protocols

In this section we prove that SCover and CCover problems are both Expspace-complete for rendez-vous protocols. To this end, we present the following reductions: CCover reduces to Cover[NB-CM] and Cover[NB+R-CM] reduces to SCover. This will prove that CCover is in Expspace and SCover is Expspace-hard (from Theorem 3.3 and Theorem 3.5). As SCover is an instance of CCover, the two reductions suffice to prove Expspace-completeness for both problems.

### 4.1 From Rendez-vous Protocols to NB-CM

Let $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$ a rendez-vous protocol and $C_F$ a configuration of $\mathcal{P}$ to be covered. We shall also decompose $C_F$ as a sum of multisets $\wr \mathbf{q}_1 \wr + \wr \mathbf{q}_2 \wr + \dots + \wr \mathbf{q}_s \wr$. Observe that there might be $\mathbf{q}_i = \mathbf{q}_j$ for $i \neq j$. We build the NB-CM $M = (\mathrm{Loc}, X, \Delta_b, \Delta_{nb}, \ell_{in})$ with $X = Q$. A configuration $C$ of $\mathcal{P}$ is meant to be represented in $M$ by $(\ell_{in}, v)$, with $v(q) = C(q)$ for all $q \in Q$. The only meaningful location of $M$ is then $\ell_{in}$. The other ones are here to ensure correct updates of the counters when simulating a transition. We let $\mathrm{Loc} = \{\ell_{in}\} \cup \{\ell^1_{(t,t')}, \ell^2_{(t,t')}, \ell^3_{(t,t')} \mid t = (q, !a, q'), t' = (p, ?a, p') \in T\} \cup \{\ell_t, \ell^a_{t,p_1}, \cdots, \ell^a_{t,p_k} \mid t = (q, !a, q') \in T, R(a) = \{p_1, \dots, p_k\}\} \cup \{\ell_q \mid t = (q, \tau, q') \in T\} \cup \{\ell_1 \dots \ell_s\}$, with final location $\ell_f = \ell_s$, where

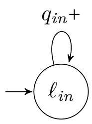


**Figure 6** Incrementing $q_{in}$

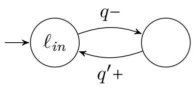


**Figure 7** Transitions for $(q,\tau,q') \in T$

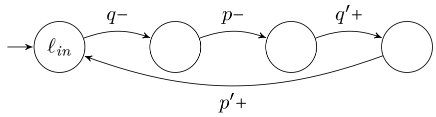


**Figure 8** Transitions for a rendez-vous $(q,!a,q')$, $(p,?a,p') \in T$

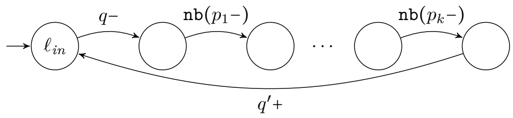


**Figure 9** Transitions for a non-blocking sending $(q,!a,q') \in T$ and $R(a) = \{p_1 \dots p_k\}$

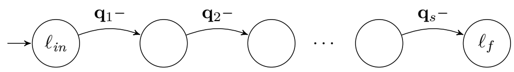


**Figure 10** Verification for the coverability of $C_F = \wr \mathbf{q}_1 \wr + \wr \mathbf{q}_2 \wr + \cdots + \wr \mathbf{q}_s \wr$

$R(m)$ for a message $m \in \Sigma$ has been defined in Section 2. The sets $\Delta_b$ and $\Delta_{nb}$ are shown Figures 6–10. Transitions pictured Figures 6–8 and 10 show how to simulate a rendez-vous protocol with the classical rendez-vous mechanism. The non-blocking rendez-vous are handled by the transitions pictured Figure 9. If the NB-CM $M$ faithfully simulates $\mathcal{P}$, then this loop of non-blocking decrements is taken when the values of the counters in $R(a)$ are equal to 0, and the configuration reached still corresponds to a configuration in $\mathcal{P}$. However, it could be that this loop is taken in $M$ while some counters in $R(a)$ are strictly positive. In this case, a blocking rendez-vous has to be taken in $\mathcal{P}$, e.g. $(q,!a,q')$ and $(p,?a,p')$ if the counter $p$ in $M$ is strictly positive. Therefore, the value of the reached configuration $(\ell_{in}, v)$ and the corresponding configuration $C$ in $\mathcal{P}$ will be different, nonetheless $C \geq v$.Then, if it is possible to reach a configuration $(\ell_{in}, v)$ in $M$ whose counters are high enough to cover $\ell_F$, then the corresponding initial execution in $\mathcal{P}$ will reach a configuration $C \geq v$ which covers $C_F$.

▶ **Theorem 4.1.** *CCover over rendez-vous protocols is in Expspace.*

## 4.2 From NB+R-CM to Rendez-Vous Protocols

The reduction from Cover[NB+R-CM] to SCover in rendez-vous protocols mainly relies on the mechanism that can ensure that at most one process evolves in some given set of states, as explained in Example 2.5. This will allow to somehow select a “leader” among the processes that will simulate the behaviour of the NB+R-CM whereas other processes will simulate the values of the counters. Let $M = (\mathrm{Loc}, X, \Delta_b, \Delta_{nb}, \ell_{in})$ a NB+R-CM and $\ell_f \in \mathrm{Loc}$ a final target location. We build the rendez-vous protocol $\mathcal{P}$ pictured in Figure 11, where $\mathcal{P}(M)$ is the part that will simulate the NB+R-CM $M$. The locations $\{1_{\mathtt{x}} \mid \mathtt{x} \in X\}$ will allow to encode the values of the different counters during the execution: for a configuration $C$, $C(1_{\mathtt{x}})$ will represent the value of the counter $\mathtt{x}$. We give then $\mathcal{P}(M) = (Q_M, \Sigma_M, \ell_{in}, \ell_f, T_M)$ with $Q_M = \mathrm{Loc} \cup \{\ell_\delta \mid \delta \in \Delta_b\}$, $\Sigma_M = \{\mathrm{inc}_{\mathtt{x}}, \overline{\mathrm{inc}}_{\mathtt{x}}, \mathrm{dec}_{\mathtt{x}}, \overline{\mathrm{dec}}_{\mathtt{x}}, \mathrm{nbdec}_{\mathtt{x}} \mid \mathtt{x} \in X\}$, and $T_M = \{(\ell_i, !\mathrm{inc}_{\mathtt{x}}, \ell_\delta), (\ell_\delta, ?\overline{\mathrm{inc}}_{\mathtt{x}}, \ell_j) \mid \delta = (\ell_i, \mathtt{x+}, \ell_j) \in \Delta_b\} \cup \{(\ell_i, !\mathrm{dec}_{\mathtt{x}}, \ell_\delta), (\ell_\delta, ?\overline{\mathrm{dec}}_{\mathtt{x}}, \ell_j) \mid \delta = (\ell_i, \mathtt{x-}, \ell_j) \in \Delta_b\} \cup \{(\ell_i, !\mathrm{nbdec}_{\mathtt{x}}, \ell_j) \mid (\ell_i, \mathtt{nb(x-)}, \ell_j) \in \Delta_{nb}\} \cup \{(\ell_i, \tau, \ell_j) \mid (\ell_i, \bot, \ell_j) \in \Delta_b\}$. Here, the reception of a message $\overline{\mathrm{inc}}_{\mathtt{x}}$ (respectively $\overline{\mathrm{dec}}_{\mathtt{x}}$) works as an acknowledgement, ensuring that a process has indeed received the message $\mathrm{inc}_{\mathtt{x}}$ (respectively $\mathrm{dec}_{\mathtt{x}}$), and that the corresponding counter has been incremented (resp. decremented). For non-blocking decrement, obviously no acknowledgement is required. The protocol $\mathcal{P} = (Q, \Sigma, q_{in}, \ell_f, T)$ is then defined with $Q = Q_M \cup \{1_{\mathtt{x}}, q_{\mathtt{x}}, q'_{\mathtt{x}} \mid \mathtt{x} \in X\} \cup \{q_{in}, q, q_\bot\}$, $\Sigma = \Sigma_M \cup \{L, R\}$ and $T$ is the set of transitions $T_M$ along with the transitions pictured in Figure 11. Note that there is a transition $(\ell, ?L, q_\bot)$ for all $\ell \in Q_M$.

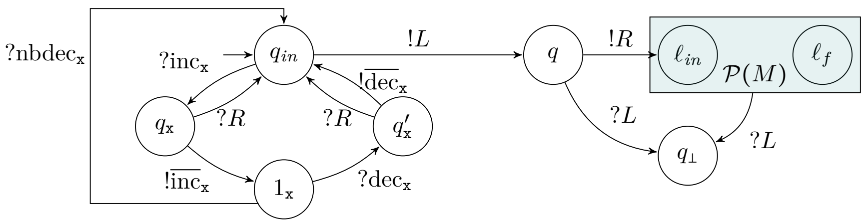


**Figure 11** The rendez-vous protocol $\mathcal{P}$ built from the NB+R-CM $M$. Note that there is one gadget with states $\{q_x, q'_x, 1_x\}$ for each counter $x \in X$.

With two non-blocking transitions on $L$ and $R$ at the beginning, the protocol $\mathcal{P}$ can faithfully simulate the NB+R-CM $M$ without further ado. Conversely, an initial execution of $\mathcal{P}$ can send multiple processes into the $\mathcal{P}(M)$ zone, which can mess up the simulation. However, the construction of the protocol ensures that there can only be one process in the set of states $\{q_x, q'_x \mid x \in X\}$. Then, each new process entering $\mathcal{P}(M)$ will send the message $L$, which will send the process already in $\{q\} \cup Q_M$ in the deadlock state $q_\perp$, and send the message $R$, which will be received by any process in $\{q_x, q'_x \mid x \in X\}$. Therefore, sending a new process in the $\mathcal{P}(M)$ zone simply mimicks a restore transition of $M$. So every initial execution of $\mathcal{P}$ corresponds to an initial execution of $M$.

▶ **Theorem 4.2.** *SCover and CCover over rendez-vous protocols are Expspace complete.*

## 5 Coverability for Wait-Only Protocols

In this section, we study a restriction on rendez-vous protocols in which we assume that a process waiting to answer a rendez-vous cannot perform another action by itself. This allows for a polynomial time algorithm for solving CCover.

### 5.1 Wait–Only Protocols

We say that a protocol $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$ is *wait-only* if the set of states $Q$ can be partitioned into $Q_A$ - the *active states* - and $Q_W$ - the *waiting* states - with $q_{in} \in Q_A$ and:

- for all $q \in Q_A$, for all $(q', ?m, q'') \in T$, we have $q' \neq q$;
- for all $q \in Q_W$, there exists $q' \in Q$ and $m \in \Sigma$ such that $(q, ?m, q') \in T$ and there does not exist $q'' \in Q$ such that $(q, \tau, q'') \in T$ or $(q, !m', q'') \in T$ for some $m' \in \Sigma$.

Hence, with such protocols, when a process is in a waiting state from $Q_W$, he is not able to request rendez-vous nor to perform an internal action. Examples of wait-only protocols are given by Figures 12 and 13.

In the sequel, we will often refer to the paths of the underlying graph of the protocol. Formally, a *path* in a protocol $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$ is either a control state $q \in Q$ or a finite sequence of transitions in $T$ of the form $(q_0, a_0, q_1)(q_1, a_1, q_2) \dots (q_k, a_k, q_{k+1})$, the first case representing a path from $q$ to $q$ and the second one from $q_0$ to $q_{k+1}$.

### 5.2 Abstract Sets of Configurations

To solve the coverability problem for wait-only protocols in polynomial time, we rely on a sound and complete abstraction of the set of reachable configurations. In the sequel, we consider a wait-only protocol $\mathcal{P} = (Q, \Sigma, q_{in}, q_f, T)$ whose set of states is partitioned into a set of active states $Q_A$ and a set of waiting states $Q_W$. An *abstract set of configurations* $\gamma$ is a pair $(S, \mathit{Toks})$ such that:

- $S \subseteq Q$ is a subset of states, and,
- *Toks* $\subseteq Q_W \times \Sigma$ is a subset of pairs composed of a waiting state and a message, and,
- $q \notin S$ for all $(q, m) \in$ *Toks*.

We abstract then the set of reachable configurations as a set of states of the underlying protocol. However, as we have seen, some states, like states in $Q_A$, can host an unbounded number of processes together (this will be the states in $S$), when some states can only host a bounded number (in fact, 1) of processes together (this will be the states stored in *Toks*). This happens when a waiting state $q$ answers a rendez-vous $m$, that has necessarily been requested for a process to be in $q$. Hence, in *Toks*, along with a state $q$, we remember the last message $m$ having been sent in the path leading from $q_{in}$ to $q$, which is necessarily in $Q_W$. Observe that, since several paths can lead to $q$, there can be $(q, m_1), (q, m_2) \in$ *Toks* with $m_1 \neq m_2$. We denote by $\Gamma$ the set of abstract sets of configuration.

Let $\gamma = (S, \mathit{Toks})$ be an abstract set of configurations. Before we go into the configurations represented by $\gamma$, we need some preliminary definitions. We note $\mathsf{st}(\mathit{Toks})$ the set $\{q \in Q_W \mid$ there exists $m \in \Sigma$ such that $(q, m) \in \mathit{Toks}\}$ of control states appearing in *Toks*. Given a state $q \in Q$, we let $\mathrm{Rec}(q)$ be the set $\{m \in \Sigma \mid$ there exists $q' \in Q$ such that $(q, ?m, q') \in T\}$ of messages that can be received in state $q$ (if $q$ is not a waiting state, this set is empty). Given two different waiting states $q_1$ and $q_2$ in $\mathsf{st}(\mathit{Toks})$, we say $q_1$ and $q_2$ are *conflict-free* in $\gamma$ if there exist $m_1, m_2 \in \Sigma$ such that $m_1 \neq m_2$, $(q_1, m_1), (q_2, m_2) \in \mathit{Toks}$ and $m_1 \notin \mathrm{Rec}(q_2)$ and $m_2 \notin \mathrm{Rec}(q_1)$. We now say that a configuration $C \in \mathcal{C}(\mathcal{P})$ *respects* $\gamma$ if and only if for all $q \in Q$ such that $C(q) > 0$ one of the following two conditions holds:

1. $q \in S$, or,
2. $q \in \mathsf{st}(\mathit{Toks})$ and $C(q) = 1$ and for all $q' \in \mathsf{st}(\mathit{Toks}) \setminus \{q\}$ such that $C(q') = 1$, we have that $q$ and $q'$ are conflict-free.

Let $[\![\gamma]\!]$ be the set of configurations respecting $\gamma$. Note that in $[\![\gamma]\!]$, for $q$ in $S$ there is no restriction on the number of processes that can be put in $q$ and if $q$ in $\mathsf{st}(\mathit{Toks})$, it can host at most one process. Two states from $\mathsf{st}(\mathit{Toks})$ can both host a process if they are conflict-free.

Finally, we will only consider abstract sets of configurations that are *consistent*. This property aims to ensure that concrete configurations that respect it are indeed reachable from states of $S$. Formally, we say that an abstract set of configurations $\gamma = (S, \mathit{Toks})$ is *consistent* if $(i)$ for all $(q, m) \in \mathit{Toks}$, there exists a path $(q_0, a_0, q_1)(q_1, a_1, q_2) \ldots (q_k, a_k, q)$ in $\mathcal{P}$ such that $q_0 \in S$ and $a_0 = !m$ and for all $1 \leq i \leq k$, we have that $a_i = ?m_i$ and that there exist $(q'_i, !m_i, q''_i) \in T$ with $q'_i \in S$, and $(ii)$ for two tokens $(q, m), (q', m') \in \mathit{Toks}$ either $m \in \mathrm{Rec}(q')$ and $m' \in \mathrm{Rec}(q)$, or, $m \notin \mathrm{Rec}(q')$ and $m' \notin \mathrm{Rec}(q)$. Condition $(i)$ ensures that processes in $S$ can indeed lead to a process in the states from $\mathsf{st}(\mathit{Toks})$. Condition $(ii)$ ensures that if in a configuration $C$, a set of states in $\mathsf{st}(\mathit{Toks})$ are pairwise conflict-free, then they can all host a process together.

▶ **Lemma 5.1.** *Given $\gamma \in \Gamma$ and a configuration $C$, there exists $C' \in [\![\gamma]\!]$ such that $C' \geq C$ if and only if $C \in [\![\gamma]\!]$. Checking that $C \in [\![\gamma]\!]$ can be done in polynomial time.*

## 5.3 Computing Abstract Sets of Configurations

Our polynomial time algorithm is based on the computation of a polynomial length sequence of consistent abstract sets of configurations leading to a final abstract set characterising in a sound and complete manner (with respect to the coverability problem), an abstraction for the set of reachable configurations. This will be achieved by a function $F : \Gamma \to \Gamma$, that inductively computes this final abstract set starting from $\gamma_0 = (\{q_{in}\}, \varnothing)$.

| **Construction of intermediate states $S''$ and $\mathit{Toks}''$** |
|---|
| **1.** $S \subseteq S''$ and $\mathit{Toks} \subseteq \mathit{Toks}''$ |
| **2.** for all $(p,\tau,p') \in T$ with $p \in S$, we have $p' \in S''$ |
| **3.** for all $(p,!a,p') \in T$ with $p \in S$, we have: |
| **a.** $p' \in S''$ if $a \notin \mathrm{Rec}(p')$ or if there exists $(q,?a,q') \in T$ with $q \in S$; |
| **b.** $(p',a) \in \mathit{Toks}''$ otherwise (i.e. when $a \in \mathrm{Rec}(p')$ and for all $(q,?a,q') \in T$, $q \notin S$); |
| **4.** for all $(q,?a,q') \in T$ with $q \in S$ or $(q,a) \in \mathit{Toks}$, we have $q' \in S''$ if there exists $(p,!a,p') \in T$ with $p \in S$; |
| **5.** for all $(q,?a,q') \in T$ with $(q,m) \in \mathit{Toks}$ with $m \neq a$, we have: |
| **a.** $q' \in S''$ if $m \notin \mathrm{Rec}(q')$ and there exists $(p,!a,p') \in T$ with $p \in S$; |
| **b.** $(q',m) \in \mathit{Toks}''$ if $m \in \mathrm{Rec}(q')$ and there exists $(p,!a,p') \in T$ with $p \in S$; |
| **Construction of state $S'$, the smallest set including $S''$ and such that:** |
| **6.** for all $(q_1,m_1),(q_2,m_2) \in \mathit{Toks}''$ such that $m_1 \neq m_2$ and $m_2 \notin \mathrm{Rec}(q_1)$ and $m_1 \in \mathrm{Rec}(q_2)$, we have $q_1 \in S'$; |
| **7.** for all $(q_1,m_1),(q_2,m_2) \in \mathit{Toks}''$ s.t $m_1 \neq m_2$ and $(q_2,?m_1,q_3) \in T$, we have $q_1 \in S'$; |
| **8.** for all $(q_1,m_1),(q_2,m_2),(q_3,m_3) \in \mathit{Toks}''$ such that $m_1 \neq m_2$ and $m_1 \neq m_3$ and $m_2 \neq m_3$ and $m_1 \notin \mathrm{Rec}(q_2)$, $m_1 \in \mathrm{Rec}(q_3)$ and $m_2 \notin \mathrm{Rec}(q_1)$, $m_2 \in \mathrm{Rec}(q_3)$, and $m_3 \in \mathrm{Rec}(q_2)$ and $m_3 \in \mathrm{Rec}(q_1)$, we have $q_1 \in S'$. |
| **Construction of state $\mathit{Toks}'$** |
| $\mathit{Toks}' = \{(q,m) \in \mathit{Toks}'' \mid q \notin S'\}$. |

**Table 1** Definition of $F(\gamma) = (S', \mathit{Toks}')$ for $\gamma = (S, \mathit{Toks})$.

Formal definition of the function $F$ is given by Table 1, and relies on intermediate sets $S'' \subseteq Q$ and $\mathit{Toks}'' \subseteq Q \times \Sigma$, which are the smallest sets satisfying the conditions described. Observe that it might be that a state is both added to $S''$ and $\mathit{Toks}''$; in that case, it will be removed from $\mathit{Toks}'$ by application of the last rule of $F$. Hence, a state belongs either to $S'$ or to $\mathsf{st}(\mathit{Toks}')$.

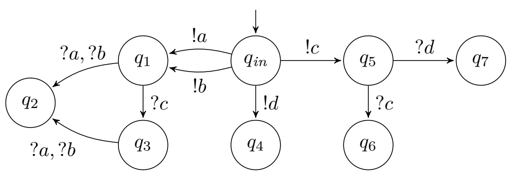


**Figure 12** Wait-only protocol $\mathcal{P}_1$.

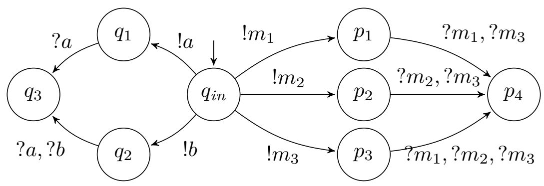


**Figure 13** Wait-only protocol $\mathcal{P}_2$.

▶ **Example 5.2.** Consider the wait-only protocol $\mathcal{P}_1$ depicted on Figure 12. We have $F((\{q_{in}\},\emptyset)) = (\{q_{in},q_4\},\{(q_1,a),(q_1,b),(q_5,c)\})$. In $\mathcal{P}_1$, it is indeed possible to reach a configuration with as many processes as one wishes in the state $q_4$ by repeating the transition $(q_{in},!d,q_4)$ (rule 3a). On the other hand, it is possible to put *at most* one process in the waiting state $q_1$ (rule 3b), because any other attempt from a process in $q_{in}$ will yield a reception of the message $a$ (resp. $b$) by the process already in $q_1$. Similarly, we can put at most one process in $q_5$. Note that in $F((\{q_{in}\},\emptyset))$, the states $q_1$ and $q_5$ are conflict-free and it is hence possible to have simultaneously one process in both of them.

