Petri Net Analysis using Invariant Generation

Sriram Sankaranarayanan, Henny Sipma and Zohar Manna *

Computer Science Department
Stanford University
Stanford, USA
{srirams,sipma,manna}@cs.stanford.edu

Abstract. Petri nets have been widely used to model and analyze concurrent systems. Their wide-spread use in this domain is, on one hand, facilitated by their simplicity and expressiveness. On the other hand, the analysis of Petri nets for questions like reachability, boundedness and deadlock freedom can be surprisingly hard. In this paper, we model Petri nets as transition systems. We exploit the special structure in these transition systems to provide an exact and closed-form characterization of all its inductive linear invariants. We then exploit this characterization to provide an invariant generation technique that we demonstrate to be efficient and powerful in practice. We compare our work with those in the literature and discuss extensions.

1 Introduction

Petri nets provide a concise and expressive representation to model the behavior of concurrent systems [13, 12]. They have been widely used to model a variety of systems including communication protocols and flexible manufacturing systems. Due to their expressiveness, the analysis of Petri nets is hard. Because of this many restricted classes of Petri nets have evolved to make analysis such as reachability and checking for deadlock feasible for practical systems.

In this paper, we analyze the reachable markings of general Petri nets using *invariants*. An invariant is an assertion that is true in all reachable states of a program. Invariants have been well studied for the analysis of many types of programs [7, 9]. Traditionally, invariants have been generated using *abstract interpretation* [6]. This technique generates invariants iteratively, starting from an initial estimate, and improving the estimate at each stage iteratively until no more improvements can be made. The convergence of the algorithm is not guaranteed, and often termination is imposed by a guessing operation called *widening* [7, 9, 2]. However, widening introduces inaccuracies in the process, often resulting in invariants that are too weak to be useful. Some tools like HyTech forgo widening in favor of stronger invariants [10].

* This research was supported in part by NSF grants CCR-01-21403, CCR-02-20134 and CCR-02-09237, by ARO grant DAAD19-01-1-0723, by ARPA/AF contracts F33615-00-C-1693 and F33615-99-C-3014, and by NAVY/ONR contract N00014-03-1-0939.

In [5] we introduced a new method for invariant generation that uses a direct method to compute linear invariants. The conditions for being an invariant are encoded as a set of inequalities. The solution of this set of inequalities then provides a generator for the coefficients of a family of invariants. For the general case of transition systems, the set of inequalities is nonlinear (quadratic) and hence computing the solution is, with current constraint-solving tools, impractical for all but the smallest systems. In this paper we specialize this method for Petri nets and show that by exploiting the structure of transitions in general Petri nets, the same set of conditions can be encoded in systems of linear inequalities, thus making this technique applicable to much larger systems.

Our method of invariant generation is sound and complete for inductive linear inequalities. It marks an advance over earlier methods that use *state equations* [15]. Even though the *state equations* are useful in deducing different structural properties of the Petri net, their use in the analysis of Petri nets is made difficult partly because Petri nets are inherently non-deterministic and many analyses using these equations do not handle the transition guards (which are inequalities) exactly. In this paper, we use the theory of invariants of programs which were originally developed to prove partial and total correctness of imperative programs to Petri nets in order to generate invariant assertions that hold at all the reachable states of the net.

The rest of the paper is organized as follows: Section 2 defines transition systems and Petri nets. In Section 3 we describe our technique for invariant generation and in Section 4 we show how this technique can be specialized to the case of Petri nets. Section 5 demonstrates the technique on a moderately sized manufacturing system example reported in the literature. Section 6 discusses a scheme to strengthen the invariants and Section 7 concludes the paper with some observations regarding extensibility of our results.

2 Preliminaries

We first define transition systems and Petri nets, and then show that Petri nets can be modeled as transition systems.

2.1 Transition Systems

Definition 1 (Transition Systems). *[11] A* Transition System $\Psi : \langle V, \mathcal{T}, \Theta \rangle$ *has the following components:*

- V*: a set of* state variables*. In the rest of the paper we assume* $V = \{x_1, \ldots, x_n\}$ *unless otherwise stated. A state s is an interpretation of* V*.*
- $\mathcal{T}$ *a set of* transitions*; each transition* $\tau \in \mathcal{T}$ *is defined by a transition relation* $\rho_\tau(V, V')$*, a first-order formula in which the unprimed variables refer to the values in the current state and the primed variables refer to the values in the next state;*
- Θ*: an assertion over* V *that represents the* initial condition*. The assertion* Θ *is assumed to be satisfiable.*

Definition 2 (Run). *A sequence of states $s_0, s_1, \ldots$ is a* run *of a transition system $\Psi : \langle V, \mathcal{T}, \Theta\rangle$ if (1) the initial state satisfies the initial condition, that is, $s_0 \models \Theta$, and (2) for each pair of consecutive states s_i, s_{i+1} there exists a transition $\tau \in \mathcal{T}$ that leads from s_i to s_{i+1}, that is, $(s_i, s_{i+1}) \models \rho_\tau$*

2.2 Linear Transition Systems

A *linear inequality* is a constraint $a_1x_1 + \ldots + a_nx_n + b \leq 0$ where $a_1, \ldots, a_n, b,$ are real-valued coefficients and $x_1, \ldots, x_n,$ are real-valued variables. A *linear assertion* is a conjunction of linear inequalities. Given an assertion ψ, the assertion ψ' is obtained by replacing all the variables with their primed counterparts.

A *linear transition system* is a transition system in which all variables in V are real-valued and all transition relations and the initial condition are linear assertions.

For a transition τ in a linear transition system the transition relation ρ_τ can be written as

$$\begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n + a'_{11}x'_1 + \cdots + a'_{1n}x'_n + b_1 \leq 0 \\ \vdots \qquad\qquad \vdots \qquad\quad \vdots \qquad \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n + a'_{m1}x'_1 + \cdots + a'_{mn}x'_n + b_m \leq 0 \end{bmatrix}$$

2.3 Petri Nets

Definition 3 (Petri nets and Markings). *[13] A* Petri net structure *is a tuple $\langle P, T, I, O\rangle$, where P denotes a set of places, T denotes a set of transitions, $I : T \mapsto P^\infty$ is a multi-set of* input places *for each transition and $O : T \mapsto P^\infty$ represents the multi-set of output places for each transition. A* marking *μ is an assignment of a non-negative number of* marks *to each place in the* Petri net structure*. A* Petri net *consists of a Petri net structure and an initial marking μ_0.*

Informally, a Petri net starts from its initial marking and changes state by firing transitions. A transition τ can fire whenever each input place p has a given minimum number of tokens. The effect of firing a transition is to remove the tokens from the input places and add them to the output places.

