



Checking deadlock-freedom of parametric component-based systems

Marius Bozga, Radu Iosif, Joseph Sifakis

► To cite this version:

Marius Bozga, Radu Iosif, Joseph Sifakis. Checking deadlock-freedom of parametric component-based systems. *Journal of Logical and Algebraic Methods in Programming*, 2020, 119, pp.100621. 10.1016/j.jlamp.2020.100621 . hal-03040227

HAL Id: hal-03040227

<https://hal.science/hal-03040227>

Submitted on 14 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Checking Deadlock-Freedom of Parametric Component-Based Systems

Marius Bozga, Radu Iosif, Joseph Sifakis

Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000 Grenoble France

Abstract

We propose an automated method for computing inductive invariants used to proving deadlock freedom of parametric component-based systems. The method generalizes the approach for computing structural trap invariants from bounded to parametric systems with general architectures. It symbolically extracts trap invariants from interaction formulae defining the system architecture. The paper presents the theoretical foundations of the method and proves its soundness. It also reports on a preliminary experimental evaluation on several textbook examples.

Keywords: parametric systems, deadlock-freedom, invariants, trap invariants, interaction logic, cardinality constraints, quantifier elimination, positivation

1. Introduction

Modern computing systems exhibit dynamic and reconfigurable behavior. To tackle the complexity of such systems, engineers extensively use architectures that enforce, by construction, essential properties, such as fault tolerance or mutual exclusion. Architectures can be viewed as parametric operators that take as arguments instances of components of given types and enforce a characteristic property. For instance, client-server architectures enforce atomicity and resilience of transactions, for any numbers of clients and servers. Similarly, token-ring architectures enforce mutual exclusion between any number of components in the ring.

Parametric verification is an extremely relevant and challenging problem in systems engineering. In contrast to the verification of bounded systems, consisting of a known set of components, there exist no general methods and tools successfully applied to parametric systems. Verification problems for very simple parametric systems, even with finite-state components, are typically intractable [1, 2]. Most work in this area puts emphasis on limitations determined mainly by three criteria (1) the topology of the architecture, (2) the coordination primitives, and (3) the properties to be verified.

Email addresses: Marius.Bozga@univ-grenoble-alpes.fr (Marius Bozga),
Radu.Iosif@univ-grenoble-alpes.fr (Radu Iosif), Joseph.Sifakis@univ-grenoble-alpes.fr (Joseph Sifakis)

¹Institute of Engineering Univ. Grenoble Alpes

Seminal works in parameterized verification consider *rendez-vous* communication, with participants placed in a ring [3, 4] or a clique [1] of arbitrary size. Recently, classes of graphs (with bounded tree- and clique-width) definable in monadic second order logic and point-to-point rendez-vous communication have been considered [5]. Most approaches to define decidable problems focus on manually proving a *cut-off* bound $c \geq 2$ such that correctness for at most c processes implies correctness for any number of processes [3, 4, 6, 7, 8]. Other methods identify systems with well-structured transition relations [1, 9, 10]. An exhaustive chart of decidability results for verification of parameterized systems is drawn in [2]. When decidability is not of concern, over-approximation and semi-algorithmic techniques such as *regular model checking* [11, 12], SMT-based *bounded model checking* [13, 14], *abstraction* [15, 16] and *automata learning* [17] can be used to deal with more general classes of systems.

The efficiency of a verification method crucially relies on its ability to synthesize an *inductive safety invariant*, i.e., an infinite set of configurations that contains the initial configurations, is closed under the transition relation, and excludes the error configurations. In general, automatically synthesizing invariants requires computationally expensive fixpoint iterations [18]. In the particular case of parameterized systems, invariants can be either *global*, relating the local states of all processes [19], or *modular*, relating the local states of a bounded number of processes, whose identities are irrelevant [20, 21].

In contrast with existing invariant synthesis methods for parameterized verification, we synthesize parameterized invariants directly from the interaction formula of a system, without iterating its transition relation. Such invariants depend only on the structure (and not on the operational semantics) of an infinite family of Petri Nets, one for each instance of the system, and are thus *structural* invariants. Essentially, the invariants we infer use the *traps*² of the system, which are sets W of local states with the property that, if a process is initially in a state from W , then always some process will be in a state from W . Following [23], we call them (parameterized) *trap invariants*. Computing trap invariants can be done via a syntactic transformation of the logical formula describing the architecture of the system and the result is expressed using the quantifier-free fragment of boolean algebra with cardinality constraints [24]. Thus invariant computation is fairly cheap and the verification problem (proving the emptiness of the intersection between the invariant and the set of error states) is reduced to the unsatisfiability of a cardinality constraint. In practice, this check can be carried out quite efficiently by existing tools, such as CVC4 [25].

We briefly describe our approach below. A system is the composition of a finite number of component instances of given types, using interactions that follow the Behaviour-Interaction-Priorities (BIP) paradigm [26]. To simplify the technical part, we assume that components and interactions are finite abstractions of real-life systems. An instance is a finite-state transition system whose edges are labeled by ports. The instances communicate synchronously via a number of simultaneous interactions involving a set of ports each, such that no data is exchanged during interactions. If the number of instances in the system is fixed and known in advance, we say that the system is *bounded*, otherwise it is *parametric*.

For instance, the bounded system in Figure 1a consist of component types *Semaphore*, with one instance, and *Task*, with two instances. A semaphore goes from the free state

²Called in this way by analogy with the notion of traps for Petri Nets [22].

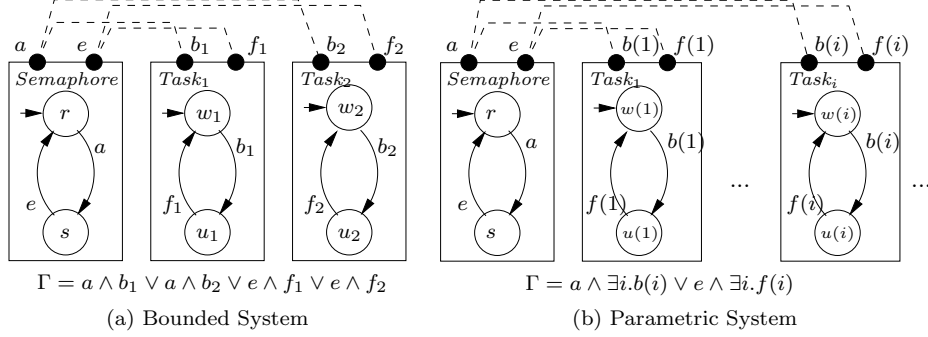


Figure 1: Mutual Exclusion Example

r to the taken state s by an acquire action a , and viceversa from s to r by a release action e . A task goes from waiting w to busy u by action b and viceversa, by action f . For the bounded system in Figure 1a, the interactions are $\{a, b_1\}, \{a, b_2\}, \{e, f_1\}$ and $\{e, f_2\}$, depicted with dashed lines. Since the number of instances is known in advance, we can view an interaction as a minimal satisfying valuation of the boolean formula $\Gamma = (a \wedge b_1) \vee (a \wedge b_2) \vee (e \wedge f_1) \vee (e \wedge f_2)$, where the port symbols are propositional variables. Because every instance has finitely many states, we can write a boolean formula $\Delta = [\neg r \vee \neg(w_1 \vee w_2)] \wedge [\neg s \vee \neg(u_1 \vee u_2)]$, this time over propositional state variables, which defines the configurations in which all interactions are disabled (deadlock). Proving that no deadlock configuration is reachable from the initial configuration $r \wedge w_1 \wedge w_2$, requires finding an over-approximation (invariant) I of the reachable configurations, such that the conjunction $I \wedge \Delta$ is not satisfiable.

The basic idea of our method, supported by the D-FINDER deadlock detection tool [23] for bounded component-based systems, is to compute an invariant straight from the interaction formula, without going through costly abstract fixpoint iterations. The invariants we are looking for are in fact solutions of a system of boolean constraints $\Theta(\Gamma)$, of size linear in the size of Γ (written in DNF). These constraints capture the trap condition, namely that for each interaction in the system, if the pre-state of some port involved in the interaction is in the invariant, then the post-state of some (not necessarily the same) port of the interaction will be in the invariant. In our example, $\Theta(\Gamma) = \bigwedge_{i=1,2} (r \vee w_i) \leftrightarrow (s \vee u_i)$. Finding the (minimal) solutions of this constraint can be done, as currently implemented in D-FINDER, by exhaustive model enumeration using a SAT solver. Here we propose a more efficient solution, which consists in writing $\Theta(\Gamma)$ in DNF and remove the negative literals from each minterm. In our case, this gives the invariant $I = (r \vee s) \wedge \bigwedge_{i=1,2} (w_i \vee u_i) \wedge (r \vee u_1 \vee u_2) \wedge (s \vee w_1 \vee w_2)$ and $I \wedge \Delta$ is proved unsatisfiable using a SAT solver.

The main contribution of this paper is the generalization of this invariant generation method to the parametric case. To understand the problem, consider the parametric system from Figure 1, in which a *Semaphore* interacts with n *Tasks*, where $n > 0$ is not known in advance. The interactions are described by a fragment of first order logic, in which the ports are either propositional or monadic predicate symbols, in our case $\Gamma = [\exists i. (a \wedge b(i))] \vee [\exists i. (e \wedge f(i))]$. This logic, called *Monadic Interaction Logic*

(MIL), is also used to express the constraints $\Theta(\Gamma)$ and compute their solutions. In our case, we obtain $I = (r \vee s) \wedge [\forall i . w(i) \vee u(i)] \wedge [r \vee \exists i . u(i)] \wedge [s \vee \exists i . w(i)]$. As in the bounded case, we can give a parametric description of deadlock configurations $\Delta = [\neg r \vee \neg \exists i . w(i)] \wedge [\neg s \vee \neg \exists i . u(i)]$ and prove that $I \wedge \Delta$ is unsatisfiable, using the decidability of MIL, based on an early small model property result due to Löwenheim [27]. In practice, we avoid the model enumeration suggested by this result and check the satisfiability of such queries using a decidable theory of sets with cardinality constraints [24], available in the CVC4 SMT solver [28].

The use of traps as invariants of the system, useful in proving safety properties, such as absence of deadlocks or mutual exclusion violations, is tightly connected to a semantic model that views a concurrent component-based system as a 1-safe Petri net, in which each place represents a local control state of a component and the presence of (at most) one token in the place indicates that the component is in that control state. In contrast, existing approaches [1] consider general Petri nets, or equivalently vector addition systems, in which the tokens in a place represent the components that are in that control state. We consider 1-safe Petri nets of unbounded (parametric) size, in order to define the trap invariant by a first-order logic constraint, as a natural generalization of the propositional (boolean) constraint used to define traps in a bounded-size 1-safe Petri net [22].

The paper is structured as follows: §2 presents existing results for checking deadlock-freedom of bounded systems using invariants, §3 formalizes the approach for computing invariants using MIL, §4 introduces cardinality constraints for invariant generation, §5 presents the integration of the above results within a verification technique for parametric systems and §6 reports on experiments carried out with a prototype tool. Finally, §7 presents concluding remarks and future work directions.

A short version of the paper has been published in [29]. This extended version includes the proofs for all technical results related to cardinality constraints, some of them being non-trivial to obtain and potentially useful for applications beyond the scope of this paper.

2. Bounded Component-based Systems

In this section, we recall the definition of bounded component-based systems, their execution semantics based on 1-safe Petri nets and the trap invariant synthesis method in this context [23]. The next section (3) is concerned with the generalization of these results to the parametric case, in which the number of components is finite but unknown.

A *component* is a tuple $\mathcal{C} = \langle P, S, s_0, \Delta \rangle$, where $P = \{p, q, r, \dots\}$ is a finite set of *ports*, S is a finite set of *states*, $s_0 \in S$ is an initial state and $\Delta \subseteq S \times P \times S$ is a set of *transitions* written $s \xrightarrow{p} s'$. To simplify the technical details, we assume there are *no two different transitions with the same port*, i.e. if $s_1 \xrightarrow{p_1} s'_1, s_2 \xrightarrow{p_2} s'_2 \in \Delta$ and $s_1 \neq s_2$ or $s'_1 \neq s'_2$ then $p_1 \neq p_2$. In general, this restriction can be lifted, at the cost of cluttering the presentation.

A *bounded system* $\mathcal{S} = \langle \mathcal{C}^1, \dots, \mathcal{C}^n, \Gamma \rangle$ consists of a fixed number (n) of components $\mathcal{C}^k = \langle P^k, S^k, s_0^k, \Delta^k \rangle$ and an *interaction formula* Γ , describing the allowed interactions. Since the number of components is known in advance, we write interaction formulae using boolean logic over the set of propositional variables $\text{BVar} \stackrel{\text{def}}{=} \bigcup_{k=1}^n (P^k \cup S^k)$.

A *boolean interaction formula* is either $a \in \mathbf{BVar}$, $f_1 \wedge f_2$ or $\neg f_1$, where f_i are formulae, for $i = 1, 2$, respectively. We define the usual shorthands $f_1 \vee f_2 \stackrel{\text{def}}{=} \neg(\neg f_1 \wedge \neg f_2)$, $f_1 \rightarrow f_2 \stackrel{\text{def}}{=} \neg f_1 \vee f_2$, $f_1 \leftrightarrow f_2 \stackrel{\text{def}}{=} (f_1 \rightarrow f_2) \wedge (f_2 \rightarrow f_1)$. A literal is either a variable or its negation and a minterm is a conjunction of literals. A formula is in disjunctive normal form (DNF) if it is written as $\bigvee_{i=1}^n \bigwedge_{j=1}^{m_i} \ell_{ij}$, where ℓ_{ij} is a literal. A formula is *positive* if and only if each variable occurs under an even number of negations, or, equivalently, its DNF forms contains no negative literals. We assume interaction formulae of bounded systems to be always positive. As will be explained next, this restriction loses no generality.

A *boolean valuation* $\beta : \mathbf{BVar} \rightarrow \{\top, \perp\}$ maps each propositional variable to either true (\top) or false (\perp). We write $\beta \models f$ if and only if $f = \top$, when replacing each boolean variable a with $\beta(a)$ in f . We say that β is a *model* of f in this case and write $f \equiv g$ for $\llbracket f \rrbracket = \llbracket g \rrbracket$, where $\llbracket f \rrbracket \stackrel{\text{def}}{=} \{\beta \mid \beta \models f\}$. Given two valuations β_1 and β_2 we write $\beta_1 \subseteq \beta_2$ if and only if $\beta_1(a) = \top$ implies $\beta_2(a) = \top$, for each variable $a \in \mathbf{BVar}$. We write $f \equiv^\mu g$ for $\llbracket f \rrbracket^\mu = \llbracket g \rrbracket^\mu$, where $\llbracket f \rrbracket^\mu \stackrel{\text{def}}{=} \{\beta \in \llbracket f \rrbracket \mid \text{for all } \beta' : \beta' \subseteq \beta \text{ and } \beta' \neq \beta \text{ only if } \beta' \notin \llbracket f \rrbracket\}$ is the set of minimal models of f .

In the rest of this section, we fix a bounded system $\mathcal{S} = \langle \mathcal{C}^1, \dots, \mathcal{C}^n, \Gamma \rangle$, where $\mathcal{C}^k = \langle \mathbf{P}^k, \mathbf{S}^k, s_0^k, \Delta^k \rangle$, for all $k \in [1, n]$ and Γ is a positive boolean formula, over propositional variables denoting ports. The semantics of an interaction formula is given by the set of its minimal models. For this reason, it is sufficient to consider positive formulae only, for the description of the interactions in a bounded component-based system.

2.1. Execution Semantics of Bounded Systems

We use 1-safe marked Petri Nets to define the set of executions of a bounded system. A *Petri Net* (PN) is a tuple $N = \langle S, T, E \rangle$, where S is a set of *places*, T is a set of *transitions*, $S \cap T = \emptyset$, and $E \subseteq S \times T \cup T \times S$ is a set of *edges*. The elements of $S \cup T$ are called *nodes*. Given nodes $m, n \in S \cup T$, we write $E(m, n) \stackrel{\text{def}}{=} 1$ if $(m, n) \in E$ and $E(m, n) \stackrel{\text{def}}{=} 0$, otherwise. For a node n , let $\bullet n \stackrel{\text{def}}{=} \{m \in S \cup T \mid E(m, n) = 1\}$, $n^\bullet \stackrel{\text{def}}{=} \{m \in S \cup T \mid E(n, m) = 1\}$ and lift these definitions to sets of nodes, as usual.

A *marking* for a PN $N = \langle S, T, E \rangle$ is a function $m : S \rightarrow \mathbb{N}$. A transition t is *enabled* in m if and only if $m(s) > 0$ for each place $s \in \bullet t$. The *transition relation* of N is defined as follows. For all markings m, m' and all transitions t , we write $m \xrightarrow{t} m'$ whenever t is enabled in m and $m'(s) = m(s) - E(s, t) + E(t, s)$, for all $s \in S$. Given two markings m and m' , a finite sequence of transitions $\sigma = t_1, \dots, t_n$ is a *firing sequence*, written $m \xrightarrow{\sigma} m'$ if and only if either (i) $n = 0$ and $m = m'$, or (ii) $n \geq 1$ and there exist markings m_1, \dots, m_{n-1} such that $m \xrightarrow{t_1} m_1 \dots m_{n-1} \xrightarrow{t_n} m'$.

A *marked Petri net* is a pair $\mathcal{N} = (N, m_0)$, where m_0 is the *initial marking* of $N = \langle S, T, E \rangle$. A marking m is *reachable* in \mathcal{N} if and only if there exists a firing sequence σ such that $m_0 \xrightarrow{\sigma} m$. We denote by $\mathcal{R}(\mathcal{N})$ the set of reachable markings of \mathcal{N} . A set of markings \mathcal{M} is an *invariant* of $\mathcal{N} = (N, m_0)$ if and only if $m_0 \in \mathcal{M}$ and for each $m \xrightarrow{t} m'$ such that $m \in \mathcal{M}$, we have $m' \in \mathcal{M}$. A marked PN \mathcal{N} is *1-safe* if $m(s) \leq 1$, for each $s \in S$ and each $m \in \mathcal{R}(\mathcal{N})$. In the following, we consider only 1-safe marked PNs. In this case, any (necessarily finite) set of reachable markings can be defined by a boolean formula, which identifies markings with the induced boolean valuations.

