Reset Nets Between Decidability and Undecidability

C. Dufourd, A. Finkel, and Ph. Schnoebelen

Lab. Specification and Verification,* ENS de Cachan & CNRS URA 2236.

Abstract. We study Petri nets with Reset arcs (also Transfer and Doubling arcs) in combination with other extensions of the basic Petri net model. While Reachability is undecidable in all these extensions (indeed they are Turing-powerful), we exhibit unexpected frontiers for the decidability of Termination, Coverability, Boundedness and place-Boundedness. In particular, we show counter-intuitive separations between seemingly related problems. Our main theorem is the very surprising fact that boundedness is undecidable for Petri nets with Reset arcs.

1 Introduction

> *"In general, it seems that any extension which does not allow zero testing will not actually increase the modeling power (or decrease the decision power) of Petri nets but merely result in another equivalent formulation of the basic Petri net model. (Modeling convenience may be increased.)"* [Pet81], page 203.

Extensions of Petri nets. The above quote from [Pet81] is a fair summary of current beliefs in the Petri net community regarding extensions of the basic Petri net model: extensions are either Turing-powerful or they are not real extensions. It explains why there exist very few studies of decidability issues for small extensions of Petri nets (with the notable exception of Valk's Post-SM nets) compared to the hundreds of papers investigating subclasses of Petri nets (free-choice nets, conflict-free nets, 1-safe nets, ...).

Reset arcs. Reset arcs from a transition t to a place p are a new kind of arcs used to reset p (i.e. to empty it) whenever t fires. Their modeling convenience has been investigated e.g. in [Bil91,LC94].

There are some obvious connections between "reseting" and "testing for emptiness" (see the proof of Theorem 11). It is widely known that Petri nets with inhibitory (or "zero test") arcs are Turing-powerful. By contrast, the study of decidability issues for Petri nets with Reset arcs only started in [AK77] where the Reachability problem is shown undecidable. Then, language-theoretical properties and extensions of p-semi flows techniques were studied in [Cia94] for Reset

* 61, av. Pdt Wilson; 94235 Cachan Cedex; FRANCE. email: {dufourd, finkel, phs}@lsv.ens-cachan.fr.

(and other) arcs. Recently, [KCK+97] announced that the Boundedness problem is decidable for Petri nets extended with Reset arcs (actually for their more general PCN model).

It turns out that Reset arcs push Petri nets closer towards the frontiers of decidability. The scarcity of results in this domain is partly explained by the difficulty of the remaining open questions. But the study of such borderline models is an important topic. Indeed there exists an annual conference, "Verification of Infinite State Systems", devoted to the algorithmic aspects of decision methods for such models.

Our contribution. In this paper, we study decidability issues for Reset arcs (and related extensions) in a general framework, aiming at a better understanding of the situation. We introduce G-nets, a general framework containing in a natural way all the extensions we are interested in, and where we can smoothly isolate relevant subclasses. We study in a systematic way the decidability of the Coverability, the Termination, the Reachability, and the two Boundedness problems for G-nets and several relevant subclasses. Our three most important results are:

- The decidability of the Coverability problem for a very large extension of Petri nets, using a surprisingly simple new algorithm.
- The undecidability of the Boundedness problem for Petri nets with Reset arcs, a deep result countering our earlier intuitions (and the decidability proof from [KCK+97]).
- The proof that the Coverability, Boundedness and place-Boundedness problems can be separated in counter-intuitive ways.

Related Works. Valk introduced and studied Self-Modifying nets (that contain Reset Petri nets) and Post Self-Modifying nets [Val78b,Val78a]. He showed that SM-nets can simulate two-counters machines with inhibitor arcs, and then that almost all properties (like Reachability, Boundedness or Termination) are undecidable. He also proved that Reachability is undecidable for Post SM-nets and that the place-Boundedness problem is decidable (with the non-primitive-recursive Karp and Miller algorithm) for Post SM-nets. Lakos and Christensen [LC94] compared different sets of primitives for extended arcs, essentially from a modelization point of view. Billington modelized the Cambridge Ring Network using Reset arcs [Bil91]. See the bibliographies in [Cia94,LC94] for other *applications* of Petri nets with Reset and Transfer arcs. Because we are not concerned with true concurrency issues, we see Read arcs [Vog97] as classical arcs.

Plan of the paper. We define G-nets and relevant subclasses in Sections 2 and 3. Then we study Coverability and Termination (Section 4), Boundedness (Section 5), place-Boundedness (Section 6), and Reachability (Section 7) in turn.

