Technische Universität München
TUM School of Computation, Information and Technology

Observation Petri Nets

Chana Yvonne Marie Weil-Kennedy

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Helmut Seidl

Prüfer der Dissertation:

1. Prof. Dr. Francisco Javier Esparza Estaun
2. Prof. Serge Haddad

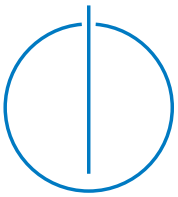
Die Dissertation wurde am 15.11.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 23.02.2023 angenommen.

Department of Informatics
Technical University of Munich

Dissertation in Informatics

Observation Petri Nets

Chana Weil-Kennedy



Acknowledgments

First and foremost, I would like to thank my advisor Javier Esparza. He is an excellent researcher, professor and communicator, and a prime illustration of leading by example. He is also very generous with his time, and precise in his always-constructive criticism – I feel lucky to have had him as an advisor.

I want to thank my co-authors Mikhail and Bala. Mikhail was an invaluable help at the beginning of my doctorate, always patient with me as I worked through his ideas and slowly added my own. I will not forget his kind words when I was feeling down. It was a pleasure to collaborate with Bala on later papers, and it hardly felt like work because of his calm confidence and contagious humor. Thanks to Christoph, who I started with and who remained my office mate in spirit throughout. And thanks to the other people of the chair who made my life here nicer, whether through shared tutorials, bouldering, coffee breaks or Wordle support – special thanks to Pascal, Marijana and Onkel Luu. I feel fortunate to have met a kind and welcoming academic community, from the very beginning. Thanks to Pierre Ganty for providing a wonderful internship opportunity, at IMDEA Software Institute in Madrid, which led directly to this thesis. Thanks to Miguel for his friendship as I took my first steps in academic life, and thanks to everyone else with whom I've had the pleasure of crossing paths during my internship, at conferences and at summer schools.

As always, I am grateful to my friends who keep me sane, in Munich and abroad. Loving thoughts go to Julie and Kyvèli, and a heartfelt thank you to Lucas. Finally, I am grateful to my parents and brother who mean the world to me.

On a related but less emotional note, I am thankful for the financial support I received during this doctorate from the European Research Council project PaVeS (No. 787367).

Abstract

Petri nets are a classic formal model for concurrent systems, extensively studied and used since the 1970s. They model fundamental features of concurrent systems such as synchronization, process creation, and choice. Many subclasses of Petri nets have been defined by forbidding one or more of these features (e.g. T-systems, free-choice nets, communication-free nets). In this thesis, we introduce a feature of Petri nets called observation, which is a restricted form of synchronization. We develop the theory of observation Petri nets, which we define to be the class of nets in which the only allowed form of synchronization is observation. We study the complexity of analysis problems for this class, and in particular we consider verification problems parameterized by the number of processes (or tokens) in the Petri net. Our class is further divided into two objects of study: immediate observation nets, in which process creation is also prohibited, and branching immediate observation nets, in which process creation is allowed. We also consider an application of our theory to population protocols, a prominent model in distributed computing, and study some extensions to our two main formalisms. Our complexity results support the relevance of studying the observation feature.

Contents

1 Introduction

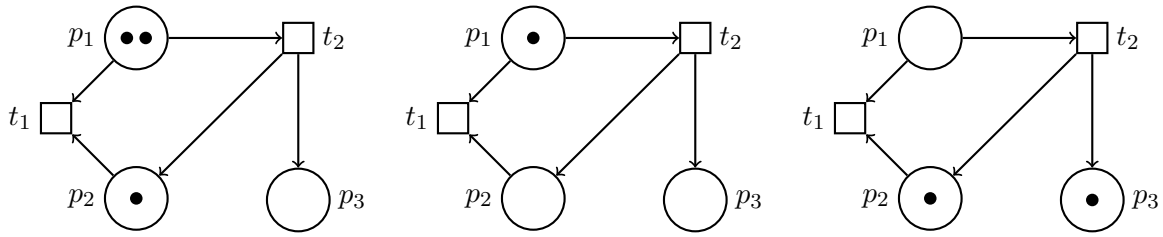
Concurrent systems are systems in which several entities interact. An entity can perform actions independently from the other entities, or it can interact with them and perform actions together. These entities can be agents, processes or resources, and the actions and interactions are specified by some common protocol. Some examples are wireless mobile networks, industrial control systems, or information processing in biological systems. The theoretical study of concurrent systems is used to develop distributed (or parallel) computing techniques, but also to analyze such systems and verify their correctness. Possible correctness problems are whether resources are distributed in such a way that every agent can perform their task, or whether a safety-critical system never enters a faulty state. In this thesis, we focus on the analysis and verification of concurrent systems as modelled by Petri nets.

Petri nets are a classic formal model for concurrent systems. They have been extensively used for modelling, design and formal analysis since their introduction in the 1960s by Carl Adam Petri (see [73] for a good introduction). They have applications in many domains, like communication protocols, business process modelling or biology (see e.g. surveys [15, 22, 32, 70, 87]). Petri nets model fundamental features of concurrent systems such as synchronization, concurrency, choice, and process creation or destruction.

1.1 Petri Nets

A Petri net is made up of *places* (graphically represented by circles), of *transitions* (graphically represented by squares) and of arcs between places and transitions. *Tokens* (graphically represented by black dots) can be in a place of the net. A *marking* of a given Petri net is a distribution of tokens on the places of the net. A transition is *enabled* at a marking if there is at least one token in each place that has an arc to this transition. An enabled transition can *fire*, resulting in a new marking which has one less token in each input place of the transition (places with an arc to the transition), and one more token in each output place of the transition (places with an arc from the transition). Intuitively, the places, transitions and arcs model the static structure of a concurrent system, while the tokens and the firing rule model the dynamic behaviour of a concurrent system.

Consider the Petri net of Figure 1.1, which has three places p_1, p_2, p_3 and three transitions t_1, t_2, t_3. Transitions t_1 and t_2 are enabled at the marking represented in Figure 1.1a. Firing t_1 leads to the marking represented in Figure 1.1b. Transition t_2 is enabled at that marking, and firing it leads to the marking represented in Figure 1.1c.



(a) *Initial marking.* **(b)** *After firing* t_1. **(c)** *After firing* t_2.

Figure 1.1: *A Petri net.*

There are different ways of seeing what tokens, places and transitions represent. In one of the earlier models of Petri nets, called C/E nets [78] or elementary nets [79, 80], places could hold either one token or no token. In this model, transitions were events, their input places were pre-conditions, and their output places were post-conditions. The presence of a token in a place indicated whether the associated condition held. In the model we consider, which used to be called P/T nets [35] and that is now called Petri nets, places may contain more than one token. Interpretations suited to this model include seeing transitions as tasks which require resources to be executed and which re-allocate or produce new resources upon completion. Places are then resource types, and a token is one unit of a resource. Another interpretation is that transitions are computation steps, input places are input data and output places are output data. In this thesis, we follow the mental image of tokens as identical agents. A token in a place is seen as an agent being in a certain state, and transitions model agents interacting, resulting in a change of states. This agent view makes sense for example in the distributed-processing setting: processes (or programs or threads) are in different states, like waiting, or reading from some shared memory, and they may communicate with each other to change states or spawn new processes according to some shared protocol.

Petri nets model fundamental features of the behaviour of concurrent systems (see e.g. [80] for early formalization of features, in particular choice and concurrency). We illustrate some of these features on the Petri net of Figure 1.1. Transition t_1 *synchronizes* the actions of a token in p_1 and a token in p_2: there must be a token in both places at the same time for t_1 to fire. Consider the marking which places one token in p_1 and one token in p_2. At this marking, the token in p_1 faces a *choice*. It can either contribute to firing transition t_1 or contribute to firing transition t_2. This feature is also called *conflict* in the literature, since transitions t_1 and t_2 are in conflict at the marking: both are enabled but only one may fire. On the other hand, at our first example marking in Figure 1.1a, there are enough tokens for both t_1 and t_2 to fire without hindering each other, they can occur *concurrently*. Transition t_1 models *process destruction*, and transition t_2 models *process creation*: firing t_1 from a given marking yields a marking with two less tokens, while firing t_2 adds an extra

token. In the agent view of tokens, an agent in p_1 and an agent in p_2 are destroyed by t_1, while t_2 can be seen as an agent in p_1 moving to p_2 and creating a new agent in p_3.

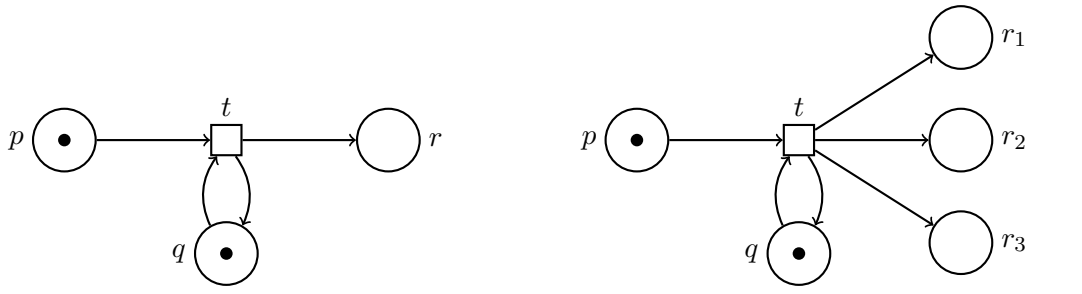
A key feature of Petri net theory is that syntactic restrictions on the structure of a Petri net can guarantee behavioural properties of the Petri net. T-nets (or T-graphs, or marked graphs [28, 51]) are Petri nets in which places have exactly one input transition and one output transition. No marking of a T-net can exhibit a situation of choice since a token in a place only enables one transition. S-nets (or S-graphs, or state machines [35]) are Petri nets in which transitions have exactly one input place and one output place. In these nets, there is no synchronization of the actions of different tokens since transitions are enabled by just one token. Additionally, there is no process creation or destruction since each transition "consumes" one token and "produces" one token. Basic Parallel Processes (or communication-free nets [27, 39, 72]) are Petri nets in which every transition has a unique input place. In these nets, the absence of synchronization is still guaranteed and process creation or destruction is allowed. Communication-free nets, the other name of BPP nets, emphasizes the agent view of tokens: the agents cannot communicate with each other through shared transitions.

A main motivation for studying subclasses of Petri nets, like those above, is the intractability of analysis in the full model. Petri nets are very expressive but the trade-off is that the analysis of Petri nets is of high complexity. In particular, the problem of reachability was recently shown to be ACKERMANN-complete [31, 66, 67], a complexity class with non-primitive recursive runtime. The reachability problem is the central problem for the analysis of Petri nets. It asks, given an initial marking and a target marking, whether the initial marking can reach the target marking by firing a sequence of transitions in a given Petri net. The application of reachability in Petri net analysis is most direct for safety problems: we check whether some bad marking, or set of bad markings, is *not* reachable. It is also useful for liveness problems where we are interested in the repeated reachability of some good markings. Additionally, the reachability problem is important because of its inter-reducibility with many problems in diverse domains of theoretical computer science. Some examples are the non-emptiness problem for data automata [20], the validity in the !-*Horn* fragment of linear logic [58], or the correctness problem for population protocols [42] (see Section 5 of [82] for more examples). A relaxation of reachability, called coverability, is also widely used and studied. Coverability asks, given an initial marking and a target marking, whether the initial marking can reach a marking with at least as many tokens on each place as the target marking. The complexity of the reachability problem is ACKERMANN-complete, as mentioned above, while that of coverability is EXPSPACE-complete [69, 74]. For T-nets and S-nets, it is well-known that the reachability problem is P-complete (straightforward consequence of [28, 51]). For BPP nets, the reachability problem is NP-complete [39].

In this thesis, we introduce a new feature called observation. We syntactically define a class of Petri nets whose behaviour enforces this feature, and we study this class of nets, called observation Petri nets.

1.2 Observation Petri Nets

We introduce a new feature called *observation*. It is a restricted form of synchronization in which, informally, a token moves from one place to another after observing the presence of tokens in other places; these observed tokens do not move. For example, consider the Petri net in Figure 1.2a. Firing transition t can be seen as the token in p observing that there is a token in q and moving to r. A transition is an *observation transition* if it has at most one input place which is not also an output place. This input place is called the *source* place. Places which are both input and output places of the transition are called *observed* places. Places which are just output places of the transition are called *destination* places. The transitions in Figures 1.2a and 1.2b are observation transitions. In Figure 1.2a, p is the source place of t, q is an observed place of t and r is a destination place of t. In Figure 1.2b, p is the source place, q is an observed place and r_1, r_2, r_3 are destination places of t. It is clear that observation transitions model the observation feature. They also model synchronization, but in a restricted form: an observation transition synchronizes the actions of the token in the source place and the tokens in the observed places (they must all be there for the transition to be enabled), but only the token in the source place can change its place. This passiveness of the tokens in the observed places means that if there are several tokens in the source place of some transition, it can be fired again and again without affecting the tokens in the observed places. This intuition will be important for understanding the properties of observation Petri nets.



(a) *An immediate observation net.* **(b)** *A branching immediate observation net.*

Figure 1.2: *Observation Petri nets.*

We study *observation Petri nets*, a class syntactically defined as the Petri nets in which every transition is an observation transition. This restriction on their structure ensures that the only allowed form of synchronization is observation. We look at two classes of

observation Petri nets. We start by studying immediate observation nets, in which process creation and destruction are prohibited. Then we study branching immediate observation nets, in which process creation and destruction are allowed.

An *immediate observation* (IO) net is a Petri net such that every transition is an observation transition with one source place, one destination place and at most one observed place. These places do not necessarily need to be distinct. Figure 1.2a is an example of an IO net with one transition. The results we will show for IO nets still hold if there is more than one observed place. For simplicity, we only consider this case in the last chapter of the thesis. IO nets restrict all synchronization to observation, and do not allow process creation or destruction. They are *conservative*: firing a transition preserves the number of tokens in the net.

For conservative Petri nets, the reachability problem and the coverability problem are PSPACE-complete [37]. We show that this is still the case for IO nets, i.e. this restriction does not improve on the complexity of its more general class, at least in the marking to marking case. We turn our attention to *generalized reachability problems*, a generalization of reachability queries to possibly infinite sets of markings. These problems are expressed using boolean combinations (union, intersection, negation), forward reachability ($post^*$), backward reachability (pre^*) and a class of sets of markings called *cubes*. In particular, one can express the *cube-reachability* problem, which asks whether there exists a marking in a given cube which can reach some marking in another given cube. For these problems, IO nets have much better complexity than conservative Petri nets: we show that all such problems can be solved in polynomial space. In comparison, we show that cube-reachability for conservative Petri nets is as hard as reachability in general Petri nets, i.e. ACKERMANN-hard [31, 66]. In essence, this result for generalized reachability problems expresses that in IO nets, it is not harder to verify reachability problems for single markings than it is to verify them for (infinite) sets of markings. This allows us to consider parameterized verification problems with ease – problems in which the numbers of tokens in given places are parameters, which we express using our cubes.

Our second type of observation Petri net is *branching immediate observation* (BIO) nets. This class of nets restricts all synchronization to observation *and* does not forbid process creation or destruction. BIO nets are the Petri nets such that each transition is an observation transition with one source place, multiple destination places and at most one observed place. Figure 1.2b is an example of a BIO net. Like for IO nets, the results we will show for BIO nets still hold if there are several observed places, and they also hold if we allow transitions to have no source place. This most general case corresponds to the full class of observation Petri nets. It is treated in the last chapter for simplicity. Notice that BIO nets are a generalization of both IO nets and BPP nets (the nets in which each transition has a unique input place).

We show that for BIO nets, the classic analysis problems of reachability and coverability are much easier than for general Petri nets (we do not compare with conservative nets since BIO nets are not conservative). The problems are in PSPACE, as opposed to ACKERMANN and EXPSPACE respectively in the general case. Moreover, we examine generalized reachability problems for this BIO class as well. Many of the problems remain solvable in PSPACE, allowing us to conclude like for the IO class: the complexity of the reachability problem does not change whether we check it on single markings or on sets of markings. This is even more remarkable for BIO nets, since it is a much more expressive class than IO nets.

These complexity results encourage the study of observation Petri nets, and in this thesis we establish a picture of the properties and complexity results for our two observation classes.

A motivation for studying the observation feature is its relevance in the field of distributed computing. In 2004, Angluin *et al.* [6] introduced population protocols, a formalism for the study of ad hoc networks of tiny computing devices without any infrastructure. Since then they have been intensely studied, in particular in recent years (for example [1–3, 30, 36, 40]). In [7], Angluin *et al.* define subclasses of population protocols corresponding to different communication primitives between devices, one of which is immediate observation protocols. In these, the communication is only one-way; an example is a network of passively mobile sensors. IO nets were introduced to model these immediate observation protocols (thus the name), and we apply our results to verifying the correctness of such protocols.

The observation feature also appears in the field of chemical reaction networks [9, 48], where Petri net modelling is sometimes used [5, 59, 60, 86]. Chemical reaction networks are a discrete model of chemistry in which molecules change their state due to collisions. IO nets model networks of enzymatic chemical reactions, of the form $A + C \to C + B$, in which an enzyme C catalyzes the formation of product B from substrate A [12, 71]. An example of application of Petri net techniques to such a network is presented in [4].[1] We expect that BIO nets can find a similar application, as they are a natural model for enzymatic catalytic reactions of the form $A + C \to C + B_1 + \cdots + B_n$ with more than one product. For example, catalase degrades hydrogen peroxide into water and oxygen, a reaction of the form $A + C \to C + B_1 + B_2$ [23].

Additionally, BIO nets (actually, reverse BIO nets) appear in the field of type-directed program synthesis [49, 53]. In [49], the authors use Petri nets to automatically generate program sketches, which they then complete using SAT solvers. The Petri nets represent relationships between API components: the places are types, the transitions are methods and the tokens represent the number of program variables of a certain type. The transitions either "output" a single type depending on some multiset of "input" types, or are cloning

[1] The Petri nets of [4] are in fact slightly more general than IO nets, but equivalent to them for properties that depend only on the reachability graph, as are the net properties studied in [4].

transitions, which upon observing a token produce a new one so as to reuse program variables.

1.3 Outline and Publications

We first give the outline of the thesis, then list our publications.

Outline. Chapter 2 introduces mathematical notations, recalls the basics of Petri nets and defines cubes and generalized reachability problems. The main content is then structured into four chapters as follows.

1. Chapter 3 introduces our first main object of study: *immediate observation (IO) nets*. After definitions, examples and basic properties, we turn our attention to studying reachability problems, first simple then parameterized by possibly infinite sets of markings called cubes. We first give a complexity lower bound for these problems by simulating a bounded-tape Turing machine using IO nets. To give complexity upper bounds, we set up a framework where tokens are "de-anonymized", i.e. given identities. Reasoning in this framework, we prove properties on firing sequences and reachability sets of IO nets. We derive results for IO nets, in particular that a host of cube-parameterized problems, called *generalized reachability problems*, can be solved in polynomial space. This is an interesting property because the complexity is not higher than in the non-parameterized single marking setting. This contrasts with the situation for conservative Petri nets, of which IO nets are a subclass: in the single marking setting reachability problems are solvable in polynomial space, but in the parameterized setting we show they are as hard as the reachability problem for general Petri nets. Other properties are shown, for example that the reachability set of a cube is itself a finite union of cubes.

2. Chapter 4 applies the results of the previous chapter to *immediate observation population protocols*. This model was introduced in [7]. Population protocols are an extensively studied [3, 36, 40] model of distributed computation in which anonymous finite-state agents interact together to change their states, following a common protocol. In a *correct* population protocol, the agents compute a predicate: the input is the initial configuration of the agents' states, and the agents interact in pairs to eventually reach a consensus which provides the output. Petri nets can and have been used to model population protocols [42]. Checking correctness is the main verification question for population protocols. In [42], it is shown that this problem is decidable but at least as hard as Petri net reachability. This means there is a non-primitive recursive lower bound [31]. We use IO nets to study IO population

protocols (and this is where their name comes from). For IO population protocols, we show that the correctness problem is significantly easier than in the general case: it is PSPACE-complete.

3. Chapter 5 introduces our second main object of study: *branching immediate observation (BIO) nets*. BIO nets are a natural generalization of IO nets and BPP nets, also called communication-free nets. BPP nets are a well studied subclass of Petri nets (e.g. [27, 39, 72]) which models branching processes, but where processes cannot synchronize with each other. In a BIO net, a token (or process) observes another token and then branches into a certain number of tokens in different places. We study the complexity for BIO nets of the same generalized reachability problems as in Chapter 3 for IO nets. We again set up a framework for this study by giving tokens identities. This time the trajectories are more complex than for IO nets. In particular, the reachability set from a cube is no longer guaranteed to be "simple" (i.e., here, semilinear). However, we show that many of the same complexity bounds hold. In particular, reachability, even parameterized by cubes, remains a PSPACE-complete problem.

4. Chapter 6 contains additional material connected to IO and BIO nets. First, we show that the results continue to hold when we allow multiple observations as conditions for a transition to fire, instead of just one. For BIO nets, we also show that having no source place does not change the results. Next, we look at reconfigurable broadcast networks (RBN) [33, 34], an existing model for networks of identical finite-state agents which communicate by broadcasts. One agent broadcasts and changes its state, and the neighboring agents receive this broadcast and change their state. In RBN, the neighborhood topology can reconfigure during an execution. We show that IO nets can be seen as a subclass of RBN in which the broadcaster does not change states. We then investigate a new model at the intersection of RBN and BIO nets, called branching reconfigurable broadcast networks (BRBN). We give some complexity results for this model. Finally, we take a look at temporal logic for IO and BIO nets, and state some model checking results.

We describe which publications contributed to each chapter. The publications are listed below for the reader's convenience. Most of the results in Chapter 3 are published in [47] and [77]. The results of Chapter 4 come from [47]. Some of the results of Chapter 5 are published in [77], and many are not yet published. Finally, Chapter 6 contains mostly unpublished results, except for those on RBN which were published in [11]. More details about which results come from which papers can be found in the discussion sections at the end of each chapter.

Publications. We list our papers published during the doctorate in chronological order.

Verification of Immediate Observation Population Protocols. J. Esparza, P. Ganty, R. Majumdar, C. Weil-Kennedy. In conference proceedings of CONCUR 2018. [44]

Parameterized Analysis of Immediate Observation Petri Nets. J. Esparza, M. Raskin, C. Weil-Kennedy. In conference proceedings of Petri Nets 2019. [47]

The Complexity of Verifying Population Protocols. J. Esparza, S. Jaax, M. Raskin, C. Weil-Kennedy. In Distributed Computing journal. [46]

Flatness and Complexity of Immediate Observation Petri Nets. J. Esparza, M. Raskin, C. Weil-Kennedy. In conference proceedings of CONCUR 2020. [77]

Efficient Restrictions of Immediate Observation Petri Nets. M. Raskin, C. Weil-Kennedy. In conference proceedings of Reachability Problems 2020. [75]

Reconfigurable Broadcast Networks and Asynchronous Shared-Memory Systems are Equivalent. A. R. Balasubramanian, C. Weil-Kennedy. In conference proceedings of GandALF 2021. [11]

Parameterized Analysis of Reconfigurable Broadcast Networks. A. R. Balasubramanian, L. Guillou, C. Weil-Kennedy. In conference proceedings of FoSSaCS 2022. [10]

As mentioned above, [47], [77], [46] and [11] contribute to the results of this thesis. The results of [44] are not included as they are subsumed by the results of [47]. However, [44] was the first step in the direction of studying observation Petri nets and generalized reachability problems, and it introduced cubes for a model close to IO nets. Article [75] contains additional results on IO nets which have not been included in the thesis. Article [10] focuses on the Reconfigurable Broadcast Network (RBN) formalism. Though we mention some links between RBN and IO nets in this thesis, the results of this article are out of scope and not included.

2 Preliminaries

In this chapter, we introduce some definitions and notations which will be useful in the rest of the thesis.

2.1 Multisets

A *multiset* on a finite set E is a mapping $C\colon E \to \mathbb{N}$, i.e. for any $e \in E$, $C(e)$ denotes the number of occurrences of element e in C. We let $\mathbb{M}(E)$ denote the set of all multisets on E. Let $\wr e_1, \ldots, e_n \wr$ denote the multiset C such that $C(e) = |\{j \mid e_j = e\}|$. We sometimes write multisets using set-like notation. For example, $\wr 2 \cdot a, b \wr$ and $\wr a, a, b \wr$ denote the same multiset. Given $e \in E$, we denote by $\boldsymbol{e}$ the multiset consisting of one occurrence of element e, that is $\wr e \wr$. Operations on $\mathbb{N}$ like addition or comparison are extended to multisets by defining them component-wise on each element of E. Subtraction is allowed in the following way: if C, D are multisets on set E then for all $e \in E$, $(C - D)(e) = \max(C(e) - D(e), 0)$. We call $|C| \stackrel{\text{def}}{=} \sum_{e \in E} C(e)$ the *size* of C.

2.2 Petri Nets

We recall the model of Petri nets and give notation. In the introduction we presented Petri nets without weights on the arcs for simplicity, but in the rest we use Petri nets with weighted arcs. A *Petri net* $\mathcal{N}$ is a triple (P, T, F) consisting of a finite set of *places* P, a finite set of *transitions* T and a *flow function* $F\colon (P \times T) \cup (T \times P) \to \mathbb{N}$. A *marking* M is a multiset on P, and we say that a marking M puts $M(p)$ *tokens* in place p of P. The *size* of M, denoted by $|M|$, is the total number of tokens in M. The *preset* ${}^\bullet t$ and *postset* $t^\bullet$ of a transition t are the multisets on P given by ${}^\bullet t(p) = F(p, t)$ and $t^\bullet(p) = F(t, p)$. A transition t is *enabled* at a marking M if ${}^\bullet t \leq M$, i.e. ${}^\bullet t$ is component-wise smaller or equal to M. If t is enabled then it can be *fired*, leading to a new marking $M' = M - {}^\bullet t + t^\bullet$. We let $M \xrightarrow{t} M'$ denote this. Given $\sigma = t_1 \ldots t_n$ we write $M \xrightarrow{\sigma} M_n$ when $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \ldots \xrightarrow{t_n} M_n$, and call σ a *firing sequence*.

Example 1. Figure 2.1 represents a Petri net with three places p_1, p_2, p_3 and three transitions t_1, t_2, t_3. Places are represented by circles, transitions by squares, and tokens by black dots. The arcs between places and transitions represent the flow function. No arc means that the flow function has value 0. If there is an arc and it has no number written on it, then the flow function's value on that arc is 1 (we say the arc has weight 1). Otherwise, it is the number written on the arc: the arc from t_1 to p_3 has weight 2, i.e. $F(t_1, p_3) = 2$. The initial marking M_0 puts one token in place p_1, one in p_2 and zero in

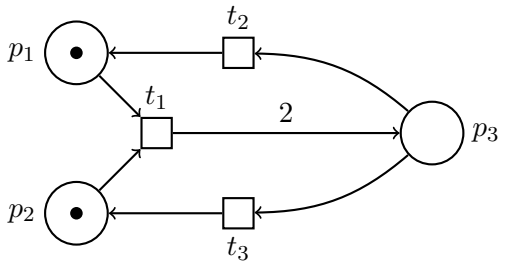


Figure 2.1: *A Petri net with initial marking* $M_0 = (1, 1, 0)$.

p_3. We can write M_0 either as $(1, 1, 0)$ or as $\wr p_1, p_2 \wr$. A firing sequence from M_0 to M_0 is $M_0 \xrightarrow{t_1} (0, 0, 2) \xrightarrow{t_2} (1, 0, 1) \xrightarrow{t_3} M_0$.

Reachability, coverability and liveness. We write $M' \xrightarrow{*} M''$ if $M' \xrightarrow{\sigma} M''$ for some $\sigma \in T^*$, and say that M'' is *reachable* from M'. A marking M *covers* another marking M', written $M \geq M'$ if $M(p) \geq M'(p)$ for all places p. A marking M is *coverable* from M' if there exists a marking M'' such that $M' \xrightarrow{*} M'' \geq M$. A marking M is *live* if for all markings M' such that $M \xrightarrow{*} M'$, for all transitions $t \in T$, there exists a marking M'' such that $M' \xrightarrow{*} M''$ and t is enabled at M''. Given a Petri net $\mathcal{N} = (P, T, F)$ and two markings M, M', the *reachability problem* consists of deciding whether M' is reachable from M. The *coverability problem* consists of deciding whether M' is coverable from M. The *liveness problem* consists of deciding whether M is live.

Reachability sets. Let $\mathcal{S}$ be a set of markings of $\mathcal{N}$. The *backward reachability set* of $\mathcal{S}$ is $pre^*(\mathcal{S}) \stackrel{\text{def}}{=} \{M' | \exists M \in \mathcal{S} \,.\, M' \xrightarrow{*} M\}$, and the *forward reachability set* of $\mathcal{S}$ is $post^*(\mathcal{S}) \stackrel{\text{def}}{=} \{M | \exists M' \in \mathcal{S} \,.\, M' \xrightarrow{*} M\}$. The *one-step backward reachability set* of $\mathcal{S}$ is $pre(\mathcal{S}) \stackrel{\text{def}}{=} \{M' | \exists M \in \mathcal{S} \,.\, M' \to M\}$, and the *one-step forward reachability set* of $\mathcal{S}$ is $post(\mathcal{S}) \stackrel{\text{def}}{=} \{M | \exists M' \in \mathcal{S} \,.\, M' \to M\}$.

Conservative nets. A Petri net $\mathcal{N} = (P, T, F)$ is *conservative*[1] if for every transition the sum of the weights of its input arcs is equal to the sum of the weights of its output arcs, i.e. if $\sum_{p \in P} |{}^\bullet t(p)| = \sum_{p \in P} |t^\bullet(p)|$ for all $t \in T$. It follows immediately from the definitions that if $\mathcal{N}$ is conservative and $M \xrightarrow{*} M'$, then $\sum_{p \in P} M(p) = \sum_{p \in P} M'(p)$.

Monotonicity. Petri nets have the following *monotonicity* property: given markings M, M', L of a Petri net $\mathcal{N}$, if $M \xrightarrow{*} M'$, then $M + L \xrightarrow{*} M' + L$. The proof follows directly from the fact that tokens in a Petri net cannot disable the occurrence of a transition.

[1] These nets are sometimes called **1**-conservative in the literature, e.g. in [57].

2.3 Cubes and Counting Sets

Given a finite set P, a *cube* $\mathcal{C}$ is a subset of $\mathbb{M}(P)$ described by a lower bound $L\colon P \to \mathbb{N}$ and an upper bound $U\colon P \to \mathbb{N} \cup \{\infty\}$ such that $\mathcal{C} = \{C \mid L \leq C \leq U\}$. Abusing notation, we identify the set $\mathcal{C}$ with the pair (L, U). Notice that since $U(p)$ can be ∞ for some element $p \in P$, a cube can contain an infinite number of multisets.

A finite union of cubes $\bigcup_{i=1}^{m}(L_i, U_i)$ is called a *counting constraint* and the set of multisets $\bigcup_{i=1}^{m}\mathcal{C}_i$ it describes is called a *counting set*. Given a counting constraint Γ, we let $[\Gamma]$ denote the counting set described by Γ. Notice that two different counting constraints may describe the same counting set. For example, let $P = \{p\}$ and let $(L, U) = (1, 3)$, $(L', U') = (2, 4)$, $(L'', U'') = (1, 4)$. The counting constraints $(L, U) \cup (L', U')$ and (L'', U'') define the same counting set. It is easy to show that counting constraints and counting sets are closed under Boolean operations.

Norms. Let $\mathcal{C} = (L, U)$ be a cube. Let $\|\mathcal{C}\|_l$ be the sum of the components of L. Let $\|\mathcal{C}\|_u$ be the sum of the finite components of U if there are any, and 0 otherwise. The *norm* of $\mathcal{C}$ is the maximum of $\|\mathcal{C}\|_l$ and $\|\mathcal{C}\|_u$, denoted by $\|\mathcal{C}\|$. We define the norm of a counting constraint $\Gamma = \bigcup_{i=1}^{m}\mathcal{C}_i$ as $\|\Gamma\| \stackrel{\text{def}}{=} \max_{i\in[1,m]}\{\|\mathcal{C}_i\|\}$. The norm of a counting set $\mathcal{S}$ is the smallest norm of a counting constraint describing $\mathcal{S}$, that is, $\|\mathcal{S}\| \stackrel{\text{def}}{=} \min_{\mathcal{S}=[\Gamma]}\{\|\Gamma\|\}$. The norms of the union, intersection and complement can be bounded in the following way.

Proposition 1. Let $\mathcal{S}_1, \mathcal{S}_2$ be counting sets over a finite set P. The norms of the union, intersection and complement satisfy:

- $\|\mathcal{S}_1 \cup \mathcal{S}_2\| \leq \max\{\|\mathcal{S}_1\|, \|\mathcal{S}_2\|\}$,
- $\|\mathcal{S}_1 \cap \mathcal{S}_2\| \leq \|\mathcal{S}_1\| + \|\mathcal{S}_2\|$, and
- $\|\overline{\mathcal{S}_1}\| \leq |P| \cdot \|\mathcal{S}_1\| + |P|$.

Proof. In the following, let Γ_1 and Γ_2 be counting constraints describing $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. We prove the bounds on $\|\Gamma_1\|, \|\Gamma_2\|$, and they hold for $\|\mathcal{S}_1\|, \|\mathcal{S}_2\|$ by definition of the norm of a counting set.

Union. Let Γ be the union of the two counting constraints Γ_1, Γ_2. It is still a counting constraint as a union of cubes, and the result follows from the definition of the norm.

Intersection. For this proof, we consider a cube representation (L, U) as a collection of constraints over $n = |P|$ variables $x_1, \ldots, x_n$ of the form $l_i \leq x_i$ or $x_i \leq u_i$ with $l_i \in \mathbb{N}$ and $u_i \in \mathbb{N} \cup \infty$. Each variable x_i is associated to an element $p_i \in P$, for an arbitrary ordering of $P = \{p_1, \ldots, p_n\}$. A cube representation can now be seen as a conjunction of such constraints, one lower bound and one upper bound for each x_i for $i \in \{1, \ldots, n\}$. We call such a $2n$-conjunction a *minterm*. Counting constraints Γ_1, Γ_2 are thus disjunctions of minterms, noted γ_1, γ_2 respectively. The intersection of Γ_1, Γ_2 is the conjunction $\gamma_1 \wedge \gamma_2$.

We rearrange this conjunction into a disjunction of minterms by using the following steps: put $\gamma_1 \wedge \gamma_2$ in disjunctive normal form. Remove conjunctions containing the unsatisfiable constraints $l \leq x_i \wedge x_i \leq u$ with $l > u$. Remove redundant constraints inside conjunctions: replace $(l_1 \leq x \wedge l_2 \leq x)$ by $\max\{l_1, l_2\} \leq x$, and replace $(x \leq u_1 \wedge x \leq u_2)$ by $x \leq \min\{u_1, u_2\}$. If a conjunction does not contain a lower bound (respectively upper bound) for x_i, add $0 \leq x_i$ (respectively $x_i \leq \infty$), thus making it a minterm. The disjunction of these minterms is the counting constraint Γ we are looking for. Each lower bound in Γ is at most the maximal lower bound in the minterms of Γ_1 and Γ_2, so $\|\Gamma\|_l \leq \|\Gamma_1\|_l + \|\Gamma_2\|_l$. Each upper bound in Γ is at most the minimal upper bound in the minterms of Γ_1 and Γ_2, so $\|\Gamma\|_u \leq \min\{\|\Gamma_1\|_u, \|\Gamma_2\|_u\}$. Thus norm of Γ is smaller or equal to $\|\Gamma_1\| + \|\Gamma_2\|$.

Complement. We reuse the constraint and minterm formalism above. The complement is represented by the negation of the disjunction of minterms. We rearrange it into a disjunction of minterms by first putting it in disjunctive normal form, then replacing $\neg(x_i \leq c)$ by $x_i \geq c + 1$ if $c \neq \infty$ or removing the enclosing conjunction otherwise, then replacing $\neg(x_i \geq c)$ by $x_i \leq c - 1$ if $c \in \mathbb{N} \setminus \{0\}$ or removing the enclosing conjunction otherwise, and finally by using the same rules as above, i.e removing unsatisfiable constraints from conjunctions, replacing redundant ones and adding lower and upper bounds if they are missing. We obtain a disjunction of minterms giving us the counting constraint Γ we are looking for. It has minterms with lower bounds of the form $u + 1$, with u an upper bound in a minterm of the original constraint, meaning $u \leq \|\Gamma_1\|$. The reasoning is similar for the new upper bounds of the minterms; this yields $\|\Gamma\| \leq n\|\Gamma_1\| + n$. □

Example 2. Consider the cubes $\mathcal{C}_1 = (L_1, U_1)$ and $\mathcal{C}_2 = (L_2, U_2)$ over a set $P = \{p_1, p_2, p_3\}$, with

$$\begin{array}{cccc} 0 & \infty & 10 & 14 \\ L_1 = 3 & U_1 = 6 & L_2 = 2 & U_2 = \infty \\ 4 & \infty & 0 & \infty \end{array}$$

The norm of $\mathcal{C}_1$ is $\|\mathcal{C}_1\| = \max\{\|\mathcal{C}_1\|_l, \|\mathcal{C}_1\|_u\} = \max\{7, 6\} = 7$. The norm of $\mathcal{C}_2$ is $\|\mathcal{C}_2\| = \max\{\|\mathcal{C}_2\|_l, \|\mathcal{C}_2\|_u\} = \max\{12, 14\} = 14$.

Counting constraint $\mathcal{C}_1 \cup \mathcal{C}_2$ has norm $\|\mathcal{C}_1 \cup \mathcal{C}_2\| = \max\{\|\mathcal{C}_1\|, \|\mathcal{C}_2\|\} = 14$. Counting constraint $\mathcal{C}_1 \cap \mathcal{C}_2$ is equal to cube $\mathcal{C}_3 = (L_3, U_3)$, where

$$\begin{array}{cc} 10 & 14 \\ L_3 = 3 & U_3 = 6 \\ 4 & \infty \end{array}$$

The norm of $\mathcal{C}_1 \cap \mathcal{C}_2$ is $\|\mathcal{C}_1 \cap \mathcal{C}_2\| = \|\mathcal{C}_3\| = 20$. Counting constraint $\overline{\mathcal{C}_1}$ is equal to counting constraint $\mathcal{C}_4 \cup \mathcal{C}_5$ with $\mathcal{C}_4 = (L_4, U_4)$ and $\mathcal{C}_5 = (L_5, U_5)$, where

$$\begin{array}{rrrr} 0 & \infty & 0 & \infty \\ L_4 = 0 & U_4 = 2 & L_5 = 7 & U_5 = \infty \\ 0 & 3 & 0 & 3 \end{array}$$

The norm of $\overline{\mathcal{C}_1}$ is $\|\overline{\mathcal{C}_1}\| = \max\{\|\mathcal{C}_4\|, \|\mathcal{C}_5\|\} = 7$.

Reachability, coverability and liveness. Given a Petri net $\mathcal{N} = (P, T, F)$, the reachability problem can be generalized to cubes and counting sets. Cubes and counting sets over P are seen as sets of markings. The *cube-reachability* problem consists of deciding, given two cubes $\mathcal{C}, \mathcal{C}'$ over P, whether there exist markings $M \in \mathcal{C}$ and $M' \in \mathcal{C}'$ such that M' is reachable from M in $\mathcal{N}$. If this is the case, we say $\mathcal{C}'$ is reachable from $\mathcal{C}$. The *counting set-reachability* problem consists of deciding, given two counting sets $\mathcal{S}, \mathcal{S}'$, whether there exist markings $M \in \mathcal{S}$ and $M' \in \mathcal{S}'$ such that M' is reachable from M in $\mathcal{N}$. We define *cube-coverability* and *counting set-coverability* in an analogous way. The *cube-liveness* problem consists of deciding if, given a net N and a cube $\mathcal{M}$ of markings of N, every marking of $\mathcal{M}$ is live. The *counting set-liveness* is defined in an analogous way. The complexity results in this thesis are true irrespective of whether the bounds in a given input cube or input counting constraint are encoded in unary or in binary.

Semilinear sets. Let $d \geq 1$. A *linear set*[2] is a set $L(\boldsymbol{b}, Pe)$ of vectors of $\mathbb{N}^d$ defined using a *base* vector $\boldsymbol{b} \in \mathbb{N}^d$ and a finite set of *period* vectors $Pe = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_n}\} \subseteq \mathbb{N}^d$ such that

$$L(\boldsymbol{b}, Pe) = \{\boldsymbol{b} + \lambda_1 \boldsymbol{p_1} + \ldots + \lambda_n \boldsymbol{p_n} \mid \lambda_i \geq 0 \text{ for } 1 \leq i \leq n\}.$$

A *semilinear set* is a finite union of linear sets.

A cube is a semilinear set. Let $\mathcal{C} = (L, U)$ be a cube over a finite set $P = \{p_1, \ldots, p_d\}$. Notice that multisets over P can be seen as vectors of $\mathbb{N}^d$ and vice versa. Let $\boldsymbol{p_i}$ be the multiset over P consisting of one occurrence of p_i. Let Pe be the set of $\boldsymbol{p_i}$ such that $U(p_i) = \infty$. The cube $\mathcal{C}$ is a semilinear set as a finite union of the linear sets $L(\boldsymbol{b}, Pe)$ for every multiset $\boldsymbol{b}$ over P such that $\boldsymbol{b}(p) = L(p)$ for $p \in P$ such that $U(p) = \infty$ and $L(p) \leq \boldsymbol{b}(p) \leq U(p)$ otherwise. A counting set is a semilinear set as a finite union of linear sets.

[2] We use the notations of [54].

2.4 Generalized Reachability Problems

Let $\mathcal{N} = (P, T, F)$ be a Petri net. A *generalized reachability expression* of $\mathcal{N}$ is an expression that is constructed by the following syntax:

$$E := \Gamma \mid \mathit{post}^*(E) \mid \mathit{pre}^*(E) \mid E \cap E \mid E \cup E \mid \overline{E}$$

where Γ is any counting constraint over P.