A marking m is a *deadlock* if for no marking m' and no transition t , do we have $m \xrightarrow{t} m'$. Let $\mathcal{D}(\mathcal{N})$ be the set of deadlocks of N . A marked PN \mathcal{N} is *deadlock-free* if and

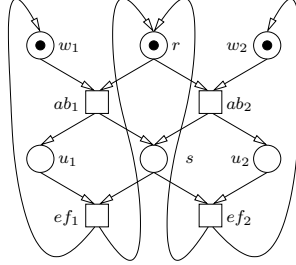


Figure 2: Petri Net for Mutual Exclusion Example

only if $\mathcal{R}(\mathcal{N}) \cap \mathcal{D}(\mathcal{N}) = \emptyset$. A sufficient condition for deadlock freedom is $\mathcal{M} \cap \mathcal{D}(\mathcal{N}) = \emptyset$, for some invariant \mathcal{M} of \mathcal{N} .

The set of executions of the bounded system \mathcal{S} is given by the 1-safe marked PN $\mathcal{N}_{\mathcal{S}} = (N, m_0)$, where $N = (\bigcup_{i=1}^n S^i, T, E)$, $m_0(s) = 1$ if and only if $s \in \{s_0^i \mid i \in [1, n]\}$ and T, E are as follows. For each minimal model $\beta \in \llbracket \Gamma \rrbracket^\mu$, we have a transition $t_\beta \in T$ and edges $(s_i, t_\beta), (t_\beta, s'_i) \in E$, for all $i \in [1, n]$ such that $s_i \xrightarrow{p_i} s'_i \in \Delta^i$ and $\beta(p_i) = \top$. Moreover, nothing else is in T or E .

For example, the marked PN from Figure 2 describes the set of executions of the bounded system from Figure 1a. Note that each transition of the PN corresponds to a minimal model of the interaction formula $\Gamma = (a \wedge b_1) \vee (a \wedge b_2) \vee (e \wedge f_1) \vee (e \wedge f_2)$, or equivalently, to the set of (positive) literals of some minterm in the DNF of Γ .

2.2. Proving Deadlock Freedom of Bounded Systems

A bounded system \mathcal{S} is deadlock-free if and only if its corresponding marked PN $\mathcal{N}_{\mathcal{S}}$ is deadlock-free. In the following, we prove deadlock-freedom of a bounded system, by defining a class of invariants that are particularly useful for excluding unreachable deadlock markings.

Given a Petri Net $N = (S, T, E)$, a set of places $W \subseteq S$ is called a *trap* if and only if $W^\bullet \subseteq {}^\bullet W$. A trap W of N is a *marked trap* of the marked PN $\mathcal{N} = (N, m_0)$ if and only if $m_0(s) = 1$ for some $s \in W$. A *minimal marked trap* is a marked trap such that none of its strict subsets is a marked trap. A marked trap defines an invariant of the PN because some place in the trap will always be marked, no matter which transition is fired. The *trap invariant* of \mathcal{N} is the least set of markings that mark each trap of \mathcal{N} . Clearly, the trap invariant of \mathcal{N} subsumes the set of reachable markings of \mathcal{N} , because the latter is the least invariant of \mathcal{N} and invariants are closed under intersection³.

Lemma 1. *Given a bounded system \mathcal{S} , the boolean formula:*

$$\text{Trap}(\mathcal{N}_{\mathcal{S}}) \stackrel{\text{def}}{=} \bigwedge \{ \bigvee_{i=1}^k s_i \mid \{s_1, \dots, s_k\} \text{ is a marked trap of } \mathcal{N}_{\mathcal{S}} \}$$

defines an invariant of $\mathcal{N}_{\mathcal{S}}$.

³The intersection of two or more invariants is again an invariant.

PROOF. Let $\mathcal{N}_S = (N, m_0)$, where $N = (S, T, E)$. First, we prove that $m_0 \models \text{Trap}(\mathcal{N}_S)$. Let $S = \{s_1, \dots, s_k\}$ be a marked trap of \mathcal{N}_S . Since S is marked, $m_0(s_i) = 1$ for some $i \in [1, k]$, thus $m_0 \models \bigvee_{i=1}^k s_i$. Because the choice of S is arbitrary, we have $m_0 \models \text{Trap}(\mathcal{N}_S)$. Second, let $m \models \text{Trap}(\mathcal{N}_S)$ and $t \in T$ such that $m \xrightarrow{t} m'$. We prove that $m' \models \text{Trap}(\mathcal{N}_S)$. Let $S = \{s_1, \dots, s_k\}$ be a marked trap of \mathcal{N}_S . Then $m \models s_i$ for some $i \in [1, k]$ and, because S is a trap, $m' \models s_j$ for some $j \in [1, k]$. Since the choice of S was arbitrary, we obtain $m' \models \text{Trap}(\mathcal{N}_S)$. \square

Next, we describe a method for computing trap invariants that does not explicitly enumerate all the marked traps of a marked PN. First, we consider a *trap constraint* $\Theta(\Gamma)$, derived from the interaction formula Γ , in linear time. By slight abuse of notation, we define, for a given port $p \in P^i$ of the component \mathcal{C}^i , for some $i \in [1, n]$, the pre- and post-state of p in \mathcal{C}^i as $\bullet p \stackrel{\text{def}}{=} s$ and $p \bullet \stackrel{\text{def}}{=} s'$, where $s \xrightarrow{p} s'$ is the unique rule⁴ involving p in Δ^i , and $\bullet p = p \bullet \stackrel{\text{def}}{=} \perp$ if there is no such rule. Assuming that the interaction formula is written in DNF as $\Gamma = \bigvee_{k=1}^N \bigwedge_{\ell=1}^{M_k} p_{k\ell}$, we define the trap constraint:

$$\Theta(\Gamma) \stackrel{\text{def}}{=} \bigwedge_{k=1}^N \left(\bigvee_{\ell=1}^{M_k} \bullet p_{k\ell} \right) \rightarrow \left(\bigvee_{\ell=1}^{M_k} p_{k\ell} \bullet \right)$$

Here N is the number of clauses of the interaction formula, or equivalently, the number of interactions in the bounded component-based system, and M_k is the number of ports in the k -th interaction. We also consider the formula $\text{Init}(\mathcal{S}) \stackrel{\text{def}}{=} \bigvee_{k=1}^n s_0^k$ defining a non-empty intersection with initial marking of the system, and prove the following:

Lemma 2. *Let \mathcal{S} be a bounded system with interaction formula Γ and β be a boolean valuation. Then $\beta \in \llbracket \Theta(\Gamma) \wedge \text{Init}(\mathcal{S}) \rrbracket$ iff $\{s \mid \beta(s) = \top\}$ is a marked trap of \mathcal{N}_S . Moreover, $\beta \in \llbracket \Theta(\Gamma) \wedge \text{Init}(\mathcal{S}) \rrbracket^\mu$ iff $\{s \mid \beta(s) = \top\}$ is a minimal marked trap of \mathcal{N}_S .*

PROOF. Let $\mathcal{C}^i = \langle P^i, S^i, s_0^i, \Delta^i \rangle$, for all $i \in [1, n]$, $\mathcal{N}_S = (N, m_0)$ and $N = \langle Q, T, E \rangle$, where $Q = \bigcup_{i=1}^n S^i$ and $T = \{t_\beta \mid \beta \in \llbracket \Gamma \rrbracket^\mu\}$. Given a trap $S \subseteq Q$ of N , we have the following equivalences:

$$\begin{aligned} S^\bullet &\subseteq \bullet S && \iff \\ \bigwedge_{s \in S} [s \in S] &\rightarrow \{t \in T \mid (s, t) \in E\} \subseteq \{t \in T \mid (t, s) \in E\} && \iff \\ \bigwedge_{s \in S} [s \in S] &\rightarrow (\bigwedge_{t \in T} s \in \bullet t \rightarrow \bigvee_{s' \in S} s' \in S \wedge s' \in t \bullet) && \iff \\ \bigwedge_{s \in S} \bigwedge_{t \in T} (s \in S \wedge s \in \bullet t &\rightarrow \bigvee_{s' \in S} s' \in S \wedge s' \in t \bullet) && \iff \\ \bigwedge_{t \in T} (\bigvee_{s \in \bullet t} s \in S &\rightarrow \bigvee_{s' \in t \bullet} s' \in S) \end{aligned}$$

Assume that $[s \in S]$ is a propositional variable. Then for each transition $t_\beta \in T$, we have:

$$\begin{aligned} \bigvee_{s \in \bullet t_\beta} [s \in S] &\iff \bigvee_{\beta(p)=\top} \bullet p \\ \bigvee_{s \in t_\beta \bullet} [s \in S] &\iff \bigvee_{\beta(p)=\top} p \bullet \end{aligned}$$

Clearly, for any valuation $\beta \in \llbracket \Theta(\Gamma) \rrbracket$ of the propositional variables corresponding to the places in Q that satisfies $\Theta(\Gamma)$, the set $S_\beta = \{s \mid \beta(s) = \top\}$ is a trap of N . If, moreover, $\beta \models \bigvee_{i=1}^n s_0^i$ then S_β is a marked trap of \mathcal{N}_S . Furthermore, β is a minimal model of $\Theta(\Gamma) \wedge \bigwedge_{i=1}^n s_0^i$ iff for each valuation $\beta' \subseteq \beta$, such that $\beta' \neq \beta$, we have $\beta' \not\models \Theta(\Gamma) \wedge \bigvee_{i=1}^n s_0^i$. But then, no strict subset of $\{s \mid \beta(s) = \top\}$ is a marked trap of \mathcal{N}_S , thus $\{s \mid \beta(s) = \top\}$ is a minimal marked trap of \mathcal{N}_S . \square

⁴We have assumed that each port is associated a unique transition rule.

Because $\Theta(\Gamma)$ and $Init(\mathcal{S})$ are boolean formulae, it is, in principle, possible to compute the trap invariant $Trap(\mathcal{N}_S)$ by enumerating the (minimal) models of $\Theta(\Gamma) \wedge Init(\mathcal{S})$ and applying the definition from Lemma 1. However, model enumeration is inefficient and, moreover, does not admit generalization for the parametric case, in which the size of the system is unknown. For these reasons, we prefer a computation of the trap invariant, based on two symbolic transformations of boolean formulae, described next.

For a formula f we denote by f^+ the positive formula obtained by deleting all negative literals from the DNF of f . In lack of a better term, we shall call this operation *positivation*. Second, for a positive boolean formula f , we define the formula $(f)^\sim$ recursively on the structure of f , as follows: $(f_1 \wedge f_2)^\sim \stackrel{\text{def}}{=} f_1^\sim \vee f_2^\sim$, $(f_1 \vee f_2)^\sim \stackrel{\text{def}}{=} f_1^\sim \wedge f_2^\sim$ and $a^\sim \stackrel{\text{def}}{=} \neg a$, for any $a \in \text{BVar}$. Note that f^\sim is equivalent to the negation of the formula obtained from f by substituting each variable a with $\neg a$ in f . This operation, called *dualization*, applies only to positive formulae and is undefined elsewhere.

Lemma 3. *Given boolean formulae f and g , we have $f \equiv g$ only if $(f^+)^\sim \equiv (g^+)^\sim$.*

PROOF. If $f \equiv g$, the set of minterms in the DNF of f is identical to the one of g , modulo commutativity of conjunctions. Then the set of minterms in the DNF of f^+ equals the one of g^+ , thus $f^+ \equiv g^+$. Second, the CNF of $(f^+)^\sim$ is the same of the CNF of $(g^+)^\sim$, as both are obtained directly from the DNF of f^+ and g^+ , respectively, by interchanging disjunctions with conjunctions. \square

The following theorem gives the main result of this section, the symbolic computation of the trap invariant of a bounded system, directly from its interaction formula.

Theorem 1. *For any bounded system \mathcal{S} , with interaction formula Γ , we have:*

$$Trap(\mathcal{N}_S) \equiv \left([\Theta(\Gamma) \wedge Init(\mathcal{S})]^+ \right)^\sim$$

PROOF. For a boolean valuation β , we denote by μ_β the complete minterm $\bigwedge_{\beta(a)=\top} a \wedge \bigwedge_{\beta(a)=\perp} \neg a$. By Lemma 2 we obtain the equivalence:

$$\Theta(\Gamma) \wedge Init(\mathcal{N}_S) \equiv \bigvee \left\{ \mu_\gamma \mid \{s \mid \gamma(s) = \top\} \text{ is a marked trap of } \mathcal{N} \right\}$$

and thus:

$$\begin{aligned} [\Theta(\Gamma) \wedge Init(\mathcal{N}_S)]^+ &\equiv \bigvee \left\{ \mu_\gamma^+ \mid \{s \mid \gamma(s) = \top\} \text{ is a marked trap of } \mathcal{N}_S \right\} \\ &\equiv \bigvee \left\{ \bigvee_{i=1}^k s_i \mid \{s_1, \dots, s_k\} \text{ is a marked trap of } \mathcal{N}_S \right\} \\ &\equiv (Trap(\mathcal{N}_S))^\sim \end{aligned}$$

The equivalence of the statement is obtained by applying Lemma 3. \square

Just as any invariants, trap invariants can be used to prove absence of deadlocks in a bounded system. Assuming, as before, that the interaction formula is given in DNF as $\Gamma = \bigvee_{k=1}^N \bigwedge_{\ell=1}^{M_k} p_{k\ell}$, we define the set of deadlock markings of \mathcal{N}_S by the formula $\Delta(\Gamma) \stackrel{\text{def}}{=} \bigwedge_{k=1}^N \bigvee_{\ell=1}^{M_k} \neg(\bullet p_{k\ell})$. This is the set of configurations in which all interactions are disabled. With this definition, proving deadlock freedom amounts to proving unsatisfiability of a boolean formula.

Corollary 1. *A bounded system \mathcal{S} with interaction formula Γ is deadlock-free if the boolean formula $\left([\Theta(\Gamma) \wedge \text{Init}(\mathcal{S})]^+\right)^\sim \wedge \Delta(\Gamma)$ is unsatisfiable.*

PROOF. Let $\mathcal{N}_\mathcal{S} = (N, m_0)$, where $N = (S, T, E)$ and define the set of deadlock markings:

$$\text{Dead}(\mathcal{N}_\mathcal{S}) \stackrel{\text{def}}{=} \bigwedge_{t \in T} \bigvee_{s \in \bullet_t} \neg s$$

Suppose, by contradiction, that \mathcal{S} is not deadlock-free, thus $\mathcal{R}(\mathcal{N}_\mathcal{S}) \wedge \text{Dead}(\mathcal{N}_\mathcal{S})$ has a satisfying valuation β . Because $\text{Trap}(\mathcal{N}_\mathcal{S})$ defines an invariant of $\mathcal{N}_\mathcal{S}$ and $\mathcal{R}(\mathcal{N}_\mathcal{S})$ defines its least invariant, we have $\mathcal{R}(\mathcal{N}_\mathcal{S}) \rightarrow \text{Trap}(\mathcal{N}_\mathcal{S})$ thus $\beta \models \text{Trap}(\mathcal{N}_\mathcal{S}) \wedge \text{Dead}(\mathcal{N}_\mathcal{S})$. By Theorem 1, we have $\text{Trap}(\mathcal{N}_\mathcal{S}) \rightarrow \left([\Theta(\Gamma) \wedge \text{Init}(\mathcal{N}_\mathcal{S})]^+\right)^\sim$ and, from the definition of $\mathcal{N}_\mathcal{S}$, one also obtains that $\text{Dead}(\mathcal{N}_\mathcal{S}) \rightarrow \Delta(\Gamma)$ leading to $\beta \models \left([\Theta(\Gamma) \wedge \text{Init}(\mathcal{N}_\mathcal{S})]^+\right)^\sim \wedge \Delta(\Gamma)$, which contradicts $\left([\Theta(\Gamma) \wedge \text{Init}(\mathcal{N}_\mathcal{S})]^+\right)^\sim \wedge \Delta(\Gamma) \rightarrow \perp$. \square

3. Parametric Component-based Systems

From now on we shall focus on parametric systems, consisting of a fixed set of component types $\mathcal{C}^1, \dots, \mathcal{C}^n$, such that the number of instances of each type is not known in advance. These numbers are given by a function $M : [1, n] \rightarrow \mathbb{N}$, where $M(k)$ denotes the number of components of type \mathcal{C}^k that are active in the system. To simplify the technical presentation of the results, we assume that all instances of a component type are created at once, before the system is started⁵. For the rest of this section, we fix a parametric system $\mathcal{S} = \langle \mathcal{C}^1, \dots, \mathcal{C}^n, M, \Gamma \rangle$, where each component type $\mathcal{C}^k = \langle P^k, S^k, s_0^k, \Delta^k \rangle$ has the same definition as a component in a bounded system and Γ is an interaction formula, written in the fragment of first order logic, defined next.

3.1. Monadic Interaction Logic

For each component type \mathcal{C}^k , where $k \in [1, n]$, we assume a set of index variables Var^k and a set of predicate symbols $\text{Pred}^k \stackrel{\text{def}}{=} P^k \cup S^k$. We also define the sets $\text{Var} \stackrel{\text{def}}{=} \bigcup_{k=1}^n \text{Var}^k$ and $\text{Pred} \stackrel{\text{def}}{=} \bigcup_{k=1}^n \text{Pred}^k$. Moreover, we consider that $\text{Var}^k \cap \text{Var}^\ell = \emptyset$ and $\text{Pred}^k \cap \text{Pred}^\ell = \emptyset$, for all $1 \leq k < \ell \leq n$. For simplicity's sake, we assume that all predicate symbols in Pred are of arity one. For component types \mathcal{C}^k , such that $M(k) = 1$ and predicate symbols $\text{pred} \in \text{Pred}^k$, we shall write pred instead of $\text{pred}(1)$, as in the interaction formula of the system from Figure 1b. The syntax of the *monadic interaction logic* (MIL) is given below:

$$\begin{aligned} i, j &\in \text{Var} && \text{index variables} \\ \phi &:= i = j \mid \text{pred}(i) \mid \phi_1 \wedge \phi_2 \mid \neg \phi_1 \mid \exists i . \phi_1 \end{aligned}$$

⁵This is not a limitation, since dynamic creation of a finite number of instances can be simulated by considering that all instances are initially in a waiting state, which is left as result of an interaction involving a designated “spawn” port. This is w.l.o.g., since we consider safety properties, that require reasoning about finite executions.

where, for each predicate atom $\text{pred}(i)$, if $\text{pred} \in \text{Pred}^k$ and $i \in \text{Var}^\ell$ then $k = \ell$. We use the shorthands $\forall i . \phi_1 \stackrel{\text{def}}{=} \neg(\exists i . \neg\phi_1)$ and $\text{distinct}(i_1, \dots, i_m) \stackrel{\text{def}}{=} \bigwedge_{1 \leq j < \ell \leq m} \neg i_j = i_\ell$ ⁶. A *sentence* is a formula in which all variables are in the scope of a quantifier. A formula is *positive* if each predicate symbol occurs under an even number of negations. The semantics of MIL is given in terms of structures $\mathcal{I} = (\mathfrak{U}, \nu, \iota)$, where:

- $\mathfrak{U} \stackrel{\text{def}}{=} [1, \max_{k=1}^n \mathbf{M}(k)]$ is the *universe* of instances, over which variables range,
- $\nu : \text{Var} \rightarrow \mathfrak{U}$ is a *valuation* mapping variables to elements of the universe,
- $\iota : \text{Pred} \rightarrow 2^\mathfrak{U}$ is an *interpretation* of predicates as subsets of the universe.