If we apply the function $F$ one more time, we first get $S'' = \{q_{in},q_2,q_4,q_6,q_7\}$ and $\mathit{Toks}'' = \{(q_1,a),(q_1,b),(q_3,a),(q_3,b),(q_5,c)\}$. We can put at most one process in $q_3$: to add one, a process will take the transition $(q_1,?c,q_3)$. Since $(q_1,a)$, $(q_1,b) \in \mathit{Toks}$, there can be at most one process in state $q_1$, and this process arrived by a path in which the last request of rendez-vous was $!a$ or $!b$. Since $\{a,b\} \subseteq \mathrm{Rec}(q_3)$, by rule 5b, $(q_3,a),(q_3,b)$ are added. On the other hand we can put as many processes as we want in the state $q_7$ (rule 5a): from a configuration with one process on state $q_5$, successive non-blocking request on letter $c$, and rendez-vous on letter $d$ will allow to increase the number of processes in state $q_7$. Now, observe that the tokens $(q_5,c)$, $(q_1,a)$, $(q_3,a)$ allow for application of rule 7, since $(q_1,?c,q_3) \in T$, and yields $q_5$ in $S'$. Once two processes have been put on states $q_1$ and $q_5$ respectively (remember that $q_1$ and $q_5$ are conflict-free in $F(\gamma)$), iterating rendez-vous on letter $c$ (with transition $(q_1, ?c, q_3)$) and rendez-vous on letter $a$ put as many processes as one wants on state $q_5$. Finally, $F(F(\{q_{in}\}, \varnothing)) = (\{q_{in}, q_2, q_4, q_5, q_6, q_7\}, \{(q_1, a), (q_1, b), (q_3, a), (q_3, b)\})$. Since $q_1$ and $q_3$ are not conflict-free, they won't be reachable together in a configuration.

We consider now the wait-only protocol $\mathcal{P}_2$ depicted on Figure 13. In that case, to compute $F((\{q_{in}\}, \varnothing))$ we will first have $S'' = \{q_{in}\}$ and $\mathit{Toks}'' = \{(q_1, a), (q_2, b), (p_1, m_1), (p_2, m_2), (p_3, m_3)\}$ (using rule 3b), to finally get $F((\{q_{in}\}, \varnothing)) = (\{q_{in}, q_1, p_1\}, \{(q_2, b), (p_2, m_2), (p_3, m_3)\}))$. Applying rule 6 to tokens $(q_1, a)$ and $(q_2, b)$ from $\mathit{Toks}''$, we obtain that $q_1 \in S'$: whenever one manages to obtain one process in state $q_2$, this process can answer the requests on message $a$ instead of processes in state $q_1$, allowing one to obtain as many processes as desired in state $q_1$. Now since $(p_1, m_1)$, $(p_2, m_2)$ and $(p_3, m_3)$ are in $\mathit{Toks}''$ and respect the conditions of rule 8, $p_1$ is added to the set $S'$ of unbounded states. This case is a generalisation of the previous one, with 3 processes. Once one process has been put on state $p_2$ from $q_{in}$, iterating the following actions: rendez-vous over $m_3$, rendez-vous over $m_1$, non-blocking request of $m_2$, will ensure as many processes as one wants on state $p_1$. Finally applying successively $F$, we get in this case the abstract set $(\{q_{in}, q_1, q_3, p_1, p_2, p_3, p_4\}, \{(q_2, b)\})$.

We show that $F$ satisfies the following properties.

▶ **Lemma 5.3.** **1.** *$F(\gamma)$ is consistent and can be computed in polynomial time for all consistent $\gamma \in \Gamma$.*
**2.** *If $(S', \mathit{Toks}') = F(S, \mathit{Toks})$ then $S \subsetneq S'$ or $\mathit{Toks} \subseteq \mathit{Toks}'$.*
**3.** *For all consistent $\gamma \in \Gamma$, if $C \in [\![\gamma]\!]$ and $C \to C'$ then $C' \in [\![F(\gamma)]\!]$.*
**4.** *For all consistent $\gamma \in \Gamma$, if $C' \in [\![F(\gamma)]\!]$, then there exists $C'' \in \mathcal{C}$ and $C \in [\![\gamma]\!]$ such that $C'' \geq C'$ and $C \to^* C''$.*

## 5.4 Polynomial Time Algorithm

We now present our polynomial time algorithm to solve CCover for wait-only protocols. We define the sequence $(\gamma_n)_{n \in \mathbb{N}}$ as follows : $\gamma_0 = (\{q_{in}\}, \varnothing)$ and $\gamma_{i+1} = F(\gamma_i)$ for all $i \in \mathbb{N}$. First note that $\gamma_0$ is consistent and that $[\![\gamma_0]\!] = \mathcal{I}$ is the set of initial configurations. Using Lemma 5.3, we deduce that $\gamma_i$ is consistent for all $i \in \mathbb{N}$. Furthermore, each time we apply $F$ to an abstract set of configurations $(S, \mathit{Toks})$ either $S$ or $\mathit{Toks}$ increases. Hence for all $n \geq |Q|^2 * |\Sigma|$, we have $\gamma_{n+1} = F(\gamma_n) = \gamma_n$. Let $\gamma_f = \gamma_{|Q|^2 * |\Sigma|}$. Using Lemma 5.3, we get:

▶ **Lemma 5.4.** *Given $C \in \mathcal{C}$, there exists $C_0 \in \mathcal{I}$ and $C' \geq C$ such that $C_0 \to^* C'$ if and only if there exists $C'' \in [\![\gamma_f]\!]$ such that $C'' \geq C$.*

We need to iterate $|Q|^2 * |\Sigma|$ times the function $F$ to compute $\gamma_f$ and each computation of $F$ can be done in polynomial time. Furthermore checking whether there exists $C'' \in [\![\gamma_f]\!]$ such that $C'' \geq C$ for a configuration $C \in \mathcal{C}$ can be done in polynomial time by Lemma 5.1, hence using the previous lemma we obtain the desired result.

▶ **Theorem 5.5.** *CCover and SCover restricted to wait-only protocols are in Ptime.*

# 6 Undecidability of Synchro

It is known that Cover[CM] is undecidable in its full generality [17]. This result holds for a very restricted class of counter machines, namely Minsky machines (Minsky-CM for short), which are CM over 2 counters, $\mathtt{x}_1$ and $\mathtt{x}_2$. Actually, it is already undecidable whether there

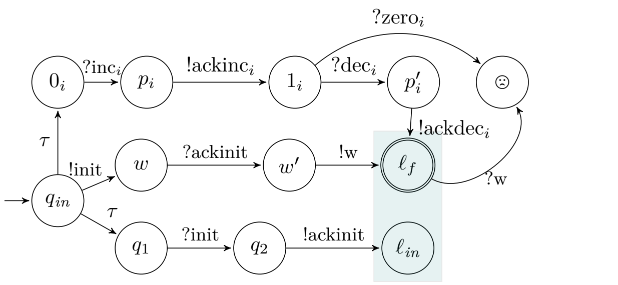


**Figure 14** The protocol $\mathcal{P}$ - The coloured zone contains transitions pictured in Figures 15–17

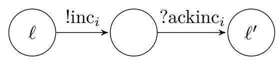


**Figure 15** Translation of $(\ell, x_i+, \ell')$

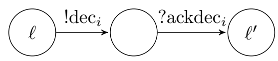


**Figure 16** Translation of $(\ell, x_i-, \ell')$.

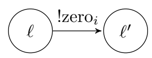


**Figure 17** Translation of $(\ell, x_i = 0, \ell')$.

is an execution $(\ell_{in}, \mathbf{0}_{\{\mathtt{x}_1,\mathtt{x}_2\}}) \rightsquigarrow^* (\ell_f, \mathbf{0}_{\{\mathtt{x}_1,\mathtt{x}_2\}})$. Reduction from this last problem gives the following result.

▶ **Theorem 6.1.** *Synchro is undecidable, even for wait-only protocols.*

Fix $M = (\mathrm{Loc}, \ell_0, C = \{\mathtt{x}_1, \mathtt{x}_2\}, \Delta)$ with $\ell_f \in \mathrm{Loc}$ the final state. Wlog, we assume that there is no outgoing transition from state $\ell_f$ in the machine. The protocol $\mathcal{P}$ is described in Figures 14–16. The states $\{0_i, p_i, 1_i, p'_i \mid i = 1, 2\}$ will be visited by processes simulating values of counters, while the states in Loc will be visited by a process simulating the different locations in the Minsky-CM. If at the end of the computation, the counters are equal to 0, it means that each counter has been incremented and decremented the same number of times, so that all processes simulating the counters end up in the state $\ell_f$. The first challenge is to appropriately check when a counter equals 0. This is achieved thanks to the non-blocking semantics: the process sends a message !zero$_i$ to check if the counter $i$ equals 0. If it is does not, the message will be received by a process that will end up in the deadlock state ☹. The second challenge is to ensure that only one process simulates the Minsky-CM in the states in Loc. This is ensured by the states $\{w, w'\}$. Each time a process arrives in the $\ell_{in}$ state, another must arrive in the $w'$ state, as a witness that the simulation has begun. This witness must reach $\ell_f$ for the computation to be an instance of Synchro, but it should be the first to do so, otherwise a process already in $\ell_f$ will receive the message "w" and reach the deadlock state ☹. Thus, if two processes simulate the Minsky-CM, there will be two witnesses, and they won't be able to reach $\ell_f$ together.

## 7 Conclusion

We have introduced the model of parameterised networks communicating by non-blocking rendez-vous, and showed that safety analysis of such networks becomes much harder than in the framework of classical rendez-vous. Indeed, CCover and SCover become Expspace-complete and Synchro undecidable in our framework, while these problems are solvable in polynomial time in the framework of [13]. We have introduced a natural restriction of protocols, in which control states are partitioned between *active* states (that allow requesting of rendez-vous) and *waiting* states (that can only answer to rendez-vous) and showed that CCover can then be solved in polynomial time. Future work includes finding further restrictions that would yield decidability of Synchro. A candidate would be protocols in which waiting states can only receive *one* message. Observe that in that case, the reduction of Section 6 can be adapted to simulate a test-free CM, hence Synchro for this subclass of protocols is as hard as reachability in Vector Addition Systems with States, i.e. non-primitive recursive [15]. Decidability remains open though.

## References


1. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *LICS'96*, pages 313–321. IEEE Computer Society, 1996.
2. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1-2):109–127, 2000.
3. K. R. Apt and D. C. Kozen. Limits for automatic verification of finite-state concurrent systems. *Inf. Process. Lett.*, 22(6):307–309, 1986.
4. A. R. Balasubramanian, J. Esparza, and M. A. Raskin. Finding cut-offs in leaderless rendez-vous protocols is easy. In *FOSSACS'21*, volume 12650 of *LNCS*, pages 42–61. Springer, 2021.
5. G. Delzanno, J. F. Raskin, and L. Van Begin. Towards the automated verification of multithreaded java programs. In *TACAS'02*, volume 2280 of *LNCS*, pages 173–187. Springer, 2002.
6. G. Delzanno, A. Sangnier, R. Traverso, and G. Zavattaro. On the complexity of parameterized reachability in reconfigurable broadcast networks. In *FSTTCS'12*, volume 18 of *LIPIcs*, pages 289–300. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.
7. A. Durand-Gasselin, J. Esparza, P. Ganty, and R. Majumdar. Model checking parameterized asynchronous shared-memory systems. *Formal Methods in System Design*, 50(2-3):140–167, 2017.
8. J. Esparza. Decidability and complexity of petri net problems—an introduction. In *Advanced Course on Petri Nets*, pages 374–428. Springer, 1998.
9. J. Esparza. Keeping a crowd safe: On the complexity of parameterized verification (invited talk). In Ernst W. Mayr and Natacha Portier, editors, *Proceedings of 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *LIPIcs*, pages 1–10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014.
10. J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *LICS'99*, pages 352–359. IEEE Comp. Soc. Press, July 1999.
11. J. Esparza, P. Ganty, and R. Majumdar. Parameterized verification of asynchronous shared-memory systems. In *CAV'13*, volume 8044 of *LNCS*, pages 124–140. Springer-Verlag, 2013.
12. A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.
13. S. M. German and A. P. Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, 1992.
14. F. Horn and A. Sangnier. Deciding the existence of cut-off in parameterized rendez-vous networks. In *CONCUR'20*, volume 171 of *LIPIcs*, pages 46:1–46:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
15. Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *FOCS'21*, pages 1241–1252. IEEE, 2021.
16. R.J. Lipton. *The reachability problem requires exponential space*. Research report (Yale University. Department of Computer Science). Department of Computer Science, Yale University, 1976.
17. Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., 1967.
18. C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.
19. J. F. Raskin and L. Van Begin. Petri nets with non-blocking arcs are difficult to analyze. In *INFINITY'03*, volume 98 of *Electronic Notes in Theoretical Computer Science*, pages 35–55. Elsevier, 2003.

# A Proofs of Section 3

We present here the omitted proofs of Section 3.

## A.1 Proof of Theorem 3.3

We will in fact prove the EXPSPACE upper bound for a more general model: Non-Blocking Vector Addition Systems (NB-VAS). A NB-VAS is composed of a set of transitions over vectors of dimension $d$, sometimes called counters, and an initial vector of $d$ non-negative integers, like in VAS. However, in a NB-VAS, a transition is a couple of vectors: one is a vector of $d$ integers and is called the *blocking* part of the transition and the other one is a vector of $d$ *non-negative* integers and is called the *non-blocking* part of the transition.

▶ **Definition A.1.** *Let $d \in \mathbb{N}$. A Non-blocking Vector Addition System (NB-VAS) of dimension $d$ is a tuple $(T, v_0)$ such that $T \subseteq \mathbb{Z}^d \times \mathbb{N}^d$ and $v_0 \in \mathbb{N}^d$.*

Formally, for two vectors $v, v' \in \mathbb{N}^d$, and a transition $t = (t_b, t_{nb}) \in T$, we write $v \overset{t}{\rightsquigarrow} v'$ if there exists $v'' \in \mathbb{N}^d$ such that $v'' = v + t_b$ and, for all $i \in [1, d]$, $v'(i) = \max(0, v''(i) - t_{nb}(i))$. We write $\rightsquigarrow$ for $\bigcup_{t \in T} \overset{t}{\rightsquigarrow}$. We define an execution as a sequence of vectors $v_1 v_2 \dots v_k$ such that for all $1 \leq i < k$, $v_i \rightsquigarrow v_{i+1}$.

Intuitively, the blocking part $t_b$ of the transition has a strict semantics: to be taken, it needs to be applied to a vector large enough so no value goes below 0. The non-blocking part $t_{nb}$ can be taken even if it decreases one component below 0: the corresponding component will simply be set to 0.

We can now define what is the SCOVER problem on NB-VAS.

▶ **Definition A.2.** *SCOVER problem for a NB-VAS $V = (T, v_0)$ of dimension $d \in \mathbb{N}$ and a target vector $v_f$, asks if there exists $v \in \mathbb{N}^d$, such that $v \geq v_f$ and $v_0 \rightsquigarrow^* v$.*

Adapting the proof of [18] to the model of NB-VAS yields the following result.

▶ **Lemma A.3.** *The SCOVER problem for NB-VAS is in EXPSPACE.*

**Proof.** Fix a NB-VAS $(T, v_0)$ of dimension $d$, we will extend the semantics of NB-VAS to a slighter *relaxed* semantics: let $v, v' \in \mathbb{N}^d$ and $t = (t_b, t_{nb}) \in T$, we will write $v \overset{t}{\rightharpoonup} v'$ when for all $1 \leq j \leq d$, $v'(j) = \max(0, (v + t_b - t_{nb})(j))$.

Note that $v \overset{t}{\rightsquigarrow} v'$ implies that $v \overset{t}{\rightharpoonup} v'$ but the converse is false: consider an NB-VAS of dimension $d = 2$, with $t = (t_b, t_{nb}) \in T$ such that $t_b = (-3, 0)$ and $t_{nb} = (0, 1)$, and let $v = (1, 2)$ and $v' = (0, 1)$. One can easily see that there does not exist $v'' \in \mathbb{N}^2$ such that $v'' = v + t_b$, as $1 - 3 < 0$. So, $t$ cannot be taken from $v$ and it is not the case that $v \overset{t}{\rightsquigarrow} v'$, however, $v \overset{t}{\rightharpoonup} v'$.

We use $\rightharpoonup$ for $\bigcup_{t \in T} \overset{t}{\rightharpoonup}$.

Let $J \subseteq [1, d]$, a path $v_0 \rightharpoonup v_1 \rightharpoonup \dots \rightharpoonup v_m$ is said to be $J$-*correct* if for all $v_i$ such that $i < m$, there exists $t = (t_b, t_{nb}) \in T$, such that $v_i \rightharpoonup^t v_{i+1}$ and for all $j \in J$, $(v_i + t_b)(j) \geq 0$. We say that the path is correct if the path is $[1, d]$-correct.

It follows from the definitions that for all $v, v' \in \mathbb{N}^d$, $v \rightsquigarrow^* v'$ if and only if there exists a correct path between $v$ and $v'$.

Fix a target vector $v_f \in \mathbb{N}^d$, and define $N = |v_f| + \max_{(t_b, t_{nb}) \in T}(|t_b| + |t_{nb}|)$, where $|\cdot|$ is the norm 1 of vectors in $\mathbb{Z}^d$. Let $\rho = v_0 \rightharpoonup v_1 \rightharpoonup \dots \rightharpoonup v_m$ and $J \subseteq [1, d]$. We say the path $\rho$ is $J$-*covering* if it is $J$-correct and for all $j \in J$, $v_m(j) \geq v_f(j)$. Let $r \in \mathbb{N}$, we say that $\rho$ is $(J, r)$-bounded if for all $v_i$, for all $j \in J$, $v_i(j) < r$. Let $v \in \mathbb{N}^d$, we define $m(J, v)$ as the length of the shortest $J$-covering path starting with $v$, 0 if there is none.

Note $\mathcal{J}_i = \{J \subseteq [1,d] \mid |J| = i\}$ and we define the function $f$ as follows: for $1 \leq i \leq d$, $f(i) = \max\{m(J_i, v) \mid J_i \in \mathcal{J}_i, v \in \mathbb{N}^d\}$. We will see that $f$ is always well defined, in $\mathbb{N}$.

▷ Claim A.4. $f(0) = 1$.

**Proof.** From any vector $v \in \mathbb{N}^d$, the path with one element $v$ is $\emptyset$-covering. ◀

▷ Claim A.5. For all $0 \leq i < d$, $f(i+1) \leq (N.f(i))^{i+1} + f(i)$.

**Proof.** Let $J \in \mathcal{J}_{i+1}$ and $v \in \mathbb{N}^d$ such that there exists a $J$-covering path starting with $v$. Note $\rho = v_0 \xrightarrow{t^1} \ldots \xrightarrow{t^m} v_m$ the shortest such path.

**First case: $\rho$ is $(J, N.f(i))$-bounded.** Assume, for sake of contradiction, that for some $k < \ell$, for all $j \in J$, $v_k(j) = v_\ell(j)$. Then we show that $v_0 \rightarrow \ldots v_k \rightarrow \overline{v}_{\ell+1} \ldots \rightarrow \overline{v}_m$ is also a $J$-correct path, with the vectors $(\overline{v}_{\ell'})_{\ell < \ell' \leq m}$, defined as follows.

$$\overline{v}_{\ell+1}(j) = \begin{cases} v_{\ell+1}(j) & \text{for all } j \in J \\ \max(0, (v_k(j) + t_b^{\ell+1}(j) - t_{nb}^{\ell+1}(j))) & \text{otherwise.} \end{cases}$$

And for all $\ell + 1 < \ell' \leq m$,

$$\overline{v}_{\ell'}(j) = \begin{cases} v_{\ell'}(j) & \text{for all } j \in J \\ \max(0, (\overline{v}_{\ell'-1}(j) + t_b^{\ell'}(j) - t_{nb}^{\ell'}(j))) & \text{otherwise.} \end{cases}$$

Then $v_0 \rightarrow \ldots v_k \rightarrow \overline{v}_{\ell+1} \ldots \rightarrow \overline{v}_m$ is also a $J$-correct path. Indeed, since $v_k(j) = v_\ell(j)$ for all $j \in J$, we have that $\overline{v}_{\ell+1}(j) = v_{\ell+1}(j) = \max(0, (v_\ell(j) + t_b^{\ell+1}(j) - t_{nb}^{\ell+1}(j))) = \max(0, (v_k(j) + t_b^{\ell+1}(j) - t_{nb}^{\ell+1}(j)))$. Moreover, for $j \in J$, since $v_\ell(j) + t_b^{\ell+1}(j) \geq 0$, we get that $v_k(j) + t_b^{\ell+1}(j) \geq 0$. By definition, for $j \notin J$, $\overline{v}_{\ell+1}(j) = \max(0, (v_k(j) + t_b^{\ell+1}(j) - t_{nb}^{\ell+1}(j)))$. Hence, $v_k \rightarrow^{t^{\ell+1}} \overline{v}_{\ell+1}$, and $v_0 \rightarrow^{t^1} \ldots v_k \rightarrow^{t^{\ell+1}} \overline{v}_{\ell+1}$ is $J$-correct. Now let $\ell < \ell' < m$. By definition, for $j \in J$, $\overline{v}_{\ell'+1}(j) = v_{\ell'+1}(j)$. Then, $\overline{v}_{\ell'+1}(j) = \max(0, (v_{\ell'}(j) + t_b^{\ell'+1}(j) - t_{nb}^{\ell'+1}(j))) = \max(0, (\overline{v}_{\ell'}(j) + t_b^{\ell'+1}(j) - t_{nb}^{\ell'+1}(j)))$. Again, since $\rho$ is $J$-correct, we deduce that for $j \in J$, $v_{\ell'}(j) + t_b^{\ell'+1}(j) \geq 0$, hence $\overline{v}_{\ell'}(j) + t_b^{\ell'+1}(j) \geq 0$. For $j \notin J$, $\overline{v}_{\ell'+1}(j) = \max(0, (\overline{v}_{\ell'}(j) + t_b^{\ell'+1}(j) - t_{nb}^{\ell'+1}(j)))$. So $\overline{v}_{\ell'} \rightarrow^{t^{\ell'+1}} \overline{v}_{\ell'+1}$, and $v_0 \rightarrow^{t^1} \ldots v_k \rightarrow^{t^{\ell'+1}} \overline{v}_{\ell'+1}$ is $J$-correct.

Then, $\rho' = v_0 \rightarrow \ldots v_k \rightarrow \overline{v}_{\ell+1} \ldots \rightarrow \overline{v}_m$ is a $J$-correct path, and since $\overline{v}_m(j) = v_m(j)$ for all $j \in J$, it is also $J$-covering, contradicting the fact that $\rho$ is minimal.

Hence, for all $k < \ell$, there exists $j \in J$ such that $v_k(j) \neq v_\ell(j)$. The length of such a path is at most $(N.f(i))^{i+1}$, so $m(J, v) \leq (N.f(i))^{i+1} \leq (N.f(i))^{i+1} + f(i)$.

