Nets with tokens which carry data

Ranko Lazić[1,⋆], Tom Newcomb[2], Joël Ouaknine[2],
A.W. Roscoe[2], and James Worrell[2]

[1] Department of Computer Science, University of Warwick, UK
[2] Computing Laboratory, University of Oxford, UK

Abstract. We study data nets, a generalisation of Petri nets in which tokens carry data from linearly-ordered infinite domains and in which whole-place operations such as resets and transfers are possible. Data nets subsume several known classes of infinite-state systems, including multiset rewriting systems and polymorphic systems with arrays.
We show that coverability and termination are decidable for arbitrary data nets, and that boundedness is decidable for data nets in which whole-place operations are restricted to transfers. By providing an encoding of lossy channel systems into data nets without whole-place operations, we establish that coverability, termination and boundedness for the latter class have non-primitive recursive complexity. The main result of the paper is that, even for unordered data domains (i.e., with only the equality predicate), each of the three verification problems for data nets without whole-place operations has non-elementary complexity.

1 Introduction

Petri nets (e.g., [1]) are a fundamental model of concurrent systems. Being more expressive than finite-state machines and less than Turing-powerful, Petri nets have an established wide range of applications and a variety of analysis tools (e.g., [2]).

The analysis tools are based on the extensive literature on decidability and complexity of verification problems ([3] is a comprehensive survey). In this paper, we focus on three basic decision problems, to which a number of other verification questions can be reduced:

Coverability: Is a marking reachable which is greater than or equal to a given marking?
Termination: Are all computations finite?
Boundedness: Is the set of all reachable markings finite?

By the results in [4, 5], each of coverability, termination and boundedness is ExpSpace-complete for Petri nets.

Many extensions of Petri nets preserve decidability of various verification problems. Notably, affine well-structured nets were formulated in [6] as an elegant extension of Petri nets by whole-place operations. The latter are resets,

⋆ Supported by the EPSRC (GR/S52759/01) and the Intel Corporation.

which empty a place, and transfers, which take all tokens from a place and put them onto one or more specified places (possibly several times). Hence, two subclasses of affine WSNs are reset nets and transfer nets, in which whole-place operations are restricted to resets and to transfers, respectively. As shown in [6], coverability and termination for affine WSNs, and boundedness for transfer nets, are decidable. However, compared with Petri nets, there is a dramatic increase in complexity: it follows from the results on lossy channel systems in [7] that coverability and termination for reset nets and transfer nets, and boundedness for transfer nets, are not primitive recursive.[1] It was proved in [9] that boundedness for reset nets is undecidable.

Another important direction of extending Petri nets is by allowing tokens to carry data from infinite domains. (Data from finite domains do not increase expressiveness.) For example, in timed Petri nets [10], each token is equipped with a real-valued clock which represents the age of the token. Multiset rewriting specifications over constraint systems $\mathcal{C}$ [11, 12] can be seen as extensions of Petri nets in which tokens may carry data from the domain of $\mathcal{C}$ and transitions can be constrained using $\mathcal{C}$. In mobile synchronizing Petri nets [13], tokens may carry identifiers from an infinite domain, and transitions may require that an identifier be fresh (i.e., not currently carried by any token).

In this paper, we focus on the following two questions:

(1) Is there a general extension of Petri nets in which tokens carry data from infinite domains, in which whole-place operations are possible, and such that coverability, termination and boundedness are decidable (either for the whole class of extended nets or for interesting subclasses)?

(2) If the answer to the previous question is positive, and if we restrict to the subclass without whole-place operations, do coverability, termination and boundedness remain ExpSpace-complete (as for Petri nets), or are their complexities greater? What happens if we restrict further to the simplest data domains, i.e. those with only the equality predicate?

Data nets. To answer question (1), we define data nets, in which tokens carry data from linearly-ordered infinite domains. As in Petri nets, transitions consume and produce tokens. For a transition to be firable, we can require that the data which are carried by the tokens to be consumed are ordered in a certain way. In addition to such data, transitions can choose finitely many other data, which satisfy further ordering constraints and which may or may not be present in the current marking. In the production phase, tokens which carry either kind of data can be put into the marking. Data nets also support whole-place operations.

In the next few paragraphs, we introduce data nets in an informal but detailed manner, for clarity of the subsequent discussion of contributions of the paper and relations to the literature. As an alternative order of presentation, the reader may wish to postpone the following and read it in conjunction with Section 2.2, where data nets are defined formally.

[1] Recall the Ritchie-Cobham property [8, page 297]: a decision problem (i.e. a set) is primitive recursive iff it is solvable in primitive recursive time/space.

Data nets are based on affine WSNs [6]. Markings of an affine WSN are vectors in $\mathbb{N}^P$, where P is the finite set of all places. A transition t of an affine WSN is given by vectors $F_t, H_t \in \mathbb{N}^P$ and a square matrix $G_t \in \mathbb{N}^{P \times P}$. Such a transition is firable from a marking m iff $m \geq F_t$, and in that case it produces the marking $(m - F_t)G_t + H_t$. Whole-place operations are performed by the multiplication with G_t.

Since a linear ordering $\preceq$ is the only operation available on data, markings of data nets are finite sequences of vectors in $\mathbb{N}^P \setminus \{\mathbf{0}\}$. Each index j of such a marking s corresponds to an implicit datum d_j, and we have that $j \leq j'$ iff $d_j \preceq d_{j'}$. For each $p \in P$, $s(j)(p)$ is the number of tokens which carry d_j and are at place p. We say that such tokens are at index j. Now, each transition t has an arity $\alpha_t \in \mathbb{N}$. For a transition t to be fired from a marking s, we choose nondeterministically α_t mutually distinct data. Some of those data may be fresh (i.e., not carried by any token in s), so picking the α_t data is formalised by first expanding s to a finite sequence $s_\dagger$ by inserting the vector $\mathbf{0}$ at arbitrary positions, and then picking an increasing (in particular, injective) mapping

$$\iota : \{1, \dots, \alpha_t\} \to \{1, \dots, |s_\dagger|\}$$

such that each occurrence of $\mathbf{0}$ is in its range. Now, such a mapping ι partitions $\{1, \dots, |s_\dagger|\}$ into α_t singletons and $\alpha_t + 1$ contiguous "regions" as follows, where the $Reg_{(i,i+1)}$ are region identifiers:

$$\underbrace{1, \dots, \iota(1) - 1}_{Reg_{(0,1)}}, \iota(1), \underbrace{\iota(1) + 1, \dots, \iota(2) - 1}_{Reg_{(1,2)}}, \dots, \iota(\alpha_t), \underbrace{\iota(\alpha_t) + 1, \dots, |s_\dagger|}_{Reg_{(\alpha_t, \alpha_t+1)}}$$

The action of t on s with respect to $s_\dagger$ and ι is determined by vectors F_t and H_t, and a square matrix G_t, whose elements are natural numbers, and which are indexed by

$$(\{1, \dots, \alpha_t\} \cup \{Reg_{(i,i+1)} : 0 \leq i \leq \alpha_t\}) \times P$$

It consists of the following stages, where $i, i' \in \{1, \dots, \alpha_t\}$, $R, R' \in \{Reg_{(i,i+1)} : 0 \leq i \leq \alpha_t\}$ and $p, p' \in P$.

subtraction: for each i and p, $F_t(i, p)$ tokens at index $\iota(i)$ are taken from p;[2]

multiplication: all tokens are taken simultaneously, and then:
- for each token taken from p at index $\iota(i)$, $G_t(i, p, i', p')$ tokens are put onto p' at index $\iota(i')$, and for each j' in region R', $G_t(i, p, R', p')$ tokens are put onto p' at index j';
- for each token taken from p at index j in region R, $G_t(R, p, i', p')$ tokens are put onto p' at index $\iota(i')$, and $G_t(R, p, R, p')$ tokens are put onto p' at index j;

addition: for each i and p, $H_t(i, p)$ tokens are put onto p at index $\iota(i)$, and for each j in region R and p, $H_t(R, p)$ tokens are put onto p at index j.

[2] In order to have well-structuredness (see Proposition 7) and for simplicity, entries $F_t(R, p)$ are not used, and neither are entries $G_t(R, p, R', p')$ with $R \neq R'$, so they are assumed to be 0.

Example 1. Consider $P = \{p_1, p_2\}$ and a transition t with $\alpha_t = 1$ given by:

F_t	$Reg_{(0,1)}$		1		$Reg_{(1,2)}$	
	0	0	1	1	0	0
	p_1	p_2	p_1	p_2	p_1	p_2

H_t	$Reg_{(0,1)}$		1		$Reg_{(1,2)}$	
	0	0	2	1	6	0
	p_1	p_2	p_1	p_2	p_1	p_2

G_t	$Reg_{(0,1)}$		1		$Reg_{(1,2)}$		
$Reg_{(0,1)}$	0	1	0	0	0	0	p_1
	1	0	0	0	0	0	p_2
1	0	0	2	0	3	0	p_1
	0	0	0	1	3	0	p_2
$Reg_{(1,2)}$	0	0	0	0	1	0	p_1
	0	0	0	2	0	1	p_2
	p_1	p_2	p_1	p_2	p_1	p_2	

From a marking s, in terms of data represented by the indices of s, transition t is fired as follows:

1. a datum d is chosen nondeterministically, such that each of p_1 and p_2 contain at least 1 token carrying d (so, d cannot be fresh);
2. for each datum $d' \prec d$, all tokens at p_1 carrying d' are transferred to p_2, and vice-versa;
3. for each token at p_1 or p_2 carrying d, and each $d' \succ d$, 3 tokens carrying d' are put onto p_1;
4. the number of tokens at p_1 carrying d is multiplied by 2;
5. for each token at p_2 carrying $d' \succ d$, 2 tokens carrying d are put onto p_2.