For a structure $\mathcal{I} = (\mathfrak{U}, \nu, \iota)$ and a formula ϕ , the satisfaction relation $\mathcal{I} \models \phi$ is defined as:

$$\begin{array}{llll} \mathcal{I} \models \perp & \Leftrightarrow & \text{never} & \mathcal{I} \models i = j & \Leftrightarrow & \nu(i) = \nu(j) \\ \mathcal{I} \models p(i) & \Leftrightarrow & \nu(i) \in \iota(p) & \mathcal{I} \models \phi_1 \wedge \phi_2 & \Leftrightarrow & \mathcal{I} \models \phi_1 \text{ and } \mathcal{I} \models \phi_2 \\ \mathcal{I} \models \neg\phi_1 & \Leftrightarrow & \mathcal{I} \not\models \phi_1 & \mathcal{I} \models \exists i . \phi_1 & \Leftrightarrow & (\mathfrak{U}, \nu[i \leftarrow m], \iota) \models \phi_1 \\ & & & & & \text{for some } m \in [1, \mathbf{M}(k)] \\ & & & & & \text{provided that } i \in \text{Var}^k \end{array}$$

where $\nu[i \leftarrow m]$ is the valuation that acts as ν , except for i , which is assigned to m . Whenever $\mathcal{I} \models \phi$, we say that \mathcal{I} is a *model* of ϕ . It is known that, if a MIL formula has a model, then it has a model with universe of cardinality at most exponential in the size (number of symbols) of the formula [27]. This result, due to Löwenheim, is among the first decidability results for a fragment of first order logic.

Given structures $\mathcal{I}_i = (\mathfrak{U}, \nu_i, \iota_i)$, for $i = 1, 2$, we write $\mathcal{I}_1 \subseteq \mathcal{I}_2$ iff $\iota_1(p) \subseteq \iota_2(p)$, for all $p \in \text{Pred}$ and $\mathcal{I}_1 \subset \mathcal{I}_2$ iff $\mathcal{I}_1 \subseteq \mathcal{I}_2$ and $\mathcal{I}_1 \neq \mathcal{I}_2$. As before, we define the sets $\llbracket \phi \rrbracket = \{\mathcal{I} \mid \mathcal{I} \models \phi\}$ and $\llbracket \phi \rrbracket^\mu = \{\mathcal{I} \in \llbracket \phi \rrbracket \mid \forall \mathcal{I}' . \mathcal{I}' \subset \mathcal{I} \rightarrow \mathcal{I}' \notin \llbracket \phi \rrbracket\}$ of models and minimal models of a MIL formula, respectively. Given formulae ϕ_1 and ϕ_2 , we write $\phi_1 \equiv \phi_2$ for $\llbracket \phi_1 \rrbracket = \llbracket \phi_2 \rrbracket$ and $\phi_1 \equiv^\mu \phi_2$ for $\llbracket \phi_1 \rrbracket^\mu = \llbracket \phi_2 \rrbracket^\mu$.

3.2. Execution Semantics of Parametric Systems

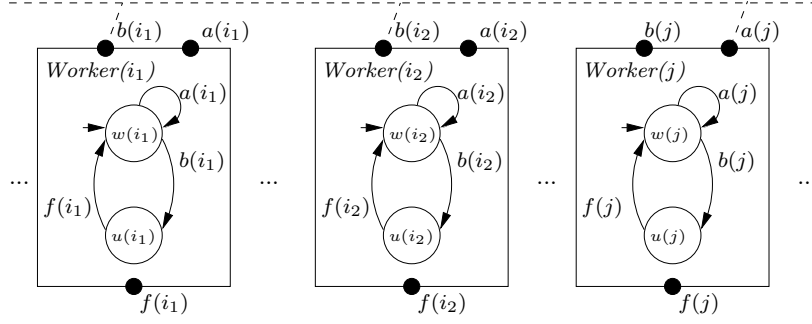
We consider the interaction formulae Γ of parametric systems to be finite disjunctions of interaction formulae clauses \mathcal{C} of the form below:

$$\exists i_1 \dots \exists i_\ell . \varphi \wedge \bigwedge_{j=1}^{\ell} p_j(i_j) \wedge \bigwedge_{j=\ell+1}^{\ell+m} \forall i_j . \psi_j \rightarrow p_j(i_j) \quad (1)$$

where $\varphi, \psi_{\ell+1}, \dots, \psi_{\ell+m}$ are conjunctions of equalities and disequalities involving index variables such that moreover for any distinct j_1, j_2 such that $j_1 < j_2$ and $p_{j_1}, p_{j_2} \in \text{Pred}^k$ it holds either (a) $j_1 \leq \ell, j_2 \leq \ell$ and φ implies $\neg i_{j_1} = i_{j_2}$ or (b) $j_1 \leq \ell, \ell < j_2$ and ψ_{j_2} implies $\neg i_{j_1} = i_{j_2}$. Intuitively, the formulae (1) state that there are at most ℓ component instances that engage in a valid multiparty rendez-vous interaction on ports $p_1(i_1), \dots, p_\ell(i_\ell)$, together with a broadcast to the ports $p_{\ell+1}(i_{\ell+1}), \dots, p_{\ell+m}(i_{\ell+m})$ of the instances that fulfill the constraints $\psi_{\ell+1}, \dots, \psi_{\ell+m}$. Due to the enforced constraints, all these ports belong to distinct components. Observe that, if $m = 0$, the above formula corresponds to a multiparty (generalized) rendez-vous interaction $\exists i_1 \dots \exists i_\ell . \varphi \wedge \bigwedge_{j=1}^{\ell} p_j(i_j)$. An example of peer-to-peer rendez-vous is the parametric system from Figure 1. Another example of broadcast is given below.

⁶Throughout this paper, we consider that $\bigwedge_{i \in I} \phi_i = \top$ if $I = \emptyset$.

Example 1. Consider the parametric system obtained from an arbitrary number of *Worker* components (Figure 3), where $\mathcal{C}^1 = \text{Worker}$, $\text{Var}^1 = \{i, i_1, i_2, j\}$ and $\text{Pred}^1 = \{a, b, f, u, w\}$. Any pair of instances can jointly execute the *b* (*begin*) action provided *all* others are taking the *a* (*await*) action. Any instance can also execute alone the *f* (*finish*) action. \square



$$\Gamma = [\exists i_1 \exists i_2 . i_1 \neq i_2 \wedge b(i_1) \wedge b(i_2) \wedge \forall j . j \neq i_1 \wedge j \neq i_2 \rightarrow a(j)] \vee \exists i . f(i)$$

Figure 3: Parametric System with Broadcast

The execution semantics of a parametric system \mathcal{S} is the marked PN $\mathcal{N}_{\mathcal{S}} = (N, m_0)$, where $N = (\bigcup_{k=1}^n S^k \times [1, M(k)], T, E)$, $m_0((s_0^k, i)) = 1$, for all $k \in [1, n]$ and $i \in [1, M(k)]$, and the sets of transitions T and edges E are defined next. For each interaction formulae clause \mathcal{C} belonging to Γ , for each minimal model $\mathcal{I} = (\mathfrak{U}, \nu, \iota) \in \llbracket \mathcal{C} \rrbracket^\mu$, we have a transition $t_{\mathcal{I}} \in T$ and the edges $((s_i, k), t_{\mathcal{I}}), (t_{\mathcal{I}}, (s'_i, k)) \in E$ for all $i \in [1, n]$ such that $s_i \xrightarrow{p_i} s'_i \in \Delta^i$ and $k \in \iota(p_i)$. Moreover, nothing else is in T or E .

As a remark, unlike in the case of bounded systems, the size of the marked PN $\mathcal{N}_{\mathcal{S}}$, that describes the execution semantics of a parametric system \mathcal{S} , depends on the maximum number of instances of each component type. The definition of the trap invariant $\text{Trap}(\mathcal{N}_{\mathcal{S}})$ is the same as in the bounded case, except that, in this case, the size of the boolean formula depends on the (unbounded) number of instances in the system. The challenge, addressed in the following, is to define trap invariants using MIL formulae of a fixed size.

3.3. Computing Parametric Trap Invariants

To start with, we define the trap constraint $\Theta(\Gamma)$ of an interaction formula Γ consisting of a finite disjunction of (1) clauses, as the finite conjunction of formulae of the form:

$$\forall i_1 \dots \forall i_\ell \quad \left[\varphi \wedge \left(\bigvee_{j=1}^\ell \bullet p_j(i_j) \vee \bigvee_{j=\ell+1}^{\ell+m} \exists i_j . \psi_j \wedge \bullet p_j(i_j) \right) \right] \rightarrow \left[\left(\bigvee_{j=1}^\ell p_j \bullet(i_j) \vee \bigvee_{j=\ell+1}^{\ell+m} \exists i_j . \psi_j \wedge p_j \bullet(i_j) \right) \right]$$

where, for a port $p \in \mathbf{P}^k$ of some component type \mathcal{C}^k , $\bullet p(i)$ and $p(i) \bullet$ denote the unique predicate atoms $s(i)$ and $s'(i)$, such that $s \xrightarrow{p} s' \in \Delta^k$ is the (unique) transition involving p in T^k , or \perp if there is no such rule.

Example 2. The interaction formula of the parametric (rendez-vous) system in Figure 1b is

$$\Gamma = \exists i. (a \wedge b(i)) \vee \exists i. (e \wedge f(i))$$

The corresponding trap constraint is obtained by applying the syntactic transformation, separately on the two clauses, and taking their conjunction, that is:

$$\begin{aligned} \Theta(\Gamma) &= \forall i. [(\bullet a \vee \bullet b(i)) \rightarrow [(a^\bullet \vee b^\bullet(i))]] \quad \wedge \quad \forall i. [(\bullet e \vee \bullet f(i)) \rightarrow [(e^\bullet \vee f^\bullet(i))]] \\ &= \forall i. [r \vee w(i)] \rightarrow [s \vee u(i)] \quad \wedge \quad \forall i. [s \vee u(i)] \rightarrow [r \vee w(i)] \end{aligned}$$

Analogously, the interaction formula for the parametric (broadcast) system in Figure 3 is:

$$\Gamma = [\exists i_1 \exists i_2. i_1 \neq i_2 \wedge b(i_1) \wedge b(i_2) \wedge \forall j. j \neq i_1 \wedge j \neq i_2 \rightarrow a(j)] \vee \exists i. f(i)$$

Notice that the first clause defines a broadcast interaction, and the second a unary (single port) interaction. The corresponding trap constraint is again obtained by applying the syntactic transformation separately on the two clauses, and taking their conjunction, that is:

$$\begin{aligned} \Theta(\Gamma) &= \forall i_1. \forall i_2. [i_1 \neq i_2 \wedge (\bullet b(i_1) \vee \bullet b(i_2) \vee \exists j. (j \neq i_1 \wedge j \neq i_2 \wedge \bullet a(j)))] \rightarrow \\ &\quad [(b^\bullet(i_1) \vee b^\bullet(i_2) \vee \exists j. (j \neq i_1 \wedge j \neq i_2 \wedge a^\bullet(j)))] \quad \wedge \\ &\quad \forall i. [\bullet f(i) \rightarrow f^\bullet(i)] \\ &= \forall i_1. \forall i_2. [i_1 \neq i_2 \wedge (w(i_1) \vee w(i_2) \vee \exists j. (j \neq i_1 \wedge j \neq i_2 \wedge w(j)))] \rightarrow \\ &\quad [(u(i_1) \vee u(i_2) \vee \exists j. (j \neq i_1 \wedge j \neq i_2 \wedge w(j)))] \quad \wedge \\ &\quad \forall i. [u(i) \rightarrow w(i)] \end{aligned}$$

□

To prove the correctness of the above parametric trap constraint definition, we define a translation of MIL formulae into boolean formulae of unbounded size. Given a function $M : [1, n] \rightarrow \mathbb{N}$, the *unfolding* of a MIL sentence ϕ is the boolean formula $B_M(\phi)$ obtained by replacing:

- each existential quantifier $\exists i. \psi(i)$, for $i \in \text{Var}^k$, by a finite disjunction $\bigvee_{\ell=1}^{M(k)} \psi[\ell/i]$,
 - each universal quantifier $\forall i. \psi(i)$, for $i \in \text{Var}^k$, by a finite conjunction $\bigwedge_{\ell=1}^{M(k)} \psi[\ell/i]$,
- where the substitution of the constant $\ell \in M(k)$ for the variable i is defined recursively as usual, except for $\text{pred}(i)[\ell/i] \stackrel{\text{def}}{=} (\text{pred}, \ell)$, which is a propositional variable. Further, we relate structures to boolean valuations of unbounded sizes as follows. For a structure $\mathcal{I} = (\mathcal{U}, \nu, \iota)$ we define the boolean valuation $\beta_{\mathcal{I}}((\text{pred}, \ell)) = \top$ if and only if $\ell \in \iota(\text{pred})$, for each predicate symbol pred and each integer constant ℓ . Conversely, for each valuation β of the propositional variables (pred, ℓ) , there exists a structure $\mathcal{I}_\beta = (\mathcal{U}, \nu, \iota)$ such that $\iota(\text{pred}) \stackrel{\text{def}}{=} \{\ell \mid \beta((\text{pred}, \ell)) = \top\}$, for each $\text{pred} \in \text{Pred}$. The following lemma relates the semantics of MIL formulae with that of their boolean unfoldings:

Lemma 4. *Given a MIL sentence ϕ and a function $M : [1, n] \rightarrow \mathbb{N}$, the following hold:*

1. *for each structure $\mathcal{I} \in \llbracket \phi \rrbracket$, we have $\beta_{\mathcal{I}} \in \llbracket B_M(\phi) \rrbracket$ and conversely, for each valuation $\beta \in \llbracket B_M(\phi) \rrbracket$, we have $\mathcal{I}_\beta \in \llbracket \phi \rrbracket$.*
2. *for each structure $\mathcal{I} \in \llbracket \phi \rrbracket^\mu$, we have $\beta_{\mathcal{I}} \in \llbracket B_M(\phi) \rrbracket^\mu$ and conversely, for each valuation $\beta \in \llbracket B_M(\phi) \rrbracket^\mu$, we have $\mathcal{I}_\beta \in \llbracket \phi \rrbracket^\mu$.*

PROOF. (1) By induction on the structure of ϕ . (2) First, it is routine to prove that, for any two structures $\mathcal{I}_1 \subseteq \mathcal{I}_2$, we have $\beta_{\mathcal{I}_1} \subseteq \beta_{\mathcal{I}_2}$ and, conversely, for any two valuations $\beta_1 \subseteq \beta_2$, we have $\mathcal{I}_{\beta_1} \subseteq \mathcal{I}_{\beta_2}$. Next, let $\mathcal{I} \in \llbracket \phi \rrbracket^\mu$. By the point (1), we have $\beta_{\mathcal{I}} \in \llbracket \mathbf{B}_M(\phi) \rrbracket$. Suppose $\beta_{\mathcal{I}} \notin \llbracket \mathbf{B}_M(\phi) \rrbracket^\mu$, which means that there exists $\beta' \subsetneq \beta_{\mathcal{I}}$ such that $\beta' \in \llbracket \mathbf{B}_M(\phi) \rrbracket$. By the point (1), $\mathcal{I}_{\beta'} \in \llbracket \phi \rrbracket$ and, moreover, $\mathcal{I}_{\beta'} \subsetneq \mathcal{I}$, which contradicts the minimality of \mathcal{I} . Thus $\beta_{\mathcal{I}} \in \llbracket \mathbf{B}_M(\phi) \rrbracket^\mu$. The other direction is symmetric. \square

Considering the MIL formula $\text{Init}(\mathcal{S}) \stackrel{\text{def}}{=} \bigvee_{k=1}^n \exists i_k . s_0^k(i_k)$, that defines a non-empty intersection with the set of initial configurations of a parametric system \mathcal{S} , the following lemma proves the correctness of the above parametric trap constraint definition:

Lemma 5. *Let \mathcal{S} be a parametric system with interaction formula Γ and \mathcal{I} be a structure. Then $\mathcal{I} \models \Theta(\Gamma) \wedge \text{Init}(\mathcal{S})$ iff $\{(s, k) \mid k \in \iota(s)\}$ is a marked trap of $\mathcal{N}_{\mathcal{S}}$. Moreover, $\mathcal{I} \in \llbracket \Theta(\Gamma) \wedge \text{Init}(\mathcal{S}) \rrbracket^\mu$ iff $\{(s, k) \mid k \in \iota(s)\}$ is a minimal marked trap of $\mathcal{N}_{\mathcal{S}}$.*

PROOF. Let $\mathcal{C}^k = \langle \mathbf{P}^k, \mathbf{S}^k, s_0^k, \Delta^k \rangle$ and define the bounded system:

$$\begin{aligned} \mathcal{U}(\mathcal{S}) &\stackrel{\text{def}}{=} \langle \{\mathcal{C}^{k,i} \mid k \in [1, n], i \in [1, M(k)]\}, \mathbf{B}_M(\Gamma) \rangle \\ \mathcal{C}^{k,i} &\stackrel{\text{def}}{=} \langle \mathbf{P}^k \times \{i\}, \mathbf{S}^k \times \{i\}, s_0^k \times \{i\}, \{(s, i) \xrightarrow{(p,i)} (s', i) \mid s \xrightarrow{p} s' \in \Delta^k\} \rangle \end{aligned}$$

It is not hard to prove that $\mathcal{N}_{\mathcal{S}}$ is the same as $\mathcal{N}_{\mathcal{U}(\mathcal{S})}$, thus their marked traps coincide. The following equivalences follow from Lemma 4:

$$\begin{aligned} \mathcal{I} \models \Theta(\Gamma) &\iff \beta_{\mathcal{I}} \models \Theta(\mathbf{B}_M(\Gamma)) \\ \mathcal{I} \models \text{Init}(\mathcal{S}) &\iff \beta_{\mathcal{I}} \models \text{Init}(\mathcal{U}(\mathcal{S})) \end{aligned}$$

Moreover, $\{(s, k) \mid k \in \iota(s)\} = \{(s, k) \mid \beta_{\mathcal{I}}((s, k)) = \top\}$ and we apply Lemma 2. \square

We are currently left with the task of computing a MIL formula which defines the trap invariant $\text{Trap}(\mathcal{N}_{\mathcal{S}})$ of a parametric component-based system $\mathcal{S} = \langle \mathcal{C}^1, \dots, \mathcal{C}^n, \mathbf{M}, \Gamma \rangle$. The difficulty lies in the fact that the size of $\mathcal{N}_{\mathcal{S}}$ and thus, that of the boolean formula $\text{Trap}(\mathcal{N}_{\mathcal{S}})$ depends on the number $M(k)$ of instances of each component type $k \in [1, n]$. As we aim at computing an invariant able to prove safety properties, such as deadlock freedom, independently of how many components are present in the system, we must define the trap invariant using a formula depending exclusively on Γ , i.e. not on \mathbf{M} .