Definition 4 (Run). *Let $n_p(I(t))$ denote the multiplicity of p in the set of input places for transition t, and similar for the output places. A sequence of markings, $\gamma_0, \gamma_1, \ldots$ is a* run *of a Petri net if (1) the first marking is equal to the initial marking, that is, $\gamma_0 = \mu_0$, and (2) for each pair of consecutive markings, γ_i, γ_{i+1}, there exists a transition $t \in T$ such that t is enabled, that is for each place $p \in P$, $\gamma_i(p) \geq n_p(I(t))$, and the transition leads to γ_{i+1}, that is, for each place $p \in P$ $\gamma_{i+1}(p) = (\gamma_i(p) - n_p(I(t))) + n_p(O(t))$*

Definition 5 (Petri nets as Transition Systems). *Given a Petri net $\mathcal{P}$: $\langle P, T, I, O, \mu_0\rangle$, the transition system $\Psi : \langle V, \mathcal{T}, \Theta\rangle$ is called the associated transition system of $\mathcal{P}$ if*

– *for each $p \in P$ there exists a variable x_p in V;*
– *for each $t \in T$ there exists a transition $\tau \in \mathcal{T}$ with transition relation*

$$\bigwedge_{p \in P} x_p \geq n_p(I(t)) \;\wedge\; x'_p = (x_p - n_p(I(t))) + n_p(O(t))$$

– *$\Theta = \bigwedge_{p \in P}(x_p = \mu_0(p))$. The initial marking may be parametric, that is, a place or a set of places is initialized to contain an unknown number of tokens, taken as a parameter. For each parameter v we introduce a new variable x_v and add the conjunct $x'_v = x_v$ to the transition relation of each transition.*

An assignment of integer values to $\boldsymbol{x}$ corresponds naturally to a marking and vice-versa. Therefore we can refer to the reachable markings of a transition system without any ambiguity. Thus, the conversion preserves reachable markings.

Theorem 1 (Safety of Conversion). *Let Ψ be the associated transition system of Petri Net $\mathcal{P}$. Then a marking μ is reachable in $\mathcal{P}$ iff the corresponding variable assignment is reachable in Ψ.*

With each transition t of a Petri net $\mathcal{P}$, we associate a *guard vector* $\boldsymbol{g_\tau} = \langle g_1, \ldots, g_n \rangle$, and an *update vector* $\boldsymbol{u_\tau} = \langle u_1, \ldots, u_n \rangle$, where $g_i = n_i(I(t))$ is the minimum number of tokens required to be present in place i for the transition to fire, and $u_i = n_i(O(t)) - n_i(I(t))$ is the change in the number of tokens in place i when the transition fires.

From definition 5 it is easy to see that the corresponding transition in the associated transition system can be written in terms of $\boldsymbol{g_\tau}$ and $\boldsymbol{u_\tau}$ as follows:

$$\begin{bmatrix}
-x_1 & & & & & & & + g_1 \leq 0 \\
& -x_2 & & & & & & + g_2 \leq 0 \\
& & \ddots & & & & & \vdots \quad \vdots \\
& & & -x_n & & & & + g_n \leq 0 \\
& & & & & & & \\
-x_1 & & & & +x'_1 & & & - u_1 = 0 \\
& -x_2 & & & & +x'_2 & & - u_2 = 0 \\
& & \ddots & & & & \ddots & \vdots \quad \vdots \\
& & & -x_n & & & & +x_n - u_n = 0
\end{bmatrix}$$

Example 1. Figure 1 shows a Petri net with three places p_1, p_2, p_3 and two transitions t_1, t_2. The input function is shown by arrows from places to transitions, where the label in the arrow indicates the multiplicity of the input place for that transition, with a default value of 1. For example $I(t_1) = \{p_1 : 1,\ p_2 : 2,\ p_3 : 2\}$. Similarly, the output function is shown as arrows from the transitions to the places. The initial marking (not shown in the figure) has one token in p_1 and two tokens in $p_2,\ p_3$.

The associated transition system of this Petri net is as follows:

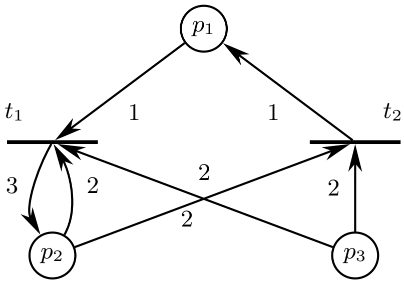

Fig. 1. Example Petri net with three locations and two transitions

$$
\begin{gathered}
V = \{x_1, x_2, x_3\} \\
\mathcal{T} = \{\tau_1, \tau_2\} \\
\Theta \equiv (x_1 = 1 \ \wedge \ x_2 = 2 \ \wedge \ x_3 = 2) \\
\rho_{\tau_1} \equiv \begin{bmatrix} x_1 \geq 1 \\ x_2 \geq 2 \\ x_3 \geq 2 \\ x_1' - x_1 = -1 \\ x_2' - x_2 = 1 \\ x_3' - x_3 = -2 \end{bmatrix} \quad \rho_{\tau_2} \equiv \begin{bmatrix} x_1 \geq 0 \\ x_2 \geq 2 \\ x_3 \geq 2 \\ x_1' - x_1 = 1 \\ x_2' - x_2 = -2 \\ x_3' - x_3 = -2 \end{bmatrix}
\end{gathered}
$$

The $\boldsymbol{u}$ and $\boldsymbol{g}$ vectors for the two transitions are:

$$
\begin{gathered}
\boldsymbol{g_{\tau_1}} : \langle 1, 2, 2\rangle \ \ \boldsymbol{u_{\tau_1}} : \langle -1, 1, -2\rangle \\
\boldsymbol{g_{\tau_2}} : \langle 0, 2, 2\rangle \ \ \boldsymbol{u_{\tau_2}} : \langle 1, -2, -2\rangle
\end{gathered}
$$

2.4 Linear Constraints

As mentioned earlier, a linear assertion is a conjunction of constraints of the form $a_1x_1 + \ldots + a_nx_n + b \leq 0$. A linear inequality is said to be *homogeneous* if $b = 0$ or *inhomogeneous* otherwise. Geometrically the set of points in $\mathfrak{R}^n$ satisfying a homogeneous assertion is a polyhedral cone. Inhomogeneous constraints correspond to polyhedra. Any polyhedron can be represented in terms of its constraints or in terms of its generators (vertices, lines and rays). The generators of a polyhedron (polyhedral cone) can be viewed as the set of *basic solutions* to the constraints that define the polyhedron (cone). Every other solution to the constraints lies in the convex-hull (conic-hull) formed by these basic solutions. For linear constraints operations like satisfiability, projection, intersection, convex union and computing generators can be carried out efficiently in theory and practice. The details are available in any textbook or survey on this topic [14, 3].

3 Invariant Generation

We will give a brief description of our method for invariant generation. A more detailed presentation of our invariant generation technique may be found in our earlier work [5].

Definition 6 (Invariant). *Given a transition system Ψ, an assertion ϕ is ' an* invariant *for Ψ if it holds at each reachable state of Ψ.*

Definition 7 (Inductive). *An assertion ϕ is* inductive *for a transition system Ψ : $\langle V, \mathcal{T}, \Theta \rangle$ iff (1. Initiation) ϕ holds initially, that is, $\Theta \models \phi$, and (2. Consecution) if ϕ holds prior to a transition being taken then it holds in any state obtained after the transition is taken, that is, for all $\tau \in \mathcal{T}$, $\phi \wedge \rho_\tau \models \phi'$.*

It is easy to show that any inductive assertion is also an invariant assertion.