If E is a generalized reachability expression, then the *length* of E, denoted by $|E|$, is defined as follows:

- if $E = \Gamma$ where Γ is a counting constraint, then $|E| = 1$;
- if $E = \mathit{post}^*(E_1)$ or $E = \mathit{pre}^*(E_1)$, then $|E| = |E_1| + 1$;
- if $E = E_1 \cup E_2$ or $E = E_1 \cap E_2$, then $|E| = |E_1| + |E_2|$;
- if $E = \overline{E_1}$, then $|E| = |E_1| + 1$.

We denote by $[E]$ the set of markings that is described by a generalized reachability expression E, and it is defined as follows:

- if $E = \Gamma$ where Γ is a counting constraint, then $[E] = [\Gamma]$;
- if $E = \mathit{post}^*(E_1)$, then $[E] = \mathit{post}^*([E_1])$;
- if $E = \mathit{pre}^*(E_1)$, then $[E] = \mathit{pre}^*([E_1])$;
- if $E = E_1 \triangle E_2$, then $[E] = [E_1] \triangle [E_2]$ for $\triangle \in \{\cup, \cap\}$;
- if $E = \overline{E_1}$, then $[E] = \overline{[E_1]}$.

Given a Petri net $\mathcal{N}$, a generalized reachability expression E and a marking M, the *generalized reachability membership* problem consists of deciding whether M is in $[E]$. Given a Petri net $\mathcal{N}$ and a generalized reachability expression E, the *generalized reachability emptiness* problem consists of deciding whether $[E]$ is empty. These two problems are called the *generalized reachability problems*. For example, cube-reachability can be expressed as a generalized reachability problem: a cube $\mathcal{C}$ can reach a cube $\mathcal{C}'$ if and only if the set represented by the generalized reachability expression $\mathcal{C}' \cap \mathit{post}^*(\mathcal{C})$ is non empty.

2.5 Complexity

We use the usual complexity classes P, NP, PSPACE, and EXPSPACE, and we point the reader to [61] or any other classic book for formal definitions.

A non-usual class we evoke is the ACKERMANN class. It is a class of problems decidable by a Turing machine that may require runtime which is non-primitive recursive – the runtime is seen as a function of the input. A primitive recursive function is, informally, a function which is computable by a program in which all the loops are **for**-loops; **for**-loops

provide an upper bound on the number of iterations of the loop, so these programs are guaranteed to terminate. An example of a primitive recursive function is $n \mapsto exp_n(n)$ for $n \in \mathbb{N}$, where the exp_k are defined inductively using $exp_0(x) = x$ and $exp_{k+1}(x) = 2^{exp_k(x)}$. A non-primitive recursive function is a function that cannot be bounded above by any primitive recursive function. So a problem that is ACKERMANN-hard can only be computed by a Turing machine that uses *at least* non-primitive recursive runtime. The reachability problem for Petri nets is an example of an ACKERMANN-hard problem. A reader interested in a detailed definition of this class can refer to [81].

3 Immediate Observation Nets

In this chapter we define and study our first class of observation Petri nets, immediate observation (IO) nets. Section 3.1 gives the formal definition and basic properties. Section 3.2 proves complexity lower bounds for classic analysis problems. Section 3.3 introduces tools and techniques that are then used in Section 3.4 to prove our main results for IO nets, including upper complexity bounds for generalized reachability problems.

3.1 Definition and Examples

An immediate observation net is a Petri net with transitions of a certain shape: informally, a token moves from a place p to a place p' by *observing* the presence of another token in q.

Definition 2. A transition t of a Petri net is an *immediate observation transition* (IO transition) if there are places p_s, p_d, p_o, not necessarily distinct, such that ${}^\bullet t = \wr p_s, p_o \int$ and $t^\bullet = \wr p_d, p_o \int$, or such that ${}^\bullet t = \wr p_s \int$ and $t^\bullet = \wr p_d \int$. We call p_s, p_d, p_o the *source*, *destination*, and *observed* places of t, respectively. A Petri net is an *immediate observation net* (IO net) if all its transitions are IO transitions.

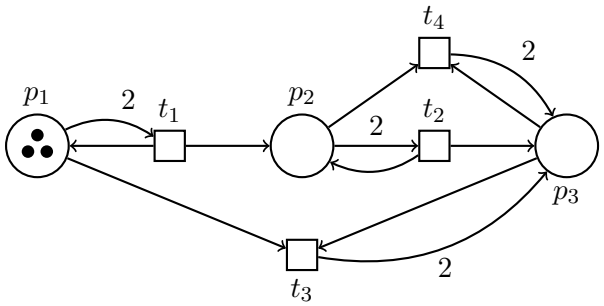


Figure 3.1: *An IO net from [7].*

In the following, for simplicity, we consider only IO nets in which every transition has an observed place. Given an IO net in which this is not the case, it suffices to add an extra marked place which acts as observed place for all the transitions without one. In the following, we sometimes write $t = x \xrightarrow{y} z$ when x is the source place, y the observed place, and z the destination place of a transition t. We equate an IO net (P, T, F) with the pair (P, δ), where $\delta = \left\{(x, y, z) \in P^3 \mid t = x \xrightarrow{y} z \in T\right\}$.

Example 3. In the Petri net of Figure 2.1 , transitions t_2 and t_3 are IO transitions (without observed places). However, transition t_1 is not, so it is not an IO net. Figure 3.1 shows an IO net taken from the literature on population protocols (Section 8.1 of [7]). What it

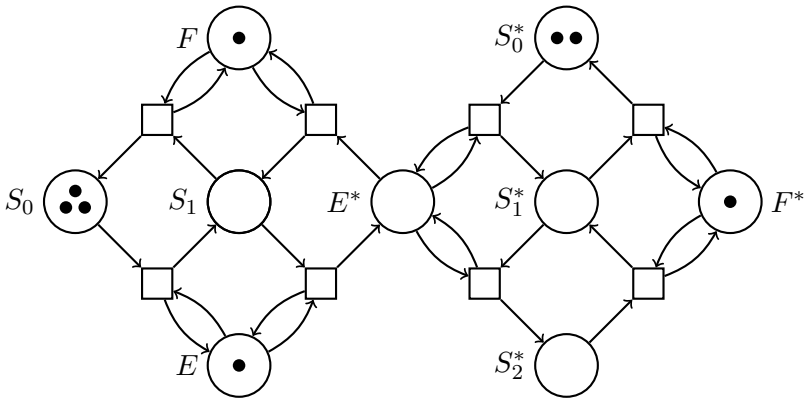


Figure 3.2: *An IO net from [4].*

models is detailed later in Example 9. Figure 3.2 gives another example of an IO net, this time taken from the literature on chemical reaction networks ([4], Section 6.2)[1]. Intuitively, the transitions model chemical reactions triggered by enzymes. For example, substrate S_0 is converted into product S_1 by enzyme E. The net models a cascade of reactions in which some products, for example E^*, may act as an enzyme for the following stage.

Given an arc a from x to y of weight k for $(x, y) \in (P \times T) \cup (T \times P)$ and $k \geq 0$, we call the arc from y to x of weight k the *reverse* arc of a. If we take an IO net $\mathcal{N}$ and replace all its arcs with their reverse arcs, the new Petri net $\mathcal{N}'$ obtained is still an IO net. Given two markings M, M' over the set of places of $\mathcal{N}$ (equal to the set of places of $\mathcal{N}'$), M' is reachable from M in $\mathcal{N}$ if and only if M is reachable from M' in $\mathcal{N}'$.

Remark 3. We will also consider immediate multiple observation (IMO) nets, which are like IO nets except that there may be more than one observed place. This case is treated in Chapter 6. The results for IO nets shown in the present chapter also hold for IMO nets, with bounds changing only by a factor corresponding to the maximal number of observations. We treat this case separately to make the proofs of this chapter cleaner.

Unlike general Petri nets, IO nets have a *copycat* property, which we describe informally here. Consider a firing sequence $M \xrightarrow{*} M'$. We can think of the tokens as each following a trajectory in the firing sequence, going from place to place following the transitions. If we add a token to a place q of M already containing an token, i.e. such that $M(q) \geq 1$, then this new token can "copy" the trajectory of the old token, mimicking its transitions along the firing sequence. The intuition is that this is possible for IO nets because the token in the observed place allowing the transition is not consumed when the transition is fired. This copycat property is implicitly used in many of the proofs of the following sections. It is formally defined in Lemma 12 of the next section.

[1]The Petri net presented here is simplified, but equivalent to that of [4] for properties that depend only on the reachability graph, as are the net properties studied in [4].

Example 4. Consider the firing sequence in the IO net of Figure 3.1: $(3,0,0) \xrightarrow{t_1} (2,1,0) \xrightarrow{t_1} (1,2,0) \xrightarrow{t_2} (1,1,1)$. One of the tokens goes to places p_1, p_2, p_2, p_3. Let us add a token to p_1 in the first marking. It can copy the "trajectory" of the previous token, yielding the firing sequence $(4,0,0) \xrightarrow{t_1} (3,1,0) \xrightarrow{t_1} (2,2,0) \xrightarrow{t_1} (1,3,0) \xrightarrow{t_2} (1,2,1) \xrightarrow{t_2} (1,1,2)$.

On the contrary, consider the Petri net of Figure 2.1, and firing sequence $(1,1,0) \xrightarrow{t_1} (0,2,0) \xrightarrow{t_2} (1,0,1) \xrightarrow{t_3} (1,1,0)$. Let us add a token to p_1 in the first marking. This token cannot copy the trajectory of any other token in the firing sequence, because t_1 can be fired once but not twice from $(2,1,0)$. Transition t_1 is fired by taking a token from p_1 and a token from p_2, synchronizing their actions.

Observe that IO transitions have a preset of size two and a postset of size two, so we have:

Fact 4. Immediate observation nets are conservative.

The transitions of the Petri net of Figure 2.1 also each have a preset and a postset of equal size. It is an example of a conservative Petri net which is not an IO net. For conservative Petri nets, the following holds.

Theorem 5. For conservative Petri nets:

- Reachability, coverability, and liveness are in PSPACE.
- Cube-reachability and cube-coverability are as hard as reachability and coverability for general Petri nets, so non-primitive recursive and EXPSPACE-hard, respectively.

Proof. The first part is proved in [57], we give a sketch here. By Savitch's Theorem, NPSPACE=PSPACE, so it is enough to provide a nondeterministic algorithm. Let $\mathcal{N}$ be conservative Petri net, let M and M' be markings of $\mathcal{N}$. Since the net is conservative, the size of markings along a firing sequence is constant. To check if M' is reachable (respectively coverable) from M the algorithm guesses a sequence of markings, step-by-step reachable, only remembering the current one and the next one, and checking whether it equals (respectively covers) M'. If it does, the algorithm stops and answers that M' is reachable (respectively coverable) from M. To show that there exists an algorithm running in polynomial space for liveness, the proof uses that coPSPACE=PSPACE and gives an algorithm for checking if M is *not* live. The algorithm guesses a transition t and a marking M_1 reachable from M, checking that it is indeed reachable with the above algorithm. Then it checks for each marking M_2 reachable from M_1 whether t is enabled at M_2. If t is enabled at no marking reachable from M_1, the algorithm answers that M is not live.

For the second part, let $\mathcal{N} = (P, T, F)$ be an arbitrary Petri net. We construct a Petri net $\mathcal{N}' = (P \cup \{r, s\}, T, F')$, where r and s are two new places, the *repository* and the *sink*, and F' is defined so that, intuitively, transitions of $\mathcal{N}'$ neither create nor destroy tokens. Formally, for every transition t:

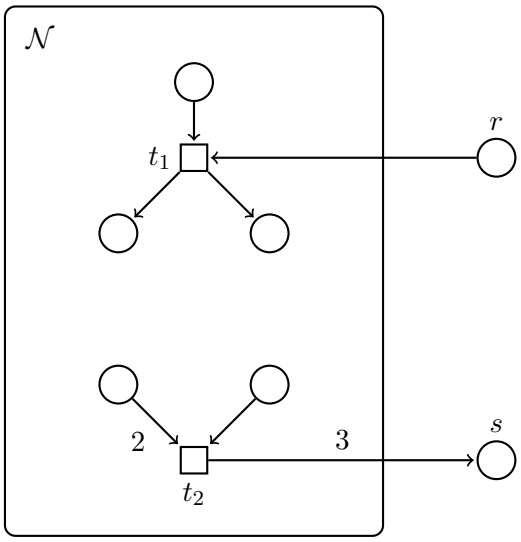


Figure 3.3: *Conservative net $\mathcal{N}'$ constructed from Petri net $\mathcal{N}$ in the proof of Theorem 5.*

- $F'(p,t) = F(p,t)$ and $F'(t,p) = F(t,p)$ for every $p \in P$.
- $F'(t,r) = 0$, and $F'(r,t) = \max\{0, |t^\bullet| - |{}^\bullet t|\}$.
- $F'(s,t) = 0$, and $F'(t,s) = \max\{0, |{}^\bullet t| - |t^\bullet|\}$.

In $\mathcal{N}'$ we have $\sum_{p \in P \cup \{r,s\}} F'(p,t) = \sum_{p \in P \cup \{r,s\}} F'(t,p)$ for every transition t, and so $\mathcal{N}'$ is conservative. The construction is sketched in Figure 3.3: a Petri net $\mathcal{N}$ contains a transition t_1 which has an input arc and two output arcs, all of weight 1. We add a repository place r and an arc from r to t_1, so that t_1 no longer creates tokens. The net also contains a transition t_2 with an input arc of weight 2 and an input arc of weight 1. We add a sink place s and an arc from t_2 to s of weight 3, so that t_2 no longer destroys tokens.

Given a marking M of $\mathcal{N}$, let (L_M, U_M) be the cube of $\mathcal{N}'$ given by $L_M(p) = M(p) = U_M(p)$ for every $p \in P$, $L_M(r) = L_M(s) = 0$ and $U_M(r) = U_M(s) = \infty$. Clearly, we have: M_2 is reachable (coverable) from M_1 in $\mathcal{N}$ if and only if (L_{M_2}, U_{M_2}) is reachable (coverable) from (L_{M_1}, U_{M_1}) in $\mathcal{N}'$, and we are done. □

IO nets inherit these complexity upper bounds, as they are conservative. But we will show that for IO nets, the cube-parameterized versions of reachability, coverability and liveness are also in PSPACE. This pinpoints the essential property of the class: loosely speaking, deciding standard problems for infinitely many markings is not harder than deciding them for one marking. More precisely, we will show a theorem stating that any generalized reachability problem can be solved in polynomial space. That is, membership or emptiness for any combination of atoms using boolean operations, pre^* and $post^*$ can be evaluated in polynomial space, where an atom is a counting set.

The tools to prove this are presented in Section 3.3, and the results are proved in Section 3.4. But first we show that IO nets can simulate bounded-tape Turing machines, providing us with PSPACE lower bounds for reachability, coverability, and liveness for IO nets.

3.2 Lower Bound

The standard simulation of bounded-tape Turing machines by 1-safe Petri nets, as described for example in [24, 37], can be modified to produce an IO net (actually, a 1-safe IO net). Using this result, we can then prove that the reachability, coverability, and liveness problems are PSPACE-hard. Since a set consisting of a single marking is a special case of a cube, the result carries over to the cube-versions of the problems. This underlines the importance of the result that reachability and coverability are still solvable in PSPACE when *parameterized* by cubes.

We fix a deterministic Turing machine $\mathcal{M}$ with set of control states Q, alphabet Σ containing the empty symbol $\sqcup$, and partial transition function $\delta\colon Q \times \Sigma \to Q \times \Sigma \times D$ ($D = \{-1, +1\}$). We let K denote an upper bound on the number of tape cells visited by the computation of $\mathcal{M}$ on empty tape. The *implementation* of $\mathcal{M}$ is the IO Petri net $N_{\mathcal{M}}$ described below.

Places of $N_{\mathcal{M}}$. The net $N_{\mathcal{M}}$ contains two sets of *cell places* and *head places* modelling the state of the tape cells and the head, respectively. The cell places are:

- $\mathit{off}[\sigma, n]$ for each $\sigma \in \Sigma$ and $1 \leq n \leq K$. A token on $\mathit{off}[\sigma, n]$ denotes that cell n contains symbol σ, and the cell is "off", i.e., the head is not on it.
- $\mathit{on}[\sigma, n]$ for each $\sigma \in \Sigma$ and $1 \leq n \leq K$, with analogous intended meaning.

The head places are:

- $\mathit{at}[q, n]$ for each $q \in Q$ and $1 \leq n \leq K$. A token on $\mathit{at}[q, n]$ denotes that the head is in control state q and at cell n.
- $\mathit{move}[q, \sigma, n, d]$ for each $q \in Q$, $\sigma \in \Sigma$, $1 \leq n \leq K$ and every $d \in D$ such that $1 \leq n + d \leq K$. A token on $\mathit{move}[q, \sigma, n, d]$ denotes that head is in control state q, has left cell n after writing symbol σ on it, and is currently moving in the direction given by d.

Transitions of $N_{\mathcal{M}}$. Intuitively, the implementation of $\mathcal{M}$ contains a set of *cell transitions* in which a cell observes the head and changes its state, and a set of *head transitions* in which the head observes a cell. Further, each of these sets contains transitions of two types. The set of cell transitions contains:

- Type **1a**: $(\mathit{off}[\sigma, n]\ ,\ \mathit{at}[q, n]) \mapsto (\mathit{on}[\sigma, n]\ ,\ \mathit{at}[q, n])$ for every state $q \in Q$, symbol $\sigma \in \Sigma$, and cell $1 \leq n \leq K$.
  The n-th cell, currently *off*, observes that the head is on it, and switches itself *on*.

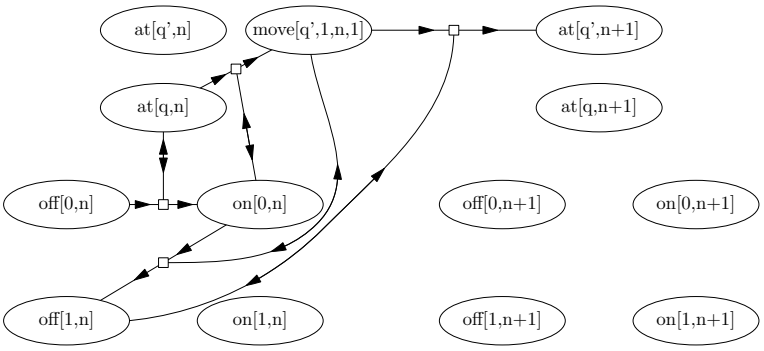


Figure 3.4: *Some of the places and transitions involved in modelling a Turing machine.*

- Type **1b**: $(on[\sigma, n]\ ,\ move[q, \sigma', n, d]) \mapsto (off[\sigma', n]\ ,\ move[q, \sigma', n, d])$ for every $q \in Q$, $\sigma \in \Sigma$, and $1 \leq n \leq K$ such that $1 \leq n + d \leq K$.
  The n-th cell, currently *on*, observes that the head has left after writing σ', and switches itself *off* (accepting the character the head intended to write).

The set of head transitions contains:

- Type **2a**: $(at[q, n]\ ,\ on[\sigma, n]) \mapsto (move[\delta_Q(q, \sigma), \delta_\Sigma(q, \sigma), n, \delta_D(q, \sigma)]\ ,\ on[\sigma, n])$ for every $q \in Q$, $\sigma \in \Sigma$, and $1 \leq n \leq K$ such that $1 \leq n + \delta_D(q, \sigma) \leq K$.
  The head, currently on cell n, observes that the cell is *on*, writes the new symbol on it, and leaves.
- Type **2b**: $(move[q, \sigma, n, d]\ ,\ off[\sigma, n]) \mapsto (at[q, n + d]\ ,\ off[\sigma, n])$ for every $q \in Q$, $\sigma \in \Sigma$, and $1 \leq n \leq K$ such that $1 \leq n + d \leq K$.
  The head, currently moving, observes that the old cell has turned *off*, and places itself on the new cell.

This concludes the definition of $N_{\mathcal{M}}$. In Theorem 9 below we formalize the relation between the Turing machine $\mathcal{M}$ and its implementation $N_{\mathcal{M}}$, using the following definition.

Definition 6. Given a configuration c of $\mathcal{M}$ with control state q, tape content $\sigma_1\sigma_2 \cdots \sigma_K$, and head on cell $n \leq K$, we denote M_c the marking that puts a token in $off[\sigma_i, i]$ for each $1 \leq i \leq K$, a token in $at[q, n]$, and no tokens elsewhere.

Figure 3.4 illustrates transitions involved in modelling a single step of a Turing machine that reads 0, writes 1, moves head to the right and switches the control state from q to q'.

Definition 7. A marking of $N_{\mathcal{M}}$ is a *modelling marking* if the following conditions hold.

1. If a place is marked, it is marked with a single token.
2. For every $1 \leq n \leq K$ exactly one of the $2|\Sigma|$ places $on[\sigma, n]$, $off[\sigma, n]$ is marked.
   (Intuitively: every cell is either *on* or *off* and contains exactly one symbol.)
3. Exactly one of all the head places is marked.

4. If a cell place $on[\sigma, n]$ is marked, then a head place $at[q, n]$ or $move[q, \sigma', n, d]$ is marked for some σ' and d.
5. If a head place $move[q, \sigma, n, d]$ is marked, either $on[\sigma', n]$ is marked for some σ', of $\mathit{off}[\sigma, n]$ is marked.

Note that for every configuration c of $\mathcal{M}$ the marking M_c is a modelling marking.

Lemma 8. For every modelling marking M of $N_{\mathcal{M}}$:

(1) M enables at most one transition.
(2) If M enables no transitions, then it marks places $on[\sigma, n]$ and $at[q, n]$ for some $q \in Q$, $\sigma \in \Sigma$, and $1 \leq n \leq K$.
(3) If $M \to M'$, then M' is also a modelling marking.

Proof. (1) All possible transitions require tokens at two places, one of type $on[\cdot, n]$ or $\mathit{off}[\cdot, n]$ and one of type $at[\cdot, n]$ or $move[\cdot, n, \cdot, \cdot]$, with the same n. But the modelling condition requires that there can be at most one such pair marked at M.
(2) By point 3 of Definition 7, M marks exactly one head place. If it is a place $move[q, \sigma, n, d]$, then by point 5 either $on[\sigma', n]$ or $\mathit{off}[\sigma, n]$ is marked. So there is a transition of type **1b** or **2b** that is enabled. If the marked head place is a place $at[q, n]$, then by point 2 there is a unique σ such that one of $on[\sigma, n]$ and $\mathit{off}[\sigma, n]$ is marked. If it is $\mathit{off}[\sigma, n]$ then a a transition of type **1a** is enabled. The only case left in which M may enable no transition is if the marked head place is of the form $on[\sigma, n]$ and the marked cell place with same n is $on[\sigma, n]$. Other cell places marked at M do not enable any transition. The only transition type possibly enabled is **2a**; and these are not defined for every $at[q, n]$ and $on[\sigma, n]$.
(3) Every transition consumes and produces one token at $\mathit{off}[\cdot, n]$ or $on[\cdot, n]$ place, and the new place has the same n. Every transition consumes and produces one token at a $move[\cdot, \cdot, \cdot, \cdot]$ or $at[\cdot, \cdot]$ place. If an $on[\cdot, n]$ place becomes marked after a transition, it has the same n as the marked $at[\cdot, n]$ place of the markings before and after firing the transition; if an $on[\cdot, n]$ place stays marked, the token is moved from a $at[\cdot, n]$ to a $move[\cdot, n, \cdot, \cdot]$ place with the same n. When $move[q, \sigma, n, d]$ becomes marked, the transition needs a marked $on[\cdot, n]$ place. When $move[q, \sigma, n, d]$ stays marked, the transition marks a $\mathit{off}[\sigma, n]$ place. □

Now we state our simulation theorem and hardness result.

Theorem 9. For every two configurations c, c' of $\mathcal{M}$: $c \to c'$ if and only if $M_c \xrightarrow{t_1 t_2 t_3 t_4} M_{c'}$ in $N_{\mathcal{M}}$ for some transitions t_1, t_2, t_3, t_4 of types **1a**, **2a**, **1b**, **2b**, respectively.

Proof. Let c be a configuration of $\mathcal{M}$ with control state q, tape content $\sigma_1 \sigma_2 \cdots \sigma_K$, and head on cell $n \leq K$. Let c' be a configuration of $\mathcal{M}$ with control state q', tape content $\sigma'_1 \sigma'_2 \cdots \sigma'_K$, and head on cell $n' \leq K$.

If $c \to c'$, then $\delta_Q(q, \sigma_n) = q', \delta_\Sigma(q, \sigma_n) = \sigma'_n, n' = n + \delta_D(q, \sigma_n)$ with $1 \leq n' \leq K$, and for all $i \neq n, \sigma_i = \sigma'_i$. By definition of M_c, a sequence $\xrightarrow{t_1 t_2 t_3 t_4}$ is enabled where

$$
\begin{aligned}
t_1 &= (\mathit{off}[\sigma_n, n], at[q, n]) \mapsto (\mathit{on}[\sigma_n, n], at[q, n]),\\
t_2 &= (at[q, n], \mathit{on}[\sigma_n, n]) \mapsto (\mathit{move}[q', \sigma'_n, n, \delta_D(q, \sigma_n)], \mathit{on}[\sigma_n, n]),\\
t_3 &= (\mathit{on}[\sigma_n, n], \mathit{move}[q', \sigma'_n, n, \delta_D(q, \sigma_n)]) \mapsto (\mathit{off}[\sigma'_n, n], \mathit{move}[q', \sigma'_n, n, \delta_D(q, \sigma_n)]),\\
t_4 &= (\mathit{move}[q', \sigma'_n, n, \delta_D(q, \sigma_n)], \mathit{off}[\sigma'_n, n]) \mapsto (at[q', n'], \mathit{off}[\sigma'_n, n]).
\end{aligned}
$$

The marking reached from M_c by firing t_1, t_2, t_3, t_4 is $M_{c'}$.

Assume now that $M_c \xrightarrow{t_1 t_2 t_3 t_4} M_{c'}$ in $N_\mathcal{M}$ for some transitions t_1, t_2, t_3, t_4 of types **1a**, **2a**, **1b**, **2b**, respectively. Transition t_1 must be equal to $(\mathit{off}[\sigma_n, n], at[q, n]) \mapsto (\mathit{on}[\sigma_n, n], at[q, n])$ since the only head place marked at M_c is $at[q, n]$ and all marked cell places are of type *off*. After firing t_1 from M_c, place $\mathit{on}[\sigma_n, n]$ is marked and it is the only marked cell place of type *on*. So transition t_2 of type **2a** must be equal to $(at[q, n], \mathit{on}[\sigma_n, n]) \mapsto (\mathit{move}[\delta_Q(q, \sigma_n), \delta_\Sigma(q, \sigma_n), n, \delta_D(q, \sigma_n)], \mathit{on}[\sigma_n, n])$. Similarly we have a unique choice for t_3 of type **1b** and t_4 of type **2b**. The existence of t_2 implies that $\delta(q, \sigma_n)$ is defined and that $1 \leq n + \delta_D(q, \sigma_n) \leq K$. Since $M_c \xrightarrow{t_1 t_2 t_3 t_4} M_{c'}$, this entails that $\delta_Q(q, \sigma_n) = q', \delta_\Sigma(q, \sigma_n) = \sigma'_n, n' = n + \delta_D(q, \sigma_n)$ and for all $i \neq n, \sigma_i = \sigma'_i$. Thus in $\mathcal{M}$, configuration c can reach in one step the configuration with control state q', tape content $\sigma'_1 \sigma'_2 \cdots \sigma'_K$, and head on cell n', i.e. c'. □

Theorem 10. The reachability, coverability and liveness problems for IO nets are PSPACE-hard.

Proof. The proof is routine. Let p be a fixed polynomial satisfying $p(n) \geq n$ for all n. Consider the set of deterministic Turing machines whose set of states contains two distinct distinguished states q_{acc}, q_{rej}, and whose computation on empty tape satisfies the following conditions:

- The computation never visits more than $p(n)$ cells, where n is the size of $\mathcal{M}$, and visits the set $\{q_{acc}, q_{rej}\}$ of states exactly once.
- The computation ends in a configuration c with empty tape, head on the first cell, and control state either q_{acc} or q_{rej}.

We say that the machine *accepts* (*rejects*) if it terminates in q_{acc} (q_{rej}). It is well known that the problem whether such a machine accepts on empty tape is PSPACE-hard. Given such a machine $\mathcal{M}$, let $N_\mathcal{M}$ be its associated IO net, and let M_0 and M be the modelling markings describing the initial configuration and the unique accepting configuration. Then $\mathcal{M}$ accepts if and only if M is reachable from M_0. Acceptance is thus reduced to reachability in IO nets. In fact, $\mathcal{M}$ accepts if and only if some marking reachable from M_0 covers M;

since the number of tokens in M and M_0 is the same and since the token number stays constant in a firing sequence, this is equivalent to M being reachable from M_0. Acceptance is thus reduced to coverability in IO nets.

Now we reduce acceptance of bounded-tape Turing machines to liveness of IO nets. Consider a Turing machine $\mathcal{M}$ with accepting state q_{acc} and K an upper bound on the number of tape cells visited by a computation of $\mathcal{M}$ on empty tape. Let $N_{\mathcal{M}}$ be its associated IO net. We add two additional places to $N_{\mathcal{M}}$, places *observer* and *success*. We add the following transitions for all n such that $1 \leq n \leq K$, and for all places p, q including *observer* and *success*:

- $(observer, at[q_{acc}, n]) \mapsto success, (at[q_{acc}, n])$, and
- $(p, success) \mapsto (q, success)$.

Initially, we place the tokens according to the initial control state and tape contents, and additionally put one token into *observer*. Now, if the Turing machine cannot reach the accepting state, the net will never be able to execute any transition into *success* (so it will not be live). If the Turing machine can reach the accepting state, the only possible sequence of transitions of the net will lead to marking of some place $at[q_{acc}, n]$. Afterwards, the net can optionally switch a cell place from *off* to *on*, but cannot continue further without firing a transition that marks the *success* place.

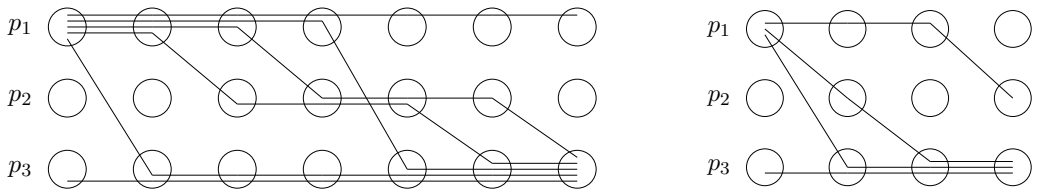
Once the *success* place is marked it stays marked. Additionally, it allows moving tokens between any two places. Our net contains at least two other tokens, which can thus be moved to fire any transition. Therefore if the Turing machine reaches the accepting state, the Petri net is live. This concludes the reduction, which implies PSPACE-hardness of IO net liveness. □

3.3 Pruning and Boosting

The results for IO nets are proved using the idea of "de-anonymizing" tokens, as well as two main lemmas: the Pruning Lemma and the Boosting Lemma. We start by introducing some notions required to state the lemmas, and then the lemmas themselves.

Trajectories and histories. Since the transitions of IO nets do not create or destroy tokens, we can give tokens identities. Given a firing sequence, each token of the initial marking follows a *trajectory* through the places of the net until it reaches the final marking of the sequence. The trajectories of the tokens between given source and target markings constitute a *history*.

Fix an IO net $\mathcal{N}$. A *trajectory* of $\mathcal{N}$ of length k is a sequence $\tau = p_1 \dots p_k$ of places of $\mathcal{N}$. We let $\tau(i)$ denote the i-th place of τ. The *i-th step* of τ is the pair $\tau(i)\tau(i+1)$. If $\tau(i) = \tau(i+1)$ we say the step is *horizontal*, and otherwise it is *non-horizontal*. A *history*



(a) *A realizable history.* **(b)** *A non-realizable history.*

Figure 3.5: *A realizable history and a non-realizable history of the IO net of Figure 3.1.*

H of length h is a multiset of trajectories of length h. Given an index $1 \leq i \leq h$, *the i-th marking of H*, denoted M_H^i, is defined as follows: for every place p, $M_H^i(p)$ is the number of trajectories $\tau \in H$ such that $\tau(i) = p$. The markings M_H^1 and M_H^h are the *initial* and *final* markings of H, and we write $M_H^1 \xrightarrow{H} M_H^h$.

A history H of length $h \geq 1$ is *realizable* if there exist transitions $t_1, \ldots, t_{h-1}$ and numbers $k_1, \ldots, k_{h-1} \geq 0$ such that

- $M_H^1 \xrightarrow{t_1^{k_1}} M_H^2 \cdots M_H^{h-1} \xrightarrow{t_{h-1}^{k_{h-1}}} M_H^h$, where for every t we define $M' \xrightarrow{t^0} M$ iff $M' = M$.
- For every $1 \leq i \leq h-1$, there are exactly k_i trajectories $\tau \in H$ such that $\tau(i)\tau(i+1) = p_s p_d$, where p_s, p_d are the source and target places of t_i, and all other trajectories $\tau \in H$ satisfy $\tau(i) = \tau(i+1)$. Moreover, there is at least one trajectory τ in H such that $\tau(i)\tau(i+1) = p_o p_o$, where p_o is the observed place of t_i.

We say that $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ realizes H. Intuitively, at a step of a realizable history only one transition occurs, although perhaps multiple times, for different tokens. From the definition of realizable history we immediately obtain: $M' \xrightarrow{*} M$ if and only if there exists a realizable history with M' and M as initial and final markings.

We define a measure of the length of firing sequences that abstracts from the number of times a transition is consecutively executed: Let σ be a firing sequence. Let $k_1, \ldots, k_m$ be the unique positive natural numbers such that $\sigma = t_1^{k_1} t_2^{k_2} \ldots t_m^{k_m}$ and $t_i \neq t_{i+1}$ for every $i = 1, \ldots, m-1$. We say that σ has *accelerated length* m, and let $|\sigma|_a$ denote the accelerated length of σ. From the definition of realizable history we immediately obtain: every firing sequence that realizes a history of length h has accelerated length at most h.

Example 5. Figure 3.5a shows a realizable history of the IO net of Figure 3.1. It consists of six trajectories of length seven, where trajectories are represented by paths that go from place to place. The places are represented by circles in a column, with place p_1 at the top, place p_2 in the middle and place p_3 at the bottom. The columns are repeated to graphically represent the order of the steps. Between each column, a trajectory takes either a horizontal step or a non-horizontal step. The multiset of trajectories in the places of column number i gives the i-th marking M_H^i of H. The first column corresponds to

the initial marking $M_H^1 = (5, 0, 1)$, and the last column corresponds to the final marking $M_H^7 = (1, 0, 5)$.

Consider the top-most path made up of only horizontal steps from p_1 to p_1. It represents the trajectory of a token that stays in place p_1. Consider the non-horizontal fourth step p_1p_3 of the second top-most trajectory. It corresponds to a token in p_1 observing a token in p_3 and moving to p_3 by transition t_3. The observed token in p_3 is represented by the horizontal fourth step of the either the last or next-to-last trajectory. The history is realized by the firing sequence $t_3t_1t_1t_3t_2t_4$.

Figure 3.5b shows a non-realizable history of the IO net of Figure 3.1. It consists of four trajectories of length three. The first step contradicts realizability because there are two trajectories with non equal non-horizontal steps: the second top-most trajectory takes step p_1p_2 and the third top-most trajectory takes step p_1p_3. The third step also contradicts realizability: the top-most trajectory takes step p_1p_2 but there is no horizontal third step p_op_o such that $p_1 \xrightarrow{p_o} p_2$ is a transition of the net.

Bunches and Pruning Lemma. A *bunch* is a multiset of trajectories with the same length and the same initial and final place. The Pruning Lemma states that every realizable history containing a bunch of trajectories from p to p' of size larger than the number of places n can be "pruned", meaning that the bunch can be replaced by a smaller one, also leading from p to p', while keeping the history realizable (however, the smaller bunch cannot always be chosen as a sub-multiset of the original one).

Intuitively, if a bunch B contains many trajectories, we can replace it by a new bunch of trajectories constructed from it that keep the rest of the history realizable. The trajectories of the bunch may contain horizontal steps enabling non-horizontal steps in the rest of the history, where the horizontal steps correspond to tokens in the observed place of a transition t, and the non-horizontal steps correspond to tokens going from the source to the destination place of t. The idea is to "fix" one token in each observed place of a transition realizing H. We use the trajectories of B to build new trajectories that "fix" each such token in an observed place at the earliest moment possible, and then "unfix" each such token at the last moment possible – this is enough to guarantee realizability of the rest of the history.

Lemma 11 (Pruning Lemma)**.** Let $\mathcal{N} = (P, \delta)$ be an IO net with n places. Let H be a realizable history of $\mathcal{N}$ containing a bunch $B \subseteq H$ of size larger than n. There exists a bunch B' of size at most n with the same initial and final places as B, such that the history $H' \stackrel{\text{def}}{=} H - B + B'$ (where $+$ and $-$ denote multiset addition and subtraction) is also realizable in $\mathcal{N}$.

Proof. Let P_B be a set of all places visited by at least one trajectory in the bunch B. For every $p \in P_B$ let $f(p)$ and $l(p)$ be the earliest and the latest moment in time when this

place has been used by any of the trajectories (the first and the last occurrence can be in different trajectories). Let $\tau_p, p \in P_B$ be a trajectory that first goes to p by the moment $f(p)$, then waits there until $l(p)$, then goes from p to the final place. To go to and from p it uses fragments of trajectories of B. The portion between is made of horizontal steps stationary in p. We take $B' = \{\tau_p \mid p \in P_B\}$ and prove that the history H' obtained by replacing B with B' in H is still realizable. Note that we can copy the same fragment of a trajectory multiple times.

We build τ_p by taking fragments of existing trajectories and using them at the exact same moments as they are used in H, and by adding some horizontal fragments. Therefore, the set of non-horizontal steps in B' is a subset (if we ignore multiplicity) of the set of non-horizontal steps in B.

Consider any non-horizontal step in H' in any trajectory at position $(i, i+1)$. By construction, the same step at the same position is also present in H. History H is realizable in $\mathcal{N}$, so this step corresponds to a $p_s p_d$ such that $t = p_s \xrightarrow{p_o} p_d \in \delta$ and $M_H^i \xrightarrow{t^k} M_H^{i+1}$ for some $k \geq 1$. Also H contains an enabling horizontal step $p_o p_o$ in some trajectory at that position $(i, i+1)$. There are two cases: either that step $p_o p_o$ was provided by a trajectory in B, or not. In the first case, note that the place p_o of this horizontal step must be first observed no later than i, and last observed not earlier than $i+1$. This implies $f(p_o) \leq i < i+1 \leq l(p_o)$. As H' contains a horizontal step $p_o p_o$ for all positions between $f(p_o)$ and $l(p_o)$, in particular it contains it at position $(i, i+1)$. In the second case the same horizontal step is present in H' as a part of the same trajectory.

Thus H' is realizable. □

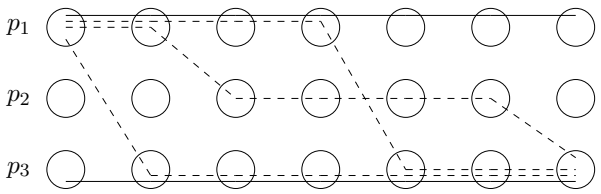


Figure 3.6: *The realizable history of Figure 3.5a after pruning.*

Example 6. The realizable history H of Figure 3.5a, leading from $(5, 0, 1)$ to $(1, 0, 5)$, has a bunch B of size $4 \geq n$ from p_1 to p_3. It visits all places, and $f(p_1) = 1, l(p_1) = 4$, $f(p_2) = 3, l(p_2) = 6$, and $f(p_3) = 2, l(p_3) = 7$. Figure 3.6 shows a history H', leading from $(4, 0, 1)$ to $(1, 0, 4)$, resulting from the application of the Pruning Lemma to H and B. The new bunch B' from p_1 to p_3 given by the Pruning Lemma is drawn in dashed trajectories. Notice that the trajectory of B' that passes through p_2 does not appear in B. The firing sequence $t_3 t_1 t_3 t_4$ realizes H'.

Notice that our definition of realizable histories integrates an important aspect of IO nets, which is that *one* token in the observed place p_o of some transition $p_s \xrightarrow{p_o} p_d$ is enough

(a) *A realizable history.* **(b)** *Boosting of the history on the left.*

Figure 3.7: *A realizable history of the IO net of Figure 3.1 before and after boosting.*

to enable an arbitrary number of tokens to go from place p_s to p_d. In particular, adding a trajectory with a non-horizontal i-th step $p_s p_d$ to a realizable history already containing a non-horizontal i-th step $p_s p_d$ does not alter realizability, since the necessary horizontal i-th step $p_o p_o$ is still there.

Boosting Lemma. We present a Boosting Lemma, which states that duplicating a trajectory of a history of an IO net preserves realizability. Intuitively, duplicating a trajectory corresponds to adding a new "copycat" token that follows another token wherever it goes. This is the formalization of the copycat property informally defined in Section 3.1.

Lemma 12 (Boosting Lemma)**.** Let H be a realizable history of an IO net containing a trajectory τ. The history $H + \wr\tau\wr$ is also realizable.

Proof. Let h be the length of H, and let $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ be a realization of H. For every $1 \leq i \leq h-1$ define k'_i as follows: if $\tau(i) = \tau(i+1)$, then $k'_i \stackrel{\text{def}}{=} k_i$; if $\tau(i) \neq \tau(i+1)$, then $k'_i \stackrel{\text{def}}{=} k_i + 1$. We claim that $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}}$ is a realization of $H + \tau$.

Assume $h = 1$. Then H is realizable by t^0 for any transition t, and so is $H + \tau$. Assume that the induction property holds for some $h \geq 1$, and let H be of length $h+1$, realizable by $t_1^{k_1} \cdots t_h^{k_h}$. By induction, the history $H + \tau$ truncated of its last step is realizable by $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}}$. If $\tau(h) \neq \tau(h+1)$ in H, then since $\tau \in H$ and H is realizable, $\tau(h)\tau(h+1) = p_s p_d$ for p_s and p_d the source and destination places of t_h. Additionally, there are $k_h - 1$ other trajectories τ' such that $\tau'(h)\tau'(h+1) = p_s p_d$, and there is at least one trajectory τ' such that $\tau'(h)\tau'(h+1) = p_o p_o$. Thus $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}} t_h^{k_h+1}$ realizes $H + \tau$. If $\tau(h) = \tau(h+1)$ in H, then $H + \tau$ is realized by $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}} t_h^{k_h}$.