Observe first that $\text{Trap}(\mathcal{N}_{\mathcal{S}})$ can be equivalently defined using only the minimal marked traps of $\mathcal{N}_{\mathcal{S}}$, which, by Lemma 5, are exactly the sets $\{(s, k) \mid k \in \iota(s)\}$, defined by some structure $(\mathcal{U}, \nu, \iota) \in \llbracket \Theta(\Gamma) \wedge \text{Init}(\mathcal{S}) \rrbracket^\mu$. Assuming that the set of structures $\llbracket \Theta(\Gamma) \wedge \text{Init}(\mathcal{S}) \rrbracket^\mu$, or an over-approximation of it, can be defined by a positive MIL formula, the trap invariant is defined using a generalization of boolean dualisation to predicate logic, defined recursively, as follows:

$$\begin{aligned} (i = j)^\sim &\stackrel{\text{def}}{=} \neg i = j & (\phi_1 \vee \phi_2)^\sim &\stackrel{\text{def}}{=} \phi_1^\sim \wedge \phi_2^\sim & (\exists i . \phi_1)^\sim &\stackrel{\text{def}}{=} \forall i . \phi_1^\sim \\ (\neg i = j)^\sim &\stackrel{\text{def}}{=} i = j & (\phi_1 \wedge \phi_2)^\sim &\stackrel{\text{def}}{=} \phi_1^\sim \vee \phi_2^\sim & (\forall i . \phi_1)^\sim &\stackrel{\text{def}}{=} \exists i . \phi_1^\sim \\ p(i)^\sim &\stackrel{\text{def}}{=} p(i) \end{aligned}$$

The crux of the method is the ability of defining, given an arbitrary MIL formula ϕ , a positive MIL formula ϕ^\oplus that preserve its minimal models, formally $\phi \equiv^\mu \phi^\oplus$. Because

of quantification over unbounded domains, a MIL formula ϕ does not have a disjunctive normal form and thus one cannot define ϕ^\oplus by simply deleting the negative literals in DNF, as was done for the definition of the positivation operation $(.)^+$, in the propositional case. For now we assume that the transformation $(.)^\oplus$ of monadic predicate formulae into positive formulae preserving minimal models is defined (a detailed presentation of this step is given next in §4) and close this section with a parametric counterpart of Theorem 1.

Before giving the proof of the main result of this section, we shall be needing a few technical lemmas. For a set \mathcal{S} of boolean valuations, let $\mathcal{S}^\uparrow \stackrel{\text{def}}{=} \{\beta \mid \exists \beta' \in \mathcal{S} . \beta' \subseteq \beta\}$ be its *upward closure*. A set \mathcal{S} of boolean valuations is upward-closed iff $\mathcal{S} = \mathcal{S}^\uparrow$. The following lemma shows that the set of models of a positive boolean formula is upward-closed and thus uniquely determined by its minimal elements.

Lemma 6. *Given a positive boolean formula f , we have $\llbracket f \rrbracket = \llbracket f \rrbracket^\uparrow = (\llbracket f \rrbracket^\mu)^\uparrow$.*

PROOF. The inclusions $\llbracket f \rrbracket \subseteq \llbracket f \rrbracket^\uparrow \subseteq (\llbracket f \rrbracket^\mu)^\uparrow$ are immediate. To show that $(\llbracket f \rrbracket^\mu)^\uparrow \subseteq \llbracket f \rrbracket$, observe that, if f is positive and $\beta \models f$ then any valuation β' such that $\beta \subseteq \beta'$ is also model of f . Let $\beta \in \llbracket f \rrbracket^\mu$ be a valuation. Then there exists $\beta' \in \llbracket f \rrbracket^\mu$ such that $\beta' \subseteq \beta$. Since $\beta' \models f$ and f is positive, we obtain $\beta \in \llbracket f \rrbracket$.

Lemma 7. *Given a MIL sentence ϕ with quantified variables i^1, \dots, i^n and a function $M : [1, n] \rightarrow \mathbb{N}$, we have $B_M(\phi)^+ \equiv B_M(\phi^\oplus)$.*

PROOF. Because $B_M(\phi)^+$ and $B_M(\phi^\oplus)$ are both positive boolean formulae, we have $\llbracket B_M(\phi)^+ \rrbracket = (\llbracket B_M(\phi)^+ \rrbracket^\mu)^\uparrow$ and $\llbracket B_M(\phi^\oplus) \rrbracket = (\llbracket B_M(\phi^\oplus) \rrbracket^\mu)^\uparrow$, by Lemma 6. It is sufficient to show $B_M(\phi)^+ \equiv^\mu B_M(\phi^\oplus)$, i.e. $\llbracket B_M(\phi)^+ \rrbracket^\mu = \llbracket B_M(\phi^\oplus) \rrbracket^\mu$ to obtain the equivalence in general. To prove $\llbracket B_M(\phi)^+ \rrbracket^\mu = \llbracket B_M(\phi^\oplus) \rrbracket^\mu$, we show that $\llbracket B_M(\phi)^+ \rrbracket^\mu \subseteq \llbracket B_M(\phi^\oplus) \rrbracket^\mu$ and $\llbracket B_M(\phi^\oplus) \rrbracket^\mu \subseteq \llbracket B_M(\phi)^+ \rrbracket^\mu$, respectively.

$\llbracket B_M(\phi)^+ \rrbracket^\mu \subseteq \llbracket B_M(\phi^\oplus) \rrbracket^\mu$ Let $\beta \in \llbracket B_M(\phi)^+ \rrbracket^\mu$ be a valuation. Then, we also have $\beta \in \llbracket B_M(\phi) \rrbracket^\mu$, since $\phi^+ \equiv^\mu \phi$, in general for any boolean formula ϕ . Then, by Lemma 4 (2), there exists a structure $\mathcal{I} \in \llbracket \phi \rrbracket^\mu$ such that $\beta = \beta_{\mathcal{I}}$. Hence we obtain $\mathcal{I} \in \llbracket \phi^\oplus \rrbracket^\mu \subseteq \llbracket \phi^\oplus \rrbracket$. But then $\beta \in \llbracket B_M(\phi^\oplus) \rrbracket$, by Lemma 4 (1).

$\llbracket B_M(\phi^\oplus) \rrbracket^\mu \subseteq \llbracket B_M(\phi)^+ \rrbracket^\mu$ Let $\beta \in \llbracket B_M(\phi^\oplus) \rrbracket^\mu$ be a boolean valuation. By Lemma 4 (2), we obtain a structure $\mathcal{I} \in \llbracket \phi^\oplus \rrbracket^\mu$ such that $\beta = \beta_{\mathcal{I}}$. But then $\mathcal{I} \in \llbracket \phi \rrbracket^\mu$ and $\beta \in \llbracket B_M(\phi) \rrbracket^\mu$, by Lemma 4 (2). Hence $\beta \in \llbracket B_M(\phi)^+ \rrbracket^\mu$. \square

Theorem 2. *For any parametric system $\mathcal{S} = \langle \mathcal{C}^1, \dots, \mathcal{C}^n, M, \Gamma \rangle$, we have*

$$Trap(\mathcal{N}_{\mathcal{S}}) \equiv B_M \left(\left((\Theta(\Gamma) \wedge Init(\mathcal{S}))^\oplus \right)^\sim \right)$$

PROOF. By Theorem 1, we have $Trap(\mathcal{N}_{\mathcal{S}}) \equiv \left((\Theta(B_M(\Gamma)) \wedge B_M(Init(\mathcal{S})))^+ \right)^\sim$. We

obtain the following equivalences:

$$\begin{aligned}
& \left((\Theta(B_M(\Gamma)) \wedge B_M(Init(\mathcal{S})))^+ \right)^\sim && \text{since } \Theta(B_M(\Gamma)) \equiv B_M(\Theta(\Gamma)) \\
\equiv & \left((B_M(\Theta(\Gamma)) \wedge B_M(Init(\mathcal{S})))^+ \right)^\sim \\
\equiv & \left((B_M(\Theta(\Gamma) \wedge Init(\mathcal{S})))^+ \right)^\sim && \text{by Lemma 7} \\
\equiv & \left(B_M \left((\Theta(\Gamma) \wedge Init(\mathcal{S}))^\oplus \right) \right)^\sim \\
\equiv & B_M \left(\left((\Theta(\Gamma) \wedge Init(\mathcal{S}))^\oplus \right)^\sim \right)
\end{aligned}$$

□

4. Cardinality Constraints

This section is concerned with the definition of a *positivation* operator $(.)^\oplus$ for MIL sentences, whose only requirements are that ϕ^\oplus is positive and $\phi \equiv^\mu \phi^\oplus$. For this purpose, we use a logic of quantifier-free *boolean cardinality constraints* [24, 28] as an equivalent intermediate language, on which the positive formulae are defined. The translation of MIL into cardinality constraints is done by an equivalence-preserving quantifier elimination procedure, described in §4.1. As a byproduct, since the satisfiability of quantifier-free cardinality constraints is NP-complete [24] and integrated with SMT [28], we obtain a practical decision procedure for MIL that does not use model enumeration, as suggested by the small model property [27]. Finally, the definition of a positive MIL formula from a boolean combination of quantifier-free cardinality constraints is given in §4.2.

We start by giving the definition of cardinality constraints. Given the set of monadic predicate symbols Pred , a *boolean term* is generated by the syntax:

$$t := p \in \text{Pred} \mid \neg t_1 \mid t_1 \wedge t_2 \mid t_1 \vee t_2$$

When there is no risk of confusion, we borrow the terminology of boolean logic and say that a term is in DNF if it is a disjunction of conjunctions (minterms). We also write $t_1 \rightarrow t_2$ if and only if the implication is valid when t_1 and t_2 are interpreted as boolean formulae, with each predicate symbol viewed as a propositional variable. Two boolean terms t_1 and t_2 are said to be *compatible* if and only if $t_1 \wedge t_2$ is satisfiable, when viewed as a boolean formula.

For a boolean term t and a first-order variable $i \in \text{Var}$, we define the shorthand $t(i)$ recursively, as $(\neg t_1)(i) \stackrel{\text{def}}{=} \neg t_1(i)$, $(t_1 \wedge t_2)(i) \stackrel{\text{def}}{=} t_1(i) \wedge t_2(i)$ and $(t_1 \vee t_2)(i) \stackrel{\text{def}}{=} t_1(i) \vee t_2(i)$. Given a positive integer $n \in \mathbb{N}$ and t a boolean term, we define the following *cardinality constraints*, by MIL formulae:

$$\begin{aligned}
|t| \geq n & \stackrel{\text{def}}{=} \exists i_1 \dots \exists i_n . \text{distinct}(i_1, \dots, i_n) \wedge \bigwedge_{j=1}^n t(i_j) \\
|t| \leq n & \stackrel{\text{def}}{=} \neg(|t| \geq n + 1)
\end{aligned}$$

We shall further use cardinality constraints with $n = \infty$, by defining $|t| \geq \infty \stackrel{\text{def}}{=} \perp$ and $|t| \leq \infty \stackrel{\text{def}}{=} \top$. The intuitive semantics of cardinality constraints is formally defined in terms of structures $\mathcal{I} = (\mathcal{U}, \nu, \iota)$ by the semantics of monadic predicate logic, given in the previous. For instance, $|p \wedge q| \geq 1$ means that the intersection of the sets p and q is not empty, whereas $|\neg p| \leq 0$ means p contains all elements from the universe.

4.1. Quantifier Elimination

Given a sentence ϕ , written in MIL, we build an equivalent boolean combination of cardinality constraints $\text{qe}(\phi)$, using quantifier elimination. Actually, this construction is a direct consequence of the quantifier elimination procedure for the first-order theory of Boolean algebra [30, 27]. We describe the elimination of a single existential quantifier and the generalization to several existential or universal quantifiers is immediate. Assume that $\phi = \exists i_1 . \bigvee_{k \in K} \psi_k(i_1, \dots, i_m)$, where K is a finite set of indices and, for each $k \in K$, ψ_k is a quantifier-free conjunction of atomic propositions of the form $i_j = i_\ell$, $P(i_j)$ and their negations, for some $j, \ell \in [1, m]$. We write, equivalently, $\phi \equiv \bigvee_{k \in K} \varphi_k \wedge \exists i_1 . \theta_k(i_1, \dots, i_m)$, where φ_k does not contain occurrences of i_1 and θ_k is a conjunction of literals of the form $P(i_1)$, $\neg P(i_1)$, $i_1 = i_j$ and $\neg i_1 = i_j$, for some $j \in [2, m]$. For each $k \in K$, we distinguish the following cases:

1. if $i_1 = i_j$ is a consequence of θ_k , for some $j > 1$, let $\text{qe}(\exists i_1 . \theta_k) \stackrel{\text{def}}{=} \theta_k[i_j/i_1]$.
2. else, $\theta_k = \bigwedge_{j \in J_k} \neg i_1 = i_j \wedge t_k(i_1)$ for some $J_k \subseteq [2, m]$ and boolean term t_k , and let:

$$\begin{aligned} \text{qe}(\exists i_1 . \theta_k) &\stackrel{\text{def}}{=} \bigwedge_{J \subseteq J_k} \left[\text{distinct}(\{i_j\}_{j \in J}) \wedge \bigwedge_{j \in J} t_k(i_j) \right] \rightarrow |t_k| \geq ||J|| + 1 \\ \text{qe}(\phi) &\stackrel{\text{def}}{=} \bigvee_{k \in K} \varphi_k \wedge \text{qe}(\exists i_1 . \theta_k) \end{aligned}$$

Example 3. The following examples illustrate quantifier elimination as defined above:

$$\begin{aligned} \exists i. P(i) &\equiv |P| \geq 1 \\ \exists i. \neg P(i) &\equiv |\neg P| \geq 1 \\ \exists i. P(i) \wedge Q(i) &\equiv |P \wedge Q| \geq 1 \\ \exists i. P(i) \wedge \neg Q(i) &\equiv |P \wedge \neg Q| \geq 1 \\ \exists i. \neg i = j \wedge P(i) &\equiv (P(j) \rightarrow |P| \geq 2) \wedge (|P| \geq 1) \\ \exists i. \neg i = j \wedge P(i) \wedge \neg Q(i) &\equiv ((P(j) \wedge \neg Q(j)) \rightarrow |P \wedge \neg Q| \geq 2) \wedge (|P \wedge \neg Q| \geq 1) \\ \exists i. \neg i = j \wedge \neg i = k \wedge P(i) &\equiv ((\neg j = k \wedge P(j) \wedge P(k)) \rightarrow |P| \geq 3) \wedge \\ &\quad (P(j) \rightarrow |P| \geq 2) \wedge (P(k) \rightarrow |P| \geq 2) \wedge |P| \geq 1 \end{aligned}$$

□

Universal quantification is dealt with using the duality $\text{qe}(\forall i_1 . \psi) \stackrel{\text{def}}{=} \neg \text{qe}(\exists i_1 . \neg \psi)$. For a prenex formula $\phi = Q_n i_n \dots Q_1 i_1 . \psi$, where $Q_1, \dots, Q_n \in \{\exists, \forall\}$ and ψ is quantifier-free, we define, recursively $\text{qe}(\phi) \stackrel{\text{def}}{=} \text{qe}(Q_n i_n . \text{qe}(Q_{n-1} i_{n-1} \dots Q_1 i_1 . \psi))$. It is easy to see that, if ϕ is a sentence, $\text{qe}(\phi)$ is a boolean combination of cardinality constraints. The correctness of the construction is a consequence of the following lemma:

Lemma 8. *Given a MIL formula $\phi = Q_n i_n \dots Q_1 i_1 . \psi$, where $Q_1, \dots, Q_n \in \{\forall, \exists\}$ and ψ is a quantifier-free conjunction of equality and predicate atoms, we have $\phi \equiv \text{qe}(\phi)$.*

PROOF. We give the proof only for the case $n = 1$ and $Q_1 = \exists$, the general case being an easy consequence. Suppose that $\psi = \varphi \wedge \theta(i_1)$, where i_1 does not occur within φ . If $\theta \models i_1 = i_j$ for some $j \neq 1$ then $\exists i_1 . \theta \equiv \theta[i_j/i_1]$. Otherwise, let $\theta = \bigwedge_{j \in J} \neg i_1 = i_j \wedge t_j(i_1)$, for some boolean terms t_j and show:

$$\exists i_1 . \theta \equiv \bigwedge_{K \subseteq J} \left(\text{distinct}(\{i_k\}_{k \in K}) \wedge \bigwedge_{k \in K} t_k(i_k) \right) \rightarrow |t| \geq ||K|| + 1$$

“ \Rightarrow ” Let $(\mathfrak{U}, \nu[i_1 \leftarrow u], \iota) \models \bigwedge_{j \in J} \neg i_1 = i_j \wedge t(i_1)$, for some $u \in \mathfrak{U}$ and let K be the maximal subset of J such that $\nu(i_{k_1}) \neq \nu(i_{k_2})$, for all $k_1 \neq k_2 \in K$ and $\nu(i_j) \in \iota(t)$. Since, moreover, $\nu(i_1) \notin \{\nu(i_k)\}_{k \in K}$, we obtain $\|\iota(t)\| \geq \|K\| + 1$.