**Second case: $\rho$ is not $(J, N.f(i))$-bounded.** We can then split $\rho$ into two paths $\rho_1\rho_2$ such that $\rho_1$ is $(J, N.f(i))$-bounded and $\rho_2 = v'_0 \ldots v'_n$ is such that $v'_0(j) \geq N.f(i)$ for some $j \in J$. As we have just seen, $|\rho_1| \leq (N.f(i))^{i+1}$.

Note $J' = J \setminus \{j\}$ with $j$ such that $v'_0(j) \geq N.f(i)$. Note that $\rho_2$ is $J'$-covering, therefore, by definition of $f$, there exists a $J'$-covering execution $\overline{\rho} = w_0 \ldots w_k$ with $w_0 = v'_0$, and such that $|\overline{\rho}| \leq f(i)$. Also, by definition of $N$, for all $1 \leq j' \leq d$, for all $(t_b, t_{nb}) \in T$, $N \geq |t_b(j')| + |t_{nb}(j')|$, then $t_b(j') \geq -N$, and $t_b(j') - t_{nb}(j') \geq -N$. Hence, for all $v \in \mathbb{N}^d$, $1 \leq j' \leq d$, and $c \in \mathbb{N}$ such that $v(j') \geq N + c$, for all $(t_b, t_{nb}) \in T$, $(v + t_b)(j') \geq c$ and $(v + t_b - t_{nb})(j') \geq c$. Now, since $w_0 = v'_0$, we get $w_0(j) \geq N.f(i)$. We deduce two things: first, for all $0 \leq \ell < k$, if $t = (t_b, t_{nb}) \in T$ is such that $w_\ell \rightarrow^t w_{\ell+1}$, it holds that $(w_\ell + t_b)(j) \geq N.(f(i) - \ell - 1)$. Since $k = f(i) - 1$, it yields that $\overline{\rho}$ is $J$-correct. Second, for all $0 \leq \ell \leq k$, $w_\ell(j) \geq N(f(i) - \ell)$. Again, $k = f(i) - 1$, so $w_k(j) \geq N \geq v_f(j)$. Hence $\overline{\rho}$ is also $J$-covering.

Since $\rho$ is the shortest $J$-covering path, we conclude that $|\rho| \leq (N.f(i))^{i+1} + f(i)$, and so $m(J, v) \leq (N.f(i))^{i+1} + f(i)$. ◀

We define a function $g$ such that $g(0) = 1$ and $g(i+1) = (N+1)^d (g(i))^d$ for $0 \leq i < d$; then $f(i) \leq g(i)$ for all $1 \leq i \leq d$. Hence, $f(d) \leq g(d) \leq (N+1)^{d^{d+1}} \leq 2^{2^{cn \log n}}$ for some $n \geq \max(d, N, |v_0|)$ and a constant $c$ which does not depend on $d$, $v_0$, nor $v_f$ or the NB-VAS. Hence, we can cover vector $v_f$ from $v_0$ if and only if there exists a path (from $v_0$) of length $\leq 2^{2^{cn \log n}}$ which covers $v_f$. Hence, there is a non-deterministic procedure that guesses a path of length $\leq 2^{2^{cn \log n}}$, checks if it is a valid path and accepts it if and only if it covers $v_f$. As $|v_0| \leq n$, $|v_f| \leq n$ and for all $(t_b, t_{nb}) \in T$, $|t_b| + |t_{nb}| \leq n$, this procedure takes an exponential space in the size of the protocol. By Savitch theorem, there exists a deterministic procedure in exponential space for the same problem. ◀

We are now ready to prove that the SCover problem for NB-VAS is as hard as the SCover problem for NB-CM.

▶ **Lemma A.6.** *Cover[NB-CM] reduces to SCover in NB-VAS.*

**Proof.** Let a NB-CM $M = (\mathrm{Loc}, X, \Delta_b, \Delta_{nb}, \ell_{in})$, for which we assume wlog that it does not contain any self-loop (replace a self loop on a location by a cycle using an additional internal transition and an additional location). We note $X = \{\mathtt{x}_1, \ldots, \mathtt{x}_m\}$, and $\mathrm{Loc} = \{\ell_1 \ldots \ell_k\}$, with $\ell_1 = \ell_{in}$ and $\ell_k = \ell_f$, and let $d = k + m$. We define the NB-VAS $V = (T, v_0)$ of dimension $d$ as follows: it has one counter by location of the NB-CM, and one counter by counter of the NB-CM. The transitions will ensure that the sum of the values of the counters representing the locations of $M$ will always be equal to 1, hence a vector during an execution of $V$ will always represent a configuration of $M$. First, for a transition $\delta = (\ell_i, op, \ell_{i'}) \in \Delta$, we define $(t_\delta, t'_\delta) \in \mathbb{Z}^d \times \mathbb{N}^d$ by $t_\delta(i) = -1, t_\delta(i') = 1$ and,

- if $op = \bot$, then $t_\delta(y) = 0$ for all other $1 \leq y \leq d$, and $t'_\delta = \mathbf{0}_d$ (where $\mathbf{0}_d$ is the null vector of dimension $d$), i.e. no other modification is made on the counters.
- if $op = \mathtt{x}_j+$, then $t_\delta(k+j) = 1$, and $t_\delta(y) = 0$ for all other $1 \leq y \leq d$, and $t'_\delta = \mathbf{0}_d$, i.e. the blocking part of the transition ensures the increment of the corresponding counter, while the non-blocking part does nothing.
- if $op = \mathtt{x}_j-$, then $t_\delta(k+j) = -1$, and $t_\delta(y) = 0$ for all other $1 \leq y \leq d$, and $t'_\delta = \mathbf{0}_d$, i.e. the blocking part of the transition ensures the decrement of the corresponding counter, while the non-blocking part does nothing. .
- if $op = \mathtt{nb}(\mathtt{x}_j-)$, then $t_\delta(y) = 0$ for all other $1 \leq y \leq d$, and $t'_\delta(k+j) = -1$ and $t'_\delta(y) = 0$ for all other $1 \leq y \leq d$, i.e. the blocking part of the transition only ensures the change in the location, and the non-blocking decrement of the counter is ensured by the non-blocking part of the transition.

We then let $T = \{t_\delta \mid \delta \in \Delta\}$, and $v_0$ is defined by $v_0(1) = 1$ and $v_0(y) = 0$ for all $2 \leq y \leq d$. We also fix $v_f$ by $v_f(k) = 1$, and $v_f(y) = 0$ for all other $1 \leq y \leq d$. One can prove that $v_f$ is covered in $V$ if and only if $\ell_f$ is covered in $M$. ◀

Putting together Lemma A.3 and Lemma A.6, we obtain the proof of Theorem 3.3.

## A.2 Proof of Theorem 3.5

In this subsection, we prove Theorem 3.5 by proving that the SCover[NB+R-CM] problem is Expspace hard. Put together with Theorem 3.3, it will prove the Expspace-completeness of SCover[NB+R-CM].

### A.2.1 Proofs on the Pocedural NB-CM Defined in Section 3

We formalize some properties on the procedural NB-CM presented in Section 3 used in the proof.

About the procedural NB-CM $\texttt{TestSwap}_i$, we use this proposition from [16, 8].

▶ **Proposition A.7** ([16, 8]). *Let $0 \le i < n$, and $\overline{\mathtt{x}} \in \overline{Y}_i$. For all $v, v' \in \mathbb{N}^{X'}$, for $\ell \in \{\ell_z^{\mathtt{TS},i,\mathtt{x}}, \ell_{nz}^{\mathtt{TS},i,\mathtt{x}}\}$, we have $(\ell_{in}^{\mathtt{TS},i}, v) \rightsquigarrow^* (\ell, v')$ in $\texttt{TestSwap}_i(\overline{\mathtt{x}})$ if and only if :*

- *(PreTest1): for all $0 \le j < i$, for all $\overline{\mathtt{x}}_j \in \overline{Y}_j$, $v(\overline{\mathtt{x}}_j) = 2^{2^j}$ and for all $\mathtt{x}_j \in Y_j$, $v(\mathtt{x}_j) = 0$;*
- *(PreTest2): $v(\overline{\mathtt{s}}_i) = 2^{2^i}$ and $v(\mathtt{s}_i) = 0$;*
- *(PreTest3): $v(\mathtt{x}) + v(\overline{\mathtt{x}}) = 2^{2^i}$;*
- *(PostTest1): For all $\mathtt{y} \notin \{\mathtt{x}, \overline{\mathtt{x}}\}$, $v'(\mathtt{y}) = v(\mathtt{y})$;*
- *(PostTest2): either (i) $v(\overline{\mathtt{x}}) = v'(\mathtt{x}) = 0$, $v(\mathtt{x}) = v'(\overline{\mathtt{x}})$ and $\ell = \ell_z^i$, or (ii) $v'(\overline{\mathtt{x}}) = v(\overline{\mathtt{x}}) > 0$, $v'(\mathtt{x}) = v(\mathtt{x})$ and $\ell = \ell_{nz}^{\mathtt{TS},i,\mathtt{x}}$.*

*Moreover, if for all $0 \le j \le n$, and any counter $\mathtt{x} \in Y_j \cup \overline{Y}_j$, $v(\mathtt{x}) \le 2^{2^j}$, then for all $0 \le j \le n$, and any counter $\mathtt{x} \in Y_j \cup \overline{Y}_j$, the value of $\mathtt{x}$ will never go above $2^{2^j}$ during the execution.*

Note that for a valuation $v \in \mathbb{N}^{X'}$ that meets the requirements (PreTest1), (PreTest2) and (PreTest3), there is only one configuration $(\ell, v')$ with $\ell \in \{\ell_z^{\mathtt{TS},i,\mathtt{x}}, \ell_{nz}^{\mathtt{TS},i,\mathtt{x}}\}$ such that $(\ell_{in}, v) \rightsquigarrow^* (\ell, v')$.

#### Procedural NB-CM $\texttt{Rst}_i$.

We shall now prove that the procedural NB-CMs we defined and displayed in Section 3 meet the desired requirements. For all $0 \le i \le n$, any procedural NB-CM $\texttt{Rst}_i$ enjoys the following property:

▶ **Proposition A.8.** *For all $0 \le i \le n$, for all $v \in \mathbb{N}^{X'}$ such that*

- *(PreRst1): for all $0 \le j < i$, for all $\overline{\mathtt{x}} \in \overline{Y}_j$, $v(\overline{\mathtt{x}}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v(\mathtt{x}) = 0$,*

*for all $v' \in \mathbb{N}^{X'}$, if $(\ell_{in}^{\mathtt{R},i}, v) \rightsquigarrow^* (\ell_{out}^{\mathtt{R},i}, v')$ in $\texttt{Rst}_\texttt{i}$ then*

- *(PostRst1): for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v'(\mathtt{x}) = \max(0, v(\mathtt{x}) - 2^{2^i})$,*
- *(PostRst2): for all $\mathtt{x} \notin Y_i \cup \overline{Y}_i$, $v'(\mathtt{x}) = v(\mathtt{x})$.*

**Proof of Proposition A.8.** For $\texttt{Rst}_0$, (PreRst1) trivially holds, and it is easy to see that (PostRst1) and (PostRst2) hold. Now fix $0 \le i < n$, and consider the procedural-NB-CM $\texttt{Rst}_{\texttt{i+1}}$. Let $v_0 \in \mathbb{N}^{X'}$ such that for all $0 \le j < i+1$, for all $\overline{\mathtt{x}} \in \overline{Y}_j$, $v_0(\overline{\mathtt{x}}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v_0(\mathtt{x}) = 0$, and let $v_f$ such that $(\ell_{in}^{\mathtt{R},i}, v_0) \rightsquigarrow^+ (\ell_{out}^{\mathtt{R},i}, v_f)$ in $\texttt{Rst}_i$.

First, we show the following property.

**Property** $(*)$: if there exist $v, v' \in \mathbb{N}^{X'}$ such that $v(\overline{\mathtt{z}}_i) = k$, $(\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{z}}, v')$ with no other visit of $\ell_z^{\mathtt{TS},i,\mathtt{z}}$ in between, then $v'(\overline{\mathtt{z}}_i) = 2^{2^i}$, $v'(\mathtt{z}_i) = 0$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v'(\mathtt{x}) = \max(0, v(\mathtt{x}) - k)$, and $v'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$.

If $k = 0$, then Proposition A.7 ensures that $v'(\overline{\mathtt{z}}_i) = 2^{2^i}$, $v'(\mathtt{z}_i) = 0$, and for all other $\mathtt{x} \in X'$, $v'(\mathtt{x}) = v(\mathtt{x})$. Otherwise, assume that the property holds for some $k \ge 0$ and consider $(\ell_{in}^{\mathtt{TS},i,\overline{\mathtt{z}}}, v) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\overline{\mathtt{z}}}, v')$ with no other visit of $\ell_z^{\mathtt{TS},i,\mathtt{z}}$ in between, and $v(\overline{\mathtt{z}}_i) = k+1$. Here, since $v(\overline{\mathtt{z}}_i) = k+1$, Proposition A.7 and the construction of the procedural-NB-CM ensure that $(\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v) \rightsquigarrow^* (\ell_{nz}^{\mathtt{TS},i,\mathtt{z}}, v) \rightsquigarrow (\ell_2^{\mathtt{R},i+1}, v) \rightsquigarrow^* (\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v_1)$ with $v_1(\overline{\mathtt{z}}_i) = k$, $v_1(\mathtt{z}_i) = v(\mathtt{z}_i) + 1$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v_1(\mathtt{x}) = \max(0, v(\mathtt{x}) - 1)$, and for all other $\mathtt{x} \in X'$, $v_1(\mathtt{x}) = v(\mathtt{x})$.

Induction hypothesis tells us that $(\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v_1) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{z}}, v')$ with $v'(\overline{\mathtt{z}}_i) = 2^{2^i}$, $v'(\mathtt{z}_i) = 0$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v'(\mathtt{x}) = \max(0, v(\mathtt{x}) - k - 1)$, and $v'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$.

Next, we show the following.

**Property** $(**)$: if there exist $v, v' \in \mathbb{N}^{X'}$ such that $v(\overline{\mathtt{y}}_i) = k$, $v(\overline{\mathtt{z}}_i) = 2^{2^i}$, $v(\mathtt{z}_i) = 0$, and $(\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{y}}, v')$ with no other visit of $\ell_z^{\mathtt{TS},i,\mathtt{y}}$ in between, then $v'(\overline{\mathtt{y}}_i) = 2^{2^i}$, $v'(\overline{y_i}) = 0$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v'(\mathtt{x}) = \max(0, v(\mathtt{x}) - k.2^{2^i})$, and $v'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$.

If $k = 0$, then Proposition A.7 ensures that $v'(\overline{\mathtt{y}}_i) = 2^{2^i}$, $v'(\mathtt{y}_i) = 0$, and $v'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$. Otherwise, assume that the property holds for some $k \geq 0$ and consider $(\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{y}}, v')$ with no other visit of $\ell_z^{\mathtt{TS},i,\mathtt{y}}$ in between, and $v(\overline{\mathtt{y}}_i) = k+1$. Again, since $v(\overline{\mathtt{y}}_i) = k+1$, Proposition A.7 and the construction of the procedural-NB-CM ensure that $(\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v) \rightsquigarrow^* (\ell_{nz}^{\mathtt{TS},i,\mathtt{y}}, v) \rightsquigarrow (\ell_{in}^{\mathtt{R},i+1}, v) \rightsquigarrow^* (\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v_1) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{z}}, v_1') \rightsquigarrow (\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v_1')$, with $v_1(\overline{\mathtt{y}}_i) = v(\overline{\mathtt{y}}_i) - 1 = k$, $v_1(\mathtt{y}_i) = v(\mathtt{y}_i) + 1$, $v_1(\overline{\mathtt{z}}_i) = v(\overline{\mathtt{z}}_i) - 1 = 2^{2^i} - 1$, $v_1(\mathtt{z}_i) = v(\mathtt{z}_i) + 1 = 1$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v_1(\mathtt{x}) = \max(0, v(\mathtt{x}) - 1)$, and for all other $\mathtt{x} \in X'$, $v_1(\mathtt{x}) = v(\mathtt{x})$. By Property $(*)$, $v_1'(\overline{\mathtt{z}}_i) = 2^{2^i}$, $v_1'(\mathtt{z}_i) = 0$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v_1'(\mathtt{x}) = \max(0, v(\mathtt{x}) - 2^{2^i})$, and $v_1'(\mathtt{x}) = v_1(\mathtt{x})$ for all other $\mathtt{x} \in X'$. Induction hypothesis allows to conclude that since $(\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v_1') \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{y}}, v')$, $v'(\overline{\mathtt{y}}_i) = 2^{2^i}$, $v'(\mathtt{y}_i) = 0$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v'(\mathtt{x}) = \max(0, v_1'(\mathtt{x}) - k.2^{2^i}) = \max(0, v(\mathtt{x}) - (k+1).2^{2^i})$, and $v'(\mathtt{x}) = v_1'(\mathtt{x}) = v(\mathtt{x})$ for all other $\mathtt{x} \in X'$.

Since $(\ell_{in}^{\mathtt{R},i}, v_0) \rightsquigarrow^+ (\ell_{out}^{\mathtt{R},i}, v_f)$, we know that $(\ell_{in}^{\mathtt{R},i}, v_0) \rightsquigarrow^* (\ell_{in}^{\mathtt{TS},i,\mathtt{z}}, v) \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{z}}, v') \rightsquigarrow (\ell_{in}^{\mathtt{TS},i,\mathtt{y}}, v') \rightsquigarrow^* (\ell_z^{\mathtt{TS},i,\mathtt{y}}, v'') \rightsquigarrow (\ell_{out}^{\mathtt{R},i}, v_f)$. By construction, $v(\overline{\mathtt{y}}_i) = 2^{2^i} - 1$, $v(\overline{\mathtt{z}}_i) = 2^{2^i} - 1$, $v(\mathtt{z}_i) = 1$, $v(\mathtt{z}_i) = 1$, for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v(\mathtt{x}) = \max(0, v_0(\mathtt{x}) - 1)$, and for all other counter $\mathtt{x}$, $v(\mathtt{x}) = v_0(\mathtt{x})$. By Property $(*)$, $v'(\overline{\mathtt{z}}_i) = 2^{2^i} = v_0(\overline{\mathtt{z}}_i)$, $v'(\mathtt{z}_i) = 0 = v_0(\mathtt{z}_i)$, for all $\mathtt{x} \in Y_i \cup \overline{Y_{i+1}}$, $v'(\mathtt{x}) = \max(0, v_0(\mathtt{x}) - 2^{2^i})$ and for all other $\mathtt{x} \in X'$, $v'(\mathtt{x}) = v(\mathtt{x})$. By Property $(**)$, $v''(\overline{\mathtt{y}}_i) = 2^{2^i} = v_0(\overline{\mathtt{y}}_i)$, $v''(\mathtt{y}_i) = 0 = v_0(\mathtt{y}_i)$, for all $\mathtt{x} \in Y_i \cup \overline{Y_{i+1}}$, $v''(\mathtt{x}) = \max(0, v_0(\mathtt{x}) - 2^{2^i} - (2^{2^i} - 1).2^{2^i}) = \max(0, v_0(\mathtt{x}) - 2^{2^i}.2^{2^i}) = \max(0, v_0(\mathtt{x}) - 2^{2^{i+1}})$, and for all other $\mathtt{x} \in X'$, $v''(\mathtt{x}) = v'(\mathtt{x}) = v_0(\mathtt{x})$. ◀

We get the immediate corollary:

▶ **Lemma A.9.** *Let $0 \leq i \leq n$, and $v \in \mathbb{N}^{X'}$ satisfying (PreRst1) for $\mathtt{Rst}_i$. If $v$ is $i$-bounded, then the unique configuration such that $(\ell_{in}^{\mathtt{R},i}, v) \rightsquigarrow^+ (\ell_{out}^{\mathtt{R},i}, v')$ in $\mathtt{Rst}_i$ is defined $v'(\mathtt{x}) = 0$ for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$ and $v'(\mathtt{x}) = v(\mathtt{x})$ for all $\mathtt{x} \notin Y_i \cup \overline{Y}_i$.*

▶ **Proposition A.10.** *Let $0 \leq i \leq n$, and let $v \in \mathbb{N}^{X'}$ satisfying (PreRst1) for $\mathtt{Rst_i}$. If for all $0 \leq j \leq n$, $v$ is $j$-bounded, then for all $(\ell, v') \in Loc^{\mathtt{R},i} \times \mathbb{N}^{X'}$ such that $(\ell_{in}^{\mathtt{R},i}, v) \rightsquigarrow^* (\ell, v')$ in $\mathtt{Rst}_i$, $v'$ is $j$-bounded for all $0 \leq j \leq n$.*

**Proof.** We will prove the statement of the property along with some other properties: (1) if $\ell$ is not a state of $\mathtt{TestSwap}_i(\overline{\mathtt{z}}_i)$ or $\mathtt{TestSwap}_i(\overline{\mathtt{y}}_i)$, then for all $0 \leq j < i$, for all $\mathtt{x} \in \overline{Y_j}$, $v'(\mathtt{x}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v'(\mathtt{x}) = 0$, and $v'(\overline{\mathtt{s}_i}) = 2^{2^i}$ and $v'(\mathtt{s}_i) = 0$. (2) if $\ell$ is not a state of $\mathtt{TestSwap}_i(\overline{\mathtt{z}}_i)$ or $\mathtt{TestSwap}_i(\overline{\mathtt{y}}_i)$ and if $\ell \neq \ell_1^{\mathtt{R},i+1}$, then $v'(\mathtt{y}_i) + v'(\overline{\mathtt{y}}_i) = 2^{2^i}$, and if $\ell \neq \ell_3^{\mathtt{R},i+1}$, then $v'(\mathtt{z}_i) + v'(\overline{\mathtt{z}}_i) = 2^{2^i}$.