□

Example 7. Figure 3.7b shows a realizable history of the IO net of Figure 3.1. It consists of three trajectories, and the initial and final markings are $(3, 0, 0)$ and $(1, 1, 1)$. The history is realized by the firing sequence $t_1 t_1 t_2$. History H has a trajectory $\tau = p_1 p_2 p_2 p_3$. Figure 3.7b shows a history H', leading from $(4, 0, 0)$ to $(1, 1, 2)$, resulting from the application of the Boosting Lemma to H and τ. The firing sequence $t_1 t_1 t_1 t_2 t_2$ realizes H'. The reader

with an eye for detail may have noticed this is the same example as was used to illustrate the copycat property in Example 4 of Section 3.1.

3.4 Results

Given an initial and a final marking, the Pruning and Boosting Lemmas allow us to isolate a small number of "essential" trajectories needed to go from one to the other. Using these lemmas we prove results on IO firing sequences, culminating in the Generalized Reachability Theorem.

3.4.1 Shortening and Flatness

First, we show a Shortening Theorem: if a marking can reach another, then there exists a firing sequence of bounded *accelerated* length between the two markings. Given a firing sequence $M' \xrightarrow{\sigma} M$, the idea is to take its history, prune it down to have small bunches, then boost up selected trajectories to create a new history entailing a shorter firing sequence from M' to M.

Theorem 13 (IO Shortening)**.** Let $\mathcal{N}$ be an IO net with n places, and let M', M be two markings of $\mathcal{N}$. If $M' \xrightarrow{*} M$, then $M' \xrightarrow{\sigma} M$ for some σ of accelerated length $|\sigma|_a \leq (n^3+1)^n$.

Proof. Let H be a realizable history such that $M' \xrightarrow{H} M$, and let h be the length of H. For every two places p, q, let $B_{p,q}$ denote the bunch of all trajectories of H leading from p to q, and let $T_{p,q}$ be equal to the size of $B_{p,q}$. Applying the Pruning Lemma to all bunches $B_{p,q}$ such that $T_{p,q} \geq n$, we obtain a new realizable history $\widetilde{H}$ satisfying

$$\widetilde{T}_{p,q} = \min\{n, T_{p,q}\} \quad \text{for every } p, q \in P. \tag{3.1}$$

So $\widetilde{H}$ has $\sum_{p,q\in P} \widetilde{T}_{p,q} \leq n^3$ trajectories. Let $M^1_{\widetilde{H}} \xrightarrow{t_1^{k_1}} M^2_{\widetilde{H}} \cdots M^{h-1}_{\widetilde{H}} \xrightarrow{t_{h-1}^{k_{h-1}}} M^h_{\widetilde{H}}$ be a realization of $\widetilde{H}$. Since $\widetilde{H}$ hast at most n^3 trajectories, we have $M^i_{\widetilde{H}}(p) \leq n^3$ for every $p \in P$ and $1 \leq i \leq n$. If $h \geq (n^3+1)^n$, then there are $1 \leq i \neq j \leq h$ such that $M^i_{\widetilde{H}} = M^j_{\widetilde{H}}$, and the history $\widetilde{H'}$ obtained by "cutting out" the fragment of $\widetilde{H}$ between $M^i_{\widetilde{H}}$ and $M^j_{\widetilde{H}}$ is also realizable. (Formally, $\widetilde{H'}$ is the result of replacing every trajectory $\tau \in \widetilde{H}$ by $\tau(1)\cdots\tau(i)\tau(j+1)\cdots\tau(h)$.) So w.l.o.g. we can assume $\widetilde{h} < (n^3+1)^n$.

Since $\widetilde{H}$ is realizable, we have $\widetilde{M'} \xrightarrow{\widetilde{H}} \widetilde{M}$ for some markings $\widetilde{M'}, \widetilde{M}$. We examine the relation between M' and $\widetilde{M'}$, and between M and $\widetilde{M}$. For every place p, the initial (final) number of tokens of p in H is equal to the number of trajectories of H of starting in p

(ending in p), and similarly for $\widetilde{H}$. So we have

$$\begin{array}{llll} M'(p) = \sum_{q\in P} T_{p,q} & \text{and} & M(p) = \sum_{q\in P} T_{q,p} \\ \widetilde{M'}(p) = \sum_{q\in P} \widetilde{T}_{p,q} & \text{and} & \widetilde{M}(p) = \sum_{q\in P} \widetilde{T}_{q,p} \ . \end{array}$$

Further, for every place $p \in P$:

(a) $\widetilde{M'}(p) \leq M'(p)$, and $\widetilde{M}(p) \leq M(p)$.
Follows immediately from $\widetilde{T}_{p,q} \leq T_{p,q}$ for every $q \in P$ (Equation 3.1).

(b) If $\widetilde{M'}(p) = 0$ then $M'(p) = 0$, and if $\widetilde{M}(p) = 0$ then $M(p) = 0$.
If $\widetilde{M'}(p) = 0$ then $\widetilde{T}_{p,q} = 0$ for every $q \in P$. So, by Equation 3.1, $\widetilde{T}_{p,q} = T_{p,q}$ for every $q \in P$, and so $M'(p) = \sum_{q\in P} T_{p,q} = \sum_{q\in P} \widetilde{T}_{p,q} = \widetilde{M'}(p) = 0$. The proof for the target markings is analogous.

Let $\overline{H}$ be the history obtained from $\widetilde{H}$ as follows: for every $p, q \in P$, if $\widetilde{T}_{p,q} > 0$ then pick a trajectory $\tau \in B_{p,q}$, and set $\overline{B}_{p,q} = \widetilde{B}_{p,q} + (\widetilde{T}_{p,q} - T_{p,q} - 1) \cdot \tau$ By the Boosting Lemma, $\overline{H}$ is realizable, and so there are markings $\overline{M'}, \overline{M}$ such that $\overline{M'} \xrightarrow{\overline{H}} \overline{M}$. Further, by (a) and (b) above we have $\overline{T}_{p,q} = T_{p,q}$ for every $p, q \in P$, and so for every $p \in P$:

$$\overline{M'}(p) = \sum_{q\in P} \overline{T}_{p,q} = \sum_{q\in P} T_{p,q} = M'(p)$$

So we get $M' \xrightarrow{\overline{H}} M$. Since $\widetilde{H}$ and $\overline{H}$ have the same length, we get $\overline{h} < (n^3+1)^n$. So every firing sequence realizing $\overline{H}$ has accelerated length at most $(n^3+1)^n$, and we are done. □

A consequence of this Shortening Theorem is that the reachability relation for IO nets is flat. Flatness has been used as a property for reachability in several systems, among which Petri nets [29, 50, 68]. We use the definition of Leroux and Sutre [68]. A net $\mathcal{N} = (P, T, F)$ is *globally flat* if there exist transition words $w_1, w_2, \ldots, w_k \in T^*$ such that for every two markings M', M, if $M' \xrightarrow{*} M$, then there exist $j_1, \ldots, j_k \geq 0$ satisfying $M' \xrightarrow{w_1^{j_1} \ldots w_k^{j_k}} M$. Observe that the words $w_1, w_2, \ldots, w_k$ are independent of both M and M'.

Theorem 14. IO nets are globally flat.

Proof. Let $\mathcal{N} = (P, T, F)$ be an IO net with n places , and let $K = (n^3+1)^n$. By Theorem 13, for every two markings M' and M of $\mathcal{N}$ such that $M' \xrightarrow{*} M$, there is a firing sequence σ of accelerated length at most K such that $M' \xrightarrow{\sigma} M$. Let $T = \{t_1, \ldots, t_m\}$. Every such firing sequence belongs to the regular language $(t_1^* t_2^* \cdots t_m^*)^K$. Let $w_1, w_2, \ldots, w_{m \cdot K} \in T^*$ be the words given by $w_i = t_{((i-1) \bmod m)+1}$ for every $1 \leq i \leq m \cdot K$. These transition words witness that $\mathcal{N}$ is globally flat. □

A net $\mathcal{N} = (P, T, F)$ is *locally post*-flat*[2] if for every M' there exist transition words $w_1, w_2, \ldots, w_k \in T^*$ such that for every M satisfying $M' \xrightarrow{*} M$ there exist $j_1, \ldots, j_k \geq 0$ such that $M' \xrightarrow{w_1^{j_1} \ldots w_k^{j_k}} M$. Here the transition words depend on the choice of source marking M', whereas they are independent in the definition of globally flat. In particular, the fact that IO are globally flat implies that they are locally *post**-flat. A Petri net being locally *post**-flat is equivalent to the reachability set of the net being semilinear [65]. In our case, this means that for any marking M of an IO net, $post^*(M)$ is a semilinear set. This result will be subsumed by the Closure Theorem in our next section, which shows that $post^*(M)$ is in fact a counting set (recall that counting sets are a subclass of semilinear sets).

The fact that IO nets are locally flat allows for analyzing nets by applying existing symbolic model checking tools which use acceleration techniques (e.g. FAST [13], LASH [19], TREX [8]). Intuitively, the tools decide reachability by computing transitive closures of sequences of transitions (the words w_i in our flatness definitions) with the help of symbolic representations. They use semi-decision procedures which may not terminate in general, but are guaranteed to terminate if the system is flat [68].

3.4.2 Closure under Reachability

Our main result for the verification of IO is the existence of algorithms running in polynomial space to decide generalized reachability problems. The crucial result that directly leads to this is that counting sets are closed under reachability operators $post^*$ and pre^*, and that the norms can be bounded polynomially.

Theorem 15 (IO Closure)**.** Let $\mathcal{N} = (P, \delta)$ be an IO net with n places. Let $\mathcal{C}$ be a cube over P. Then $post^*(\mathcal{C})$ is a counting set and

$$\|post^*(\mathcal{C})\| \leq \|\mathcal{C}\| + n^3$$

The same holds for $pre^*(\mathcal{C})$.

We need a preparatory technical lemma before proving our Closure Theorem. Given a cube $\mathcal{C}$ and a marking M' in $post^*(\mathcal{C})$, this lemma constructs a cube $\mathcal{C}'$ containing M' and itself contained in $post^*(\mathcal{C})$, such that the norm of $\mathcal{C}'$ is bounded.

Lemma 16. Let $\mathcal{N} = (P, \delta)$ be an IO net with n places, and let $\mathcal{C}$ be a cube. For all $M' \in post^*(\mathcal{C})$, there exists a cube $\mathcal{C}'$ such that

1. $M' \in \mathcal{C}' \subseteq post^*(\mathcal{C})$, and
2. $\|\mathcal{C}'\| \leq \|\mathcal{C}\| + n^3$.

[2] What we call locally *post**-flat is called locally flat, or just flat, in [68].

Proof. Let L, U be multisets such that $\mathcal{C} = (L, U)$. Let M' be a marking of $post^*(\mathcal{C})$. There exists a marking $M \in \mathcal{C}$ such that $M \xrightarrow{*} M'$, and $M \geq L$ by definition of $\mathcal{C}$. Let H be a history such that $M \xrightarrow{H} M'$. We want to construct a cube $\mathcal{C}'$ satisfying conditions (1) and (2). For this, we choose appropriate lower and upper bounds L', U', and set $\mathcal{C}' = (L', U')$.

Lower bound L'. Let H^L be an arbitrary sub-multiset of H with multiset of initial places L. Let H' denote $H - H^L$. Further, for every $p, p' \in P$, let $H'_{p,p'}$ be the bunch of all trajectories of H' with p and p' as initial and final places, respectively. Let $T'_{p,p'}$ be the number of trajectories in $H'_{p,p'}$. We have

$$H' = \sum_{p,p' \in P} H'_{p,p'}$$

So H' is the union of n^2 (possibly empty) bunches. Applying the Pruning Lemma (Lemma 11) to each bunch of H' with more than n trajectories yields a new history

$$H'' = \sum_{p,p' \in p} H''_{p,p'}$$

where the sum represents multiset addition, and such that history $H'' + H^L$ is realizable, and

$$T''_{p,p'} = \min\{n, T'_{p,p'}\} \quad \text{for every } p, p' \in P. \tag{3.2}$$

Let D and D' be the initial and final markings of $H'' + H^L$. They satisfy the properties:

- $D \xrightarrow{*} D'$, because $H'' + H^L$ is realizable.
- $D \leq M$ and $D' \leq M'$, which follows from $T''_{p,p'} \leq T'_{p,p'}$ for every $p, p' \in P$ (Equation 3.2).
- $D \geq L$, because $H^L \leq H'' + H^L$.
- $|D| = |D'| \leq |L| + n^3$ because $|H'' + H^L| = \sum_{p,p'} T''_{p,p'} + |H^L| \leq n^2 \cdot n + |H^L| = |L| + n^3$.

Since $M \in \mathcal{C}$, we have $U \geq M \geq D \geq L$. So $D \in \mathcal{C}$, and therefore $D' \in post^*(\mathcal{C})$.

Set $L' \stackrel{\text{def}}{=} D'$ as lower bound of our cube $\mathcal{C}'$.

Upper bound U'. Recall H is a history from M to M'. For $q \in P$, define $U'(q)$ as follows:

(i) If some trajectory of H ending in q starts from a place r such that $U(r) = \infty$, then set $U'(q) \stackrel{\text{def}}{=} \infty$.

(ii) If every trajectory of H ending in q starts from a place r such that $U(r) < \infty$, then set $U'(q) = M'(q)$.

We prove that $\mathcal{C}' \stackrel{\text{def}}{=} (L', U')$ satisfies the conditions of the lemma.

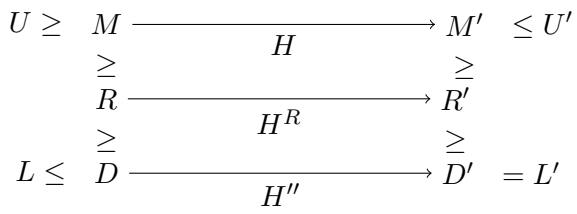


Figure 3.8: *Construction of the proof of Lemma 16*

Property 1: $M' \in \mathcal{C}' \subseteq post^*(\mathcal{C})$.
Since $\mathcal{C}' = (L', U')$, we first prove $L' \leq M' \leq U'$. The inequality $M' \geq L'$ follows from $M' \geq D' = L'$ (see Figure 3.8). Inequality $M'(q) \leq U'(q)$ holds for every place q. If $U'(q) = \infty$ there is nothing to show. If $U'(q)$ is finite, i.e., if (ii) above holds, then $U'(q) = M'(q)$. Thus $M' \in \mathcal{C}'$.

It remains to prove $(L', U') \subseteq post^*(\mathcal{C})$, which requires more effort. We show that for every marking $R' \in (L', U')$ there exists a history H^R leading to R' from a marking $R \in \mathcal{C}$, i.e. satisfying $L \leq R \leq U$. Since $R' \in (L', U')$ and $L' = D'$, we have $R' \geq D'$. We construct H^R by repeatedly applying the Boosting Lemma (Lemma 12) to H'' and some well-chosen trajectories of H''. Since H'' starts in D, this guarantees that H^R starts in a marking R such that $R \geq D \geq L$ (see Figure 3.8). Further, to ensure that H^R ends at R', for every $q \in P$ we boost H'' by exactly $(R'(q) - D'(q))$ trajectories ending at q. It remains to choose these trajectories in such a way that $R \leq U$ holds. We add trajectories so that $R(q) \leq M(q)$ holds, which, since $M(q) \leq U(q)$ (see Figure 3.8), ensures $R(q) \leq U(q)$. We decide which trajectories to add according to two cases, similar to the cases (i) and (ii) above:

(i′) H'' contains a trajectory τ ending in q from a place r such that $U(r) = \infty$.
In this case we add $(R'(q) - D'(q))$ copies of τ.

(ii′) Every trajectory of H'' ending in q from some place r satisfies $U(r) < \infty$.
In this case, by the definition of U' (see (ii) above), we have $U'(q) = M'(q)$. Since $R' \leq U'$ by hypothesis, we get $D'(q) \leq R'(q) \leq M'(q)$, and so $(R'(q) - D'(q)) \leq (M'(q) - D'(q))$. We need to add at most $M'(q) - D'(q)$ trajectories to H''.
For each place $r \in P$, let $T_{r,q}$ and $T''_{r,q}$ be the size of the bunch of trajectories leading from r to q in H and H'' , respectively. By this definition, and the definition of the pruning operation, we have

(a) $M'(q) - D'(q) = \sum_{r \in P} (T_{r,q} - T''_{r,q})$.
(b) For every $r \in P$: $T_{r,q} \geq T''_{r,q}$, and
(c) For every $r \in P$: $T_{r,q} \geq 1$ implies $T''_{r,q} \geq 1$.

We add trajectories as follows: we loop through the places r such that $T_{r,q} \geq 1$. We take any trajectory of H'' leading from r to q (which exists by (c)), and replicate it

$T_{r,q} - T''_{r,q}$ times or less, until the quota of $R'(q) - D'(q)$ trajectories has been reached. The quota is eventually reached by (a). By the Boosting Lemma, the resulting history H^R is still realizable.

We claim that this procedure produces a history H^R such that $T^R_{r,q} \leq T_{r,q}$ for every $r, q \in P$ such that $U(r) < \infty$. Indeed, fix r such that $U(r) < \infty$. If q satisfies (i′), then no trajectory from r to q is replicated, i.e., $T^R_{r,q} = T''_{r,q} \leq T_{r,q}$. If q satisfies (ii′), then our procedure adds at most $T_{r,q} - T''_{r,q}$ trajectories to the bunch of size $T''_{r,q}$ of H'', therefore $T^R_{r,q} \leq T_{r,q}$. This entails that $R(r) \leq M(r)$ for every place r such that $U(r) < \infty$. Since $M \leq U$, we have $R \leq U$, and so we have shown that $R \in \mathcal{C}$.

Property 2: $\|\mathcal{C}'\| \leq \|\mathcal{C}\| + n^3$.
We show $\|\mathcal{C}'\|_l \leq \|\mathcal{C}\| + n^3$ and $\|\mathcal{C}'\|_u \leq \|\mathcal{C}\|$, from which the property is deduced, using $\|\mathcal{C}'\| = \max(\|\mathcal{C}'\|_l, \|\mathcal{C}'\|_u)$. For the l-norm, recall that $L' = D'$. By construction of H'' (see above), we have $|D'| \leq |L| + n^3$, and so

$$\|(L', U')\|_l \leq |L| + n^3 \leq \|(L, U)\| + n^3 \ .$$

For the u-norm, notice that by (i) and (ii), every trajectory of H ending in a place q satisfying $U'(q) < \infty$ starts in a place r satisfying $U(r) < \infty$. Using this remark, we get:

$$\begin{aligned}
& \|(L', U')\|_u & & & & \\
= & \sum_{q|U'(q)<\infty} & U'(q) & & & \\
= & \sum_{q\in P|U'(q)<\infty} & M'(q) & & & \text{(Def. of } U') \\
= & \sum_{q\in P|U'(q)<\infty} & \sum_{r\in P} & T_{r,q} & & \text{(Def. of } T_{r,q}) \\
\leq & \sum_{q\in P} & \sum_{r\in P|U(r)<\infty} & T_{r,q} & & \text{(Remark above)} \\
= & \sum_{r\in P|U(r)<\infty} & \sum_{q\in P} & T_{r,q} & & \text{(Algebra)} \\
= & \sum_{r\in P|U(r)<\infty} & M(r) & & & (H \text{ starts in } M) \\
\leq & \sum_{r\in P|U(r)<\infty} & U(r) & & & (M \leq U) \\
= & \|(L, U)\|_u & & & &
\end{aligned}$$

□

Example 8. Let $\mathcal{C} = (L, U)$ be a cube of the IO net of Figure 3.1, with $L = (0, 0, 0)$ and $U = (6, 3, \infty)$. Let M' be the marking $(1, 0, 5)$. It is in $post^*(\mathcal{C})$ because it is reachable

from $M = (5, 0, 1) \in \mathcal{C}$, as witnessed by the realizable history H of Figure 3.5a. Following the proof of Lemma 16, we construct cube $\mathcal{C}' = (L', U')$ such that $M' \in \mathcal{C}' \subseteq post^*(\mathcal{C})$ and $\|\mathcal{C}'\| \leq \|\mathcal{C}\| + n^3 = 9 + 3^3 = 36$. History H is pruned down to the history from $(4, 0, 1)$ to $(1, 0, 4)$ illustrated in Figure 3.6, in which no bunch has more than $n = 3$ trajectories. We fix $L' = (1, 0, 4)$. The only trajectory of H ending in p_1 starts in p_1, and $U(p_1) < \infty$ so $U'(p_1) = M'(p_1) = 1$. No trajectory ends in p_2, so $U'(p_2) = M'(p_2) = 0$. There is a trajectory from p_3 to p_3 and $U(p_3) = \infty$, so $U'(p_3) = \infty$ and we have $U' = (1, 0, \infty)$. The norm of $\mathcal{C}'$ is 5, and $\mathcal{C}'$ is indeed included in $post^*(\mathcal{C})$: every marking in $\mathcal{C}'$ is of the form $(1, 0, 4 + k)$ for $k \geq 0$, and it is reachable from marking $(4, 0, 1 + k) \in \mathcal{C}$ by firing $t_3t_1t_3t_4$.

The proof of the Closure Theorem follows from the above lemma and the fact that there are only a finite number of cubes, given bounds on their norms.

Proof of the Closure Theorem (Theorem 15). By Lemma 16, for every marking $M' \in post^*(\mathcal{C})$ there is a cube $\mathcal{C}_{M'}$ such that $M' \in \mathcal{C}_{M'}$, $\mathcal{C}_{M'} \subseteq post^*(\mathcal{C})$, and $\|\mathcal{C}_{M'}\| \leq \|\mathcal{C}\| + n^3$. So $post^*(\mathcal{C}) = \bigcup_{M' \in post^*(\mathcal{C})} \mathcal{C}_{M'}$. Given a bound on their lower and upper norms, there are only finitely many cubes $\mathcal{C}_{M'}$. Therefore $post^*(\mathcal{C}) = \bigcup_{i=1}^{k} \mathcal{C}_i$ for some k, and so it is a counting set as a finite union of cubes. Since $\|\mathcal{C}_i\| \leq \|\mathcal{C}\| + n^3$ for every $1 \leq i \leq k$, by definition of counting set norms we have $\|post^*(\mathcal{C})\| \leq \|\mathcal{C}\| + n^3$.

The result for $pre^*(\mathcal{C})$ is proved by considering the net $\mathcal{N}'$ obtained by reversing the arcs of our given IO net $\mathcal{N}$. This is still an IO net. A marking of $\mathcal{N}$ is a marking of $\mathcal{N}'$ and vice versa, and it is easy to see that $M \xrightarrow{*} M'$ in $\mathcal{N}$ if and only if $M' \xrightarrow{*} M$ in $\mathcal{N}'$ for any markings M, M'. Thus the set of markings $pre^*_{\mathcal{N}}(\mathcal{C})$ is equal to the set of markings $post^*_{\mathcal{N}'}(\mathcal{C})$. We get that $pre^*_{\mathcal{N}}(\mathcal{C})$ is a counting set and $\|pre^*_{\mathcal{N}}(\mathcal{C})\| \leq \|\mathcal{C}\| + n^3$, by application of the above to $\mathcal{N}'$ and $post^*_{\mathcal{N}'}(\mathcal{C})$. □

The above result also holds for counting sets. Indeed recall that counting sets can be represented as finite unions of cubes, and that the norm of a counting set is the maximum of the norms of these cubes. With the fact that counting sets are closed under boolean operations, we obtain the following closure result.

Corollary 17. Counting sets of IO nets are closed under $post^*$, pre^* and boolean operations.

Notice that given a non-empty cube $\mathcal{C} = (L, U)$, the marking equal to $L(p)$ for each place p is in $\mathcal{C}$. Using this and the Closure Theorem, we deduce the existence of a polynomial size witness for reachability between two counting sets.

Lemma 18. Let $\mathcal{N}$ be an IO net with n places. Let $\mathcal{S}', \mathcal{S}$ be two counting sets. If $\mathcal{S}$ is reachable from $\mathcal{S}'$, then there exist $C' \in \mathcal{S}', C \in \mathcal{S}$ such that $C' \xrightarrow{*} C$ and $|C'| = |C| \leq \|\mathcal{S}'\| + \|\mathcal{S}\| + n^3$.

Proof. If $\mathcal{S}$ is reachable from $\mathcal{S}'$, then the set of markings described by $\mathcal{S} \cap post^*(\mathcal{S}')$ is non-empty. By the Closure Theorem (Theorem 15) and Proposition 1, $\mathcal{S} \cap post^*(\mathcal{S}')$ is a counting set of norm at most $\|\mathcal{S}\| + \|\mathcal{S}'\| + n^3$. Let $\cup_i \mathcal{C}_i$ be a counting constraint for $\mathcal{S} \cap post^*(\mathcal{S}')$ whose norm is less than $\|\mathcal{S} \cap post^*(\mathcal{S}')\|$. Let $\mathcal{C} = (L, U)$ a cube in $\cup_i \mathcal{C}_i$. Marking C equal to L on all places is in $\mathcal{C}$. By definition of the norm, $|C| \leq \|\mathcal{S} \cap post^*(\mathcal{S}')\|$. Thus $|C| \leq \|\mathcal{S}'\| + \|\mathcal{S}\| + n^3$. Since $C \in \mathcal{S} \cap post^*(\mathcal{S}')$, there exists $C' \in \mathcal{S}'$ such that $C' \xrightarrow{*} C$ and we are done. □

3.4.3 IO Generalized Reachability Theorem

In the Closure Theorem we have shown that, given a cube $\mathcal{C}$, $post^*(\mathcal{C})$ and $pre^*(\mathcal{C})$ are counting sets of bounded size. Using this, we show our main result: generalized reachability problems can be solved in polynomial space. That is, membership or emptiness for any combination of atoms using boolean operations, pre^* and $post^*$ can be evaluated in polynomial space, where an atom is a counting set. The intuition behind this Generalized Reachability Theorem is that the norms of the counting sets obtained by such combinations are "small", and so we only need to examine small markings to verify them, thus yielding an algorithm for checking correctness which runs in polynomial space .

Recall the notion of generalized reachability expressions from Section 2.4, expressions which are constructed using boolean combinations, $post^*$ and pre^* over counting constraints. By Corollary 17, any generalized reachability expression E is a counting constraint, and the set of markings $[E]$ it represents is a counting set. Recall that the generalized reachability problems are the generalized reachability membership problem and the generalized reachability emptiness problem.

Theorem 19 (IO Generalized Reachability Theorem)**.** Let $\mathcal{N}$ be an IO net with n places. Let E be a generalized reachability expression of length $|E|$, and let N be the maximum norm of the counting constraints appearing in E. Then

- $[E]$ is a counting set of norm $O(|E| \cdot N \cdot n^{|E|})$,
- the generalized reachability membership problem for IO nets is in PSPACE, and
- the generalized reachability emptiness problem for IO nets is in PSPACE.

Proof. Set $[E]$ is a counting set, by Corollary 17. We denote by $\|E\|$ the norm of $[E]$. The bounds for the norms follow from Proposition 1 and Theorem 15. We proceed by structural induction on E. If E is a counting constraint Γ, then $\|E\| \leq \|\Gamma\| \leq N \in O(|E| \cdot N \cdot n^{|E|})$. Let E_1 and E_2 be generalized reachability expressions with norm $O(|E_1| \cdot N \cdot n^{|E_1|})$ and $O(|E_2| \cdot N \cdot n^{|E_2|})$ respectively. If $E = E_1 \triangle E_2$ with $\triangle \in \{\cup, \cap\}$, then its norm is smaller or equal to $\|E_1\| + \|E_2\|$ and $|E| = |E_1| + |E_2|$, so $\|E\| \in O(|E| \cdot N \cdot n^{|E|})$. If $E = \overline{E_1}$, then $\|E\| \leq n\|E_1\| + n$ and $|E| = |E_1| + 1$, so $\|E\| \in O(|E| \cdot N \cdot n^{|E|})$. Finally, if $E = post^*(E_1)$ or $E = pre^*(E_1)$, then $\|E\| \leq \|E_1\| + n^3$ and $|E| = |E_1| + 1$, so $\|E\| \in O(|E| \cdot N \cdot n^{|E|})$.

We consider now the generalized reachability membership problem, where the input is $\mathcal{N}$, E and a marking M. We again proceed by structural induction on E. If E is a counting constraint, then it is a finite union of cubes, and membership of M in $[E]$ is done by checking if $M(p)$ is in the bounds of one of these cubes for each place p. If membership in E_1 and E_2 can be checked in PSPACE for some generalized reachability expressions E_1, E_2, then it is easy to see that membership in $E_1 \cup E_2$, in $E_1 \cap E_2$ and in $\overline{E_1}$ is also in PSPACE. (This may require storing the input of the subsidiary PSPACE checks, which can be done in polynomial space.) We show that membership in $post^*([E_1])$ is in PSPACE, where E_1 is a generalized reachability expression for which membership is in PSPACE and $[E_1]$ is a counting set with norm $O(|E_1| \cdot N \cdot n^{|E_1|})$. The case for membership in $pre^*([E_1])$ is symmetrical.

By Savitch's Theorem, NPSPACE=PSPACE, so it is enough to provide a nondeterministic algorithm. The algorithm first guesses a marking $M_0 \in [E_1]$ of the same size as M. It verifies that M_0 belongs to $[E_1]$ by using the membership algorithm running in polynomial space that we have by assumption. It then guesses a firing sequence starting at M_0, step by step. The algorithm accepts if the marking reached at some step is M. Notice that all intermediate markings have the same size as M. At any moment in time the algorithm only stores three markings, the current one, the next marking in the sequence, and the input M, which can be done in polynomial space. This concludes the discussion regarding the membership complexity.

To see that checking emptiness of E is in PSPACE, notice that if E is nonempty, then it has an element of size at most $\|E\|$. We can guess such an element M in polynomial space, and verify that M is indeed in E by means of the above membership algorithm. Storing M encoded in binary requires space at most polynomial in $|E|$, $\log(N)$ and n, by our bound on $\|E\|$.

Notice that for both our algorithms, the encoding of the input marking M and of the counting constraints of expression E (whose bounds are smaller or equal to N) can be in unary or in binary, and the algorithms still run in polynomial space. □

This result is a powerful tool which can be used to prove that a host of problems are in PSPACE for IO. We expose some of them below. For instance, the cube-reachability problem for cubes $\mathcal{C}$ and $\mathcal{C}'$ is just checking if $post^*(\mathcal{C}) \cap \mathcal{C}'$ is empty.

Theorem 20. The cube-reachability and cube-coverability problems for IO nets are in PSPACE.

Proof. Let us first consider cube-reachability. Cube $\mathcal{C}'$ can reach $\mathcal{C}$ if and only if the generalized reachability expression $\mathcal{C} \cap post^*(\mathcal{C}')$ is non empty. This can be checked in PSPACE by Theorem 19.

Now for cube-coverability. Cube $\mathcal{C}'$ can cover $\mathcal{C}$ if and only if $\mathcal{C}'$ can reach $\mathcal{C}_\infty$, where $\mathcal{C}_\infty$ is the cube with same lower bounds as $\mathcal{C}$ but upper bound ∞ on every place. As above, this can be checked in PSPACE by Theorem 19. □

Notice that cube-reachability and coverability can be extended to counting set-reachability and coverability by virtue of a counting set being a finite union of cubes.

Recall that a marking M_0 of an IO net N is *live* if for every marking M reachable from M_0 and for every transition t of N, some marking reachable from M enables t. The *cube-liveness* problem consists of deciding if, given a net N and a cube $\mathcal{M}$ of markings of N, every marking of $\mathcal{M}$ is live.

Theorem 21. The cube-liveness problem for IO nets is in PSPACE.

Proof. Let $\mathcal{N}$ be an IO net with set of places P, and $\mathcal{C}$ a cube. Let $t = p_s \xrightarrow{p_o} p_d$ be a transition of $\mathcal{N}$. The set $En(t)$ of markings that enable t contains the markings that put at least one token in p_s and at least one token in p_o (unless $p_s = p_o$ in which case there should be at least two tokens in that place). Clearly, $En(t)$ is a cube. Then $\overline{pre^*(En(t))}$ is the set of markings M from which one cannot execute transition t anymore by any firing sequence starting in M. So the set $\mathcal{L}$ of live markings of $\mathcal{N}$ is given by

$$\mathcal{L} = \overline{pre^*\left(\bigcup_{t \in T} \overline{pre^*(En(t))}\right)}$$

Deciding whether $\mathcal{C} \subseteq \mathcal{L}$ is equivalent to deciding whether $\mathcal{C} \cap \overline{\mathcal{L}} = \emptyset$ holds. This can be checked in PSPACE by Theorem 19. □

The *structural liveness* problem for Petri nets is: given a net $\mathcal{N}$, does there exist a marking M such that M is live? A PSPACE upper bound can be shown by using Theorem 19 and the counting set of live markings $\mathcal{L}$ from the previous proof. Let $\mathcal{C}$ be the cube representing all markings, i.e. with lower bound 0, and upper bound ∞ on all places. The answer to the structural liveness problem is yes if and only if $\mathcal{C} \cap \mathcal{L}$ is non-empty. This can be checked in polynomial space by Theorem 19. The authors of [84] give an alternative proof for the PSPACE upper bound, showing that if an IO net has a live marking, then it has one such that there are at most two tokens in each place.

The Generalized Reachability Theorem also has consequences for population protocols, which is the subject of the next chapter.

3.5 Summary and Discussion

We defined immediate observation (IO) nets, a class of Petri nets defined syntactically, in which the only possible form of synchronization is observation, and no process creation nor

destruction is possible. IO nets are included in the class of conservative Petri nets, for which classic problems like reachability, coverability and liveness are PSPACE-complete [37]. By reduction from the acceptance problem for bounded-tape Turing machines in Section 3.2, we showed that these problems are still PSPACE-hard (and thus complete) for IO nets. We showed that cube-parameterized versions of reachability and coverability for conservative Petri nets have the same complexity as reachability and coverability for general Petri nets. This is not the case for IO nets, for which we showed that the complexity is still PSPACE-complete, like in the single marking case. In fact, this PSPACE upper bound for IO nets holds for any generalized reachability problem, as shown in our Generalized Reachability Theorem. To prove it, we used techniques described in Section 3.3, de-anonymizing token trajectories, and looking at histories instead of firing sequences. These techniques lead to additional results, like the IO Shortening Theorem stating that if there exists a firing sequence between two markings then there exists one with few transition alternations, and the fact that IO nets are globally flat. We proved the IO Closure Theorem, which states that counting sets (finite unions of cubes) are closed under forward reachability ($post^*$) and backward reachability (pre^*), and that the size of the reachability sets is polynomial in the initial counting set. This result helps to get a clear picture of the expressivity of IO nets, and it leads directly to our Generalized Reachability Theorem.

Flatness for Petri nets [50, 68] is a notion which expresses that the reachability set is somehow simple. The fact that IO nets are (globally and thus locally) flat paves the way for the use of symbolic model checkers with acceleration techniques in practical analysis cases. These tools, like FAST [13], LASH [19] and TREX [8], use semi-decision procedures. They work well in practice but may not terminate. However, it is shown in [68] that they are guaranteed to terminate if the system is flat. It would be interesting to investigate how the algorithms of these symbolic model checkers can be tailored to the case of IO nets to produce efficient verification techniques, using the results on the reachability of IO nets provided in this chapter. As mentioned in the introduction, IO nets model networks of enzymatic chemical reactions. Properties of these sometimes translate as properties of the reachability graph [4], providing a setting in which to apply the above tools.

In [84] the authors study the structural liveness problem for IO nets. They prove that the problem is PSPACE-complete, adapting the bounded-tape Turing machine simulation for the lower bound, and showing that an IO net is structurally live if and only if it is live from a marking with at most 2 tokens in any place. The result is elegant and the proof uses classic Petri net theory tools like siphons. For future work, it may be interesting to look more at the interaction of IO nets with tools such as siphons, traps or place/transition-invariants.

Finally, another perspective offered by this chapter for future work is the use of token de-anonymization and histories. We are surely not the first to use this approach, but we can spread its use, as it provides a good angle for studying systems, like Petri nets, in

which the tokens or processes can be seen as agents (and not as different entities at each moment).

Sources. IO nets were introduced in [47] as a Petri net model inspired by immediate observation population protocols (IO protocols, see next Chapter). The lower bounds, the Pruning and Boosting Lemmas, the IO Closure Theorem as well as the upper bounds for cube-reachability, cube-coverability and cube-liveness were published in [47]. The Shortening and Flatness Theorems come from [77]. The Generalized Reachability Theorem appears in [46](Lemma 6.3), though in a different form.

An earlier paper [44] considers the model of IO population protocols, a model close to IO nets, and studies a problem called the correctness problem. Using a Rackoff style argument [74], it gives a doubly-exponential bound on the norms of $post^*$ and pre^* of a counting set. This leads to an EXPSPACE upper bound on the correctness problem, in the same way that the Closure Theorem leads to the Generalized Reachability Problem (using an algorithm which guesses configurations of size doubly-exponential, which are encoded in exponential space). The paper also contains a PSPACE-hardness proof for the correctness problem by reduction from the acceptance problem for bounded-tape Turing machines. The correctness problem for IO protocols can be expressed as a generalized reachability problem, and the results of [44] are subsumed by the later results of [47]. However, [44] can be considered as the first paper of the line of research presented in this thesis.

4 Application to Population Protocols

Population protocols were introduced by Angluin *et al.* as a model of distributed computation [6], and since then they have been extensively studied [1–3, 30, 36, 40]. They were originally introduced to model mobile sensor networks with limited computational capacity, and they also find motivation in the field of chemical reaction networks [83]. The model postulates a "soup" of finite-state, indistinguishable agents interacting in pairs. A protocol has a set of initial configurations. Intuitively, each initial configuration corresponds to an input, and the purpose of a protocol is to compute a boolean output, 0 or 1, for each input. A protocol outputs b for a given initial configuration C if in all fair runs starting at C (with respect to a certain fairness condition), all agents eventually agree to output b. So, loosely speaking, population protocols compute by reaching a stable consensus. The *predicate* computed by a protocol is the function that assigns to each initial configuration C the boolean output computed by the protocol when started at C.

Petri nets can model population protocols, and in the last years, this connection was exploited to address the problem of proving population protocols correct. The fundamental *correctness* problem for population protocols asks, given a protocol and a predicate, whether the protocol computes the predicate. This question was proved decidable in [41, 42], but, unfortunately, the same papers also showed that the correctness problem is at least as hard as Petri net reachability, and so of non-primitive recursive complexity [31].

In their seminal paper on the expressive power of population protocols [7], Angluin *et al.* defined subclasses corresponding to different communication primitives between agents. In particular, they introduced *immediate observation protocols*, called IO protocols for short. We show that these are modelled by IO nets, and that the correctness problem can be expressed as a generalized reachability problem.

Section 4.1 gives definitions for population protocols and IO protocols, and describes their connection to Petri nets. Section 4.2 focuses on solving the correctness problem for IO protocols, applying the results of the previous chapter.

4.1 Primer on Population Protocols

Definition and connection to Petri nets. A *population protocol* consists of a set of states Q, a set of transitions $T \subseteq Q^2 \times Q^2$, a set of *input* states $I \subseteq Q$ and an *output* function $O\colon Q \to \{0,1\}$ assigning a boolean value to each state. A transition $((q_1,q_2),(q_3,q_4)) \in T$ is denoted $(q_1,q_2) \mapsto (q_3,q_4)$. A *configuration* is a multiset over the states Q. Given two configurations C and C' we say that there is a *step* $C \xrightarrow{t} C'$ if there

exists $t = (q_1, q_2) \mapsto (q_3, q_4) \in T$, if $C(q) \geq \boldsymbol{q_1} + \boldsymbol{q_2}$, and if $C' = C - \boldsymbol{q_1} - \boldsymbol{q_2} + \boldsymbol{q_3} + \boldsymbol{q_4}$. A *run* is a sequence of steps.

The connection to Petri nets is immediate: the Petri net modelling a protocol has one place for each state, and one transition for every transition of the protocol. If transition t of the Petri net models $(q_1, q_2) \mapsto (q_3, q_4)$, then ${}^\bullet t = \wr q_1, q_2 \wr$, and $t^\bullet = \wr q_3, q_4 \wr$. An agent in state q is modeled by a token in place q. A configuration C with $C(q)$ agents in state q is modeled by the marking putting $C(q)$ tokens in place q for every $q \in Q$. A run in the protocol corresponds to a firing sequence in the Petri net and vice versa. Observe that the transitions of the net each have a preset and a postset of equal size, so we have:

Fact 22. Petri nets obtained from population protocols are conservative.

Computation. Population protocols are designed to compute predicates $\varphi\colon \mathbb{N}^k \to \{0,1\}$. We first give an informal explanation of how a protocol computes a predicate, and then a formal definition using Petri net terminology. A protocol for φ has a k input states $\{q_1, q_2, \ldots, q_k\} \subseteq Q$. Assume for example $k = 2$. In order to compute $\varphi(n_1, n_2)$, we first place n_i agents in q_i for $j = 1, 2$, and zero agents elsewhere. This is the *initial configuration* of the protocol for the input (n_1, n_2). Then we let the protocol run. The protocol satisfies that in every *fair* run starting at the initial configuration (fair is defined formally below), eventually all agents reach states labeled with 1 by the output function, and stay in such states forever, or they reach states labeled with 0 by the output function, and stay in such states forever. So, intuitively, in all fair runs all agents eventually "agree" on a boolean value. By definition, this value is the result of the computation, i.e, the value of $\varphi(n_1, n_2)$.