2 Generalized Self-Modifying Nets

Let $P = \{p_1, \ldots, p_k\}$. We write $\mathbb{N}[P]$ or $\mathbb{N}[p_1, \cdots, p_k]$ the set of polynomials over k variables with coefficients from $\mathbb{N}$. We adopt the usual convention $P \cup \mathbb{N} \subseteq \mathbb{N}[P]$.

All the Petri nets extensions studied in this paper will be subclasses of *Generalized Self-Modifying nets*, a very general new class of extended nets. A Generalized Self-Modifying net (a G-net for short), with k places $p_1, \ldots, p_k$, is a net where each arc is labeled by a polynomial Q from $\mathbb{N}[p_1, p_2, \ldots, p_k]$ of the special form $\Sigma_{j \in J} \lambda_j p_{i_j}^{n_j}$ (J finite, λ_j, $n_j \in \mathbb{N}$ and $1 \leq i_j \leq k$). Generalized Self-Modifying nets naturally extend Self-Modifying nets defined twenty years ago by Valk [Val78b,Val78a] where only polynomials of the form $\Sigma_{i=1}^{k} \lambda_i p_i$ are considered.

Why we use this notion of G-nets is mostly a matter of clarity and convenience. It has convenience because computing with simple polynomials is easy. It has clarity because we wanted to show that our approach is quite general and smoothly go beyond the simple linear functions used in Self-Modifying nets.

Definition 1. A Generalized Self-Modifying net (shortly a G-net) is a 4-tuple $N = \langle P, T, F, m_0 \rangle$ where
- $P = \{p_1, \cdots, p_{|P|}\}$ is a finite set of *places*,
- T is a finite set of *transitions* (with $P \cap T = \emptyset$),
- $F : (P \times T) \cup (T \times P) \longrightarrow \mathbb{N}[P]$ is a *flow function* such that : $\forall x, y \in P \cup T$, $F(x, y)$ has the form $\Sigma_{j \in J} \lambda_j p_{i_j}^{n_j}$ where J is a finite set, $\lambda_j \in \mathbb{N}$, $n_j \in \mathbb{N}$ and $1 \leq i_j \leq |P|$,
- $m_0 \in \mathbb{N}^{|P|}$ is the *initial marking*.

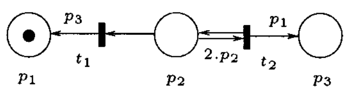

Fig. 1. A G-net computing Fibonacci numbers

Figure 1 shows a G-net computing the Fibonacci numbers. We follow the usual convention that arcs with omitted labels have a weight of 1, a constant polynomial. We use vector notation to denote markings: in the example $m_0 = (1, 0, 0)$. Given a marking m, we write $m(p)$ to denote the number of tokens in place p. We write $m \leq m'$ when $m(p) \leq m'(p)$ for all p.

A transition t is *firable* from a marking $m \in \mathbb{N}^{|P|}$, written $m \xrightarrow{t}$, if for any place p_i :

$$m(p_i) \geq \Sigma_j \lambda_j m(p_{i_j})^{n_j} \text{ where } F(p_i, t) = \Sigma_j \lambda_j p_{i_j}^{n_j}$$

In the example we have $F(p_2, t_2) = 2p_2$ so that $m \xrightarrow{t_2}$ only if $m(p_2) \geq 2m(p_2)$, i.e. iff p_2 is empty in m. This shows how inhibitory arcs are a special case of our extended arcs.

When $m \xrightarrow{t}$, firing t from m leads to a new marking m' where for any place p_i, $m'(p_i) = m(p_i) - \Sigma_j \lambda_j m(p_{i_j})^{n_j} + \Sigma_j \mu_j m(p_{i_j})^{n_j}$ where $F(p_i, t) = \Sigma_j \lambda_j p_{i_j}^{n_j}$ and $F(t, p_i) = \Sigma_j \mu_j p_{i_j}^{n_j}$.

In our example, t_2 is firable from $(1,0,0)$. Firing it leads to add 1 token into p_2 and to add the current content of p_1 into p_3. We have $m_0 = (1,0,0) \xrightarrow{t_2} (1,1,1)$. Now t_1 is firable.

An *execution* of N is a sequence of markings $m_0 \xrightarrow{t_{i_1}} m_1 \xrightarrow{t_{i_2}} m_2 \ldots$ successively reachable from m_0. An execution of the Fibonacci net is $m_0 \xrightarrow{t_2} (1,1,1) \xrightarrow{t_1} (2,0,1) \xrightarrow{t_2} (2,1,3) \xrightarrow{t_1} (5,0,3) \ldots$. For odd k, we reach the marking $m_k = (\mathit{fib}(k+1), 0, \mathit{fib}(k))$.