For $\mathtt{Rst}_0$, the property is trivial. Let $0 \leq i < n$, and a valuation $v \in \mathbb{N}^{X'}$ such that for all $0 \leq j \leq i$, for all $\overline{\mathtt{x}} \in \overline{Y}_j$, $v(\overline{\mathtt{x}}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v(\mathtt{x}) = 0$, and such that, for all $0 \leq j \leq n$, $v$ is $j$-bounded. Let now $(\ell, v')$ such that $(\ell_{in}^{\mathtt{R},i+1}, v) \rightsquigarrow^* (\ell, v')$ in $\mathtt{Rst}_{i+1}$. We prove the property by induction on the number of occurences of $\ell_{in}^{\mathtt{TS},i,z}$ and $\ell_{in}^{\mathtt{TS},i,y}$. If there is no occurence of such state between in $(\ell_{in}^{\mathtt{R},i+1}, v) \rightsquigarrow^* (\ell, v')$, then, for all $\mathtt{x} \in Y_j \cup \overline{Y_j} \cup \{\mathtt{s}_i, \overline{\mathtt{s}_i}\}$ and $j \neq i$, $j \neq i+1$, then $v'(\mathtt{x}) = v(\mathtt{x})$ and so $v'$ is $j$-bounded. Furthermore, for $\mathtt{x} \in Y_i \cup Y_{i+1} \cup \overline{Y_{i+1}}$,

$v'(\mathtt{x}) \leq v(\mathtt{x})$, and for all $\mathtt{x} \in \overline{Y_i}$, $v'(\mathtt{x}) \leq v(\mathtt{x}) + 1 = 1$. The property (2) is easily verified. Hence the properties hold.

Assume now we proved the properties for $k$ occurrences of $\ell_{in}^{\mathtt{TS},i,z}$ and $\ell_{in}^{\mathtt{TS},i,y}$, and let us prove the clam for $k+1$ such occurrences. Note $\ell_{k+1} \in \{\ell_{in}^{\mathtt{TS},i,z}, \ell_{in}^{\mathtt{TS},i,y}\}$ the last occurence such that: $(\ell_{in}^{\mathtt{R},i+1}, v) \rightsquigarrow^+ (\ell_k, v_k) \rightsquigarrow (\ell_{k+1}, v_{k+1}) \rightsquigarrow^* (\ell, v')$. By induction hypothesis, $v_k$ is $j$-bounded for all $0 \leq j \leq n$ and it respects (1) and (2), and by construction, $(\ell_k, \bot, \ell_{k+1})$ and $\ell_k \neq \ell_1^{\mathtt{R},i+1}$, $\ell_k \neq \ell_3^{\mathtt{R},i+1}$, hence $v_{k+1}$ is $j$-bounded for all $0 \leq j \leq n$ and respects (PreTest1), (PreTest2), and (PreTest3) for $\mathtt{TestSwap}_i(\overline{\mathtt{z}}_i)$ and $\mathtt{TestSwap}_i(\overline{\mathtt{y}}_i)$. As a consequence, if $\ell$ is a state of one of this machine such that $(\ell_{k+1}, v_{k+1}) \rightsquigarrow^* (\ell, v')$, then by Proposition A.7, for all $0 \leq j \leq n$, as $v_{k+1}$ is $j$-bounded, so is $v'$.

Assume now $\ell$ to not be a state of one of the two machines. And keep in mind that $v_{k+1}$ respects (1) and (2). Then, either $\ell = \ell_{out}^{\mathtt{R},i+1}$ and so $v'(\mathtt{x}) = v_{k+1}(\mathtt{x})$ for all $\mathtt{x} \in Y_j \cup \overline{Y}_j$ for all $j \neq i$, and $v'(\overline{\mathtt{y}_i}) = 2^{2^i}$ and $v'(\mathtt{y}_i) = 0$ and so the claim holds, either $\ell \in \{\ell_{in}^{\mathtt{R},i+1}, \ell_{j'}^{\mathtt{R},i+1}\}_{j'=1,2,3,4,5,6,\ldots,7}$. In this case, the execution is such that: $(\ell_{k+1}, v_{k+1}) \rightsquigarrow^+ (\ell_{nz,k+1}, v_{k+1}) \rightsquigarrow^* (\ell, v')$, where if $\ell_{k+1} = \ell_{in}^{\mathtt{TS},i,z}$, $\ell_{nz,k+1} = \ell_{nz}^{\mathtt{TS},i,z}$ and otherwise $\ell_{nz,k+1} = \ell_{nz}^{\mathtt{TS},i,y}$. In any cases, for all $j \neq i$, $j \neq i+1$, $\mathtt{x} \in Y_j \cup \overline{Y}_j \cup \{\mathtt{s}_i, \overline{\mathtt{s}_i}\}$, $v'(\mathtt{x}) = v_{k+1}(\mathtt{x})$, hence (1) holds and $v'$ is $j$-bounded for all $j < i$ and $j > i+1$.

Observe as well that for all $\mathtt{x} \in Y_{i+1} \cup \overline{Y}_{i+1}$, $v'(\mathtt{x}) \leq v_{k+1}(\mathtt{x})$, and so $v'$ is $i+1$-bounded. The last thing to prove is that (2) holds. This is direct from the fact that $v_{k+1}$ respects (2). ◀

About the procedural NB-CM $\mathtt{Inc}_i$, we use this proposition from [16, 8].

▶ **Proposition A.11** ([16, 8]). *For all $0 \leq i < n$, for all $v, v' \in \mathbb{N}^{X'}$, $(\ell_{in}^{\mathtt{Inc},i}, v) \rightsquigarrow^* (\ell_{out}^{\mathtt{Inc},i}, v')$ in $\mathtt{Inc}_i$ if and only if:*

- *(PreInc1) for all $0 \leq j < i$, for all $\mathtt{x} \in \overline{Y}_j$, $v(\mathtt{x}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v(\mathtt{x}) = 0$;*
- *(PreInc2) for all $\mathtt{x} \in \overline{Y}_i$, $v(\mathtt{x}) = 0$,*
- *(PostInc1) for all $\mathtt{x} \in \overline{Y}_i$, $v'(\mathtt{x}) = 2^{2^i}$;*
- *(PostInc2) for all $\mathtt{x} \notin Y_i$, $v'(\mathtt{x}) = v(\mathtt{x})$.*

*Moreover, if for all $0 \leq j \leq n$, $v$ is $j$-bounded, then for all $(\ell, v'')$ such that $(\ell_{in}^{\mathtt{Inc},i}, v) \rightsquigarrow^* (\ell, v'')$ in $\mathtt{Inc}_i$, then $v''$ is $j$-bounded for all $0 \leq j \leq n$.*

## Procedural NB-CM RstInc.

We shall now prove the properties in the procedural NB-CM RstInc defined in Section 3. The next proposition establishes the correctness of the construction RstInc.

▶ **Proposition A.12.** *Let $v \in \mathbb{N}^{X'}$ be a valuation such that for all $0 \leq i \leq n$ and for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v(\mathtt{x}) \leq 2^{2^i}$. Then the unique valuation $v' \in \mathbb{N}^{X'}$ such that $(\ell_a, v) \rightsquigarrow^* (\ell_b, v')$ in* RstInc *satisfies the following: for all $0 \leq i \leq n$, for all $\mathtt{x} \in \overline{Y}_i$, $v'(\mathtt{x}) = 2^{2^i}$ and for all $\mathtt{x} \in Y_i$, $v'(\mathtt{x}) = 0$. Moreover, for all $(\ell, v'')$ such that $(\ell_a, v) \rightsquigarrow^* (\ell, v'')$ in* RstInc*, for all $0 \leq i \leq n$, $v''$ is $i$-bounded.*

**Proof of Proposition A.12.** We can split the execution in $(\ell_a, v) \rightsquigarrow (\ell_{in}^{\mathtt{R},0}, v) \rightsquigarrow^* (\ell_{out}^{\mathtt{R},0}, v_0) \rightsquigarrow (\ell_{in}^{\mathtt{Inc},0}, v_0) \rightsquigarrow^* (\ell_{out}^{\mathtt{Inc},0}, v_0') \rightsquigarrow (\ell_{in}^{\mathtt{R},1}, v_0') \rightsquigarrow^* (\ell_{out}^{\mathtt{R},1}, v_1) \rightsquigarrow^* (\ell_{in}^{\mathtt{Inc},n-1}, v_{n-1}) \rightsquigarrow^* (\ell_{out}^{\mathtt{Inc},n-1}, v_{n-1}') \rightsquigarrow (\ell_{in}^{\mathtt{R},n}, v_{n-1}') \rightsquigarrow^* (\ell_{out}^{\mathtt{R},n}, v_n) \rightsquigarrow (\ell_b, v')$, with $v' = v_n$ and $v = v_{-1}'$. We show that for all $0 \leq i \leq n$:

- $P_1(i)$: For all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v_i(\mathtt{x}) = 0$, and for all $\mathtt{x} \notin (Y_i \cup \overline{Y}_i)$, $v_i(\mathtt{x}) = v_{i-1}'(\mathtt{x})$.
- $P_2(i)$: For all $0 \leq j < i$, for all $\mathtt{x} \in Y_j$, $v_{i-1}'(\mathtt{x}) = 0$ and for all $\mathtt{x} \in \overline{Y}_j$, $v_{i-1}'(\mathtt{x}) = 2^{2^j}$, and for all other $\mathtt{x} \in X'$, $v_i'(\mathtt{x}) = v_i(\mathtt{x})$.

- $P_3(i)$: For all $v''$ such that $(\ell_a, v) \rightsquigarrow^* (\ell, v'') \rightsquigarrow^* (\ell_{out}^{\mathtt{R},i}, v_i)$, $v''$ is $i$-bounded, for all $0 \leq i \leq n$.

For $k = 0$, Lemma A.9 implies that for all $\mathtt{x} \in Y_0 \cup \overline{Y}_0$, $v_0(\mathtt{x}) = 0$, and that for all other $\mathtt{x} \in X'$, $v_0(\mathtt{x}) = v(\mathtt{x})$. Moreover, for all $v''$ such that $(\ell_{in}^{\mathtt{R},0}, v) \rightsquigarrow^* (\ell, v'') \rightsquigarrow^* (\ell_{out}^{\mathtt{R},0}, v_0)$, Proposition A.10 ensures that $v''$ is $i$-bounded, for all $0 \leq i \leq n$. $P_2(0)$ is trivially true.

Let $0 \leq k < n$, and assume that $P_1(k)$, $P_2(k)$ and $P_3(k)$ hold. $P_1(k)$ and $P_2(k)$ and Proposition A.11 imply that for all $\mathtt{x} \in \overline{Y}_k$, $v'_k(\mathtt{x}) = 2^{2^k}$, and that for all other counter $\mathtt{x} \in X'$, $v'_k(\mathtt{x}) = v_k(\mathtt{x})$. Thanks to $P_1(k)$, $P_2(k+1)$ holds. Moreover, we also know by Proposition A.11 that for all $v''$ such that $(\ell_{out}^{\mathtt{R},k}, v_k) \rightsquigarrow (\ell_{in}^{\mathtt{Inc},k}, v_k) \rightsquigarrow^* (\ell, v'') \rightsquigarrow^* (\ell_{out}^{\mathtt{Inc},k}, v'_k)$, $v''$ is $i$-bounded for all $0 \leq i \leq n$. Since $v'_k$ is then $i$-bounded for all $0 \leq i \leq n$, and since $P_2(k)$ holds, Lemma A.9 implies that $v_{k+1}(\mathtt{x}) = 0$ for all $\mathtt{x} \in Y_{k+1} \cup \overline{Y}_{k+1}$, and that, for all other $\mathtt{x} \in X'$, $v_{k+1}(\mathtt{x}) = v'_k\mathtt{x})$. So $P_1(k+1)$ holds. Moreover, by Proposition A.10, for all $v''$ such that $(\ell_{out}^{\mathtt{Inc},k}, v'_k) \rightsquigarrow (\ell_{in}^{\mathtt{R},k+1}, v'_k) \rightsquigarrow^* (\ell, v'') \rightsquigarrow^* (\ell_{out}^{\mathtt{R},k+1}, v_{k+1})$, $v''$ is $i$-bounded for all $0 \leq i \leq n$. Hence $P_3(k+1)$ holds.

By $P_1(n)$, $v'(\mathtt{x}) = 0$ for all $\mathtt{x} \in Y_n$, and since $\overline{Y}_n = \emptyset$, $v'(\mathtt{x}) = 2^{2^n}$ for all $\mathtt{x} \in \overline{Y}_n$. Let $\mathtt{x} \notin (Y_n \cup \overline{Y}_n)$. Then $v'(\mathtt{x}) = v'_{n-1}(\mathtt{x})$, and by $P_2(n)$, for all $0 \leq i < n$, for all $\mathtt{x} \in \overline{Y}_i$, $v'(\mathtt{x}) = 2^{2^i}$, and for all $\mathtt{x} \in Y_i$, $v'(\mathtt{x}) = 0$. By $P_3(n)$, for all $(\ell, v'')$ such that $(\ell_a, v) \rightsquigarrow^* (\ell, v'')$ in $\mathtt{RstInc}$, for all $0 \leq i \leq n$, $v''$ is $i$-bounded. ◀

### A.2.2 Proofs of the Reduction

We are now ready to prove Theorem 3.5, i.e that the reduction is sound and complete. For some subset of counters $Y$, we will note $v_{|Y}$ for the valuation $v$ on counters $Y$, formally, $v_{|Y} : Y \to \mathbb{N}$ and is equal to $v$ on its domain.

▶ **Lemma A.13.** *If there exists $v \in \mathbb{N}^X$ such that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow_M^* (\ell_f, v)$, then there exists $v' \in \mathbb{N}^{X'}$ such that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell_f, v')$.*

**Proof.** From Proposition A.12, we have that $(\ell'_{in}, \mathbf{0}'_X) \rightsquigarrow_N^* (\ell_{in}, v_0)$ where $v_0$ is such that, for all $0 \leq j \leq n$, for all $\mathtt{x} \in \overline{Y}_j$, $v_0(\mathtt{x}) = 2^{2^j}$ and for all $\mathtt{x} \in Y_j$, $v_0(\mathtt{x}) = 0$. By construction of $N$, $(\ell_{in}, v_0) \rightsquigarrow_N^* (\ell_f, v')$ with $v'$ defined by: for all $0 \leq i < n$, for all $\mathtt{x} \in \overline{Y}_j$, $v'(\mathtt{x}) = 2^{2^j}$, for all $\mathtt{x} \in Y_j$, $v'(\mathtt{x}) = 0$, and, for all $\mathtt{x} \in X$, $v'(\mathtt{x}) = v(\mathtt{x})$. Note that in this path, there is no restore step. ◀

▶ **Lemma A.14.** *If there exists $v' \in \mathbb{N}^{X'}$ such that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell_f, v')$, then there exists $v \in \mathbb{N}^X$ such that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow_M^* (\ell_f, v)$.*

**Proof.** We will note $v_0$ the function such that for all $0 \leq i \leq n$, and for all $\mathtt{x} \in \overline{Y}_i$, $v_0(\mathtt{x}) = 2^{2^i}$ and for all $\mathtt{x} \in Y_i$, $v_0(\mathtt{x}) = 0$. Observe that there might be multiple visits of location $\ell_{in}$ in the execution of $N$, because of the restore transitions. The construction of $\mathtt{RstInc}$ ensures that, every time a configuration $(\ell_{in}, v)$ is visited, $v = v_0$. Formally, we show that for all $(\ell_{in}, v)$ such that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell_{in}, v)$, we have that $v = v_0$. First let $(\ell'_{in}, w) \rightsquigarrow_N^* (\ell'_{in}, w')$, with $w(\mathtt{x}) \leq 2^{2^i}$, and $\ell'_{in}$, $\ell_{in}$ not visited in between. Then for all $0 \leq i \leq n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $w'(\mathtt{x}) \leq 2^{2^i}$. Indeed, let $(\ell, \overline{w})$ be such that $(\ell'_{in}, w) \rightsquigarrow_N^* (\ell, \overline{w}) \rightsquigarrow_N (\ell'_{in}, w')$. By Proposition A.12, we know that, for all $0 \leq i \leq n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $\overline{w}(\mathtt{x}) \leq 2^{2^i}$. Since the last transition is a restore transition, we deduce that, for all $0 \leq i \leq n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $w'(\mathtt{x}) = \overline{w}(\mathtt{x}) \leq 2^{2^i}$.

- Let $v \in \mathbb{N}^{X'}$ be such that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell_{in}, v)$, and $(\ell_{in}, v)$ is the first configuration where $\ell_{in}$ is visited. The execution is thus of the form $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow_N^* (\ell'_{in}, w) \rightsquigarrow_N^* (\ell_{in}, v)$, with

$(\ell'_{in}, w)$ the last time $\ell'_{in}$ is visited. We have stated above that $w(\mathtt{x}) \leq 2^{2^i}$. Then, we have that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow^*_N (\ell'_{in}, w) \rightsquigarrow_N (\ell_a, w) \rightsquigarrow^*_N (\ell_b, v) \rightsquigarrow_N (\ell_{in}, v)$, and by Proposition A.12, $v = v_0$.

- Let now $v_k, v_{k+1} \in \mathbb{N}^{X'}$ be such that $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow^*_N (\ell_{in}, v_k) \rightsquigarrow^*_N (\ell_{in}, v_{k+1})$, and $v_k$ and $v_{k+1}$ are respectively the $k^{\text{th}}$ and the $(k+1)^{\text{th}}$ time that $\ell_{in}$ is visited, for some $k \geq 0$. Assume that $v_k = v_0$. We have $(\ell_{in}, v_k) \rightsquigarrow^*_N (\ell, v) \rightsquigarrow_N (\ell'_{in}, v) \rightsquigarrow^*_N (\ell'_{in}, \overline{v}) \rightsquigarrow_N (\ell_a, \overline{v}) \rightsquigarrow^*_N (\ell_b, v_{k+1}) \rightsquigarrow_N (\ell_{in}, v_{k+1})$. Since the test-free CM $M$ is 2EXP-bounded, and $v_k = v_0$, we obtain that for all $\mathtt{x} \in X = Y_n$, $v(\mathtt{x}) \leq 2^{2^n}$. For all $0 \leq i < n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v(\mathtt{x}) = v_0(\mathtt{x})$, then for all $0 \leq i \leq n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$, $v(\mathtt{x}) \leq 2^{2^i}$. Then, as proved above, $\overline{v}(\mathtt{x}) \leq 2^{2^i}$ for all $0 \leq i \leq n$, for all $\mathtt{x} \in Y_i \cup \overline{Y}_i$. By Proposition A.12, $v' = v_0$.

Consider now the execution $(\ell'_{in}, \mathbf{0}_{X'}) \rightsquigarrow^*_N (\ell_{in}, v) \rightsquigarrow^*_N (\ell_f, v')$, where $(\ell_{in}, v)$ is the last time the location $\ell_{in}$ is visited. Then, as proved hereabove, $v = v_0$. From the execution $(\ell_{in}, v) \rightsquigarrow^*_N (\ell_f, v')$, we can deduce an execution $(\ell_{in}, v_{|X}) \rightsquigarrow^*_M (\ell_f, v'_{|X})$. Since $v = v_0$ and for all $\mathtt{x} \in X = Y_n$, $v(\mathtt{x}) = 0$, we can conclude the proof. ◀

The two previous lemmas prove that the reduction is sound and complete. By Theorem 3.4, we proved the EXPSPACE-hardness of the problem, and so Theorem 3.5.

# B Proofs of Section 4

In this section, we present proofs omitted in Section 4.

## B.1 Proof of Theorem 4.1

We present here the proof of Theorem 4.1, the two lemmas of this subsection prove the soundness and completeness of the reduction presented in Section 4.1, put together with Theorem 3.3, it proves Theorem 4.1.

▶ **Lemma B.1.** *Let $C_0 \in \mathcal{I}$, $C_f \geq C_F$. If $C_0 \rightarrow^*_{\mathcal{P}} C_f$, then there exists $v \in \mathbb{N}^Q$ such that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_f, v)$.*

**Proof.** For all $q \in Q$, we let $v_q(q) = 1$ and $v_q(q') = 0$ for all $q' \in X$ such that $q' \neq q$. Let $n = ||C_0|| = C_0(q_{in})$, and let $C_0 C_1 \cdots C_m C_f$ be the configurations visited in $\mathcal{P}$. Then, applying the transition $(\ell_{in}, q_{in}+, \ell_{in})$, we get $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow (\ell_{in}, v^1) \rightsquigarrow \ldots \rightsquigarrow (\ell_{in}, v^n)$ with $v_0 = v^n$ and $v_0(q_{in}) = n$ and $v_0(\mathtt{x}) = 0$ for all $\mathtt{x} \neq q_{in}$. Let $i \geq 0$ and assume that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_{in}, C_i)$. We show that $(\ell_{in}, C_i) \rightsquigarrow^* (\ell_{in}, C_{i+1})$.

- If $C_i \xrightarrow{m}_{\mathcal{P}} C_{i+1}$, let $t = (q_1, !m, q'_1), t' = (q_2, ?m, q'_2) \in T$ such that $C_i(q_1) > 0$, $C_i(q_2) > 0$, $C_i(q_1) + C_i(q_2) \geq 2$, and $C_{i+1} = C_i - \wr q_1, q_2 \wr + \wr q'_1, q'_2 \wr$. Then $(\ell_{in}, C_i) \rightsquigarrow (\ell^1_{(t,t')}, v^1_i) \rightsquigarrow (\ell^2_{(t,t')}, v^2_i) \rightsquigarrow (\ell^3_{(t,t')}, v^3_i) \rightsquigarrow (\ell_{in}, v^4_i)$, with $v^1_i = C_i - v_{q_1}$, $v^2_i = v^1_i - v_{q_2}$, $v^3_i = v^2_i + v_{q'_1}$, $v^4_i = v^3_i + v_{q'_2}$. Observe that $v^4_i = C_{i+1}$ and then $(\ell_{in}, C_i) \rightsquigarrow^* (\ell_{in}, C_{i+1})$.
- If $C_i \xrightarrow{\tau}_{\mathcal{P}} C_{i+1}$, let $t = (q, \tau, q')$ such that $C_i(q) > 0$ and $C_{i+1} = C_i - \wr q \wr + \wr q' \wr$. Then, $(\ell_{in}, C_i) \rightsquigarrow (\ell_q, v^1_i) \rightsquigarrow (\ell_{in}, v^2_i)$ with $v^1_i = C_i - v_q$ and $v^2_i = v^1_i + v_{q'}$. Observe that $v^2_i = C_{i+1}$, then $(\ell_{in}, C_i) \rightsquigarrow^* (\ell_{in}, C_{i+1})$.
- If $C_i \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C_{i+1}$, let $t = (q, !m, q')$ such that $C_{i+1} = C_i - \wr q \wr + \wr q' \wr$, and $R(m) = \{q_1, \ldots, q_k\}$. Then $C_i(p_j) = 0$ for all $1 \leq j \leq k$. We then have that $(\ell_{in}, C_i) \rightsquigarrow (\ell_t, v^1_i) \rightsquigarrow (\ell^m_{t,q_1}, v^1_i) \rightsquigarrow \cdots \rightsquigarrow (\ell^m_{t,q_k}, v^1_i) \rightsquigarrow (\ell_{in}, v^2_i)$ with $v^1_i = C_i - v_q$ and $v^2_i = v^1_i + v_{q'}$. Indeed, $v^1_i(q_j) = 0$ for all $q_j \in R(m)$, so the transitions $(\ell^m_{t,q_j}, \mathtt{nb}(q_{j+1}-)), \ell^m_{t,q_{j+1}})$ do not change the value of the counters. Hence, $v^2_i = C_{i+1}$ and $(\ell_{in}, C_i) \rightsquigarrow^* (\ell_{in}, C_{i+1})$.