Since $H_t = F_t G_t$, the addition stage of performing t exactly "undoes" the subtraction stage, so t performs only whole-place operations.

In Section 2.2, the above will be formalised so that t is firable from s with respect to $s_\dagger$ and ι iff $s_\dagger \geq [\![F_t]\!]_\iota^{|s_\dagger|}$, and in that case it produces the marking obtained from $(s_\dagger - [\![F_t]\!]_\iota^{|s_\dagger|})[\![G_t]\!]_\iota^{|s_\dagger|} + [\![H_t]\!]_\iota^{|s_\dagger|}$ by removing each entry $\mathbf{0}$, where $[\![F_t]\!]_\iota^{|s_\dagger|}$, $[\![G_t]\!]_\iota^{|s_\dagger|}$ and $[\![H_t]\!]_\iota^{|s_\dagger|}$ are appropriate "expansions" of F_t, G_t and H_t, indexed by $\{1, \ldots, |s_\dagger|\} \times P$.

Since vectors $\mathbf{0}$ which correspond to fresh data can be inserted at arbitrary positions to fire a transition, the linear ordering on data is assumed to be dense and without least and greatest elements. Having a least or greatest element can easily be simulated, and density is not a restriction when considering only finite computations (as is the case for the coverability problem).

We show that affine WSNs [6] are equivalent to a class of data nets whose transitions have arity 1. Data nets also subsume timed Petri nets [10] and timed networks [14], in the sense that systems obtained after quotienting by time regions can be simulated by data nets, where the data domain is fractional parts of clock values. Monadic multiset rewriting specifications over order constraints on rationals or reals [11] and over gap-order constrains on integers [12] can be translated to data nets, subject to the remarks above about density. Mobile synchronizing petri nets [13], lossy channel systems [15], and polymorphic systems with one array of type $\langle X, \leq \rangle \to \{1, \ldots, n\}$ or with two arrays of types $\langle X, = \rangle \to \langle Y, \leq \rangle$ and $\langle X, = \rangle \to \{1, \ldots, n\}$ [16, 17], can also be expressed using data nets.

Decidability. Using the theory of well-structured transition systems [18], we prove that coverability and termination for arbitrary data nets, and boundedness for data nets in which whole-place operations are restricted to transfers, are decidable. Thus, question (1) posed above is answered positively. The decidability of coverability for data nets subsumes the results in [6, 10, 14, 11–13, 15–17] that coverability is decidable for the respective classes of infinite-state systems mentioned above, and in most cases the proof in this paper is more succinct.

Hardness. To question (2) above, we obtain the following answers. We say that a data net is *Petri* iff it does not contain whole-place operations, and *unordered* iff it makes use only of equality between data (and not of the linear ordering).

- By providing a translation from lossy channel systems to Petri data nets, we establish that coverability, termination and boundedness for the latter class are not primitive recursive. The encoding uses the linear ordering on the data domain, for picking fresh data which are employed in simulating writes to channels.
- The main result of the paper is that coverability, termination and boundedness for unordered Petri data nets are not elementary, i.e., their computational complexities cannot be bounded by towers of exponentials of fixed heights. That is a surprising result, since unordered Petri data nets are highly constrained systems. In particular, they do not provide a mechanism for ensuring that a datum chosen in a transition is fresh (i.e., not present in the current marking). The result is proved by simulating a hierarchy of bounded counters, which is reminiscent of the "rulers" construction of Meyer and Stockmeyer (e.g., [19]).

By translating Petri data nets and unordered Petri data nets to subclasses of systems in [11–13, 16, 17], the two hardness results yield the same lower bounds for corresponding decision problems for such subclasses. In particular, we obtain non-elementariness of verifying monadic multiset rewriting specifications with only equality constraints [11] and of verifying polymorphic systems with two arrays of types $\langle X, = \rangle \to \langle Y, = \rangle$ and $\langle X, = \rangle \to \{1, \ldots, n\}$ [16].

Paper organisation. Section 2 contains preliminaries, including definitions of data nets and of several relevant subclasses, some basic results, and an example. In Section 3, we present the translation from lossy channel systems to Petri data nets. Sections 4 and 5 contain the decidability and hardness results. Some remaining open problems are discussed in Section 6.

2 Preliminaries

Sets, quasi-orders and mappings. For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. We write $\mathbb{N}_\omega$ for $\mathbb{N} \cup \{\omega\}$. The linear ordering $\leq$ on $\mathbb{N}$ is extended to $\mathbb{N}_\omega$ by having $n < \omega$ for each $n \in \mathbb{N}$.

A set A and a relation $\preceq$ on A form a *quasi-order* iff $\preceq$ is reflexive and transitive. We write $a_1 \prec a_2$ iff $a_1 \preceq a_2$ and $a_2 \not\preceq a_1$.

For any $A' \subseteq A$, its upward closure is $\uparrow A' = \{a \in A : \exists a' \in A' \cdot a' \preceq a\}$. We say that A' is upwards-closed iff $A' = \uparrow A'$. A *basis* of an upwards-closed set A' is a subset A'' such that $A' = \uparrow A''$. Downward closure (written $\downarrow A'$), closedness and bases are defined symmetrically.

A mapping f from a quasi-order $\langle A, \preceq \rangle$ to a quasi-order $\langle A', \preceq' \rangle$ is *increasing* iff $a_1 \prec a_2 \Rightarrow f(a_1) \prec' f(a_2)$.

Vectors and matrices. For sets A and B, let A^B denote the set of all B-indexed vectors of elements of A, i.e., the set of all mappings $B \to A$. For example, $\mathbb{N}^{[n]\times[n']}$ is the set of all $n \times n'$ matrices of natural numbers. For $a \in A$, let $\mathbf{a} \in A^B$ denote the vector whose each entry equals a. Let $\mathit{Id} \in \mathbb{N}^{B\times B}$ denote the identity square matrix.

A quasi-ordering $\preceq$ on A induces the following quasi-ordering on A^B: $v \preceq v'$ iff $v(b) \preceq v'(b)$ for all $b \in B$.

Sequences and bags. For a set A, let $\mathit{Seq}(A)$ denote the set of all finite sequences of elements of A. For $s \in \mathit{Seq}(A)$, let $|s|$ denote the length of s, and $s(1), \ldots, s(|s|)$ denote its elements.

For $s, s' \in \mathit{Seq}(A)$ and $a \in A$, we say that s' is an *a-expansion* of s (equivalently, s is the *a-contraction* of s') iff s is obtained by removing each occurrence of a from s'.

For $s, s' \in \mathit{Seq}(A)$, we write $s \sim s'$ iff s' can be obtained from s by permuting its entries. We define the set $\mathit{Bag}(A)$ of all finite bags (i.e., multisets) of elements of A as the set of all equivalence classes of $\sim$. Let $\overline{s}$ denote the equivalence class of s, i.e., the bag with the same elements as s.

Suppose $\langle A, \preceq \rangle$ is a quasi-order. The quasi-ordering $\preceq$ induces quasi-orderings on $\mathit{Seq}(A)$ and $\mathit{Bag}(A)$ as follows. For $s, s' \in \mathit{Seq}(A)$, we write $s \preceq s'$ iff there exists an increasing $\iota : [|s|] \to [|s'|]$ such that $s(i) \preceq s'(\iota(i))$ for all $i \in [|s|]$. For $b, b' \in \mathit{Bag}(A)$, we write $b \preceq b'$ iff there exist $s \in b$ and $s' \in b'$ such that $s \preceq s'$.

Well-quasi-orderings. A quasi-ordering $\preceq$ on a set A is a well-quasi-ordering iff, for every infinite sequence $a_1, a_2, \ldots \in A$, there exist $i < j$ such that $a_i \preceq a_j$.

Proposition 2 ([20]). *Whenever $\preceq$ is a well-quasi-ordering on a set A, the induced orderings on $\mathit{Seq}(A)$ and $\mathit{Bag}(A)$ also are well-quasi-orderings.*

2.1 Affine well-structured nets

We recall the notion of affine well-structured net [6].[3] Such a net is a tuple $\langle P, T, F, G, H \rangle$ such that P is a finite set of places, T is a finite set of transitions, and for each $t \in T$, F_t and H_t are vectors in $\mathbb{N}^P$, and G_t is a matrix in $\mathbb{N}^{P\times P}$.

[3] For technical reasons, the formalisation of affine WSNs in this paper is slightly different, but equivalent.