A net *terminates* if there exists no infinite execution (*Termination Problem*). E.g. the Fibonacci net does not terminate. A marking m' is *reachable* from m, written $m \xrightarrow{*} m'$, if there exists a sequence $\sigma \in T^*$ such that $m \xrightarrow{\sigma} m'$ (*Reachability Problem*). The *reachability set* of N, denoted $RS(N)$,is $\{m | m_0 \xrightarrow{*} m\}$. A marking $m \in \mathbb{N}^{|P|}$ is *coverable* in N, if there exists a marking $m' \in RS(N)$ such that $m' \geq m$ (*Coverability Problem*). A G-net is *bounded* if its reachability set is finite (*Boundedness Problem*). A place $p \in P$ is *bounded* if there exists a $k \in \mathbb{N}$ such that $\forall m \in RS(N), m(p) \leq k$ (*place-Boundedness Problem*). In our example, place p_2 is bounded, but places p_1 and p_3 are not.

These five problems are crucial for verification. The finiteness of the behavior (Termination) or the finiteness of the reachability set (Boundedness) are among the first properties of interest. When some places are used to model buffers or files, implementation issues may require to check Boundedness for these places only. The Coverability and the Reachability Problems are key notions for decidability of temporal logics. Coverability is an abstract problem containing Determinism, Quasi-Liveness, Control-State Reachability, ...

3 Some Relevant Families of G-nets

Our Fibonacci example already illustrates several common kinds of arcs. A flow function $F(p,t) = 2p$ is in fact an inhibitory arc from p to t. A flow function $F(p,t) = p$ is a Reset arc: firing t will set p to zero. We usually draw such arcs with a crossed edge from t to p to emphasize the postcondition side of such arcs. See figure 3. A Transfer arc is used to transfer all tokens from p into some p' when t is fired. Because it empties p, there is an obvious connection with Reset arcs. With $F(p,t) = F(t,p') = p$, G-nets allow Transfer arcs.

We can now define formally how well-known families of extended nets are subclasses of G-nets (see also figure 3).

- Valk's Self-Modifying nets (*SM-nets*) are G-nets such that the flow function F uses polynomials of degree at most 1.
- A *Post G-net* is a G-net where only post-arcs are extended arcs: $\forall p \in P$, $\forall t \in T$, $F(p,t) \in \mathbb{N}$.
- Similarly, a *Post-SM* net is a SM-net such that for every $p \in P$, $t \in T$, $F(p,t) \in \mathbb{N}$.

- A *Petri net* is a G-net with only classical arcs: for every x, y, $F(x,y) \in \mathbb{N}$.
- A *Reset Post G-net* is a G-net where pre-arcs are Reset arcs or classical Petri arcs: for all $p \in P$ and $t \in T$, $F(p,t) = p$ or $F(p,t) \in \mathbb{N}$.
- A *Reset Petri net* is a Reset Post G-net such that all post-arcs are classical: $F(t,p) \in \mathbb{N}$.
- A *Transfer Post G-net* is a Reset Post G-net such that whenever there is a reset arc $F(p,t) = p$ then there is a p' with $F(t,p') = p$. (As a whole, the two arcs (p,t) and (t,p') are what we call a Transfer arc.)
- A *Transfer Petri net* is a Transfer Post G-net such that all arcs are classical arcs or Transfert arcs.
- A *Double Petri net* is a Post G-net such that for every $p \in P$, $t \in T$, $F(t,p) = p$ or $F(t,p) \in \mathbb{N}$. When $F(t,p) = p$, (t,p) is called a *Doubling arc*.

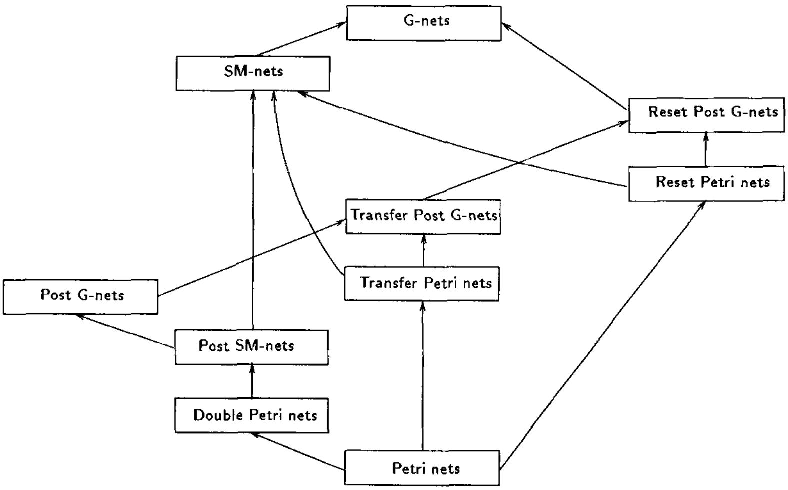

Fig. 2. Subclasses of G-nets and inclusions between them

Self-Modifying nets were defined in 1978 by Valk in [Val78a]. Reset Petri nets were introduced in 1977 by Araki and Kasami in [AK77]. Transfer Petri nets are defined in [Cia94]. Post-SM nets are a subclass of SM-nets defined and studied by Valk in [Val78a]. Petri nets have been defined in 1962 by Petri! The other classes are natural extensions of the previous one.