So we know that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_{in}, C_f)$. Moreover, since $C_f \geq C_F$, it holds that $C_f \geq v_{\mathbf{q_1}} + v_{\mathbf{q_2}} + \cdots + v_{\mathbf{q}_s}$. Then $(\ell_{in}, C_f) \rightsquigarrow^s (\ell_f, v)$ with $v = C_f - (v_{\mathbf{q_1}} + v_{\mathbf{q_2}} + \cdots + v_{\mathbf{q}_s})$. ◀

▶ **Lemma B.2.** *Let $v \in \mathbb{N}^Q$. If $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_f, v)$, then there exists $C_0 \in \mathcal{I}$, $C_f \geq C_F$ such that $C_0 \rightarrow^*_{\mathcal{P}} C_f$.*

**Proof.** Let $(\ell_{in}, v_0), (\ell_{in}, v_1) \ldots (\ell_{in}, v_n)$ be the projection of the execution of $M$ on $\{\ell_{in}\} \times \mathbb{N}^X$. We prove that, for all $0 \leq i \leq n$, there exists $C_0 \in \mathcal{I}$, and $C \geq v_i$ such that $C_0 \rightarrow^*_{\mathcal{P}} C$. For $i = 0$, we let $C_0$ be the empty multiset, and the property is trivially true. Let $0 \leq i < n$, and assume that there exists $C_0 \in \mathcal{I}$, $C \geq v_i$ such that $C_0 \rightarrow^*_{\mathcal{P}} C$.

- If $(\ell_{in}, v_i) \overset{\delta}{\rightsquigarrow} (\ell_{in}, v_{i+1})$ with $\delta = (\ell_{in}, q_{in}+, \ell_{in})$, then $v_{i+1} = v_i + v_{q_{in}}$. The execution $C_0 \rightarrow^*_{\mathcal{P}} C$ built so far cannot be extended as it is, since it might not include enough processes. Let $N$ be such that $C_0 \rightarrow_{\mathcal{P}} C_1 \rightarrow_{\mathcal{P}} \ldots \rightarrow_{\mathcal{P}} C_N = C$, and let $C'_0 \in \mathcal{I}$ with $C'_0(q_{in}) = C_0(q_{in}) + N + 1$. We build, for all $0 \leq j \leq N$, a configuration $C'_j$ such that $C'_0 \rightarrow^j_{\mathcal{P}} C'_j$, $C'_j \geq C_j$ and $C'_j(q_{in}) > C_j(q_{in}) + N - j$. For $j = 0$ it is trivial. Assume now that, for $0 \leq j < N$, $C'_j \geq C_j$ and that $C'_j(q_{in}) > C_j(q_{in}) + N - j$.
  If $C_j \xrightarrow{m}_{\mathcal{P}} C_{j+1}$ for $m \in \Sigma$, with $t_1 = (q_1, !m, q'_1)$ and $t_2 = (q_2, ?m, q'_2)$. Then, $C_{j+1} = C_j - \{\!\{q_1, q_2\}\!\} + \{\!\{q'_1, q'_2\}\!\}$. Moreover, $C'_j(q_1) \geq C_j(q_1) > 0$ and $C'_j(q_2) \geq C_j(q_2) > 0$ and $C'_j(q_1) + C'_j(q_2) \geq C_j(q_1) + C_j(q_2) \geq 2$. We let $C'_{j+1} = C'_j - \{\!\{q_1, q_2\}\!\} + \{\!\{q'_1, q'_2\}\!\}$, and $C'_j \xrightarrow{m}_{\mathcal{P}} C'_{j+1}$. It is easy to see that $C'_{j+1} \geq C_{j+1}$. Moreover, $C'_{j+1}(q_{in}) > C_{j+1}(q_{in}) + N - j > C_{j+1} + N - j - 1$.
  If $C_j \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C_{j+1}$ and for all $q \in R(m)$, $C'_j - \{\!\{q_1\}\!\}(q) = 0$, with $t = (q_1, !m, q_2)$, (respectively $C_j \xrightarrow{\tau}_{\mathcal{P}} C_{j+1}$ with $t = (q_1, \tau, q_2)$), we let $C'_{j+1} = C'_j - \{\!\{q_1\}\!\} + \{\!\{q_2\}\!\}$, and $C'_j \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C'_{j+1}$ (respectively $C'_j \xrightarrow{\tau}_{\mathcal{P}} C'_{j+1}$). Again, thanks to the induction hypothesis, we get that $C'_{j+1} \geq C_{j+1}$, and $C'_{j+1}(q_{in}) > C_{j+1}(q_{in}) + N - j > C_{j+1}(q_{in}) + N - j - 1$.
  If now $C_j \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C_{j+1}$, with $t_1 = (q_1, !m, q_2)$ and there exists $q'_1 \in R(m)$ such that $C'_j - \{\!\{q_1\}\!\}(q'_1) > 0$. Let $(q'_1, ?m, q'_2) \in T$, and then $C'_{j+1} = C'_j - \{\!\{q_1, q'_1\}\!\} + \{\!\{q_2, q'_2\}\!\}$. Since $C'_j \geq C_j$, $C'_j(q_1) \geq 1$, and since $C'_j - \{\!\{q_1\}\!\}(q'_1) > 0$, $C'_j(q'_1) \geq 1$ and $C'_j(q_1) + C'_j(q'_1) \geq 2$. Hence, $C'_j \xrightarrow{m}_{\mathcal{P}} C'_{j+1}$. We have that $C'_j(q'_1) > C_j(q'_1)$, so $C'_{j+1}(q'_1) \geq C_{j+1}(q'_1)$ and $C'_{j+1}(q) \geq C_{j+1}(q)$ for all other $q \in Q$. Hence $C'_{j+1} > C_{j+1}$. Also, $C_{j+1}(q_{in}) = C_j(q_{in}) + x$, with $x \in \{0, 1\}$. If $q'_1 \neq q_{in}$, then $C'_{j+1}(q_{in}) = C'_j(q_{in}) + y$, with $y \geq x$. Hence, since $C'_j(q_{in}) > C_j(q_{in}) + N - j$, we get $C'_{j+1}(q_{in}) > C_{j+1}(q_{in}) + N - j > C_{j+1}(q_{in}) + N - j - 1$. If $q'_1 = q_{in}$, then we can see that $C'_{j+1}(q_{in}) = C'_j(q_{in}) + y$, with $x - 1 \leq y \leq x$. In that case, $C'_{j+1}(q_{in}) > C_j(q_{in}) + N - j + y \geq C_j(q_{in}) + N - j + x - 1 \geq C_{j+1}(q_{in}) + N - j - 1$.
  So we have built an execution $C'_0 \rightarrow^*_{\mathcal{P}} C'_N$ such that $C'_N \geq C_N$ and $C'_N(q_{in}) > C_N(q_{in})$. Hence, $C'_N \geq v_{i+1}$.
- If $(\ell_{in}, v_i) \rightsquigarrow (\ell^1_{(t,t')}, v^1_i) \rightsquigarrow (\ell^2_{(t,t')}, v^2_i) \rightsquigarrow (\ell^3_{(t,t')}, v^3_i) \rightsquigarrow (\ell_{in}, v_{i+1})$, with $t = (q_1, !m, q_2)$ and $t' = (q'_1, ?m, q'_2)$, then $v^1_i = v_i - v_{q_1}$, $v^2_i = v^1_i - v_{q'_1}$, $v^3_i = v^2_i + v_{q_2}$, and $v_{i+1} = v^3_i + v_{q'_2}$. Then by induction hypothesis, $C(q_1) \geq 1$, $C(q'_1) \geq 1$, and $C(q_1) + C(q'_1) \geq 2$. We let $C' = C - \{\!\{q_1, q'_1\}\!\} + \{\!\{q_2, q'_2\}\!\}$. We have $C \xrightarrow{m}_{\mathcal{P}} C'$ and $C' \geq v_{i+1}$.
- If $(\ell_{in}, v_i) \rightsquigarrow (\ell_q, v^1_i) \rightsquigarrow (\ell_{in}, v_{i+1})$ with $(q, \tau, q') \in T$ and $v^1_i = v_i - v_q$ and $v_{i+1} = v^1_i + v_{q'}$, then by induction hypothesis, $C \geq 1$, and if we let $C' = C - \{\!\{q\}\!\} + \{\!\{q'\}\!\}$, then $C \xrightarrow{\tau}_{\mathcal{P}} C'$, and $C' \geq v_{i+1}$.
- If $(\ell_{in}, v_i) \rightsquigarrow (\ell_t, v^1_i) \rightsquigarrow (\ell^m_{t,p_1}, v^2_i) \rightsquigarrow \ldots \rightsquigarrow (\ell^m_{t,p_k}, v^{k+1}_i) \rightsquigarrow (\ell_{in}, v_{i+1})$ with $t = (q, !m, q')$ and $R(m) = \{p_1, \ldots, p_k\}$, and $(C - \{\!\{q\}\!\})(p) = 0$ for all $p \in R(m)$. We let $C' = C - \{\!\{q\}\!\} + \{\!\{q'\}\!\}$, hence $C \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C'$. Moreover, $v^1_i = v_i - v_q$, and, for all $1 \leq j < k$, it holds that $v^{j+1}_i(p_j) =$

$\max(0, v_i^j(p_j) - 1)$ and $v_i^{j+1}(p) = v_i^j(p)$ for all $p \neq p_j$. By induction hypothesis, $C \geq v_i$, hence $v_i^j(p) = 0$ for all $p \in R(m)$, for all $1 \leq j \leq k+1$. Hence, $v_{i+1} = v_i^{k+1} + v_{q'} = v_i^1 + v_{q'}$, and $C' \geq v_{i+1}$.

- If $(\ell_{in}, v_i) \rightsquigarrow (\ell_t, v_i^1) \rightsquigarrow (\ell^m_{t,p_1}, v_i^2) \rightsquigarrow \ldots \rightsquigarrow (\ell^m_{t,p_k}, v_i^{k+1}) \rightsquigarrow (\ell_{in}, v_{i+1})$ with $t = (q, !m, q')$ and $R(m) = \{p_1, \ldots, p_k\}$, and $(C - \Lbag q \Rbag)(p_j) > 0$ for some $p_j \in R(m)$. Let $(p_j, ?m, p'_j) \in T$ and $C' = C - \Lbag q, p_j \Rbag + \Lbag q', p'_j \Rbag$. Obviously, $C \xrightarrow{m}_{\mathcal{P}} C'$. It remains to show that $C' \geq v_{i+1}$. This is due to the fact that in the NB+R-CM $M$, the counter $p'_j$ will not be incremented, unlike $C(p'_j)$. Moreover, in the protocol $\mathcal{P}$, only $p_j$ will lose a process, whereas in $M$, other counters corresponding to processes in $R(m)$ may be decremented. Formally, by definition and by induction hypothesis, $C - \Lbag q \Rbag \geq v_i^1$. Also, for all $p \in R(m)$, either $v_i^1(p) = v_i^{k+1}(p) = 0$, or $v_i^{k+1}(p) = v_i^1(p) - 1$. Remark that since $C \geq v_i$, then $C - \Lbag q \Rbag \geq v_i - v_q = v_i^1$, hence $(C - \Lbag q, p_j \Rbag)(p_j) = (C - \Lbag q \Rbag)(p_j) - 1 \geq v_i^1(p_j) - 1$. Also, $(C - \Lbag q \Rbag)(p_j) - 1 \geq 0$, hence $(C - \Lbag q \Rbag)(p_j) - 1 \geq \max(0, v_i^1(p_j) - 1) = v_i^{k+1}(p_j)$. Observe also that, for all $p \neq p_j \in R(m)$, if $v_i^1(p) > 0$, then $(C - \Lbag q, p_j \Rbag)(p) = (C - \Lbag q \Rbag)(p) \geq v_i^1(p) > v_i^{k+1}(p)$. If $v_i^1(p) = 0$, then $(C - \Lbag q, p_j \Rbag)(p) \geq v_i^1(p) = v_i^{k+1}(p)$. For all other $p \in Q$, $(C - \Lbag q, p_j \Rbag)(p) = (C - \Lbag q \Rbag)(p) \geq v_i^1(p) = v_i^{k+1}(p)$. Hence, $C - \Lbag q, p_j \Rbag \geq v_i^{k+1}$. By definition, $v_{i+1} = v_i^{k+1} + v_{q'}$. Hence, $(C - \Lbag q, p_j \Rbag + \Lbag q', p'_j \Rbag)(p) \geq v_{i+1}(p)$, for all $p \neq p'_j$, and $(C - \Lbag q, p_j \Rbag + \Lbag q', p'_j \Rbag)(p'_j) > v_{i+1}(p'_j)$. So, $C' > v_{i+1}$.

Now we know that the initial execution of $M$ is : $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_{in}, v_n) \rightsquigarrow^* (\ell_f, v_f)$ with $v_f = v_n - (v_{\mathbf{q_1}} + v_{\mathbf{q_2}} + \cdots + v_{\mathbf{q}_s})$. Thus $v_n > v_{\mathbf{q_1}} + v_{\mathbf{q_2}} + \cdots + v_{\mathbf{q}_s}$. We have proved that we can build an initial execution of $P$: $C_0 \rightarrow^*_{\mathcal{P}} C_n$ and that $C_n \geq v_{\mathbf{q_1}} + v_{\mathbf{q_2}} + \cdots + v_{\mathbf{q}_s}$. Hence $C_n \geq C_F$. ◀

## B.2 Proofs of Theorem 4.2

To prove Theorem 4.2, we shall use Theorem 4.1 along with the reduction presented in Section 4.2. If the reduction is sound and complete, it will prove that SCover is Expspace-hard. As SCover is a particular instance of the CCover problem, this is sufficient to prove Theorem 4.2. The two lemmas of this subsection prove the soundness and completeness of the reduction presented in Section 4.2, put together with Theorem 3.5, it proves that SCover is Expspace-hard.

▶ **Lemma B.3.** *For all $v \in \mathbb{N}^d$, if $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^*_M (\ell_f, v)$, then there exists $C_0 \in \mathcal{I}$, $C_f \in \mathcal{F}_\exists$ such that $C_0 \rightarrow^*_{\mathcal{P}} C_f$.*

**Proof.** For all $\mathtt{x} \in X$, we let $N_\mathtt{x}$ be the maximal value taken by $\mathtt{x}$ in the initial execution $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^* (\ell_f, v)$, and $N = \Sigma_{\mathtt{x} \in X} N_\mathtt{x}$. Now, we let $C_0 \in \mathcal{I} \cap C_{N+1}$ be the initial configuration with $N+1$ processes. In the initial execution of $\mathcal{P}$ that we will build, one of the processes will evolve in the $\mathcal{P}(M)$ part of the protocol, simulating the execution of the NB+R-CM, the others will simulate the values of the counters in the execution.

Now, we show by induction on $k$ that, for all $k \geq 0$, if $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^k (\ell, w)$, then $C_0 \rightarrow^* C$, with $C(1_\mathtt{x}) = w(\mathtt{x})$ for all $\mathtt{x} \in X$, $C(\ell) = 1$, $C(q_{in}) = N - \Sigma_{\mathtt{x} \in X} w(\mathtt{x})$, and $C(s) = 0$ for all other $s \in Q$.

$C_0 \xrightarrow{\mathbf{nb}(L)} C_0^1 \xrightarrow{\mathbf{nb}(R)} C_0^2$, and $C_0^2(q_{in}) = N$, $C_0^2(\ell_{in}) = 1$, and $C_0^2(s) = 0$ for all other $s \in Q$. So the property holds for $k = 0$. Suppose now that the property holds for $k \geq 0$ and consider $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^k (\ell, w) \overset{\delta}{\rightsquigarrow} (\ell', w')$.

- if $\delta = (\ell, \mathtt{x}+, \ell')$, then $C \xrightarrow{\mathrm{inc}_\mathtt{x}}_{\mathcal{P}} C_1$ with $C_1 = C - \Lbag \ell, q_{in} \Rbag + \Lbag \ell_\delta, q_\mathtt{x} \Rbag$. Indeed, by induction hypothesis, $C(\ell) = 1 > 0$, and $C(q_{in}) > 0$, otherwise $\Sigma_{\mathtt{x} \in X} w(\mathtt{x}) = N$ and $w(\mathtt{x})$ is already the maximal value taken by $\mathtt{x}$ so no increment of $\mathtt{x}$ could have happened at that point of the execution of $M$. We also have $C_1 \xrightarrow{\overline{\mathrm{inc}_\mathtt{x}}}_{\mathcal{P}} C'$, since $C_1(\ell_\delta) > 0$ and $C_1(q_\mathtt{x}) > 0$ by construction, and $C' = C_1 - \{\!\!\{\ell_\delta, q_\mathtt{x}\}\!\!\} + \{\!\!\{\ell', 1_\mathtt{x}\}\!\!\}$. So $C'(\ell') = 1$, for all $\mathtt{x} \in X$, $C'(1_\mathtt{x}) = w'(\mathtt{x})$, and $C'(q_{in}) = N - \Sigma_{\mathtt{x}\in X} w'(\mathtt{x})$.

- if $\delta = (\ell, \mathtt{x}-, \ell')$, then $C(\ell) = 1 > 0$ and $C(1_\mathtt{x}) > 0$ since $w(\mathtt{x}) > 0$. Then $C \xrightarrow{\mathrm{dec}_\mathtt{x}}_{\mathcal{P}} C_1$ with $C_1 = C - \{\!\!\{\ell, 1_\mathtt{x}\}\!\!\} + \{\!\!\{\ell_\delta, q'_\mathtt{x}\}\!\!\}$. Then $C_1 \xrightarrow{\overline{\mathrm{dec}_\mathtt{x}}}_{\mathcal{P}} C'$, with $C' = C_1 - \{\!\!\{q'_\mathtt{x}, \ell_\delta\}\!\!\} + \{\!\!\{q_{in}, \ell'\}\!\!\}$. So $C'(\ell') = 1$, $C'(1_\mathtt{x}) = C(1_\mathtt{x}) - 1$, $C'(q_{in}) = C(q_{in}) + 1$.
- if $\delta = (\ell, \mathtt{nb}(\mathtt{x}-), \ell')$ and $w(\mathtt{x}) > 0$ then $C \xrightarrow{\mathrm{nbdec}_\mathtt{x}}_{\mathcal{P}} C'$, and $C' = C - \{\!\!\{\ell, 1_\mathtt{x}\}\!\!\} + \{\!\!\{\ell', q_{in}\}\!\!\}$ and the case is proved.
- if $\delta = (\ell, \mathtt{nb}(\mathtt{x}-), \ell')$ and $w(\mathtt{x}) = 0$ then by induction hypothesis, $C(1_\mathtt{x}) = 0$ and $C \xrightarrow{\mathbf{nb}(\mathrm{nbdec}_\mathtt{x})}_{\mathcal{P}} C'$, with $C' = C - \{\!\!\{\ell\}\!\!\} + \{\!\!\{\ell'\}\!\!\}$. Then, $C'(1_\mathtt{x}) = 0 = w'(\mathtt{x})$, and $C'(\ell') = 1$.
- if $\delta = (\ell, \bot, \ell')$, then $C \xrightarrow{\tau}_{\mathcal{P}} C'$, avec $C' = C - \{\!\!\{\ell\}\!\!\} + \{\!\!\{\ell'\}\!\!\}$. This includes the restore transitions.

Then $C_0 \rightarrow^* C$ with $C(\ell_f) = 1$ and $C \in \mathcal{F}_\exists$. ◀

▶ **Lemma B.4.** *Let $C_0 \in \mathcal{I}$, $C_f \in \mathcal{F}_\exists$ such that $C_0 \rightarrow^*_{\mathcal{P}} C_f$, then $(\ell_0, \mathbf{0}_X) \rightsquigarrow^*_M (\ell_f, v)$ for some $v \in \mathbb{N}^X$.*

Before proving this lemma we establish the following useful result.

▶ **Lemma B.5.** *Let $C_0 \in \mathcal{I}$. For all $C \in \mathcal{C}$ such that $C_0 \rightarrow^+_{\mathcal{P}} C$, we have $\Sigma_{p\in\{q\}\cup Q_M} C(p) = 1$.*

**Proof of Lemma B.4.** Note $C_0 \rightarrow C_1 \rightarrow \ldots \rightarrow C_n = C_f$. Now, thanks to Lemma B.5, for all $1 \le i \le n$, we can note $\mathsf{leader}(C_i)$ the unique state $s$ in $\{q\} \cup Q_M$ such that $C_i(s) = 1$. In particular, note that $\mathsf{leader}(C_n) = \ell_f$. We say that a configuration $C$ is $M$-compatible if $\mathsf{leader}(C) \in \mathrm{Loc}$. For any $M$-compatible configuration $C \in \mathcal{C}$, we define the configuration of the NB+R-CM $\pi(C_i) = (\mathsf{leader}(C), v)$ with $v = C(1_\mathtt{x})$ for all $\mathtt{x} \in X$.

We let $C_{i_1} \cdots C_{i_k}$ be the projection of $C_0 C_1 \ldots C_n$ onto the $M$-compatible configurations.

We show by induction on $j$ that :

$P(j)$: For all $1 \le j \le k$, $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^*_M \pi(C_{i_j})$, and $\Sigma_{\mathtt{x}\in X} C_{i_j}(q_\mathtt{x}) + C_{i_j}(q'_\mathtt{x}) = 0$. Moreover, for all $C$ such that $C_0 \rightarrow^*_{\mathcal{P}} C \rightarrow_{\mathcal{P}} C_{i_j}$, $\Sigma_{\mathtt{x}\in X} C(q_\mathtt{x}) + C(q'_\mathtt{x}) \le 1$.

By construction of the protocol, $C_0 \xrightarrow{\mathbf{nb}(L)} C_1 (\xrightarrow{L})^k C_2 \xrightarrow{\mathbf{nb}(R)} C_{i_1}$ for some $k \in \mathbb{N}$. So $\pi(C_{i_1}) = (\ell_{in}, \mathbf{0}_X)$, and for all $C$ such that $C_0 \rightarrow^*_{\mathcal{P}} C \rightarrow_{\mathcal{P}} C_{i_1}$, $\Sigma_{\mathtt{x}\in X} C(q_\mathtt{x}) + C(q'_\mathtt{x}) = 0$, so $P(0)$ holds true.