Traditional methods for invariant generation compute a super-set of the set of reachable states by some form of symbolic forward propagation with widening until a fixed point is reached. Our method, on the other hand, computes invariants directly by encoding the two conditions for inductiveness, Initiation and Consecution, as a set of constraints on the coefficients of the target invariant. Any solution of this set of constraints corresponds to an inductive invariant.

The technique is based on Farkas' Lemma [14], which provides a sound and complete method for reasoning with systems of linear inequalities.

Theorem 2 (Farkas' Lemma). *Consider the following system of linear inequalities over real-valued variables $x_1, \ldots, x_n$,*

$$S: \begin{bmatrix} a_{11}x_1 + \cdots + & a_{1n}x_n + & b_1 \le 0 \\ \vdots & \vdots & \vdots \\ a_{m1}x_1 + \cdots + & a_{mn}x_n + & b_m \le 0 \end{bmatrix}$$

When S is satisfiable, it entails a given linear inequality

$$\psi: \; c_1x_1 + \cdots + c_nx_n + d \le 0$$

if and only if there exist non-negative real numbers $\lambda_0, \lambda_1, \ldots, \lambda_m$, such that

$$c_1 = \sum_{i=1}^{m} \lambda_i a_{i1}, \quad \ldots \quad , c_n = \sum_{i=1}^{m} \lambda_i a_{in}, \; d = (\sum_{i=1}^{m} \lambda_i b_i) - \lambda_0$$

Furthermore, S is unsatisfiable if and only if the inequality $1 \le 0$ can be derived as shown above.

In the rest of this paper we will represent applications of this lemma using the following tabular notation:

$$
\begin{array}{r|rcrcrl}
\lambda_0 & & & & & -1 \leq 0 & \\
\lambda_1 & a_{11}x_1 + \cdots + & & a_{1n}x_n + & & b_1 \leq 0 & \multirow{4}{*}{$\left.\begin{array}{c}\\ \\ \\ \end{array}\right\} S$} \\
\vdots & \vdots & & \vdots & & \vdots & \\
\lambda_m & a_{m1}x_1 + \cdots + & & a_{mn}x_n + & & b_m \leq 0 & \\
\hline
 & c_1x_1 + \cdots + & & c_nx_n + & & d \leq 0 & \leftarrow \psi \\
 & & & & & 1 \leq 0 & \leftarrow \mathit{false}
\end{array}
$$

The antecedents are placed above the line and the consequences below. For each column, the sum of the column entries above the line, with the appropriate multipliers, must be equal to the entry below the line. If a row corresponds to an inequality, the corresponding multiplier is required to be non-negative. This requirement is dropped for rows corresponding to equalities.

Farkas' Lemma can be used to generate inductive assertions for transition systems. The key idea is to consider the co-efficients $c_1, \ldots, c_n, d$ of a target invariant

$$\varphi: \; c_1x_1 + \cdots + c_nx_n + d \leq 0$$

as unknowns [1], and obtain constraints corresponding to initiation and consecution, using Farkas' Lemma.

Initiation: The constraint for initiation $\Theta \models \varphi$ is:

$$
\begin{array}{r|rrrl}
\lambda_0 & & & -\; 1 \leq 0 & \\
\lambda_1 & a_{11}x_1 + \cdots + & a_{1n}x_n + & b_1 \leq 0 & \\
\vdots & \vdots & \vdots & \vdots & \Big\} \Theta \\
\lambda_m & a_{m1}x_1 + \cdots + & a_{mn}x_n + & b_m \leq 0 & \\
\hline
 & c_1x_1 + \cdots + & c_nx_n + & d \leq 0 & \leftarrow \varphi
\end{array}
$$

where $\lambda_0, \ldots, \lambda_m \geq 0$. Note that we omitted the $1 \leq 0$ case, because the initial condition Θ is assumed to be satisfiable. The set of solutions for c_i (expressed as a set of constraints on c_i) are obtained by eliminating the (existentially quantified) multipliers $\lambda_0, \ldots, \lambda_m$.

Example 2. Consider again the transition system in Example 1. The initiation constraint for $\Theta : (x_1 = 1 \wedge x_2 = 2 \wedge x_3 = 2)$ is represented by the following table:

$$
\begin{array}{r|ccr}
\lambda_0 & & & -\; 1 \leq 0 \\
\lambda_1 & x_1 & & -\; 1 = 0 \\
\lambda_2 & & x_2 & -\; 2 = 0 \\
\lambda_3 & & & x_3 - 2 = 0 \\
\hline
 & \multicolumn{3}{c}{c_1x_1 + c_2x_2 + c_3x_3 + d \leq 0}
\end{array}
$$

yielding the following constraints on the coefficients c_1, c_2, c_3, and d:

$$
\begin{array}{l}
c_1 = \lambda_1, \; c_2 = \lambda_2, \; c_3 = \lambda_3, \; \lambda_0 \geq 0 \\
-\lambda_0 - \lambda_1 - 2\lambda_2 - 2\lambda_3 = d
\end{array}
$$

[1] Henceforth, as a convention, a, b with subscripts denote known real coefficients and c, d with subscripts denote unknown real coefficients

On elimination of $\boldsymbol{\lambda}$, we get $\psi_\Theta : \; c_1 + 2c_2 + 2c_3 + d \le 0$.

Consecution For each transition τ, the consecution condition, $\varphi \wedge \rho_\tau \models \varphi'$, is encoded as follows:

$$
\begin{array}{r|rcrcrl}
\mu & c_1x_1 + \cdots + c_nx_n & & & + & d \le 0 & \leftarrow \varphi \\
\lambda_0 & & & & - & 1 \le 0 & \\
\lambda_1 & a_{11}x_1 + \cdots + a_{1n}x_n & + & a'_{11}x'_1 + \cdots + a'_{1n}x'_n & + & b_1 \le 0 & \\
\vdots & \vdots \qquad\quad \vdots & & \vdots \qquad\quad \vdots & & \vdots & \Big\} \; \rho_\pi \\
\lambda_m & a_{m1}x_1 + \cdots + a_{mn}x_n & + & a'_{m1}x'_1 + \cdots + a'_{mn}x'_n & + & b_m \le 0 & \\
\hline
 & & & c_1x'_1 + \cdots + c_nx'_n & + & d \le 0, & \leftarrow \varphi' \\
 & & & & & 1 \le 0 & \leftarrow \textit{disabled}
\end{array}
$$

where $\mu, \lambda_0, \ldots, \lambda_m \ge 0$. In this case we must include the row $1 \le 0$ to account for transitions that are never enabled, that is, transitions for which the conjunction of the invariant and the transition relation is unsatisfiable.

The set of solutions for c_i (expressed as a set of constraints on c_i) are obtained by eliminating the (existentially quantified) multipliers $\lambda_0, \ldots, \lambda_m, \; \mu$.

Theorem 3. *The invariant generation technique is sound and complete for inductive linear inequalities.*

Proof: This follows directly from Farkas' Lemma and our constraint generation technique.