“ \Leftarrow ” Let $(\mathfrak{U}, \nu, \iota)$ be a model of the right-hand side formula and let $K \subseteq J$ be a set such that $\nu(x_{k_1}) \neq \nu(x_{k_2})$ for all $k_1 \neq k_2 \in K$ and $\{\nu(x_k)\}_{k \in K} \in \iota(t)$. Then, since $\|\iota(t)\| \geq \|K\| + 1$, there exists $u \in \iota(t) \setminus \{\nu(x_k)\}_{k \in K}$ and thus $(\mathfrak{U}, \nu[i_1 \leftarrow u], \iota) \models \bigwedge_{j \in J} \neg x_1 = x_j \wedge t(x_1)$. \square

Example 4. (contd. from Example 2) The constraint $\Theta(\Gamma) \wedge \text{Init}(\mathcal{S})$ for the parametric system in Figure 1b is

$$\forall i. [r \vee w(i)] \rightarrow [s \vee u(i)] \wedge \forall i. [s \vee u(i)] \rightarrow [r \vee w(i)] \wedge (r \vee \exists i. w(i))$$

Let focus on the quantifier elimination of the first term of the conjunction. First, we use the double negation duality to rewrite the universal quantifier into an existential one, then standard Boolean algebra rules such that to achieve miniscoping of the existential quantifier. The term is progressively rewritten as follows:

$$\begin{aligned} \forall i. [r \vee w(i)] \rightarrow [s \vee u(i)] &\equiv \neg \exists i. \neg([r \vee w(i)] \rightarrow [s \vee u(i)]) \\ &\equiv \neg \exists i. ((r \vee w(i)) \wedge \neg(s \vee u(i))) \\ &\equiv \neg \exists i. ((r \vee w(i)) \wedge \neg s \wedge \neg u(i)) \\ &\equiv \neg \exists i. ((r \wedge \neg s \wedge \neg u(i)) \vee (w(i) \wedge \neg s \wedge \neg u(i))) \\ &\equiv \neg(\exists i. (r \wedge \neg s \wedge \neg u(i)) \vee \exists i. (w(i) \wedge \neg s \wedge \neg u(i))) \\ &\equiv \neg((r \wedge \neg s \wedge \exists i. \neg u(i)) \vee (\neg s \wedge \exists i. (w(i) \wedge \neg u(i)))) \end{aligned}$$

We can now perform quantifier elimination by taking respectively $\exists i. \neg u(i) \equiv |\neg u| \geq 1$ and $\exists i. (w(i) \wedge \neg u(i)) \equiv |w \wedge \neg u| \geq 1$. Therefore, the result of quantifier elimination of the first term proceeds as follows:

$$\begin{aligned} \forall i. [r \vee w(i)] \rightarrow [s \vee u(i)] &\equiv \neg((r \wedge \neg s \wedge |\neg u| \geq 1) \vee (\neg s \wedge |w \wedge \neg u| \geq 1)) \\ &\equiv \neg(r \wedge \neg s \wedge |\neg u| \geq 1) \wedge \neg(\neg s \wedge |w \wedge \neg u| \geq 1) \\ &\equiv (\neg r \vee s \vee |\neg u| \leq 0) \wedge (s \vee |w \wedge \neg u| \leq 0) \\ &\equiv s \vee (\neg r \vee |\neg u| \leq 0) \wedge |w \wedge \neg u| \leq 0 \\ &\equiv s \vee (\neg r \wedge |w \wedge \neg u| \leq 0) \vee (|\neg u| \leq 0 \wedge |w \wedge \neg u| \leq 0) \\ &\equiv s \vee (\neg r \wedge |w \wedge \neg u| \leq 0) \vee |\neg u| \leq 0 \end{aligned}$$

In the last step, we used that $|\neg u| \leq 0$ implies $|w \wedge \neg u| \leq 0$. Now, a similar result is obtained for the second term of our initial conjunction. Moreover, for the third term quantifier elimination can be directly applied by taking $\exists i. w(i) \equiv |w| \geq 1$. Finally, the complete result of quantifier elimination within $\Theta(\Gamma) \wedge \text{Init}(\mathcal{S})$ is

$$(s \vee (\neg r \wedge |w \wedge \neg u| \leq 0) \vee |\neg u| \leq 0) \wedge (r \vee (\neg s \wedge |u \wedge \neg w| \leq 0) \vee |\neg w| \leq 0) \wedge (r \vee |w| \geq 1)$$

which, after transformation into disjunctive normal form and simplification, becomes:

$$\begin{aligned} &(\neg r \wedge \neg s \wedge |w \wedge \neg u| \leq 0 \wedge |u \wedge \neg w| \leq 0 \wedge 1 \leq |w|) \vee \\ &(\neg r \wedge |w \wedge \neg u| \leq 0 \wedge |\neg w| \leq 0 \wedge 1 \leq |w|) \vee \\ &(s \wedge r) \vee (r \wedge |u| \leq 0) \vee (s \wedge |\neg w| \leq 0 \wedge 1 \leq |w|) \vee \\ &(\neg s \wedge |\neg u| \leq 0 \wedge |u \wedge \neg w| \leq 0 \wedge 1 \leq |w|) \vee \\ &(|\neg u| \leq 0 \wedge |\neg w| \leq 0 \wedge 1 \leq |w|) . \end{aligned}$$

The same procedure can be applied for the parametric system in Figure 3. As several (nested) variables are used in the definition of the trap constraint, and moreover subject to disequalities, quantifier elimination generates cardinality constraints of higher order (instead of 0 and 1 as for the previous example). The result after simplification and transformation into disjunctive normal form is

$$\begin{aligned} & (|u \wedge \neg w| \leq 0 \wedge 3 \leq |w|) \vee \\ & (|w \wedge \neg u| \leq 1 \wedge |u \wedge \neg w| \leq 0 \wedge 2 \leq |w|) \vee \\ & (|\neg u| \leq 1 \wedge |\neg u \wedge \neg w| \leq 0 \wedge |u \wedge \neg w| \leq 0 \wedge 1 \leq |w|) \vee \\ & (|w \wedge \neg u| \leq 0 \wedge |u \wedge \neg w| \leq 0 \wedge 1 \leq |w|) . \end{aligned}$$

□

4.2. Building Positive Formulae that Preserve Minimal Models

Let ϕ be a MIL formula, not necessarily positive. We shall build a positive formula ϕ^\oplus , such that $\phi \equiv^\mu \phi^\oplus$. By the result of the last section, ϕ is equivalent to a boolean combination of cardinality constraints $\text{qe}(\phi)$, obtained by quantifier elimination. Thus we assume w.l.o.g. that the DNF of ϕ is a disjunction of conjunctions of the form $\bigwedge_{i \in L} |t_i| \geq \ell_i \wedge \bigwedge_{j \in U} |t_j| \leq u_j$, for some sets of indices L, U and some positive integers $\{\ell_i\}_{i \in L}$ and $\{u_j\}_{j \in U}$.

For a boolean combination of cardinality constraints ψ , we denote by $P(\psi)$ the set of predicate symbols that occur in a boolean term of ψ and by $P^+(\psi)$ ($P^-(\psi)$) the set of predicate symbols that occur under an even (odd) number of negations in ψ . The following proposition allows to restrict the form of ϕ even further, without losing generality:

Proposition 1. *Given MIL formulae ϕ_1 and ϕ_2 , for any positivation operator $(.)^\oplus$, the following hold:*

1. $(\phi_1 \vee \phi_2)^\oplus \equiv^\mu \phi_1^\oplus \vee \phi_2^\oplus$,
2. $(\phi_1 \wedge \phi_2)^\oplus \equiv^\mu \phi_1^\oplus \wedge \phi_2^\oplus$, provided that $P(\phi_1) \cap P(\phi_2) = \emptyset$.

From now on, we assume that ϕ is a conjunction of cardinality constraints that cannot be split as $\phi = \phi_1 \wedge \phi_2$, such that $P(\phi_1) \cap P(\phi_2) = \emptyset$.

Let us consider a cardinality constraint $|t| \geq \ell$ that occurs in ϕ . Given a set \mathcal{P} of predicate symbols, for a set of predicates $S \subseteq \mathcal{P}$, the *complete* boolean minterm corresponding to S with respect to \mathcal{P} is $t_S^\mathcal{P} \stackrel{\text{def}}{=} \bigwedge_{p \in S} p \wedge \bigwedge_{p \in \mathcal{P} \setminus S} \neg p$. Moreover, let $\mathcal{S}_t \stackrel{\text{def}}{=} \{S \subseteq P(\phi) \mid t_S \rightarrow t\}$ be the set of sets S of predicate symbols for which the complete minterm t_S implies t . Finally, each cardinality constraint $|t| \geq \ell$ is replaced by the equivalent disjunction⁷, in which each boolean term is complete with respect to $P(\phi)$:

$$|t| \geq \ell \equiv \bigvee \left\{ \bigwedge_{S \in \mathcal{S}_t} |t_S^{\mathcal{P}(\phi)}| \geq \ell_S \mid \text{for some constants } \{\ell_S \in \mathbb{N}\}_{S \in \mathcal{S}_t} \text{ such that } \sum_{S \in \mathcal{S}_t} \ell_S = \ell \right\}$$

Note that because any two complete minterms t_S and t_T , for $S \neq T$, are incompatible, then necessarily $|t_S \vee t_T| = |t_S| + |t_T|$. Thus $|t_S \vee t_T| \geq \ell$ if and only if there exist $\ell_1, \ell_2 \in \mathbb{N}$ such that $\ell_1 + \ell_2 = \ell$ and $|t_S| \geq \ell_1$, $|t_T| \geq \ell_2$, respectively.

⁷The constraints $|t| \leq u$ are dealt with as $\neg(|t| \geq u + 1)$.

Notice that, restricting the sets of predicates in \mathcal{S}_t to subsets of $P(\phi)$, instead of the entire set of predicates, allows to apply Proposition 1 and reduce the number of complete minterms to be considered. That is, whenever possible, we write each minterm $\bigwedge_{i \in L} |t_i| \geq \ell_i \wedge \bigwedge_{j \in U} |t_j| \leq u_j$ in the DNF of ϕ as $\psi_1 \wedge \dots \wedge \psi_k$, such that $P(\psi_i) \cap P(\psi_j) = \emptyset$ for all $1 \leq i < j \leq k$. In practice, this optimisation turns out to be quite effective, as shown by the small execution times of our test cases, reported in §6.

The second step is building, for each conjunction $C = \bigwedge \{\ell_S \leq |t_S^{\mathcal{P}(\phi)}| \wedge |t_S^{\mathcal{P}(\phi)}| \leq u_S \mid S \subseteq P(\phi)\}$ ⁸, as above, a positive formula C^\oplus , that preserves its set of minimal models $\llbracket C \rrbracket^\mu$. The generalization to arbitrary boolean combinations of cardinality constraints is a direct consequence of Proposition 1. Let $\mathcal{L}^+(\phi)$ (resp. $\mathcal{L}^-(\phi)$) be the set of positive boolean combinations of predicate symbols $p \in \mathcal{P}^+(\phi)$ (resp. $\neg p$, where $p \in \mathcal{P}^-(\phi)$). Further, for a complete minterm $t_S^{\mathcal{P}}$, we write $t_S^{\mathcal{P}^+}$ ($t_S^{\mathcal{P}^-}$) for the conjunction of the positive (negative) literals in $t_S^{\mathcal{P}}$. Then, we define:

$$C^\oplus \stackrel{\text{def}}{=} \bigwedge \left\{ |\tau| \geq \sum_{t_S^{\mathcal{P}^+} \rightarrow \tau} \ell_S \mid \tau \in \mathcal{L}^+(\phi) \right\} \wedge \bigwedge \left\{ |\tau| \leq \sum_{t_S^{\mathcal{P}^-} \rightarrow \tau} u_S \mid \tau \in \mathcal{L}^-(\phi) \right\}$$

Here $\sum_{t_S^{\mathcal{P}^+} \rightarrow \tau} \ell_S$ (resp. $\sum_{t_S^{\mathcal{P}^-} \rightarrow \tau} u_S$) is the sum over all lower (resp. upper) bounds of the cardinalities of the positive (resp. negative) minterms that imply τ , in the boolean sense. It is not hard to see that C^\oplus is a positive MIL formula, because:

- for each $\tau \in \mathcal{L}^+(\phi)$, we have
 $|\tau| \geq k \equiv \exists i_1 \dots \exists i_k . \text{distinct}(i_1, \dots, i_k) \wedge \bigwedge_{j=1}^k \tau(j)$
- for each $\tau \in \mathcal{L}^-(\phi)$, we have
 $|\tau| \leq k \equiv \forall i_1 \dots \forall i_{k+1} . \text{distinct}(i_1, \dots, i_{k+1}) \rightarrow \bigvee_{j=1}^{k+1} \neg \tau(i_j)$

The following lemma proves that the above definition meets the second requirement of positivation operators, concerning the preservation of minimal models.

Lemma 9. *Given \mathcal{P} a finite set of monadic predicate symbols, $\{\ell_S \in \mathbb{N}\}_{S \subseteq \mathcal{P}}$ and $\{u_S \in \mathbb{N} \cup \{\infty\}\}_{S \subseteq \mathcal{P}}$ sets of constants, for any conjunction $C = \bigwedge \{\ell_S \leq |t_S^{\mathcal{P}}| \wedge |t_S^{\mathcal{P}}| \leq u_S \mid S \subseteq \mathcal{P}\}$, we have $C \equiv^\mu C^\oplus$.*

PROOF. completed in section 4.3 □

Example 5. (contd. from Example 4) Consider the first minterm of the DNF of the cardinality constraint obtained by quantifier elimination in Example 4, from the system in Figure 1b. The result of positivation for this minterm is given below:

$$(\neg r \wedge \neg s \wedge |w \wedge \neg u| \leq 0 \wedge |u \wedge \neg w| \leq 0 \wedge 1 \leq |w|)^\oplus = 1 \leq |u \wedge w|$$

Intuitively, the negative literals $\neg r$ and $\neg s$ may safely disappear, because no minimal model will assign r or s to true. Further, the constraints $|w \wedge \neg u| \leq 0$ and $|u \wedge \neg w| \leq 0$ are equivalent to the fact that, in any structure $\mathcal{I} = (\mathfrak{U}, \nu, \iota)$, we must have $\iota(u) = \iota(w)$. Finally, because $|w| \geq 1$, then necessarily $|u \wedge w| \geq 1$.

Similarly, the result of positivation applied to the second conjunct of the DNF cardinality constraint corresponding to the system in Figure 3 is given below:

$$(2 \leq |w| \wedge |w \wedge \neg u| \leq 1 \wedge |u \wedge \neg w| \leq 0)^\oplus = 2 \leq |w| \wedge 1 \leq |u \wedge w|$$

⁸Missing lower bounds ℓ_S are replaced with 0 and missing upper bounds u_S with ∞ .

Here, the number of elements in w is at least 2 and, in any structure $\mathcal{I} = (\mathfrak{U}, \nu, \iota)$, we must have $\iota(u) \subseteq \iota(w)$ and at most one element in $\iota(w) \setminus \iota(u)$. Consequently, the intersection of the sets $\iota(u)$ and $\iota(w)$ must contain at least one element, i.e. $|u \wedge w| \geq 1$. \square

4.3. The proof of positivation lemma

Lemma 9 states the most intricate technical result of the paper. Its proof requires several additional notions, which are the concern of this subsection. If t is any boolean term, its interpretation in the structure $\mathcal{I} = (\mathfrak{U}, \nu, \iota)$ is the set $t^{\mathcal{I}} \subseteq \mathfrak{U}$ defined recursively, as follows:

$$p^{\mathcal{I}} \stackrel{\text{def}}{=} \iota(p) \quad (\neg t)^{\mathcal{I}} \stackrel{\text{def}}{=} \mathfrak{U} \setminus t^{\mathcal{I}} \quad (t_1 \wedge t_2)^{\mathcal{I}} \stackrel{\text{def}}{=} t_1^{\mathcal{I}} \cap t_2^{\mathcal{I}} \quad (t_1 \vee t_2)^{\mathcal{I}} \stackrel{\text{def}}{=} t_1^{\mathcal{I}} \cup t_2^{\mathcal{I}}$$

Next, we generalize upward closures and upward closed sets from boolean valuations to first order structures as follows. If \mathcal{S} is a set of structures sharing the same universe, then $\mathcal{S} \uparrow \stackrel{\text{def}}{=} \{\mathcal{I} \mid \exists \mathcal{I}' \in \mathcal{S} . \mathcal{I}' \subseteq \mathcal{I}\}$ denotes its upward closure. Moreover, \mathcal{S} is upward closed iff $\mathcal{S} = \mathcal{S} \uparrow$. Then we have the following facts, whose proofs are folklore:

Fact 1. *Given a positive MIL formula ϕ , the set $\llbracket \phi \rrbracket$ is upward closed.*

PROOF. By induction on the structure of ϕ . \square

Fact 2. *Given MIL formulae ϕ_1 and ϕ_2 , we have $\phi_1 \equiv^\mu \phi_2$ if and only if $\llbracket \phi_1 \rrbracket \uparrow = \llbracket \phi_2 \rrbracket \uparrow$. If, moreover, ϕ_2 is positive, $\phi_1 \equiv^\mu \phi_2$ if and only if $\llbracket \phi_1 \rrbracket \uparrow = \llbracket \phi_2 \rrbracket$.*

PROOF. The first point is due to the observation $\llbracket \phi_i \rrbracket \uparrow = \{\mathcal{I} \mid \exists \mathcal{I}' \in \llbracket \phi_i \rrbracket^\mu . \mathcal{I}' \subseteq \mathcal{I}\}$, for $i = 1, 2$. The second point is obtained applying Fact 1. \square

Given a conjunction $C = \bigwedge \{\ell_S \leq |t_S^{\mathcal{P}}| \wedge |t_S^{\mathcal{P}}| \leq u_S \mid S \subseteq \mathcal{P}\}$ of cardinality constraints involving all complete minterms with respect to \mathcal{P} , for some arbitrary MIL formula ϕ , Lemma 9 requires showing that $\llbracket C \rrbracket \uparrow = \llbracket C^\oplus \rrbracket$. We shall do this in two stages:

1. We express $\llbracket C \rrbracket \uparrow$ using the reachability set of a vector addition system with states of a special form, that is, moreover, definable as the set of solutions of an integer linear system.
2. We use Hoffman's Circulation Theorem [31, Theorem 11.2] to show that the set of solutions of the linear system above defines $\llbracket C^\oplus \rrbracket$.