4 Decidability of Coverability and Termination

The Coverability problem has usually been associated with the coverability tree algorithms [Rac78,KM69,Hac76,Fin90] and then with the Boundedness (and place-Boundedness) problems. Recently, Abdulla *et al.* [AČJY96] and Finkel and Schnoebelen [FS98] have proposed another algorithm for the Coverability problem. This algorithm works on general so-called Well-Structured Transition Systems. It works "backward" (computes predecessors of states), contrasting the earlier "forward" algorithm for coverability trees.

Definition 2. [Fin90,AČJY96] A *well-structured transition system* (a WSTS) is a structure $S = \langle Q, \rightarrow, \leq \rangle$ such that:
- $Q = \{m, \ldots\}$ is a set of states,
- $\rightarrow \subseteq Q \times Q$ is a set of transitions,
- $\leq \subseteq Q \times Q$ is a well-quasi-ordering (a wqo) on the set of states, satisfying the *simple monotonicity* property

$m \rightarrow m'$ and $m_1 \geq m$ imply $m_1 \rightarrow m'_1$ for some $m'_1 \geq m'$ (1)

Thus a WSTS is a transition system where the transitions have the monotonicity property w.r.t. some wqo. (Recall that a wqo is any reflexive and transitive relation such that for any infinite sequence $m_1, m_2, \ldots$, there exists two indexes $i < j$ s.t. $m_i \leq m_j$.)

Theorem 3. [AČJY96,FS98] *For WSTS's with an effective wqo and effective pred-basis, Coverability is decidable.*

(This result is called "decidability of control-state reachability" in [AČJY96].) Here we give the ideas of the algorithm.

Pred-basis is related to one-step coverability: to any $m \in Q$, we associate $pb(m)$, a finite set $\{m_1, \ldots, m_k\}$ such that it is possible to cover m from m' in one step iff m' covers some $m_i \in pb(m)$. Formally,

$$(\exists m_i \in pb(m), m' \geq m_i) \text{ iff } (\exists m'' \geq m, m' \rightarrow m'')$$

The set pb is well-defined because $\{m' \mid \exists m'' \geq m, m' \rightarrow m''\}$ is upward-closed (a consequence of monotonicity) and upward-closed sets have finite basis (a property of well-quasi-orderings). When pb is effective, it is possible to build a sequence $K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots$ of finite sets with $K_0 \stackrel{\text{def}}{=} \{m\}$, $K_{j+1} \stackrel{\text{def}}{=} K_j \cup pb(K_j)$. Because $\leq$ is a well-quasi-ordering, the sequence eventually stabilizes, i.e. there exists an index n s.t. $\forall m_i \in K_{n+1}, \exists m_j \in K_n, m_j \leq m_i$. Because $\leq$ is effective, stabilization can be detected effectively, hence n can be computed. At stabilization, K_n answers the coverability problem: it is possible to cover m from m' iff m' covers some $m_i \in K_n$.

This algorithm applies to all effective WSTS's, including Petri nets, lossy channels systems, normed BPA processes, ... see [AČJY96,FS98].

Now Reset Post G-nets enjoy simple monotonicity w.r.t. "$\leq$", the usual ordering between markings. Further, $\leq$ is an effective wqo, effectivity of pb is clear (see example below), hence Reset Post G-nets are effective WSTS's. The corollary is

Theorem 4. *Coverability is decidable for Reset Post G-nets.*

Let us illustrate the algorithm with the Reset Petri net from figure 3. Assume we are interested into covering the target marking $m = (1,2,0,0)$. We start with $K_0 = \{(1,2,0,0)\}$. Now let us compute $pb(\{(1,2,0,0)\})$.

With t_1, we can cover $(1,2,0,0)$ in one step if we start from $(1,3,0,0)$ or any larger marking. With t_2 is is impossible to cover m in one step because t_2 resets p_2. With t_3, we can cover m if we start from $(1,1,1,1)$ or above. With t_4, we need to start from $(0,1,1,0)$ or above. Eventually, we end up with $K_1 = \{(1,2,0,0),(1,3,0,0),(1,1,1,1),(0,1,1,0)\}$. For convenience, we remove non-minimal elements, writing $K_1 = \{(1,2,0,0),(0,1,1,0)\}$, before going on with the computation of K_2.