The constraints for Consecution (the enabled case) are nonlinear (quadratic to be precise), because the multiplier μ is multiplied with the, unknown, coefficients c_i. As was noted in our earlier work [5], this limits the practicality of our method, because of the limitations of nonlinear constraint solving tools. In the rest of the paper we show that for general Petri nets these nonlinear constraints can be avoided by exploiting the structure of the transitions in the associated transition system, thereby making this method practical for this class of systems.

4 Analysis of Petri Nets

We now specialize the method presented in the previous section to transition systems that are derived from Petri nets as described in Section 2. Since the constraints derived from the initial condition and the disabled case of consecution are always linear, our primary focus here is on the enabled case of consecution.

4.1 Deriving Constraints

Using the transition representation from Section 2, the constraints for the consecution condition can be written as

$$
\begin{array}{l|ccccccccll}
\mu & \multicolumn{4}{l}{c_1x_1 + c_2x_2 + \ldots + c_nx_n} & & & & & + d & \leq 0 \\
\lambda_0 & & & & & & & & & - 1 & \leq 0 \\
\lambda_{g,1} & - \; x_1 & & & & & & & & + g_1 & \leq 0 \\
\lambda_{g,2} & & - \; x_2 & & & & & & & + g_2 & \leq 0 \\
\ldots & & & \ddots & & & & & & \vdots & \\
\lambda_{g,n} & & & & - \; x_n & & & & & + g_n & \leq 0 \\
\lambda_{u,1} & - \; x_1 & & & & + \; x'_1 & & & & - u_1 & = 0 \\
\lambda_{u,2} & & - \; x_2 & & & & + \; x'_2 & & & - u_2 & = 0 \\
\ldots & & & \ddots & & & & \ddots & & \vdots & \\
\lambda_{u,n} & & & & - \; x_n & & & & + \; x'_n & - u_n & = 0 \\
\hline
 & & & & & \multicolumn{5}{r}{c_1x'_1 + c_2x'_2 + \ldots + c_nx'_n + d} & \leq 0
\end{array}
$$

Recall that for each column the sum of the column entries above the line, with the appropriate multiplier, must be equal to the entry below the line. Thus we have

$$\lambda_{u,1} = c_1, \;\; \lambda_{u,2} = c_2, \;\; \ldots \;\; , \lambda_{u_n} = c_n$$

that is,

$$\boldsymbol{\lambda_u} \; = \; \boldsymbol{c}$$

Furthermore

$$\mu c_1 - \lambda_{g,1} - \lambda_{u,1} = 0, \;\; \ldots \;\; , \;\; \mu c_n - \lambda_{g,n} - \lambda_{g,n} = 0$$

or

$$\mu \boldsymbol{c} - \boldsymbol{\lambda_g} - \boldsymbol{\lambda_u} = 0$$

which, with $\boldsymbol{\lambda_u} = \boldsymbol{c}$ gives

$$(\mu - 1)\boldsymbol{c} \; = \; \boldsymbol{\lambda_g}$$

and finally, for the constant column,

$$\mu d - \lambda_0 + \lambda_{g,1} g_1 + \ldots \lambda_{g,n} g_n - \lambda_{u,1} u_1 - \ldots - \lambda_{u,n} u_n = d$$

which can be rewritten as

$$(\mu - 1)d + \boldsymbol{g}^{\mathrm{T}} \boldsymbol{\lambda_g} - \boldsymbol{u}^{\mathrm{T}} \boldsymbol{\lambda_u} = \lambda_0$$

Thus the coefficients $\boldsymbol{c}$ of inequalities that satisfy the enabled case of the consecution condition are characterized by

$$\exists\mu \geq 0 \; \exists\lambda_0 \geq 0 \; \exists\boldsymbol{\lambda_g} \geq 0 \; \exists\boldsymbol{\lambda_u} \left[\begin{array}{c} \boldsymbol{\lambda_u} = \boldsymbol{c} \\ \wedge \\ (\mu-1)\boldsymbol{c} = \boldsymbol{\lambda_g} \\ \wedge \\ (\mu-1)d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{\lambda_g} - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{\lambda_u} = \lambda_0 \end{array} \right]$$

Elimination of λ_0, $\boldsymbol{\lambda_g}$, and $\boldsymbol{\lambda_u}$ results in

$$\exists\mu \geq 0 \left[\begin{array}{c} (\mu-1)\boldsymbol{c} \geq 0 \\ \wedge \\ (\mu-1)(d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c}) - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \geq 0 \end{array} \right]$$

To eliminate μ we consider four cases separately: $\mu = 1$, $0 < \mu < 1$, $\mu = 0$, and $\mu > 1$. Before we do so, we first introduce an auxiliary lemma that allows simplification of the latter two cases.

Lemma 1 (Auxiliary). *For $\alpha, x, y \in \Re$:*

$$\exists\alpha > 0 \; (\alpha x + y \geq 0) \;\equiv\; (x > 0 \;\vee\; y > 0 \;\vee\; (x = 0 \wedge y = 0))$$

Proof. ($\Rightarrow$) Assume $(\alpha x + y \geq 0)$ holds for some $\alpha > 0$. Then it cannot be the case that both x and y are strictly negative. Hence either $x > 0$ or $y > 0$ or both are zero.
($\Leftarrow$) Assume $(x > 0 \;\vee\; y > 0 \;\vee\; (x = 0 \wedge y = 0))$. If $x > 0$ then choose $\alpha > |y/x|$. Alternatively, if $y > 0$ and $x < 0$, choose $\alpha < |y/x|$. Both cases result in $(\alpha x + y) \geq 0$. The remaining cases, that is $y > 0, x = 0$, and $y = 0, x = 0$ are obvious.

Case $\mu = 1$: For $\mu = 1$ the constraints simplify to

$$\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \leq 0$$

Case $\mu > 1$: For $\mu > 1$, taking $\alpha = \mu - 1$, the constraints can be rewritten as

$$\exists\alpha > 0 \left[\begin{array}{c} \alpha\boldsymbol{c} \geq 0 \\ \wedge \\ \alpha(d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c}) - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \geq 0 \end{array} \right]$$

Applying lemma 1 this simplifies to the disjunction

$$\boldsymbol{c} \geq 0 \;\wedge\; \left(\begin{array}{c} d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} > 0 \\ \vee \\ \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} < 0 \\ \vee \\ d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} = 0 \wedge \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} = 0 \end{array} \right)$$

The second and third disjunct are subsumed by the condition $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \leq 0$ from the first case, leaving only

$$\boldsymbol{c} \geq 0 \;\wedge\; d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} > 0$$

Case $\mu = 0$: For $\mu = 0$ the constraints simplify to

$$\boldsymbol{c} \leq 0 \;\wedge\; d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \leq 0$$

Case $0 < \mu < 1$: For $0 < \mu < 1$ the constraints can be rewritten as

$$\boldsymbol{c} \leq 0 \;\wedge\; -(d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c}) - \frac{1}{1-\mu}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \geq 0$$

or, taking $\alpha = \frac{\mu}{1-\mu}$,

$$\boldsymbol{c} \leq 0 \;\wedge\; -(d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c}) - \alpha\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \geq 0$$

Applying Lemma 1 results in

$$\boldsymbol{c} \leq 0 \;\wedge\; \begin{pmatrix} d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} < 0 \\ \vee \\ \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} < 0 \\ \vee \\ d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} = 0 \;\wedge\; \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} = 0 \end{pmatrix}$$

Since the first and third disjunct are subsumed by the case $\mu = 0$ and the second disjunct is subsumed by the case $\mu = 1$, this case does not add any new constraints.