Let now $1 \le j < k$, and suppose that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow^*_M \pi(C_{i_j})$, and $\Sigma_{\mathtt{x}\in X} C_{i_j}(q_\mathtt{x}) + C_{i_j}(q'_\mathtt{x}) = 0$. We know that $C_{i_j} \rightarrow^+ C_{i_{j+1}}$.

- If there is no $C \in \mathcal{C}$ such that $C(q) = 1$ and $C_{i_j} \rightarrow^+ C \rightarrow^* C_{i_{j+1}}$, the only possible transitions from $C_{i_j}$ are in $T_M$. Let $\pi(C_{i_j}) = (\ell, v)$.
  - if $C_{i_j} \xrightarrow{\mathrm{inc}_\mathtt{x}} C$ then $C = C_{i_j} - \{\!\!\{\ell, q_{in}\}\!\!\} + \{\!\!\{\ell_\delta, q_\mathtt{x}\}\!\!\}$ for $\delta = (\ell, \mathtt{x}+, \ell') \in \Delta_b$. $\Sigma_{\mathtt{x}\in X} C(q_\mathtt{x}) + C(q'_\mathtt{x}) = 1$. Note that the message $\mathrm{inc}_\mathtt{x}$ is necessarily received by some process, otherwise $C(q_\mathtt{x}) = 0$ and $C$ has no successor, which is in contradiction with the fact the the execution reaches $C_f$. Moreover, the only possible successor configuration is $C \xrightarrow{\overline{\mathrm{inc}_\mathtt{x}}} C_{i_{j+1}}$, with $C_{i_{j+1}} = C - \{\!\!\{q_\mathtt{x}, \ell_\delta\}\!\!\} + \{\!\!\{1_\mathtt{x}, \ell'\}\!\!\}$. Hence, obviously, $\pi(C_{i_j}) \rightsquigarrow \pi(C_{i_{j+1}})$.
  - if $C_{i_j} \xrightarrow{\mathrm{dec}_\mathtt{x}} C$ then $C = C_{i_j} - \{\!\!\{\ell, 1_\mathtt{x}\}\!\!\} + \{\!\!\{\ell_\delta, q'_\mathtt{x}\}\!\!\}$ for $\delta = (\ell, \mathtt{x}-, \ell') \in \Delta_b$. $\Sigma_{\mathtt{x}\in X} C(q_\mathtt{x}) + C(q'_\mathtt{x}) = 1$. Note that the message $\mathrm{dec}_\mathtt{x}$ is necessarily received by some process, otherwise $C(q'_\mathtt{x}) = 0$ and $C$ has no successor, which is in contradiction with the fact the the execution reaches $C_f$. Besides, $C_{i_j}(1_\mathtt{x}) > 0$ hence $v(\mathtt{x}) > 0$. Moreover, the only possible

successor configuration is $C \xrightarrow{\overline{\mathrm{dec}_{\mathtt{x}}}} C_{i_{j+1}}$, with $C_{i_{j+1}} = C - \{\!\{q'_{\mathtt{x}}, \ell_\delta\}\!\} + \{\!\{q_{in}, \ell'\}\!\}$. Hence, obviously, $\pi(C_{i_j}) \rightsquigarrow \pi(C_{i_{j+1}})$.

- if $C_{i_j} \xrightarrow{\mathrm{nbdec}_{\mathtt{x}}} C_{i_{j+1}}$ then $C_{i_{j+1}} = C_{i_j} - \{\!\{\ell, 1_{\mathtt{x}}\}\!\} + \{\!\{\ell', q_{in}\}\!\}$ for $\delta = (\ell, \mathbf{nb}(\mathtt{x}-), \ell') \in \Delta_{nb}$. $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 0$. Besides, $C_{i_j}(1_{\mathtt{x}}) > 0$ hence $v(\mathtt{x}) > 0$. Hence, obviously, $\pi(C_{i_j}) \rightsquigarrow \pi(C_{i_{j+1}})$.
- if $C_{i_j} \xrightarrow{\mathbf{nb}(\mathrm{nbdec}_{\mathtt{x}})} C_{i_{j+1}}$ then $C_{i_{j+1}} = C_{i_j} - \{\!\{\ell\}\!\} + \{\!\{\ell'\}\!\}$ for $\delta = (\ell, \mathbf{nb}(\mathtt{x}-), \ell') \in \Delta_{nb}$. $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 0$. Besides, $C_{i_j}(1_{\mathtt{x}}) = 0$ hence $v(\mathtt{x}) = 0$. Hence, obviously, $\pi(C_{i_j}) \overset{\mathbf{nb}(\mathtt{x}-)}{\rightsquigarrow} \pi(C_{i_{j+1}})$.
- if $C_{i_j} \xrightarrow{\tau} C_{i_{j+1}}$ then $C_{i_{j+1}} = C_{i_j} - \{\!\{\ell\}\!\} + \{\!\{\ell'\}\!\}$ for $\delta = (\ell, \perp, \ell') \in \Delta_{nb}$. $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 0$. Besides, $C_{i_j}(1_{\mathtt{x}}) = C'_{i_{j+1}}(1_{\mathtt{x}})$ for all $\mathtt{x} \in X$. Hence, obviously, $\pi(C_{i_j}) \overset{\perp}{\rightsquigarrow} \pi(C_{i_{j+1}})$.

- Otherwise, let $C$ be the first configuration such that $C(q) = 1$ and $C_{i_j} \rightarrow^+ C \rightarrow^* C_{i_{j+1}}$. The transition leading to $C$ is necessarily a transition where the message $L$ has been sent. Remember also that by induction hypothesis, $\Sigma_{\mathtt{x}\in X} C_{i_j}(q_{\mathtt{x}}) + C_{i_j}(q'_{\mathtt{x}}) = 0$.
  - if $C_{i_j} \xrightarrow{L} C$, then $C(q) = 1$, and by induction hypothesis, $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 0$. Then the only possible successor configuration is $C \xrightarrow{\mathbf{nb}(R)} C_{i_{j+1}}$, with $\Sigma_{\mathtt{x}\in X} C_{i_{j+1}}(q_{\mathtt{x}}) + C_{i_{j+1}}(q'_{\mathtt{x}}) = 0$, and $\pi(C_{i_{j+1}}) = (\ell_{in}, v)$, so $\pi(C_{i_j}) \overset{\perp}{\rightsquigarrow} \pi(C_{i_{j+1}})$, by a restore transition.
  - if $C_{i_j} \xrightarrow{\mathrm{inc}_{\mathtt{x}}} C_1 \xrightarrow{L} C$ then $C_1 = C_{i_j} - \{\!\{\ell, q_{in}\}\!\} + \{\!\{\ell_\delta, q_{\mathtt{x}}\}\!\}$ for $\delta = (\ell, \mathtt{x}+, \ell') \in \Delta_b$ and $\Sigma_{\mathtt{x}\in X} C_1(q_{\mathtt{x}}) + C_1(q'_{\mathtt{x}}) = 1$. Now, $C = C_1 - \{\!\{\ell_\delta, q_{in}\}\!\} + \{\!\{q_\perp, q\}\!\}$, so $C(q) = 1 = C(q_{\mathtt{x}})$, and $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 1$.
    * If $C \xrightarrow{R} C_{i_{j+1}}$, then $C_{i_{j+1}} = C - \{\!\{q, q_{\mathtt{x}}\}\!\} + \{\!\{\ell_{in}, q_{in}\}\!\}$, then $\Sigma_{\mathtt{x}\in X} C_{i_{j+1}}(q_{\mathtt{x}}) + C_{i_{j+1}}(q'_{\mathtt{x}}) = 0$ and $\pi(C_{i_{j+1}}) = (\ell_{in}, v)$, hence $\pi(C_{i_j}) \overset{\perp}{\rightsquigarrow} \pi(C_{i_{j+1}})$ by a restore transition.
    * Now $C(q_{\mathtt{x}}) = 1$ so it might be that $C \xrightarrow{\mathbf{nb}(\overline{\mathrm{inc}_{\mathtt{x}}})} C'$, with $C' = C - \{\!\{q_{\mathtt{x}}\}\!\} + \{\!\{1_{\mathtt{x}}\}\!\}$. Here, $\Sigma_{\mathtt{x}\in X} C'(q_{\mathtt{x}}) + C'(q'_{\mathtt{x}}) = 0$. However, $\mathtt{leader}(C') = \{q\}$ so $C'$ is not $M$-compatible. The only possible transition from $C'$ is now $C' \xrightarrow{\mathbf{nb}(R)} C_{i_{j+1}}$ with $C_{i_{j+1}} = C' - \{\!\{q\}\!\} + \{\!\{\ell_{in}\}\!\}$. Hence, $C_{i_{j+1}}(1_{\mathtt{x}}) = C'(1_{\mathtt{x}}) = C_{i_j}(1_{\mathtt{x}}) + 1 = v(\mathtt{x}) + 1$, and $C_{i_{j+1}}(1_{\mathtt{y}}) = C'(1_{\mathtt{y}}) = C_{i_j}(1_{\mathtt{y}}) = v(\mathtt{y})$ for all $\mathtt{y} \neq \mathtt{x}$. So $\pi(C_{i_j}) = (\ell, v) \overset{\delta}{\rightsquigarrow} (\ell', v + v_{\mathtt{x}}) \overset{\perp}{\rightsquigarrow} (\ell_{in}, v + v_{\mathtt{x}}) = \pi(C_{i_{j+1}})$, the last step being a restore transition. Finally, $\Sigma_{\mathtt{x}\in X} C_{i_{j+1}}(q_{\mathtt{x}}) + C_{i_{j+1}}(q'_{\mathtt{x}}) = 0$.
  - if $C_{i_j} \xrightarrow{\mathrm{dec}_{\mathtt{x}}} C_1 \xrightarrow{L} C$, then $C_1 = C_{i_j} - \{\!\{\ell, 1_{\mathtt{x}}\}\!\} + \{\!\{\ell_\delta, q'_{\mathtt{x}}\}\!\}$ for $\delta = (\ell, \mathtt{x}-, \ell') \in \Delta_b$ and $\Sigma_{\mathtt{x}\in X} C_1(q_{\mathtt{x}}) + C_1(q'_{\mathtt{x}}) = 1$. Now, $C = C_1 - \{\!\{\ell_\delta, q_{in}\}\!\} + \{\!\{q_\perp, q\}\!\}$, so $C(q) = 1 = C(q'_{\mathtt{x}})$, and $\Sigma_{\mathtt{x}\in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 1$. Again, two transitions are available :
    * If $C \xrightarrow{R} C_{i_{j+1}}$, then $C_{i_{j+1}} = C - \{\!\{q, q'_{\mathtt{x}}\}\!\} + \{\!\{\ell_{in}, q_{in}\}\!\}$, then $\Sigma_{\mathtt{x}\in X} C_{i_{j+1}}(q_{\mathtt{x}}) + C_{i_{j+1}}(q'_{\mathtt{x}}) = 0$ and $\pi(C_{i_{j+1}}) = (\ell_{in}, v)$, hence $\pi(C_{i_j}) \overset{\perp}{\rightsquigarrow} \pi(C_{i_{j+1}})$ by a restore transition.
    * Now $C(q'_{\mathtt{x}}) = 1$ so it might be that $C \xrightarrow{\mathbf{nb}(\overline{\mathrm{dec}_{\mathtt{x}}})} C'$, with $C' = C - \{\!\{q'_{\mathtt{x}}\}\!\} + \{\!\{q_{in}\}\!\}$. Here, $\Sigma_{\mathtt{x}\in X} C'(q_{\mathtt{x}}) + C'(q'_{\mathtt{x}}) = 0$. However, $\mathtt{leader}(C') = \{q\}$ so $C'$ is not $M$-compatible. The only possible transition from $C'$ is now $C' \xrightarrow{\mathbf{nb}(R)} C_{i_{j+1}}$ with $C_{i_{j+1}} = C' - \{\!\{q\}\!\} + \{\!\{\ell_{in}\}\!\}$. Hence, $C_{i_{j+1}}(1_{\mathtt{x}}) = C'(1_{\mathtt{x}}) = C_{i_j}(1_{\mathtt{x}}) - 1 = v(\mathtt{x}) - 1$, and $C_{i_{j+1}}(1_{\mathtt{y}}) = C'(1_{\mathtt{y}}) = C_{i_j}(1_{\mathtt{y}}) = v(\mathtt{y})$ for all $\mathtt{y} \neq \mathtt{x}$. So $\pi(C_{i_j}) = (\ell, v) \overset{\delta}{\rightsquigarrow} (\ell', v - v_{\mathtt{x}}) \overset{\perp}{\rightsquigarrow} (\ell_{in}, v + v_{\mathtt{x}}) = \pi(C_{i_{j+1}})$, the last step being a restore transition. Finally, $\Sigma_{\mathtt{x}\in X} C_{i_{j+1}}(q_{\mathtt{x}}) + C_{i_{j+1}}(q'_{\mathtt{x}}) = 0$.
  - If $C_{i_j} \xrightarrow{\mathbf{nb}(\mathrm{inc}_{\mathtt{x}})} C_1$ then, it means that $C_{i_j}(q_{in}) = 0$. In that case, let $\delta = (\ell, \mathtt{x}+, \ell') \in \Delta_b$, and $C_1 = C_{i_j} - \{\!\{\ell\}\!\} + \{\!\{\ell_\delta\}\!\}$. Since, by induction hypothesis, $C_1(q_{\mathtt{x}}) = C_{i_j}(\mathtt{x}) = 0$, the only possible transition from $C_1$ would be $C_1 \xrightarrow{L} C_{i_{j+1}}$. However, $C_{i_j}(q_{in}) = C_1(q_{in}) = 0$, so

this transition is not possible, and $C_1$ is a deadlock configuration, a contradiction with the hypothesis that $C_{i_j} \to C_{i_{j+1}}$.

- If $C_{i_j} \xrightarrow{\mathbf{nb}(\mathrm{dec}_{\mathtt{x}})} C_1$ then it means that $C_{i_j}(1_{\mathtt{x}}) = 0$. In that case, let $\delta = (\ell, \mathtt{x}-, \ell') \in \Delta_b$, and $C_1 = C_{i_j} - \wr \ell \wr + \wr \ell_\delta \wr$. Since, by induction hypothesis, $\Sigma_{\mathtt{x} \in X} C_1(q_{\mathtt{x}}) + C_1(q'_{\mathtt{x}}) = \Sigma_{\mathtt{x} \in X} C_{i_j}(q_{\mathtt{x}}) + C_{i_j}(q'_{\mathtt{x}}) = 0$, the only possible transition from $C_1$ is $C_1 \xrightarrow{L} C$, with $C = C_1 - \wr q_{in}, \ell_\delta \wr + \wr q, q_\perp \wr$. Again, $\Sigma_{\mathtt{x} \in X} C(q_{\mathtt{x}}) + C(q'_{\mathtt{x}}) = 0$, and $C(\ell) =$ for all $\ell \in Q_M$, so the only possible transition is $C \xrightarrow{\mathbf{nb}(R)} C_{i_{j+1}}$. Observe that $C_{i_{j+1}}$ is $M$-compatible, with $C_{i_{j+1}}(\ell_{in}) = 1$, and $C_{i_{j+1}}(1_{\mathtt{x}}) = C_{i_j}(1_{\mathtt{x}})$ for all $\mathtt{x} \in X$. Hence $\pi(C_{i_{j+1}}) = (\ell_{in}, v)$, and $\pi(C_{i_j}) \overset{\perp}{\rightsquigarrow} \pi(C_{i_{j+1}})$, thanks to a restore transition of $M$.

We then have, by $P(k)$, that $(\ell_{in}, \mathbf{0}_X) \rightsquigarrow_M^* \pi(C_{i_k})$, with $C_{i_k}$ $M$-compatible and such that $C_{i_k} \to^* C_f$, and $C_{i_k}$ is the last $M$-compatible configuration. Then, by definition of an $M$-compatible configuration, $C_{i_k} = C_f$, and $\pi(C_{i_k}) = (\ell_f, v)$ for some $v \in \mathbb{N}^X$. ◀

# C Proof of Section 5

We present here omitted proofs of Section 5

## C.1 Technical Lemma

We provide here a lemma which will be useful in different parts of this section.

▶ **Lemma C.1.** *Let $\mathcal{P}$ be rendez-vous protocol and $C, C' \in \mathcal{C}$ such that $C = C_0 \to C_1 \cdots \to C_\ell = C'$. Then we have the two following properties.*

1. *For all $q \in Q$ verifying $C(q) = 2.\ell + a$ for some $a \in \mathbb{N}$, we have $C'(q) \geq a$.*
2. *For all $D_0 \in \mathcal{C}$ such that $D_0 \geq C_0$, there exist $D_1, \ldots, D_\ell$ such that $D_0 \to D_1 \cdots \to D_\ell$ and $D_i \geq C_i$ for all $1 \leq i \leq \ell$.*

**Proof.** According to the semantics associated to (non-blocking) rendez-vous protocols, each step in the execution from $C$ to $C'$ consumes at most two processes in each control state $q$, hence the result of the first item.

Let $C, C' \in \mathcal{C}$ such that $C \to C'$. Let $D \in \mathcal{C}$ such that $D \geq C$. We reason by a case analysis on the operation performed to move from $C$ to $C'$ and show that there exists $D'$ such that $D \to D'$ and $D' \geq C'$. (To obtain the final result, we repeat $k$ times this reasoning).

- Assume $C \xrightarrow{m}_{\mathcal{P}} C'$ then there exists $(q_1, !m, q'_1) \in T$ and $(q_2, ?m, q'_2) \in T$ such that $C(q_1) > 0$ and $C(q_2) > 0$ and $C(q_1) + C(q_2) \geq 2$ and $C' = C - \wr q_1, q_2 \wr + \wr q'_1, q'_2 \wr$. But since $D \geq C$, we have as well $D(q_1) > 0$ and $D(q_2) > 0$ and $D(q_1) + D(q_2) \geq 2$ and as a matter of fact $D \xrightarrow{m}_{\mathcal{P}} D'$ for $D' = D - \wr q_1, q_2 \wr + \wr q'_1, q'_2 \wr$. Since $D \geq C$, we have $D' \geq C'$.
- The case $C \xrightarrow{\tau}_{\mathcal{P}} C'$ can be treated in a similar way.
- Assume $C \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} C'$, then there exists $(q_1, !m, q'_1) \in T$, such that $C(q_1) > 0$ and $(C - \wr q_1 \wr)(q_2) = 0$ for all $(q_2, ?m, q'_2) \in T$ and $C' = C - \wr q_1 \wr + \wr q'_1 \wr$. We have as well that $D(q_1) > 0$. But we need to deal with two cases :
  1. If $(D - \wr q_1 \wr)(q_2) = 0$ for all $(q_2, ?m, q'_2) \in T$. In that case we have $D \xrightarrow{\mathbf{nb}(m)}_{\mathcal{P}} D'$ for $D' = D - \wr q_1 \wr + \wr q'_1 \wr$ and $D' \geq C'$.
  2. If there exists $(q_2, ?m, q'_2) \in T$ such that $(D - \wr q_1 \wr)(q_2) > 0$. Then we have that $D \xrightarrow{m}_{\mathcal{P}} D'$ for $D' = D - \wr q_1, q_2 \wr + \wr q'_1, q'_2 \wr$. Note that since $(C - \wr q_1 \wr)(q_2) = 0$ and $D \geq C$, we have here again $D' \geq C'$.

◀

## C.2 Properties of Consistent Abstract Sets of Configurations

### C.2.1 Proof of Lemma 5.1

**Proof.** Let $C' \in [\![\gamma]\!]$ such that $C' \geq C$. Let $q \in Q$ such that $C(q) > 0$. Then we have $C'(q) > 0$. If $q \notin S$, then $q \in \mathsf{st}(\mathit{Toks})$ and $C'(q) = 1$ and $C(q) = 1$ too. Furthermore for all $q' \in \mathsf{st}(\mathit{Toks}) \setminus \{q\}$ such $C(q') = 1$, we have that $C'(q') = 1$ and $q$ and $q'$ are conflict-free. This allows us to conclude that $C \in [\![\gamma]\!]$.

Checking whether $C$ belongs to $[\![\gamma]\!]$ can be done in polynomial time applying the definition of $[\![\cdot]\!]$. ◀

### C.2.2 Building Configurations from a Consistent Abstract Set

▶ **Lemma C.2.** *Let $\gamma$ be a consistent abstract set of configurations. Given a subset of states $U \subseteq Q$, if for all $N \in \mathbb{N}$ and for all $q \in U$ there exists $C_q \in [\![\gamma]\!]$ and $C'_q \in \mathcal{C}$ such that $C_q \rightarrow^* C'_q$ and $C'_q(q) \geq N$, then for all $N \in \mathbb{N}$, there exists $C \in [\![\gamma]\!]$ and $C' \in \mathcal{C}$ such that $C \rightarrow^* C'$ and $C'(q) \geq N$ for all $q \in U$.*

**Proof.** We suppose $\gamma = (S, \mathit{Toks})$ and reason by induction on the number of elements in $U \setminus S$. The base case is obvious. Indeed assume $U \setminus S = \emptyset$ and let $N \in \mathbb{N}$. We define the configuration $C$ such that $C(q) = N$ for all $q \in S$ and $C(q) = 0$ for all $q \in \mathsf{st}(\mathit{Toks})$. It is clear that $C \in [\![\gamma]\!]$ and that $C(q) \geq N$ for all $q \in U$ (since $U \setminus S = \emptyset$, we have in fact $U \subseteq S$).

We now assume that the property holds for a set $U$ and we shall see it holds for $U \cup \{p\}$, $p \notin S$. We assume hence that for all $N \in \mathbb{N}$ and for all $q \in U \cup \{p\}$ there exists $C_q \in [\![\gamma]\!]$ and $C'_q \in \mathcal{C}$ such that $C_q \rightarrow^* C'_q$ and $C'_q(q) \geq N$. Let $N \in \mathbb{N}$. By induction hypothesis, there exists $C_U \in [\![\gamma]\!]$ and $C'_U \in \mathcal{C}$ such that $C_U \rightarrow^* C'_U$ and $C'_U(q) \geq N$ for all $q \in U$. We denote by $\ell_U$ the number of steps in the execution from $C_U$ to $C'_U$. We will see that that we can build a configuration $C \in [\![\gamma]\!]$ such that $C \rightarrow^* C''_U$ with $C''_U \geq C_U$ and $C''_U(p) \geq N + 2 * \ell_U$. Using Lemma C.1, we will then have that $C''_U \rightarrow^* C'$ with $C' \geq C'_U$ and $C'(p) \geq N$. This will allow us to conclude.