Eventually we reach $K_5 = \{(1,0,0,0),(0,0,1,0)\}$ and notice that $K_6 = K_5$. We have reached stabilization: it is possible to cover m from some m_0 iff m_0 has at least one token in p_1 or p_3. Hence m is coverable in N.

Theorem 5. *Termination is decidable for Reset Post G-nets.*

Proof. Again we can apply a general decidability result (from [Fin90]) for WSTS's with effective $\leq$ and effective one-step successors mapping. □

Theorems 4 and 5 cannot be extended beyond Reset Post G-nets:

Theorem 6. [Val78b,Val78a] *Coverability and Termination are undecidable for SM-nets (and hence for G-nets).*

Proof. Strictly speaking, undecidability of Coverability and Termination for SM-nets is not considered in [Val78b,Val78a] but his encoding of Minsky's counter machines into SM-nets can be reused with no difficulty. □

Theorem 4 generalizes a result of Valk [Val78b,Val78a] who decides Coverability for Post SM-nets by using the Karp and Miller coverability tree algorithm on Post SM-nets. His proof cannot be extended because *there does not exist effective coverability trees* (nor finite coverability sets) for Reset Post G-nets, as a consequence of Theorem 8.

5 Decidability of Boundedness

Transfer Post G-nets are Well-Structured Transition Systems with additional structure. They enjoy *strict monotonicity*:

$$m \to m' \text{ and } m_1 > m \text{ imply } m_1 \to m'_1 \text{ for some } m'_1 > m' \tag{2}$$

(while Reset Petri nets only enjoy simple monotonicity). With strict monotonicity, boundedness is decided by searching for a sequence $m_0 \xrightarrow{*} m_1 \xrightarrow{\sigma} m_2$ with $m_1 < m_2$. Then σ is iterated: $m_1 \xrightarrow{\sigma} m_2 \xrightarrow{\sigma} m_3 \cdots$ yielding $m_1 < m_2 < m_3 < \cdots$ and the net is unbounded. This gives

Theorem 7. *Boundedness is decidable for Transfer Post G-nets.*

Proof. Strict monotonicity makes Transfer Post G-nets 1'-well-structured transition systems in the sense of [Fin90], hence boundedness is decidable. $\square$

Reset Petri nets do not enjoy strict monotonicity, making the situation less comfortable: in a Reset Petri net, when $m_1 \xrightarrow{\sigma} m_2$, with $m_2 > m_1$, σ can be iterated, but $m_{i+1} = m_i$ is possible in the sequence $m_1 \xrightarrow{\sigma} m_2 \xrightarrow{\sigma} m_3 \xrightarrow{\sigma} m_4 \cdots$. [KCK+97] thought they could overcome this difficulty by claiming that a Reset Petri net is unbounded iff there is a $m_0 \xrightarrow{*} m_1 \xrightarrow{\sigma} m_2$ with $m_1 < m_2$ and more precisely with $m_1(p) < m_2(p)$ for some place p that is not reset by any transition in σ (see Theorem 7 from [KCK+97]). They conclude that Boundedness is decidable for Reset Petri nets.

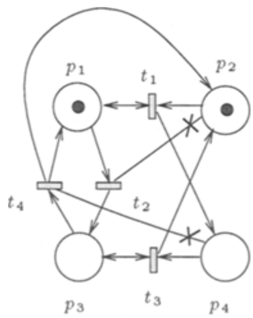

Fig. 3. An unbounded Reset Petri net with no iterated sequence

It turns out their claim is false. Consider the net from figure 3. This net is unbounded but its only unbounded behaviour has the form:

$$(1,1,0,0) \xrightarrow{t_1 t_2 t_3 t_4} (1,2,0,0) \xrightarrow{t_1^2 t_2 t_3^2 t_4} (1,3,0,0) \ldots (1,i,0,0) \xrightarrow{t_1^i t_2 t_3^i t_4} (1,i+1,0,0)\,.$$

and no $m_1 \xrightarrow{\sigma} m_2$ can be found with $m_1 < m_2$ and $m_1(p) < m_2(p)$ for a p that is not reset.
In fact, we cannot extend theorem 7 beyond Transfer Post G-net. Surprisingly