Markings of an affine WSN $\langle P, T, F, G, H\rangle$ are vectors in $\mathbb{N}^P$. A marking m' can be obtained from a marking m by firing a transition $t \in T$, written $m \xrightarrow{t} m'$, iff $m \geq F_t$ and $m' = (m - F_t)G_t + H_t$.

As was shown in [6], Petri nets and many of their known extensions are special cases of affine WSNs. In particular, Petri nets and their extensions by (generalised) resets and transfers are equivalent to the classes of affine WSNs $\langle P, T, F, G, H\rangle$ determined by the following restrictions:

Petri nets: $\forall t \in T \cdot G_t = Id$
reset nets: $\forall t \in T \cdot G_t \leq Id$
transfer nets: $\forall t \in T, p \in P \cdot \exists p' \in P \cdot G_t(p, p') > 0$

2.2 Data nets

Given $n \in \mathbb{N}$, let $Regs(n) = \{Reg_{(i,i+1)} : 0 \leq i \leq n\}$. For each $0 \leq i \leq n$, $m \geq n$ and increasing $\iota : [n] \to [m]$, let $[\![Reg_{(i,i+1)}]\!]^m_\iota = \{j \in [m] : \iota(i) < j < \iota(i+1)\}$, where by convention $\iota(0) = 0$ and $\iota(n+1) = m+1$.

A *data net* is a tuple $\langle P, T, \alpha, F, G, H\rangle$ such that:

- P is a finite set of places;
- T is a finite set of transitions;
- for each $t \in T$, $\alpha_t \in \mathbb{N}$ specifies the arity of t;
- for each $t \in T$, $F_t \in \mathbb{N}^{([\alpha_t] \cup Regs(\alpha_t)) \times P}$, and $F_t(R, p) = 0$ whenever $R \in Regs(\alpha_t)$ and $p \in P$;
- for each $t \in T$, $G_t \in \mathbb{N}^{(([\alpha_t] \cup Regs(\alpha_t)) \times P)^2}$, and $G_t(R, p, R', p') = 0$ whenever $R, R' \in Regs(\alpha_t)$, $R \neq R'$ and $p, p' \in P$;
- for each $t \in T$, $H_t \in \mathbb{N}^{([\alpha_t] \cup Regs(\alpha_t)) \times P}$.

Suppose $\langle P, T, \alpha, F, G, H\rangle$ is a data net, and $t \in T$. Any $m \geq \alpha_t$ and increasing $\iota : [\alpha_t] \to [m]$ determine the following instances of F_t, G_t and H_t:

- $[\![F_t]\!]^m_\iota \in \mathbb{N}^{[m] \times P}$ is defined by

$$[\![F_t]\!]^m_\iota(\iota(i), p) = F_t(i, p) \qquad [\![F_t]\!]^m_\iota(j, p) = F_t(R, p) \text{ for } j \in [\![R]\!]^m_\iota$$

- $[\![G_t]\!]^m_\iota \in \mathbb{N}^{([m] \times P)^2}$ is defined by

$$\begin{array}{rll} [\![G_t]\!]^m_\iota(\iota(i), p, \iota(i'), p') &= G_t(i, p, i', p') & \\ [\![G_t]\!]^m_\iota(\iota(i), p, j', p') &= G_t(i, p, R, p') & \text{for } j' \in [\![R]\!]^m_\iota \\ [\![G_t]\!]^m_\iota(j, p, \iota(i'), p') &= G_t(R, p, i', p') & \text{for } j \in [\![R]\!]^m_\iota \\ [\![G_t]\!]^m_\iota(j, p, j, p') &= G_t(R, p, R, p') & \text{for } j \in [\![R]\!]^m_\iota \\ [\![G_t]\!]^m_\iota(j, p, j', p') &= 0 & \text{otherwise} \end{array}$$

- $[\![H_t]\!]^m_\iota \in \mathbb{N}^{[m] \times P}$ is defined in the same way as $[\![F_t]\!]^m_\iota$.

A *marking* of a data net $\langle P, T, \alpha, F, G, H\rangle$ is a finite sequence of vectors in $\mathbb{N}^P \setminus \{\mathbf{0}\}$. A marking s' can be obtained from a marking s by firing a transition $t \in T$, written $s \xrightarrow{t} s'$, iff there exist a $\mathbf{0}$-expansion $s_\dagger$ of s and an increasing $\iota : [\alpha_t] \to [|s_\dagger|]$ such that:[4]

[4] In (ii) and (iii), $s_\dagger$ is treated as a vector in $\mathbb{N}^{[|s_\dagger|] \times P}$.

(i) $\{j : s_\dagger(j) = \mathbf{0}\} \subseteq \mathit{Range}(\iota)$;
(ii) $s_\dagger \geq [\![F_t]\!]_\iota^{|s_\dagger|}$;
(iii) s' is the $\mathbf{0}$-contraction of $(s_\dagger - [\![F_t]\!]_\iota^{|s_\dagger|})[\![G_t]\!]_\iota^{|s_\dagger|} + [\![H_t]\!]_\iota^{|s_\dagger|}$.

We may also write $s \xrightarrow{t,s_\dagger,\iota} s'$, or just $s \to s'$.

Proposition 3. *For any data net, its transition system $\langle \mathit{Seq}(\mathbb{N}^P \setminus \{\mathbf{0}\}), \to\rangle$ is finitely branching.*

2.3 Decision problems

We consider the following standard problems:

Coverability: Given a data net, and markings s and s', to decide whether some marking $s'' \geq s'$ is reachable from s.
Termination: Given a data net, and a marking s, to decide whether all computations from s are finite.
Boundedness: Given a data net, and a marking s, to decide whether the set of all markings reachable from s is finite.

Coverability, termination and boundedness for affine WSNs are defined in the same way.

2.4 Classes of data nets

We now define several classes of data nets. Figure 1 shows the inclusions among classes of data nets and affine well-structured nets in Propositions 5, 6, 8 and 9 below. In addition, the mapping $\mathcal{N} \mapsto \widetilde{\mathcal{N}}$ and its inverse (see Proposition 6) provide a correspondence between unary transfer data nets (resp., unary Petri data nets) and transfer nets (resp., Petri nets). The dashed line represents the fact that Proposition 9 does not provide a reduction for the boundedness problem.

Unordered data nets. A data net $\langle P, T, \alpha, F, G, H\rangle$ is unordered iff:

(i) for each $t \in T$, $R, R' \in \mathit{Regs}(\alpha_t)$ and $p, p' \in P$, we have $G_t(R, p, R, p') = G_t(R', p, R', p')$ and $H_t(R, p) = H_t(R', p)$;
(ii) for each $t \in T$ and permutation π of $[\alpha_t]$, there exists $t' \in T$ such that $F_{t'}$, $G_{t'}$ and $H_{t'}$ are obtained from F_t, G_t and H_t (respectively) by applying π to each index in $[\alpha_t]$.

Given an unordered data net $\langle P, T, \alpha, F, G, H\rangle$, we write $t \sim t'$ iff t and t' have the property in (ii) above. That defines an equivalence relation on T, and we write $\overline{t}$ for the equivalence class of t. From the following proposition, the same-bag relation $\sim$ between markings is a bisimulation on the transition system of $\langle P, T, \alpha, F, G, H\rangle$.[5]

Proposition 4. *For any unordered data net, whenever $s_1 \xrightarrow{t} s_2$ and $s'_1 \sim s_1$, we have $s'_1 \xrightarrow{t'} s'_2$ for some $t' \sim t$ and $s'_2 \sim s_2$.*

[5] Conditions (i) and (ii) in the definition of unordered data nets suggest an alternative formalisation, where only one region is used for indexing F, G and H, and only

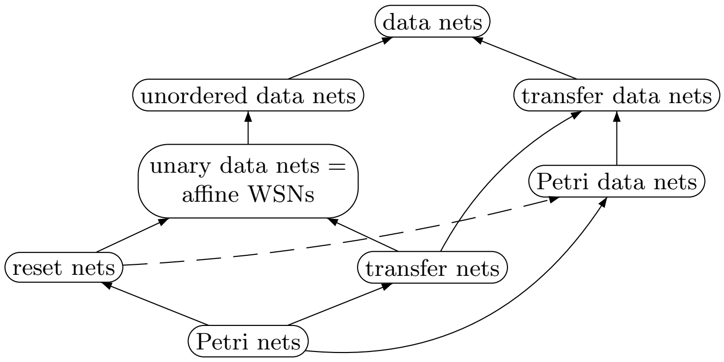

Fig. 1. Inclusions among classes of data nets

Unary data nets. A data net $\langle P, T, \alpha, F, G, H\rangle$ is unary iff:

(i) for each $t \in T$, $\alpha_t = 1$;
(ii) for each $t \in T$, there exists $p \in P$ such that $F_t(1, p) > 0$;
(iii) for each $t \in T$, $R \in \mathit{Regs}(1)$ and $p, p' \in P$, we have $G_t(1, p, R, p') = 0$, $G_t(R, p, 1, p') = 0$, $G_t(R, p, R, p) = 1$, $G_t(R, p, R, p') = 0$ if $p \neq p'$, and $H_t(R, p) = 0$.