Formally, and in Petri net terms, fix a Petri net $\mathcal{N} = (P, T, F)$ with $|{}^\bullet t| = 2 = |t^\bullet|$ for every transition t. Further, fix a set $I = \{p_1, \ldots, p_k\}$ of *input places*, and a *function* $O\colon P \to \{0,1\}$. We identify the tuple $(\mathcal{N}, I, O)$ and the population protocol it models. A marking M is a b-*consensus* if $M(p) > 0$ implies $O(p) = b$. A b-consensus M is *stable* if every marking reachable from M is also a b-consensus. A firing sequence $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \cdots$ of $\mathcal{N}$ is *fair* if it is finite and ends at a deadlock marking, or if it is infinite and the following condition holds for all markings M, M' and $t \in T$: if $M \xrightarrow{t} M'$ and $M = M_i$ for infinitely many $j \geq 0$, then $M_j \xrightarrow{t_{j+1}} M_{j+1} = M \xrightarrow{t} M'$ for infinitely many $j \geq 0$. In other words, if a fair sequence reaches a marking infinitely often, then all the transitions enabled at that marking will be fired infinitely often from that marking. A fair firing sequence *converges to* b if there is $j \geq 0$ such that M_j is a b-consensus for every marking $j \geq i$ of the sequence. For every $\boldsymbol{v} \in \mathbb{N}^k$ with $|\boldsymbol{v}| \geq 2$ let $M_{\boldsymbol{v}}$ be the marking given by $M_{\boldsymbol{v}}(p_i) = \boldsymbol{v}_i$ for every $p_i \in I$, and $M_{\boldsymbol{v}}(p) = 0$ for every $p \in P \setminus I$. We call $M_{\boldsymbol{v}}$ the *initial marking for input* $\boldsymbol{v}$. We say $(\mathcal{N}, I, O)$ *computes the predicate* $\varphi\colon \mathbb{N}^k \to \{0,1\}$ if for every $\boldsymbol{v} \in \mathbb{N}^k$, every fair firing sequence starting at $M_{\boldsymbol{v}}$ converges to b.

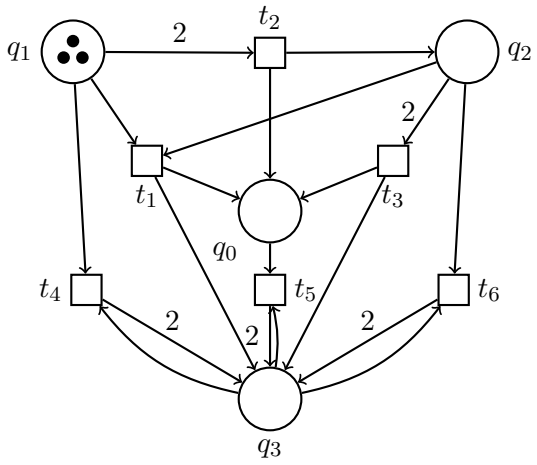


Figure 4.1: *Petri net underlying population protocol $\mathcal{P}_2$.*

Example 9. We exhibit two population protocols that compute the predicate $\varphi(x) \stackrel{\text{def}}{=} [x \geq 3]$, and their corresponding Petri nets.

The Petri net $\mathcal{N}_1$ for the first protocol $\mathcal{P}_1$ is shown in Figure 3.1. Protocol $\mathcal{P}_1$ has states $P = \{p_1, p_2, p_3\}$ and transitions $(p_a, p_a) \mapsto (p_{a+1}, p_a)$ and $(p_a, p_3) \mapsto (p_3, p_3)$ for $a = 1, 2$. The only input state is p_1. States p_1 and p_2 are labeled with 0, and state p_3 with 1. The initial marking in $\mathcal{N}_1$ for input x puts x tokens on the place p_1 corresponding to state p_1, and no token elsewhere. If $x \geq 3$, then every fair firing sequence of $\mathcal{N}_1$ eventually reaches the deadlock marking with x tokens in p_3 and no tokens elsewhere (indeed, transitions t_3 and t_4 ensure that after a token reaches p_3, eventually all other tokens move to p_3 as well). So the tokens eventually reach consensus 1. If $x < 3$, then no firing sequence ever puts a token in q_3 and so, since both p_1 and p_2 have output 0, the tokens reach consensus 0.

The Petri net $\mathcal{N}_2$ for the second protocol $\mathcal{P}_2$ is shown in Figure 4.1. Protocol $\mathcal{P}_2$ has state set $Q = \{q_0, q_1, q_2, q_3\}$, and transitions $(q_a, q_b) \mapsto (q_0, q_{\min(a+b,3)})$ for $0 < a, b < 3$, and $(q_a, q_3) \mapsto (q_3, q_3)$ for $0 \leq a < 3$. Again, the only input state is q_1. States q_0, q_1, q_2 are labeled with 0, and state q_3 is labeled with 1. The intuition is that a token is in q_i if it "knows" that there are at least i tokens in the net, for $i \in \{0, \ldots, 3\}$. For example, the initial marking $\wr 3 \cdot q_1 \wr$ of $\mathcal{N}_2$ shown in Figure 4.1 has three tokens which each "know" that there is at least one token in the net (themself). By firing t_2, one of the tokens can "transfer" its information to another, and forget its own: one token moves to q_2 with the information that there are at least two tokens, and the other token moves to q_0 with the (non-)information that there are at least zero tokens. As in the first protocol, the tokens eventually reach consensus 1 from an input x if and only if $x \geq 3$. If $x \geq 3$, then in every fair firing sequence of $\mathcal{N}_2$ a token will eventually reach place q_3. Transitions t_4, t_5 and t_6 then ensure that all tokens move to q_3, which has output 1, and stay there. If $x < 3$, then no firing sequence can put a token in q_3 and so all tokens stay in places with output 0.

Both these protocols could be generalized to compute $[x \geq n]$ for any natural $n \geq 1$.

Immediate observation protocols. When two agents of a population protocol communicate, they can both simultaneously change their states. This corresponds to communication by *rendez-vous*. In [7], Angluin *et al.* introduced *immediate observation* protocols, corresponding to a more restricted communication mechanism. One of the agents observes the state of the other agent, and updates its own state accordingly; the observed agent does not change its state, since it may not even know that it is being observed. Transitions are of the form $(q_s, q_o) \mapsto (q_d, q_o)$, where q_o is the state of the observed agent.

Example 10. Protocol $\mathcal{P}_1$ of Example 9 is immediate observation, but $\mathcal{P}_2$ is not.

The connection to IO nets is immediate. Transitions $(q_s, q_o) \mapsto (q_d, q_o)$ of the protocol are simply IO transitions of the form $q_s \xrightarrow{q_o} q_d$. Given an IO protocol, the Petri net modelling it is an IO net.

Verifying population protocols. Not every population protocol is well designed. For some inputs $(n_1, \ldots, n_k)$ the protocol can have fair runs that never converge, or fair runs converging to the wrong value $1 - \varphi(n_1, \ldots, n_k)$. This raises the question of how to automatically verify that a protocol correctly computes a predicate. The main difficulty is to prove convergence to the right value *for each of the infinitely many possible inputs*. In the next section, we show how to formalize this as a generalized reachability problem.

Example 11. Consider the population protocol $\mathcal{P}_1$ of Example 9 whose corresponding Petri net is shown in Figure 3.1. Let us add a transition $t_5 = (p_3, p_3) \mapsto (p_1, p_3)$. We do not prove it formally, but it is easy to see that this new protocol has fair runs that never converge. We reason in the corresponding Petri nets. A fair firing sequence can only be finite if it ends at a deadlock, and this was the case for fair firing sequences of $\mathcal{P}_1$ that started with three or more tokens in p_1: they ended in the 1-consensus deadlock marking with all tokens in p_3. In the new protocol, t_5 is enabled at this marking so it is no longer a deadlock, and the marking reached by firing t_5 is not a 1-consensus. A fair firing sequence of the new protocol starting with three or more tokens in p_1 cannot converge to 1, and in fact it also cannot converge to 0.

4.2 Correctness of Immediate Observation Population Protocols

The *correctness* problem for immediate observation protocols asks, given a protocol and a predicate, whether the protocol computes the predicate. In order to study the complexity of the problem we need to restrict ourselves to a class of predicates representable by finite

means. A characterization of the predicates computable by IO protocols was given in [7]: they compute predicates representable by counting constraints (called **COUNT**$_*$ in [7]). The predicates *representable by counting constraints* are the predicates $\varphi\colon \mathbb{N}^k \to \{0,1\}$ for which there is a counting constraint Γ such that $\varphi(\boldsymbol{v}) = 1$ if and only if $\boldsymbol{v}$ is in $[\Gamma]$. So we can formulate the correctness problem as follows: given a counting constraint Γ and an IO protocol with a suitable set of input states, does it compute the predicate represented by Γ?

In Petri net terms, the correctness problem for IO nets asks, given an IO net $\mathcal{N}$, an input set I, an output function O and a counting constraint Γ, whether $(\mathcal{N}, I, O)$ computes Γ (formally defined in Section 4.1). We use the Generalized Reachability Theorem (Theorem 19) to show that the correctness problem for IO nets, and so for IO protocols, is PSPACE-complete.

We present a proposition that characterizes the nets $\mathcal{N}$ that compute a given predicate $\varphi\colon \mathbb{N}^k \to \{0,1\}$. On top of the definitions of Section 4.1, we need some notations. For IO net $\mathcal{N} = (P, T, F)$, input set $I \subseteq P$, output function $O\colon P \to \{0,1\}$, and $b \in \{0,1\}$:

- $\mathcal{I}_b = \{M_{\boldsymbol{v}} \mid \varphi(\boldsymbol{v}) = b\}$, i.e., $\mathcal{I}_1$ ($\mathcal{I}_0$) denotes the initial markings of $\mathcal{N}$ for the input vectors satisfying (not satisfying) φ.
- $\mathcal{C}_b$ denotes the set of b-consensuses of $\mathcal{N}$.
- $\mathcal{ST}_b \stackrel{\text{def}}{=} \overline{pre^*\left(\overline{\mathcal{C}_b}\right)}$ denotes the set of stable consensuses of $\mathcal{N}$ (the complement of the markings from which one can reach a non-b-consensus).

Proposition 23. Let $\mathcal{N}$ be an IO net, let I be a set of input places, let $O\colon P \to \{0,1\}$ be an output function, and let $\varphi : \mathbb{N}^k \to \{0,1\}$ be a predicate where $k = |I|$. Then $(\mathcal{N}, I, O)$ computes φ if and only if $post^*(\mathcal{I}_b) \subseteq pre^*(\mathcal{ST}_b)$ holds for $b \in \{0,1\}$.

Proof. Assume that $post^*(\mathcal{I}_b) \subseteq pre^*(\mathcal{ST}_b)$ holds for $b \in \{0,1\}$. Recall: $(\mathcal{N}, I, O)$ computes φ if for $b = 0, 1$, for every initial marking $M \in \mathcal{I}_b$, every fair firing sequence starting in M converges to b. Let $\pi = M_0, M_1, \ldots$ be a fair firing sequence with $M_0 \in \mathcal{I}_b$ for some $b \in \{0,1\}$. We show that π converges to b.

A marking M of $\mathcal{N}$ is called a *bottom marking* if $M \stackrel{*}{\to} M'$ implies $M' \stackrel{*}{\to} M$ for every marking M'. In other words, M is a bottom marking if it belongs to a bottom strongly connected component (SCC) of the reachability graph of the net. If $M \stackrel{*}{\to} M'$ then $|M| = |M'|$. The set of markings reachable from M is finite for any M, so every fair firing sequence eventually visits a bottom marking. In particular, π contains a bottom marking M of a bottom SCC B. By assumption, we know that $\mathcal{ST}_b$ is reachable from M, so there exists $M' \in \mathcal{ST}_b$ such that $M \stackrel{*}{\to} M'$. This entails $M' \in B$. Since for all $D \in \mathcal{ST}_b$, if $D \stackrel{*}{\to} D'$ then $D' \in \mathcal{ST}_b$, we obtain that $B \subseteq \mathcal{ST}_b$. Every marking of $\mathcal{ST}_b$ is a b-consensus so π converges to b.

Assume that $(\mathcal{N}, I, O)$ computes φ, i.e. that every fair firing sequence starting in $\mathcal{I}_b$ converges to b for $b \in \{0,1\}$. Let us show that $post^*(\mathcal{I}_b) \subseteq pre^*(\mathcal{ST}_b)$ holds. Consider

$M \in post^*(\mathcal{I}_b)$. There exists $M_0 \in \mathcal{I}_b$ such that $M_0 \xrightarrow{*} M$. We show that this finite firing sequence can be extended to a fair infinite firing sequence π.

Let $Conf$ be the set of markings of the net, and let π be a finite firing sequence. Fix an infinite sequence $\rho = M_0, M_1, \ldots$ of markings such that every marking of $Conf$ appears infinitely often in ρ. Define the infinite firing sequence $\pi_0\,\pi_1\,\pi_2\ldots$ and the infinite subsequence $M_{i_0}, M_{i_1}, M_{i_2}\ldots$ of ρ inductively as follows. For $i = 0$, let $\pi_0 := \pi$ and $M_{i_0} := M_0$. For every $j \geq 0$, let $\pi_0 \ldots \pi_j\,\pi_{j+1}$ be any firing sequence leading to the first marking of ρ after M_{i_j} that is reachable from the last marking of $\pi_0 \ldots \pi_j$. It is easy to see that $\pi_0\,\pi_1\,\pi_2\ldots$ is fair.

By the reasoning on bottom markings, our fair firing sequence contains a bottom marking M' of a bottom SCC B. If $B \subseteq \mathcal{ST}_b$ then $M \in pre^*(\mathcal{ST}_b)$ and our proof is done. Suppose this is not the case, i.e. $B \cap \overline{\mathcal{ST}_b} \neq \emptyset$. This means that there is a marking $\hat{M} \notin \mathcal{C}_b$ that is in B. It is thus reachable from any marking of π and so by fairness it is reached infinitely often. Thus π does not converge to b, contradicting the correctness assumption. □

Thus correctness can be solved by checking if $post^*(\mathcal{I}_b) \subseteq pre^*(\mathcal{ST}_b)$ holds for $b \in \{0,1\}$. We show that $\mathcal{I}_b$ and $\mathcal{ST}_b$ are counting sets, allowing us to apply the Generalized Reachability Theorem to solve this inclusion question.

Lemma 24. Let $\mathcal{N} = (P, T, F)$ be an IO net, let $I \subseteq P$ be a set of input places of size k, let $O\colon P \to \{0,1\}$ be an output function, and let $\varphi : \mathbb{N}^k \to \{0,1\}$ be a predicate represented by some counting constraint Γ_φ. Then for $b \in \{0,1\}$, the sets $\mathcal{I}_b, \mathcal{C}_b$ and $\mathcal{ST}_b$ are counting sets with norms polynomial in $|P|, \Gamma_\varphi$.

Proof. The set $\mathcal{C}_b$ is equal to the cube such that there are 0 tokens in states q with $O(q) = 1 - b$ (i.e. an upper and a lower bound of 0), and an arbitrary number of tokens elsewhere (i.e. an upper bound of ∞ and a lower bound of 0). It has norm equal to 0. By Proposition 1 and Theorem 15, $\mathcal{ST}_b = \overline{pre^*\left(\overline{\mathcal{C}_b}\right)}$ is a counting set of norm polynomial in $|P|$. Set $\mathcal{I}_b$ is a counting set described by either Γ_φ or its complement. □

Checking $post^*(\mathcal{I}_b) \subseteq pre^*(\mathcal{ST}_b)$ is equivalent to checking $post^*(\mathcal{I}_b) \cap \overline{pre^*(\mathcal{ST}_b)} = \emptyset$. Since $\mathcal{I}_b$ and $\mathcal{ST}_b$ are counting sets, this is a generalized reachability problem.

Theorem 25. The correctness problem for IO nets is PSPACE-complete.

Proof. Let $\mathcal{N} = (P, T, F)$ be an IO net, let $I \subseteq P$ be a set of input places of size k, let $O\colon P \to \{0,1\}$ be an output function, and let $\varphi : \mathbb{N}^k \to \{0,1\}$ be a predicate represented by some counting constraint Γ_φ. The condition for correctness of Proposition 23 can be rewritten as

$$post^*(\mathcal{I}_b) \cap \overline{pre^*(\mathcal{ST}_b)} = \emptyset. \tag{4.1}$$

By Lemma 24, $\mathcal{I}_b$ and $\mathcal{ST}_b$ are counting sets. Thus $post^*(\mathcal{I}_b) \cap \overline{pre^*(\mathcal{ST}_b)}$ is a generalized reachability expression. Its emptiness can be checked in PSPACE by Theorem 19.

The proof for PSPACE-hardness reduces from the acceptance problem for deterministic Turing machines running in linear space. We prove that the correctness problem is PSPACE-hard, using the construction from the proof of Theorem 10.

Given a Turing machine $\mathcal{M}$ with initial state q_{init} and a size bound, we construct the corresponding IO net $\mathcal{N}_{\mathcal{M}}$ (of Theorem 10) and apply some changes. We restrict $(success, p) \mapsto (success, q)$ transitions to $(success, p) \mapsto (success, success)$ transitions for any place p. We add transitions such that if there are two tokens in the head places, or two tokens in the cell places for the same cell, or two tokens in the *observer* place, one of them can move to *success*. We fix our input places to be the places $off[0, \cdot]$, $at[q_{init}, 1]$ and the *observer* place. We fix our output function to return 1 for *success* and 0 otherwise. We fix the predicate "there are initially at least two tokens in one of the input places".

If the Turing machine accepts the empty tape without going out of bounds, the protocol is not correct, as we can put exactly one token in every input and run the simulation until the acceptance will lead to one of the $(at[q_{acc}, \cdot], observer) \mapsto (at[q_{acc}, \cdot], success)$ transitions firing. Then all the tokens can go to *success*, leading to a 1-consensus deadlock marking even thought the predicate is not true.

Otherwise, if the Turing machine does not accept, the protocol is correct. Initial markings M smaller or equal to the marking M_0 with one token in every input place do not satisfy the predicate. No firing sequence starting in such an M can mark *success* because then by monotonicity, marking M_0 could too. But the *success* place can only be marked from M_0 if the Turing machine can reach the accepting state, and it cannot by assumption. So all fair firing sequences starting in such an M converge to 0, since they only visit 0-consensus markings. The remaining markings are bigger than M_0, which means they put more than one token in one of the input places: they satisfy the predicate. Using our added transitions, fair firing sequences starting in such markings will converge to 1, because eventually they reach the marking with all the tokens in the *success* place, which is a 1-consensus deadlock marking. □

In [7], Angluin et al. show that IO protocols compute exactly the predicates representable by counting constraints. They do so by first showing that given a predicate representable by a counting constraint, there exists an IO protocol computing it; and second, by showing that any correct IO protocol computes a predicate representable by a counting constraint. The first part is done by showing that one can construct a IO protocol computing any "threshold predicate" of the form $\varphi(x) \stackrel{\text{def}}{=} [x \geq k]$ for $k \geq 0$, like in Example 9, and that the class of predicates computable by IO protocols is closed under boolean operations. The second part is involved and uses Higman's Lemma. We can reprove it here in a simple manner using the Closure Theorem.

Lemma 26. Let $\mathcal{P}$ be a IO protocol that computes the predicate $\varphi : \mathbb{N}^k \to \{0, 1\}$. Then φ is representable by counting constraints.

Proof. Let $\mathcal{N}$ be an IO net, let I be a set of input places, and let O be an output function such that $(\mathcal{N}, I, O)$ models $\mathcal{P}$. Recall that φ is representable by counting constraints if there is a counting constraint Γ such that $\varphi(\boldsymbol{v}) = 1$ if and only if $\boldsymbol{v}$ is in the counting set $[\Gamma]$ described by Γ. Let $\mathcal{I}$ be the counting set of initial markings defined by the cube which puts an arbitrary number of tokens in initial states of I, and 0 elsewhere. The set $\mathcal{I} \cap \overline{pre^*(\overline{pre^*(\mathcal{ST}_b)})}$ is the set of initial markings from which all runs of $\mathcal{P}$ converge to b for b in $\{0, 1\}$. Informally: a marking M in this set is initial, and it cannot reach a marking which can reach a marking in $\overline{pre^*(\mathcal{ST}_b)}$, that is a marking which cannot reach a stable b-consensus. In other words, there is no way for M to not be able to reach a stable b-consensus, and so by fairness it eventually will. By Lemma 24 and the Closure Corollary 17, it is a counting set. Since $(\mathcal{N}, I, O)$ computes φ, by definition $\mathcal{I}_b = \mathcal{I} \cap \overline{pre^*(\overline{pre^*(\mathcal{ST}_b)})}$. The set $\{\boldsymbol{v} \mid \varphi(\boldsymbol{v}) = b\}$ is equal to $\mathcal{I}_b$ restricted to the initial states I, and so we are done. □

4.3 Summary and Discussion

In this chapter we recall the formalism of population protocols, introduced in [6]. The main verification problem for these is the correctness problem, which asks if a given protocol computes a given predicate. The authors of [6] introduced in [7] a subclass called immediate observation population protocols. We explain how IO nets model these IO protocols, and we characterize the correctness problem for IO protocols as a generalized reachability problem for IO nets. Using our Generalized Reachability Theorem from the previous chapter, we conclude that IO protocol correctness is in PSPACE. This closes a complexity gap we had opened in [44], and contributed to our article [46] which gives the complexity of the correctness problem for all of the population protocol subclasses originally introduced in [7]. Our results on IO nets from the previous chapter also allow us to reprove a result from [7] in a succinct manner, namely that IO protocols compute predicates definable by counting sets. It is also worth noting that IO nets take their name from IO protocols, because we introduced them in [47] with the goal of modelling IO protocols.

We show that checking the correctness problem for IO protocols is equivalent to checking a generalized reachability problem which is expressed using both $post^*$ and pre^* of counting sets. We then get the complexity result by invoking the Generalized Reachability Theorem. This fully uses the fact that, for IO nets, both the forward and backward reachability set of a counting set is a (small) counting set. This is not the case for BIO nets.

Sources. The result on IO protocol correctness was published in [47]. The result reproving that IO protocols compute counting sets appeared in the appendix of [10] (Lemma 7). A PSPACE lower bound and an EXPSPACE upper bound on the correctness problem appeared in [44], but are subsumed by the results of [47].

5 Branching Immediate Observation Nets

In this chapter we define and study our second class of observation Petri nets, branching immediate observation (BIO) nets. Section 5.1 gives the formal definition and basic properties. Section 5.2 introduces tools and techniques, echoing those for IO nets in Section 3.3, that are then used in Section 5.3 to prove our main results for BIO nets.

5.1 Definition and Examples

Branching immediate observation nets, or *BIO nets*, are a generalization of IO nets and BPP nets. Informally, a BIO net is an IO net in which a token may branch into several tokens upon *observing* the presence of another token.

Definition 27. A transition t of a Petri net is a *branching IO transition* (BIO transition) if there is $k \geq 0$ and places $p_s, p_{d_1}, \ldots, p_{d_k}, p_o$, not necessarily distinct, such that ${}^\bullet t = \wr p_s, p_o \wr$ and $t^\bullet = \wr p_{d_1}, \ldots, p_{d_k}, p_o \wr$, or such that ${}^\bullet t = \wr p_s \wr$ and $t^\bullet = \wr p_{d_1}, \ldots, p_{d_k} \wr$. We call p_s the *source* place, p_o the *observed* place, and the p_{d_i} the *destination* places of t. A Petri net is a *branching IO net* (BIO net) if all its transitions are BIO transitions.

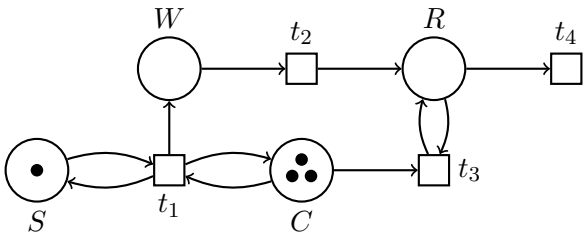


Figure 5.1: *A BIO net.*

Like for IO nets, in the proofs of this chapter we consider only BIO nets in which every transition has an observed place. Given a BIO net in which this is not the case, it suffices to add an extra marked place which acts as observed place for all the transitions without one. In the following, we sometimes denote by $p_s \xrightarrow{p_o} \wr p_1, \ldots, p_k \wr$ the transition t with source place p_s, observed place p_o and $t^\bullet = \wr p_1, \ldots, p_k \wr$. We equate a BIO net (P, T, F) with the pair (P, δ), where δ is the set of tuples $(p_s, p_o, \wr p_1, \ldots, p_k \wr)$ such that $p_s \xrightarrow{p_o} \wr p_1, \ldots, p_k \wr$ is in T.

Example 12. Figure 5.1 shows a BIO net representing a client server interaction. If the server S observes a client C, it creates a worker W, which creates a response R and terminates. The client C "leaves" after observing a response. Responses may expire.

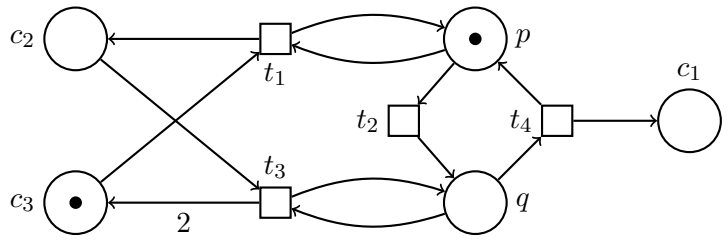


Figure 5.2: *A non-flat BIO net.*

Notice that unlike IO nets, BIO nets are not conservative, and firing transitions can add more tokens to the net. Also unlike IO nets, the reverse of a BIO net – that is, the net in which all arcs are reversed – is no longer a BIO net.

Remark 28. We will also consider branching immediate multiple observation (BIMO) nets, which are like BIO nets except there may be more than one observed place. This case is treated in Chapter 6. The results for BIO nets hold also for BIMO nets, with bounds changing only by a factor corresponding to the maximal number of observations. We treat this case separately to make the proofs of this chapter cleaner.

BIO nets also have the *copycat* property. Consider a firing sequence $M \xrightarrow{*} M'$. We can think of the tokens as each producing a trajectory *tree* by following the firing sequence: a token in a place can branch into a multiset of children following a transition. If we add a token to a place q of M already containing an token, i.e. such that $M(q) \geq 1$, then this new token can "copy" the trajectory tree of the old token, mimicking its transitions along the firing sequence. The intuition is the same as for IO nets: the token in the observed place allowing the transition to fire is not consumed when the transition is fired, so additional tokens can also take the transition. This property is formally defined in Lemma 30 of the next section.

Example 13. We write the markings of the BIO net of Figure 5.1 using the ordering S,W,C,R on the places. Consider the firing sequence: $(1,1,3,0) \xrightarrow{t_2} (1,0,3,1) \xrightarrow{t_3} (1,0,2,1) \xrightarrow{t_4} (1,0,2,0)$. One of the tokens is in place W, then moves to place R, then dissapears. Let us add a token to W in the first marking. It can copy the "trajectory tree" (here non-branching) of the previous token, yielding the firing sequence $(1,2,3,0) \xrightarrow{t_2}\xrightarrow{t_2} (1,0,3,2) \xrightarrow{t_3} (1,0,2,2) \xrightarrow{t_4}\xrightarrow{t_4} (1,0,2,0)$.

The next example shows that BIO nets may have non-semilinear sets of reachable markings. It is taken from Hopcroft and Pansiot's well-known example of a Petri net with a non-semilinear reachability set (Lemma 2.8 of [56]).

Example 14 ([56])**.** Consider the BIO net $\mathcal{N}$ of Figure 5.2, with states p, q, c_1, c_2, c_3 and initial marking $M_0 = (1,0,0,0,1)$. The set $post^*(M_0)$ of markings reachable from M_0 in $\mathcal{N}$

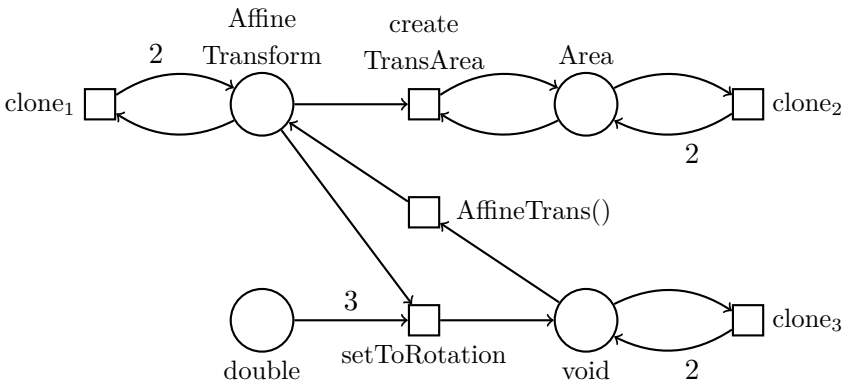


Figure 5.3: *A MIO net from [49].*

is characterized by the condition $(\boldsymbol{p} = 1 \wedge \boldsymbol{q} = 0 \wedge 0 < \boldsymbol{c_2} + \boldsymbol{c_3} \leq 2^{\boldsymbol{c_1}}) \vee (\boldsymbol{p} = 0 \wedge \boldsymbol{q} = 1 \wedge 0 < 2\boldsymbol{c_2} + \boldsymbol{c_3} \leq 2^{\boldsymbol{c_1}+\boldsymbol{1}})$, where $\boldsymbol{c}$ denotes the number of tokens in some place c. Informally, one token cycles between p and q, putting a new token in c_1 at every new cycle. When p is marked, tokens in c_3 can move to c_2, and when q is marked, tokens in c_2 can move to c_3 while doubling their number (see Lemma 2.8 of [56]). It is well known that semilinear sets coincide with the sets of natural numbers definable in Presburger arithmetic [52, 54], i.e. in $\mathsf{FO}(\mathbb{N}, +)$, in which one cannot express an exponential. Clearly $post^*(M_0)$ is not semilinear. And therefore also the reachability relation of this BIO net is not semilinear.

This directly implies that BIO nets do not have a globally flat reachability relation like IO nets, since global flatness is equivalent to semilinearity of the reachability set [65]. However we will show that they are locally flat, for a certain definition of locally flat [68], which still allow us to analyze the nets applying existing symbolic model checking tools.

BIO nets are an extension of IO nets, and they inherit the PSPACE lower bounds for reachability, coverability and liveness. Even though BIO nets are more expressive than IO nets (non-semilinearity), we show that they enjoy many similar properties. If a marking can reach another, it can do so by a "short" firing sequence (this time dependent on the target marking). If a cube can reach another, there exists a polynomial size marking firing sequence witnessing this. Moreover, BIO nets have a Generalized Reachability Theorem stating that any combination of atoms using boolean operations and pre^* (but not $post^*$) can be evaluated in polynomial space, where an atom is a counting set. This entails that, like for IO nets, deciding standard problems (reachability, coverability, liveness) for infinitely many markings is not harder than deciding them for one marking.

The tools to prove all this are presented in Section 5.2, the results are proved in Section 5.3.

Remark 29. One can also define *merging immediate observation* (MIO) nets. These are the Petri nets obtained from BIO nets by reversing all the arcs. Reversing all the arcs in a

BIO net yields a MIO net, and vice versa. Our results for BIO nets imply dual results for MIO nets: any result on pre^* for BIO nets implies the same result on $post^*$ for MIO nets.

Figure 5.3 illustrates (a fragment of) a MIO net taken from the literature on type-driven component-based synthesis [49]. The net represents relationships between API components: the places are types, the transitions are methods and the tokens represent the number of program variables of a certain type. Additionally, there are clone transitions which allow program variables to be reused.

5.2 Branching Histories

Like for IO nets, the results for BIO nets are proved using the idea of "de-anonymizing" tokens. However, since BIO nets can create and destroy tokens, trajectories must be generalized to branching trajectories, which are trees of places; intuitively, the tree captures the cascade of tokens created by a token of the initial marking.

We fix a BIO net $\mathcal{N} = (P, T, F)$ with n places, and let $m_d := \max_{t \in T} |t^\bullet - {}^\bullet t|$ denote the maximum number of tokens created by a transition.

Branching trajectories. A *branching trajectory* of $\mathcal{N}$ is a nonempty, directed tree β whose nodes are labeled with places of P. A node labeled by p is called a *p-node*. The i-th level of β, denoted by $\beta(i)$, is the (possibly empty) set of nodes of β at distance $(i-1)$ from the root. We let $M_\beta(i)$ denote the multiset of places labeling the nodes of $\beta(i)$. Observe that $M_\beta(i)$ is a marking. We say that β has *length* l if $\beta(l) \neq \emptyset$ and $\beta(l+1) = \emptyset$. For example, the first branching trajectory β_c in the history of Figure 5.4 has length 9 and $M_\beta(3) = \wr p, q \wr$.

Histories and realizable histories. A *history* H of length h is a forest of branching trajectories of length at most h. We use histories to describe a behaviour from an initial marking; the history contains a branching trajectory for each token of the initial marking.

Given a history H of length h and an index $1 \leq i \leq h$, the i-th level of H is the set $H(i) = \bigcup_{\beta \in H} \beta(i)$, and the *$i$-th marking of* H, denoted M_H^i, is the multiset $M_H^i = \sum_{\beta \in H} M_\beta(i)$. The markings M_H^1 and M_H^h are called the *initial* and *final* markings of H, and we write $M_H^1 \xrightarrow{H} M_H^h$. If the length of H is longer than the length of its branching trajectories, the final marking of H is the zero marking. Two histories are *equivalent* if they have the same initial and final markings.

A history H of length $h \geq 1$ is *realizable* if there exist transitions $t_1, \ldots, t_{h-1} \in T$ and numbers $k_1, \ldots, k_{h-1} \geq 0$ such that for every $1 \leq i \leq h-1$ the set $H(i)$ can be partitioned into two sets:

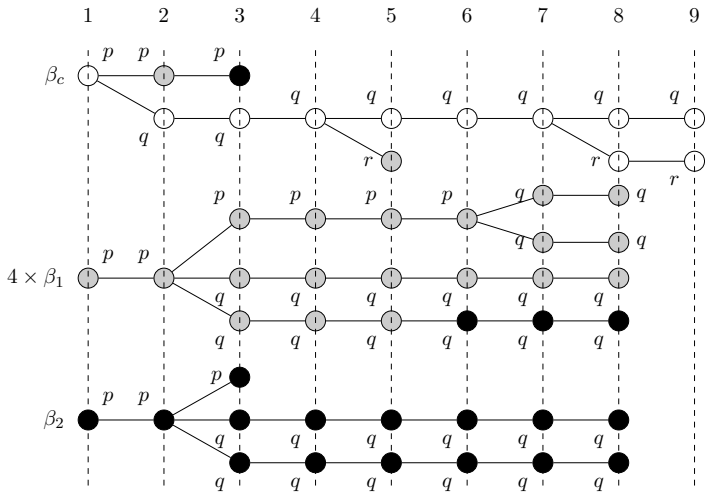


Figure 5.4: *A decorated realizable history of a BIO net.*

- A set $H_a(i)$ containing exactly k_i *active* nodes labeled by the source place of t_i. Given a particular active node, say v, the multiset of labels of its children is the (possibly empty) multiset $\wr p_{d_1}, \ldots, p_{d_k} \wr$ of destination places of t_i.
- A set $H_p(i)$ containing *passive* nodes, each of them with exactly one child, carrying the same label as their parents. This set must contain at least one node labeled by the place p_o observed by t_i.

We say that the sequence $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ *realizes* H. It follows easily from the definitions that $M_H^1 \xrightarrow{t_1^{k_1}} M_H^2 \cdots M_H^{h-1} \xrightarrow{t_{h-1}^{k_{h-1}}} M_H^h$ holds (where $M \xrightarrow{t^0} M'$ iff $M = M'$). From this definition we easily obtain: $M \xrightarrow{*} M'$ if and only if there exists a realizable history with M and M' as initial and final markings.

Like for IO, we define the accelerated length of a firing sequence: Let σ be a firing sequence. let $k_1, \ldots, k_m$ be the unique positive natural numbers such that $\sigma = t_1^{k_1} t_2^{k_2} \ldots t_m^{k_m}$ and $t_i \neq t_{i+1}$ for every $i = 1, \ldots, m-1$. Then σ has accelerated length m, and $|\sigma|_a$ denotes this accelerated length. From the definition of realizable history we immediately obtain: every firing sequence that realizes a history of length h has accelerated length at most h.

Example 15. Figure 5.4 shows a realizable history H of a BIO net with places $\{p, q, r\}$ (for the moment, ignore the shades of gray). H consists of six branching trajectories: β_c, four copies of β_1, and β_2. The history has length 9, and its initial and final markings are $M_H^1 = \wr 6p \wr$ and $M_H^9 = \wr q, r \wr$. The transition t_i executed at step i is

$$t_1 = p \xrightarrow{p} \wr q, p \wr \quad t_2 = p \xrightarrow{p} \wr 2q, p \wr \quad t_3 = p \xrightarrow{q} \emptyset \quad t_4 = q \xrightarrow{q} \wr q, r \wr$$
$$t_5 = r \xrightarrow{p} \emptyset \quad t_6 = p \xrightarrow{q} \wr 2q \wr \quad t_7 = t_4 \quad t_8 = q \xrightarrow{r} \emptyset$$

The firing sequence that realizes H is $t_1\, t_2^5\, t_3^2\, t_4\, t_5\, t_6^4\, t_7\, t_8^{18}$. While the final marking of H is produced by β_c only, β_c is not realizable on its own. For example, the r-node of β_c at level 5 is destroyed in the next step by the firing of t_5, but t_5 can only occur if there is at least one token in place p; this token is supplied by β_1 or β_2. We can think of β_1 and β_2 as branching trajectories that eventually become extinct, but before extinction provide tokens that need to be observable to fire some transitions.

Like for IO nets, our definition of realizable histories integrates the important aspect of BIO nets which is that *one* token in the observed place p_o of some transition $p_s \xrightarrow{p_o} \wr p_{d_1}, \ldots, p_{d_k} \wr$ is enough to enable an arbitrary number of tokens to branch from place p_s to $\wr p_{d_1}, \ldots, p_{d_k} \wr$. In a realizable history, the token in the observed place is represented by a passive node labeled by the observed place, and the tokens in the corresponding source place are represented by active nodes labeled by the source place. We present a Boosting Lemma for BIO nets. It states that duplicating a branching trajectory of a history preserves realizability. It is the formalization of the copycat property informally defined in Section 5.1.

Lemma 30 (BIO Boosting Lemma)**.** Let H be a realizable history of a BIO net containing a branching trajectory β. The history $H + \wr\beta\wr$ is also realizable.

Proof. Let h be the length of H, and let $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ be a realization of H. For every $1 \leq i \leq h-1$ define k'_i as follows: if there are no nodes of $\beta(i)$ in the active nodes $H_a(i)$ of $H(i)$, then $k'_i \stackrel{\text{def}}{=} k_i$; otherwise let b_i be the number of nodes of $\beta(i)$ in the active nodes $H_a(i)$. Define $k'_i \stackrel{\text{def}}{=} k_i + b_i$. We show by induction on h that $H' \stackrel{\text{def}}{=} H + \wr\beta\wr$ is realizable by $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}}$. Assume $h = 1$. Then H is realizable by t^0 for any transition t, and so is H'.

Assume that the induction property holds for some $h \geq 1$, and let H be of length $h+1$, realizable by $t_1^{k_1} \cdots t_h^{k_h}$. By induction, the history H' truncated of its last step is realizable by $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}}$. We partition the nodes of $H'(h)$ into

- a set $H'_a(h)$: it contains the set $H_a(h)$ of active nodes of $H(h)$, as well as a copy of v for each node v of $\beta(h)$ which is in $H_a(h)$.
- a set $H'_p(h)$: it contains the set $H_p(h)$ of passive nodes of $H(h)$, as well as a copy of v for each node v of $\beta(h)$ which is in $H_p(h)$.

Since $H'_p(h)$ contains $H_p(h)$, $H'_p(h)$ has at least one node labeled by the place p_o observed by t_h. Since every v added to $H_a(h)$ to obtain $H'_a(h)$ is a copy of a node already in $H_a(h)$, the multiset of labels of its children is guaranteed to be the multiset $\wr p_{d_1}, \ldots, p_{d_k} \wr$ of destination places of t_h. There are $k_i + b_i$ nodes in $H'_a(h)$ by definition of b_i and realizability of H. Thus H' is realizable, and realized by $t_1^{k'_1} \cdots t_{h-1}^{k'_{h-1}} t_h^{k'_h}$. □

5.2.1 Decorated Histories

To prove our next results, we need to add information to our histories. A *decoration* $\widehat{H}$ of a history H consists of the history H itself and a partition of the nodes of H into *cargo*, *fuel*, and *smoke* nodes. Figure 5.4 shows a history H with a decoration $\widehat{H}$. Graphically, cargo nodes are white, gray nodes are fuel, and black nodes are smoke.

Before giving the formal definition of a decoration, let us provide some intuition. Think of the sequence of markings of a history as the sequence of states of a ship. All nodes of the final marking are cargo, they are what the ship "delivers" in the end. At any other marking, the cargo nodes are the "causal predecessors" of the final cargo nodes. In the history of Figure 5.4, only branching trajectory β_c has nodes at the final level, and thus in the final marking. These nodes labeled q and r must be (white) cargo nodes, and their predecessors in β_c – the nodes with a path to them – must also be cargo nodes.

Every decoration has the same cargo nodes, they only differ in the partition of the other nodes into fuel and smoke. Intuitively, a decoration reserves the right to use fuel nodes to fire transitions (a p-node can be "used" to fire a transition that observes p), and commits to never using a smoke node or its descendants. In the ship analogy, the fuel is "useful" and the smoke is "useless". In the realizable history of Figure 5.4, whenever there is a (black) smoke node labeled by a place, there is also a (gray) fuel node labeled by the same place. This means that if p_o is the observed place of a transition t_i realizing H at step i, then we can always pick a cargo or a fuel p_o-node to be the necessary passive p_o-node of $H(i)$. The most conservative decoration (which always exists) is the one that declares all non-cargo nodes as fuel. Our first goal will be to show that every history has an equivalent *fuel-efficient* history that delivers the same cargo but admits a low-fuel decoration.

Decorations. Formally, a *decoration* $\widehat{H}$ of a history H of length h is a partition of the nodes of H into *cargo*, *fuel*, and *smoke* nodes satisfying the following conditions:

- A node of H is a *cargo node* if and only if it has at least one descendant in $H(h)$.
- All descendants of smoke nodes are smoke nodes.
- For every place p and level i, if $H(i)$ contains smoke p-nodes, then it also contains fuel p-nodes. ("No smoke without fuel". Intuitively, the smoke p-nodes are not needed because the fuel p-nodes can be used instead.)