The developments of the two points rely on the observation that, each model of a cardinality constraint is uniquely defined, up to the renaming of its elements, by the positive cardinality of each complete minterm. In the following we shall consider this fact implicit and work with mappings of minterms into positive integer values, instead of first order structures.

4.3.1. Vector Addition Systems

The goal of this subsection is to reduce the problem $\llbracket C \rrbracket \uparrow = \llbracket C^\oplus \rrbracket$ of equivalence between sets of first order structures to the resolution of a linear integer system. To begin with observe that, given an arbitrary MIL formula ϕ , if $\mathcal{I} \models \phi$, then any structure obtained from \mathcal{I} by a renaming of its elements is also a model of ϕ . This is because ϕ uses only equalities and disequalities, which cannot distinguish the particular identity of elements. In other words, $\llbracket \phi \rrbracket$ is closed under isomorphic transformations of structures.

In the following, we assume that the finite set \mathcal{P} of predicate symbols is indexed by a total order \triangleleft . Then any set $S \subseteq \mathcal{P}$ corresponds to a word w_S which is the sequence of its elements, in the \triangleleft order. Moreover, let \triangleleft_{lex} be the lexicographic order induced by \triangleleft . The following definition introduces a total order on sets of predicate symbols, that is compatible with the subset ordering.

Definition 1. Given sets $S, T \subseteq \mathcal{P}$, where \mathcal{P} is totally ordered via \triangleleft , we define the total order $S \trianglelefteq^\dagger T$ if and only if one of the following holds:

1. $S \subseteq T$, or
2. $S \not\subseteq T$ and $w_S \triangleleft_{lex} w_T$.

As usual, we write $S \triangleleft^\dagger T$ for $S \trianglelefteq^\dagger T$ and $S \neq T$.

Let $\mathbf{t}^\mathcal{I} \stackrel{\text{def}}{=} \langle \|(t_S^p)^\mathcal{I}\| \rangle_{S \subseteq \mathcal{P}}$ be the vector of cardinalities of the interpretations for all complete minterms in the structure \mathcal{I} , arranged in the \trianglelefteq^\dagger order. In the following, we sometimes refer to this vector as the *cardinality vector* of the structure \mathcal{I} .

Since the interpretations of the complete minterms w.r.t \mathcal{P} are pairwise disjoint, for any predicate symbol $p \in \mathcal{P}$, we have $p^\mathcal{I} = \bigcup_{p \in S} (t_S^p)^\mathcal{I}$. Using the recursive definitions above, we can write any boolean term as a finite union of complete minterms, which corresponds to the DNF of the boolean formula associated with it. Hence, for any cardinality constraint $|t| \geq n$, we have $\mathcal{I} \models |t| \geq n$ if and only if $\sum_{i=1}^k \|(t_{S_i}^p)^\mathcal{I}\| \geq n$, where $t_{S_1}^p, \dots, t_{S_k}^p$ is the set of complete minterms that occur in the DNF of t . In general, for a boolean combination of cardinality constraints ϕ , we write $\mathbf{t}^\mathcal{I} \models \phi$ if and only if the formula obtained by replacing each term $|t|$ with the sum above is logically valid. A formal definition can be given recursively, on the structure of ϕ .

At this point, we can identify the set of models $\llbracket \phi \rrbracket$, where ϕ is any boolean combination of cardinality constraints, by the set of vectors $\{\mathbf{t}^\mathcal{I} \mid \mathbf{t}^\mathcal{I} \models \phi\}$, up to isomorphism of first order structures. It remains now to define upward closures in the same way. A first remark is that, because the set $\llbracket \phi \rrbracket$ is closed under isomorphism, so is its upward closure $\llbracket \phi \rrbracket^\uparrow$. However, the definition of $\llbracket \phi \rrbracket^\uparrow$ in terms of vectors $\mathbf{t}^\mathcal{I}$ requires a partial order that captures the pointwise inclusion between structures $\mathcal{I} \subseteq \mathcal{I}'$.

Definition 2. Given structures \mathcal{I} and \mathcal{I}' with the same universe, we define the relation $\mathbf{t}^{\mathcal{I}'} \prec_1 \mathbf{t}^\mathcal{I}$ if and only if there exists a set $S \subseteq \mathcal{P}$ and a predicate symbol $p \in S$ such that:

1. $\|(t_S^p)^\mathcal{I}\| = \|(t_S^p)^{\mathcal{I}'}\| + 1$,
2. $\|(t_{S \setminus \{p\}}^p)^\mathcal{I}\| = \|(t_{S \setminus \{p\}}^p)^{\mathcal{I}'}\| - 1$,
3. $\|(t_T^p)^\mathcal{I}\| = \|(t_T^p)^{\mathcal{I}'}\|$, for all $T \subseteq \mathcal{P}$, such that $T \neq S$ and $T \neq S \setminus \{p\}$.

We denote by \preceq the reflexive and transitive closure of the \prec_1 relation.

Lemma 10. For a boolean combination of cardinality constraints ϕ , the following hold:

1. $\llbracket \phi \rrbracket = \{\mathcal{I} \mid \mathbf{t}^\mathcal{I} \models \phi\}$,
2. $\llbracket \phi \rrbracket^\uparrow = \{\mathcal{I} \mid \exists \mathcal{I}' \in \llbracket \phi \rrbracket . \mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^\mathcal{I}\}$.

PROOF. (1) One shows that, for any structure \mathcal{I} , we have $\mathcal{I} \models \phi \iff \mathbf{t}^\mathcal{I} \models \phi$, by induction on the structure of ϕ . The base case $\phi = |t| \geq n$ is by definition and the inductive steps are routine. (2) We show that, for any structure $\mathcal{I} = (\mathcal{U}, \nu, \iota)$, the following are equivalent:

- (i) there exists $\mathcal{I}' \in \llbracket \phi \rrbracket$ such that $\mathcal{I}' \subseteq \mathcal{I}$, and

(ii) there exists $\mathcal{I}'' \in \llbracket \phi \rrbracket$ such that $\mathbf{t}^{\mathcal{I}''} \preceq \mathbf{t}^{\mathcal{I}}$.
(i) \Rightarrow (ii) We let $\mathcal{I}'' = \mathcal{I}'$ and prove $\mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}$. If, for all predicate symbols $p \in \mathcal{P}$, we have $p^{\mathcal{I}'} = p^{\mathcal{I}}$, then $\mathcal{I}' = \mathcal{I}$ and $\mathbf{t}^{\mathcal{I}'} = \mathbf{t}^{\mathcal{I}}$ follows. Assuming that this is not the case, let p be an arbitrary predicate symbol such that $p^{\mathcal{I}'} \subset p^{\mathcal{I}}$. We build a sequence of structures $\mathcal{I} = \mathcal{I}_0, \dots, \mathcal{I}_k$ such that $p^{\mathcal{I}_0} \supset \dots \supset p^{\mathcal{I}_k}$ and $\mathbf{t}^{\mathcal{I}_0} \succ_1 \dots \succ_1 \mathbf{t}^{\mathcal{I}_k}$. Let $u \in p^{\mathcal{I}} \setminus p^{\mathcal{I}'}$ be an element and let $S_u \stackrel{\text{def}}{=} \{q \in \mathcal{P} \mid u \in q^{\mathcal{I}}\}$. Clearly, we have that $p \in S_u$. Let $\mathcal{I}_1 = (\mathfrak{U}, \nu, \iota_1)$ be the structure such that $\iota_1(p) = \iota(p) \setminus \{u\}$ and $\iota_1(q) = \iota(q)$ for all $q \in \mathcal{P} \setminus \{p\}$. It is not hard to see that:

- $\|(t_{S_u}^{\mathcal{I}})^{\mathcal{I}}\| = \|(t_{S_u}^{\mathcal{I}_1})^{\mathcal{I}_1}\| + 1$,
- $\|(t_{S_u \setminus \{p\}}^{\mathcal{I}})^{\mathcal{I}}\| = \|(t_{S_u \setminus \{p\}}^{\mathcal{I}_1})^{\mathcal{I}_1}\| - 1$,
- $\|(t_T^{\mathcal{I}})^{\mathcal{I}}\| = \|(t_T^{\mathcal{I}_1})^{\mathcal{I}_1}\|$, for all $T \subseteq \mathcal{P}$, such that $T \neq S_u$ and $T \neq S_u \setminus \{p\}$.

By Definition 2, we have $\mathbf{t}^{\mathcal{I}_0} \succ_1 \mathbf{t}^{\mathcal{I}_1}$. We continue choosing elements $u \in p^{\mathcal{I}} \setminus p^{\mathcal{I}'}$ until no such elements can be found, then pick another predicate symbol for which \mathcal{I} and \mathcal{I}' differ. In this way we obtain a finite sequence of structures $\{\mathcal{I}_j\}_{j=0}^n$, such that $\mathcal{I}_j \succ_1 \mathcal{I}_{j+1}$ for all $0 \leq j < n$, thus $\mathbf{t}^{\mathcal{I}} \succeq \mathbf{t}^{\mathcal{I}'}$, as required.

(ii) \Rightarrow (i) By induction on the length of the sequence of structures $\mathcal{I} = \mathcal{I}_0, \dots, \mathcal{I}_k = \mathcal{I}''$ such that $\mathbf{t}^{\mathcal{I}_0} \succ_1 \dots \succ_1 \mathbf{t}^{\mathcal{I}_k}$. In the base case $k = 0$, we have $\mathbf{t}^{\mathcal{I}''} = \mathbf{t}^{\mathcal{I}}$, thus we have $\mathbf{t}^{\mathcal{I}} \models \phi$ and consequently $\mathcal{I} \in \llbracket \phi \rrbracket$, by point (1). For the induction step $k > 0$, we observe that $\mathbf{t}^{\mathcal{I}_0} \succ_1 \mathbf{t}^{\mathcal{I}_1}$ implies the existence of a structure $\mathcal{I}'_1 \subset \mathcal{I}_0$ which is isomorphic to \mathcal{I}_1 , thus $\mathbf{t}^{\mathcal{I}'_1} = \mathbf{t}^{\mathcal{I}_1}$. By the induction hypothesis, there exists $\mathcal{I}' \in \llbracket \phi \rrbracket$ such that $\mathcal{I}' \subseteq \mathcal{I}'_1$, hence $\mathcal{I}' \subseteq \mathcal{I}$, as required. \square

In the following, we define a vector addition system whose reachability relation matches the \preceq partial order on cardinality vectors $\mathbf{t}^{\mathcal{I}}$.

Definition 3. An n -dimensional vector addition system (VAS) is a finite set of vectors $V = \{\mathbf{v}_1, \dots, \mathbf{v}_k\} \subseteq \mathbb{Z}^n$.

A configuration of $V = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is a vector $\mathbf{c} \in \mathbb{N}^n$. The one-step reachability relation in V is $\mathbf{c} \xrightarrow{\mathbf{v}_i}_V \mathbf{c}'$ if and only if $\mathbf{c}' = \mathbf{c} + \mathbf{v}_i$, for some $1 \leq i \leq k$. The fact that $\mathbf{c}, \mathbf{c}' \in \mathbb{N}^n$ is important here, because configurations of a VAS are not allowed to contain negative values. For a finite sequence $\sigma = \mathbf{v}_{i_1} \dots \mathbf{v}_{i_k}$ of vectors from V , we write $\mathbf{c} \xrightarrow{\sigma}_V \mathbf{c}'$ for the sequence of transitions $\mathbf{c} \xrightarrow{\mathbf{v}_{i_1}}_V \mathbf{c}_1 \xrightarrow{\mathbf{v}_{i_2}}_V \dots \xrightarrow{\mathbf{v}_{i_k}}_V \mathbf{c}'$. Moreover, we write $\mathbf{c} \xrightarrow{*}_V \mathbf{c}'$ when σ is not important.

For a vector $\mathbf{v} \in \{-1, 0, 1\}^{2^{|\mathcal{P}|}}$ and a set $S \subseteq \mathcal{P}$, let $\mathbf{v}(S)$ be the entry in \mathbf{v} corresponding to S . Moreover, for some predicate symbol $p \in S$, we denote by $\mathbf{v}(S, p)$ the vector \mathbf{u} such that $\mathbf{u}(S) = -1$, $\mathbf{u}(S \setminus \{p\}) = 1$ and $\mathbf{u}(T) = 0$, for all $T \subseteq \mathcal{P}$ such that $T \neq S$ and $T \neq S \setminus \{p\}$. Intuitively, $\mathbf{v}(S, p)$ transfers an element from $t_S^{\mathcal{P}}$ into $t_{S \setminus \{p\}}^{\mathcal{P}}$, thus decreasing the cardinality of $t_S^{\mathcal{P}}$ and increasing that of $t_{S \setminus \{p\}}^{\mathcal{P}}$ by one, respectively.

We now define the $2^{|\mathcal{P}|}$ -dimensional VAS $V_{\mathcal{P}} \stackrel{\text{def}}{=} \{\mathbf{v}(S, p) \mid S \subseteq \mathcal{P}, p \in S\}$. This particular VAS captures the \preceq partial order on cardinality vectors as a reachability relation, as stated by the lemma below:

Lemma 11. For any two structures \mathcal{I} and \mathcal{I}' sharing the same universe, we have $\mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}$ if and only if $\mathbf{t}^{\mathcal{I}} \xrightarrow{*}_{V_{\mathcal{P}}} \mathbf{t}^{\mathcal{I}'}$.

PROOF. “ \Rightarrow ” For any two structures \mathcal{I}_1 and \mathcal{I}_2 , sharing the same universe, we have $\mathbf{t}^{\mathcal{I}_1} \prec_1 \mathbf{t}^{\mathcal{I}_2}$ iff there exists a set $S \subseteq \mathcal{P}$ and a predicate symbol $p \in S$ such that:

- $\mathbf{t}^{\mathcal{I}_2}(S) = \mathbf{t}^{\mathcal{I}_1}(S) + 1$,
- $\mathbf{t}^{\mathcal{I}_2}(S \setminus \{p\}) = \mathbf{t}^{\mathcal{I}_1}(S \setminus \{p\}) - 1$,
- $\mathbf{t}^{\mathcal{I}_2}(T) = \mathbf{t}^{\mathcal{I}_1}(T)$, for all $T \subseteq \mathcal{P}$, such that $T \neq S$ and $T \neq S \setminus \{p\}$.

Then, using the fact that $\mathbf{t}^{\mathcal{I}}(S) = \|(t_S^{\mathcal{I}})^{\mathcal{I}}\|$, for all $S \subseteq \mathcal{P}$, we establish that $\mathbf{t}^{\mathcal{I}_1} \prec_1 \mathbf{t}^{\mathcal{I}_2}$. Hence $\mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}$ implies the existence of a sequence σ of vectors from $V_{\mathcal{P}}$ such that $\mathbf{t}^{\mathcal{I}} \xrightarrow{\sigma_{V_{\mathcal{P}}}} \mathbf{t}^{\mathcal{I}'}$. “ \Leftarrow ” For any two configurations \mathbf{c} and \mathbf{c}' , if $\mathbf{c} \xrightarrow{v_{V_{\mathcal{P}}}} \mathbf{c}'$ for some vector $\mathbf{v} \in V_{\mathcal{P}}$, then $\mathbf{c} \succ_1 \mathbf{c}'$, by Definition 2. Consequently, $\mathbf{t}^{\mathcal{I}} \xrightarrow{\sigma_{V_{\mathcal{P}}}} \mathbf{t}^{\mathcal{I}'}$ implies $\mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}$, by straightforward induction on the length of σ . \square

For a tuple of variables $\mathbf{x} = \langle x_1, \dots, x_k \rangle$ and a valuation ν mapping these variables into \mathbb{Z} , we denote by $\nu(\mathbf{x})$ the tuple of integers $\langle \nu(x_1), \dots, \nu(x_k) \rangle$. The following lemma gives an equivalent condition for the existence of an execution in $V_{\mathcal{P}}$, that ends in a given configuration:

Lemma 12. *Let $\mathbf{x} = [x_S]_{S \subseteq \mathcal{P}}$ and $\mathbf{y} = [y_S]_{S \subseteq \mathcal{P}}$ be column vectors of variables and $\{k_{S,p} \mid S \subseteq \mathcal{P}, p \in S\}$ be variables. Then for any positive valuation ν of the variables \mathbf{x} and \mathbf{y} , the following are equivalent:*

1. ν can be extended to a positive solution of the integer linear system:

$$\mathbf{x} = \sum_{S \subseteq \mathcal{P}, p \in S} k_{S,p} \cdot \mathbf{v}(S, p) + \mathbf{y}$$

2. $\nu(\mathbf{y}) \xrightarrow{*_{V_{\mathcal{P}}}} \nu(\mathbf{x})$.