Proposition 5. *Any unary data net is an unordered data net.*

Given a unary data net $\mathcal{N} = \langle P, T, \alpha, F, G, H\rangle$, let $\widetilde{\mathcal{N}} = \langle P, T, \tilde{F}, \tilde{G}, \tilde{H}\rangle$ be the affine WSN such that $\tilde{F}$, $\tilde{G}$ and $\tilde{H}$ are obtained from F_t, G_t and H_t (respectively) by removing entries which involve indices from $\mathit{Regs}(1)$. Observe that, conversely, for each affine WSN $\mathcal{N}'$ in which no transition is firable from $\mathbf{0}$, there is a unique unary data net $\mathcal{N}$ such that $\widetilde{\mathcal{N}} = \mathcal{N}'$. Both $\mathcal{N} \mapsto \widetilde{\mathcal{N}}$ and its inverse are computable in logarithmic space.

Proposition 6. *(a) For any unary data net $\mathcal{N}$, we have that $s \xrightarrow{t} s'$ iff $|s'| = |s|$ and there exists $i \in [|s|]$ with $s(i) \xrightarrow{t} s'(i)$ in $\widetilde{\mathcal{N}}$ and $s'(j) = s(j)$ for all $j \neq i$.*
(b) Coverability of s' from s in a unary data net $\mathcal{N}$ is equivalent to existence of an increasing $\iota : [|s'|] \to [|s|]$ such that $s'(i)$ is coverable from $s(\iota(i))$ in $\widetilde{\mathcal{N}}$ for each $i \in [|s'|]$.
Termination (resp., boundedness) from s in a unary data net $\mathcal{N}$ is equivalent to $\widetilde{\mathcal{N}}$ being terminating (resp., bounded) from $s(i)$ for each $i \in [|s|]$.
(c) Coverability of m' from m, termination from m and boundedness from m in an affine well-structured net $\widetilde{\mathcal{N}}$ are equivalent to coverability of $\langle m'\rangle$ from $\langle m\rangle$, termination from $\langle m\rangle$ and boundedness from $\langle m\rangle$ (respectively) in $\mathcal{N}$.

one transition from each equivalence class is represented. Such a formalisation is more succinct (exponentially in transition arities), but that issue is not important in this paper. In addition, by Proposition 4, markings of unordered data nets can be regarded as bags.

Note that Proposition 6 (c) can be extended to affine WSN with transitions firable from $\mathbf{0}$ by adding an auxiliary place in which a single token is kept.

Transfer data nets. A data net $\langle P, T, \alpha, F, G, H \rangle$ is transfer iff:

(i) for each $t \in T$, $i \in [\alpha_t]$ and $p \in P$, we have $G_t(i, p, i', p') > 0$ for some $i' \in [\alpha_t]$ and $p' \in P$;
(ii) for each $t \in T$, $R \in Regs(\alpha_t)$ and $p \in P$, either we have $G_t(R, p, i', p') > 0$ for some $i' \in [\alpha_t]$ and $p' \in P$, or we have $G_t(R, p, R, p') > 0$ for some $p' \in P$.

Observe that (i) and (ii) are satisfied by the transition t in Example 1.

Proposition 7. *(a) Whenever $s_1 \xrightarrow{t} s_2$ in a data net and $s_1' \geq s_1$, there exists $s_2' \geq s_2$ such that $s_1' \xrightarrow{t} s_2'$.*
(b) Whenever $s_1 \xrightarrow{t} s_2$ in a transfer data net and $s_1' > s_1$, there exists $s_2' > s_2$ such that $s_1' \xrightarrow{t} s_2'$.

Petri data nets. In Petri data nets, whole-place operations are not allowed, and transitions can produce tokens carrying only data which were chosen during the firing. Formally, a data net $\langle P, T, \alpha, F, G, H \rangle$ is Petri iff:

– for each $t \in T$, $G_t = Id$;
– for each $t \in T$, $R \in Regs(\alpha_t)$ and $p \in P$, $H_t(R, p) = 0$.

Proposition 8. *Any Petri data net is a transfer data net.*

2.5 Example: a file system

As an illustration, we now show how a file system which permits unboundedly many users, user processes and files can be modelled as a data net. A variety of other examples of systems expressible using data nets can be found in [10, 14, 11–13, 15, 16], including a real-timed mutual exclusion protocol, a distributed authentication protocol, a communication protocol over unreliable channels, and a leader election algorithm.

We suppose there are two categories of users: administrators and staff members. Let `Administrator` be a finite set consisting of all possible states which an administrator process can be in, and let `Staff` be such a set for staff-member processes. (We assume that `Administrator` and `Staff` are disjoint.) We consider two file permissions, so let `Permissions` $= \{$`private`, `public`$\}$. We also suppose `Contents` is a finite set of all possible file contents. If file contents is unbounded, the `Contents` set may consist of finitary abstractions, which include information such as file names.

The set of places is

$$P = \texttt{Administrator} \cup \texttt{Staff} \cup (\texttt{Permissions} \times \texttt{Contents})$$

Tokens represent user processes and files, and data which they carry represents user identities. More specifically:

- a token at place $a \in \texttt{Administrator}$ carrying datum d represents a process of administrator d and which is in state a;
- a token at place $b \in \texttt{Staff}$ carrying datum d represents a process of staff member d and which is in state b;
- a token at place $\langle r, c\rangle \in \texttt{Permissions} \times \texttt{Contents}$ carrying datum d represents a file owned by user d, and with permission r and contents c.

To express a write by a staff-member process in state b to a file with contents c, which changes b to b' and c to c', we define a transition $\texttt{write}(b, b', c, c')$. It involves one user, so $\alpha_{\texttt{write}(b,b',c,c')} = 1$. Firstly, it takes one token from place b and one token from place c. They must carry the same datum, which ensures that the user owns the file.

$$F_{\texttt{write}(b,b',c,c')}(1, b) = 1 \qquad F_{\texttt{write}(b,b',c,c')}(1, c) = 1$$

The transition involves no whole-place operations, so $G_{\texttt{write}(b,b',c,c')} = \mathit{Id}$. Finally, it puts one token onto place b' and one token onto place c', which carry the same datum as the two tokens taken in the first stage.

$$H_{\texttt{write}(b,b',c,c')}(1, b') = 1 \qquad H_{\texttt{write}(b,b',c,c')}(1, c') = 1$$

The remaining entries of $F_{\texttt{write}(b,b',c,c')}$ and $H_{\texttt{write}(b,b',c,c')}$ are 0.

As a slightly more complex example, we can express a change of ownership of a file with permission r and contents c from an administrator to a staff member. It involves an administrator process which changes state from a to a', and a staff-member processes which changes state from b to b'. Since two users are involved, we have $\alpha_{\texttt{change}(r,c,a,a',b,b')} = 2$. As in the previous example, $G_{\texttt{change}(r,c,a,a',b,b')} = \mathit{Id}$ and we show only entries which are not 0:

$$\begin{array}{rr}
F_{\texttt{change}(r,c,a,a',b,b')}(1, \langle r, c\rangle) = 1 & H_{\texttt{change}(r,c,a,a',b,b')}(2, \langle r, c\rangle) = 1 \\
F_{\texttt{change}(r,c,a,a',b,b')}(1, a) = 1 & H_{\texttt{change}(r,c,a,a',b,b')}(1, a') = 1 \\
F_{\texttt{change}(r,c,a,a',b,b')}(2, b) = 1 & H_{\texttt{change}(r,c,a,a',b,b')}(2, b') = 1
\end{array}$$

In the $\texttt{change}(r, c, a, a', b, b')$ transition, it is assumed that the administrator identity is smaller than the staff-member identity. To cover the opposite case, and to have an unordered data net, we define a transition $\texttt{change}(r, c, b, b', a, a')$. The definition is the same as that of $\texttt{change}(r, c, a, a', b, b')$, except that indices 1 and 2 are swapped when defining $F_{\texttt{change}(r,c,b,b',a,a')}$ and $H_{\texttt{change}(r,c,b,b',a,a')}$.

The data net having the three sets of transitions introduced so far is unordered and Petri. Implementing the following action makes it no longer Petri, in fact not even a transfer data net: all processes and files of a staff member who has a process which is in state b are removed from the system. We have $\alpha_{\texttt{crash}(b)} = 1$, $F_{\texttt{crash}(b)}(1, b) = 1$, the remaining entries of $F_{\texttt{crash}(b)}$ and all entries of $H_{\texttt{crash}(b)}$ are 0, and:

$$\begin{array}{rl}
G_{\texttt{crash}(s)}(1, p, 1, p') = 0 & \text{for } p, p' \in P \\
G_{\texttt{crash}(s)}(1, p, R, p') = 0 & \text{for } R \in \mathit{Regs}(1) \text{ and } p, p' \in P \\
G_{\texttt{crash}(s)}(R, p, 1, p') = 0 & \text{for } R \in \mathit{Regs}(1) \text{ and } p, p' \in P \\
G_{\texttt{crash}(s)}(R, p, R, p) = 1 & \text{for } R \in \mathit{Regs}(1) \text{ and } p \in P \\
G_{\texttt{crash}(s)}(R, p, R', p') = 0 & \text{otherwise}
\end{array}$$

Many interesting properties of the file system can be formalised as coverability, termination or boundedness properties. For example, that there is never a user who is both an administrator and a staff member amounts to none of the markings $s_{a,b}$ for $a \in \texttt{Administrator}$ and $b \in \texttt{Staff}$ being coverable, where $|s_{a,b}| = 1$, $s_{a,b}(1)(a) = s_{a,b}(1)(b) = 1$, and $s_{a,b}(1)(p) = 0$ for all $p \in P \setminus \{a, b\}$.