A *decorated history* is a pair consisting of H and a decoration of H. Observe that along all paths cargo comes before fuel, and fuel before smoke. Graphically, white nodes (if any) come before gray nodes (if any), and gray nodes before black nodes (if any).

5.2.2 Fuel-efficient Histories

We prove that every realizable history has an equivalent realizable history with a *fuel-efficient* decoration, defined as follows.

Definition 31. Let $\widehat{H}$ be a decorated history. A place p is *wasteful at level i* if $\widehat{H}(i)$ contains more than n fuel p-nodes. A place p is *wasteful* in $\widehat{H}$ if it is wasteful at some level; otherwise p is *fuel-efficient* in $\widehat{H}$. Finally, $\widehat{H}$ is *fuel-efficient* if all places are fuel-efficient.

Example 16. Since $n = 3$, in the decorated history of Figure 5.4 place p is wasteful at levels 1 to 6, and q is wasteful at levels 3 to 8. The history is not fuel-efficient.

The proof is based on a Replacement Lemma, which plays a role similar to that of the Pruning Lemma for IO nets. We need a definition.

Definition 32. The (p, i)-*bunch* of H, denoted $B_p(i)$, is the set of subtrees of H rooted at the p-nodes of $H(i)$.

Loosely speaking, the Replacement Lemma shows that if i is the earliest level at which p is wasteful, then the bunch $B_p(i)$ of trajectories can be replaced so that the new history has a decoration where p is not wasteful anymore. The lemma shows how to do this while ensuring that the histories before and after the replacement are equivalent. Repeated applications of the Replacement Lemma yield a fuel-efficient history.

Formally, given a history B'_p with p-nodes as roots and with the same number of trees as $B_p(i)$, we let $H[B'_p/B_p(i)]$ denote the result of replacing each tree of $B_p(i)$ by a different tree of B'_p. For this we assume that $B_p(i)$ and B'_p have been enumerated in some way, and the j-th tree of $B_p(i)$ is replaced by the j-th tree of B'_p. We state the Replacement Lemma:

Lemma 33 (Replacement Lemma)**.** Let $\widehat{H}$ be a decoration of a realizable history H such that p is wasteful, and i is the earliest level at which p is wasteful. There exists a history B'_p such that $H' \stackrel{\text{def}}{=} H[B'_p/B_p(i)]$ is realizable, equivalent to H, and has a decoration whose fuel-efficient places contain all fuel-efficient places of $\widehat{H}$ and p.

Before giving the proof, we present an example.

Example 17. Consider the decorated history $\widehat{H}$ of Figure 5.4. Place p is wasteful at level 1 and $B_p(1) = H$. So all of H is replaced by B'_p whose existence is given by the Replacement Lemma applied to H, p and $i := 1$. Figure 5.5 shows a realizable history $H' = B'_p$ with decoration $\widehat{H'}$. The histories H' and H are equivalent: H' leads from $\wr 6p \int$ to $\wr q, r \int$. It is realized by $t_1\, t_2\, t_3\, t_4\, t_5\, t_6^5\, t_7\, t_8^{12}$. Place p is no longer wasteful in $\widehat{H'}$, and in fact all places are fuel-efficient.

We give ourselves a few more definitions to help in the construction of B'_p. We call smoke and fuel nodes *transportation* nodes. Given a decorated history $\widehat{H}$, let *last*(p) denote the

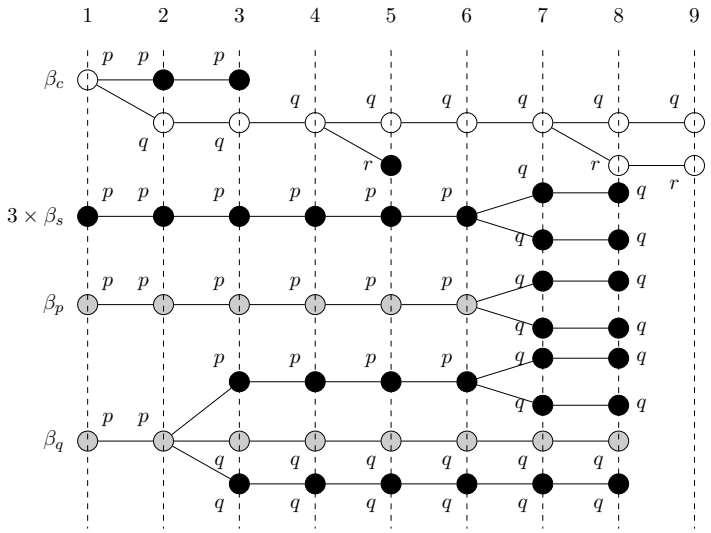


Figure 5.5: *Result of replacing $B_p(1)$ in the history of Figure 5.4.*

last level i such that $\widehat{H}(i)$ contains a transportation p-node. A *place-level* is a pair (q, j), where q is a place and j is a level of H. A *path* of place-levels is a concatenation of "steps" of two types: "doing nothing" steps from (r, l) to $(r, l+1)$, and "transportation history" steps from (r, l) to $(s, l+1)$ such that some transportation r-node of $\widehat{H}(l)$ has an s-child in $\widehat{H}(l+1)$. If H is realizable, this "transportation history" step corresponds to an r-node being active at $H(l)$ and having s in the multiset of labels of its children. We say that (q, j) is *reachable* from (p, i) if there is a path from (p, i) to (q, j), and let $\mathcal{R}_{p,i}$ be the set of all place-levels (q, j) reachable from (p, i).

Before giving the proof proper, we describe an example construction of history B'_p for the decorated history of Figure 5.4 and place p.

Example 18. Let $\widehat{H}$ be the decorated history of Figure 5.4, in which p is already wasteful at level $i = 1$. We construct B'_p, illustrated in Figure 5.5. Recall that since $B_p(1) = H$, we have $H[B'_p/B_p(1)] = B'_p$.

In our example $\widehat{H}$, we have $\mathcal{R}_{p,1} = \{(p,1), \ldots, (p,6), (q,3), \ldots, (q,8)\}$. (Observe that $(r, 5)$ does not belong to $\mathcal{R}_{p,1}$, because its parent is a cargo node.) B'_p is the union of three sets of branching trajectories, B_c, B_f, and B_s (where c, f, s stand for cargo, fuel, and smoke):

- B_c contains all branching trajectories of $B_p(i)$ rooted at a cargo node. (In Figure 5.5, B_c is the singleton set $\{\beta_c\}$.) The decoration conserves the cargo nodes but turns the other nodes to smoke. Intuitively, B_c ensures that H' delivers the same cargo as H.
- B_f contains a branching trajectory β_q for every q such that $(q, j) \in \mathcal{R}_{p,i}$ for some j. (In Figure 5.5, B_f contains the two trees β_p and β_q.) Intuitively, these trajectories guarantee that the new set $\mathcal{R}_{p,i}$ of $\widehat{H'}$ is a superset of the old one, and so that any

transition firing that relies on observing some place q at level j can still occur, because (q, j) is still reachable from (p, i).

Let us now define β_q. (Figure 5.6 shows β_q for the history of Figure 5.5.) Let $\mathit{first}(q)$ be the smallest j such that $(q, j) \in \mathcal{R}_{p,i}$. There is a shortest path from (p, i) to $(q, \mathit{first}(q))$, and each step of the path corresponds to doing nothing or to executing a transition once. (In Figure 5.6 we have $(p, i) = (p, 1)$, $(q, \mathit{first}(q)) = (q, 3)$. The shortest path between them is $(p, 1)(p, 2)(q, 3)$, and it corresponds to doing nothing in the first step then firing t_2.) Let δ_q be the corresponding branching trajectory – a more precise definition is given in the proof, but intuitively it is the branching trajectory produced by either doing nothing or taking a transition following the steps of the shortest path considered. (In Figure 5.6, δ_q is the tree contained in the blue area.) First we append a path to each leaf of δ_q: if the leaf is, say, an r-node at level j, then we append to it a path of r-nodes from level j to level $\mathit{last}(r)$. (Red area of Figure 5.6.) Then, we append to the end of each such path a *r-destroyer*, i.e., a tree that makes the token disappear. We choose for this any subtree of $\widehat{H}$ rooted in a transportation node of $(r, \mathit{last}(r))$. (Green area of Figure 5.6; in order to destroy a p-node we first transform it into two q-nodes by firing t_6, wait while t_7 is fired in another part of the history, and then destroy the q-nodes by firing t_8 twice. The two q-nodes are destroyed by firing t_8 twice.) The decoration of β_q is chosen so that there is a fuel path rooted in (p, i) containing q-nodes from levels $\mathit{first}(q)$ to $\mathit{last}(q)$, and the rest is smoke.

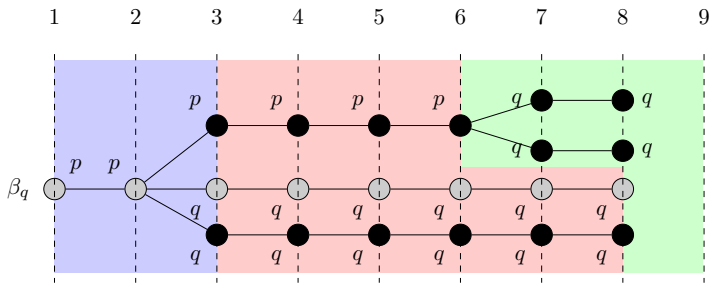


Figure 5.6: *Illustration of the construction of the set B_f of trees.*

- B_s contains $|B_p(i)| - |B_c| - |B_f|$ copies of a tree of smoke nodes β_s consisting of a path of p-nodes of length $\mathit{last}(p) - i + 1$ appended with a p-destroyer. Intuitively, this is smoke added to ensure that $H(i) = H'(i)$.

This concludes the description of B'_p. There are at most $|B_f| \leq n$ fuel nodes per level in B'_p, so p is fuel-efficient.

Proof. (Replacement Lemma.) We first construct $H' \stackrel{\text{def}}{=} H[B'_p/B_p(i)]$ and show that it is realizable and equivalent to H. Then, we define a decoration $\widehat{H'}$ of H', and show that it realizes the condition of the lemma.

Construction of H'. We define B'_p as the union of three sets of branching trajectories, B_c, B_f, and B_s (where c, f, s stand for cargo, fuel, and smoke):

- B_c contains all branching trajectories of $B_p(i)$ rooted at a cargo node.
- B_f contains a branching trajectory β_q for every $q \in \mathcal{R}_{p,i}$.
  We define β_q. Let $\mathit{first}(q)$ be the smallest j such that $(q, j) \in \mathcal{R}_{p,i}$. Notice that $\mathit{first}(q) \leq \mathit{last}(q)$ for all $q \in P$, since by definition of reachability there exists a transportation q-node in level $\mathit{first}(q)$. There is a shortest path ρ_q from (p, i) to $(q, \mathit{first}(q))$. This path induces a branching trajectory δ_q: intuitively it is produced by either doing nothing or taking a transition following the steps of the shortest path considered. We construct δ_q as follows. Let $\rho_q = (p_0, i)(p_1, i+1)\dots(p_k, i+k)$, where $p_0 = p$ and $(p_k, i+k) = (q, \mathit{first}(q))$. The root of δ_q is a p-node v_0. For j from 0 to $k-1$, if $(p_j, i+j)(p_{j+1}, i+j+1)$ is a "doing nothing" step of ρ_q, then add to v_j a child node v_{j+1} labeled by p_{j+1}. If $(p_j, i+j)(p_{j+1}, i+j+1)$ is a "transportation history" step of ρ_q, then there is an active p_j-node in $H(i+j)$ such that the multiset of labels of its children is the multiset $\wr p_{d_1}, \dots, p_{d_k} \int$ of destination places of some t_i with source place p_j and such that p_{j+1} is equal to one of the p_{d_l}. Add to v_j a set of children nodes with multiset of labels $\wr p_{d_1}, \dots, p_{d_k} \int$, and call v_{j+1} one of the nodes labeled by p_{j+1}. This procedure gives us δ_q. We are building a branching trajectory β_q that we will root in level i of H, so intuitively the nodes in level j of β_q will be at level $i+j$ of our new history. We build upon δ_q as follows. First, we append a path to each leaf of δ_q: if the leaf is an r-node at level j of δ_q for some r and j, then we append to it a path of r-nodes from level j to level $\mathit{last}(r) - i$ (thus the last node of this path will be at level $\mathit{last}(r)$ of the new history where β_q is rooted in a node of level i). Then, we append to the end of each such path a *r-destroyer*, i.e., a tree that makes the token disappear. We choose for this any subtree γ_r of $\widehat{H}$ rooted in a transportation node of $(r, \mathit{last}(r))$. This concludes the construction of β_q.
- B_s contains $|B_p(i)| - |B_c| - |B_f|$ copies of a tree β_s consisting of a path of p-nodes of length $\mathit{last}(p) - i + 1$ appended with a p-destroyer γ_p.

We define the replacement $H' = H[B'_p/B_p(i)]$: we replace the trees of $B_p(i)$ with a cargo root in $\widehat{H}$ by the same tree in B_c, we replace some trees of $B_p(i)$ with a fuel root in $\widehat{H}$ by the trees of B_f (in any order), and the rest of the trees of $B_p(i)$ by the trees of B_s. This is well-defined because

- the trees of B'_p all have p-nodes as root,
- there are more than n trees with fuel roots in $B_p(i)$ since p is wasteful at i,

- there are *not* more than n trees in B_f since there is at most one tree per $q \in P$, and
- there are as many trees overall in B'_p as in $B_p(i)$.

History H' is equivalent to history H: the trees added in $B'_p \setminus B_c$ all end in destroyers, and the other trees of H' were already in H, so H' has the same final marking. In case $i = 1$, the number of p-nodes in $H(i)$ and $H'(i)$ is the same so H' has the same initial marking.

History H' is realizable. History H is realizable, and we note $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ a sequence that realizes it, for some transitions $t_1, \ldots, t_{h-1} \in T$ and numbers $k_1, \ldots, k_{h-1} \geq 0$. We show that H' is realizable using the same transitions but different numbers $l_1, \ldots, l_{h-1} \geq 0$. Let $1 \leq j \leq h-1$. Let $H'_p(j)$ be the set of nodes of $H'(j)$ which have exactly one child with the same label, and let $H'_a(j)$ be the rest.

We claim that for every node v' in $H'_a(j)$ with label r and multiset of children labels c, there exists a node v in $H_a(j)$ with label r and multiset of children labels c. By realizability of H this entails that v' is labeled with the source place p_s of t_j, and the multiset of labels of its children is the multiset $\wr p_{d_1}, \ldots, p_{d_k} \int$ of destinations of t_j.

Now to show our claim. Let v' a node of $H'_a(j)$. If v' is not a node of the subtree B'_p, or if v' is a node of B_c, then we are done. Let us assume this is not the case, i.e. $v' \in B_f \cup B_s$.

- If v' is in a tree β_s, then it is in a destroyer (since v' is not in $H'_p(j)$) and so it is in a copy of a subtree of $\widehat{H}$.
- Assume v' is in a tree β_q for some $q \in \mathcal{R}_{p,i}$. If v' is in a destroyer then it is in a copy of a subtree of $\widehat{H}$, and we are done. Otherwise, v' is in the tree δ_q induced by the shortest path ρ_q from (p, i) to $(q, \mathit{first}(q))$ in H. Since v' is not passive, i.e. $v' \notin H'_p(j)$, there exists a "transportation history" step $(r, j)(r', j+1)$ in ρ_q for some $r' \in c$. By definition of δ_q, this means that there is an r-node v active at $H(j)$ with the same multiset c of children labels.

We now show that the set $H'_p(j)$ contains a node labeled by the place p_o observed by t_j. If there is a node labeled p_o in $H_p(j)$ that is not in $B_p(i)$, then it is also in $H'_p(j)$ and we are done. Let us assume that the only nodes of $H_p(j)$ labeled p_o are in $B_p(i)$. If there is a cargo node labeled p_o in $\widehat{H}_p(j)$ then it is also in $H'_p(j)$, so we are done. Otherwise there exists a transportation node v labeled p_o in $\widehat{H}_p(j)$, and $j \leq \mathit{last}(p_o)$ by definition. Since v is in $B_p(i)$, either v is in a tree with a cargo root, or place-level (p_o, j) is reachable from (p, i). If v is in a tree of $B_p(i)$ with a cargo root, it is also in $B_c \subseteq B'_p$. Otherwise $(p_o, j) \in \mathcal{R}_{p,i}$, and therefore by construction there is a node in B'_p labeled p_o at every level between $\mathit{first}(p_o)$ and $\mathit{last}(p_o)$, in particular at j.

Decoration of H'. Let $\widehat{H'}$ be the following decoration of H'. We start with the nodes of B'_p. We define the cargo nodes of B_c to be the cargo nodes of $B_p(i)$ in $\widehat{H}$, and let the rest of the nodes of B_c be smoke. In each tree β_q in B_f, constructed around the tree induced by a

shortest path ρ_q from (p, i) to $(q, \mathit{first}(q))$, we let the nodes along the path ρ_q be fuel nodes, along with the nodes along one branch from $(q, \mathit{first}(q))$ to $(q, \mathit{last}(q))$. All the other nodes of β_q are defined as smoke nodes. We let all the nodes of the trees β_s be smoke nodes.

The nodes of $H' \setminus B'_p$ are decorated in two steps. First, we set $\widehat{H'}$ to be equal to $\widehat{H}$ on the nodes of $H' \setminus B'_p$, which is possible because $H' \setminus B'_p = H \setminus B_p(i)$. Then, we do the following "re-decoration". Let (q, j) be a place level reachable from (p, i) in H'. If there are any fuel nodes labeled q in $(H' \setminus B_f)(j)$, redecorate them and all their descendants as smoke nodes in $\widehat{H'}$. Do this for every (q, j) reachable from (p, i). Notice that this re-decoration only affects nodes of $H' \setminus B'_p$, as the only fuel nodes of B'_p are in B_f. We now show that this decoration is well-defined.

The decoration is well-defined. The cargo nodes in $\widehat{H'}$ are well defined, as the cargo nodes of $\widehat{H'}$ are the cargo nodes of $\widehat{H}$. The order of "cargo then fuel then smoke" is respected along the branching trajectories of $\widehat{H'}$ because they are respected in B'_p, by construction. We now show that the "no smoke without fuel" property holds.

First remark that, for any place p, the last level index $\mathit{last}(p)$ at which there is a transportation p-node in $\widehat{H'}$ is equal to $\mathit{last}(p)$ in $\widehat{H}$ by construction. Let v be a smoke q-node at level $\widehat{H'}(j)$, for some q and j. If (q, j) is reachable from (p, i) in H then there exists a fuel q-node in $\widehat{H'}(j)$ provided by β_q, since $j \leq \mathit{last}(q)$ by virtue of v being smoke. If (q, j) is not reachable from (p, i) in H, then there is no subtree of $B_p(i)$ rooted in a transportation p-node with a descendant labeled q. Therefore in $\widehat{H'}$, node v is not in B_f. Since it is also not in B_s, whose trees are only p nodes until $\mathit{last}(p)$, v is in either a tree of B_c or in no tree of B'_p, and therefore v exists also in $\widehat{H}$ as a smoke node. Since the smoke/fuel partition of $\widehat{H}$ is well defined, there exists a fuel q-node v' in $\widehat{H}(j)$. Since (q, j) is not reachable from (p, i) in H, v' is either in B_c or not part of $B_p(i)$ and so v' is also in $\widehat{H'}(j)$.

Fuel-efficient places. For every place-level (q, j) in H' reachable from (p, i), there are at most n fuel q-nodes in $\widehat{H'}(j)$. Indeed, by definition, the only fuel nodes labeled q in $\widehat{H'}(j)$ are in $B_f(j)$. By definition of $B_f(j)$, the only fuel nodes labeled q in $B_f(j)$ are in the trees β_r for some $r \in \mathcal{R}_{p,i}$. There are a most n such trees, and in each tree there is at most one fuel node per level. Therefore there are no wasteful places q at some level j such that (q, j) is reachable from (p, i) in H'. In particular, p is fuel-efficient since i is the earliest level at which p is wasteful in H. If there is a wasteful place-level in $\widehat{H'}$, then it is unreachable from (p, i) in H'. By definition of H', this means that it is also a wasteful place-level in $H \setminus B'_p$ and thus in H. Thus the fuel-efficient places of $\widehat{H'}$ contain all the fuel-efficient places of $\widehat{H}$, as well as the place p. □

Notice that repeated applications of the Replacement Lemma to some history H yield the existence of a fuel-efficient decoration $\widehat{H'}$ of a history H' equivalent to H.

5.2.3 Smoke Irrelevance and Unique Footprint

Our next results for BIO histories are the Smoke Irrelevance Lemma and the Unique Footprint Lemma. Intuitively, the Smoke Irrelevance Lemma shows that we can always deliver the same cargo using the same fuel *independently* of the initial amount of smoke. We use this to prove our second lemma: the Unique Footprint Lemma shows that for every history there exists an equivalent history in which any two levels differ in the cargo, the fuel, or the *support* of the smoke. This allows us to bound the length of histories for given initial and a final markings.

Let $\widehat{H}_c(i)$, $\widehat{H}_f(i)$, $\widehat{H}_s(i)$ denote the multisets of cargo, fuel, and smoke nodes of $\widehat{H}(i)$, for $\widehat{H}$ a decorated history and i a level of H.

Lemma 34 (Smoke Irrelevance Lemma)**.** Let $\widehat{H}$ be a realizable decorated history of length h, and let μ be any multiset of places such that $\|\mu\| \subseteq \|\widehat{H}_s(1)\|$. There exists a realizable decorated history $\widehat{H'}$ of length h such that $\widehat{H'}_s(1) = \mu$ and, for every level $1 \leq i \leq h$, $\widehat{H'}_c(i) = \widehat{H}_c(i)$ and $\widehat{H'}_f(i) = \widehat{H}_f(i)$.

We present an example before giving the proof.

Example 19. Consider the realizable decorated history $\widehat{H}$ of Figure 5.5. The smoke nodes of $\widehat{H}_s(1)$ are the roots of the three β_s trees, and their multiset of labels is $\wr 3 \cdot p \wr$. Let $\mu = \wr p \wr$. We have $\|\mu\| \subseteq \|\widehat{H}_s(1)\| = \{p\}$. Let $\widehat{H'}$ be the decorated history equal to $\widehat{H}$ with two of the three trees β_s removed. It is such that $\widehat{H'}_s(1) = \mu$ and it is realized by $t_1\, t_2\, t_3\, t_4\, t_5\, t_6^3\, t_7\, t_8^8$. At each level i from 1 to 9, it has the same cargo and fuel nodes as $\widehat{H}$.

Notice that the new history given by the Smoke Irrelevance Lemma has the same final marking, since the final level is only cargo nodes.

Proof. Rename $\nu := \widehat{H}_s(1)$ for clarity. To construct $\widehat{H'}$, start with $\widehat{H}$, and do the following for every place $p \in P$. If $\mu(p) \leq \nu(p)$, then delete $\nu(p) - \mu(p)$ smoke p-nodes from $\widehat{H}(1)$ as well as all their descendants (which are all smoke nodes by definition). If $\mu(p) > \nu(p)$, then add to $\widehat{H}$ $(\mu(p) - \nu(p))$ copies of an arbitrary tree β of smoke nodes of $\widehat{H}$ rooted in $(p, 1)$. This tree exists because $p \in \|\mu\|$, and so $p \in \|\nu\|$. The addition of the copies of β maintains the "no smoke without fuel" property, because it was already fulfilled in $\widehat{H}$ by the nodes of β. The smoke nodes of $\widehat{H'}(1)$ thus constructed are labelled by the places of μ, and fuel and cargo nodes are neither added nor removed.

We prove that $\widehat{H'}$ is realizable. Let $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ be a sequence that realizes $\widehat{H}$, for some transitions $t_1, \ldots, t_{h-1} \in T$ and numbers $k_1, \ldots, k_{h-1} \geq 0$. Removing trees of smoke nodes from $\widehat{H'}$ does not affect realizability: if there is a smoke p_o-node with a p_o-node child in some level $\widehat{H}(i)$ where p_o is the observed place of t_i, then there is also a fuel p_o-node in $\widehat{H}(i)$ with a p_o-node child by the "no smoke without fuel" property. These fuel nodes are still in $\widehat{H'}$ because we only remove trees of smoke nodes. Removing the trees translates

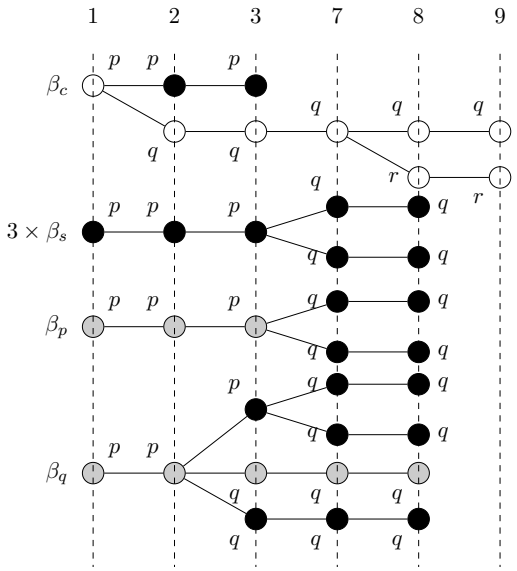


Figure 5.7: *Result of splicing out levels between 3 and 6 in the history of Figure 5.5.*

as decreasing the iterations of some transitions in the realizing sequence of $\widehat{H}$. The trees of smoke nodes that we add to $\widehat{H}'$ also do not affect realizability: they only increase the iterations of the transitions in the realizing sequence, as in the proof of the Replacement Theorem. □

We use the lemma to show that for every history there exists an equivalent decorated history in which any two levels differ in the cargo, the fuel, or the *support* of the smoke.

Definition 35. Given a level $\widehat{H}(i)$ of a decorated history, define its *footprint* as the triple $(\widehat{H}_c(i), \widehat{H}_f(i), \|\widehat{H}_s(i)\|)$ (that is, we only take the support of $\widehat{H}_s(i)$, not $\widehat{H}_s(i)$ itself).

Lemma 36 (Unique Footprint Lemma)**.** Every realizable history has an equivalent fuel-efficient realizable decorated history in which every level has a different footprint.

Before giving the proof, we provide an example which describes the proof idea.

Example 20. Consider the realizable decorated history $\widehat{H}$ in Figure 5.5. It is of length nine and goes from $\wr 6p \wr$ to $\wr q, r \wr$. Level 3 and 6 have the same footprint: $\widehat{H}_c(i) = \wr q \wr$, $\widehat{H}_f(i) = \wr p, q \wr$ and $\|\widehat{H}_s(i)\| = \{p\}$ for $i = 3, 6$. Applying the Smoke Irrelevance Lemma to $\mu := \widehat{H}_s(3)$ and to the history starting in $\widehat{H}(6)$ yields the existence of a realizable decorated history $\widehat{H'}$ of length four such that $\widehat{H'}(1) = \widehat{H}(3)$ and such that the final marking is $\wr q, r \wr$. We construct a new realizable decorated history by taking the first two levels of $\widehat{H}$ and appending $\widehat{H'}$. The new history is represented in Figure 5.7. It is equivalent to H, it is realized by $t_1\, t_2\, t_6^5\, t_7\, t_8^{12}$ and it has a unique footprint at each level.

Proof. Let H be a realizable history, and let $\widehat{H}$ be a decoration of H (one always exists, for example the conservative decoration that declares all non-cargo nodes as fuel). By the Replacement Lemma, we can assume w.l.o.g. that $\widehat{H}$ is fuel-efficient. Assume further that $\widehat{H}$ has minimal length h, i.e., every equivalent decorated history that is also fuel-efficient has length at least h. We claim that every level of $\widehat{H}$ has a different footprint. Assume this is not the case. Then there exist two indices $1 \leq i < j \leq h$ such that $(\widehat{H}_c(i), \widehat{H}_f(i), \|\widehat{H}_s(i)\|) = (\widehat{H}_c(j), \widehat{H}_f(j), \|\widehat{H}_s(j)\|)$. The truncated history $\widehat{H}(j)\widehat{H}(j+1)\dots\widehat{H}(h)$ is clearly realizable. Since $\|\widehat{H}_s(i)\| = \|\widehat{H}_s(j)\|$, we can apply the Smoke Irrelevance Lemma with $\mu := \widehat{H}_s(i)$ and obtain a decorated history $\widehat{H'}$ of length $h - j + 1$ such that $(\widehat{H}_c(i), \widehat{H}_f(i), \widehat{H}_s(i)) = (\widehat{H'}_c(1), \widehat{H'}_f(1), \widehat{H'}_s(1))$ (notice: now $\widehat{H}_s(i) = \widehat{H'}_s(1)$, instead of only $\|\widehat{H}_s(i)\| = \|\widehat{H'}_s(1)\|$). But this implies $\widehat{H}(i) = \widehat{H'}(1)$, and so the concatenation $H(1)\cdots H(i-1)H'(1)\cdots H'(h-j+1)$ is also a realizable history. By the Smoke Irrelevance Lemma we have $\widehat{H'}_c(h-j+1) = \widehat{H}_c(h)$. Since the last levels of a decorated history only contain cargo nodes, this implies $\widehat{H'}(h-j+1) = \widehat{H}(h)$, and so the concatenation is equivalent to H. Further, since $\widehat{H'}$ has the same cargo and fuel nodes as $\widehat{H}(j)\widehat{H}(j+1)\dots\widehat{H}(h)$, the concatenation is also fuel-efficient, contradicting that $\widehat{H}$ has minimal length. □

5.3 Results

Given an initial and a final marking, the set of tools of the previous section show us that only a small number of trajectory trees are "essential" to go from one to the other. We use this to derive our main results for BIO nets. In Section 5.3.1 we show BIO versions of the Shortening Theorem and flatness results for IO nets. In Section 5.3.2 we show a BIO Closure Theorem from which we derive our main result on generalized reachability problems for BIO, presented in Section 5.3.3.

5.3.1 Shortening and Local Flatness

Like for IO nets, we show a Shortening Theorem for BIO nets: if a marking can reach another, then there exists a firing sequence of bounded *accelerated* length between the two markings. Unlike for IO nets, this bound depends not only on the size of the net, but also on the final (target) marking. Example 14 of Section 5.1 illustrates that for BIO nets the bound cannot be independent of both the initial and final marking.

Example 21. Recall the BIO net of Example 14 with states p, q, c_1, c_2, c_3. It is easy to see that for $j \geq 1$ the marking $M_j \stackrel{\text{def}}{=} (1, 0, j, 0, 2^j)$ is reachable only via the firing sequence

$$(t_1t_2t_3t_4)(t_1^2t_2t_3^2t_4)\dots(t_1^it_2t_3^it_4)\dots(t_1^jt_2t_3^jt_4).$$

This sequence has accelerated length $4j$, which depends on the target marking M_j.

The BIO Shortening Theorem has an added feature: it bounds the size of the markings along the shorter firing sequence. In IO the token number stays constant along a firing sequence, but this is not the case for BIO.

Theorem 37 (BIO Shortening)**.** Let $\mathcal{N}$ be a BIO net with n places, let M', M be two markings of $\mathcal{N}$, and let $|M'| = m'$, $|M| = m$. Let $m_d := \max_{t \in T} |t^\bullet - {}^\bullet t|$ denote the maximum number of tokens created by a transition of $\mathcal{N}$. If $M' \xrightarrow{*} M$, then $M' \xrightarrow{\sigma} M$ for some σ of accelerated length $|\sigma|_a \leq 2^n(m+1)^n(n+1)^n$. Further, the intermediate markings along σ have size at most $(m' + 2^n(m+1)^n(n+1)^n(m+n)m_d)m_d^n$.

Given a history for firing sequence $M' \xrightarrow{\sigma} M$, the proof idea for the Shortening Theorem is to apply the Unique Footprint Lemma to get a short history, from which we also derive the token bound.

Proof. We first prove the bound on the accelerated length. By the Unique Footprint Lemma, there is a history H such that $M' \xrightarrow{H} M$ and H has a decoration $\widehat{H}$ where every level has a different footprint. So the length of $\widehat{H}$ is bounded by the number of possible footprints of the histories leading from M' to M. Since, by definition, the number of cargo nodes cannot decrease from a level to the next, and the last level consists of only cargo, every level has between 0 and m cargo nodes per place. Since $\widehat{H}$ is fuel-efficient, every level has between 0 and n fuel nodes per place. Finally, there are at most 2^n possible supports in a net with n places. So the number of footprints, and so the length of $\widehat{H}$, and the accelerated length of any firing sequence realizing $\widehat{H}$, is at most $2^n(m+1)^n(n+1)^n$.

Let us now prove the token bound. To bound the number of smoke nodes in each level, we apply the following operation. Replace every largest tree of smoke nodes (since the children of smoke nodes are smoke, this means trees rooted at smoke nodes whose parents are cargo or fuel) by the tree β_s defined as in the Replacement Lemma: β_s is a path of smoke p-nodes ending at level $last(p)$, appended by a p-destroyer tree. This maintains realizability, because (by the "no smoke without fuel" property in $\widehat{H}$), it does not decrease the support of the multiset of places of any level. We call $\widehat{H'}$ the resulting realizable history with decorated nodes. Note that the "no smoke without fuel" property may not hold in $\widehat{H'}$, so it is not formally a decorated history, but it is sufficient to conclude the proof. $\widehat{H'}$ has the following property: smoke p-nodes can only create other nodes (which, by definition, are also smoke) at the level $last(p)$, and it can create at most m_d of them.

At all other levels j of $\widehat{H'}$, only cargo and fuel nodes can create nodes. There are at most $h' \leq 2^n(m+1)^n(n+1)^n$ levels, and at most $(m+n)$ cargo and fuel nodes per place. Each transition has a unique source place, and all the nodes are added to the initial m' nodes corresponding to the tokens of M'. Thus there are at most $m' + h'(m+n)m_d$ nodes at the first level $last(p)$ in which a smoke node creates nodes. At most all of the nodes are smoke, so at most $(m' + h'(m+n)m_d)m_d$ nodes are created. There are at most n levels $last(p)$,

which each create at most the total amount of nodes times m_d nodes. Thus at every level of the history there are at most $(m' + h'(m+n)m_d)m_d^n$ nodes, concluding the proof. □

Analogously to IO nets, this Shortening Theorem entails that the reachability relation for BIO nets is flat, at least in a local sense. In Section 3.4.1, we gave the definition of a locally *post**-flat Petri net. We say a Petri net is locally *pre**-flat if and only if its reverse net, i.e. the net with reverse arcs, is locally *post**-flat.

Theorem 38. BIO nets are locally *pre**-flat, but neither globally flat nor locally *post**-flat.

Proof. (a) We show that BIO nets are locally *pre**-flat. Let $\mathcal{N} = (P, T, F)$ be a BIO net with n places and $T = \{t_1, \ldots, t_l\}$, let M be a marking of $\mathcal{N}$ with $|M| = m$, and let $K = 2^n(m+1)^n(n+1)^n$. By Theorem 37, for every marking M' of $\mathcal{N}$ there is a firing sequence $t_{i_1}^{j_1} \cdots t_{i_K}^{j_K}$ leading from M' to M. Since every such sequence belongs to the regular language $(t_1^* t_2^* \cdots t_l^*)^K$, the words $w_1, w_2, \ldots, w_{l \cdot K}$ given by $w_i = t_{((i-1) \bmod l)+1}$ for every $1 \leq i \leq l \cdot K$ witness that $\mathcal{N}$ is locally *pre**-flat.

(b) We show that BIO nets are not locally *post**-flat, and so also not globally flat. Consider the BIO net of Figure 5.2 with states p, q, c_1, c_2, c_3. Recall that for all $j \geq 1$, M_0 only reaches the marking $M_j \stackrel{\text{def}}{=} (1, 0, j, 0, 2^j)$ via $(t_1 t_2 t_3 t_4)(t_1^2 t_2 t_3^2 t_4) \ldots (t_1^i t_2 t_3^i t_4) \ldots (t_1^j t_2 t_3^j t_4)$. So in order to reach M_j it is necessary to fire j times a sequence of the form $t_1^k t_2^{k_2} t_3^k t_4^{k_4}$, which proves the result. □

The Shortening Theorem for BIO nets can also be used to prove that reachability between single markings is in PSPACE: using the bound on the number of tokens in the intermediary markings of a firing sequence, we only need to check if there exists a firing sequence between markings of bounded size. But in the next sections, we show that there is a Closure Theorem and thus a Generalized Reachability Theorem for BIO nets that subsumes this result in a stronger result on generalized reachability expressions.

5.3.2 Closure under Backwards Reachability

Recall that for an IO net with n places and a counting set $\mathcal{S}$, we have that $post^*(\mathcal{S})$ is also a counting set and $\|post^*(\mathcal{S})\| \leq \|\mathcal{S}\| + n^3$. The same holds for $pre^*(\mathcal{S})$. This result can be partially extended to BIO nets. This cannot be the case for both *pre** and *post**: a counting set is a semilinear set, and we have already seen that BIO nets may have a marking M_0 such that $post^*(M_0)$ is non-semilinear (Example 14). However, we show that *pre** of a counting set is a counting set with polynomially bounded norm.

First, and using the tools from Section 5.2, we show that *pre** of a singleton marking is a counting set with polynomially bounded norm. Essentially we show that token counts above $|M| + n$ are not distinguishable from the point of view of reachability of a marking M.

Lemma 39. Let $\mathcal{N} = (P, T, F)$ be a BIO net with n places. Let M be a marking of $\mathcal{N}$. Then $pre^*(M)$ is a counting set and

$$\|pre^*(M)\| \leq n|M| + n^2.$$

Proof. Let M' be a marking in $pre^*(M)$. There exists $\widehat{H}$, a realizable decorated history from M' to M. By the Replacement Lemma (Lemma 33), we can assume w.l.o.g. that $\widehat{H}$ is fuel-efficient. Given a place p, if $M'(p) \geq |M| + n + 1$, then there exists a smoke p-node in $\widehat{H}(1)$. Indeed, there are at most $|M|$ cargo nodes at any level of $\widehat{H}$, and at most n fuel p-nodes by fuel-efficiency. By the Smoke Irrelevance Lemma (Lemma 34), we can decrease or increase the number of smoke p-nodes in M' by any amount and still reach M. That is, for any $k \geq 0$, the marking equal to $\widehat{H}_f(1)(p) + \widehat{H}_c(1)(p) + k$ on p and equal to M' everywhere else is in $pre^*(M)$. We deduce that $pre^*(M)$ can be represented by a counting constraint as a finite union of cubes with bounds on places p of the form $a \leq p \leq a$ for $a \in \{0, \ldots, |M| + n\}$ or $|M| + n \leq p \leq \infty$. Thus $pre^*(M)$ is a counting set, and by definition of the norm it has norm smaller or equal to n times $|M| + n$. □

We extend this result to pre^* of a cube. In the proof, we add token-destroying transitions to the net in order to reduce the problem to single-marking (backwards) reachability.

Theorem 40 (BIO Closure)**.** Let $\mathcal{N} = (P, T, F)$ be a BIO net with n places. Let $\mathcal{C}$ be a cube over P. Then $pre^*(\mathcal{C})$ is a counting set and

$$\|pre^*(\mathcal{C})\| \leq n\|\mathcal{C}\| + n^2$$

Proof. Every cube $\mathcal{C}$ can be decomposed into a finite union of *simple cubes*: cubes with bounds on each place p of the form $b_p \leq p \leq b_p$ or $b_p \leq p \leq \infty$ for some $b_p \in \mathbb{N}$ such that $b_p \leq \|\mathcal{C}\|$.

Let $\mathcal{C}_s$ be a simple cube. For each place p with upper bound ∞, we add a transition t with preset ${}^\bullet t = \{p\}$ and postset $t^\bullet = \emptyset$. It is easy to see that for every marking M', the modified net $\mathcal{N}'$ has a firing sequence from M' to the lower bound L of $\mathcal{C}_s$ if and only if the original net $\mathcal{N}$ has a firing sequence from M' to some marking in $\mathcal{C}_s$. It then remains to apply Lemma 39 to L and $\mathcal{N}'$, which has the same place count as $\mathcal{N}$. We obtain $\|pre^*(\mathcal{C}_s)\| \leq n|L| + n^2$ and we conclude by applying $|L| \leq \|\mathcal{C}_s\|$.

The bounds still apply to an arbitrary cube $\mathcal{C}$: it is a union of simple cubes, and the norm of a union is the maximum of the norms. □

The above result also holds for counting sets, as finite unions of cubes. With the fact that counting sets are closed under boolean operations, we obtain the following closure result for BIO nets.

Corollary 41. Counting sets of BIO nets are closed under pre^* and boolean operations.

Like for IO nets, we deduce the existence of a polynomial size witness for reachability between two counting sets. For IO nets we used the fact that $\mathcal{S}$ being reachable from $\mathcal{S}'$ entails that $\mathcal{S} \cap post^*(\mathcal{S})$ is non-empty. This time we use that $\mathcal{S}$ being reachable from $\mathcal{S}'$ entails that $\mathcal{S}' \cap pre^*(\mathcal{S})$ is non-empty.

Lemma 42. Let $\mathcal{N} = (P, T, F)$ be a BIO net with n places. Let $\mathcal{S}', \mathcal{S}$ be two counting sets. If $\mathcal{S}$ is reachable from $\mathcal{S}'$, then there exist $C' \in \mathcal{S}', C \in \mathcal{S}$ such that $C' \xrightarrow{*} C$ and $|C'| \leq \|\mathcal{S}'\| + n\|\mathcal{S}\| + n^2$.

Proof. If $\mathcal{S}$ is reachable from $\mathcal{S}'$, then the set of markings described by $\mathcal{S}' \cap pre^*(\mathcal{S})$ is non-empty. By the Closure Theorem (Theorem 40) and Proposition 1, $\mathcal{S}' \cap pre^*(\mathcal{S})$ is a counting set of norm at most $\|\mathcal{S}'\| + n\|\mathcal{S}\| + n^2$. Let $\cup_i \mathcal{C}_i$ be a counting constraint for $\mathcal{S}' \cap pre^*(\mathcal{S})$ whose norm is less than or equal to $\|\mathcal{S}' \cap pre^*(\mathcal{S})\|$. Let $\mathcal{C} = (L, U)$ a cube in $\cup_i \mathcal{C}_i$. Marking C' equal to L on on all places is in $\mathcal{C}$. By definition of the norm, $|C'| \leq \|\mathcal{S}' \cap pre^*(\mathcal{S})\|$. Thus $|C'| \leq \|\mathcal{S}'\| + n\|\mathcal{S}\| + n^2$. Since $C' \in \mathcal{S}' \cap pre^*(\mathcal{S})$, there exists $C \in \mathcal{S}$ such that $C' \xrightarrow{*} C$ and we are done. □

5.3.3 BIO Generalized Reachability Theorem

Like for IO nets, we show a Generalized Reachability Theorem using the Closure Theorem. It is weaker than for IO nets, because it is not true that $post^*$ of a counting set is a counting set, this only holds for pre^*. To state the BIO Generalized Reachability Theorem, we restrict the generalized reachability expressions and problems defined in Section 2.4 so they cannot use the $post^*$ operator.