PROOF. “ \Rightarrow ” Consider the sequence of vectors $\nu(\mathbf{y}) = \mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_k = \nu(\mathbf{x})$, such that $\mathbf{c}_{i+1} = \mathbf{c}_i + \mathbf{v}(S_{i+1}, p_{i+1})$ for all $1 \leq i < k$ and the sequence of vectors $\mathbf{v}(S_1, p_1), \dots, \mathbf{v}(S_k, p_k)$ occur in order, each vector $\mathbf{v}(S_i, p_i)$ occurring $\nu(k_{S_i, p_i}) \geq 0$ times in the sequence. To show that this is an execution of $V_{\mathcal{P}}$, observe that each sequence of entries $\mathbf{c}_0(S), \dots, \mathbf{c}_k(S)$, for some $S \subseteq \mathcal{P}$ is first increased, then decreased, zero or more times. Because $\mathbf{c}_0(S) \geq 0$ and $\mathbf{c}_k(S) \geq 0$, we have that $\mathbf{c}_i(S) \geq 0$, for all $0 \leq i \leq k$. Since the choice of S was arbitrary, every vector \mathbf{c}_i has only positive entries, hence the sequence is an execution of $V_{\mathcal{P}}$. “ \Leftarrow ” Immediate, since in every execution $\nu(\mathbf{y}) \xrightarrow{*_{V_{\mathcal{P}}}} \nu(\mathbf{x})$, each vector $\mathbf{v}(S, p)$ occurs a positive number of times and let $\nu(k_{S,p})$ be that number. \square

Turning back to the original problem $\llbracket C \rrbracket^{\uparrow} = \llbracket C^{\oplus} \rrbracket$, we notice that the set of vectors $\{\mathbf{t}^{\mathcal{I}} \mid \exists \mathcal{I}' . \mathbf{t}^{\mathcal{I}'} \models C \wedge \mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}\}$, which corresponds (up to isomorphism) to the left-hand side of the required equality, is the set of vectors $\nu(\mathbf{y})$, where ν is a positive solutions of the linear system below:

$$\mathbf{x} = \sum_{S \subseteq \mathcal{P}, p \in S} k_{S,p} \cdot \mathbf{v}(S, p) + \mathbf{y} \wedge \bigwedge_{S \subseteq \mathcal{P}} \ell_S \leq \mathbf{x}(S) \leq u_S \quad (2)$$

The formal argument combines the results of Lemmas 10, 11 and 12. Next, we show that the right-hand side corresponds to the linear system obtained by eliminating the \mathbf{x} and $\{k_{S,p} \mid S \subseteq \mathcal{P}, p \in S\}$ variables from the above system.

4.3.2. Circulations in a Weighted Graph

We eliminate the $k_{S,p}$ variables from (2) using Hoffman's Circulation Theorem, given below. Let $G = (V, E)$ be a directed graph, where V is a finite set of vertices and $E \subseteq V \times V$ a set of edges. Further, we associate each edge in G a lower and upper capacity, formally $L : E \rightarrow \mathbb{N}$ and $U : E \rightarrow \mathbb{N} \cup \{\infty\}$, such that $L(e) \leq U(e)$, for all $e \in E$. For brevity, we call $G = (V, E, L, U)$ a *capacitated graph* in the following. Given a vertex $v \in V$, we denote by \bullet_v ($v\bullet$) the set of incoming (outgoing) edges with destination (source) v . We lift these notations to sets of vertices in the usual way. A *circulation* is a mapping $X : E \rightarrow \mathbb{N}$ such that, for all $v \in V$, we have $\sum_{e \in \bullet_v} X(e) = \sum_{e \in v\bullet} X(e)$ and $L(e) \leq X(e) \leq U(e)$, for all $e \in E$. The following is known as Hoffman's Circulation Theorem [31, Theorem 11.2]:

Theorem 3. *Given a capacitated graph $G = (V, E, L, U)$, there exists a circulation in G if and only if $\sum_{e \in \bullet_S} L(e) \leq \sum_{e \in S\bullet} U(e)$, for each set of vertices $S \subseteq V$.*

We encode the existence of positive solutions of the linear integer system (2) as a circulation problem in the capacitated graph $G_{\mathcal{P}}[\mathbf{y}] = (2^{\mathcal{P}} \cup \{\zeta\}, E_{\mathcal{P}}, L_{\mathcal{P}}, U_{\mathcal{P}})$, where:

- $\zeta \notin 2^{\mathcal{P}}$ is a special vertex, not a subset of \mathcal{P} ,
- \mathbf{y} is a tuple of parameters, indexed by sets of predicate symbols,
- for each set $S \subseteq \mathcal{P}$ there exists an edge $e = (\zeta, S)$, with $L_{\mathcal{P}}(e) = U_{\mathcal{P}}(e) = \mathbf{y}(S)$,
- for each set $S \subseteq \mathcal{P}$, there exists an edge $e = (S, \zeta)$, with $L_{\mathcal{P}}(e) = \ell_S$ and $U_{\mathcal{P}}(e) = u_S$,
- for each nonempty set $S \subseteq \mathcal{P}$ and each predicate symbol $p \in S$, there exists an edge $e = (S, S \setminus \{p\})$, with $L(e) = 0$ and $U(e) = \infty$.

Moreover, nothing else is in $E_{\mathcal{P}}$, $L_{\mathcal{P}}$ and $U_{\mathcal{P}}$, respectively. For example, given $\mathcal{P} = \{a, b, c\}$, the graph $G_{\mathcal{P}}$ is depicted in Figure 4. The following lemma relates the existence of positive solutions of the linear integer system (2) with the existence of a circulation in $G_{\mathcal{P}}[\mathbf{y}]$.

Lemma 13. *Given a set \mathcal{P} of predicate symbols and a positive valuation ν of the variables \mathbf{y} , the following are equivalent:*

- ν can be extended to a positive solution of the integer system (2),*
- the capacitated graph $G_{\mathcal{P}}[\nu(\mathbf{y})]$ has a circulation.*

PROOF. (a) \Rightarrow (b) Assume that ν is a positive solution of (2). We define the mapping $X : E \rightarrow \mathbb{N}$ as follows, for all $S \subseteq \mathcal{P}$:

- $X(e) = \nu(\mathbf{y}_S)$ if $e = (\zeta, S)$,
- $X(e) = \nu(\mathbf{x}_S)$ if $e = (S, \zeta)$,
- $X(e) = \nu(k_{S,p})$ if $e = (S, S \setminus \{p\})$, for some $p \in S$.

We prove that X is a circulation in $G_{\mathcal{P}}$. The condition $L_{\mathcal{P}}(e) \leq X(e) \leq U_{\mathcal{P}}(e)$, for all $e \in E_{\mathcal{P}}$ is immediate, because either $L_{\mathcal{P}}(e) = 0$ and $U_{\mathcal{P}}(e) = \infty$ or it follows directly from (2). It remains to check that $\sum_{e \in \bullet_u} X(e) = \sum_{e \in u\bullet} X(e)$, for any vertex $u \in 2^{\mathcal{P}} \cup \{\zeta\}$. If $u = \zeta$, we have $\sum_{S \subseteq \mathcal{P}} \nu(\mathbf{x}(S)) = \sum_{S \subseteq \mathcal{P}} \nu(\mathbf{y}(S))$, because the sum of the elements of each vector $\mathbf{v}(S, p)$ is zero, for any $S \subseteq \mathcal{P}$ and $p \in S$. Else, if u is some set $S \subseteq \mathcal{P}$, we have $\nu(\mathbf{x}(S)) + \sum_{p \in S} \nu(k_{S,p}) = \nu(\mathbf{y}(S)) + \sum_{q \notin S} \nu(k_{S \cup \{q\}, q})$.

(b) \Rightarrow (a) Given a circulation X in $G_{\mathcal{P}}$, we define ν as follows, for all $S \subseteq \mathcal{P}$:

- $\nu(\mathbf{y}(S)) = X(e)$, where $e = (\zeta, S)$,
- $\nu(\mathbf{x}(S)) = X(e)$, where $e = (S, \zeta)$,

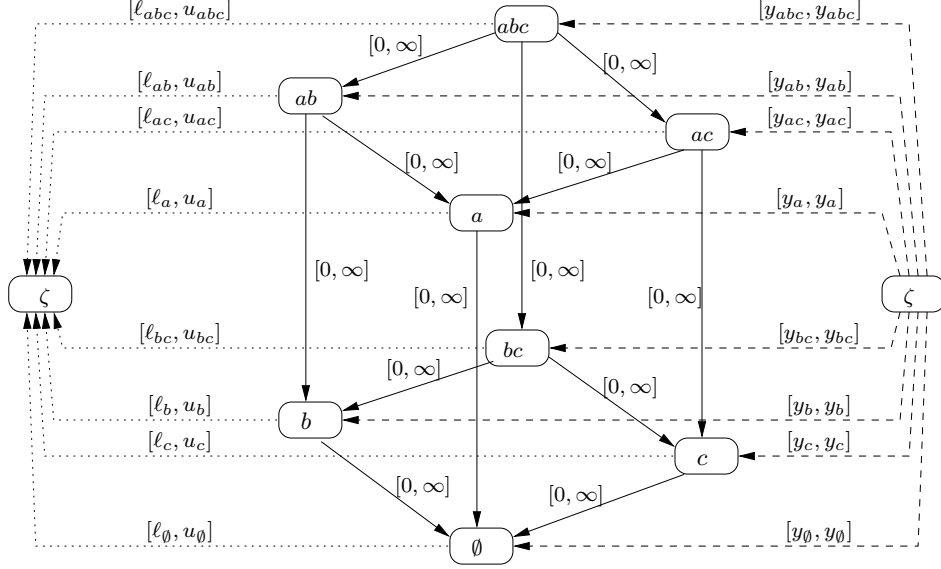


Figure 4: The Capacitated Graph $G_{\{a,b,c\}}[y]$ — the ζ node is duplicated, for clarity.

- $\nu(k_{S,p}) = X(e)$, where $e = (S, S \setminus \{p\})$, for some $p \in S$.

By definition, ν is a positive valuation. It remains to show that ν is indeed a solution of (2). The condition $\ell_S \leq \nu(\mathbf{x}(S)) \leq u_S$ is clearly satisfied for each $S \subseteq \mathcal{P}$, because $L_{\mathcal{P}}(e) = \ell_S$ and $U_{\mathcal{P}}(e) = u_S$, for each edge $e = (S, \zeta)$. To prove the remaining condition, observe that $\nu(\mathbf{x}(S)) + \sum_{p \in S} \nu(k_{S,p}) = \nu(\mathbf{y}(S)) + \sum_{q \notin S} \nu(k_{S \cup \{q\}, q})$, for each $S \subseteq \mathcal{P}$, which leads to $\nu(\mathbf{x}) = \sum_{S \subseteq \mathcal{P}, p \in S} \nu(k_{S,p}) \cdot \mathbf{v}(S, p) + \nu(\mathbf{y})$, as required. \square

Theorem 3 gives an equivalent condition for the existence of a circulation in $G_{\mathcal{P}}[y]$. In the following, we write another linear system, with unknowns y only, that captures this condition. A set of sets $\mathcal{S} \subseteq 2^{\mathcal{P}}$ is *downward closed* iff any subset of a set in \mathcal{S} is in \mathcal{S} . Dually, \mathcal{S} is *upward closed* iff any superset of a set in \mathcal{S} is in \mathcal{S} . It is easy to check that the complement of a downward (upward) closed set is upward (downward) closed.

Consider any capacitated graph $G_{\mathcal{P}}[y]$, e.g. refer to Figure 4 for an example. If $\mathcal{S} \subseteq 2^{\mathcal{P}}$ is not downward closed, then there exists an outgoing edge $e \in \mathcal{S}^\bullet$ with $U(e) = \infty$. Consequently, we have $\sum_{e \in \mathcal{S}^\bullet} U(e) = \infty$ in this case, thus the condition of Theorem 3 is trivially satisfied, for such sets. In the light of this remark, it is obvious that we need to consider only downward closed sets in order to characterize circulations in $G_{\mathcal{P}}[y]$, as in the example below:

Example 6. Consider the capacitated graph $G_{\{a,b,c\}}[y]$ from Figure 4. The necessary and sufficient condition for the existence of a circulation in $G_{\{a,b,c\}}[y]$, are partly shown below. By considering sets of vertices \mathcal{S} for every downward closed set $\mathcal{S} \subseteq 2^{\mathcal{P}}$ we obtain

the following inequalities:

$$\begin{aligned}
y_{\emptyset} &\leq u_{\emptyset} \\
y_a + y_{\emptyset} &\leq u_a + u_{\emptyset} \\
y_b + y_{\emptyset} &\leq u_b + u_{\emptyset} \\
y_c + y_{\emptyset} &\leq u_c + u_{\emptyset} \\
y_a + y_b + y_{\emptyset} &\leq u_a + u_b + u_{\emptyset} \\
y_b + y_c + y_{\emptyset} &\leq u_b + u_c + u_{\emptyset} \\
y_a + y_c + y_{\emptyset} &\leq u_a + u_c + u_{\emptyset} \\
y_a + y_b + y_c + y_{\emptyset} &\leq u_a + u_b + u_c + u_{\emptyset} \\
&\dots \\
y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_b + y_c + y_a + y_{\emptyset} &\leq u_{abc} + u_{bc} + u_{ac} + u_{ab} + u_b + u_c + u_a + u_{\emptyset}
\end{aligned}$$

These inequalities correspond to the enumeration of the downward closed sets \mathcal{S} in order $\{\emptyset\}, \{\{a\}, \emptyset\}, \{\{b\}, \emptyset\}, \{\{c\}, \emptyset\}, \{\{a\}, \{b\}, \emptyset\}, \{\{b\}, \{c\}, \emptyset\}, \{\{a\}, \{c\}, \emptyset\}, \{\{a\}, \{b\}, \{c\}, \emptyset\}, \dots, \{\{a, b, c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a\}, \{b\}, \{c\}, \emptyset\} = 2^{\mathcal{P}}$. In similar way, by considering sets of vertices $\mathcal{S} \cup \{\zeta\}$ for every downward closed set $\mathcal{S} \subseteq 2^{\mathcal{P}}$ we obtain the following inequalities:

$$\begin{aligned}
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_b + \ell_c + \ell_a + \ell_{\emptyset} &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_b + y_c + y_a + y_{\emptyset} \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_b + \ell_c + \ell_a &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_b + y_c + y_a \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_b + \ell_c &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_b + y_c \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_a + \ell_c &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_a + y_c \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_a + \ell_b &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_a + y_b \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_c &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_c \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_a &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_a \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} + \ell_b &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} + y_b \\
\ell_{abc} + \ell_{bc} + \ell_{ac} + \ell_{ab} &\leq y_{abc} + y_{bc} + y_{ac} + y_{ab} \\
&\dots \\
\ell_{abc} &\leq y_{abc}
\end{aligned}$$

These inequalities correspond to the enumeration of downward closed sets in order $\emptyset, \{\emptyset\}, \{\{a\}, \emptyset\}, \{\{b\}, \emptyset\}, \{\{c\}, \emptyset\}, \{\{a\}, \{b\}, \emptyset\}, \{\{b\}, \{c\}, \emptyset\}, \{\{a\}, \{c\}, \emptyset\}, \{\{a\}, \{b\}, \{c\}, \emptyset\}, \dots, \{\{a, b\}, \{b, c\}, \{a, c\}, \{a\}, \{b\}, \{c\}, \emptyset\} = 2^{\mathcal{P}} \setminus \{\{a, b, c\}\}$. \square

At this point it is easy to generalize the above example and infer an equivalent condition for the existence of a circulation in $G_{\mathcal{P}}[\mathbf{y}]$:

$$\bigwedge_{S \in 2^{\mathcal{P}}} \left(\sum_{S \in \mathcal{S}} y_S \leq \sum_{S \in \mathcal{S}} u_S \wedge \sum_{S \notin \mathcal{S}} \ell_S \leq \sum_{S \notin \mathcal{S}} y_S \right) \quad (3)$$

Let us now turn to the definition of C^{\oplus} , given in terms of complete minterms, and notice the following facts:

1. for each positive boolean combination $\tau \in \mathcal{L}^+(\phi)$, the set $\{S \subseteq \mathcal{P} \mid t_S^{\mathcal{P}+} \rightarrow \tau\}$ is upward closed and its set of minimal elements corresponds to the minterms of τ in DNF,
2. dually, for each negative boolean combination $\tau \in \mathcal{L}^-(\phi)$, the set $\{S \subseteq \mathcal{P} \mid t_S^{\mathcal{P}-} \rightarrow \tau\}$ is downward closed and its set of maximal elements corresponds to the minterms of τ in DNF,

3. because the complete minterms are pairwise disjoint, in each structure \mathcal{I} , we have $\tau^{\mathcal{I}} = \bigcup_{t_S^{\mathcal{P}} \rightarrow \tau} (t_S^{\mathcal{P}})^{\mathcal{I}}$, for all $\tau \in \mathcal{L}^+(\phi)$ and $\tau^{\mathcal{I}} = \bigcup_{t_S^{\mathcal{P}} \rightarrow \tau} (t_S^{\mathcal{P}})^{\mathcal{I}}$, for all $\tau \in \mathcal{L}^-(\phi)$,
4. for each positive solution ν of (3), there exists a structure $\mathcal{I} \in \llbracket C^{\oplus} \rrbracket$ such that $\nu(y_S) = \|(t_S^{\mathcal{P}})^{\mathcal{I}}\|$ and viceversa, each structure $\mathcal{I} \in \llbracket C^{\oplus} \rrbracket$ induces a positive solution of (3), where $\nu(y_S) = \|(t_S^{\mathcal{P}})^{\mathcal{I}}\|$, for all $S \subseteq \mathcal{P}$.

To summarize, we prove that $C \equiv^{\mu} C^{\oplus}$ by proving the equivalent statement $\llbracket C \rrbracket \uparrow = \llbracket C^{\oplus} \rrbracket$. Since both the left and the right-hand side of this equality are sets of structures closed under isomorphism, we reduce the problem to an equivalence between sets of integer tuples $\{\mathbf{t}^{\mathcal{I}} \mid \mathcal{I} \in \llbracket C \rrbracket \uparrow\} = \{\mathbf{t}^{\mathcal{I}} \mid \mathcal{I} \in \llbracket C^{\oplus} \rrbracket\}$. By Lemma 10, this is equivalent to $\{\mathbf{t}^{\mathcal{I}} \mid \exists \mathcal{I}' \in \llbracket C \rrbracket . \mathbf{t}^{\mathcal{I}'} \preceq \mathbf{t}^{\mathcal{I}}\} = \{\mathbf{t}^{\mathcal{I}} \mid \mathcal{I} \in \llbracket C^{\oplus} \rrbracket\}$. Subsequently, Lemmas 11 and 12 prove that the left-hand side of the latter equality is the set of positive solutions of the linear system (2), restricted to the tuple of variables $\mathbf{y} = \langle y_S \rangle_{S \subseteq \mathcal{P}}$. By Hoffman's Circulation Theorem (Theorem 3), this is the set of positive solutions to the linear system (3), obtained from the elimination of the \mathbf{x} and $k_{S,p}$ variables from (2). Finally, this set is exactly the right-hand side of the equality above, as a result of interpreting the definition of C^{\oplus} in terms of vertices of the capacitated graph $G_{\mathcal{P}}[\mathbf{y}]$, on which the circulation theorem was applied.