3 Reset nets and lossy channel systems

In this section, we first show how Petri data nets can express reset nets, which establishes the dashed inclusion in the diagram in Section 2.4. The translation preserves coverability and termination properties of reset nets.

Secondly, we show that Petri data nets can also express lossy channel systems [15]. The translation provides reductions of the location reachability and termination problems for lossy channel systems to the coverability, termination and boundedness problems for Petri data nets. Thus, the latter three problems will be shown non-primitive recursive: see Theorem 14.

Proposition 9. *(a) Coverability for reset nets is Turing reducible in polynomial space to coverability for Petri data nets.*
(b) Termination for reset nets is reducible in polynomial space to termination for Petri data nets, and to boundedness for Petri data nets.

Proof. We define a translation from reset nets $\mathcal{N} = \langle P, T, F, G, H\rangle$ to Petri data nets $\widehat{\mathcal{N}} = \langle \hat{P}, \hat{T}, \alpha, \hat{F}, \hat{G}, \hat{H}\rangle$. For each $t \in T$, let s^0_t be a sequence consisting of all $p \in P$ which are reset by t, i.e., such that $G(p, p) = 0$ (each occurring once).

The set of places of $\widehat{\mathcal{N}}$ is formed by adding a place to P: $\hat{P} = P \uplus \{\hat{p}\}$. In $\widehat{\mathcal{N}}$, each place $p \in P$ will store a single token, carrying a datum which represents the place p of $\mathcal{N}$. The place $\hat{p}$ will store as many tokens carrying the datum which represents a place p as there are tokens at p in $\mathcal{N}$. More precisely, for markings m of $\mathcal{N}$ and s of $\widehat{\mathcal{N}}$, we write $m \approx s$ iff for each $p \in P$, there exists $j_p \in [|s|]$ such that: $s(j_p)(p) = 1$, $s(j')(p) = 0$ for all $j' \neq j_p$, and $s(j_p)(\hat{p}) = m(p)$. The relation $\approx$ will be a bisimulation between $\mathcal{N}$ and $\widehat{\mathcal{N}}$.

The transitions of $\widehat{\mathcal{N}}$ are pairs of transitions of $\mathcal{N}$ and enumerations of P: $\hat{T} = \{\hat{t}_\pi \,:\, t \in T \wedge [|P|] \overset{\pi}{\leftrightarrow} P\}$. Suppose $m \approx s$, and let π be the enumeration of P such that $\pi^{-1}(p) < \pi^{-1}(p')$ iff $j_p < j_{p'}$. We shall have that:

(i) only transitions of the form $\hat{t}_\pi$ are firable from s;

(ii) $m \xrightarrow{t} m'$ implies $s \xrightarrow{\hat{t}_\pi} s'$ for some $m' \approx s'$;

(iii) $s \xrightarrow{\hat{t}_\pi} s'$ implies $m \xrightarrow{t} m'$ for some $m' \approx s'$.

Consider any $\hat{t}_\pi \in \hat{T}$. We set $\alpha_{\hat{t}_\pi} = |P| + |s^0_t|$. Indices $i \in [|P|]$ will be used to pick data which represent the places of $\mathcal{N}$, and indices $|P| + i$ will be used to pick fresh data (which are greater than all existing data) to simulate the resets of t. Since $\hat{G}_{\hat{t}_\pi} = Id$ is required for $\widehat{\mathcal{N}}$ to be a Petri data net, it remains to define

$\hat{F}_{\hat{t}_\pi}$ and $\hat{H}_{\hat{t}_\pi}$ so that (i)–(iii) above are satisfied. Each entry not listed below is set to 0:

$$\begin{array}{rcll} \hat{F}_{\hat{t}_\pi}(i,\pi(i)) = 1 & & \hat{F}_{\hat{t}_\pi}(i,\hat{p}) = F_t(\pi(i)) & (i \in [|P|]) \\ \hat{H}_{\hat{t}_\pi}(i,\pi(i)) = 1 & & \hat{H}_{\hat{t}_\pi}(i,\hat{p}) = H_t(\pi(i)) & (\pi(i) \notin s_t^0) \\ \hat{H}_{\hat{t}_\pi}(|P|+i, s_t^0(i)) = 1 & & \hat{H}_{\hat{t}_\pi}(|P|+i,\hat{p}) = H_t(s_t^0(i)) & (i \in [|s_t^0|]) \end{array}$$

Since any enumeration π of P is storable in polynomial space, we have that polynomial space suffices for the translation.

Given a marking m of $\mathcal{N}$, let s be a marking of $\widehat{\mathcal{N}}$ such that $m \approx s$. For (a), we have by (i)–(iii) above that a given marking m' is coverable from m in $\mathcal{N}$ iff some minimal s' such that $m' \approx s'$ is coverable from s in $\widehat{\mathcal{N}}$. For the first half of (b), we have by (i)–(iii) above that $\mathcal{N}$ terminates from m iff $\widehat{\mathcal{N}}$ terminates from s. For the second half, let $\widehat{\mathcal{N}}'$ be obtained from $\widehat{\mathcal{N}}$ (in logarithmic space) by adding a place $\hat{p}'$ and ensuring that each transition increases the number of tokens at $\hat{p}'$. Let s' be an arbitrary extension of s to place $\hat{p}'$. We have that $\mathcal{N}$ terminates from m iff $\widehat{\mathcal{N}}'$ is bounded from s'. $\square$

A *lossy channel system* is a tuple $\mathcal{S} = \langle Q, C, \Sigma, \Delta \rangle$, where Q is a finite set of locations, C is a finite set of channels, Σ is a finite alphabet, and $\Delta \subseteq Q \times C \times \{!, ?\} \times \Sigma \times Q$ is a set of transitions.

A state of $\mathcal{S}$ is a pair $\langle q, w \rangle$, where $q \in Q$ and $w : C \to \Sigma^*$. For each $c \in C$, the word $w(c)$ is the contents of channel c at state $\langle q, w \rangle$.

To define computation steps, we first define perfect computation steps, which either write a letter to the end of a channel, or read a letter from the beginning of a channel. For states $\langle q_1, w_1 \rangle$ and $\langle q_2, w_2 \rangle$, we write $\langle q_1, w_1 \rangle \rightarrow_{perf} \langle q_2, w_2 \rangle$ iff there exist $c \in C$ and $a \in \Sigma$ such that:

- either $\langle q_1, c, !, a, q_2 \rangle \in \Delta$ and $w_2 = w_1[c \mapsto (w_1(c))a]$,
- or $\langle q_1, c, ?, a, q_2 \rangle \in \Delta$ and $w_1 = w_2[c \mapsto a(w_2(c))]$.

Let $\sqsubseteq$ denote the "subword" well-quasi-ordering on Σ^*, obtained by lifting the equality relation on Σ (see Proposition 2). For example, we have $abba \sqsubseteq abracadabra$. For states $\langle q, w \rangle$ and $\langle q', w' \rangle$, we write $\langle q, w \rangle \sqsupseteq \langle q', w' \rangle$ iff $q = q'$ and $w(c) \sqsupseteq w'(c)$ for all $c \in C$, i.e., $\langle q', w' \rangle$ is obtained from $\langle q, w \rangle$ by losing zero or more letters.

A computation step $\langle q, w \rangle \rightarrow \langle q', w' \rangle$ of $\mathcal{S}$ consists of zero or more losses, followed by a perfect computation step, followed by zero or more losses. Thus, the $\rightarrow$ relation is defined by composing the $\rightarrow_{perf}$ and $\sqsupseteq$ relations: $\rightarrow \; = \; \sqsupseteq \rightarrow_{perf} \sqsupseteq$.

The following are two key decision problems for lossy channel systems:

Location reachability: Given a lossy channel system, a state $\langle q, w \rangle$ and a location q', to decide whether some state $\langle q', w' \rangle$ is reachable from $\langle q, w \rangle$.

Termination: Given a lossy channel system, and a state $\langle q, w \rangle$, to decide whether all computations from $\langle q, w \rangle$ are finite.

Proposition 10. *(a) Location reachability for lossy channel systems is reducible in logarithmic space to coverability for Petri data nets.*

(b) Termination for lossy channel systems is reducible in logarithmic space to termination for Petri data nets, and to boundedness for Petri data nets.

Proof. Given a lossy channel system $\mathcal{S} = \langle Q, C, \Sigma, \Delta \rangle$, we define a Petri data net $\mathcal{N}_\mathcal{S} = \langle P, T, \alpha, F, G, H \rangle$ as follows. We shall have that $\mathcal{N}_\mathcal{S}$ is computable in logarithmic space.