Let $\mathcal{N} = (P, T, F)$ be a Petri net. A *pre^*-generalized reachability expression* of $\mathcal{N}$ is an expression that is constructed by the following syntax:

$$E := \Gamma \mid pre^*(E) \mid E \cap E \mid E \cup E \mid \overline{E}$$

where Γ is any counting constraint over P. The set $[E]$ represented by E and the length $|E|$ of a pre^*-generalized reachability expression is defined in the same way as for generalized reachability expressions (see Section 2.4). Notice that by Corollary 41, any pre^*-generalized reachability expression E is a counting constraint, and the set of markings $[E]$ is a counting set.

Given a Petri net $\mathcal{N}$, a pre^*-generalized reachability expression E and a marking M, the *pre^*-generalized reachability membership* problem consists of deciding whether M is in $[E]$. Given a Petri net $\mathcal{N}$ and a pre^*-generalized reachability expression E, the *pre^*-generalized reachability emptiness* problem consists of deciding whether $[E]$ is empty. These two problems are called the *pre^*-generalized reachability problems*.

Theorem 43 (BIO Generalized Reachability Theorem)**.** Let $\mathcal{N}$ be a BIO net with n places. Let E be a pre^*-generalized reachability expression of length $|E|$, and let N be the maximum norm of the counting constraints appearing in E. Then

- $[E]$ is a counting set of norm $O(|E| \cdot N \cdot n^{|E|})$,
- the pre^*-generalized reachability membership problem for BIO nets is in PSPACE, and
- the pre^*-generalized reachability emptiness problem for BIO nets is in PSPACE.

Proof. The proof is similar in parts to the proof of Theorem 19 for IO generalized reachability expressions.

Set $[E]$ is a counting set, by Corollary 41. We denote by $\|E\|$ the norm of $[E]$. The bounds for the norms follow from Proposition 1 and Theorem 40. The proof is done in the same way as in the proof of Theorem 19, except that if $E = post^*(E_1)$ or $E = pre^*(E_1)$ for some pre^*-generalized reachability expression E_1, then $\|E\| \leq n\|E_1\| + n^2$ instead of $\|E\| \leq \|E_1\| + n^3$.

We consider the generalized reachability membership problem, where the input is $\mathcal{N}$, E and a marking M. Like for Theorem 19, we proceed by structural induction on E. If E is a counting constraint, if $E = E_1 \cup E_2$, if $E = E_1 \cap E_2$ or if $E = \overline{E_1}$ for some pre^*-generalized reachability expressions E_1, E_2, then the proof is the same as in Theorem 19. The only case which differs is if $E = pre^*([E_1])$. We show that membership in $pre^*([E_1])$ is in PSPACE, where E_1 is a pre^*-generalized reachability expression for which membership is in PSPACE and $[E_1]$ is a counting set with norm $O(|E_1| \cdot N \cdot n^{|E_1|})$.

By Savitch's Theorem, NPSPACE=PSPACE, so it is enough to provide a nondeterministic algorithm. In the proof of Theorem 19 for IO nets, we guessed a marking in $[E_1]$ of same size as M, and checked if there was a firing sequence between them. This procedure is not suited for BIO nets, because the markings along a firing sequence do not all have the same size. Instead, we guess a cube in E_1.

Set $[E_1]$ is a counting set, and thus it is equal to a finite union of cubes, each of norm smaller or equal to $\|E_1\|$. Without loss of generality, we can assume each of these cubes is simple, like in the proof of Theorem 40: each place of the cube either has upper bound equal to ∞ or upper bound equal to the lower bound. The algorithm first guesses a simple cube $\mathcal{C} = (L, U)$ from this finite union. Intuitively $\mathcal{C}$ is such that if $M \in pre^*([E_1])$, then $M \in pre^*(\mathcal{C})$. Storing the cube (with bounds encoded in binary) uses space at most polynomial in $|E|$, $\log(N)$ and n, since its bounds are smaller or equal to $\|E\|$.

We want to verify that $\mathcal{C} \subseteq [E_1]$. This is equivalent to checking whether $\mathcal{C} \cap \overline{[E_1]} = \emptyset$. Because coPSPACE=PSPACE, it is sufficient to give an algorithm running in polynomial space that checks whether $\mathcal{C} \cap \overline{[E_1]} \neq \emptyset$. If the intersection is not empty, i.e. if $\mathcal{C} \cap \overline{[E_1]} \neq \emptyset$, then it has an element M' of size at most $\|\mathcal{C} \cap \overline{[E_1]}\| \leq \|\mathcal{C}\| + n\|[E_1]\| + n$ (by Proposition 1). We guess such an M' and check if it is in $\mathcal{C}$ by comparing it to the bounds (L, U), then

check if it is in $\overline{[E_1]}$ by using the membership algorithm running in polynomial space for $[E_1]$ that we have by assumption. Storing M' encoded in binary requires space at most polynomial in $\log(\|\mathcal{C}\|)$, $\log(\|[E_1]\|)$ and n, i.e. space at most polynomial in $|E|$, $\log(N)$ and n, since the cube and E_1's norms are smaller or equal to $\|E\|$.

We now modify the net as in the proof of Theorem 40: for each place p with upper bound ∞ in $\mathcal{C}$, we add a transition t with preset ${}^\bullet t = \{p\}$ and postset $t^\bullet = \emptyset$. Let L be the lower bound of $\mathcal{C}$. If there exists a firing sequence from our input marking M to L in this modified net, then there exists a firing sequence from M to some marking in $\mathcal{C}$ in the original net. In the modified net, starting in M, the algorithm guesses a firing sequence, step by step, guessing each time a marking of size bounded by Theorem 37 with $m' = |M|$ and $m = |L|$. The algorithm accepts if the marking reached at some step is L.

At any moment in time the algorithm only stores the current marking, the next marking in the sequence, and cube $\mathcal{C} = (L, U)$. A seen above, storing the cube requires space at most polynomial in $|E|$, $\log(N)$ and n. The size of the markings in the sequence are bounded by Theorem 37, i.e. by $(|M|+2^n(|L|+1)^n(n+1)^n(|L|+n)m_d)m_d^n$, where $m_d := \max_{t\in T} |t^\bullet - {}^\bullet t|$. Note that since the added transitions have no destination places, m_d is the same in the original net and in the modified net. Encoding the marking in binary thus requires space at most polynomial in $|E|$, $\log(|M|)$, $\log(N)$, $\log(m_d)$ and n. This concludes the discussion regarding the membership complexity.

To see that checking emptiness of E is in PSPACE, we use the same proof as in Theorem 19. If E is nonempty, then it has an element of size at most $\|E\|$. We guess such an element M in polynomial space, and verify that M is in E using the PSPACE membership algorithm. Storing M encoded in binary requires space at most polynomial in $|E|$, $\log(N)$ and n.

Like for IO nets in Theorem 19, the encoding of the input marking M and of the counting constraints of expression E (whose bounds are smaller or equal to N) can be in unary or in binary, and our algorithms still run in polynomial space. □

Even without *post**, this result is a powerful tool which can be used to prove that many problems are in PSPACE for BIO nets. For instance, the cube-reachability problem from cube $\mathcal{C}'$ to cube $\mathcal{C}$ can be decided by checking if $pre^*(\mathcal{C}) \cap \mathcal{C}'$ is empty. Notice first that since IO nets are a subclass of BIO nets, BIO nets inherit the PSPACE-hardness of reachability, coverability and liveness (Theorem 10).

Theorem 44. The reachability, coverability and liveness problems for BIO nets are PSPACE-hard.

Now we give a theorem echoing the IO net Theorems 20 and 21.

Theorem 45. The cube-reachability, cube-coverability and cube-liveness problems for BIO nets are in PSPACE.

Proof. Let us first consider cube-reachability. For IO nets, we decided it by checking whether $\mathcal{C} \cap post^*(\mathcal{C}')$ was empty. For BIO nets we want to replace $post^*$ by pre^*. Cube $\mathcal{C}'$ can reach $\mathcal{C}$ if and only if the pre^*-generalized reachability expression $\mathcal{C}' \cap pre^*(\mathcal{C})$ is non empty. This can be checked in PSPACE by Theorem 43.

Cube-coverability is reduced to cube-reachability in the same way as in the proof of Theorem 20: cube $\mathcal{C}'$ can cover cube $\mathcal{C}$ if and only if $\mathcal{C}'$ can reach $\mathcal{C}_\infty$, where $\mathcal{C}_\infty$ is the cube with same lower bounds as $\mathcal{C}$ but upper bound ∞ on every place. This can be checked in PSPACE by Theorem 43.

Cube-liveness is treated as in the proof of Theorem 21: the set of live markings of a BIO net $\mathcal{N} = (P, T, F)$ is expressed as

$$\mathcal{L} = \overline{pre^*\left(\bigcup_{t \in T} \overline{pre^*(En(t))}\right)}$$

where $En(t)$ is the cube of markings that enable t. Deciding whether $\mathcal{C} \subseteq \mathcal{L}$ is equivalent to deciding whether $\mathcal{C} \cap \overline{\mathcal{L}} = \emptyset$ holds. This is checked in PSPACE by Theorem 43.

□

Recall that these cube problems can be extended to counting set problems, by virtue of a counting set being a finite union of cubes.

Finally, just like for IO nets, the *structural liveness* problem can be expressed as a generalized reachability problem. It can thus be decided in polynomial space using Theorem 43. This is also proved in [84], using a different technique: the authors show that a BIO net with n places and maximum edge weight w is structurally live if and only if it is live from a marking with at most nw tokens in any place.

Remark 46. As mentioned in Remark 29, each result on pre^* for BIO nets immediately implies the same result on $post^*$ for MIO nets. In particular, the following results hold for MIO nets.

- MIO nets are locally $post^*$-flat but not locally pre^*-flat.
- Let $\mathcal{N} = (P, T, F)$ be a MIO net with n places. Let $\mathcal{C}$ be a cube over P. Then $post^*(\mathcal{C})$ is a counting set and $\|post^*(\mathcal{C})\| \leq n\|\mathcal{C}\| + n^2$.
- The $post^*$-generalized reachability problems for MIO nets are in PSPACE, where $post^*$-generalized reachability problems are expressed using generalized reachability expressions constructed without pre^*.

5.4 Summary and Discussion

We define branching immediate observation (BIO) nets, a class of Petri nets defined syntactically, in which the only possible form of synchronization is observation. It is a

generalization of IO nets: we no longer disallow process creation and destruction. It is also a generalization of BPP nets, or communication free nets. This is a well studied class of Petri nets (see e.g. [27, 39, 50, 64, 68, 72, 85]) in which synchronization is prohibited by having transitions with exactly one input place; in BIO nets we allow observation as well.

We show that a subclass of generalized reachability problems for BIO nets are in PSPACE. This entails that cube-parameterized versions of reachability, coverability and liveness can be solved in polynomial space. This is much better than the complexities for general Petri nets even in the single marking case of these problems (reachability is non-primitive recursive [31], reachability and liveness are recursively equivalent [55], and coverability is EXPSPACE-complete [69, 74]). We develop techniques similar to the ones for IO nets, de-anonymizing token trajectories, and looking at histories instead of firing sequences. This time the trajectories are trees instead of paths, since tokens may "branch" into new tokens. We show a BIO Shortening Theorem which says that if there exists a firing sequence from a target marking to a source marking, then there exists one with few transition alternations; this "few" now depends on the size of the target marking, unlike for IO. Additionally, the theorem bounds the size of the markings in the new firing sequence, i.e. it bounds the number of tokens needed to go from a marking to another. BIO nets may have a non-semilinear forward reachability set, by Hopcroft and Pansiot's well-known example [56], which happens to be a BIO net. This implies that the forward reachability set of a BIO net may not be flat, but we show that BIO nets always have a flat backward reachability set. Our BIO Closure Theorem shows that counting sets are closed under backward reachability (for IO nets they are closed under backward *and* forward reachability), and that the size of the backward reachability set is polynomial in the initial counting set (like for IO nets). This leads to our BIO Generalized Reachability Theorem, which says that pre^*-generalized reachability problems (generalized reachability problems not using $post^*$) can be solved in polynomial space.

BIO nets are essentially the fullest class defined by restricting all synchronization to observation (except for having multiple observations and possibly no source place, a case treated in the last chapter which does not change the complexity results). As far as we know, BIO nets are the first class of Petri nets for which the reachability problem is proved to be easier than in the general case while also having a forward reachability set which may be non-semilinear. The classes of unbounded Petri nets for which the reachability problem is demonstrably simpler than for arbitrary Petri nets – like BPP-nets, reversible nets, and IO nets – all have semilinear reachability sets. The local flatness of BIO nets already allows the application of symbolic model checkers that use acceleration techniques. Indeed, these use semi-decision procedures to compute reachability sets. These procedures terminate if the reachability sets of the net are flat [68], so for BIO nets these can be used to compute the backward reachability set of a net. Like for IO nets, an idea for future work is to see how to tailor their use for BIO nets specifically and apply it in practical cases.

As explained in the introduction and in Remark 29, MIO nets (reverse BIO nets) can be found in the field of type-directed program synthesis [49, 53]. In tool SyPet of [49], Petri nets are used to provide well-typed program sketches, which are then completed using constraint solvers. We expect this to be an interesting field of application. Another domain for BIO net application is chemical reaction networks. As mentioned in the introduction, IO nets have already been used to model enzymatic chemical reactions [4]. It is reasonable to expect that BIO nets find a similar application.

Sources. BIO nets were introduced in [77]. The article contained the BIO Shortening and Flatness Theorems as well as the technical lemmas of Replacement, Smoke Irrelevance and Unique Footprint. The Closure and Generalized Reachability Theorems are unpublished, although they existed in a different form in a first version of [77], which can be found at [76]. That version gave an exponential bound on the size of pre^* of a counting set, whereas this thesis gives a polynomial bound.

6 Extensions

In this chapter we present extensions to the theory of observation Petri nets. Section 6.1 looks at IO and BIO nets which have multiple observed places, and details how the results of the previous chapters evolve. Section 6.2 links IO nets to an existing model of distributed systems called reconfigurable broadcast networks, then considers branching reconfigurable broadcast networks. Finally, Section 6.3 presents some considerations on observation Petri nets and model checking.

6.1 Multiple Observation Nets

A natural extension for IO and BIO nets is to consider multiple observation: that is, a transition with one source place can fire from a marking if and only if a multiset of observed places is covered by the marking. The results of the previous sections change very little for this extension, only adding as factor to some results the maximal number of tokens needed in an observed place to fire a transition of the net.

6.1.1 Immediate Multiple Observation Nets

We consider a natural extension of IO nets called immediate multiple observation nets, or IMO nets, in which a token may observe multiple tokens instead of just the one.

Definition 47. A transition t of a Petri net is an *immediate multiple observation* (IMO) transition if there are places $p_s, p_d, p_{o_1}, p_{o_2}, \ldots, p_{o_k}$ for some $k \in \mathbb{N}$, not necessarily distinct, such that ${}^\bullet t = \wr p_s, p_{o_1}, \ldots, p_{o_k} \int$ and $t^\bullet = \wr p_d, p_{o_1}, \ldots, p_{o_k} \int$. We call p_s and p_d the *source* and *destination* places of t, and $\wr p_d, p_{o_1}, \ldots, p_{o_k} \int$ the *observed* multiset of places of t. A Petri net is an *immediate multiple observation net* if and only if all its transitions are IMO transitions.

We let $m_o = \max(1, \max_{t \in T} \{\min({}^\bullet t(p), t^\bullet(p)) \mid p \in {}^\bullet t \cap t^\bullet\})$ denote the *maximal observation degree* of $\mathcal{N}$, that is the maximal number of tokens needed in an observed place to fire a transition of T, or 1 if this number is 0.

Notice that IMO nets are conservative Petri nets, like IO nets.

Example 22. Figure 6.1 is an IMO net with $m_o = 2$, because the observed multiset of t_1 is $\wr 2p_1 \int$. It is such that p_2 is reachable from a marking M_0 with only tokens in p_1 if and only if $M_0(p_1) \geq 3$. In contrast to the IO net of Figure 3.1, this "threshold check" can be realized in just one transition.

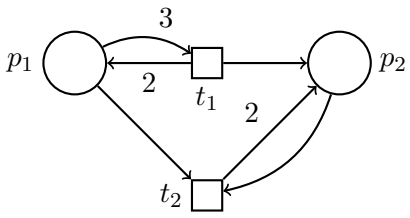


Figure 6.1: *An IMO net.*

We demonstrate that the proofs for the IO nets are applicable to IMO nets with only minor modifications. The idea is that we can reuse the same proofs and formalisms except that instead of making sure there is at least one token in each observed place, we make sure there are at least m_o.

Pruning and Boosting for IMO Nets

Defining histories in the same way as for IO nets, we say that a history H of length $h \geq 1$ is *realizable* if there exist transitions $t_1, \ldots, t_{h-1}$ and numbers $k_1, \ldots, k_{h-1} \geq 0$ such that

- $M_H^1 \xrightarrow{t_1^{k_1}} M_H^2 \cdots M_H^{h-1} \xrightarrow{t_{h-1}^{k_{h-1}}} M_H^h$, where for every t we define $M' \xrightarrow{t^0} M$ iff $M' = M$.
- For every $1 \leq i \leq h-1$, there are exactly k_i trajectories $\tau \in H$ such that $\tau(i)\tau(i+1) = p_s p_d$, where p_s, p_d are the source and target places of t_i, and all other trajectories $\tau \in H$ satisfy $\tau(i) = \tau(i+1)$. Moreover, H contains a submultiset of trajectories $\wr \tau_1', \ldots, \tau_l' \wr$ with $\tau_j'(i) = \tau_j'(i+1) = p_{o_j}$ for all $1 \leq j \leq l$, where $\wr p_{o_1}, \ldots, p_{o_l} \wr$ is the observed multiset of places of t_i.

We say that $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ realizes H. For a step i corresponding to a transition t_i, there must be l trajectories in H that each have a horizontal step at i such that the steps are labeled by the observed multiset $\wr p_{o_1}, \ldots, p_{o_l} \wr$ of t_i. In IO nets this l was always equal to one.

Defining bunches in the same way, we prove a Pruning Lemma for IMO nets in which a "big" bunch is one of size larger than nm_o, instead of n as it was for IO nets.

Lemma 48 (IMO Pruning Lemma)**.** Let $\mathcal{N} = (P, \delta)$ be an IMO net with n places and maximal observation degree m_o. Let H be a realizable history of $\mathcal{N}$ containing a bunch $B \subseteq H$ of size larger than nm_o. There exists a bunch B' of size at most nm_o with the same initial and final places as B, such that the history $H' \stackrel{\text{def}}{=} H - B + B'$ (where $+$ and $-$ denote multiset addition and subtraction) is also realizable in $\mathcal{N}$.

We give an example to illustrate why bunches of size n are too small to prune.

Example 23. Figure 6.2 shows a realizable history H of the IMO net $\mathcal{N}$ of Figure 6.1. The history starts in marking $\wr 3p_1 \wr$ and ends in marking $\wr 3p_2 \wr$. It is realized by $t_1 t_2^2$. There

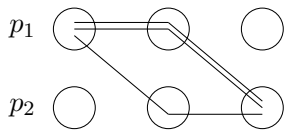


Figure 6.2: *A realizable history of the net in Figure 6.1.*

are three trajectories, all starting in p_1 and ending in p_2. The trajectories make up a bunch of size 3, where $n < 3 \leq nm_o$, since $n = 2$ and $m_o = 2$ for $\mathcal{N}$. If we prune this bunch like we would for an IO net, then we obtain the history made up of one of the top trajectories of H and the bottom trajectory of H. This history is not realizable: no marking is enabled at the initial marking $\wr 2p_1 \int$. Indeed, t_1 needs three tokens in p_1 to fire – one token in the source place (p_1) and two tokens in the observed place (also p_1).

For each of the at most n places p_o visited by a bunch, our pruning operation for IMO nets replaces a "big" bunch with a bunch that has enough horizontal steps labeled by p_o to enable any transition which has p_o in its observed multiset; such a transition may need to observe up to m_o tokens in p_o.

Proof. The proof is the same as for IO nets, except that now we take m_o copies of each trajectory τ_p, where p is a place visited by the bunch B. Let P_B be a set of all places visited by at least one trajectory in the bunch B. For every $p \in P_B$ let $f(p)$ and $l(p)$ be the earliest and the latest moment in time when this place has been used by any of the trajectories (the first and the last occurrence can be in different trajectories). Let $\tau_p, p \in P_B$ be a trajectory that first goes to p by the moment $f(p)$, then waits there until $l(p)$, then goes from p to the final place. To go to and from p it uses fragments of trajectories of B. The portion between is made of horizontal steps stationary in p. Note that we can copy the same fragment of a trajectory multiple times. We take $B' = \{\tau_p \mid p \in P_B\}$ and prove that the history H' obtained by replacing B with m_o copies of B' in H is still realizable.

We build τ_p by taking fragments of existing trajectories and using them at the exact same moments as they are used in H, and by adding some horizontal fragments. Therefore, the set of non-horizontal steps in B' is a subset (if we ignore multiplicity) of the set of non-horizontal steps in B.

Consider any non-horizontal step in H' in any trajectory at position $(i, i+1)$. By construction, the same step at the same position is also present in H. History H is realizable in $\mathcal{N}$, so this step corresponds to a $p_s p_d$ such that there is a $t \in \delta$ with ${}^\bullet t = \wr p_s, p_{o_1}, \ldots, p_{o_k} \int$ and $t^\bullet = \wr p_d, p_{o_1}, \ldots, p_{o_k} \int$, and $M_H^i \xrightarrow{t^l} M_H^{i+1}$ for some $l \geq 1$. Also, H contains an enabling horizontal step $p_o p_o$ for each $p_o \in \wr p_{o_1}, \ldots, p_{o_k} \int$ (with multiplicity) in some trajectory at that position $(i, i+1)$. We must show that for each such horizontal step in H there is a corresponding horizontal step in H', also at position $(i, i+1)$. Consider one such step $p_o p_o$ in H. There are two cases: either the step is provided by a trajectory

that is not in B, or it is provided by a trajectory that is in B. In the first case the same horizontal step is present in H' as a part of the same trajectory. In the second case, note that the place p_o of this horizontal step must be first observed no later than i, and last observed not earlier than $i+1$. This implies $f(p_o) \leq i < i+1 \leq l(p_o)$. As H' contains m_o horizontal steps $p_o p_o$ for all positions between $f(p_o)$ and $l(p_o)$, in particular it contains them at position $(i, i+1)$. The fact that there are m_o such steps in H' ensures that if transition t is enabled by m_o observations of p_o, then we can match each $p_o p_o$ step in H to a $p_o p_o$ step in H'.

Thus H' is realizable. □

The Boosting Lemma holds for IMO nets as it did for IO nets: duplicating a trajectory preserves realizability of a history.

Lemma 49 (IMO Boosting Lemma)**.** Let H be a realizable history of an IMO net containing a trajectory τ. The history $H + \wr\tau\int$ is also realizable.

Results for IMO Nets

Like for IO nets, we can now also prove that IMO nets are globally flat via a bound on the accelerated length of the firing sequences.

Lemma 50 (IMO Shortening)**.** Let $\mathcal{N}$ be an IMO net with n places and maximal observation degree m_o. Let M', M be two markings of $\mathcal{N}$. If $M' \xrightarrow{*} M$, then $M' \xrightarrow{\sigma} M$ for some σ of accelerated length $|\sigma|_a \leq (n^3 \cdot m_o + 1)^n$.

This proof is the same as for Theorem 13 for IO nets, except that the size of the bunches pruned is now nm_o instead of n. A consequence is the global flatness of IMO nets, with the same proof as for IO nets.

Theorem 51. Immediate multiple observation nets are globally flat.

The backward and forward reachability sets of a cube are "small" counting sets.

Theorem 52 (IMO Closure)**.** Let $\mathcal{N}$ be an IMO net with place count n and maximal observation degree m_o. Let $\mathcal{C}$ be a cube. Then $post^*(\mathcal{C})$ is a counting set and

$$\|post^*(\mathcal{C})\| \leq \|\mathcal{C}\| + m_o \cdot n^3$$

The same holds for $pre^*(\mathcal{C})$.

The proof is the same as for IO nets, and yields the IMO Generalized Reachability Theorem.

Theorem 53 (IMO Generalized Reachability Theorem)**.** Let $\mathcal{N}$ be an IMO net with n places and maximal observation degree m_o. Let E be a generalized reachability expression of length $|E|$, and let N be the maximum norm of the counting constraints appearing in E. Then

- $[E]$ is a counting set of norm $O(|E| \cdot N \cdot m_o \cdot n^{|E|})$,
- the generalized reachability membership problem for IMO nets is in PSPACE, and
- the generalized reachability emptiness problem for IMO nets is in PSPACE.

Like for IO nets, this results entails PSPACE-completeness of cube-parameterized results.

Theorem 54. The cube-reachability, cube-coverability, cube-liveness and stuctural liveness problems for IMO nets are PSPACE-complete.

6.1.2 Branching Immediate Multiple Observation Nets

We now consider an extension of BIO nets called branching immediate multiple observation nets, or BIMO nets, in which a token may observe multiple tokens. It also extends BIO nets by allowing transitions to have *no source place*. Like for IMO and IO nets, BIMO nets retain the properties of BIO nets.

Definition 55. A transition t of a Petri net is a *branching immediate multiple observation* (BIMO) transition if $|{}^\bullet t - t^\bullet| \leq 1$ (recall that multiset subtraction is the component-wise maximum between the subtraction of the components and 0). A Petri net $\mathcal{N} = (P, T, F)$ is a *branching immediate multiple observation net* if and only if all its transitions are BIMO transitions.

We let $m_o = \max(1, \max\{\min({}^\bullet t(p), t^\bullet(p)) \mid t \in T, p \in {}^\bullet t \cap t^\bullet\})$ denote the *maximal observation degree* of $\mathcal{N}$, that is the maximal number of tokens needed in an observed place to fire a transition of T, or 1 if this number is 0.

The interpretation of BIMO transitions in terms of observation is as follows:

- either t is of the form ${}^\bullet t = \wr p_s, p_{o_1}, \ldots, p_{o_k} \int$ and $t^\bullet = \wr p_{d_1}, \ldots, p_{d_l}, p_{o_1}, \ldots, p_{o_k} \int$,
- or t is of the form ${}^\bullet t = \wr p_{o_1}, \ldots, p_{o_k} \int$ and $t^\bullet = \wr p_{d_1}, \ldots, p_{d_l}, p_{o_1}, \ldots, p_{o_k} \int$,

with p_s the *source* place, $p_{d_1}, \ldots, p_{d_l}$ the *destination* places, and $p_{o_1}, \ldots, p_{o_k}$ the *observed* places of t, all of them not necessarily distinct. In other words, each BIMO transition is such that there is at most one source place. Additionally l and k can be equal to zero, meaning there may be no destination or no observed places. Like BIO transitions and contrary to IO and IMO transitions, a BIMO transition can destroy tokens (via transitions t with $t^\bullet = \wr p_{o_1}, \ldots, p_{o_k} \int$) or create tokens (via transitions t with $|t^\bullet| \geq |{}^\bullet t|$). Thus BIMO nets are not conservative Petri nets.

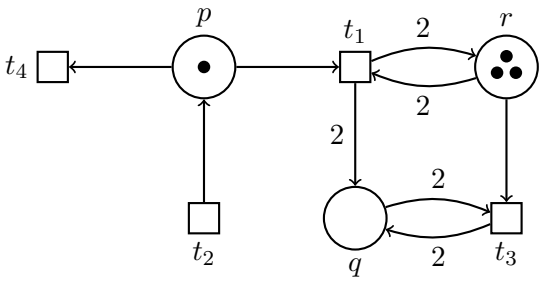


Figure 6.3: *A BIMO net.*

Example 24. Figure 6.3 shows a BIMO net with three places and four transitions. The maximal observation degree m_o is equal to 2.

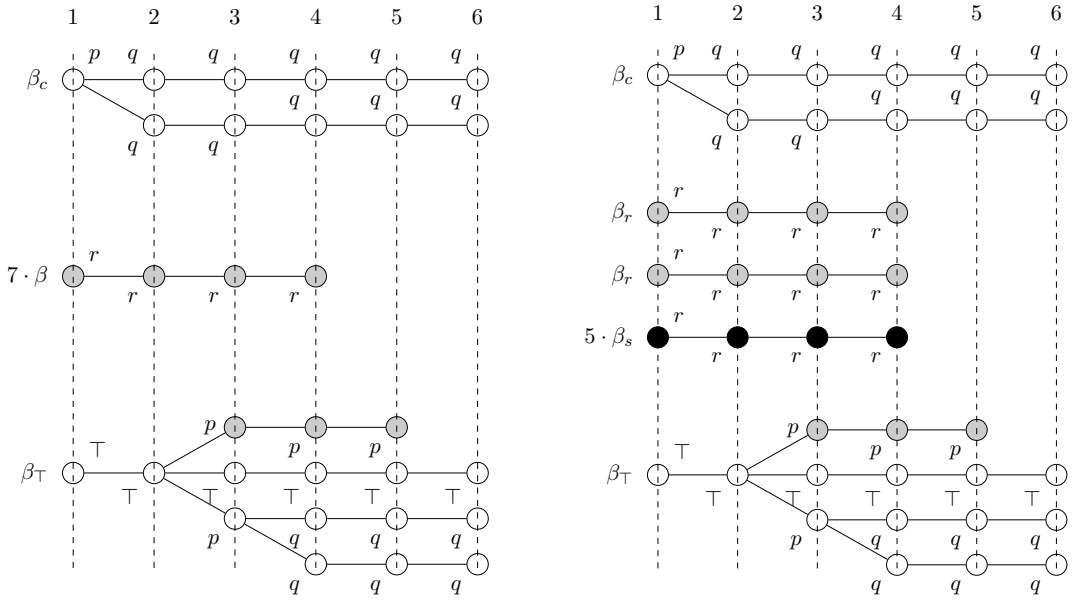
Remark 56. One can also define *merging immediate multiple observation* (MIMO) nets, akin to the MIO nets of Remark 29. These are the Petri nets obtained from BIMO nets by reversing all the transitions. Reversing all the transitions in a BIMO net yields a MIMO net, and vice versa. Like for BIO and MIO nets, our results for BIMO nets imply dual results for MIMO nets: any result on pre^* for BIMO nets implies the same result on $post^*$ for MIMO nets.

Branching Histories for BIMO Nets

We fix a BIMO net $\mathcal{N} = (P, T, F)$ with n places and maximal observation degree m_o. To make further proofs easier, we do a *preprocessing* step in which we modify all transitions to have a source place. We add a place $\top$ to P, and for every transition $t \in T$ such that ${}^\bullet t = \emptyset$, we change the preset to $\wr \top \int$ and the postset to $\wr \top \int + t^\bullet$. Intuitively, we want to place one token in $\top$ in every execution so that all originally "source-less" transitions are always enabled. Accordingly, in the following we will consider only firing sequences (respectively histories) that start from a marking M' with $M'(\top) = 1$. This entails that there is exactly one token (resp. node) in $\top$ at each marking of the firing sequence (resp. at each level of the history), since transitions with source place $\top$ also have $\top$ exactly once in their destination multiset. We have that $M' \xrightarrow{\sigma} M$ in the original net if and only if $M' + \wr \top \int \xrightarrow{\sigma} M + \wr \top \int$ in the preprocessed net (abusing notation and equating t in the original net with t in the preprocessed net). In the rest of the section, we assume that we have preprocessed our BIMO net in this way.

Example 25. Let us apply the preprocessing step to the net of Figure 6.3. Transition t_2 is the only transition without source place; we replace it with (abusing notation) t_2 such that ${}^\bullet t_2 = \wr \top \int$ and $t_2{}^\bullet = \wr \top, p \int$.

We define branching trajectories, histories and decorations in the same way as for BIO nets. Note that by our preprocessing step, P contains $\top$, so there may be nodes labelled



(a) *A realizable decorated history.* **(b)** *A fuel-efficient realizable decorated history.*

Figure 6.4: *A realizable decorated history in the BIMO net of Figure 6.3 before and after application of the BIMO Replacement Lemma.*

by the symbol $\top$. We say that a history H of length $h \geq 1$ is *realizable* if there exist transitions $t_1, \ldots, t_{h-1} \in T$ and numbers $k_1, \ldots, k_{h-1} \geq 0$ such that for every $1 \leq i \leq h-1$ the set $H(i)$ can be partitioned into two sets:

- A set $H_a(i)$ containing exactly k_i *active* nodes labeled by the source place of t_i. Given a particular active node, say v, the multiset of labels of its children is the (possibly empty) multiset $\{\!\{p_{d_1}, \ldots, p_{d_l}\}\!\}$ of destination places of t_i.
- A set $H_p(i)$ containing *passive* nodes, each of them with exactly one child, carrying the same label as their parents. This set must contain a set of nodes such that the multiset of their labels is equal to the (possibly empty) observed multiset of places of t_i: $\{\!\{p_{o_1}, \ldots, p_{o_k}\}\!\}$.

We say that the sequence $t_1^{k_1} \cdots t_{h-1}^{k_{h-1}}$ *realizes* H. Since we consider histories starting with exactly one $\top$-node, this entails that there can be at most one active $\top$-node.

Example 26. Figure 6.4a shows a realizable history H of the BIMO net of Figure 6.3. It consists of nine branching trajectories: β_c, seven identical trees β, and $\beta_\top$. The initial and final markings are $\{\!\{p, 7r, \top\}\!\}$ and $\{\!\{4q, \top\}\!\}$. The firing sequence that realizes H is $t_1\, t_2\, t_1\, t_3^7\, t_4$.

The definition of a decoration is the same, except for the "no smoke without fuel" rule. It is modified to reflect the possible need to observe m_o p-nodes. Formally, a *decoration* of H is a partition of the nodes of H into *cargo*, *fuel*, and *smoke* nodes satisfying the following conditions:

- A node of H is a *cargo node* if and only if it has at least one descendant in $H(l)$.
- All descendants of smoke nodes are smoke nodes.
- For every place p and level i, if $H(i)$ contains smoke p-nodes, then it also contains at least m_o fuel p-nodes. ("No smoke without fuel". Intuitively, the smoke p-nodes are not needed because the fuel p-nodes can be used instead.)

Note that since the histories we consider have exactly one $\top$-node at each level, this $\top$-node must be decorated as cargo.

Example 27. Figure 6.4a shows a realizable decorated history $\widehat{H}$. The decoration is the "conservative decoration", where any non-cargo node is fuel.

The definition of fuel-efficient is also modified to accommodate multiple observations.

Definition 57. Let $\widehat{H}$ be a decorated history. A place p is *wasteful at level i* if $\widehat{H}(i)$ contains more than nm_o fuel p-nodes. A place p is *wasteful* in $\widehat{H}$ if it is wasteful at some level; otherwise p is *fuel-efficient* in $\widehat{H}$. Finally, $\widehat{H}$ is *fuel-efficient* if all places are fuel-efficient.

The Replacement Lemma for BIMO nets is proved in the same way as for BIO nets, except that instead of one copy of β_q in B_f, we put m_o copies of β_q in B_f.

Lemma 58 (BIMO Replacement Lemma)**.** Let $\widehat{H}$ be a decoration of a realizable history H such that p is wasteful, and i is the earliest level at which p is wasteful. There exists a history B'_p such that $H' \stackrel{\text{def}}{=} H[B'_p/B_p(i)]$ is realizable, equivalent to H, and has a decoration whose fuel-efficient places contain all fuel-efficient places of $\widehat{H}$ and p.

Example 28. Place r is wasteful at levels 1 to 4 of the decorated history $\widehat{H}$ of Figure 6.4a. Applying the Replacement Lemma to r and $i := 1$ to $\widehat{H}$ yields the decorated history $\widehat{H'}$ of Figure 6.4b. Like H, it leads from $\wr p, 7r, \top\int$ to $\wr 4q, \top\int$. It is made up of $m_o = 2$ copies of β decorated as fuel, and five copies of β decorated as smoke. Place r is no longer wasteful in $\widehat{H'}$, and in fact all places are fuel-efficient.

The Smoke Irrelevance and Unique Footprint Lemma hold and are proved in the same way as for BIO nets.

Results for BIMO Nets

Like for BIO nets, we prove that BIMO nets are locally flat via a bound on the accelerated length of the firing sequences.

Lemma 59 (BIMO Shortening)**.** Let $\mathcal{N}$ be a BIMO net with n places and maximal observation degree m_o. Let $m_d := \max_{t\in T} |t^\bullet - {}^\bullet t|$ denote the maximum number of tokens created by a transition of $\mathcal{N}$. Let M', M be two markings of $\mathcal{N}$. If $M' \xrightarrow{*} M$, then $M' \xrightarrow{\sigma} M$

for some σ of accelerated length $|\sigma|_a \leq 2^n(m+1)^n(nm_o+1)^n$. Further, the intermediate markings along σ have size at most $(m' + 2^n(m+1)^n(nm_o+1)^n(m+nm_o)m_d)m_d^n$.

This proof is the same as for the BIO Shortening Theorem, except that the maximum number of fuel nodes in a place of a fuel-efficient history is now nm_o instead of n, modifying the bounds. The bounds of this result also hold for a BIMO net which has not gone through the preprocessing step, since $M' \xrightarrow{\sigma} M$ in the non-preprocessed net if and only if $M' + \wr\top\int \xrightarrow{\sigma} M + \wr\top\int$ in the preprocessed net.

Proof. Like in the proof for BIO nets (Theorem 37), we give ourselves a fuel-efficient decorated history of unique footprint starting in M' and ending in M. Every level has between 0 and m cargo nodes per place. Additionally, since $\widehat{H}$ is fuel-efficient, every level has between 0 and nm_o fuel nodes per place. Finally, there are at most 2^n possible supports in a net with n places. So the number of footprints, and so the length of $\widehat{H}$, and the accelerated length of any firing sequence realizing $\widehat{H}$, is at most $2^n(m+1)^n(nm_o+1)^n$.

The proof for the token bound is the same as the proof of Theorem 37 for BIO nets, except that now there are at most $2^n(m+1)^n(nm_o+1)^n$ levels in the history, and at most $(m+nm_o)$ cargo and fuel nodes per place at each level. This entails the token bound $(m' + h'(m+nm_o)m_d)m_d^n$. □

A consequence is the local *pre**-flatness of BIMO nets, with the same proof as for BIO nets. The fact that BIMO nets are not globally flat nor locally *post**-flat follows from the fact that BIO nets are BIMO nets, and they are not globally flat nor locally *post**-flat.

Theorem 60. BIMO nets are locally *pre**-flat, but neither globally flat nor locally *post**-flat.

Like for BIO nets, the backward reachability set of a cube is a "small" counting set.

Theorem 61 (BIMO Closure)**.** Let $\mathcal{N}$ be a BIMO net with n places and maximal observation degree m_o. Let $\mathcal{C}$ be a cube. Then $pre^*(\mathcal{C})$ is a counting set and

$$\|pre^*(\mathcal{C})\| \leq n \cdot \|\mathcal{C}\| + m_o \cdot n^2$$

The proof is the same as for BIO nets, and yields the BIMO Generalized Reachability Theorem. Recall that *pre**-generalized reachability expressions are generalized reachability expressions that do not use *post**, and *pre**-reachability problems are the corresponding membership and emptiness problems (see definitions in Section 5.3.3).

Theorem 62 (BIMO Generalized Reachability Theorem)**.** Let $\mathcal{N}$ be a BIMO net with n places and maximal observation degree m_o. Let E be a *pre**-generalized reachability expression of length $|E|$, and let N be the maximum norm of the counting constraints appearing in E. Then

- $[E]$ is a counting set of norm $O(|E| \cdot N \cdot m_o \cdot n^{|E|})$,
- the pre^*-generalized reachability membership problem for BIMO nets is in PSPACE, and
- the pre^*-generalized reachability emptiness problem for BIMO nets is in PSPACE.

Like for BIO nets, this results entails PSPACE-completeness of cube-parameterized results.

Theorem 63. The cube-reachability, cube-coverability, cube-liveness and stuctural liveness problems for BIMO nets are PSPACE-complete.

6.2 Reconfigurable Broadcast Networks

Reconfigurable broadcast networks (RBN) were introduced in [33, 34] and studied in [16, 17, 26] and other papers. RBN are networks consisting of finite-state, anonymous agents which run the same protocol. The agents communicate by selective broadcast: a process can broadcast a message which is received by all of its neighbors, and the set of neighbors of an agent can change arbitrarily over time. This model can be seen as an extension of IO nets: the broadcasting agent plays the role of an observed token which can move to a different place after being observed. Since the neighbors of the broadcasting agent can be any subset of the other agents, the agent does not know if it has been "observed" (i.e. if its message has been received) by other tokens – the information flow goes in only one direction, like for IO nets.

We give a simulation of IO nets by RBN, which allows us to transfer results for RBN to IO nets. We show however that IO nets cannot simulate RBN. In particular, we exhibit a result proved for IO nets in the previous sections which does not hold for RBN. Finally, inspired by the extension from IO to BIO, we initiate the study of *branching* RBN.

6.2.1 Definition

Reconfigurable broadcast networks (RBN) are networks consisting of finite-state, anonymous agents and a communication topology which specifies, for every pair of agents, whether or not there is a communication link between them. Agents with a communication link are neighbors. During a step, a single agent can broadcast a message which is received by all of its neighbors, after which both the agent and its neighbors change their state according to some transition relation. Further, in between two steps, the communication topology can change in an arbitrary manner. These reconfigurations of the communication topology can be seen as spontaneous movements of the nodes which can connect and disconnect from each other at any moment. One can also picture a fully connected network in which messages sometimes get lost: only a subset of the agents receive a broadcast. In the frame

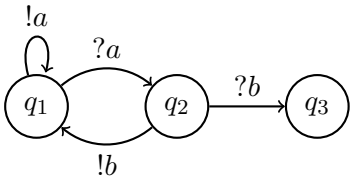


Figure 6.5: *An RBN $\mathcal{R}$ with three states.*

of this thesis, it is easier to forget the communication topology and define the semantics of an RBN directly in terms of collections of agents.