Theorem 8. *Boundedness is undecidable for Reset Petri nets.*

Proof. A full proof of this result can be found in the longer version of this paper [1]. The details take several pages but it is possible to explain the main ideas here.
1. We prove a main lemma, stating that Reset nets can compute polynomials in a weak sense: given Q in $\mathbb{N}[x_1, \ldots, x_n]$, there exists a Reset net N_Q computing Q. Here *computing* Q means that, starting from a vector of $\bar{v} \in \mathbb{N}^n$ of tokens in its input places $p_1^{\text{in}}, \ldots, p_n^{\text{in}}$, N_Q can transfer all tokens in the corresponding output places $p_1^{\text{out}}, \ldots, p_n^{\text{out}}$ by a sequence of exactly $Q(\bar{v})$ visible transitions. (Note that this is different from the more usual notion of gathering $Q(\bar{v})$ tokens in some

[1] available from the authors.

result place.) N_Q does not create new tokens (hence it is bounded), it only moves them around. Now it is not possible to enforce this exact behaviour in a Reset net, and other transitions sequences are possible. However, N_Q is such that (1) *it is not possible to fire more than $Q(\bar{v})$ visible transitions*, and (2) *when N_Q terminates properly after less than $Q(\bar{v})$ transitions, then some tokens have been lost in the transfer from the input to the output places* (this uses Reset arcs).
2. Then we show how to compare two polynomials Q and R. Clearly it is possible to check in a weak sense whether $Q(\bar{v}) = R(\bar{v})$: one synchronizes N_Q and N_R on their visible transitions and feed them with $\bar{v}$ tokens. If Q and R do not agree on $\bar{v}$, then necessarily some tokens will be lost. We use a similar construction checking for non-equality: if the two polynomials agree, then necessarily some tokens will be lost in the transfer.
3. Then we wrap this in an enumeration scheme. We enumerate all vectors $\bar{v}_1, \bar{v}_2, \bar{v}_3, \ldots$ in such a way that (1) *there exists a Reset net outputting (again in a weak sense) $\bar{v}_i$ when given $\bar{v}_{i-1}$*, and (2) *whenever some token is lost from some $\bar{v}_i$, we end up into a $\bar{v}_j$ with $j < i$.* This too uses Reset arcs. When we connect the "check $Q \neq R$" net and the tuple-enumeration net, we end up with a Reset net having the following potential behaviour: check that $Q(\bar{v}_0) \neq R(\bar{v}_0)$, compute $\bar{v}_1$ from $\bar{v}_0$, check that $Q(\bar{v}_1) \neq R(\bar{v}_1)$, compute $\bar{v}_2$ from $\bar{v}_1$, etc. This behaviour is unbounded. Two conditions make it possible: (1) *the net picks the correct behaviour for evaluating polynomials and enumerating tuples*, and (2) *for any $i \in \mathbb{N}$, $Q(\bar{v}_i)$ and $R(\bar{v}_i)$ really differ so that the comparison does not loose tokens.* Any other behaviour is bounded.
4. Thus, given Q and R, we have constructed a Reset net which is bounded iff the diophantine equation $Q(x_1, \ldots, x_n) = R(x_1, \ldots, x_n)$ has no solution. This is an effective reduction of Hilbert's Tenth Problem to our Boundedness Problem for Reset nets. □

Hence Coverability is decidable for Reset Petri nets but Boundedness is not. As far as we know, this is the first published instance of an extension of Petri nets where Coverability and Boundedness are separated from a decidability viewpoint. Because Coverability has always been associated with coverability trees, because Boundedness is decidable for Transfer Petri nets, the natural conjecture had always been that the Boundedness problem must be decidable for Reset nets.

6 Decidability of Place-Boundedness

When it is possible to extend the procedure of Karp and Miller, place-Boundedness is decidable:

Theorem 9. *Place-Boundedness is decidable for Post G-nets.*

Proof. Post G-nets are 1'-well-structured in the sense of [Fin90]. Further, they enjoy the continuity conditions required for theorem 4.16 in [Fin90]. Hence, a (generalized) Coverability tree can be built effectively for Post G-nets. This tree is used to answer place-Boundedness. □

We cannot extend this beyond Post G-nets

Theorem 10. *Place-Boundedness is undecidable for Transfer Petri nets.*

Proof. A Reset Petri net N may be simulated by a Transfer Petri net N' which mimics resets of places by transferring their contents into a (new) dummy place. N is unbounded iff one place different from the dummy place is unbounded in N'. Thus decidability of place-Boundedness for Transfer Petri nets would imply decidability of Boundedness for Reset Petri nets. □

Hence Boundedness is decidable for Transfer Petri net but place-Boundedness is not. As far as we know, this is the first time Boundedness and place-Boundedness are separated from a decidability viewpoint. (What is more, in most papers where place-Boundedness is involved, the name "Boundedness" is used, showing how the two problems have always been seen as one single general problem.)

7 Undecidability of Reachability

Reachability is an important problem. It is decidable for Petri nets and it often becomes undecidable as soon as the power of Petri nets is increased. In this section, we show that for any of the smallest extended classes of Petri nets we defined, *two* extended arcs suffice to make the Reachability Problem undecidable.