Thus, for the overall constraints for consecution of a transition τ we only need to consider the cases $\mu = 0$, $\mu = 1$, and $\mu > 1$ yielding the constraint

$$\psi_\tau :\; \underbrace{(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \leq 0)}_{\psi_{dec}} \vee \underbrace{(\boldsymbol{c} \geq 0 \;\wedge\; d + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c} > 0)}_{\psi_{dis}} \vee \underbrace{(\boldsymbol{c} \leq 0 \;\wedge\; (\boldsymbol{g}^{\mathrm{T}} + \boldsymbol{u}^{\mathrm{T}})\boldsymbol{c} + d \leq 0)}_{\psi_{loc}}$$

Each of the disjuncts represents a specific relationship between the transition and the invariant as described below.

4.2 Interpretation of the Constraints

Recall that the target invariant is $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{c} + d \leq 0$ and that Consecution requires that the invariant be preserved by all transitions $\tau \in \mathcal{T}$, that is,

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{c} + d \leq 0 \;\wedge\; \rho_\tau \;\rightarrow\; \boldsymbol{x'}^{\mathrm{T}}\boldsymbol{c} + d \leq 0$$

with

$$\rho_\tau :\; \boldsymbol{x} \geq \boldsymbol{g} \;\wedge\; \boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{u}$$

The disjunct ψ_{dec} states that the effect of the update by the transition on the invariant is to decrease the value of the invariant:

$$\boldsymbol{x'}^{\mathrm{T}}\boldsymbol{c} = (\boldsymbol{x}^{\mathrm{T}} + \boldsymbol{u}^{\mathrm{T}})\boldsymbol{c} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{c} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{c}$$

Clearly, if the invariant holds before the transition is taken, that is $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{c}+d \le 0$, then, with $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{c} \le 0$, also $\boldsymbol{x'}^{\mathrm{T}}\boldsymbol{c}+d \le 0$, and hence the invariant is preserved.

The disjunct ψ_{dis} states that the transition is always disabled. To see this, consider an arbitrary state $\boldsymbol{x}$ such that $\boldsymbol{x}$ satisfies the invariant, that is

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{c}+d \le 0$$

such that $\boldsymbol{c}, d$ satisfy ψ_{dis}. For a transition to be enabled at $\boldsymbol{x}$ we need $\boldsymbol{x} \ge \boldsymbol{g}$, or equivalently, $\boldsymbol{x} = \boldsymbol{g}+\boldsymbol{m}$, for some $\boldsymbol{m} \ge 0$. But

$$(\boldsymbol{g}+\boldsymbol{m})^{\mathrm{T}}\boldsymbol{c}+d \le 0$$

and, with $\boldsymbol{m} \ge 0$ and $\boldsymbol{c} \ge 0$, we have $\boldsymbol{m}^{\mathrm{T}}\boldsymbol{c} \ge 0$, and hence,

$$\boldsymbol{g}^{\mathrm{T}}\boldsymbol{c}+d \le 0$$

contradicting the condition $\psi_{dis}: \ \boldsymbol{c} \ge 0 \ \wedge \ \boldsymbol{g}^{\mathrm{T}}\boldsymbol{c}+d > 0$. Hence any invariant satisfying ψ_{dis} states that the transition is disabled.

Finally, the disjunct ψ_{loc} states that the transition establishes the invariant by itself, independent of its originating state. Invariants established by transitions directly are also called *local invariants*. Again consider an arbitrary state $\boldsymbol{x}$. Again the transition is enabled on $\boldsymbol{x}$ if $\boldsymbol{x} = \boldsymbol{g}+\boldsymbol{m}$ for some $\boldsymbol{m} \ge 0$, and in that case for the next state the following holds:

$$\boldsymbol{x}' = \boldsymbol{g}+\boldsymbol{m}+\boldsymbol{u}$$

It follows that

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}'+d \quad = \quad \boldsymbol{c}^{\mathrm{T}}(\boldsymbol{g}+\boldsymbol{m}+\boldsymbol{u})+d \quad = \quad \boldsymbol{c}^{\mathrm{T}}(\boldsymbol{g}+\boldsymbol{u})+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{m}+d$$

Since $\boldsymbol{c}, d$ satisfy ψ_{loc}, we have $\boldsymbol{c} \le 0$ and $\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{g}+\boldsymbol{u})+d \le 0$. Since, $\boldsymbol{m} \ge 0$, we have $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{m} \le 0$. Applying this, we obtain

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}'+d \quad = \quad \boldsymbol{c}^{\mathrm{T}}(\boldsymbol{g}+\boldsymbol{u})+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{m}+d \quad \le \quad \boldsymbol{c}^{\mathrm{T}}(\boldsymbol{g}+\boldsymbol{u})+d \quad \le \quad 0$$

establishing that the invariant is preserved.

It is interesting to note that if the transition is not decreasing with respect to the invariant, that is ψ_{dec} does not hold, then we have either $\boldsymbol{c} \ge 0$ and the transition is disabled or $\boldsymbol{c} \le 0$ and the transition establishes the invariant itself. Figures 2 and 3 provide an intuitive explanation for this seemingly strong constraint on $\boldsymbol{c}$. For each transition, the guard assertion is of the form $\boldsymbol{x} \ge \boldsymbol{g}$, where each entry in $\boldsymbol{g}$ is non-negative. Thus the guard is a *rectangular* set of points in the *positive orthant*. Whenever the transition is taken from a marking represented by $\boldsymbol{x}$, the change in the marking is always the same, given by $\boldsymbol{u}$. We claim that any invariant increasing with respect to some transition cannot intersect the guard region of the transition. Assume that such an intersection occurs at point $\boldsymbol{p}$ then $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{p}+d = 0$ holds, and upon taking the transition $\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{p}+\boldsymbol{u})+d > 0$ holds, thus violating consecution (see Figure 3). Hence, any such invariant must either *exclude* the transition guard region, thus disabling the transition or *contain* the transition guard.

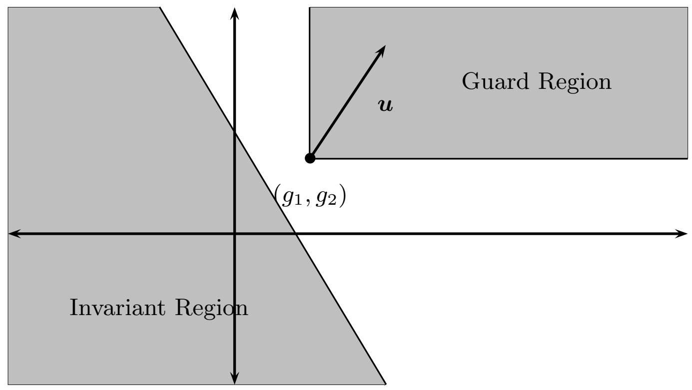

Fig. 2. The disabled case for consecution requires that the guard be excluded.