Definition 64. A *reconfigurable broadcast network* is a tuple $\mathcal{R} = (Q, \Sigma, \delta)$ where Q is a finite set of states, Σ is a finite alphabet and $\delta \subseteq Q \times \{!a, ?a \mid a \in \Sigma\} \times Q$ is the transition relation.

If $(p, !a, q)$ (resp. $(p, ?a, q)$) is a transition in δ, we denote it by $p \xrightarrow{!a} q$ (resp. $p \xrightarrow{?a} q$). We call transitions with $!a$ broadcast transitions, and transitions with $?a$ receive transitions. A *configuration* C of a RBN $\mathcal{R}$ is a multiset over Q, which intuitively counts the number of agents in each state. Given a letter $a \in \Sigma$ and two configurations C and C' we say that there is a *step* $C \xrightarrow{a} C'$ if there exists a multiset $\wr t, t_1, \ldots, t_k \wr$ of δ for some $k \geq 0$ satisfying the following: t is of the form $p \xrightarrow{!a} q$, each t_i is of the form $p_i \xrightarrow{?a} q_i$, and the configurations are such that $C \geq \boldsymbol{p} + \sum_i \boldsymbol{p_i}$ and $C' = C - \boldsymbol{p} - \sum_i \boldsymbol{p_i} + \boldsymbol{q} + \sum_i \boldsymbol{q_i}$. We sometimes write this as $C \xrightarrow{t+t_1,\ldots,t_n} C'$ or $C \xrightarrow{a} C'$. Intuitively it means that an agent at the state p broadcasts the message a and moves to q, and for each $1 \leq i \leq k$, there is an agent at the state p_i which receives this message and moves to q_i. We denote by $\xrightarrow{*}$ the reflexive and transitive closure of the step relation. A *run* is then a sequence of steps.

Let C, C', C'' be configurations of a given RBN. We say that C' is *reachable* from C if there exists a run from C to C'. We say that C'' is *coverable* from C if there exists a configuration C' which covers C'', and there exists a run from C to C'. The reachability and coverability problems as well as the cube-reachability and cube-coverability problems are defined as expected.

Example 29. Figure 6.5 illustrates a RBN $\mathcal{R} = (Q, \Sigma, \delta)$ with $Q = \{q_1, q_2, q_3\}$ and $\delta = \{t_1, t_2, t_3, t_4\}$, where

$$t_1 = q_1 \xrightarrow{!a} q_1 \qquad t_2 = q_1 \xrightarrow{?a} q_2 \qquad t_3 = q_2 \xrightarrow{!b} q_1 \qquad t_4 = q_2 \xrightarrow{?b} q_3$$

Configuration $\wr 3 \cdot q_1 \wr$ can reach $\wr 2 \cdot q_1, q_3 \wr$ in two steps. First, an agent broadcasts a, while the two other agents receive a and move to q_2. Then, one of the agents in q_2 broadcasts b and moves to q_1, while the other one receives b and moves to q_3. The run described can be

written as

$$\wr 3 \cdot q_1 \wr \xrightarrow{t_1+2\cdot t_2} \wr q_1, 2 \cdot q_2 \wr \xrightarrow{t_3+t_4} \wr 2 \cdot q_1, q_3 \wr.$$

Notice that $\wr q_3 \wr$ is only coverable from a configuration $\wr k \cdot q_1 \wr$ if $k \geq 3$.

Remark 65. RBN are conservative, in the sense that no transition adds or removes agents, and thus the number of agents stays constant along a run. In particular, this entails that the reachability and coverability problems between single configurations are in PSPACE: given an initial and final configuration, it suffices to guess a sequence of configurations, step-by-step reachable, only remembering the last one and checking whether it equals or covers the final configuration.

RBN, like Petri nets, have the monotonicity property: given configurations C, C', D of an RBN $\mathcal{R}$, if $C \xrightarrow{*} C' + D$, then $C \xrightarrow{*} C' + D$. The proof follows directly from the fact that agents in an RBN cannot disable the occurrence of a transition.

RBN also have the "copycat" property. Consider a run $C \xrightarrow{*} C'$. Like for IO nets, we can think of each agent in the run as following a trajectory. If we add an agent to a state q of C already containing an agent, i.e. such that $C(q) \geq 1$, then this new agent can "copy" the trajectory of the old agent, taking the same transitions (broadcast or receive) along the run.

Example 30. Consider the run $\wr 3 \cdot q_1 \wr \xrightarrow{t_1+2\cdot t_2} \wr q_1, 2 \cdot q_2 \wr \xrightarrow{t_3+t_4} \wr 2 \cdot q_1, q_3 \wr$ of Example 29. Let us add an agent to q_1 in the first configuration. It can copy the trajectory of the agent which goes to q_3, yielding the run

$$\wr 4 \cdot q_1 \wr \xrightarrow{t_1+3\cdot t_2} \wr q_1, 3 \cdot q_2 \wr \xrightarrow{t_3+2\cdot t_4} \wr 2 \cdot q_1, 2 \cdot q_3 \wr.$$

Or it can copy the trajectory of the agent which goes to q_2 then back to q_1, yielding the run

$$\wr 4 \cdot q_1 \wr \xrightarrow{t_1+3\cdot t_2} \wr q_1, 3 \cdot q_2 \wr \xrightarrow{t_3+t_4} \wr 2 \cdot q_1, q_2, q_3 \wr \xrightarrow{t_3} \wr 3 \cdot q_1, q_3 \wr.$$

6.2.2 Simulation of IO nets by RBN

Following the idea that RBN can be seen as an extension of IO nets, we show that RBN can simulate IO nets in the following way.

Theorem 66 (Simulation)**.** Given an IO net $\mathcal{N}$ with states $Q_\mathcal{N}$, there exists a RBN $\mathcal{R}$ with states $Q_\mathcal{R}$ such that $\mathcal{R}$ is polynomial in the size of $\mathcal{N}$, $Q_\mathcal{N} = Q_\mathcal{R}$, and $C' \xrightarrow{*} C$ in $\mathcal{N}$ if and only if $C' \xrightarrow{*} C$ in $\mathcal{R}$ for any multisets C, C' over $Q_\mathcal{N}$.

The proof essentially builds an RBN from an IO net by modelling observations as broadcasts where the broadcaster does not change states. Given an IO net or an RBN I, we let $post_I^*$ and pre_I^* denote the forward and backward reachability operators in I.

Proof. Let $\mathcal{N} = (Q, \delta)$ be an IO net. We construct an RBN that simulates $\mathcal{N}$ in which processes send messages signalling their current state. Let $\mathcal{R} = (Q', \Sigma', \delta')$ be the following RBN: the set of states Q' and the alphabet Σ' are both equal to Q. The transition relation δ' is such that for every $q \in Q$ there is a transition $q \xrightarrow{!q} q$ in δ', and for every $p \xrightarrow{q} p' \in \delta$ there is a transition $p \xrightarrow{?q} p'$ in δ'.

There is a natural bijection between configurations of $\mathcal{R}$ and markings of $\mathcal{N}$. We show that $C' \in post^*_{\mathcal{N}}(C)$ if and only if $C' \in post^*_{\mathcal{R}}(C)$ for any configurations C and C' of $\mathcal{N}$. Indeed, if C reaches C' by one step $p \xrightarrow{q} p'$ in $\mathcal{N}$, then $C \xrightarrow{t+t_1} C'$ with $t = q \xrightarrow{!q} q$ and $t_1 = p \xrightarrow{?q} p'$ in $\mathcal{R}$. Conversely, let $C \xrightarrow{t+t_1,\ldots,t_k} C'$ be a step in $\mathcal{R}$ with $t = q \xrightarrow{!q} q$ and $t_i = p_i \xrightarrow{?q} p'_i$ for some $k \geq 0$. The step must be of this form because the only broadcast transitions of $\mathcal{R}$ are of the form $q \xrightarrow{!q} q$. Then C reaches C' by the sequence of transitions $(p_1 \xrightarrow{q} p'_1), (p_2 \xrightarrow{q} p'_2), \ldots, (p_k \xrightarrow{q} p'_k)$ in $\mathcal{N}$. □

Using the Simulation Theorem above, we transfer results from RBN to IO nets.

Restrictions of cube-reachability. In [33], two restrictions of the cube-reachability problem for RBN are considered. The first restriction, dubbed $CRP[\geq 1]$ (where CRP stands for cardinality reachability problem), considers cube pairs $\mathcal{C} = (L, U), \mathcal{C}' = (L', U')$ such that $L(q) = 0$ and $U(q) \in \{0, \infty\}$ for every state q, and $L'(q) \in \{0, 1\}$ and $U'(q) = \infty$ for all q. That is, the initial cube allows for an unbounded number of agents only in some initial states, and the final cube detects the presence of at least one agent in some states. The second restriction, dubbed $CRP[\geq 1, = 0]$, considers cube pairs $\mathcal{C} = (L, U), \mathcal{C}' = (L', U')$ such that $L(q) = 0$ and $U(q) \in \{0, \infty\}$ for every state q, and $L'(q) \in \{0, 1\}$ and $U'(q) \in \{0, \infty\}$ for all q. That is, the initial cube allows for an unbounded number of agents only in some initial states, and the final cube detects the presence or absence of agents in some states. For RBN, the problems $CRP[\geq 1]$ and $CRP[\geq 1, = 0]$ are in P and in NP respectively (Theorem 3.3 and 4.3 of [33]). By the construction given above, we get:

Theorem 67. For IO nets, $CRP[\geq 1]$ and $CRP[\geq 1, = 0]$ are in P and in NP respectively.

We define IO nets equipped with leaders, inspired by RBN with leaders [11] and asynchronous shared memory systems, another distributed system model, with leaders [25, 43].

IO nets with leaders. An *IO-leader net* is an IO net $\mathcal{N} = (Q, \delta)$ with Q and δ fulfilling the following conditions. Set Q is the disjoint union of non empty sets Q_L, Q_C, where Q_L is a finite set of *leader* places and Q_C is a finite set of *contributor* places. The transition relation δ is the disjoint union of δ_L, δ_C, where $\delta_L \subseteq Q_L \times Q_C \times Q_L$ and $\delta_C \subseteq Q_C \times Q \times Q_C$. Intuitively, there is exactly one token which moves around places of Q_L (the leader). All

the other tokens moves around places of Q_C (the contributors). This is formalized as follows: a marking of such a system is defined to be a pair (q, M) where $q \in Q_L$, and M is a multiset on Q_C. A step between $C = (q, M)$ and $C' = (q', M')$ exists if one of the following is true:

- There exists $q \xrightarrow{p} q' \in \delta_L$ such that $M(p) \geq 1$ and $M' = M$.
- There exists $p \xrightarrow{r} p' \in \delta_C$ such that $M(p) \geq 1$, $M' = M - \boldsymbol{p} + \boldsymbol{p'}$, $q = q'$, and either $q = r$ or $M(r) \geq 1$.

We can then define the notion of a firing sequence for an IO-leader net in the usual way. The *IO-leader reachability* problem is to decide, given an IO-leader net $\mathcal{N}$, two leader places q_L^I, q_L^f, and a contributor place q_C^I, whether there exists a $k \geq 1$ such that the marking $(q_L^I, \{\!|k \cdot q_C^I|\!\})$ can reach a marking $C' = (q_L^f, M')$ for some M'.

We now define a special case of cube-reachability in IO nets and notice that this special case is exactly equivalent to IO-leader reachability. An *IO-leader cube* is a pair $(\mathcal{N}, \mathcal{C}, \mathcal{C}')$ of the following form: the net $\mathcal{N} = (Q, \delta)$ is such that there exists a partition of the places and transition relation as $Q = Q_C \cup Q_L, \delta = \delta_C \cup \delta_L$. And $\mathcal{C} = (L, U), \mathcal{C}' = (L', U')$ satisfy: there exists exactly two places $q_L^I, q_L^f \in Q_L$ such that $L(q_L^I) = U(q_L^I) = 1, L'(q_L^f) = U'(q_L^f) = 1$ and for every other place $q \in Q_L$, $L(q) = L'(q) = U(q) = U'(q) = 0$. Additionally, there exists exactly one place $q_C^I \in Q_C$ such that $L(q_C^I) = L'(q_C^I) = 0, U(q_C^I) = U'(q_C^I) = \infty$ and for every other place $q \in Q_C$, $L(q) = U(q) = L'(q) = 0, U'(q) = \infty$. It is easy to see that the IO-leader reachability problem is equivalent to the cube-reachability problem for IO-leader cubes.

RBN-leader protocols are defined in an analog way [11]: an *RBN-leader protocol* is an RBN $\mathcal{R} = (Q, \Sigma, \delta)$ fulfilling the following conditions. Set Q is the disjoint union of non empty sets Q_L, Q_C. The transition relation δ is the disjoint union of δ_L, δ_C, where $\delta_L \subseteq Q_L \times \{!a, ?a \mid a \in \Sigma\} \times Q_L$ and $\delta_C \subseteq Q_C \times \{!a, ?a \mid a \in \Sigma\} \times Q_C$. A configuration of such a system is defined to be a pair (q, C) where $q \in Q_L$, and C is a multiset on Q_C. A step between (q, C) and (q', C') exists if there is a step $(q, C) \xrightarrow{a} (q', C')$ in $\mathcal{R}$ for (q, C) and (q', C') seen as configurations of Q. The following result has been shown.

Theorem 68 ([11])**.** The RBN-leader reachability problem is NP-complete.

From this and our simulation we deduce the following.

Theorem 69. The IO-leader reachability problem is NP-complete.

Proof. For the upper bound, notice that given an IO-leader cube, our Simulation Theorem produces an RBN-leader cube. Since the reachability problem for RBN-leader cubes is in NP, this immediately gives us the same upper bound for IO-leader cube reachability.

For the lower bound, we give a reduction from 3-SAT. Let $\varphi = \bigwedge_{i=1}^{m} C_i$ be a 3-CNF formula over the variables $x_1, \ldots, x_n$ where each $C_i = \ell_i^1 \vee \ell_i^2 \vee \ell_i^3$. Construct an IO-leader

net as follows : the leader will have $2n + m + 1$ states $q_0, q_1, \bar{q_1}, \ldots, q_n, \bar{q_n}, p_1, \ldots, p_m$ and the contributors will have $2n + 1$ states $init, x_1, \bar{x_1}, x_2, \bar{x_2}, \ldots, x_n, \bar{x_n}$.

We allow ourselves to have no observed place for some transitions (or equivalently we add one always-marked place which serves as observed place for any transition lacking one). For every $1 \leq i \leq n$, from the state q_{i-1}, the leader can move to q_i or move to $\bar{q_i}$ (no observation). Similarly, from the state $\bar{q_{i-1}}$, the leader can move to q_i or move to $\bar{q_i}$ (no observation). These transitions intuitively correspond to the leader guessing that the variable x_i is either true or false. From the state $init$, a contributor can move to y_i if it observes a token in q_i or move to $\bar{y_i}$ if it observes a token in $\bar{q_i}$. Intuitively, for each i, some contributor receives the guess made by the leader and stores it in its finite set of states so that it can be observed by the leader later on.

Denoting the state q_n by p_0, for every $1 \leq i \leq m$, from the state p_{i-1}, the leader can observe any one of the literals $\ell_i^1, \ell_i^2, \ell_i^3$ and move to p_i. Hence, the leader can move to p_i from p_{i-1} if and only if the guesses that it made before satisfy the i-th clause.

If we now set that the leader must start at q_0 and end up at p_m and the contributors must start at $init$, then it is clear from construction that the IO-leader reachability problem for this leader net is true if and only if φ is satisfiable. □

Using the Simulation Theorem in the opposite direction, we show that reachability and coverability are PSPACE-hard for RBN. With Remark 65, this implies PSPACE-completeness of these problems.

Theorem 70. The reachability and coverability problems for RBN are PSPACE-complete.

Proof. As mentioned in Remark 65, reachability and coverability problems for RBN are in PSPACE: since the number of agents does not change along a run, it suffices to explore the finite reachability graph of configurations of a given size non-deterministically.

IO net reachability and coverability are PSPACE-hard, we reduce them to RBN reachability and coverability to show that these problems are PSPACE-hard. Let $\mathcal{N}$ an IO net of state set Q. By Theorem 66, there exists a RBN $\mathcal{R}$ with states Q, polynomial in the size of $\mathcal{N}$, and such that $C' \xrightarrow{*} C$ in $\mathcal{N}$ if and only if $C' \xrightarrow{*} C$ in $\mathcal{R}$ for any multisets C, C' over Q. So C is reachable (respectively coverable) from C' in $\mathcal{N}$ if and only if it is in $\mathcal{R}$. □

6.2.3 RBN are more complex than IO nets

We show that the converse of the Simulation Theorem (Theorem 66) is not true: IO nets cannot simulate RBN even with a more permissive definition of simulation. If this was the case, the results of the IO Closure Theorem (Theorem 15) stating that reachability sets of a cube are counting sets of polynomial norm could be transferred to RBN. But this would lead to a contradiction. We show first the transfer of results, then the contradiction.

Essentially, we show that the reachability set of a cube in an RBN is not necessarily of polynomial norm by exhibiting an exponential example.

We establish some notation and a simulation definition. Let C be a multiset over a finite set Q, where intuitively C is a configuration if Q is the state set of an RBN and a marking if Q is the place set of an IO net. Let Q' be a finite set disjoint from Q, and let h be a multiset over Q'. Then $C \cdot h$ is defined as the multiset over $Q \cup Q'$ with $(C \cdot h)(q) = C(q)$ for all $q \in Q$ and $(C \cdot h)(q) = h(q)$ for all $q \in Q'$. Given an RBN $\mathcal{R}$ with states $Q_\mathcal{R}$, we say $\mathcal{R}$ is *polynomially simulated* by an IO net $\mathcal{N}$ with places $Q_\mathcal{N}$ if $\mathcal{N}$ is polynomial in the size of $\mathcal{R}$ with $Q_\mathcal{R} \subseteq Q_\mathcal{N}$, and there exists a multiset h over $Q_\mathcal{N} \setminus Q_\mathcal{R}$ of polynomial size such that $C' \in post^*_\mathcal{R}(C)$ if and only if $C' \cdot h \in post^*_\mathcal{N}(C \cdot h)$ for any multisets C, C' over $Q_\mathcal{R}$.

Lemma 71. Assume that for every RBN $\mathcal{R}$ there exists an IO net $\mathcal{N}$ such that $\mathcal{R}$ can be polynomially simulated by $\mathcal{N}$. Then there exists a constant k such that for any RBN with n states, for any cube $\mathcal{C}$ of $\mathcal{R}$, $post^*(\mathcal{C})$ is a counting set and $\|post^*(\mathcal{C})\| \in O(\|\mathcal{C}\| + n)^k$.

Proof. The polynomial simulation of an RBN $\mathcal{R}$ with states $Q_\mathcal{R}$ by an IO net $\mathcal{N}$ with places $Q_\mathcal{N}$ entails a bijection b from configurations of $\mathcal{R}$ to a subset $\mathcal{G}$ of "good" markings of $\mathcal{N}$. Let C be a configuration of $\mathcal{R}$ and let h be the multiset h over $Q_\mathcal{N} \setminus Q_\mathcal{R}$ given by the simulation. Then $b(C) = C \cdot h$, and $\mathcal{G}$ is the set of markings $b(C)$ for C a configuration of $\mathcal{R}$.

Notice that a cube of $\mathcal{R}$ is mapped by b to a cube of $\mathcal{N}$. Intuitively, the image of a cube $\mathcal{C}'$ of $\mathcal{R}$ is its "extension" with the cube on $Q_\mathcal{N} \setminus Q_\mathcal{R}$ of lower and upper bound equal to h. The norm of $b(\mathcal{C}')$ is $\|b(\mathcal{C}')\| = \|\mathcal{C}'\| + |h|$. A cube $\mathcal{C}$ of $\mathcal{N}$ restricted to $\mathcal{G}$ is equal to the cube $\mathcal{C}|_\mathcal{G} = \mathcal{C} \cap \mathcal{H}$ where $\mathcal{H}$ is the cube of lower bound 0 and upper bound ∞ on $Q_\mathcal{R}$, and upper and lower bounds equal to h on $Q_\mathcal{N} \setminus Q_\mathcal{R}$. The reverse image of this cube by b is the cube of $\mathcal{R}$ in which we "forget" the information of $Q_\mathcal{N} \setminus Q_\mathcal{R}$. The norm of $b^{-1}(\mathcal{C}|_\mathcal{G})$ is smaller or equal to $\|\mathcal{C} \cap \mathcal{H}\| - h \leq \|\mathcal{C}\| + \|\mathcal{H}\| - h = \|\mathcal{C}\|$.

Fix an RBN $\mathcal{R} = (Q, \Sigma, \delta)$ and a cube $\mathcal{C}$ over Q. Let $\mathcal{N} = (Q_\mathcal{N}, \delta_\mathcal{N})$ be an IO net that polynomially simulates $\mathcal{R}$, and let b be the bijection induced from configurations of $\mathcal{R}$ to the subset $\mathcal{G}$ of "good markings" of $\mathcal{N}$. Since b preserves cubes, $b(\mathcal{C})$ is a cube of $\mathcal{N}$. By Theorem 15, $post^*_\mathcal{N}(b(\mathcal{C}))$ is a counting set, and thus there exist cubes $\mathcal{C}_1, \ldots, \mathcal{C}_n$ of $\mathcal{N}$ such that $post^*_\mathcal{N}(b(\mathcal{C})) = \cup_{i=1}^n \mathcal{C}_i$. Let $\mathcal{M}$ be the set $\cup_{i=1}^n b^{-1}(\mathcal{C}_i|_\mathcal{G})$ of configurations of $\mathcal{R}$.

We show that $post^*_\mathcal{R}(\mathcal{C}) = \mathcal{M}$. Let $C \in \mathcal{M}$. There exists i such that $C \in b^{-1}(\mathcal{C}_i|_\mathcal{G})$. Thus $b(C) \in \mathcal{C}_i|_\mathcal{G} \subseteq post^*_\mathcal{N}(b(\mathcal{C}))$. By polynomial simulation, $C \in post^*_\mathcal{R}(\mathcal{C})$. For the other direction of inclusion, consider $C \in post^*_\mathcal{R}(\mathcal{C})$. By polynomial simulation, $b(C) \in post^*_\mathcal{N}(b(\mathcal{C}))$, and thus there exists i such that $b(C) \in \mathcal{C}_i$. By definition, $b(C) \in \mathcal{G}$, so $b(C) \in \mathcal{C}_i|_\mathcal{G}$ and $C \in b^{-1}(\mathcal{C}_i|_\mathcal{G}) \subseteq \mathcal{M}$, concluding our proof of equality.

Since the $b^{-1}(\mathcal{C}_i|_\mathcal{G})$ are cubes, $post^*_\mathcal{R}(\mathcal{C})$ is a counting set as a finite union of cubes. The size of $post^*_\mathcal{N}(b(\mathcal{C}))$ is polynomial in $\mathcal{C}$ and $\mathcal{R}$ by Theorem 15, and thus the size of $post^*_\mathcal{R}(\mathcal{C})$ is too. □

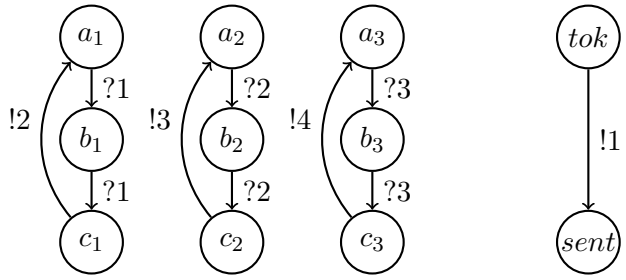


Figure 6.6: *An RBN simulating a counter to* 2^3.

Notice that the result also holds when we replace cube $\mathcal{C}$ by a counting set, since a counting set is a finite union of cubes and $post^*$ of a union is equal to the union of the $post^*$.

Deriving the contradiction. We now exhibit a contradiction to the result of Lemma 71, thus proving that IO nets do not simulate RBN in this way. Consider the RBN represented in Figure 6.6, with set of states $\{tok, sent\} \cup \{a_i, b_i, c_i | 1 \leq i \leq 3\}$. It is inspired by a similar example described in Section 5.1 of [21]. Let $\mathcal{C}_0$ be the cube which puts exactly one process in each a_i, an arbitrary number of processes in tok and 0 processes elsewhere. That is, $\mathcal{C}_0 = (L, U)$ such that $L(a_i) = U(a_i) = 1$ for all i, $L(tok) = 0$ and $U(tok) = \infty$, and $L(q) = U(q) = 0$ for all other states q. Let $\mathcal{C}_f$ be the cube which puts at least one process in c_3 and an arbitrary number elsewhere. Suppose some configuration in $\mathcal{C}_0$ reaches some configuration in $\mathcal{C}_f$. By construction, for a process to reach c_3 it must start in a_3 and receive message 3 twice. For a process to broadcast 3 it must start in a_2 and receive 2 twice, and for a process to broadcast 2 it must start in a_1 and receive 1 twice. So a run from a configuration of $\mathcal{C}_0$ to a configuration of $\mathcal{C}_f$ must contain at least 2^3 broadcasts of 1. Since the only way to broadcast 1 is for a process to go from tok to $sent$, there must be at least 2^3 processes in tok in the initial configuration of $\mathcal{C}_0$.

We can generalize this RBN to a family of RBN $\mathcal{R}_n = (Q, \Sigma, \delta)$, parameterized by $n \geq 1$, with set of states $\{tok, sent\} \cup \{a_i, b_i, c_i | 1 \leq i \leq n\}$. Let $\mathcal{C}_0$ be the cube in which there are arbitrarily many agents in tok, exactly one agent in each a_i and 0 agents in the other states. Let $\mathcal{C}_f$ be the cube in which there is a least one agent in c_n and an arbitrary number elsewhere. We claim that if we start from a configuration of $\mathcal{C}_0$, we can only reach $\mathcal{C}_f$ if we initially have 2^n or more agents in tok. Indeed we can show by induction on $i \in \{1, \ldots, n\}$ that 1 must be broadcasted 2^i times to reach c_i, and thus that 2^i agents are needed in tok initially to reach c_i. By Proposition 1 and Lemma 71, the set $S := post^*(\mathcal{C}_0) \cap \mathcal{C}_f$ is a counting set of size at most polynomial in $|Q|, \|\mathcal{C}_0\|$ and $\|\mathcal{C}_f\|$. The cubes $\|\mathcal{C}_0\|$ and $\|\mathcal{C}_f\|$ have norms n and 1 respectively, so S is of norm polynomial in n. Thus if it is non-empty it must contain a configuration of size at most polynomial in n: simply take a configuration

equal to the lower bounds L of one of the cubes in the union equal to the counting set $post^*(\mathcal{C}_0) \cap \mathcal{C}_f$. This contradicts the fact that 2^n agents are needed to reach $\mathcal{C}_f$.

This gives us the following theorem.

Theorem 72. There exists an RBN that cannot be polynomially simulated by any IO net.

6.2.4 Branching Reconfigurable Broadcast Networks

Inspired by the definition of BIO nets as generalization of IO nets, we define branching RBN, where process creation is allowed upon receiving a broadcast, with the usual reconfigurations of the neighborhood topology. We do a preliminary exploration of this model. In particular, we show that its coverability problem is EXPSPACE-complete. This shows it is more complex than RBN, for which coverability is PSPACE-complete (Theorem 70).

Definition 73. A *branching reconfigurable broadcast network* is a tuple $\mathcal{B} = (Q, \Sigma, \delta)$ where Q is a finite set of states, Σ is a finite alphabet and $\delta \subseteq (Q \times \{!a \mid a \in \Sigma\} \times Q) \cup (Q \times \{?a \mid a \in \Sigma\} \times \mathbb{N}^Q)$ is the transition relation.

Transitions are of the form $t = (p, ?a, \wr p_1, \ldots, p_k \wr)$ for receive transitions, with $k \geq 0$, and $t = (q, !a, q')$ for broadcast transitions. A *configuration* C of a BRBN $\mathcal{B}$ is a multiset over Q which counts the number of agents in each state. Given a letter $a \in \Sigma$ and two configurations C and C' we say that there is a *step* $C \xrightarrow{a} C'$ if there exists a multiset $\wr t, t_1, \ldots, t_k \wr$ of δ for some $k \geq 0$ satisfying the following: $t = q \xrightarrow{!a} q'$, each $t_i = p_i \xrightarrow{?a} \wr p_1^i, \ldots, p_{k_i}^i \wr$, $C \geq \boldsymbol{q} + \sum_i \boldsymbol{p_i}$ and $C' = C - \boldsymbol{q} - \sum_i \boldsymbol{p_i} + \boldsymbol{q'} + \sum_i \boldsymbol{p_1^i} + \ldots + \boldsymbol{p_{k_i}^i}$. We denote by $\xrightarrow{*}$ the reflexive and transitive closure of the step relation, and define the reachability and coverability problems as usual. A *run* is a sequence of steps.

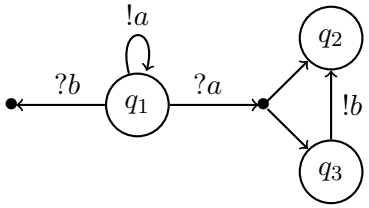


Figure 6.7: *A BRBN.*

Example 31. Figure 6.7 illustrates a BRBN $\mathcal{R} = (Q, \Sigma, \delta)$ with $Q = \{q_1, q_2, q_3\}$ and $\delta = \{t_1, t_2, t_3, t_4\}$, where

$$t_1 = q_1 \xrightarrow{!a} q_1 \qquad t_2 = q_1 \xrightarrow{?a} \wr q_2, q_3 \wr \qquad t_3 = q_3 \xrightarrow{!b} q_1 \qquad t_4 = q_1 \xrightarrow{?b} \emptyset$$

An example run is $(3,0,0) \xrightarrow{t_1+t_2} (2,1,1) \xrightarrow{t_3+2\cdot t_4} (0,2,0)$.

Like RBN, BRBN have the monotonicity property and the copycat property. However they are no longer conservative, since a transition occurring in a run may increase or decrease the number of agents in the configurations of the run.

Lower Bound

We prove that the coverability problem for BRBN protocols is EXPSPACE-hard by reduction from the coverability problem for vector addition systems with states (VASS).

VASS formalism. We fix a VASS $\mathcal{V} = (S, T)$ of dimension $n > 0$, where S is a finite set of states, and $T \subseteq S \times \mathbb{Z}^n \times S$ is a finite set of transitions. A configuration of $\mathcal{V}$ is a pair $(q, \langle c_1, \ldots, c_n \rangle)$, with $q \in S$ and $c_1, \ldots, c_n \in \mathbb{N}$. The c_i are the values of n counters, which must be non negative. Without loss of generality for our reachability queries, we can assume that the transitions of T only affect one counter c at a time, either by $+1$ or -1. If a transition from some state q to some state q' contains several increments or decrements of counters, one can add a path of new transitions and states starting in q and ending in q' such that each new transition only does one increment or one decrement. That is, every $t \in T$ is of the form $(q, \boldsymbol{v}, q')$ where $\boldsymbol{v}$ is equal to 1 or -1 on one index and 0 elsewhere. If this index is the i-th index, we denote t by $(q, inc(c_i), q')$ if its value is 1 and $(q, dec(c_i), q')$ if its value is -1.

Given a transition $t = (q, inc(c_i), q')$ and two configurations C and C', there is a *step* $C \xrightarrow{t} C'$ if $C = (q, \langle c_1, \ldots, c_n \rangle)$ and $C' = (q', \langle c_1, \ldots, c_i + 1, \ldots, c_n \rangle)$. Given a transition $t = (q, dec(c_i), q')$ and two configurations C and C', there is a *step* $C \xrightarrow{t} C'$ if $C = (q, \langle c_1, \ldots, c_n \rangle)$, $c_i > 0$ and $C' = (q', \langle c_1, \ldots, c_i - 1, \ldots, c_n \rangle)$. We denote by $\xrightarrow{*}$ the reflexive and transitive closure of the step relation, and define the reachability and coverability problems as usual. In particular, $C = (q, \langle c_1, \ldots, c_n \rangle)$ covers $C' = (q', \langle c'_1, \ldots, c'_n \rangle)$ if $q = q'$ and $c_i \geq c'_i$ for all i.

BRBN construction. We construct a BRBN $\mathcal{B}_\mathcal{V} = (Q, \Sigma, \delta)$ and a mapping $f : S \times \mathbb{N}^n \to \mathbb{N}^{|Q|}$ from configurations of $\mathcal{V}$ to configurations of $\mathcal{B}_\mathcal{V}$ verifying the following: given configurations C and C' of $\mathcal{V}$, configuration C' is coverable from configuration C if and only if configuration $f(C')$ is coverable from configuration $f(C)$ in $\mathcal{B}_\mathcal{V}$.

Intuitively, Q has a state for each state of S, n states representing the counters of $\mathcal{V}$, as well as some intermediary states. Formally, the set of states Q is comprised of the states of S, a state t for each $t \in T$, states $\perp$ and emp, and for each $i \in \{1, \ldots, n\}$: a state c_i, states $ctrl(c_i)$ and $\overline{ctrl(c_i)}$, and states c_i^t for each $t = (q, c_i, q') \in T$.

Before describing the transition relation δ, we define *good* configurations of $\mathcal{B}_\mathcal{V}$. These are the configurations D such that $\sum_{q \in S} D(q) = 1$, $D(emp) = 1$, $D(ctrl(c_i)) = 1$ and $D(c_i) \geq 0$ for each i, and for any other $q \in Q, D(q) = 0$. Note that there is a natural

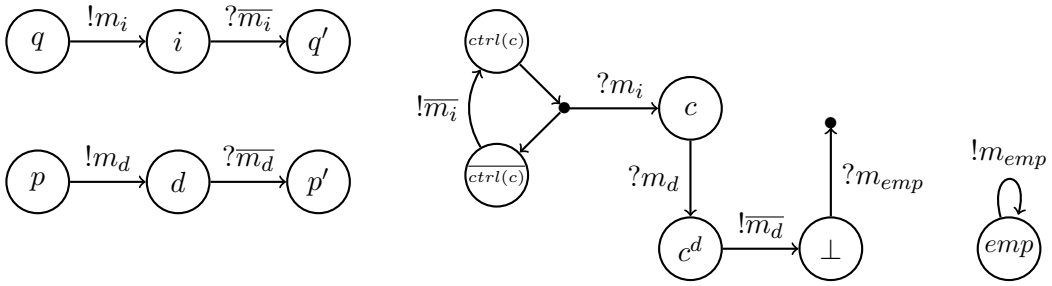


Figure 6.8: *Simulation of a VASS increment i and decrement d.*

bijection between the configurations of $\mathcal{V}$ and the good configurations of $\mathcal{B}_\mathcal{V}$: the one agent in $\sum_{q \in S} D(q)$ designates the state of the VASS configuration, while the number of agents in state c_i designates the value of the counter c_i, for each i. For C a configuration of $\mathcal{V}$, we note $f(C)$ the corresponding good configuration in $\mathcal{B}_\mathcal{V}$. For D a good configuration of $\mathcal{B}_\mathcal{V}$, we note $f^{-1}(D)$ the corresponding configuration in $\mathcal{V}$.

We define δ:

- Let $t = (q, inc(c), q')$ a transition of $\mathcal{V}$. Then δ contains transitions $(q, !m_t, t)$ and $(t, ?\overline{m_t}, \wr q' \wr)$, ensuring that the agent indicating the VASS state moves to the correct place. Additionally, δ contains transitions $(ctrl(c), ?m_t, \wr \overline{ctrl(c)}, c \wr)$ and $(\overline{ctrl(c)}, !\overline{m_t}, ctrl(c))$, ensuring that the state encoding counter c gets one more agent.
- Let $t = (q, dec(c), q')$ a transition of $\mathcal{V}$. Then δ contains transitions $(q, !m_t, \wr t \wr)$ and $(t, ?\overline{m_t}, \wr q' \wr)$, ensuring that the agent indicating the VASS state moves to the correct place. Additionally, δ contains transitions $(c, ?m_t, \wr c^t \wr)$ and $(c^t, !\overline{m_t}, \bot)$, ensuring that at least one agent moves from the state encoding counter c to the $\bot$ state (which can be seen as the "trash" state).
- Finally, δ contains transitions $(emp, !m_{emp}, emp)$ and $(\bot, ?m_{emp}, \emptyset)$. The function of these is to empty the $\bot$ state.

Accordingly, the set of messages of $\mathcal{B}_\mathcal{V}$ is $\Sigma = \{m_t, \overline{m_t} \mid t \in T\} \cup \{m_{emp}\}$. Illustrated in Figure 6.8 is the simulation of transitions $i = (q, inc(c), q')$ and $d = (p, dec(c), p')$.

Lemma 74. Let C, C' two configurations of $\mathcal{V}$. Then C' is coverable from C if and only if $f(C')$ is coverable from $f(C)$ in $\mathcal{B}_\mathcal{V}$.

Proof. If C' is coverable from C, then there exists C'' and a sequence of transitions σ such that $C \xrightarrow{\sigma} C'' \geq C'$. If σ is empty and C covers C', then also $f(C)$ covers $f(C')$, by definition of f. Otherwise, let t be a transition in σ.

- If $t = (q, inc(c), q')$ then we simulate it in $\mathcal{B}_\mathcal{V}$ by a broadcast $(q, !m_t, t)$ with receive $(ctrl(c), ?m_t, \wr \overline{ctrl(c)}, c \wr)$, followed by a broadcast by $(\overline{ctrl(c)}, !\overline{m_t}, ctrl(c))$ with receive $(t, ?\overline{m_t}, \wr q' \wr)$.

- If $t = (q, dec(c), q')$ then we simulate it in $\mathcal{B}_\mathcal{V}$ by a broadcast by $(q, !m_t, t)$ with receive $(c, ?m_t, \wr c^t \wr)$, followed by a broadcast by $(c^t, !\overline{m_t}, \bot)$ with receive $(t, ?\overline{m_t}, \wr q' \wr)$, followed by a broadcast $(emp, !m_{emp}, emp)$ with receive $(\bot, ?m_{emp}, \emptyset)$ to empty the $\bot$ state.

This entails that if $C \xrightarrow{\sigma} C''$ then there exists a run in $\mathcal{B}_\mathcal{V}$ such that $f(C) \xrightarrow{*} f(C'')$. By definition, $f(C'')$ covers $f(C')$.

For the other direction, assume D' is coverable from D for two good configurations D, D' of $\mathcal{B}_\mathcal{V}$. There exists D'' and a run ρ such that $D \xrightarrow{\rho} D'' \geq D'$. If D'' is not a good configuration, we show it can reach an $E \geq D'$ such that E is a good configuration. D'' is reachable from the good configuration D which has $\sum_{q \in S} D(q) = 1$. By construction of $\mathcal{B}_\mathcal{V}$, the number of agents in $S \cup T$ stays constant along a run. Since $D'' \geq D'$, we therefore have $\sum_{q \in S} D''(q) = 1$. Similarly, $D''(ctrl(c)) = 1$ for every counter c and $D''(emp) = 1$, using that D is good, D'' covers D and the number of agents in $ctrl(c) \cup \overline{ctrl(c)}$ stays constant along a run for every counter c. The only states which stop D'' from being good are the c^t and $\bot$. D'' can reach a good configuration E by taking broadcasts $(c^t, !\overline{m_t}, \bot)$ until every c^t is empty, then taking a broadcast $(emp, !m_{emp}, emp)$ received by all the agents in $\bot$. Configuration E covers D' because the above steps do not modify the number of agents in any other state than $\bot$ and the c^t, and $D(c^d) = D(\bot) = 0$. So without loss of generality, we assume in the rest that D'' is a good configuration.

If ρ is empty and D covers D', then also $f^{-1}(D)$ covers $f^{-1}(D')$ by definition. Otherwise, we can decompose ρ as $\rho = \rho_1 \rho_2 \dots \rho_l$ where the ρ_i are sequences of steps starting with $\xrightarrow{m_t}$ for a $t \in T$ and with no other $\xrightarrow{m_s}$ for any $s \in T$ in the rest of ρ_i (the rest of the steps are of the form $\xrightarrow{m_{emp}}$ or $\xrightarrow{\overline{m_u}}$ for $u \in T$). Let σ be the sequence of transitions in the VASS $\mathcal{V}$ equal to $t_1 t_2 \dots t_l$ where t_i is the transition of the first step of ρ_i for each i. We show that there exists a configuration C in $\mathcal{V}$ such that $f^{-1}(D) \xrightarrow{\sigma} C$ and such that C covers $f^{-1}(D'')$. By definition, $f^{-1}(D'')$ covers $f^{-1}(D')$.

Intuitively, our construction of $\mathcal{B}_\mathcal{V}$ ensures that a broadcast of an increment transition i does indeed add an agent to the appropriate counter. On the other hand, a broadcast of a decrement transition d may result in more than one agent being removed from the appropriate counter. In the run σ of $\mathcal{V}$ we take as many decrements as there are broadcasts of decrement transitions in ρ, thus possibly ending in a configuration C with counter values higher than the number of agents in the counters in D'' – this is not a problem for coverability.

All configurations $\tilde{D}$ along ρ are such that $\sum_{q \in S \cup T} \tilde{D}(q) = 1$, since they are reached from the good configuration D and since the number of agents in $S \cup T$ stays constant along a run. Also, all broadcasts $!m_t$ for $t \in T$ go from a state of S to a state of T, and the only transition leaving from a state of $t \in T$ is the receive transition $?\overline{m_t}$. Therefore there is a unique agent in $S \cup T$ which broadcasts the m_t and receives the $\overline{m_t}$. Because

D and D'' are good, this agent is in S in D and in D''. These considerations lead to the following fact.

Fact 75. The number of occurrences of $!m_t$ is smaller or equal to the number of occurrences of $!\overline{m_t}$ in ρ, for every $t \in T$.