We as well know that there exist $C_p \in [\![\gamma]\!]$ and $C'_p \in \mathcal{C}$ such that $C_p \rightarrow^* C'_p$ and $C'_p(p) \geq N + 2 * \ell_U + (k * \ell)$. We denote by $\ell_p$ the number of steps in the execution from $C_p$ to $C'_p$. We build the configuration $C$ as follows: we have $C(q) = C_U(q) + 2 * \ell_p + (k * \ell) + C_p(q)$ for all $q \in S$, and we have $C(q) = C_p(q)$ for all $q \in \mathsf{st}(\mathit{Toks})$. Note that since $C_p \in [\![\gamma]\!]$, we have that $C \in [\![\gamma]\!]$. Furthermore, we have $C \geq C_p$, hence using again Lemma C.1, we know that there exists a configuration $C''_p$ such that $C \rightarrow^* C''_p$ and $C''_p \geq C'_p$ (i.e. $C''_p(p) \geq N + 2 * \ell_U + (k * \ell)$ and $C''_p(q) \geq C_U(q) + (k * \ell) + C_p(q)$ for all $q \in S$ by Lemma C.1,Item 1)

Having $C_U \in [\![\gamma]\!]$, we name $(q_1, m_1) \dots (q_k, m_k)$ the tokens in $\mathit{Toks}$ such that $C_U(q_j) = 1$ for all $1 \leq j \leq k$, and for all $q \in \mathsf{st}(\mathit{Toks}) \setminus \{q_j\}_{1 \leq j \leq k}$, $C_U(q) = 0$. Since $\gamma$ is consistent, for each $(q_j, m_j)$ there exists a path $(q_{0,j}, !m_j, q_{1,j})(q_{1,j}, ?m_{1,j}, q_{2,j}) \dots (q_{\ell_j,j}, ?m_{\ell_j,j}, q_j)$ in $\mathcal{P}$ such that $q_{0,j} \in S$ and such that there exists $(q'_{i,j}, !m_{i,j}, q''_{i,j}) \in T$ with $q'_{i,j} \in S$ for all $1 \leq i \leq \ell_j$. We denote by $\ell = \max_{1 \leq j \leq k}(\ell_j) + 1$.

Assume there exists $1 \leq i \leq j \leq k$ such that $(q_i, m_i), (q_j, m_j) \in \mathit{Toks}$ and $C_U(q_i) = C_U(q_j) = 1$, and $m_i \in \mathrm{Rec}(q_j)$ and $m_j \in \mathrm{Rec}(q_i)$. Since $C_U$ respects $[\![\gamma]\!]$, $q_i$ and $q_j$ are conflict-free: there exist $(q_i, m), (q_j, m') \in \mathit{Toks}$ such that $m \notin \mathrm{Rec}(q_j)$ and $m' \notin \mathrm{Rec}(q_i)$. Hence, $(q_i, m_i), (q_i, m), (q_j, m_j), (q_j, m') \in \mathit{Toks}$, and $m \notin \mathrm{Rec}(q_j)$ and $m_j \in \mathrm{Rec}(q_i)$. Therefore, we have $(q_i, m), (q_j, m_j) \in \mathit{Toks}$ and $m \notin \mathrm{Rec}(q_j)$ and $m_j \in \mathrm{Rec}(q_i)$, which is in contradiction with the fact that $\gamma$ is consistent. Hence, for all $1 \leq i \leq j \leq k$, for all $(q_i, m_i), (q_j, m_j) \in \mathit{Toks}$, $m_i \notin \mathrm{Rec}(q_j)$ and $m_j \notin \mathrm{Rec}(q_i)$.

We shall now explain how from $C''_p$ we reach $C''_U$ in $k*\ell$ steps, i.e. how we put (at least) one token in each state $q_j$ such that $q_j \in \mathsf{st}(\mathit{Toks})$ and $C_U(q_j) = 1$ in order to obtain a configuration $C''_U \geq C_U$. We begin by $q_1$. Let a process on $q_{0,1}$ send the message $m_1$ (remember that $q_{0,1}$ belongs to $S$) and let $\ell_1$ other processes on states of $S$ send the messages needed for the process to reach $q_1$ following the path $(q_{0,1}, !m_1, q_{1,1})(q_{1,1}, ?m_{1,1}, q_{2,1})\dots(q_{\ell_1,1}, ?m_{\ell_1,1}, q_1)$. At this stage, we have that the number of processes in each state $q$ in $S$ is bigger than $C_U(q) + ((k-1)*\ell)$ and we have (at least) one process in $q_1$. We proceed similarly to put a process in $q_2$, note that the message $m_2$ sent at the beginning of the path cannot be received by the process in $q_1$ since, as explained above, $m_2 \notin \mathrm{Rec}(q_1)$.

We proceed again to put a process in the states $q_1$ to $q_K$ and at the end we obtain the configuration $C''_U$ with the desired properties. ◀

## C.3 Proof of Lemma 5.3

In this subsection, the different items of Lemma 5.3 have been separated in distinct lemmas.

▶ **Lemma C.3.** *$F(\gamma)$ is consistent and can be computed in polynomial time for all consistent $\gamma \in \Gamma$.*

**Proof.** The fact that $F(\gamma)$ can be computed in polynomial time is a direct consequence of the definition of $F$ (see Table 1).

Assume $\gamma = (S, \mathit{Toks}) \in \Gamma$ to be consistent. Note $(S'', \mathit{Toks}'')$ the intermediate sets computed during the computation of $F(\gamma)$, and note $F(\gamma) = (S', \mathit{Toks}')$.

To prove that $F(\gamma)$ is consistent, we need to argue that (1) for all $(q, m) \in \mathit{Toks}'' \setminus \mathit{Toks}$, there exists a finite sequence of transitions $(q_0, a_0, q_1)\dots(q_k, a_k, q)$ such that $q_0 \in S$, and $a_0 = !m$ and for all $1 \leq i \leq k$, we have that $a_i = ?m_i$ and that there exists $(q'_i, !m_i, q'_{i+1}) \in T$ with $q'_i \in S$, and (2) for all $(q, m), (q', m') \in \mathit{Toks}'$ either $m \in \mathrm{Rec}(q')$ and $m' \in \mathrm{Rec}(q)$ or $m \notin \mathrm{Rec}(q')$ and $m' \notin \mathrm{Rec}(q)$.

We start by proving property (1). If $(q, m)$ has been added to $\mathit{Toks}''$ with rule 3b, then by construction, there exists $p \in S$ such that $(p, !a, p') \in T$, and $(q, m) = (p', a)$. The sequence of transition is the single transition is $(p, !a, q)$.

If $(q, m)$ has been added to $\mathit{Toks}''$ with rule 5b, then there exists $(q', m) \in \mathit{Toks}$, and $(q', ?a, q)$ with $m \neq a$. Furthermore, $m \in \mathrm{Rec}(q)$ and there exists $(p, !a, p') \in T$ with $p \in S$. By hypothesis, $\gamma$ is consistent, hence there exists a finite sequence of transitions $(q_0, q_0, q_1)\dots(q_k, a_k, q')$ such that $q_0 \in S$, and $a_0 = !m$ and for all $1 \leq i \leq k$, we have that $a_i = ?m_i$ and that there exists $(q'_i, !m_i, q'_{i+1}) \in T$ with $q'_i \in S$. By completing this sequence with transition $(q', ?a, q)$ we get an appropriate finite sequence of transitions.

It remains to prove property (2). Assume there exists $(q, m), (q', m') \in \mathit{Toks}'$ such that $m \in \mathrm{Rec}(q')$ and $m' \notin \mathrm{Rec}(q)$, then as $\mathit{Toks}' \subseteq \mathit{Toks}''$, $(q, m), (q', m') \in \mathit{Toks}''$. By condition 6, $q \in S'$, therefore, as $\mathit{Toks}' = \{(p, a) \in \mathit{Toks}'' \mid p \notin S'\}$, we have that $(q, m) \notin \mathit{Toks}'$, and we reached a contradiction. ◀

▶ **Lemma C.4.** *If $(S', \mathit{Toks}') = F(S, \mathit{Toks})$ then $S \subsetneq S'$ or $\mathit{Toks} \subseteq \mathit{Toks}'$.*

**Proof.** From the construction of $F$ (see Table 1), we have $S \subseteq S'' \subseteq S'$.

Assume now that $S = S'$. First note that $\mathit{Toks} \subseteq \mathit{Toks}''$ (see Table 1) and that $\mathsf{st}(\mathit{Toks}) \cap S = \emptyset$. But $\mathit{Toks}' = \{(q, m) \in \mathit{Toks}'' \mid q \notin S'\} = \{(q, m) \in \mathit{Toks}'' \mid q \notin S\}$. Hence the elements that are removed from $\mathit{Toks}''$ to obtain $\mathit{Toks}'$ are not elements of $\mathit{Toks}$. Consequently $\mathit{Toks} \subseteq \mathit{Toks}'$. ◀

▶ **Lemma C.5.** *For all consistent $\gamma \in \Gamma$, if $C \in [\![\gamma]\!]$ and $C \to C'$ then $C' \in [\![F(\gamma)]\!]$.*

**Proof.** Let $\gamma = (S, \mathit{Toks}) \in \Gamma$ be a consistent abstract set of configurations, and $C \in \mathcal{C}$ such that $C \in [\![\gamma]\!]$ and $C \to C'$. Note $F(\gamma) = (S', \mathit{Toks}')$ and $\gamma' = (S'', \mathit{Toks}'')$ the intermediate sets used to compute $F(\gamma)$. We will first prove that for all state $q$ such that $C'(q) > 0$, $q \in S'$ or $q \in \mathsf{st}(\mathit{Toks}')$, and then we will prove that for all states $q$ such that $q \in \mathsf{st}(\mathit{Toks}')$ and $C'(q) > 0$, $C'(q) = 1$ and for all other state $p \in \mathsf{st}(\mathit{Toks}')$ such that $C'(p) > 0$, $p$ and $q$ are conflict-free.

Observe that $S \subseteq S'' \subseteq S'$, $\mathit{Toks} \subseteq \mathit{Toks}''$, and $\mathsf{st}(\mathit{Toks}'') \subseteq \mathsf{st}(\mathit{Toks}') \cup S'$.

First, let us prove that for every state $q$ such that $C'(q) > 0$, it holds that $q \in S' \cup \mathsf{st}(\mathit{Toks}')$. Note that for all $q$ such that $C(q) > 0$, because $C$ respects $\gamma$, $q \in \mathsf{st}(\mathit{Toks}) \cup S$. As $\mathsf{st}(\mathit{Toks}) \cup S \subseteq \mathsf{st}(\mathit{Toks}') \cup S'$, the property holds for $q$. Hence, we only need to consider states $q$ such that $C(q) = 0$ and $C'(q) > 0$. If $C \xrightarrow{\tau} C'$ then $q$ is such that there exists $(q', \tau, q) \in T$, $q'$ is therefore an active state and so $q' \in S$, (recall that $\mathit{Toks} \subseteq Q_W \times \Sigma$). Hence, $q$ should be added to $\mathsf{st}(\mathit{Toks}'') \cup S''$ by condition 2. As $\mathsf{st}(\mathit{Toks}'') \cup S'' \subseteq \mathsf{st}(\mathit{Toks}') \cup S'$, it concludes this case. If $C \xrightarrow{\mathbf{nb}(a)} C'$ then $q$ is such that there exists $(q', !a, q) \in T$, with $q'$ an active state. With the same argument, $q' \in S$ and so $q$ should be added to $\mathsf{st}(\mathit{Toks}'') \cup S''$ by condition 3a or 3b. If $C \xrightarrow{a} C'$, then $q$ is either a state such that $(q', !a, q) \in T$ and the argument is the same as in the previous case, or it is a state such that $(q', ?a, q) \in T$, and it should be added to $\mathsf{st}(\mathit{Toks}'') \cup S''$ by condition 4, 5a, or 5b. Therefore, we proved that for all state $q$ such that $C'(q) > 0$, it holds that $q \in \mathsf{st}(\mathit{Toks}') \cup S'$.

It remains to prove that if $q \in \mathsf{st}(\mathit{Toks})$, then $C'(q) = 1$ and for all $q' \in \mathsf{st}(\mathit{Toks}') \setminus \{q\}$ such that $C'(q') = 1$, we have that $q$ and $q'$ are conflict-free. Note that if $q \in \mathsf{st}(\mathit{Toks})$ and $C(q) = C'(q) = 1$, then for every state $p$ such that $p \in \mathsf{st}(\mathit{Toks})$ and $C(p) = C'(p) = 1$, it holds that $q$ and $p$ are conflict-free.

Observe that if $C \xrightarrow{\tau} C'$, then note $q$ the state such that $(q', \tau, q)$, it holds that $\{p \mid p \in \mathsf{st}(\mathit{Toks}') \text{ and } C'(p) > 0\} \subseteq \{p \mid p \in \mathsf{st}(\mathit{Toks}) \text{ and } C(p) = 1\}$: $q'$ is an active state, $q$ might be in $\mathsf{st}(\mathit{Toks})$ but it is added to $S'' \subseteq S'$ with rule 2, and for all other states, $C'(p) = C(p)$. If $p \in \mathsf{st}(\mathit{Toks}')$ and $C(p) > 0$, it implies that $C'(p) = C(p) = 1$ and $p \in \mathsf{st}(\mathit{Toks})$ (otherwise $p$ is in $S \subseteq S'$). Hence, there is nothing to do as $C$ respects $\gamma$.

Take now $q \in \mathsf{st}(\mathit{Toks}') \setminus \mathsf{st}(\mathit{Toks})$ with $C'(q) > 0$, we shall prove that $C'(q) = 1$ and for all $p \in \mathsf{st}(\mathit{Toks}')$ and $C'(p) > 0$, $q$ and $p$ are conflict-free. If $q \in \mathsf{st}(\mathit{Toks}') \setminus \mathsf{st}(\mathit{Toks})$, it implies that $C(q) = 0$ because $C$ respects $\gamma$. Hence: either (1) $C \xrightarrow{\mathbf{nb}(a)} C'$ with transition $(q', !a, q) \in T$, either (2) $C \xrightarrow{a} C'$ with transitions $(q_1, !a, q_1') \in T$ and $(q_2, ?a, q_2') \in T$ and $q = q_1'$ or $q = q_2'$. In the latter case, we should be careful as we need to prove that $q_2' \neq q_1'$, otherwise, $C'(q) = 2$.

**Case (1):** Note that as only one process moves between $C$ and $C'$ and $C(q) = 0$, it is trivial that $C'(q) = 1$. In this first case, as it is a non-blocking request on $a$ between $C$ and $C'$, it holds that: for all $p \in \mathsf{st}(\mathit{Toks})$ such that $C(p) = 1$, $a \notin \mathrm{Rec}(p)$. Take $p \in \mathsf{st}(\mathit{Toks}')$, such that $p \neq q$ and $C'(p) = 1$, then $C'(p) = C(p) = 1$ and so $p \in \mathsf{st}(\mathit{Toks})$, and $a \notin \mathrm{Rec}(p)$. Suppose $(p, m) \in \mathit{Toks}'$ such that $m \in \mathrm{Rec}(q)$, then we found two tokens in $\mathit{Toks}'$ such that $m \in \mathrm{Rec}(q)$ and $a \notin \mathrm{Rec}(p)$ which contradicts $F(\gamma)$'s consistency. Hence, $p$ and $q$ are conflict-free.

**Case (2):** Note that if $q_2' \in \mathsf{st}(\mathit{Toks}')$, then $q_2 \in \mathsf{st}(\mathit{Toks})$ (otherwise, $q_2'$ should be in $S'$ by condition 4), and note $(q_2, m) \in \mathit{Toks}$, with $(q_2', m) \in \mathit{Toks}'$. Note as well that if $q_1' \in \mathsf{st}(\mathit{Toks}')$, then $a \in \mathrm{Rec}(q_1')$ (otherwise, $q_1'$ should be in $S'$ by condition 3a) and $(q_1', a) \in \mathit{Toks}'$ by condition 3b. Furthermore, if $q_1' \in \mathsf{st}(\mathit{Toks}')$, $q_2 \in \mathsf{st}(\mathit{Toks})$ as well as otherwise $q_1'$ should be added to $S'$ by condition 3a.

We first prove that either $q_1' \in S'$, or $q_2' \in S'$. For the sake of contradiction, assume this is not the case, then there are three tokens $(q_1', a), (q_2, m), (q_2', m) \in \mathit{Toks}' \subseteq \mathit{Toks}''$, such that $(q_2, ?a, q_2') \in T$. From condition 7, $q_1'$ should be added to $S'$ and so $(q_1', a) \notin \mathit{Toks}'$. Note that, as a consequence $q_1' \neq q_2'$ or $q_1' = q_2' \in S'$. Take $q \in \mathsf{st}(\mathit{Toks}') \smallsetminus \mathsf{st}(\mathit{Toks})$ such that $C'(q) > 0$, if such a $q$ exists, then $q = q_1'$ or $q = q_2'$ and $q_1' \neq q_2'$. As a consequence, $C'(q) = 1$ (note that if $q_1' = q_2$, $C(q_2) = 1$).

Take $p \in \mathsf{st}(\mathit{Toks}') \smallsetminus \{q\}$ such that $C'(p) > 0$, it is left to prove that $q$ and $p$ are conflict-free. If $p \neq q$ and $p \in \mathsf{st}(\mathit{Toks}')$, then $C'(p) = C(p)$ (because $q_1' \in S'$ or $q_2' \in S'$). Hence, $p \in \mathsf{st}(\mathit{Toks})$ and $C'(p) = 1$.

Assume $q = q_1'$ and assume $q$ and $p$ are not conflict-free. Remember that we justified that $q_2 \in \mathsf{st}(\mathit{Toks})$, and therefore, $C(q_2) = 1$. Hence, either $C'(q_2) = 0$, or $q_2 = q_2'$ and in that case $q_2, q_2' \in S'$ or $q_2' = q_1'$ and then $q_2 = q$. In any cases, $p \neq q_2$. As $C$ respects $\gamma$, there exists $(p, m_p)$ and $(q_2, m) \in \mathit{Toks}$ such that $m_p \notin \mathrm{Rec}(q_2)$ and $m \notin \mathrm{Rec}(p)$ ($q_2$ and $p$ are conflict-free). As $p \in \mathsf{st}(\mathit{Toks}')$, $(p, m_p) \in \mathit{Toks}'$ and so $m_p \in \mathrm{Rec}(q)$ or $a \in \mathrm{Rec}(p)$ ($q$ and $p$ are not conflict-free). As $F(\gamma)$ is consistent, $m_p \in \mathrm{Rec}(q)$ and $a \in \mathrm{Rec}(p)$. Note that $a \neq m_p$ because $a \in \mathrm{Rec}(q_2)$, $a \neq m$ because $m \notin \mathrm{Rec}(p)$, and obviously $m \neq m_p$. Note also that if $m \notin \mathrm{Rec}(q)$, then we found two tokens $(q, a)$ and $(q_2, m)$ in $\mathit{Toks}'$ such that $a \in \mathrm{Rec}(q_2)$ and $m \notin \mathrm{Rec}(q)$, which contradicts the fact that $F(\gamma)$ is consistent (Lemma C.3). Hence, $m \in \mathrm{Rec}(q)$. Note that even if $q_2$ is added to $S''$, it still is in $\mathit{Toks}''$. As $\mathit{Toks}' \subseteq \mathit{Toks}''$ we found three tokens $(p, m_p), (q_2, m)$, $(q, a)$ in $\mathit{Toks}''$, satisfying condition 8, and so $p$ should be added to $S'$, which is absurd as $p \in \mathsf{st}(\mathit{Toks}')$. We reach a contradiction and so $q$ and $p$ should be conflict-free.

Finally assume $q = q_2'$. If $q = q_2$, then, because $C$ respects $\gamma$, $q$ and $p$ are conflict-free. Otherwise, as $q_2$ is conflict-free with $p$, there exists $(q_2, m)$ and $(p, m_p)$ in $\mathit{Toks}$ such that $m \notin \mathrm{Rec}(p)$ and $m_p \notin \mathrm{Rec}(q_2)$. Note that $(q, m) \in \mathit{Toks}''$ from condition 5b (otherwise, $q \in S''$ which is absurd). Hence, $(q, m) \in \mathit{Toks}'$ and, as $p \in \mathsf{st}(\mathit{Toks}')$, $(p, m_p)$ is conserved from $\mathit{Toks}$ to $\mathit{Toks}'$. It remains to show that $m_p \notin \mathrm{Rec}(q)$. Assume this is not the case, then there exists $(p, m_p)$ and $(q, m) \in \mathit{Toks}'$ such that $m \notin \mathrm{Rec}(p)$ and $m_p \in \mathrm{Rec}(q)$ which is absurd given $F(\gamma)$'s consistency. As a consequence, $q$ and $p$ are conflict-free.

We managed to prove that for all $q$ such that $C'(q) > 0$, $q \in S' \cup \mathsf{st}(\mathit{Toks}')$, and if $q \in \mathsf{st}(\mathit{Toks}')$, then $C'(q) = 1$ and for all others $p \in \mathsf{st}(\mathit{Toks}')$ such that $C'(p) = 1$, $p$ and $q$ are conflict-free.

◀

▶ **Lemma C.6.** *For all consistent $\gamma \in \Gamma$, if $C' \in [\![F(\gamma)]\!]$, then there exists $C'' \in \mathcal{C}$ and $C \in [\![\gamma]\!]$ such that $C'' \geq C'$ and $C \rightarrow^* C''$.*

**Proof.** Let $\gamma$ be a consistent abstract set of configurations and $C' \in [\![F(\gamma)]\!]$. We suppose that $\gamma = (S, \mathit{Toks})$ and $F(\gamma) = \gamma' = (S', \mathit{Toks}')$. We will first show that for all $N \in \mathbb{N}$, for all $q \in S'$ there exists a configuration $C_q \in [\![\gamma]\!]$ and a configuration $C_q' \in \mathcal{C}$ such that $C_q \rightarrow^* C_q'$ and $C_q'(q) \geq N$. This will allow us to rely then on Lemma C.2 to conclude.