5. Proving Deadlock Freedom of Parametric Systems

We have gathered all the ingredients necessary for checking deadlock freedom of parametric systems, using our method based on trap invariant generation (Figure 5). In particular, we derive a trap constraint $\Theta(\Gamma)$ directly from the interaction formula Γ , both of which are written in MIL. Second, we compute a positive formula that preserves the set of minimal models of $\Theta(\Gamma) \wedge \text{Init}(\mathcal{S})$, by first converting the MIL formula into a quantifier-free cardinality constraint, using quantifier elimination, and deriving a positive MIL formula from the latter.

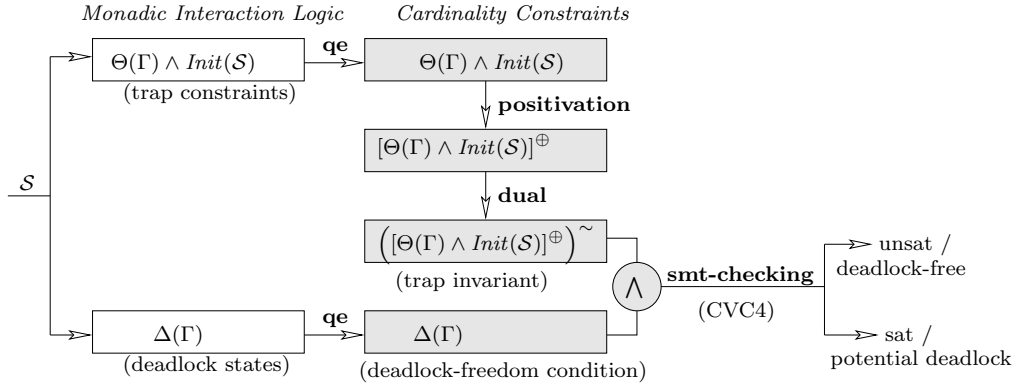


Figure 5: Verification of Parametric Component-based Systems

The conjunction between the dual of this positive formula and the formula $\Delta(\Gamma)$ that defines the deadlock states is then checked for satisfiability. Formally, given a

parametric system \mathcal{S} , with an interaction formula Γ written in the form (1), the MIL formula characterizing the deadlock states of the system is the following:

$$\Delta(\Gamma) \stackrel{\text{def}}{=} \forall i_1 \dots \forall i_\ell . \varphi \rightarrow \left[\bigvee_{j=1}^\ell \neg^\bullet p_j(i_j) \vee \bigvee_{j=\ell+1}^{\ell+m} \exists i_j . \psi_j \wedge \neg^\bullet p_j(i_j) \right]$$

We state a sufficient verification condition for deadlock freedom in the parametric case:

Corollary 2. *A parametric system $\mathcal{S} = \langle C^1, \dots, C^n, \mathbf{M}, \Gamma \rangle$ is deadlock-free if*

$$\left((\Theta(\Gamma) \wedge \text{Init}(\mathcal{S}))^\oplus \right)^\sim \wedge \Delta(\Gamma) \rightarrow \perp$$

The satisfiability check is carried out using the conversion to cardinality constraints via quantifier elimination §4.1 and an effective set theory solver for cardinality constraints, implemented in the CVC4 SMT solver [25].

6. Experimental Results

To assess our method for proving deadlock freedom of parametric component-based systems, we ran a number of experiments on systems with a small numbers of rather simple component types, but with nontrivial interaction patterns, given by MIL formulae. We check deadlock freedom of parameterized component-based systems, in which the number of instances of each component type is finite but unbounded. The **task-sem** i/n examples, $i = 1, 2, 3$, are generalizations of the parametric *Task-Semaphore* example depicted in Figure 1b, in which n *Tasks* synchronize using n *Semaphores*, such that i *Tasks* interact with a single *Semaphore* at once, in a multiparty rendez-vous. In a similar vein, the **broadcast** i/n examples, $i = 2, 3$ are generalizations of the system in Figure 3, in which i out of n *Workers* engage in rendez-vous on the b port, whereas all the other stay idle — here idling is modeled as a broadcast on the a ports. Next, in the **sync** i/n examples, $i = 1, 2, 3$, we consider systems composed of n *Workers* (Figure 1b) such that either i out of n instances simultaneously interact on the b ports, or all interact on the f ports. Notice that, for $i = 2, 3$, these systems have a deadlock if and only if $n \not\equiv 0 \pmod i$. This is because, if $n = m \pmod i$, for some $0 < m < i$, there will be m instances that cannot synchronize on their b port, in order to move from w to u , in order to engage in the f broadcast.

Finally, the **synapse** and **mesi** examples refer to models of cache coherence protocols from [32]. In both cases, the model consists of n identical processes accessing and updating their local cache containing the copy of some globally shared data. Every process can be either actively reading (r) or writing (w) the shared data, or gets notified about the reading (\bar{r}) or writing (\bar{w}) by another process. All interactions are therefore of type broadcast, that is, every active reading or writing by some process notifies instantaneously all the other processes such that they can consistently update their cache status. The property of interest was cache coherency, that is, (1) at most one process has a dirty cache and (2) no dirty cache coexists with one or more caches in shared state. For both protocols, the coherency property was successfully proven valid.

All experiments were carried out on a Intel(R) Xeon(R) CPU @ 2.00GHz virtual machine with 4GB of RAM. Table 1 shows separately the times needed to generate the proof obligations (trap invariants and deadlock states) from the interaction formulae and

example	interaction formula	t-gen	t-smt	result
task-sem 1/n	$\exists i \exists j_1. a(i) \wedge b(j_1)$ $\bigvee \exists i \exists j_1. e(i) \wedge f(j_1)$	22 ms	20 ms	unsat
task-sem 2/n	$\exists i \exists j_1 \exists j_2. j_1 \neq j_2 \wedge a(i) \wedge b(j_1) \wedge b(j_2)$ $\bigvee \exists i \exists j_1 \exists j_2. j_1 \neq j_2 \wedge e(i) \wedge f(j_1) \wedge f(j_2)$	34 ms	40 ms	unsat
task-sem 3/n	$\exists i \exists j_1 \exists j_2 \exists j_3. \text{distinct}(j_1, j_2, j_3) \wedge$ $a(i) \wedge b(j_1) \wedge b(j_2) \wedge b(j_3)$ $\bigvee \exists i \exists j_1 \exists j_2 \exists j_3. \text{distinct}(j_1, j_2, j_3) \wedge$ $e(i) \wedge f(j_1) \wedge f(j_2) \wedge f(j_3)$	73 ms	40 ms	unsat
broadcast 2/n	$\exists i_1 \exists i_2. i_1 \neq i_2 \wedge b(i_1) \wedge b(i_2) \wedge$ $\forall j. j \neq i_1 \wedge j \neq i_2 \rightarrow a(j) \bigvee \exists i. f(i)$	14 ms	20 ms	unsat
broadcast 3/n	$\exists i_1 \exists i_2 \exists i_3. \text{distinct}(i_1, i_2, i_3) \wedge$ $b(i_1) \wedge b(i_2) \wedge b(i_3) \wedge$ $\forall j. j \neq i_1 \wedge j \neq i_2 \wedge j \neq i_3 \rightarrow a(j)$ $\bigvee \exists i. f(i)$	409 ms	20 ms	unsat
sync 1/n	$\exists i. b(i) \bigvee \forall i. f(i)$	5 ms	20 ms	unsat
sync 2/n	$\exists i_1 \exists i_2. i_1 \neq i_2 \wedge b(i_1) \wedge b(i_2) \bigvee \forall i. f(i)$	7 ms	50 ms	sat
sync 3/n	$\exists i_1 \exists i_2 \exists i_3. \text{distinct}(i_1, i_2, i_3) \wedge$ $b(i_1) \wedge b(i_2) \wedge b(i_3) \bigvee \forall i. f(i)$	11 ms	40 ms	sat
synapse n	$\exists i. r(i) \wedge \forall j. j \neq i \rightarrow \bar{r}(i) \bigvee$ $\exists i. w(i) \wedge \forall j. j \neq i \rightarrow \bar{w}(i)$	45 ms	70 ms	unsat
mesi n	$\exists i. r(i) \wedge \forall j. j \neq i \rightarrow \bar{r}(i) \bigvee$ $\exists i. w(i) \wedge \forall j. j \neq i \rightarrow \bar{w}(i)$	508 ms	90 ms	unsat

Table 1: Benchmarks - *t-gen* provides the time of invariant generation (that is, quantifier elimination, positivation and dualization), *t-smt* provides the time of satisfiability checking (using the CVC4 solver) and *result* provides the verification result (that is, *unsat*: property successfully verified, *sat*: inconclusive, property cannot be verified by our method)

the times needed by CVC4 1.7 to show unsatisfiability or come up with a model. All systems considered, for which deadlock freedom could not be shown using our method, have a real deadlock scenario that manifests only under certain modulo constraints on the number n of instances. These constraints cannot be captured by MIL formulae, or, equivalently by cardinality constraints, and would require cardinality constraints of the form $|t| = n \bmod m$, for some constants $n, m \in \mathbb{N}$.

7. Conclusions

This work is part of a lasting research program on BIP linking two work directions: (1) recent work on modeling architectures using interaction logics, and (2) older work on verification by using invariants. Its rationale is to overcome as much as possible complexity and undecidability issues by proposing methods which are adequate for the verification of essential system properties.

The presented results are applicable to a large class of architectures characterized by the MIL. A key technical result is the translation of MIL formulas into cardinality constraints. This allows on the one hand the computation of the MIL formula characterizing the minimal trap invariant. On the other hand, it provides a decision procedure for MIL, that leverages from recent advances in SMT, implemented in the CVC4 solver [25].

Our approach sheds new light on the intricacy of the interaction structure between components. This clearly depends on the topology of the architecture but also on the multiplicity of interactions. Centralized control systems seem to be the easier to verify (parametric systems with single controller and without interaction between components). For distributed control systems, easier to check seem to be systems where interactions between components are uniform each component of a class interacts in the same manner with all the other components.

The hardest case corresponds to systems where interaction between components depends on a neighborhood which usually implies some arithmetic relation between indices. To model such systems MIL should be extended with arithmetic predicates on indices. This is the objective of an ongoing work direction [33].

References

- [1] S. M. German, A. P. Sistla, Reasoning about systems with many processes, J. ACM 39 (3) (1992) 675–735.
- [2] R. Bloem, S. Jacobs, A. Khalimov, I. Konnov, S. Rubin, H. Veith, J. Widder, Decidability of Parameterized Verification, Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers, 2015.
- [3] M. C. Browne, E. M. Clarke, O. Grumberg, Reasoning about networks with many identical finite state processes, Inf. Comput. 81 (1) (1989) 13–31.
- [4] E. A. Emerson, K. S. Namjoshi, Reasoning about rings, in: R. K. Cytron, P. Lee (Eds.), Conference Record of POPL’95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM Press, 1995, pp. 85–94.
- [5] B. Aminof, T. Kotek, S. Rubin, F. Spegner, H. Veith, Parameterized model checking of rendezvous systems, Distributed Comput. 31 (3) (2018) 187–222.
- [6] E. A. Emerson, V. Kahlon, Reducing model checking of the many to the few, in: D. A. McAllester (Ed.), Automated Deduction - CADE-17, 17th International Conference on Automated Deduction, Proceedings, Vol. 1831 of LNCS, Springer, 2000, pp. 236–254.

- [7] S. Außerlechner, S. Jacobs, A. Khalimov, Tight cutoffs for guarded protocols with fairness, in: B. Jobstmann, K. R. M. Leino (Eds.), *Verification, Model Checking, and Abstract Interpretation - 17th International Conference, VMCAI 2016, Proceedings*, Vol. 9583 of LNCS, Springer, 2016, pp. 476–494.
- [8] S. Jacobs, M. Sakr, Analyzing guarded protocols: Better cutoffs, more systems, more expressivity, in: I. Dillig, J. Palsberg (Eds.), *Verification, Model Checking, and Abstract Interpretation - 19th International Conference, VMCAI 2018, Proceedings*, Vol. 10747 of LNCS, Springer, 2018, pp. 247–268.
- [9] P. A. Abdulla, K. Cerans, B. Jonsson, Y. Tsay, General decidability theorems for infinite-state systems, in: *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, LICS, IEEE Computer Society*, 1996, pp. 313–321.
- [10] A. Finkel, P. Schnoebelen, Well-structured transition systems everywhere!, *Theor. Comput. Sci.* 256 (1-2) (2001) 63–92.
- [11] Y. Kesten, O. Maler, M. Marcus, A. Pnueli, E. Shahar, Symbolic model checking with rich assertional languages, *Theor. Comput. Sci.* 256 (1-2) (2001) 93–112.
- [12] P. A. Abdulla, G. Delzanno, N. B. Henda, A. Rezine, Regular model checking without transducers (on efficient verification of parameterized systems), in: O. Grumberg, M. Huth (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Proceedings*, Vol. 4424 of LNCS, Springer, 2007, pp. 721–736.
- [13] F. Alberti, S. Ghilardi, N. Sharygina, A framework for the verification of parameterized infinite-state systems, in: L. Giordano, V. Gliozzi, G. L. Pozzato (Eds.), *Proceedings of the 29th Italian Conference on Computational Logic*, Vol. 1195 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014, pp. 303–308.
- [14] S. Conchon, A. Goel, S. Krstic, A. Mebsout, F. Zaïdi, Cubicle: A parallel smt-based model checker for parameterized systems - tool paper, in: P. Madhusudan, S. A. Seshia (Eds.), *Computer Aided Verification, 24th International Conference, CAV 2012, Proceedings*, Vol. 7358 of LNCS, Springer, 2012, pp. 718–724.
- [15] K. Baukus, S. Bensalem, Y. Lakhnech, K. Stahl, Abstracting WS1S systems to verify parameterized networks, in: S. Graf, M. I. Schwartzbach (Eds.), *Tools and Algorithms for Construction and Analysis of Systems, 6th International Conference, TACAS 2000, Proceedings*, Vol. 1785 of LNCS, Springer, 2000, pp. 188–203.
- [16] A. Bouajjani, P. Habermehl, T. Vojnar, Abstract regular model checking, in: R. Alur, D. A. Peled (Eds.), *Computer Aided Verification, 16th International Conference, CAV 2004, Proceedings*, Vol. 3114 of LNCS, Springer, 2004, pp. 372–386.
- [17] Y. Chen, C. Hong, A. W. Lin, P. Rümmer, Learning to prove safety over parameterised concurrent systems, in: D. Stewart, G. Weissenbacher (Eds.), *2017 Formal Methods in Computer Aided Design, FMCAD 2017, IEEE*, 2017, pp. 76–83.
- [18] P. Cousot, R. Cousot, Systematic design of program analysis frameworks, in: A. V. Aho, S. N. Zilles, B. K. Rosen (Eds.), *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages*, ACM Press, 1979, pp. 269–282.
- [19] D. Dams, Y. Lakhnech, M. Steffen, Iterating transducers, *J. Log. Algebr. Meth. Program.* 52-53 (2002) 109–127.
- [20] A. Pnueli, S. Ruah, L. D. Zuck, Automatic deductive verification with invisible invariants, in: T. Margaria, W. Yi (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems, 7th International Conference, TACAS 2001, Proceedings*, Vol. 2031 of LNCS, Springer, 2001, pp. 82–97.
- [21] E. M. Clarke, M. Talupur, H. Veith, Environment abstraction for parameterized verification, in: E. A. Emerson, K. S. Namjoshi (Eds.), *Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Proceedings*, Vol. 3855 of LNCS, Springer, 2006, pp. 126–141.
- [22] J. Sifakis, Structural properties of petri nets, in: J. Winkowski (Ed.), *Mathematical Foundations of Computer Science 1978, Proceedings, 7th Symposium*, Vol. 64 of LNCS, Springer, 1978, pp. 474–483.
- [23] S. Bensalem, M. Bozga, T. Nguyen, J. Sifakis, D-finder: A tool for compositional deadlock detection and verification, in: A. Bouajjani, O. Maler (Eds.), *Computer Aided Verification, 21st International Conference, CAV 2009, Proceedings*, Vol. 5643 of LNCS, Springer, 2009, pp. 614–619.
- [24] V. Kuncak, H. H. Nguyen, M. C. Rinard, Deciding boolean algebra with Presburger arithmetic, *J. Autom. Reason.* 36 (3) (2006) 213–239.
- [25] C. W. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanovic, T. King, A. Reynolds,

- C. Tinelli, CVC4, in: G. Gopalakrishnan, S. Qadeer (Eds.), *Computer Aided Verification*, 23rd International Conference, CAV 2011, Proceedings, Vol. 6806 of LNCS, Springer, 2011, pp. 171–177.
- [26] A. Basu, S. Bensalem, M. Bozga, J. Combaz, M. Jaber, T. Nguyen, J. Sifakis, Rigorous component-based system design using the BIP framework, *IEEE Software* 28 (3) (2011) 41–48.
 - [27] L. Lowenheim, Über Möglichkeiten im Relativkalkül, *Math. Ann* 470 (1915) 76–447.
 - [28] K. Bansal, A. Reynolds, C. W. Barrett, C. Tinelli, A new decision procedure for finite sets and cardinality constraints in SMT, in: N. Olivetti, A. Tiwari (Eds.), *Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Proceedings*, Vol. 9706 of LNCS, Springer, 2016, pp. 82–98.
 - [29] M. Bozga, R. Iosif, J. Sifakis, Checking deadlock-freedom of parametric component-based systems, in: T. Vojnar, L. Zhang (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS 2019, Proceedings*, Vol. 11428 of LNCS, Springer, 2019, pp. 3–20.
 - [30] T. Skolem, Untersuchungen über die axiome des klassenkalküls and über "produktions- und summationsprobleme", elche gewisse klassen von aussagen betreffen, *Skrifter utgit av Videnskapsselskapet i Kristiania, I. klasse, no. 3*, Oslo (1919).
 - [31] A. Schrijver, *Combinatorial Optimization - Polyhedra and Efficiency*, Vol. A, Springer, 2003.
 - [32] G. Delzanno, Automatic verification of parameterized cache coherence protocols, in: E. A. Emerson, A. P. Sistla (Eds.), *Computer Aided Verification*, 12th International Conference, CAV 2000, Proceedings, Vol. 1855 of LNCS, Springer, 2000, pp. 53–68.
 - [33] M. Bozga, J. Esparza, R. Iosif, J. Sifakis, C. Welzel, Structural invariants for the verification of systems with parameterized architectures, in: A. Biere, D. Parker (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Proceedings*, Vol. 12078 of LNCS, Springer, 2020, pp. 228–246.