Theorem 11. *Reachability is undecidable for Double Petri nets, Reset Petri nets and Transfer Petri nets having two extended (Doubling, Reset or Transfer) arcs.*

Proof. We reduce reachability for nets with inhibitory arcs into reachability for nets with extended (Doubling, Reset or Transfer) arcs. Consider a net N with inhibitory arcs. Any place which is the input place of an inhibitory arc is called an *inhibitory place*. We build a Reset Petri net N^+ by modifying N. In N^+ we add a *twin place* p' for every inhibitory place p. The idea is that p and p' will always have the same number of tokens as long as N^+ correctly simulates N. We extend the flow relation of N^+ by setting $F(p', t) \stackrel{\text{def}}{=} F(p, t)$ and $F(t, p') \stackrel{\text{def}}{=} F(t, p)$ for every t and every inhibitory p. Finally we replace every inhibitory arc from some p to some t by a Reset arc from t to p'. See diagram:

Consider any step $m_1 \rightarrow m_2$ in N^+. The construction ensures that, for any inhibitory p, if $m_1(p) \geq m_1(p')$ then $m_2(p) \geq m_2(p')$. Furthermore, if $m_1(p) > m_1(p')$ then $m_2(p) > m_2(p')$. These two properties are summarized by "N^+ *preserves imbalances*".

Now let m be a marking of N and write m^+ for the marking of N^+ obtained by extending m to twin places: $m^+(p') \stackrel{\text{def}}{=} m(p)$. We claim that $m_0 \xrightarrow{*} m$ in N iff $m_0^+ \xrightarrow{*} m^+$ in N^+. This uses a simple induction over the length of executions. The crucial case in the induction is when N^+ erases for the first time some non empty p' with a Reset arc. This introduces an imbalance that can never be recovered. In N, this step is not possible because p is not empty and inhibits the transition.

The construction also works if we use a Transfer arc from p' to a new dummy place instead of a Reset arc.

The same construction works if we use a doubling arc from t to p' instead of a Reset arc, but this time imbalance means $M(p') > M(p)$. □

It does not seem possible to go beyond Theorem 11 because Reachability is decidable for Petri nets with only one inhibitor arc [Rei95], hence also for Petri nets with at most one Reset arc or one Transfer arc. We conjecture Reachability is decidable when only one Doubling arc is allowed.

Conclusion

In this paper we answered all the decidability questions concerning Coverability, Termination, Reachability, Boundedness and place-Boundedness for all the relevant subclasses of G-nets we gave in figure 3. These results are summarized in figure 4. Let us stress the most important ones:

- A very surprising result is that Boundedness is undecidable even for the very small class of Reset Petri nets. This is the main technical result of the paper. It is highly non-trivial and has been open for several years. That it is counter-intuitive is underlined by the fact that an (erroneous) decidability proof was published recently. Our proof required inventing a new, more faithful, way of weakly-evaluating polynomials with Reset Petri nets. A corollary is that, for Transfer Petri nets, Boundedness is decidable but place-Boundedness is not. Again, this came as a surprise. To the best of our knowledge, this is the first time these two problems are separated.
- It is possible to generalize the Karp and Miller coverability tree algorithm for Post G-nets (and then to decide place-Boundedness), but not for Reset Post G-nets, an extension of Valk's Post SM-nets. Now, for Reset Post G-nets, the Termination problem is decidable using a partial construction of coverability tree; and Coverability is decidable, using a backward algorithm, which computes sets of predecessors of markings, instead of computing sets of successors (as it is done in the coverability tree construction).