Example 3. The consecution constraint for transition τ_1 of the running example is $\psi_{\tau_1} = \psi^1_{dec} \vee \psi^1_{dis} \vee \psi^1_{loc}$ with

$$\begin{aligned} \psi^1_{dec} &: -c_1 + c_2 - 2c_3 \leq 0 \\ \psi^1_{dis} &: \boldsymbol{c} \geq 0 \ \wedge \ \left[d + c_1 + 2c_2 + 2c_3 > 0 \right] \\ \psi^1_{loc} &: \boldsymbol{c} \leq 0 \ \wedge \ \left[d + 3c_2 \leq 0 \right] \end{aligned}$$

4.3 Generating Invariants

Given a Petri net as a transition system and a target invariant $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + d \leq 0$ the constraints on $\boldsymbol{c}, d$ are the conjunction of the constraints generated by initiation and those generated for consecution for each transition, that is

$$\psi : \ \psi_{\Theta} \ \wedge \ \bigwedge_{\tau \in \mathcal{T}} \psi_{\tau}$$

The following theorem establishes soundness:

Theorem 4 (Soundness). *Let ψ be the conjunction of the constraints corresponding to the initiation and consecution requirements. For any solution $\langle \boldsymbol{c}, d \rangle = \langle \boldsymbol{a}, b \rangle$ to ψ, we have that $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + b \leq 0$ is an inductive invariant.*

Proof. This follows directly from the soundness of Farkas' Lemma for the initiation case and the derivation of each of the consecution constraints.

Theorem 5 (Completeness). *Let ψ be the conjunction of the constraints for initiation and consecution requirements. If $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + b \leq 0$ is an inductive invariant then $\langle \boldsymbol{c}, d \rangle = \langle \boldsymbol{a}, b \rangle$ is a solution to ψ.*

Proof. This follows from the case analysis for the consecution constraints obtained from Farkas' Lemma.

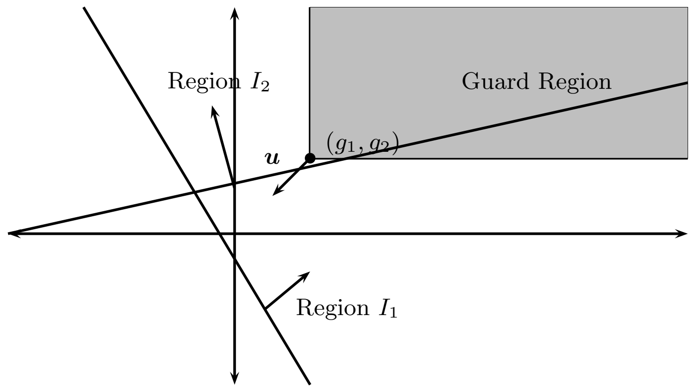

Fig. 3. Inductive invariant $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}+d \leq 0$ cannot intersect the guard. I_1 is a valid invariant but I_2 is not.

Thus to generate the invariants we compute the conjunction $\psi_\Theta \wedge \bigwedge_{\tau\in\mathcal{T}}(\psi_\tau)$ and put the expression in disjunctive normal form (DNF). Note that the disabled case is ignored since it can be shown equivalent to ψ_{dis}. Each clause in the DNF formula obtained is a conjunction of assertions involving the coefficients $\boldsymbol{c}$. This is geometrically a polyhedron. We compute the generators for each clause and recast these as invariants. In particular, the ray $a_1c_1 + \ldots + a_nc_n + bd$ is recast as the invariant $a_1x_1 + \ldots + a_nx_n + b \leq 0$ and the line $a_1c_1 + \ldots + a_nc_n + bd$ corresponds to $a_1x_1 + \ldots + a_nx_n + b = 0$. Vertices are handled similarly to rays. The final invariant is a conjunction of all the invariants obtained from the generators of all the clauses in the DNF formula.

The bottle-neck has been observed to be the computation of the DNF formula. Since each ψ_τ is a disjunction of the constraints corresponding to the three cases, we can compute the DNF form by observing that if ψ_{dis} for a transition τ_1 and ψ_{loc} for a transition τ_2 are present simultaneously in a clause, we immediately have $\boldsymbol{c} = 0$ which corresponds to the invariant $1 \geq 0$, called the *trivial invariant*. Therefore, we avoid taking conjunctions simultaneously involving terms from ψ_{dis} and ψ_{loc}. This is not surprising since we argued that for the disabled case a potential invariant must exclude the guard and for the local case the guard had to be contained. Thus requiring both to hold simultaneously even though for different transitions results in the trivial invariant. This observation saves an exponential factor in practice.

Another issue involved in the computation of the DNF result is the traversal. The traversal may be breadth-first, wherein all the clauses are simultaneously computed. Or else, we may compute each clause, one at a time, choosing one disjunctive term from each ψ_τ . While the former is time efficient, it has been observed to increase the size of the result to an extent where its handling becomes costly. The latter is space efficient while being costly in terms of time.

We advocate the breadth-first traversal for small systems and the depth-first traversal for larger systems.

The time complexity of the analysis is exponential in the number of transitions. However, the method seems to scale to medium sized programs in practice. The space complexity is polynomial in the size of the Petri net.

Example 4. Continuing with the running example, the initiation constraints are given by

$$\psi_\Theta \;\equiv\; (d + c_1 + 2c_2 + 2c_3 \leq 0)$$

For transition τ_1 the constraint ψ_{τ_1} is given by

$$\begin{aligned}
\psi^1_{dec} &: \; -c_1 + c_2 - 2c_3 \leq 0 \;\vee \\
\psi^1_{dis} &: \; \boldsymbol{c} \geq 0 \;\wedge\; \big[d + c_1 + 2c_2 + 2c_3 > 0\big] \;\vee \\
\psi^1_{loc} &: \; \boldsymbol{c} \leq 0 \;\wedge\; \big[d + 3c_2 \leq 0\big]
\end{aligned}$$

For transition τ_2 the constraint ψ_{τ_2} is given by

$$\begin{aligned}
\psi^2_{dec} &: \; c_1 - 2c_2 - 2c_3 \leq 0 \;\vee \\
\psi^2_{dis} &: \; \boldsymbol{c} \geq 0 \;\wedge\; \big[d + 2c_2 + 2c_3 > 0\big] \;\vee \\
\psi^2_{loc} &: \; \boldsymbol{c} \leq 0 \;\wedge\; \big[d + c_1 \leq 0\big]
\end{aligned}$$

The conjunction $\psi_\Theta \;\wedge\; \psi_{\tau_1} \;\wedge\; \psi_{\tau_2}$ when set in DNF form yields the following non-trivial clauses:

$$\begin{pmatrix} -c_1 + c_2 - 2c_3 \leq 0 \;\wedge \\ c_1 - 2c_2 - 2c_3 \leq 0 \;\wedge \\ d + c_1 + 2c_2 + 2c_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} c_1, c_2, c_3 \leq 0 \;\wedge \\ -c_1 + c_2 - 2c_3 \leq 0 \;\wedge \\ d + c_1 \leq 0 \;\wedge \\ d + c_1 + 2c_2 + 2c_3 \leq 0 \end{pmatrix} \vee \begin{pmatrix} c_1, c_2, c_3 \leq 0 \;\wedge \\ d + 3c_2 \leq 0 \;\wedge \\ c_1 - 2c_2 - 2c_3 \leq 0 \;\wedge \\ d + c_1 + 2c_2 + 2c_3 \leq 0 \end{pmatrix}$$