Let $P = Q \uplus C \uplus (C \times \Sigma)$. States $\langle q, w \rangle$ of $\mathcal{S}$ will be represented by markings $s \in Seq(\mathbb{N}^P \setminus \{\mathbf{0}\})$ as follows. At places in Q, there will be one token, which is at q, and which carries a datum d which is minimal in s. For each $c \in C$ with $w(c)$ empty, place c will contain one token which carries d. For each $c \in C$ with $w(c) = a_1 \cdots a_k$ and $k > 0$, there will be data $d \prec d_1^c \prec \cdots \prec d_k^c$ such that:

- place c contains one token which carries d_k^c;
- for each $a \in \Sigma$, place $\langle c, a \rangle$ contains one token carrying d_i^c for each $i \in [k]$ with $a_i = a$, and possibly some tokens carrying data greater than d_k^c.

Formally, we write $\langle q, w \rangle \approx s$ iff:

- $s(1)(q) = 1$, and $s(j)(q') = 0$ whenever either $j > 1$ or $q' \in Q \setminus \{q\}$;
- for each $c \in C$ with $w(c) = \varepsilon$, $s(1)(c) = 1$, and $s(j)(c) = 0$ for all $j > 1$;
- for each $c \in C$ with $w(c) = a_1 \cdots a_k$ and $k > 0$, there exist $1 < j_1^c < \cdots < j_k^c$ such that $s(j_k^c)(c) = 1$, $s(j')(c) = 0$ for all $j' \neq j_k^c$, and for each $1 \leq j' \leq j_k^c$ and $a' \in \Sigma$, we have

$$s(j')(c, a') = \begin{cases} 1, \text{ if there exists } i \in [k] \text{ with } j' = j_i^c \text{ and } a' = a_i \\ 0, \text{ otherwise} \end{cases}$$

For each read transition of $\mathcal{S}$, there will be $1 + |\Sigma|$ transitions of $\mathcal{N}_\mathcal{S}$, depending on whether the channel will become empty after the read, or the last letter of the new channel contents will be a':

$$\begin{aligned} T = \{& \langle q_1, c, !, a, q_2 \rangle \,:\, \langle q_1, c, !, a, q_2 \rangle \in \Delta\} \cup \\ \{& \langle q_1, c, ?, a, q_2, \varepsilon \rangle, \langle q_1, c, ?, a, q_2, a' \rangle \,:\, \langle q_1, c, ?, a, q_2 \rangle \in \Delta \wedge a' \in \Sigma\} \end{aligned}$$

When defining α_t, F_t and H_t for $t \in T$ below, we show only entries which are distinct from 0. Since $\mathcal{N}_\mathcal{S}$ is a Petri data net, we have $G_t = Id$ for each $t \in T$.

We shall have that, in computations of $\mathcal{N}_\mathcal{S}$, losses can happen only when reads are performed, but that will be sufficient for the result we are proving. Losses will occur when the datum which identifies the end of a channel and corresponds to the last letter is made smaller than the datum which corresponds to the second last letter. (Observe that, in data nets, we cannot specify that a transition be firable from a marking only if the latter contains no data which is between two particular data. If that were not so, perfect channel systems which are Turing-powerful would be expressible.)

Writes are performed using the minimal datum, which is then decreased:

$$\begin{aligned} \alpha_{\langle q_1, c, !, a, q_2 \rangle} &= 2 & H_{\langle q_1, c, !, a, q_2 \rangle}(1, q_2) &= 1 \\ F_{\langle q_1, c, !, a, q_2 \rangle}(2, q_1) &= 1 & H_{\langle q_1, c, !, a, q_2 \rangle}(1, \langle c, a \rangle) &= 1 \end{aligned}$$

Reads which make a channel c empty alter the datum carried by the token at place c to be the minimal datum:

$$\begin{array}{rr} F_{\langle q_1,c,?,a,q_2,\varepsilon\rangle}(1,q_1) = 1 & \alpha_{\langle q_1,c,?,a,q_2,\varepsilon\rangle} = 2 \\ F_{\langle q_1,c,?,a,q_2,\varepsilon\rangle}(2,c) = 1 & H_{\langle q_1,c,?,a,q_2,\varepsilon\rangle}(1,q_2) = 1 \\ F_{\langle q_1,c,?,a,q_2,\varepsilon\rangle}(2,\langle c,a\rangle) = 1 & H_{\langle q_1,c,?,a,q_2,\varepsilon\rangle}(1,c) = 1 \end{array}$$

The remaining reads from channel c decrease the datum carried by the token at place c to a value which identifies an occurrence of some a':

$$\begin{array}{rr} F_{\langle q_1,c,?,a,q_2,a'\rangle}(1,q_1) = 1 & \alpha_{\langle q_1,c,?,a,q_2,a'\rangle} = 3 \\ F_{\langle q_1,c,?,a,q_2,a'\rangle}(3,c) = 1 & H_{\langle q_1,c,?,a,q_2,a'\rangle}(1,q_2) = 1 \\ F_{\langle q_1,c,?,a,q_2,a'\rangle}(3,\langle c,a\rangle) = 1 & H_{\langle q_1,c,?,a,q_2,a'\rangle}(2,c) = 1 \\ F_{\langle q_1,c,?,a,q_2,a'\rangle}(2,\langle c,a'\rangle) = 1 & H_{\langle q_1,c,?,a,q_2,a'\rangle}(2,\langle c,a'\rangle) = 1 \end{array}$$

Now, the definition of $\mathcal{N}_\mathcal{S}$ ensures that the $\approx$ relation is an inverse simulation: whenever $\langle q,w\rangle \approx s$ and $s \to s'$, there exists $\langle q',w'\rangle$ such that $\langle q',w'\rangle \approx s'$ and $\langle q,w\rangle \to \langle q',w'\rangle$.

We write $\langle q,w\rangle \sqsubseteq\approx s$ iff there exists $\langle q^\dagger,w^\dagger\rangle$ such that $\langle q,w\rangle \sqsubseteq \langle q^\dagger,w^\dagger\rangle$ and $\langle q^\dagger,w^\dagger\rangle \approx s$. It is straightforward to check that the $\sqsubseteq\approx$ relation is a simulation: whenever $\langle q,w\rangle \sqsubseteq\approx s$ and $\langle q,w\rangle \to \langle q',w'\rangle$, there exists s' such that $\langle q',w'\rangle \sqsubseteq\approx s'$ and $s \to s'$.

To establish (a), given a state $\langle q,w\rangle$ and a location q' of $\mathcal{S}$, let s be such that $\langle q,w\rangle \approx s$, and let s' be such that $|s'| = 1$, $s'(1)(q') = 1$, and $s'(1)(p) = 0$ for all $p \in P \setminus \{q'\}$. By the properties above, we have that some state $\langle q',w'\rangle$ is reachable from $\langle q,w\rangle$ iff some marking $s'' \geq s'$ is reachable from s.

For the termination part of (b), if s is such that $\langle q,w\rangle \approx s$, then $\mathcal{S}$ has an infinite computation from $\langle q,w\rangle$ iff $\mathcal{N}_\mathcal{S}$ has an infinite computation from s. For the boundedness part, we modify $\mathcal{N}_\mathcal{S}$ by adding an auxiliary place and ensuring that each transition increases the number of tokens at that place. □

4 Decidability

The following two lemmas will be used in the proof of Theorem 13 below. The first one, due to Valk and Jantzen, provides a sufficient condition for computability of finite bases of upwards-closed sets of fixed-length tuples of natural numbers. The second lemma shows that, for computing a pred-basis of the upward closure of a marking of a data net, it suffices to consider markings up to a certain computable length.

Lemma 11 ([21]). *Suppose B is a finite set. A finite basis of an upwards-closed set $V \subseteq \mathbb{N}^B$ is computable iff it is decidable, given any $v \in \mathbb{N}^B_\omega$, whether $V \cap \downarrow\{v\} \neq \emptyset$.*

For a transition system $\langle S,\to\rangle$ and $S' \subseteq S$, we write $Pred(S')$ for $\{s \in S : \exists s' \in S' \cdot s \to s'\}$. If transitions are labelled by $t \in T$, we write $Pred_t(S')$ for $\{s \in S : \exists s' \in S' \cdot s \xrightarrow{t} s'\}$.

Lemma 12. *Given a data net $\mathcal{N}$, a transition t of $\mathcal{N}$, and a marking s' of $\mathcal{N}$, a natural number L is computable, such that whenever $s \in Pred_t(\uparrow\{s'\})$ and $|s| > L$, there exists $\bar{s} \leq s$ with $\bar{s} \in Pred_t(\uparrow\{s'\})$ and $|\bar{s}| \leq L$.*

Proof. Suppose $\mathcal{N} = \langle P, T, \alpha, F, G, H\rangle$, and let

$$L = \alpha_t + |s'| + (\alpha_t + 1) \times (2^{|P|} - 1) \times M$$

where $M = \max\{s'(i)(p) \,:\, i \in [|s'|] \wedge p \in P\}$.

Consider $s \in Pred_t(\uparrow\{s'\})$ with $|s| > L$. For some $s_\dagger$, ι and $s'' \geq s'$, we have $s \xrightarrow{t,s_\dagger,\iota} s''$. Let $s''_\dagger = (s_\dagger - [\![F_t]\!]_\iota^{|s_\dagger|})[\![G_t]\!]_\iota^{|s_\dagger|} + [\![H_t]\!]_\iota^{|s_\dagger|}$. Since s'' is the **0**-contraction of $s''_\dagger$, there exists an increasing $\iota' : [|s'|] \to [|s_\dagger|]$ such that $s'(i) \leq s''_\dagger(\iota'(i))$ for all $i \in [|s'|]$.