Take $N \in \mathbb{N}$ and $q \in S'$, if $q \in S$, then take $C_q \in [\![\gamma]\!]$ to be $\wr N \cdot q \wr$. Clearly $C_q \in [\![F(\gamma)]\!]$, $C_q(q) \geq N$ and $C_q \rightarrow^* C_q$. Now let $q \in S' \smallsetminus S$. Note $(\mathit{Toks}'', S'')$ the intermediate sets of $F(\gamma)$'s computation.

**Case 1:** $q \in S''$. As a consequence $q$ was added to $S''$ either by one of the conditions 2, 3a, 4 or 5a. In cases 2 and 3a when $a \notin \mathrm{Rec}(q)$, note $q'$ the state such that $(q', \tau, q)$ or $(q', !a, q)$, and consider the configuration $C_q = \wr N \cdot q' \wr$. By doing $N$ internal transitions or non-blocking requests, we reach $C_q' = \wr N \cdot q \wr$. Note that the requests on $a$ are non-blocking as $q' \in Q_A$ and $a \notin \mathrm{Rec}(q)$. $C_q' \in [\![F(\gamma)]\!]$.

In cases 3a with $a \in \mathrm{Rec}(q)$ and in case 4, note $(q_1, !a, q_1')$ and $(q_2, ?a, q_2')$ the two transitions realizing the conditions. As a consequence $q_1, q_2 \in S$. Take the configuration

$C_q = \wr N \cdot q_1, N \cdot q_2 \wr$. $C_q \in [\![\gamma]\!]$ and by doing $N$ successive rendez-vous on letter $a$, we reach configuration $C'_q = \wr N \cdot q'_1 \wr + \wr N \cdot q'_2 \wr$. $C'_q \in [\![F(\gamma)]\!]$, and as $q \in \{q'_1, q'_2\}$, $C'_q(q) \geq N$.

In case 5a, there exists $(q', m) \in \mathit{Toks}$ such that $(q', ?a, q) \in T$, $m \notin \mathrm{Rec}(q)$, and there exists $p \in S$ such that $(p, !a, p') \in T$. Remember that $\gamma$ is consistent, and so there exists a finite sequence of transitions $(q_0, !m, q_1)(q_1, a_1, q_2) \ldots (q_k, a_k, q')$ such that $q_0 \in S$ and for all $1 \leq i \leq k$, $a_i = ?m_i$ and there exists $(q'_i, !m_i, q''_i) \in T$ with $q'_i \in S$. Take $C_q = \wr (N-1) \cdot q_0 \wr + \wr (N-1) \cdot q'_1 \wr + \cdots + \wr (N-1) \cdot q'_k \wr + \wr N \cdot p \wr + \wr q' \wr$. Clearly $C_q \in [\![\gamma]\!]$ as all states except $q'$ are in $S$ and $q' \in \mathsf{st}(\mathit{Toks})$, $C_q(q') = 1$. We shall show how to put 2 processes on $q$ from $C_q$ and then explain how to repeat the steps in order to put $N$. Consider the following execution: $C_q \xrightarrow{a} C_1 \xrightarrow{x_m} C_2 \xrightarrow{m_1} \ldots \xrightarrow{m_k} C_{k+2} \xrightarrow{a} C_{k+3}$. The first rendez-vous on $a$ is made with transitions $(p, !a, p')$ and $(q', ?a, q)$. Then either $m \notin \mathrm{Rec}(p')$ and $x_m = \mathbf{nb}(m)$, otherwise, $x_m = m$, in any cases, the rendez-vous or non-blocking sending is made with transition $(q_0, !m, q_1)$ and the message is not received by the process on $q$ (because $m \notin \mathrm{Rec}(q)$) and so $C_2 \geq \wr q \wr + \wr q_1 \wr$. Then, each rendez-vous on $m_i$ is made with transitions $(q'_i, !m_i, q''_i)$ and $(q_i, ?m_i, q_{i+1})$ $(q_{k+1} = q')$, . Hence $C_{k+3} \geq \wr (N-2) \cdot q_0 \wr + \wr (N-2) \cdot q'_1 \wr + \cdots + \wr (N-2) \cdot q'_k \wr + \wr (N-2) \cdot p \wr + \wr 2 \cdot q \wr$. We can reiterate this execution (without the first rendez-vous on $a$) $N-2$ times to reach a configuration $C'_q$ such that $C'_q \geq \wr N \cdot q \wr$.

**Case 2:** $q \notin S''$. Hence, $q$ should be added to $S'$ by one of the conditions 6, 7, and 8. If it was added with condition 6, let $(q_1, m_1), (q_2, m_2) \in \mathit{Toks}''$ such that $q = q_1$, $m_1 \neq m_2$, $m_2 \notin \mathrm{Rec}(q_1)$ and $m_1 \in \mathrm{Rec}(q_2)$. From the proof of Lemma C.3, one can actually observe that all tokens in $\mathit{Toks}''$ correspond to "feasible" paths regarding states in $S$, i.e there exists a finite sequence of transitions $(p_0, !m_1, p_1)(p_1, a_1, p_2) \ldots (p_k, a_k, q_1)$ such that $p_0 \in S$ and for all $1 \leq i \leq k$, $a_i = ?b_i$ and there exists $(p'_i, !b_i, p''_i) \in T$ with $p'_i \in S$. The same such sequence exists for the token $(q_2, m_2)$, we note the sequence $(s_0, !m_2, s_1) \ldots (s_\ell, a_\ell, q_2)$ such that $s_0 \in S$ and for all $1 \leq i \leq \ell$, $a_i = ?c_i$ and there exists $(s'_i, !c_i, s''_i) \in T$ with $s'_i \in S$. Take $C_q = \wr N \cdot p_0 \wr + \wr N \cdot s_0 \wr + \wr N p'_1 \wr + \cdots + \wr N p'_k \wr + \wr N \cdot s'_1 \wr + \cdots + \wr N \cdot s'_\ell \wr$. Clearly, $C_q \in [\![\gamma]\!]$, as all states are in $S$. Consider the following execution: $C_q \xrightarrow{\mathbf{nb}(m_1)} C_1 \xrightarrow{b_1} \ldots \xrightarrow{b_k} C_{k+1}$, the non-blocking sending of $m_1$ is made with transition $(p_0, !m_1, p_1)$ and each rendez-vous on letter $b_i$ is made with transitions $(p'_i, !b_i, p''_i)$ and $(p_i, ?b_i, p_{i+1})$ $(p_{k+1} = q_1)$. Hence, $C_{k+1}$ is such that $C_{k+1} \geq \wr q_1 \wr$. From $C_{k+1}$, consider the following execution: $C_{k+1} \xrightarrow{x_{m_2}} C_{k+2} \xrightarrow{c_1} \ldots \xrightarrow{c_\ell} C_{k+\ell+2} \xrightarrow{m_1} C_{k+\ell+3}$, where $x_{m_2} = \mathbf{nb}(m_2)$ if no process is on a state in $R(m_2)$, or $x_{m_2} = m_2$ otherwise. In any case, as $m_2 \notin \mathrm{Rec}(q_1)$, $C_{k+2} \geq \wr q_1 \wr$. And each rendez-vous on letter $c_i$ is made with transitions $(s'_i, !c_i, s''_i)$ and $(s_i, ?c_i, s_{i+1})$ $(s_{k+1} = q_2)$, the last rendez-vous on $m_1$ is made with transitions $(p_0, !m_1, p_1)$ and $(q_2, ?m_1, q'_2)$ (such a $q'_2$ exists as $m_1 \in \mathrm{Rec}(q_2)$). Hence, $C_{k+\ell+3} \geq \wr p_1 \wr + \wr q_1 \wr$. By repeating the two sequences of steps (without the first non blocking sending of $m_1$) $N-1$ times (except for the last time where we don't need to repeat the second execution), we reach a configuration $C'_q$ such that $C'_q \geq \wr N \cdot q_1 \wr$.

If it was added with condition 7, then let $(q_1, m_1), (q_2, m_2), (q_3, m_2) \in \mathit{Toks}''$ such that $m_1 \neq m_2$ and $(q_2, ?m_1, q_3) \in T$ with $q = q_1$. From the proof of Lemma C.3, $\mathit{Toks}''$ is made of "feasible" paths regarding $S$ and so there exists a finite sequence of transitions $(p_0, !m_2, p_1)(p_1, a_1, p_2) \ldots (p_k, a_k, q_2)$ such that $p_0 \in S$ and for all $1 \leq i \leq k$, $a_i = ?b_i$ and there exists $(p'_i, !b_i, p''_i) \in T$ with $p'_i \in S$. The same such sequence exists for the token $(q_1, m_1)$, we note the sequence $(s_0, !m_1, s_1) \ldots (s_\ell, a_\ell, q_1)$ such that $s_0 \in S$ and for all $1 \leq i \leq \ell$, $a_i = ?c_i$ and there exists $(s'_i, !c_i, s''_i) \in T$ with $s'_i \in S$. Take $C_q = \wr N \cdot p_0 \wr + \wr N \cdot s_0 \wr + \wr N p'_1 \wr + \cdots + \wr N p'_k \wr + \wr N \cdot s'_1 \wr + \cdots + \wr N \cdot s'_\ell \wr$. Clearly, $C_q \in [\![\gamma]\!]$, as all states are in $S$. We do the same execution from $C_q$

to $C_{k+1}$ as in the previous case: $C_q \xrightarrow{\mathbf{nb}(m_2)} C_1 \xrightarrow{a_1} \dots \xrightarrow{a_k} C_{k+1}$. Here $C_{k+1}$ is then such that $C_{k+1} \geq \Lbag q_2 \Rbag$. Then, from $C_{k+1}$ we do the following: $C_{k+1} \xrightarrow{m_1} C_{k+2} \xrightarrow{c_1} \dots \xrightarrow{c_\ell} C_{k+\ell+2} \xrightarrow{m_2} C_{k+\ell+3}$: the rendez-vous on letter $m_1$ is made with transitons $(s_0, !m_1, s_1)$ and $(q_2, ?m_1, q_3)$. Then, each rendez-vous on letter $c_i$ is made with transitions $(s'_i, !c_i, s''_i)$ and $(s_i, ?c_i, s_{i+1})$ ($s_{k+1} = q_1$), and the last rendez-vous on letter $m_2$ is made with transitions $(p_0, !m_2, p_1)$ and $(q_3, ?m_2, q'_3)$ (such a state $q'_3$ exists as $(q_3, m_2) \in \mathit{Toks}''$ and so $m_2 \in \mathrm{Rec}(q_3)$). Hence, $C_{k+\ell+3}$ is such that $C_{k+\ell+3} \geq \Lbag q_1 \Rbag + \Lbag p_1 \Rbag$. We can repeat the steps from $C_1$ $N-1$ times (except for the last time where we don't need to repeat the second execution), to reach a configuration $C'_q$ such that $C'_q \geq \Lbag N \cdot q_1 \Rbag$.

If it was added with condition 8, then let $(q_1, m_1), (q_2, m_2), (q_3, m_3) \in \mathit{Toks}''$, such that $m_1 \neq m_2$, $m_2 \neq m_3$, $m_1 \neq m_3$, and $m_1 \notin \mathrm{Rec}(q_2)$, $m_1 \in \mathrm{Rec}(q_3)$, and $m_2 \notin \mathrm{Rec}(q_1)$, $m_2 \in \mathrm{Rec}(q_3)$ and $m_3 \in \mathrm{Rec}(q_2)$ and $m_3 \in \mathrm{Rec}(q_1)$, and $q_1 = q$. Then there exists three finite sequences of transitions $(p_0, !m_1, p_1)(p_1, ?b_1, p_2) \dots (p_k, ?b_k, p_{k+1})$, and $(s_0, !m_2, s_1)(s_1, ?c_1, s_2) \dots (s_\ell, ?c_k, s_{\ell+1})$, and $(r_0, !m_3, r_1)(r_1, ?d_1, r_2) \dots (r_j, ?d_j, r_{j+1})$ such that $p_{k+1} = q_1$, $s_{\ell+1} = q_2$ and $r_{j+1} = q_3$, and for all messages $a \in \{b_{i_1}, c_{i_2}, d_{i_3}\}_{1 \leq i_1 \leq k, 1 \leq i_2 \leq \ell, 1 \leq i_3 \leq j} = M$, there exists $q_a \in S$ such that $(q_a, !a, q'_a)$. Take $C_q = \Lbag N p_0 \Rbag + \Lbag N s_0 \Rbag + \Lbag N r_0 \Rbag + \sum_{a \in M} \Lbag N q_a \Rbag$. From $C_q$ there exists the following execution: $C_q \xrightarrow{\mathbf{nb}(m_1)} C_1 \xrightarrow{b_1} \dots \xrightarrow{b_k} C_{k+1}$ where the non-blocking sending is made with the transition $(p_0, !m_1, p_1)$ and each rendez-vous with letter $b_i$ is made with transitions $(q_{b_i}, !b_i, q'_{b_i})$ and $(p_i, ?b_i, p_{i+1})$. Hence, $C_{k+1} \geq \Lbag q_1 \Rbag$. Then, we continue the execution in the following way: $C_{k+1} \xrightarrow{x_{m_2}} C_{k+2} \xrightarrow{c_1} \dots \xrightarrow{c_\ell} C_{k+\ell+2}$ where $x_{m_2} = \mathbf{nb}(m_2)$ if there is no process on $R(m_2)$, and $x_{m_2} = m_2$ otherwise. In any case, the rendez-vous is not answered by a process on state $q_1$ because $m_2 \notin \mathrm{Rec}(q_1)$. Furthermore, each rendez-vous with letter $c_i$ is made with transitions $(q_{c_i}, !c_i, q'_{c_i})$ and $(s_i, ?c_i, s_{i+1})$. Hence, $C_{k+\ell+2} \geq \Lbag q_2 \Rbag + \Lbag q_1 \Rbag$. From $C_{k+\ell+2}$ we do the following execution: $C_{k+\ell+2} \xrightarrow{m_3} C_{k+\ell+3} \xrightarrow{d_1} \dots \xrightarrow{d_j} C_{k+\ell+j+3}$ where the rendez-vous on letter $m_3$ is made with transitions $(r_0, !m_3, r_1)$ and $(q_2, ?m_3, q'_2)$ (this transition exists as $m_3 \in \mathrm{Rec}(q_2)$). Each rendez-vous on $d_i$ is made with transitions $(q_{d_i}, !d_i, q'_{d_i})$ and $(r_i, ?d_i, r_{i+1})$. Hence, the configuration $C_{k+\ell+j+3}$ is such that $C_{k+\ell+j+3} \geq \Lbag q_3 \Rbag + \Lbag q_1 \Rbag$. Then from $C_{k+\ell+j+3}$: $C_{k+\ell+j+3} \xrightarrow{m_1} C_{k+\ell+j+4}$ where the rendez-vous is made with transitions $(p_0, !m_1, p_1)$ and $(q_3, ?m_1, q'_3)$ (this transition exists as $m_1 \in \mathrm{Rec}(q_3)$). By repeating $N-1$ times the execution from configuration $C_1$, we reach a configuration $C'_q$ such that $C'_q(q_1) \geq N$.

Hence, for all $N \in \mathbb{N}$, for all $q \in S'$, there exists $C_q \in \llbracket \gamma \rrbracket$, such that $C_q \to C'_q$ and $C'_q(q) \geq N$. From Lemma C.2, there exists $C'_N$ and $C_N \in \llbracket \gamma \rrbracket$ such that $C_N \to^* C'_N$ and for all $q \in S'$, $C_N(q) \geq N$.

Take $C' \in \llbracket F(\gamma) \rrbracket$, we know how to build for any $N \in \mathbb{N}$, a configuration $C'_N$ such that $C'_N(q) \geq N$ for all states $q \in S'$ and there exists $C_N \in \llbracket \gamma \rrbracket$, such that $C_N \to^* C'_N$, in particular for $N$ bigger than the maximal value $C'(q)$ for $q \in S'$, $C'_N$ is greater than $C'_N$ on all the states in $S'$.

To conclude the proof, we need to prove that from a configuration $C'_{N'}$ for a particular $N'$, we can reach a configuration $C''$ such that $C''(q) \geq C'(q)$ for $q \in S' \cup \mathsf{st}(\mathit{Toks}')$. As $C'$ respects $F(\gamma)$, remember that for all $q \in \mathsf{st}(\mathit{Toks}')$, $C'(q) = 1$. The execution is actually built in the manner of the end of the proof of Lemma C.2.

Note $N_{\max}$ the maximum value for any $C'(q)$. We enumerate states $q_1, \dots, q_m$ in $\mathsf{st}(\mathit{Toks}')$ such that $C'(q_i) = 1$. As $C'$ respects $F(\gamma)$, for $i \neq j$, $q_i$ and $q_j$ are conflict free.

From Lemma C.3, $F(\gamma)$ is consistent, and so we note $(p^j_0, !m^j, p^j_1)$ $(p^j_1, ?m^j_1, p^j_2)$ $\dots$ $(p^j_{k_j}, ?m^j_{k_j}, p^j_{k_j+1})$ the sequence of transitions associated to state $q_j$ such that: $p^j_{k_j+1} = q_j$, $(q_j, m^j) \in \mathit{Toks}$ and for all $m^j_i$, there exists $(q_{m^j_i}, !m^j_i, q'_{m^j_i})$ with $q_{m^j_i} \in S'$. Note that for all $i \neq j$, $q_i$ and $q_j$ are conflict-free and so there exists $(q_i, m), (q_j, m') \in \mathit{Toks}'$ such that $m \notin \mathrm{Rec}(q_j)$ and $m' \notin \mathrm{Rec}(q_i)$. As $F(\gamma)$ is consistent, it should be the case for all pairs of tokens $(q_i, a), (q_j, a')$. Hence $m^j \notin \mathrm{Rec}(q_i)$ and $m^i \notin \mathrm{Rec}(q_j)$.

Note $\ell_j = k_j + 1$. For $N' = N_{\max} + \sum_{1 \leq j \leq m} \ell_j$, there exists a configuration $C'_{N'}$ such that there exists $C_{N'} \in [\![\gamma]\!]$, $C_{N'} \to^* C'_{N'}$, and $C'_{N'}(q) \geq N'$ for all $q \in S'$. In particular, for all $q \in S'$, $C'_{N'}(q) \geq C'(q) + \sum_{1 \leq j \leq m} \ell_j$.

Then, we still have to build an execution leading to a configuration $C''$ such that for all $q \in \mathsf{st}(\mathit{Toks}')$, $C''(q) \geq C'(q)$. We then use the defined sequences of transitions for each state $q_j$. With $\ell_1$ processes we can reach a configuration $C_1$ such that $C_1(q_1) \geq 1$: $C_1 \xrightarrow{x_{m^1}} C_2 \xrightarrow{m^1_1} \dots \xrightarrow{m^1_{k_1}} C_{\ell_1+1}$. $x_{m^1} = \mathbf{nb}(m^1)$ if there is no process on $R(m^1)$, and $x_{m^1} = m^1$ otherwise. Each rendez-vous on $m^1_i$ is made with transitions $(p^1_i, ?m^1_i, p^1_{i+1})$ and $(q_{m^1_i}, !m^1_i, q'm^1_i)$. As a result, for all $q \in S'$, $C_{\ell_1+1}(q) \geq C'(q) + \sum_{2 \leq j \leq m} \ell_j$ and $C_{\ell_1+1}(q_1) \geq 1$. We then do the following execution form $C_{\ell_1+1}$: $C_{\ell_1+1} \xrightarrow{x_{m^2}} C_{\ell_1+2} \xrightarrow{m^2_1} \dots \xrightarrow{m^2_{k_2}} C_{\ell_1+\ell_2+2}$. $x_{m^2} = \mathbf{nb}(m^2)$ if there is no process on $R(m^2)$, and $x_{m^2} = m^2$ otherwise. Remember that we argued that $m^2 \notin \mathrm{Rec}(q_1)$, and therefore $C_{\ell_1+2}(q_1) \geq C_{\ell_1+1}(q_1) \geq 1$. Each rendez-vous on $m^2_i$ is made with transitions $(p^2_i, ?m^2_i, p^2_{i+1})$ and $(q_{m^2_i}, !m^2_i, q'm^2_i)$. As a result, $C_{\ell_1+\ell_2+2}(q) \geq C'(q) + \sum_{3 \leq j \leq m} \ell_j$ for all $q \in S'$ and $C_{\ell_1+\ell_2+2} \geq \wr q_1 \int + \wr q_2 \int$. We can then repeat the reasoning for each state $q_i$ and so reach a configuration $C''$ such that $C''(q) \geq C'(q)$ for all $q \in S'$ and, $C'' \geq \wr q_1 \int + \wr q_2 \int + \dots \wr q_m \int$. We built the following execution: $C_{N'} \to^* C'_{N'} \to^* C''$, such that $C'' \geq C'$, and $C'_{N'} \in [\![\gamma]\!]$.

◀

## C.4 Proof of Lemma 5.4

**Proof.** Assume that there exists $C_0 \in \mathcal{I}$ and $C' \geq C$ such that $C_0 \to C_1 \to \dots \to C_\ell = C'$. Then using iteratively Lemma C.5, we get that $C' \in [\![\gamma_\ell]\!]$. From the definition of $F$ and $[\![\cdot]\!]$, one can furthermore easily check that $[\![\gamma]\!] \subseteq [\![F(\gamma)]\!]$ for all $\gamma \in \Gamma$. Hence we have $[\![\gamma_\ell]\!] \subseteq [\![\gamma_f]\!]$ and $C' \in [\![\gamma_f]\!]$.

Before proving the other direction, we first prove by induction that for all $i \in \mathbb{N}$ and for all $D \in [\![\gamma_i]\!]$, there exists $C_0 \in \mathcal{I}$ and $D' \geq D$ such that $C_0 \to^* D'$. The base case for $i = 0$ is obvious. Assume the property holds for $\gamma_i$ and let us show it is true for $\gamma_{i+1}$. Let $E \in [\![\gamma_{i+1}]\!]$. Since $\gamma_{i+1} = F(\gamma_i)$, using Lemma C.6, we get that there exists $E' \in \mathcal{C}$ and $D \in [\![\gamma_i]\!]$ such that $E' \geq E$ and $D \to^* E'$. By induction hypothesis, there exists $C_0 \in \mathcal{I}$ and $D' \geq D$ such that $C_0 \to^* D'$. Using the monotonicity property stated in Lemma C.1, we deduce that there exists $E'' \in \mathcal{C}$ such that $E'' \geq E' \geq E$ and $C_0 \to^* D' \to^* E''$.

Suppose now that there exists $C'' \in [\![\gamma_f]\!]$ such that $C'' \geq C$. By the previous reasoning, we get that there exists $C_0 \in \mathcal{I}$ and $C' \geq C'' \geq C$ such that $C_0 \to^* C'$. ◀