The first clause yields the following generators and the corresponding invariants:

$$\begin{array}{rcr}
\textit{line}\ \langle 6, 4, -1, -12\rangle & \rightarrow & 6x_1 + 4x_2 - x_3 - 12 = 0 \\
\textit{ray}\ \langle 0, -1, 1, 0\rangle & \rightarrow & -x_2 + x_3 \leq 0 \\
\textit{ray}\ \langle -2, 0, 1, 0\rangle & \rightarrow & -2x_1 + x_3 \leq 0
\end{array}$$

Similarly the remaining clauses can be handled and the invariants obtained are

$$\begin{aligned}
&6x_1 - x_3 + 4x_2 - 12 = 0 \\
&3x_1 + 2x_2 - 6 \geq 0 \\
&2x_1 + x_2 - 4 \leq 0 \\
&x_1 + x_2 - 3 \leq 0
\end{aligned}$$

This is a triangle with vertices $\langle 2, 0, 0\rangle$, $\langle 1, 2, 2\rangle$, $\langle 0, 3, 0\rangle$, which are the three reachable states of the system.

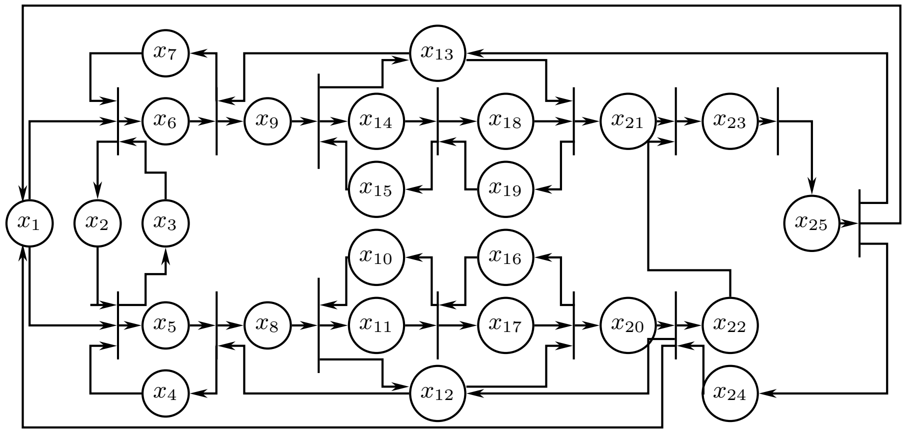

Fig. 4. Example Petri net for a manufacturing system

5 Application

We demonstrate the power of our approach by applying it to a manufacturing system first presented in [16] and later analyzed in [4, 8, 1]. The Petri net, shown in Figure 4, models an automated manufacturing system with four machines, M_1 - M_4 whose availability is modeled by x_5, x_6, x_{17}, and x_{18}, respectively, two robots, R_1 and R_2, whose availability is modeled by x_{12} and x_{13}, and two buffers, modeled by x_{10} and x_{15}. Raw material is introduced in place x_1, whose initial marking is parametric, indicating that it may initially contain any positive number of tokens. The raw material passes through two assembly lines, where it is processed by the machines and transported by the robots, and ends up in the delivery area, modeled by x_{25}. The initial marking of the system is

$$\begin{aligned}&x_1 = p\\&x_2 = x_4 = x_7 = x_{12} = x_{13} = x_{16} = x_{19} = x_{24} = 1\\&x_{10} = x_{15} = 3\end{aligned}$$

and all other places contain zero tokens.

Using a prototype implementation based on the Polyhedral Library POLKA [9] invariants were generated for this system, where we only considered the disjuncts ϕ_{dec} and ϕ_{loc} and disregarded ϕ_{dis} to limit the number of invariants generated. This resulted in around 1900 invariants, all but 20 of which were of the form

$$x_{i_1} + x_{i_2} + \cdots + x_{i_k} \geq 1$$

which correspond to initially marked traps [15]. The remaining invariants included the following structural invariants

$$\begin{aligned}
\psi_1 &: x_2 + x_3 = 1 \\
\psi_2 &: x_4 + x_5 = 1 \\
\psi_3 &: x_6 + x_7 = 1 \\
\psi_4 &: x_8 + x_{12} + x_{20} = 1 \\
\psi_5 &: x_9 + x_{13} + x_{21} + x_{23} + x_{24} = 1 \\
\psi_6 &: x_{10} + x_{11} = 3 \\
\psi_7 &: x_{14} + x_{15} = 3 \\
\psi_8 &: x_{16} + x_{17} = 1 \\
\psi_9 &: x_{18} + x_{19} = 1 \\
\psi_{10} &: x_{22} + x_{23} + x_{24} + x_{25} \leq 1
\end{aligned}$$

and the invariants

$$\begin{aligned}
\chi_1 &: x_1 \leq p \\
\chi_2 &: x_1 \geq p - 12
\end{aligned}$$

which together provide a better insight in the structure of the system. For example, from ψ_4 and ψ_5 it can be seen that the responsibilities of the two robots R_1 (x_{12}) and R_2 (x_{13}) are not symmetric. While R_1 is used to transport material from M_1 to M_3 and from M_3 to the packaging area, robot R_2 has, in addition to the corresponding tasks on the other assembly line, also the responsibility to deliver the combined product from the two assembly lines to the output area (x_{25}). From the invariants it can also be inferred that the system is bounded. The invariant generation took around 4 minutes.

The invariants generated were also used to prove deadlock freedom for this system for $1 \leq p \leq 8$ by proving that the conjunction of the invariants implies the disjunction of enabling conditions of all transitions, thus ensuring that for every reachable state at least one transition is enabled. For $p = 9$ we were able to isolate four possible deadlock states. Note however, that the presence of deadlock cannot be verified directly over abstractions in general.

Deadlock freedom had been proven before. In [16] the developers of this manufacturing system show that the system is live for $2 \leq p \leq 4$. In [4] it is shown that the system is deadlock free for $1 \leq p \leq 8$ using a mixed integer programming approach; it is also shown that the system is not deadlock free for $p > 8$ by exhibiting a transition sequence that leads to deadlock. In [1] a transformational approach is used to compute the invariants through HYTECH [10]. The analysis is done by an exact reach set computation using Presburger Arithmetic in a previous work by Fribourg and Olsen [8]. This analysis reported a running time of nearly 2 hours whereas, in [1] the running time is improved to under 2 minutes.

6 Strengthening Scheme

Having computed the invariants at a previous stage, we wish to use them as strengthening assertions to obtain new and potentially stronger invariants. Note that our approach is complete only for linear inequalities that are inductive invariants by themselves. This, however, does not preclude the existence of linear invariants that need other previously established invariants as strengthening assertions. For a general presentation of strengthening assertions, we refer the reader to a a standard text on the topic [11].