For each nonempty $P_+ \subseteq P$, let

$$s_\dagger^{P_+} = \{i \in [|s_\dagger|] \,:\, \forall p \in P \cdot s_\dagger(i)(p) > 0 \Leftrightarrow p \in P_+\}$$

Since $|s_\dagger| \geq |s|$, there exist $0 \leq j \leq \alpha_t$ and nonempty $P_+ \subseteq P$ such that $|I_j^{P_+}| > M$, where $I_j^{P_+} = ([\![Reg_{(j,j+1)}]\!]_\iota^{|s_\dagger|} \setminus Range(\iota')) \cap s_\dagger^{P_+}$.

Pick an index $i_\dagger^1 \in I_j^{P_+}$ of $s_\dagger$, and let $i^1 \in [|s|]$ be the corresponding index of s. Let $\tau_\dagger$ be the increasing mapping $[|s_\dagger| - 1] \to [|s_\dagger|]$ with $i_\dagger^1 \notin Range(\tau_\dagger)$, and τ be the increasing mapping $[|s| - 1] \to [|s|]$ with $i^1 \notin Range(\tau)$. Then let $s_\dagger^1$ (resp., s^1) be obtained from $s_\dagger$ (resp., s) by removing the entry $i_\dagger^1$ (resp., i^1), $\iota_1 = \tau_\dagger^{-1} \circ \iota$, and $s''^1_\dagger = (s^1_\dagger - [\![F_t]\!]_{\iota_1}^{|s^1_\dagger|})[\![G_t]\!]_{\iota_1}^{|s^1_\dagger|} + [\![H_t]\!]_{\iota_1}^{|s^1_\dagger|}$. By the definition of $I_j^{P_+}$ and $|I_j^{P_+}| > M$, we have that $s''^1_\dagger(i)(p) \geq M$ whenever $s''^1_\dagger(i)(p) \neq s''_\dagger(\tau_\dagger(i))(p)$. Hence, $s''^1_\dagger \geq s'$, so $s^1 \in Pred_t(\uparrow\{s'\})$.

By repeating the above, we obtain $s \geq s^1 \geq s^2 \geq \cdots s^{|s|-L} \in Pred_t(\uparrow\{s'\})$ such that $|s^k| = |s| - k$ for all k. Setting $\bar{s} = s^{|s|-L}$ completes the proof. ☐

Theorem 13. *(a) Coverability and termination for data nets are decidable.*
(b) Boundedness for transfer data nets is decidable.

Proof. Suppose $\mathcal{N} = \langle P, T, \alpha, F, G, H\rangle$ is a data net. By Propositions 2, 3 and 7, we have that the transition system of $\mathcal{N}$ is finitely-branching and well-structured with strong compatibility, and also with strict compatibility if $\mathcal{N}$ is transfer (using the terminology of [18]). Moreover, $\leq$ between markings of $\mathcal{N}$ is a decidable partial ordering, and $Succ(s) = \{s' \,:\, s \to s'\}$ is computable for markings s. Hence, termination for data nets and boundedness for transfer data nets are decidable by [18, Theorems 4.6 and 4.11].

To establish decidability of coverability by [18, Theorem 3.6], it suffices to show that, given any $t \in T$ and a marking s', a finite basis of $Pred_t(\uparrow\{s'\})$ is computable. (By Proposition 7 (a), $Pred_t(\uparrow\{s'\})$ is upwards-closed.)

First, we compute L as in Lemma 12. For any $0 \leq l \leq L$, increasing $\eta : [l] \to [l_\dagger]$ and increasing $\iota : [\alpha_t] \to [l_\dagger]$ such that $[l_\dagger] = Range(\eta) \cup Range(\iota)$, let

$$Pred^l_{t,\eta,\iota}(\uparrow\{s'\}) = \{s \,:\, l = |s| \wedge \exists s'' \geq s' \cdot s \xrightarrow{t,\eta,\iota} s''\}$$

where $s \xrightarrow{t,\eta,\iota} s''$ means that $s \xrightarrow{t,s_\dagger,\iota} s''$ for some $s_\dagger$ such that $Range(\eta) = \{j : s_\dagger(j) \neq \mathbf{0}\}$ (necessarily, $l_\dagger = |s_\dagger|$). From the definition of transition firing, we have that $s \xrightarrow{t,s_\dagger,\iota} s''$ iff $s_\dagger \geq [\![F_t]\!]_\iota^{l_\dagger}$ and s'' is the $\mathbf{0}$-contraction of $(s_\dagger - [\![F_t]\!]_\iota^{l_\dagger})[\![G_t]\!]_\iota^{l_\dagger} + [\![H_t]\!]_\iota^{l_\dagger}$. Hence, each $Pred^l_{t,\eta,\iota}(\uparrow\{s'\})$ is an upwards-closed subset of $\mathbb{N}^{P\times[l]}$. By Lemma 12, it remains to compute a finite basis of each $Pred^l_{t,\eta,\iota}(\uparrow\{s'\})$.

Suppose that l, η and ι are as above. Given any $s \in \mathbb{N}_\omega^{P\times[l]}$, we have as in [6] that $Pred^l_{t,\eta,\iota}(\uparrow\{s'\}) \cap \downarrow\{s\} \neq \emptyset$ iff $s_\dagger \geq [\![F_t]\!]_\iota^{l_\dagger}$ and $s'' \geq s'$, where $s_\dagger$ is the $\mathbf{0}$-expansion of s such that $l_\dagger = |s_\dagger|$ and $Range(\eta) = \{j : s_\dagger(j) \neq \mathbf{0}\}$, s'' is the $\mathbf{0}$-contraction of $(s_\dagger - [\![F_t]\!]_\iota^{l_\dagger})[\![G_t]\!]_\iota^{l_\dagger} + [\![H_t]\!]_\iota^{l_\dagger}$, and the required operations are extended to ω by taking limits: $\omega \geq n$, $\omega + n = n + \omega = \omega + \omega = \omega$, $\omega - n = \omega$, $0 \times \omega = 0$, and $n \times \omega = \omega$ for $n > 0$. Therefore, by Lemma 11, a finite basis of $Pred^l_{t,\eta,\iota}(\uparrow\{s'\})$ is computable. □

5 Hardness

Theorem 14. *Coverability, termination and boundedness for Petri data nets are not primitive recursive.*

Proof. As shown in [7], location reachability and termination for lossy channel systems are not primitive recursive. It remains to apply Proposition 10. □

Theorem 15. *Coverability, termination and boundedness for unordered Petri data nets are not elementary.*

Proof. For $n \in \mathbb{N}$, the *tetration* operation $a \Uparrow n$ is defined by $a \Uparrow 0 = 1$ and $a \Uparrow (n+1) = a^{a \Uparrow n}$.

The non-elementariness of the three verification problems follows from showing that, given a deterministic machine $\mathcal{M}$ of size n with finite control and two $2 \Uparrow n$-bounded counters, an unordered Petri data net $\mathcal{N}_\mathcal{M}$ which simulates $\mathcal{M}$ is constructible in logarithmic space. A counter is m-bounded iff it can have values in $\{0, \ldots, m-1\}$, i.e., it cannot be incremented beyond the maximum value $m-1$. The following counter operations may be used in $\mathcal{M}$: increment, decrement, reset, iszero and ismax.

It will be defined below when a marking of $\mathcal{N}_\mathcal{M}$ represents a configuration (i.e., state) of $\mathcal{M}$. Let us call such markings "clean". We write $s \rightarrow_{\surd} s'$ (resp., $s \rightarrow_{\times} s'$) iff $s \rightarrow s'$ and s' is clean (resp., not clean). Hence, $s \rightarrow_{\times}^{*}\rightarrow_{\surd} s'$ means that s' is clean and reachable from s by a nonempty sequence of transitions in which every intermediate marking is not clean, and $s \not\rightarrow_{\times}^{\omega}$ means that there does not exist an infinite sequence of transitions from s in which no intermediate marking is clean. $\mathcal{M}$ will be simulated in the following sense by $\mathcal{N}_\mathcal{M}$ from a certain initial marking s_I, where c_I is the initial configuration of $\mathcal{M}$:

- we have $s_I \not\rightarrow_{\times}^{\omega}$ and:
  - there exists $s_I \rightarrow_{\times}^{*}\rightarrow_{\surd} s'$ such that c_I is represented by s';
  - for all $s_I \rightarrow_{\times}^{*}\rightarrow_{\surd} s'$, c_I is represented by s';

– whenever c is represented by s, we have $s \not\rightarrow^{\omega}_{\times}$ and:
  - if c has a successor c', there exists $s \rightarrow^{*}_{\times}\rightarrow_{\surd} s'$ with c' represented by s';
  - for all $s \rightarrow^{*}_{\times}\rightarrow_{\surd} s'$, c has a successor c' which is represented by s'.