All configurations $\tilde{D}$ along ρ are such that $\tilde{D}(ctrl(c)) + \tilde{D}(\overline{ctrl(c)}) = 1$ for every counter c, since they are reached from the good configuration D for which this holds. Thus, for every broadcast $!m_i$ where $i \in T$ is an increment transition, there is at most one receive. If there is no receive, then the unique agent of $S \cup T$ is stuck in i. Therefore there is exactly one receive, which has the effect of adding one agent to the appropriate counter for every broadcast of m_i in ρ.

In the case of a decrement transition $d \in T$, there is no guarantee that m_d will be received, or be received by only one agent. However, notice that an agent can only be in c^d if it received a broadcast of m_d, and only agents in c^d can broadcast $\overline{m_d}$. Since the run starts in D which has 0 agents in c^d and ends in D'' which has 0 agents in c^d, there are as many occurrences of $?m_d$ as occurrences of $!\overline{m_d}$ in ρ. Combining this with Fact 75, the number of occurrences of $!m_d$ is smaller or equal to the number of occurrences of $?m_d$, i.e. there are at least as many agents that leave the appropriate counters as there are broadcasts of the decrement transition m_d.

Let C be the configuration of $\mathcal{V}$ obtained by taking σ from $f^{-1}(D)$. By the considerations above, $C \geq f^{-1}(D'')$, and we are done.

□

Using this reduction lemma and the fact that the coverability problem is in EXPSPACE for VASS [74], we get the announced result.

Theorem 76. The coverability problem for BRBN is EXPSPACE-hard.

Petri nets simulate BRBN

We show that Petri nets can simulate BRBN. The Petri nets coverability problem is in EXPSPACE [37], and the reachability problem is in ACKERMANN [67]. The simulation entails a transfer of complexity upper bounds from Petri nets to BRBN.

Let $\mathcal{B} = (Q, \Sigma, \delta)$ a BRBN. We construct $\mathcal{N} = (P, T, F)$ and a mapping $f : \mathbb{N}^{|Q|} \to \mathbb{N}^{|P|}$ from configurations of $\mathcal{B}$ to markings of $\mathcal{N}$ verifying the following: given configurations C and C' of $\mathcal{B}$, configuration C' is reachable from configuration C if and only if marking $f(C')$ is reachable from marking $f(C)$ in $\mathcal{N}$.

Intuitively, P has a place for each state of Q, a place for each message of Σ and transition that can broadcast it, a place *emp* signifying that no message is currently being broadcasted, as well as some intermediary states. The idea of the simulation is that there is exactly one

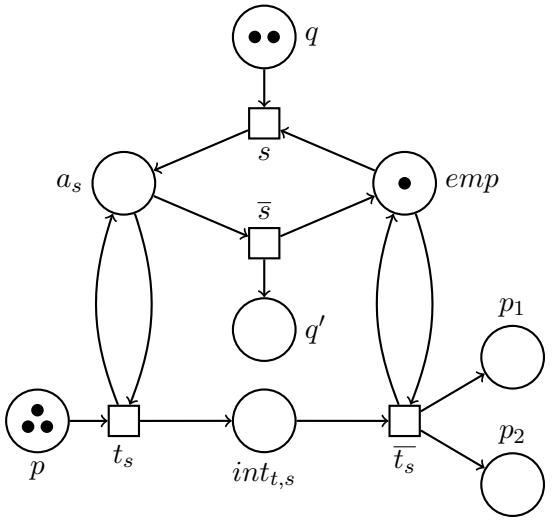


Figure 6.9: *Simulation of a BRBN broadcast s and receive t by a Petri net.*

token among the places representing messages and the *emp* place. When the token is in some place representing message $a \in \Sigma$, places of Q can "receive" the message and move to intermediary places. The token for a then moves to *emp* and the places that "received" a can move to their destinations.

Formally, the set of places P is comprised of a place q for each state $q \in Q$, a place *emp*, and, for each transition $s = (q, !a, q') \in \delta$, a place a_s and places $int_{t,s}$ for each $t \in \delta$ such that t is a receive of a. The set T is defined as follows:

- Let $s = (q, !a, q')$ be a transition of $\mathcal{B}$. Then T contains transition s such that ${}^\bullet s = \wr q, emp \wr$ and $s^\bullet = \wr a_s \wr$, and transition $\overline{s}$ such that ${}^\bullet\overline{s} = \wr a_s \wr$ and $\overline{s}^\bullet = \wr q', emp \wr$.
- Let $t = (p, ?a, \wr p_1, \ldots, p_k \wr)$ for $k \geq 0$ be a transition of $\mathcal{B}$. Then T contains transitions t_s such that ${}^\bullet t_s = \wr p, a_s \wr$ and $t_s{}^\bullet = \wr a_s, int_{t,s} \wr$, and transitions $\overline{t_s}$ such that ${}^\bullet\overline{t_s} = \wr int_{t,s}, emp \wr$ and $\overline{t_s}{}^\bullet = \wr emp, p_1, \ldots, p_k \wr$, for every $s = (q, !a, q') \in \delta$ that broadcasts message a for some q, q'.

Illustrated in Figure 6.9 is the simulation of transitions $s = (q, !a, q')$ and $t = (p, ?a, \wr p_1, p_2 \wr)$. Observe that the only non BIO transitions of this simulating Petri net are the s of the form ${}^\bullet t = \wr q, emp \wr$ and $t^\bullet = \wr a_s \wr$. The fact that *emp* is in the preset of every transition s such that s is a broadcast transition of $\mathcal{B}$ is essential to ensuring that only one broadcast is simulated at a time: between two consecutive occurrences of broadcast transitions s and s' in some firing sequence, there must be an occurrence of $\overline{s}$, which intuitively ends the simulation of broadcast s.

Note that there is a natural bijection between the configurations C of $\mathcal{B}$ and the markings of $\mathcal{N}$ such that there are $C(q)$ many tokens in $q \in Q$, one token in *emp* and 0 elsewhere. We call such markings *good*, and for C a configuration of $\mathcal{B}$, we note $f(C)$ the corresponding

good marking in $\mathcal{N}$. For M a good marking of $\mathcal{N}$, we note $f^{-1}(M)$ the corresponding configuration in $\mathcal{B}$.

Lemma 77. Let C, C' two configurations of $\mathcal{B}$. Then C' is reachable from C if and only if $f(C')$ is reachable from $f(C)$ in $\mathcal{N}$.

Proof. If C' is reachable from C, then there exists a sequence of steps σ such that $C \xrightarrow{\sigma} C'$. If σ is empty, $C = C'$ and $f(C) = f(C')$. Otherwise, let $s + t^{(1)}, \dots, t^{(n)}$ be a step in σ from some D to D', with $s = (q, !a, q')$ and $t^{(i)} = (p, ?a, \wr p_1, \dots, p_{k_i} \wr)$ for $k \geq 0, i \in \{1, \dots, n\}, n \geq 0$. We simulate it in $\mathcal{N}$ by taking $s, t_s^{(1)}, \dots, t_s^{(n)}, \overline{s}, \overline{t_s^{(1)}}, \dots, \overline{t_s^{(n)}}$ from $f(D)$ to $f(D')$. This entails that if $C \xrightarrow{\sigma} C'$ then there exists a firing sequence in $\mathcal{N}$ such that $f(C) \xrightarrow{*} f(C')$.

For the other direction, assume M' is reachable from M for two good markings M, M' of $\mathcal{N}$. There exists a firing sequence ρ such that $M \xrightarrow{\rho} M'$. If ρ is empty, $M = M'$ and $f^{-1}(M) = f^{-1}(M')$. Otherwise, we claim that there exists a sequence of broadcast steps in $\mathcal{B}$ such that $f^{-1}(M) \xrightarrow{*} f^{-1}(M')$.

A run ρ of $\mathcal{N}$ is a *good run* if there exists $\rho_1, \dots, \rho_m \in T^*$, with $m \geq 0$, such that $\rho = \rho_1 \dots \rho_m$ and every ρ_i is a *good step* of the form $s, t_s^{(1)}, \dots, t_s^{(n)}, \overline{s}, \overline{t_s^{(1)}}, \dots, \overline{t_s^{(n)}}$ where s is the name of a broadcast in $\mathcal{B}$ and $n \geq 0$. We claim that if there exists a run between two good markings M, M', then there exists a good run between M and M'.

Let ρ be a run between two good markings M, M'. Place *emp* is in the preset of every broadcast s and in the postset of every $\overline{s}$. This entails that after a transition s is fired, no other transition s' or $\overline{s'}$ can fire until $\overline{s}$ is fired, for any broadcasts s, s' of $\mathcal{B}$. Using this, the construction of $\mathcal{N}$ and the fact that ρ starts in a good marking, we get that ρ is of the form $\sigma_1 \dots \sigma_m$ where each σ_i is of the form $s, t_s^{(1)}, \dots, t_s^{(k)}, \overline{s}, \overline{u_{s_1}^{(1)}}, \dots, \overline{u_{s_l}^{(l)}}$. Since M' is good, it contains no tokens in places of the form $int_{t,s}$. So there are as many occurrences of a receive t_s as there are occurrences of $\overline{t_s}$ in ρ. We associate to each occurrence of a receive t_s in ρ a matching occurrence of $\overline{t_s}$: if there is an occurrence of t_s at index k of ρ, then it is matched with the first unmatched occurrence of $\overline{t_s}$ after index k. Thus each t_s in some σ_i is matched with a $\overline{t_s}$ in a σ_j such that $i \leq j$. If this $\overline{t_s}$ is not in σ_i, i.e. if $i < j$, moving it there does not disable any future firing. Thus the $\sigma_1 \dots \sigma_m$ can be rearranged into good steps $\rho_1 \dots \rho_m$, and ρ into a good run.

This good run directly corresponds to a sequence of broadcast steps in $\mathcal{B}$ such that $f^{-1}(M) \xrightarrow{*} f^{-1}(M')$. □

Thus we can reduce BRBN reachability to Petri net reachability and BRBN coverability to Petri net coverability (since $f(C) \geq f(C')$ if and only if $C \geq C'$ for any configurations C, C' of $\mathcal{B}$). Using this and Theorem 76 we get the announced result.

Theorem 78. The coverability problem for BRBN is EXPSPACE-complete. The reachability problem for BRBN is in ACKERMANN.

6.3 Model Checking

We examine the model checking problem for a **CTL**-style temporal logic and the observation Petri nets studied in this thesis. We use the logic **UB**, for Unified system of Branching time, introduced in [14]. Note that **UB** is a fragment of (and inspiration for) Computational Tree Logic (**CTL**). We derive our results from the article by Esparza on model checking for infinite-state concurrent systems [38]. We recall the notations and results briefly here, but direct the interested reader to the original article for a more complete presentation.

Logic Syntax and Semantics Given a Petri net $\mathcal{N}$, the syntax of **UB** formulas is

$$\varphi ::= \mathbf{true} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid E(t)\varphi \mid EF\varphi \mid EG\varphi$$

where t is a transition of $\mathcal{N}$. The formulas are interpreted on markings of $\mathcal{N}$. $E(t)\varphi$ is satisfied by marking M if t is enabled at M and firing it leads to a marking satisfying φ. $EF\varphi$ is satisfied by M if there exists a firing sequence from M to a marking satisfying φ. $EG\varphi$ is satisfied by M if there exists an infinite firing sequence from M such that all markings along it satisfy φ. The constant **true** is satisfied by all markings, and connectives $\neg$ and $\wedge$ are interpreted in the usual way.

The logics **EF** (respectively **EG**) are obtained from **UB** by removing the operator EG (respectively EF). We extend **UB** to the logic **UB** + **Cube** with additional atomic formulas of the form $\mathbf{Cube}_{\mathcal{C}}$ for each cube $\mathcal{C}$ definable on $\mathcal{N}$. $\mathbf{Cube}_{\mathcal{C}}$ is satisfied on markings belonging to $\mathcal{C}$. The logics **EF** + **Cube** and **EG** + **Cube** are obtained in the same way.

Theorem 4.4 of [38] states that the model checking problem is decidable for **EF** and classes of Petri nets whose reachability relations are effectively semilinear. The reachability relation of a net is the set of pairs (M, M') of markings such that M reaches M' in the net. The reachability relations of a class of nets are effectively semilinear if they are semilinear sets and if there is an algorithm to compute the reachability relation, given a net of the class. It is clear that IO and IMO nets have effectively semilinear reachability relations, notably via our global flatness theorems. Section 5.2 of [38] says that Theorem 4.4 still holds when we add atomic formulas to **EF**, so long as their valuations are expressible in Presburger arithmetic [54]. This is the case of semilinear sets [52], and cubes are semilinear sets (see Section 2.3). So Theorem 4.4 can be applied to **EF** + **Cube** and to IMO nets, from which we get that the model checking problem for **EF** + **Cube** formulas is decidable for IMO nets.

In the following, we show that this also holds for BIMO nets. In fact, we show a stronger result: the model checking problem for **EF** + **Cube** formulas is in EXPSPACE for BIMO nets (and thus also for BIO, IMO and IO nets).

Theorem 79. The model checking problem for **EF** + **Cube** formulas is in EXPSPACE for BIMO nets.

Proof. Let $\mathcal{N}$ be a BIMO net. We consider the problem of whether a given initial marking M of $\mathcal{N}$ satisfies a given formula φ. We define the size of a formula as the length of the binary encoding of the symbols of the formula and the bounds of the cubes in the formula. By induction over the formula size, we can see that a **EF** + **Cube** formula φ of size s represents a counting set $\mathcal{S}_\varphi$ of norm exponential in s and size of $\mathcal{N}$.

- **true** represents (or is satisfied by the markings of) the cube of all markings of $\mathcal{N}$ (lower bounds 0, upper bounds ∞)
- $\neg\varphi$ represents the counting set $\overline{\mathcal{S}_\varphi}$
- $\varphi_1 \wedge \varphi_2$ represents the counting set $\mathcal{S}_{\varphi_1} \wedge \mathcal{S}_{\varphi_2}$
- $E(t)\varphi$ represents the counting set $pre[t](\mathcal{S}_\varphi)$ of markings obtained by backward firing t from some marking of $\mathcal{S}_\varphi$ which covers $t^\bullet$
- $EF\varphi$ represents the counting set $pre^*(\mathcal{S}_\varphi)$

The size of $\mathcal{S}_\varphi$, exponential in the sizes of φ and $\mathcal{N}$, follows from Proposition 1 (on norms of counting sets under boolean operations) and Theorem 61 (the BIMO Closure Theorem). Now we need to verify whether $M \in \mathcal{S}_\varphi$. This can be done in polynomial space in the size of $\mathcal{S}_\varphi$, by the BIMO Generalized Reachability Theorem. Since the size of $\mathcal{S}_\varphi$ is exponential in the sizes of φ and $\mathcal{N}$, this check can be done in exponential space in the input. □

Theorem 4.3 from [38] states that the model checking problem for **EG** and a subclass of BPP nets is undecidable. As BIO nets are a superclass of BPP nets, this implies undecidability of the model checking problem for **EG** and BIO nets. By extension, **UB** and **CTL** model checking is also undecidable for BIO (and BIMO) nets.

Theorem 80. The model checking problem for **EG** and **UB** formulas is undecidable for BIO nets.

This still holds if we add +**Cube** to the logics. On the other hand, model checking is clearly decidable for IO nets and IMO nets: this follows from the fact that there is only a finite set of reachable markings from a given initial marking .

Theorem 81. The model checking problem for **UB** + **Cube** formulas is decidable for IMO nets.

6.4 Summary and Discussion

In this chapter, we look at an assortment of extensions to the theory of observation Petri nets presented in the previous chapters. First, we look at IO and BIO nets in which a

transition may require a multiset of tokens to be observed to be able to fire. We show that the results of IO and BIO nets hold for IMO and BIMO nets up to a factor corresponding to the maximal amount of tokens needed in an observed place for a transition to fire. Next, we consider reconfigurable broadcast networks (RBN), a pre-existing model studied in [16, 17, 26, 33, 34] in which finite-state, anonymous agents communicate by broadcast. IO nets are "nothing but" RBN nets in which the agent broadcasting a message does not change its state. We deduce some results for IO nets coming from RBN theory. We show, however, that some results of IO nets do not hold for RBN, showing that RBN are a more expressive model. Inspired by the IO to BIO generalization, we initiate the study of branching RBN. We show in particular that these have an EXPSPACE-complete coverability problem, in comparison to the PSPACE-completeness of coverability for RBN. Finally, we show results for model-checking of observation Petri nets and the temporal logic **UB** (Unified system of Branching time), a fragment of **CTL**. These are mostly proved by applying results from the article "Decidability of model checking for infinite-state concurrent systems" [38].

We initially studied the link between RBN and IO nets in the hopes that we could apply the same techniques to prove a Generalized Reachability Theorem for RBN. In [10], we published results containing a Closure Theorem and consequent Generalized Reachability Theorem for RBN. The Closure Theorem stated that counting sets were closed under reachability for RBN, and that the norms of the reachability set were *at most* exponential in the norms of the initial counting set (compared to polynomial for IO nets). Unfortunately the proof of the Closure Theorem (Theorem 2 in the paper) contains a mistake. Therefore it is still an open problem whether counting sets are closed under reachability and whether their norm is "small". From this thesis chapter we know that the forward reachability set of a counting set, if it is itself a counting set, must have *at least* exponential norm.

Applications of the BRBN model should be explored, to decide whether it is a model worth studying. The results in this last chapter are meant as a preliminary exploration of the model. In particular, there is room for improvement on the complexity of the reachability problem, which may not be as hard for BRBN as it is for general Petri nets.

Sources. The IMO and BIMO models were defined and studied in an unpublished first version of [77], which can be found at [76]. The results on RBN were published in [11]. The definition of BRBN and the results for them are new and unpublished. Finally, the model checking results appeared only in the unpublished version of [77], available at [76].

7 Conclusion

In this thesis, we introduced the observation feature to describe a restricted form of synchronization in Petri nets. We developed the theory of observation Petri nets, a class of nets we defined syntactically in which the only form of synchronization is observation. We first studied immediate observation nets, where process creation or destruction is prohibited, in Chapter 3. We applied the results to population protocols, a well-studied distributed computing model, in Chapter 4. In Chapter 5, we studied branching immediate observation nets, where process creation or destruction is allowed. Finally, Chapter 6 collects additional results pertaining to the theory of observation Petri nets: the case in which there are multiple observation places and possibly no source place, the link to the model of reconfigurable broadcast networks, and considerations on model checking.

The main analysis problems we studied for observation Petri nets are the class of generalized reachability problems. This class extends reachability queries to possibly infinite sets of markings, which are expressed as unions of cubes. It includes the cube-reachability problem, which given two sets asks if there is a marking in the first set which can reach a marking in the second set. It also includes problems like whether all the markings of a set can reach a marking in another set, or whether an observation Petri net is live from some marking in a given set. We show that this class of problems can be decided in polynomial space for IO nets, and that a slight restriction of this class can be decided in polynomial space for BIO nets. The classic reachability problem asking if a marking can reach another is already in PSPACE for both IO and BIO nets – our results for generalized reachability problems show that it is not harder to check reachability between single markings than it is to check reachability between possibly infinite sets of markings. Thus the observation feature captures a non-trivial property of Petri nets that is particularly interesting for parameterized verification.

A natural continuation of our work is to implement algorithms for verifying observation Petri nets. To decide generalized reachability problems we guess markings for which the number of tokens is bounded by an exponential in the input size, and we check reachability from or to these markings. Constructing and exploring the reachability graph containing markings of this size induces a state-space explosion. Tools often use symbolic verification to avoid this explosion. As already mentioned, tool FAST [13] uses acceleration techniques, computing transitive closures of sequences of transitions. It applies these to symbolic representations of sets of configurations. Another example is the tool UPPAAL for timed-automata [62, 63], which reduces verification of reachability properties to solving constraint systems that are easier to manipulate, and combines this with under-approximation

approaches like randomized explorations of the state-space. For our implementation we can try and combine such methods.

In the case of implementations for IO nets, we can turn to the Peregrine tool [18, 45]. Peregrine is a tool that automatically verifies the correctness of population protocols. The tool uses techniques based on so-called stage graphs, as well as constraint-solving techniques. In Chapter 4, we saw that IO nets can be seen as the underlying structure of IO population protocols, and that the correctness problem for IO population protocols can be expressed as a generalized reachability problem for IO nets. Peregrine was developed for general population protocols. One avenue of research would be to tailor its implementation to the case of IO protocols, using the results on IO net reachability sets from this thesis. Another avenue would be to take the techniques of Peregrine for checking correctness, and extend them to checking generalized reachability problems.

It was shown early on in the study of Petri nets that the reachability problem was EXPSPACE-hard [69]. This motivated studying subclasses with better reachability complexity, and one of the first examples was marked graphs [28]. Other classes followed, like BPP nets and reversible nets. However, all these classes have semilinear reachability sets, both forward and backward. In BIO nets, the reachability sets are not necessarily semilinear. In particular, Hopcroft and Pansiot's well-known example of a Petri net with a non-semilinear forward reachability set (Lemma 2.8 of [56]) is a BIO net. To the best of our knowledge, BIO nets are the first natural class of nets such that this is true and for which reachability and coverability have elementary complexity. In this thesis, we developed verification techniques for BIO nets. These rely in part on BIO nets' local (pre^*) flatness, or equivalently on the semilinearity of their backward reachability sets. A future direction of research could be to identify other Petri net classes with local but not global flatness, and apply the approaches of this thesis to them.

As mentioned before, Petri nets are sometimes used to model chemical reaction networks [4, 5, 59, 86]. In [4], the authors study the persistence property: if every species is present at the start of the reaction, then no species is eliminated in the course of the reaction. Reactions are modeled using Petri nets, with places representing species and transitions representing atomic reactions. The authors show that the persistence property can be checked on the reachability graph of the modelling Petri net. They apply their results to enzymatic mechanisms closely resembling IO nets. It would be interesting to collaborate with researchers working on chemical reaction networks to find out which properties relevant to them can be expressed as generalized reachability problems.

The expectation is that studying the observation feature will help us to better understand Petri nets on the whole, providing intuition and avenues of investigation.

List of Figures

Bibliography

[1] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, and R. L. Rivest. "Time-Space Trade-offs in Population Protocols". In: *Proc. Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2017. DOI: 10.1137/1.9781611974782.169.

[2] D. Alistarh, J. Aspnes, and R. Gelashvili. "Space-Optimal Majority in Population Protocols". In: *Proc. Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2018. DOI: 10.1137/1.9781611975031.144.

[3] D. Alistarh and R. Gelashvili. "Recent Algorithmic Advances in Population Protocols". In: *SIGACT News* 3 (2018). DOI: 10.1145/3289137.3289150.

[4] D. Angeli, P. De Leenheer, and E. D. Sontag. "A Petri net approach to the study of persistence in chemical reaction networks". In: *Mathematical biosciences* 2 (2007).

[5] D. Angeli and S. Manfredi. "A Petri Net approach to consensus in networks with joint-agent interactions". In: *Autom.* (2019). DOI: 10.1016/j.automatica.2019.06.018.

[6] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. "Computation in networks of passively mobile finite-state sensors". In: *Proc. 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 2004. DOI: 10.1145/1011767.1011810.

[7] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. "The computational power of population protocols". In: *Distributed Comput.* 4 (2007). DOI: 10.1007/s00446-007-0040-2.

[8] A. Annichini, A. Bouajjani, and M. Sighireanu. "TReX: A Tool for Reachability Analysis of Complex Systems." In: *CAV*. Ed. by G. Berry, H. Comon, and A. Finkel. Lecture Notes in Computer Science. Springer, 2001. ISBN: 3-540-42345-1.

[9] R. Aris. "Prolegomena to the rational analysis of systems of chemical reactions". In: *Archive for rational mechanics and analysis* 2 (1965).

[10] A. R. Balasubramanian, L. Guillou, and C. Weil-Kennedy. "Parameterized Analysis of Reconfigurable Broadcast Networks". In: *Foundations of Software Science and Computation Structures - 25th International Conference, FOSSACS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings*. Ed. by P. Bouyer and L. Schröder. Lecture Notes in Computer Science. Springer, 2022. DOI: 10.1007/978-3-030-99253-8_4. URL: https://doi.org/10.1007/978-3-030-99253-8%5C_4.

[11] A. R. Balasubramanian and C. Weil-Kennedy. “Reconfigurable Broadcast Networks and Asynchronous Shared-Memory Systems are Equivalent”. In: *Proceedings 12th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2021, Padua, Italy, 20-22 September 2021*. Ed. by P. Ganty and D. Bresolin. EPTCS. 2021. DOI: 10.4204/EPTCS.346.2. URL: https://doi.org/10.4204/EPTCS.346.2.

[12] P. Baldan, N. Cocco, A. Marin, and M. Simeoni. “Petri nets for modelling metabolic pathways: a survey”. In: *Natural Computing* 4 (2010).

[13] S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. “FAST: Fast Acceleration of Symbolic Transition Systems.” In: *CAV*. Ed. by W. A. H. Jr. and F. Somenzi. Lecture Notes in Computer Science. Springer, 2003. ISBN: 3-540-40524-0. URL: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/FAST-cav03.ps.

[14] M. Ben-Ari, A. Pnueli, and Z. Manna. “The Temporal Logic of Branching Time.” In: *Acta Informatica* (1983).

[15] G. Berthelot and R. Terrat. “Petri nets theory for the correctness of protocols”. In: *IEEE Transactions on Communications* 12 (1982).

[16] N. Bertrand and P. Fournier. “Parameterized Verification of Many Identical Probabilistic Timed Processes”. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*. 2013. DOI: 10.4230/LIPIcs.FSTTCS.2013.501.

[17] N. Bertrand, P. Fournier, and A. Sangnier. “Playing with Probabilities in Reconfigurable Broadcast Networks”. In: *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS*. 2014. DOI: 10.1007/978-3-642-54830-7_9.

[18] M. Blondin, J. Esparza, and S. Jaax. “Peregrine: A Tool for the Analysis of Population Protocols.” In: *CAV (1)*. Ed. by H. Chockler and G. Weissenbacher. Lecture Notes in Computer Science. Springer, 2018. ISBN: 978-3-319-96145-3. URL: http://info.usherbrooke.ca/mblondin/papers/BEJ18b.pdf.

[19] B. Boigelot. *The LASH toolset homepage*. 2014. URL: http://www.montefiore.ulg.ac.be/~boigelot/research/lash/index.html.

[20] M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. “Two-variable logic on data words”. In: *ACM Trans. Comput. Log.* 4 (2011). DOI: 10.1145/1970398.1970403.

[21] P. Bouyer, N. Markey, M. Randour, A. Sangnier, and D. Stan. “Reachability in Networks of Register Protocols under Stochastic Schedulers”. In: *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*. Ed. by I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. DOI: 10.4230/LIPIcs.ICALP.2016.106. URL: https://doi.org/10.4230/LIPIcs.ICALP.2016.106.

[22] C. Chaouiya. “Petri net modelling of biological networks”. In: *Briefings in Bioinformatics* 4 (July 2007). ISSN: 1467-5463. DOI: 10.1093/bib/bbm029. eprint: https://academic.oup.com/bib/article-pdf/8/4/210/591466/bbm029.pdf.

[23] P. Chelikani, I. Fita, and P. Loewen. “Diversity of structures and properties among catalases”. In: *Cell. Mol. Life Sci.* (2004).

[24] A. Cheng, J. Esparza, and J. Palsberg. “Complexity Results for 1-Safe Nets”. In: *Theor. Comput. Sci.* 1&2 (1995).

[25] P. Chini, R. Meyer, and P. Saivasan. “Fine-Grained Complexity of Safety Verification”. In: *J. Autom. Reason.* 7 (2020). DOI: 10.1007/s10817-020-09572-x.

[26] P. Chini, R. Meyer, and P. Saivasan. “Liveness in Broadcast Networks”. In: *Networked Systems - 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19-21, 2019, Revised Selected Papers*. Ed. by M. F. Atig and A. A. Schwarzmann. Lecture Notes in Computer Science. Springer, 2019. DOI: 10.1007/978-3-030-31277-0_4. URL: https://doi.org/10.1007/978-3-030-31277-0%5C_4.

[27] S. Christensen, Y. Hirshfeld, and F. Moller. “Decomposability, Decidability and Axiomatisability for Bisimulation Equivalence on Basic Parallel Processes”. In: *LICS*. IEEE Computer Society, 1993. ISBN: 0-8186-3140-6. URL: http://dblp.uni-trier.de/db/conf/lics/lics93.html#ChristensenHM93.

[28] F. Commoner, A. W. Holt, S. Even, and A. Pnueli. “Marked Directed Graphs”. In: *J. Comput. Syst. Sci.* 5 (1971). DOI: 10.1016/S0022-0000(71)80013-2.

[29] H. Comon and Y. Jurski. “Multiple Counters Automata, Safety Analysis and Presburger Arithmetic.” In: *CAV*. Ed. by A. J. Hu and M. Y. Vardi. Lecture Notes in Computer Science. Springer, 1998. ISBN: 3-540-64608-6. URL: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/ComJur-cav98.ps.

[30] P. Czerner, R. Guttenberg, M. Helfrich, and J. Esparza. “Fast and Succinct Population Protocols for Presburger Arithmetic”. In: *1st Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2022, March 28-30, 2022, Virtual Conference*. Ed. by J. Aspnes and O. Michail. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI: 10.4230/LIPIcs.SAND.2022.11. URL: https://doi.org/10.4230/LIPIcs.SAND.2022.11.

[31] W. Czerwinski and L. Orlikowski. "Reachability in Vector Addition Systems is Ackermann-complete". In: *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021. DOI: 10.1109/FOCS52979.2021.00120. URL: https://doi.org/10.1109/FOCS52979.2021.00120.

[32] R. David and H. Alla. "Petri nets for modeling of dynamic systems: A survey". In: *Autom.* 2 (1994). DOI: 10.1016/0005-1098(94)90024-8.

[33] G. Delzanno, A. Sangnier, R. Traverso, and G. Zavattaro. "On the Complexity of Parameterized Reachability in Reconfigurable Broadcast Networks". In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*. Ed. by D. D'Souza, T. Kavitha, and J. Radhakrishnan. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. URL: https://doi.org/10.4230/LIPIcs.FSTTCS.2012.289.

[34] G. Delzanno, A. Sangnier, and G. Zavattaro. "Parameterized Verification of Ad Hoc Networks". In: *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*. Ed. by P. Gastin and F. Laroussinie. Lecture Notes in Computer Science. Springer, 2010. DOI: 10.1007/978-3-642-15375-4_22. URL: https://doi.org/10.1007/978-3-642-15375-4%5C_22.

[35] J. Desel and W. Reisig. "Place/Transition Petri Nets". In: *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*. Ed. by W. Reisig and G. Rozenberg. Lecture Notes in Computer Science. Springer, 1996. DOI: 10.1007/3-540-65306-6_15. URL: https://doi.org/10.1007/3-540-65306-6%5C_15.

[36] R. Elsässer and T. Radzik. "Recent Results in Population Protocols for Exact Majority and Leader Election". In: *Bulletin of the EATCS* (2018).

[37] J. Esparza. "Decidability and Complexity of Petri Net Problems - An Introduction". In: *Petri Nets*. Lecture Notes in Computer Science. Springer, 1996.

[38] J. Esparza. "Decidability of Model Checking for Infinite-State Concurrent Systems." In: *Acta Inf.* 2 (1997).

[39] J. Esparza. "Petri Nets, Commutative Context-Free Grammars, and Basic Parallel Processes." In: *Fundam. Inform.* 1 (1997).

[40] J. Esparza. "Population Protocols: Beyond Runtime Analysis". In: *Reachability Problems - 15th International Conference, RP 2021, Liverpool, UK, October 25-27, 2021, Proceedings*. Ed. by P. C. Bell, P. Totzke, and I. Potapov. Lecture Notes in Computer Science. Springer, 2021. DOI: 10.1007/978-3-030-89716-1_3. URL: https://doi.org/10.1007/978-3-030-89716-1%5C_3.

[41] J. Esparza, P. Ganty, J. Leroux, and R. Majumdar. “Verification of Population Protocols”. In: *CONCUR*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[42] J. Esparza, P. Ganty, J. Leroux, and R. Majumdar. “Verification of population protocols”. In: *Acta Informatica* 2 (2017). DOI: 10.1007/s00236-016-0272-3.

[43] J. Esparza, P. Ganty, and R. Majumdar. “Parameterized Verification of Asynchronous Shared-Memory Systems”. In: *J. ACM* 1 (2016). DOI: 10.1145/2842603.

[44] J. Esparza, P. Ganty, R. Majumdar, and C. Weil-Kennedy. “Verification of Immediate Observation Population Protocols”. In: *CONCUR*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[45] J. Esparza, M. Helfrich, S. Jaax, and P. J. Meyer. “Peregrine 2.0: Explaining Correctness of Population Protocols Through Stage Graphs”. In: *Automated Technology for Verification and Analysis - 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19-23, 2020, Proceedings*. Ed. by D. V. Hung and O. Sokolsky. Lecture Notes in Computer Science. Springer, 2020. DOI: 10.1007/978-3-030-59152-6_32. URL: https://doi.org/10.1007/978-3-030-59152-6%5C_32.

[46] J. Esparza, S. Jaax, M. A. Raskin, and C. Weil-Kennedy. “The complexity of verifying population protocols”. In: *Distributed Comput.* 2 (2021). DOI: 10.1007/s00446-021-00390-x.

[47] J. Esparza, M. A. Raskin, and C. Weil-Kennedy. “Parameterized Analysis of Immediate Observation Petri Nets”. In: *Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings*. Ed. by S. Donatelli and S. Haar. Lecture Notes in Computer Science. Springer, 2019. DOI: 10.1007/978-3-030-21571-2_20. URL: https://doi.org/10.1007/978-3-030-21571-2%5C_20.

[48] M. Feinberg. “Foundations of chemical reaction network theory”. In: (2019).

[49] Y. Feng, R. Martins, Y. Wang, I. Dillig, and T. W. Reps. “Component-based synthesis for complex APIs”. In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. 2017.

[50] L. Fribourg. “Petri Nets, Flat Languages and Linear Arithmetic”. In: *Proceedings of the 9th International Workshop on Functional and Logic Programming (WFLP 2000)*. Ed. by M. Alpuente. Benicassim, Spain: Universidad Politécnica de Valencia, Spain, Sept. 2000. URL: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Fri-wflp00.ps.

[51] H. J. Genrich and K. Lautenbach. “Synchronisationsgraphen”. In: *Acta Informatica* 2 (1973).

[52] S. Ginsburg and E. H. Spanier. “Bounded Algol-Like Languages”. In: *Transactions of the American Mathematical Society* 2 (1964). ISSN: 00029947. (Visited on 10/31/2022).

[53] Z. Guo, M. James, D. Justo, J. Zhou, Z. Wang, et al. “Program synthesis by type-guided abstraction refinement”. In: *Proc. ACM Program. Lang.* POPL (2020). DOI: 10.1145/3371080.

[54] C. Haase. “A survival guide to presburger arithmetic”. In: *ACM SIGLOG News* 3 (2018). DOI: 10.1145/3242953.3242964.

[55] M. Hack. “Decidability questions for Petri Nets”. PhD thesis. Massachusetts Institute of Technology, Cambridge, MA, USA, 1976. URL: https://hdl.handle.net/1721.1/27441.

[56] J. E. Hopcroft and J.-J. Pansiot. “On the Reachability Problem for 5-Dimensional Vector Addition Systems.” In: *Theor. Comput. Sci.* (1979).

[57] N. D. Jones, L. H. Landweber, and Y. E. Lien. “Complexity of Some Problems in Petri Nets.” In: *Theoretical Computer Science* 3 (1977). DOI: 10.1016/0304-3975(77)90014-7.

[58] M. I. Kanovich. “Petri Nets, Horn Programs, Linear Logic and Vector Games”. In: *Ann. Pure Appl. Log.* 1-2 (1995). DOI: 10.1016/0168-0072(94)00060-G.

[59] I. Koch. “Petri nets–a mathematical formalism to analyze chemical reaction networks”. In: *Molecular Informatics* 12 (2010).

[60] I. Koch, W. Reisig, and F. Schreiber. *Modeling in systems biology: the Petri net approach*. Springer science & business media, 2010.

[61] D. Kozen. *Theory of Computation*. Texts in Computer Science. Springer, 2006. ISBN: 978-1-84628-297-3. DOI: 10.1007/1-84628-477-5.

[62] K. G. Larsen, P. Pettersson, and W. Yi. “Compositional and Symbolic Model-Checking of Real-Time Systems”. In: *16th IEEE Real-Time Systems Symposium, Palazzo dei Congressi, Via Matteotti, 1, Pisa, Italy, December 4-7, 1995, Proceedings*. IEEE Computer Society, 1995. DOI: 10.1109/REAL.1995.495198. URL: https://doi.org/10.1109/REAL.1995.495198.

[63] K. G. Larsen, P. Pettersson, and W. Yi. “UPPAAL in a Nutshell”. In: *Int. J. Softw. Tools Technol. Transf.* 1-2 (1997). DOI: 10.1007/s100090050010.

[64] S. Lasota. “EXPSPACE lower bounds for the simulation preorder between a communication-free Petri net and a finite-state system”. In: *Inf. Process. Lett.* 15 (2009).

[65] J. Leroux. “Presburger Vector Addition Systems.” In: *LICS*. IEEE Computer Society, 2013. ISBN: 978-1-4799-0413-6. URL: http://dblp.uni-trier.de/db/conf/lics/lics2013.html#Leroux13.

[66] J. Leroux. “The Reachability Problem for Petri Nets is Not Primitive Recursive”. In: *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021. DOI: 10.1109/FOCS52979.2021.00121. URL: https://doi.org/10.1109/FOCS52979.2021.00121.

[67] J. Leroux and S. Schmitz. “Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension”. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 2019. DOI: 10.1109/LICS.2019.8785796. URL: https://doi.org/10.1109/LICS.2019.8785796.

[68] J. Leroux and G. Sutre. “Flat Counter Automata Almost Everywhere!” In: *ATVA*. Ed. by D. A. Peled and Y.-K. Tsay. Lecture Notes in Computer Science. Springer, 2005. ISBN: 3-540-29209-8. URL: https://hal.archives-ouvertes.fr/hal-00346310/document.

[69] R. Lipton. *The reachability problem is exponential-space hard*. Tech. rep. 62. Department of Computer Science, Yale University, Jan. 1976.

[70] N. Lohmann, E. Verbeek, and R. Dijkman. “Petri Net Transformations for Business Processes – A Survey”. In: *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems*. Ed. by K. Jensen and W. M. P. van der Aalst. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN: 978-3-642-00899-3. DOI: 10.1007/978-3-642-00899-3_3. URL: https://doi.org/10.1007/978-3-642-00899-3_3.

[71] W. Marwan, A. Wagler, and R. Weismantel. “Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks”. In: *Natural Computing* 2 (2011).

[72] E. W. Mayr and J. Weihmann. “Complexity Results for Problems of Communication-Free Petri Nets and Related Formalisms”. In: *Fundam. Inform.* 1 (2015).

[73] T. Murata. “Petri nets: Properties, analysis and applications”. In: *Proc. IEEE* 4 (1989). DOI: 10.1109/5.24143.

[74] C. Rackoff. “The Covering and Boundedness Problems for Vector Addition Systems.” In: *Theor. Comput. Sci.* (1978). DOI: 10.1016/0304-3975(78)90036-1.

[75] M. Raskin and C. Weil-Kennedy. “Efficient Restrictions of Immediate Observation Petri Nets”. In: *Reachability Problems - 14th International Conference, RP 2020, Paris, France, October 19-21, 2020, Proceedings*. Ed. by S. Schmitz and I. Potapov. Lecture Notes in Computer Science. Springer, 2020. DOI: 10.1007/978-3-030-61739-4_7. URL: https://doi.org/10.1007/978-3-030-61739-4%5C_7.

[76] M. A. Raskin and C. Weil-Kennedy. “On the Flatness of Immediate Observation Petri Nets”. In: *CoRR* (2020). arXiv: 2001.09966v1.

[77] M. Raskin, C. Weil-Kennedy, and J. Esparza. “Flatness and Complexity of Immediate Observation Petri Nets”. In: *31st International Conference on Concurrency Theory (CONCUR 2020)*. Ed. by I. Konnov and L. Kovács. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. ISBN: 978-3-95977-160-3. DOI: 10.4230/LIPIcs.CONCUR.2020.45. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12857.

[78] W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science. Springer, 1985. ISBN: 3-540-13723-8. DOI: 10.1007/978-3-642-69968-9.

[79] G. Rozenberg and J. Engelfriet. “Elementary Net Systems”. In: *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*. Ed. by W. Reisig and G. Rozenberg. Lecture Notes in Computer Science. Springer, 1996. DOI: 10.1007/3-540-65306-6_14. URL: https://doi.org/10.1007/3-540-65306-6%5C_14.

[80] G. Rozenberg and P. S. Thiagarajan. “Petri Nets: Basic Notions, Structure, Behaviour”. In: *Current Trends in Concurrency, Overviews and Tutorials*. Ed. by J. W. de Bakker, W. P. de Roever, and G. Rozenberg. Lecture Notes in Computer Science. Springer, 1986. DOI: 10.1007/BFb0027048.

[81] S. Schmitz. “Complexity Hierarchies beyond Elementary”. In: *ACM Trans. Comput. Theory* 1 (2016). DOI: 10.1145/2858784.

[82] S. Schmitz. “The complexity of reachability in vector addition systems”. In: *ACM SIGLOG News* 1 (2016). DOI: 10.1145/2893582.2893585.

[83] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. “Computation with finite stochastic chemical reaction networks”. In: *Nat. Comput.* 4 (2008). DOI: 10.1007/s11047-008-9067-y.

[84] J. Valusek and P. Jancar. “Structural Liveness of Immediate Observation Petri Nets”. In: *CoRR* (2021). arXiv: 2112.15524.

[85] H. Yen. “On Reachability Equivalence for BPP-Nets”. In: *Theor. Comput. Sci.* 1-2 (1997).

[86] I. Zevedei-Oancea and S. Schuster. “Topological analysis of metabolic networks based on Petri net theory”. In: *Silico Biol.* 3 (2003).

[87] R. Zurawski and M. Zhou. “Petri nets and industrial applications: A tutorial”. In: *IEEE Trans. Ind. Electron.* 6 (1994). DOI: 10.1109/41.334574.