Consider the case when certain transitions are shown to be disabled by the generated invariants. In such a case, removing these transitions, and recomputing the invariants will necessarily yield stronger (if not strictly stronger) invariants. This is because the invariants formed by the constraints from the disabled case, were shown to preclude those formed by the constraints from the local case. The removal of disabled transitions may be argued to be a rudimentary form of strengthening. In its general form, strengthening the invariants generated requires using the previously computed invariants as additional guard assertions to each transition.

Example 5. We augment the running example 1 with a new transition τ_3 with transition relation

$$\rho_{\tau_3} : \; x_1 \geq 3 \;\wedge\; x_1' = x_1 + 3 \;\wedge\; x_2' = x_2 + 2 \;\wedge\; x_3' = x_3 + 3 \;\; .$$

It can be seen by generating the reachability tree that τ_3 is never taken. The invariants generated on the first run are

$$2x_1 + x_2 \leq 4, \;\; x_1 + x_2 \leq 3, \;\; x_3 \geq 0, \;\; 6x_1 + 4x_2 - x_3 \geq 12$$

which are strictly weaker than the invariants generated before for the same system without τ_3. For example, the conjunction of these invariants admits the (unreachable) state $\langle 1, 2, 0 \rangle$ which the previous invariants did not include. However these invariants naturally imply that $x_1 \leq 2$, which allows us to remove the disabled transition, giving us back the original invariants from example 4.

There are two approaches to the problem of strengthening invariants. The first approach uses the three cases and their interpretations in terms of Petri net behavior using the machinery already in place. The second restates the problem by diluting the strong assumptions placed on the structure of the transfer matrices and recomputing the closed form solution. Unfortunately, we have not found a convenient closed form solution for the latter approach, and hence we resort to the former approach.

Assume that φ_0 is a previously computed set of inductive assertions that are added to the guard of a Petri net transition τ. Furthermore, we assume that transitions whose guards have been shown to be disabled with respect to φ_0 are removed from the transition system. A re-interpretation of the three cases for consecution, taking the assertion φ_0 into account, leads to

- For the decreasing case, the change in the value of the expression $c_1x_1 + \cdots + c_nx_n + d$ remains the same, regardless of the strengthening. Thus, the final constraint for this case remains

$$\psi_{dec} : \; \boldsymbol{c}^{\mathrm{T}} \boldsymbol{u} \leq 0$$

– For the disabled case we encode the incompatibility of the guard and the invariant using Farkas' lemma,

$$(\varphi_0 \ \wedge \ \boldsymbol{x} \geq \boldsymbol{g} \ \wedge \ c_1 x_1 + \cdots + c_n x_n + d \leq 0) \models (1 \leq 0)$$

– For the local case we use again Farkas' Lemma, now to encode that the transition guard must imply that the value of the invariant expression after the transition being taken must be negative. This yields,

$$\varphi_0 \ \wedge \ (\boldsymbol{x} \geq \boldsymbol{g}) \ \models (c_1 x_1 + \cdots + c_n x_n + \boldsymbol{c}^{\mathrm{T}} \boldsymbol{g} + d \leq 0)$$

All three cases result in linear constraints in $\boldsymbol{c}$, $\boldsymbol{\lambda}$. After eliminating the multipliers, the resulting constraints are linear in the coefficients $\boldsymbol{c}$. Note that if φ_0 is the trivial invariant *true*, the constraints obtained are the same as those derived in the previous sections. Although we do not have a proof of completeness of these constraints, we conjecture that completeness can be shown by direct reasoning on the structure of Petri net transitions.

7 Conclusion

We have presented a general invariant generation technique for Petri nets using Farkas' Lemma to generate invariants on the unknown coefficients of an invariant to guarantee initiation and consecution conditions. Note that we have not restricted the formation of Petri nets with parametric initial markings and those with *inhibitor arcs*. Inhibitor arcs can also be incorporated into the framework of the transition system by adding a constraint $x_p = 0$ to a transition inhibited by place p. The invariant generation technique can handle these arcs with a few changes. The closed-form solution of these constraints was derived for the special case of Petri nets leading to an efficient invariant generation for general Petri nets that compares favourably with more exact and expensive techniques over the application examples presented. It is computationally inexpensive when compared to the other general analysis techniques.

The main drawbacks of the technique stem from the relative weakness of the invariant domain, which can lead to inexact results. However, we can use the geometric intuition behind the invariants to generate strengthenings that can alleviate this problem to some extent. It is as yet unclear if the technique can be extended to other types of Petri nets like colored and timed Petri nets. The technique as such does not exploit restricted classes of Petri nets that have been analyzed rigorously by the Petri net community [15, 12]. We believe that the analysis of general Petri nets and nets with inhibitors can be made much more tractable by combining many of these time-saving observations along with the use of more efficient linear constraint solving techniques.

References

1. B. Bérard and L. Fribourg. Reachability analysis of (timed) petri nets using real arithmetic. In *Proc. Intl. Conf. Concurrency Theory (CONCUR'99)*, volume 1664 of *LNCS*, 1999.

2. F. Besson, T. Jensen, and J.-P. Talpin. Polyhedral analysis of synchronous languages. In *Static Analysis Symposium, SAS'99*, Lecture Notes in Computer Science 1694, pages 51–69, 1999.
3. A. Bockmayr and V. Weispfenning. Solving numerical constraints. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 12, pages 751–842. Elsevier Science, 2001.
4. F. Chu and X.-L. Xie. Deadlock analysis of petri nets using siphons and mathematical programming. *IEEE Transactions on Robotics and Automation*, 13(6):793–804, December 1997.
5. M. Colón, S. Sankaranarayanan, and H. Sipma. Linear invariant generation using non-linear constraint solving. In *Computer Aided Verification (CAV)*, Lecture Notes in Computer Science 2725, pages 420–432, 2003.
6. P. Cousot and R. Cousot. Abstract Interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *ACM Principles of Programming Languages*, pages 238–252, 1977.
7. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among the variables of a program. In *ACM Principles of Programming Languages*, pages 84–97, Jan. 1978.
8. L. Fribourg and H. Olsén. Proving safety properties of infinite state systems by compilation into presburger arithmetic. In *Proceedings of CONCUR'97, LNCS*, volume 1243, pages 213–227, Germany, Berlin, 1997.
9. N. Halbwachs and Y.-E. Proy. *POLyhedra desK cAlculator (POLKA)*. VERIMAG, Montbonnot, France, Sept. 1995.
10. T. A. Henzinger and P. Ho. HyTech: The Cornell hybrid technology tool. In *Hybrid Systems II*, volume 999 of *LNCS*, pages 265–293, 1995.
11. Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, New York, 1995.
12. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, Apr. 1989.
13. J. Peterson. *Petri Net Theory and the Modelling of Systems*. Prentice Hall, 1983.
14. A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.
15. M. Silva, E. Teruel, and J. M. Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. *Lectures on Petri Nets I: Basic Models, LNCS*, 1491:309–373, 1998.
16. M. Zhou, F. DiCesare, and A. A. Desrochers. A hybrid methodology for synthesis of petri net models for manufacturing systems. *IEEE Transactions on Robotics and Automation*, 8(3):350–361, June 1992.