That $\mathcal{M}$ halts (i.e. reaches a halting control state from c_I) will therefore be equivalent to a simple coverability question from s_I, and to termination from s_I. After extending $\mathcal{N}_{\mathcal{M}}$ by a place whose number of tokens increases with each transition, that $\mathcal{M}$ halts becomes equivalent to boundedness from s_I.

Each clean marking s of $\mathcal{N}_{\mathcal{M}}$ will represent a valuation v of $3n$ counters C_k, C'_k and C''_k for $k \in [n]$. C_n and C'_n are the two counters of $\mathcal{M}$, and for each $k \in [n]$, C_k, C'_k and C''_k are $2 \Uparrow k$-bounded. (Counter C''_n will not be used, so it can be omitted.) $\mathcal{N}_{\mathcal{M}}$ will have places 0_D, 1_D, $\mathit{scratch}_D$, lock_D, $\mathit{checked}_D$ and $\mathit{unchecked}_D$ for each $D \in \{C_k, C'_k, C''_k : k \in [n]\}$, as well as a number (polynomial in n) of places for encoding the control of $\mathcal{M}$ and for control of $\mathcal{N}_{\mathcal{M}}$. A valuation v is represented by s as follows:

– for each $k \in [n]$ and $D \in \{C_k, C'_k, C''_k\}$, places $\mathit{scratch}_D$, lock_D and $\mathit{checked}_D$ are empty, and $\mathit{unchecked}_D$ contains exactly $2 \Uparrow (k-1)$ tokens and they carry mutually distinct data;
– for each $k \in [n]$, $D \in \{C_k, C'_k, C''_k\}$ and $i \in [2 \Uparrow (k-1)]$, if the i-th bit of $v(D)$ is $b \in \{0,1\}$, then for some datum d carried by a token at place $\mathit{unchecked}_D$, the number of tokens at b_D which carry d is i, and the number of tokens at $(1-b)_D$ which carry d is 0;
– for each $k \in [n]$ and $D \in \{C_k, C'_k, C''_k\}$, each datum carried by a token at 0_D or 1_D is carried by some token at $\mathit{unchecked}_D$.

Counters C_1, C'_1 and C''_1 are 2-bounded, so operations on them are trivial to simulate. For each $k < n$, counter operations on C_{k+1}, C'_{k+1} and C''_{k+1} are simulated using operations on C_k, C'_k and C''_k. The following shows how to implement $\mathsf{iszero}(D)$, where $D \in \{C_{k+1}, C'_{k+1}, C''_{k+1}\}$. The other four counter operations are implemented similarly.

for $C_k := 0$ to $(2 \Uparrow k) - 1$ do
{ guess a datum d and move a token carrying d from $\mathit{unchecked}_D$ to lock_D;
  for $C'_k := 0$ to C_k do { move a token carrying d from 0_D to $\mathit{scratch}_D$ };
  for $C'_k := 0$ to C_k do { move a token carrying d from $\mathit{scratch}_D$ to 0_D };
  move the token from lock_D to $\mathit{checked}_D$ };
for $C_k := 0$ to $(2 \Uparrow k) - 1$ do
{ move a token from $\mathit{checked}_D$ to $\mathit{unchecked}_D$ }

Observe that $\mathsf{iszero}(D)$ can execute completely iff, for each $i \in [2 \Uparrow k]$, the datum d guessed in the i-th iteration of the outer loop represents the i-th bit of $v(D)$ and that bit is 0. Place lock_D is used for keeping the datum d during each such iteration, and it is implicitly employed within the two inner loops.

It remains to implement routines $\mathsf{setup}(D)$ for $k \in [n]$ and $D \in \{C_k, C'_k, C''_k\}$, which start from empty 0_D, 1_D, $\mathit{scratch}_D$, lock_D, $\mathit{checked}_D$ and $\mathit{unchecked}_D$, and set up 0_D and $\mathit{unchecked}_D$ to represent D having value 0. Setting up C_1, C'_1 and C''_1 is trivial. To implement $\mathsf{setup}(D)$ for $k < n$ and $D \in \{C_{k+1}, C'_{k+1}, C''_{k+1}\}$, we use C_k, C'_k and C''_k which were set up previously. The implementation is similar to that of $\mathsf{iszero}(D)$, except that all three of C_k, C'_k and C''_k are used, since whenever a datum d is picked to be the i^{th} datum at $\mathit{unchecked}_D$ for some $i \in [2 \Uparrow k]$, two nested loops are employed to ensure that d is distinct from each of $i-1$ data which are carried by tokens already at $\mathit{unchecked}_D$. $\square$

6 Concluding remarks

We have answered questions (1) and (2) posed in Section 1. As far as we are aware, Section 5 contains the first nontrivial lower bounds on complexity of decidable problems for extensions of Petri nets by infinite data domains.

The results obtained and their proofs show that data nets are a succinct unifying formalism which is close to the underlying semantic structures, and thus a useful platform for theoretical investigations.

The proof of Theorem 13 does not provide precise upper bounds on complexity. It should be investigated whether upper bounds which match the lower bounds in the proofs of Theorems 14 and 15 are obtainable. In particular, are coverability, termination and boundedness for unordered Petri data nets primitive recursive?

Let us say that a data net is l, m*-safe* iff each place other than some l places never contains more than m tokens. It is not difficult to tighten the proofs of Theorems 14 and 15 to obtain that coverability, termination and boundedness are not primitive recursive for $1,1$-safe Petri data nets, and not elementary for $2,1$-safe unordered Petri data nets. That leaves open whether we have non-elementarity for $1,1$-safe unordered Petri data nets. That class suffices for expressing polymorphic systems with one array of type $\langle X, = \rangle \to \langle Y, = \rangle$ without whole-array operations [16, 17].

We are grateful to Alain Finkel for a helpful discussion.

References

1. Reisig, W.: Petri Nets: An Introduction. Springer (1985)
2. Girault, C., Valk, R., eds.: Petri Nets for Systems Engineering. Springer (2003)
3. Esparza, J., Nielsen, M.: Decidability issues for Petri nets – a survey. Bull. EATCS **52** (1994) 244–262
4. Lipton, R.J.: The reachability problem requires exponential space. Technical Report 62, Yale University (1976)
5. Rackoff, C.: The covering and boundedness problems for vector addition systems. Theor. Comput. Sci. **6** (1978) 223–231
6. Finkel, A., McKenzie, P., Picaronny, C.: A well-structured framework for analysing Petri net extensions. Inf. Comput. **195**(1–2) (2004) 1–29
7. Schnoebelen, P.: Verifying lossy channel systems has nonprimitive recursive complexity. Inf. Proc. Lett. **83**(5) (2002) 251–261
8. Odifreddi, P.: Classical Recursion Theory II. Elsevier (1999)
9. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset nets between decidability and undecidability. In: ICALP. Volume 1443 of LNCS., Springer (1998) 103–115

10. Abdulla, P.A., Nylén, A.: Timed Petri nets and BQOs. In: ICATPN. Volume 2075 of LNCS., Springer (2001) 53–70
11. Delzanno, G.: Constraint multiset rewriting. Technical Report DISI-TR-05-08, Università di Genova (2005) Extends [22–24].
12. Abdulla, P.A., Delzanno, G.: Constrained multiset rewriting. In: AVIS. ENTCS (2006) to appear.
13. Rosa Velardo, F., de Frutos Escrig, D., Marroquín Alonso, O.: On the expressiveness of mobile synchronizing Petri nets. In: SECCO. ENTCS (2005) to appear.
14. Abdulla, P.A., Jonsson, B.: Model checking of systems with many identical timed processes. Theor. Comput. Sci. **290**(1) (2003) 241–264
15. Abdulla, P.A., Jonsson, B.: Verifying programs with unreliable channels. Inf. Comput. **127**(2) (1996) 91–101
16. Lazić, R., Newcomb, T.C., Roscoe, A.W.: Polymorphic systems with arrays, 2-counter machines and multiset rewriting. In: Infinity '04. Volume 138 of ENTCS. (2005) 61–86
17. Lazić, R.: Decidability of reachability for polymorphic systems with arrays: A complete classification. In: Infinity '04. Volume 138 of ENTCS. (2005) 3–19
18. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! Theor. Comput. Sci. **256**(1–2) (2001) 63–92
19. Meyer, A.R.: Weak monadic second-order theory of successor is not elementary-recursive. In: Logic colloquium '72–73. Volume 453 of Lect. Not. Math., Springer (1975) 132–154
20. Higman, G.: Ordering by divisibility in abstract algebras. Proc. London Math. Soc. (3) **2**(7) (1952) 326–336
21. Valk, R., Jantzen, M.: The residue of vector sets with applications to decidability problems in Petri nets. Acta Inf. **21** (1985) 643–674
22. Delzanno, G.: An assertional language for systems parametric in several dimensions. In: VEPAS. Volume 50 of ENTCS. (2001)
23. Bozzano, M., Delzanno, G.: Beyond parameterized verification. In: TACAS. Volume 2280 of LNCS., Springer (2002) 221–235
24. Bozzano, M., Delzanno, G.: Automatic verification of invalidation-based protocols. In: CAV. Volume 2404 of LNCS., Springer (2002) 295–308