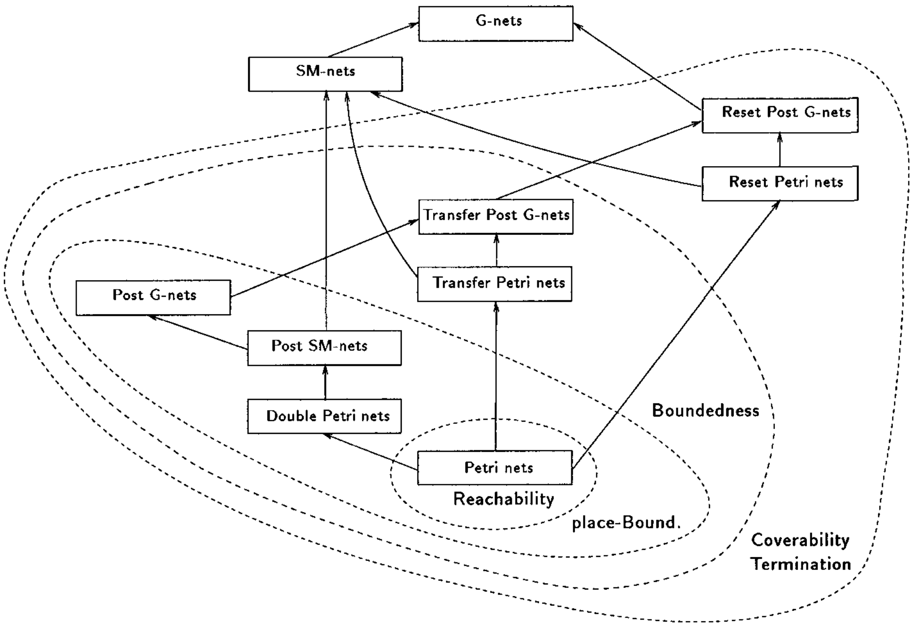

Fig. 4. What's decidable where.

Finally, we may update the opening quote:
There exist extensions of Petri nets which do not allow zero testing but that will actually increase the modeling power (e.g. in term of terminals and covering languages) and decrease the decision power (e.g. Boundedness becomes undecidable). In fact, when one considers a collection of various decision problems (not just Reachability), there are many layers between mere reformulations of the basic Petri net model (at one end), and at the other end Petri nets with inhibitory arcs (i.e. counter machines).

References

[AČJY96] P. A. Abdulla, K. Čerāns, B. Jonsson, and T. Yih-Kuen. General decidability theorems for infinite-state systems. In *Proc. 11th IEEE Symp. Logic in Computer Science (LICS'96), New Brunswick, NJ, USA, July 1996*, pages 313–321, 1996.

[AK77] T. Araki and T. Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3(1):85–104, 1977.

[Bil91] J. Billington. *Extensions to coloured Petri nets and their applications to protocols.* PhD thesis, University of Cambridge, UK, May 1991. Available as Tech. Report No.222.

[Cia94] G. Ciardo. Petri nets with marking-dependent arc cardinality: Properties and analysis. In *Proc. 15th Int. Conf. Applications and Theory of Petri Nets, Zaragoza, Spain, June 1994*, volume 815 of *Lecture Notes in Computer Science*, pages 179–198. Springer-Verlag, 1994.

[Fin90] A. Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.

[FS98] A. Finkel and Ph. Schnoebelen. Fundamental structures in well-structured infinite transition systems. In *Proc. 3rd Latin American Theoretical Informatics Symposium (LATIN'98), Campinas, Brazil, Apr. 1998*, volume 1380 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[Hac76] M. Hack. Decidability questions for Petri nets. PhD Thesis MIT/LCS/TR-161, Massachusetts Institute of Technology, Lab. for Computer Science, June 1976.

[KCK+97] M. Kishinevsky, J. Cortadella, A. Kondratyev, L. Lavagno, A. Taubin, and A. Yakovlev. Coupling asynchrony and interrupts: Place chart nets. In *Proc. 18th Int. Conf. Application and Theory of Petri Nets, Toulouse, France, June 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 328–347. Springer-Verlag, 1997.

[KM69] R. M. Karp and R. E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[LC94] C. Lakos and S. Christensen. A general approach to arc extensions for coloured Petri nets. In *Proc. 15th Int. Conf. Applications and Theory of Petri Nets, Zaragoza, Spain, June 1994*, volume 815 of *Lecture Notes in Computer Science*, pages 338–357. Springer-Verlag, 1994.

[Pet81] J. L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall Int., 1981.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223-231, 1978.

[Rei95] K. Reinhardt. Reachability in Petri nets with inhibitor arcs, November 1995. Unpublished manuscript. See `www-fs.informatik.uni-tuebingen.de/~reinhard`.

[Val78a] R. Valk. On the computational power of extended Petri nets. In *Proc. 7th Symp. Math. Found. Comp. Sci. (MFCS'78), Zakopane, Poland, Sep. 1978*, volume 64 of *Lecture Notes in Computer Science*, pages 526–535, 1978.

[Val78b] R. Valk. Self-modifying nets, a natural extension of Petri nets. In *Proc. 5th Int. Coll. Automata, Languages, and Programming (ICALP'78), Udine, Italy, Jul. 1978*, volume 62 of *Lecture Notes in Computer Science*, pages 464–476. Springer-Verlag, 1978.

[Vog97] W. Vogler. Partial order semantics and read arcs. In *Proc. 22nd Int. Symp. Math. Found. Comp. Sci. (MFCS'97), Bratislava, Slovakia, Aug. 1997*, volume 1295 of *Lecture Notes in Computer Science*, pages 508–517. Springer-Verlag, 1997.