

Soundness of Workflow Nets with an Unbounded Resource is Decidable^{*}

Vladimir A. Bashkin¹ and Irina A. Lomazova^{2,3}

¹ Yaroslavl State University, Yaroslavl, 150000, Russia
v_bashkin@mail.ru

² National Research University Higher School of Economics,
Moscow, 101000, Russia

³ Program Systems Institute of the Russian Academy of Science,
Pereslavl-Zalessky, 152020, Russia
i_lomazova@mail.ru

Abstract. In this work we consider modeling of workflow systems with Petri nets. A resource workflow net (RWF-net) is a workflow net, supplied with an additional set of initially marked resource places. Resources can be consumed and/or produced by transitions. We do not constrain neither the intermediate nor final resource markings, hence a net can have an infinite number of different reachable states.

An initially marked RWF-net is called sound if it properly terminates and, moreover, adding any extra initial resource does not violate its proper termination. An (unmarked) RWF-net is sound if it is sound for some initial resource. In this paper we prove the decidability of both marked and unmarked soundness for a restricted class of RWF-nets with a single unbounded resource place (1-dim RWF-nets). We present an algorithm for computing the minimal sound resource for a given sound 1-dim RWF-net.

1 Introduction

Petri nets constitute a popular formalism for modeling and analysis of distributed systems. In this paper we consider workflow systems, or, to be more precise, workflow processes. To model workflow processes a special subclass of Petri nets, called WF-nets [1, 2], is used.

In the context of WF-nets a crucial correctness criterion is soundness [1, 3]. We say that a workflow case execution terminates properly, iff its firing sequence (starting from the initial marking with a single token in the initial place) terminates with a single token in the final place (i.e. there are no “garbage” tokens after the termination). A model is called sound iff a process can terminate properly starting from any reachable marking.

^{*} This work is supported by the Basic Research Program of the National Research University Higher School of Economics, Russian Fund for Basic Research (projects 11-01-00737, 12-01-31508, 12-01-00281), and by Federal Program "Human Capital for Science and Education in Innovative Russia" (Contract No. 14.B37.21.0392).

Soundness of WF-nets is decidable [1]. Moreover, a number of decidable variations of soundness are established, for example, k -soundness [9], structural soundness [14] and soundness of nested models [11].

One of important aspects in workflow development concerns resource management. Resources here is a general notion for executives (people or devices), raw materials, finances, etc. To take resources into account different extensions of a base formalism were introduced, having different versions of soundness criteria.

In [5, 6] a specific class of WFR-nets with decidable soundness is studied. In [10, 13] a more general class of Resource-Constrained Workflow Nets (RCWF-nets) is defined. Informally, the authors impose two constraints on resources. First, they require that all resources that are initially available are available again after terminating of all cases. Second, they also require that for any reachable marking, the number of available resources does not override the number of initially available resources.

In [10] it is proven that for RCWF-nets with a single resource type soundness can be effectively checked in polynomial time. In [13] it is proven that soundness is also decidable in general case (by reducing to the home-space problem).

In all mentioned papers resources are assumed to be permanent, i.e. they are used (blocked) and released later on. Resources are never created, nor destroyed. Hence the process state space is explicitly bounded.

To study a more general case of arbitrary resource transformations (that can arise, for example, in open and/or adaptive workflow systems), in [8] we defined a notion of WF-nets with resources (RWF-nets). RWF-nets extend RCWF-nets from [10] in such a way that resources can be generated or consumed during a process execution without any restrictions (cf. [7]). For RWF-nets we defined notions of resources and controlled resources and studied the problem of soundness-preserving resource replacement (this problem is important for adaptive workflows).

Unfortunately, even sound RWF-nets are not bounded in general, hence existing soundness checking algorithms cannot be applied here. In [8] the decidability of soundness for RWF-nets was declared as an open problem.

In this paper we consider a restricted case — RWF-nets with a single resource place (1-dim RWF-nets). One resource type is sufficient for many practical applications (memory or money are typical examples of such resources). Note that 1-dim RWF-nets are, generally speaking, not bounded and hence this case cannot be reduced to finite-state WF-nets with resources, such as RCWF- or WFR-nets.

In this paper we use graph-theoretic properties of RWF-net control automaton to prove decidability of soundness for marked, as well as unmarked 1-dim RWF-nets. We present also an algorithm for computing minimal sound resource for a given sound 1-dim RWF-net.

The paper is organized as follows. In Section 2 basic definitions of multisets and Petri nets are given. In Section 3 we give definitions of sound RWF-nets. In Section 4 the class of 1-dim RWF-nets is defined and studied, algorithms for

checking marked soundness, soundness and finding the minimal sound resource are given. Section 5 contains some conclusions.

2 Preliminaries

Let S be a finite set. A *multiset* m over a set S is a mapping $m : S \rightarrow \text{Nat}$, where Nat is the set of natural numbers (including zero), i. e. a multiset may contain several copies of the same element.

For two multisets m, m' we write $m \subseteq m'$ iff $\forall s \in S : m(s) \leq m'(s)$ (the inclusion relation). The sum, the union and the subtraction of two multisets m and m' are defined as usual: $\forall s \in S : (m + m')(s) = m(s) + m'(s)$, $(m \cup m')(s) = \max(m(s), m'(s))$, $(m - m')(s) = m(s) \ominus m'(s)$ (where \ominus denotes the truncated subtraction). By $\mathcal{M}(S)$ we denote the set of all finite multisets over S .

Non-negative integer vectors are often used to encode multisets. Actually, the set of all multisets over finite S is a homomorphic image of $\text{Nat}^{|S|}$.

Let P and T be nonempty disjoint sets of *places* and *transitions* and let $F : (P \times T) \cup (T \times P) \rightarrow \text{Nat}$. Then $N = (P, T, F)$ is a *Petri net*. A *marking* in a Petri net is a function $M : P \rightarrow \text{Nat}$, mapping each place to some natural number (possibly zero). Thus a marking may be considered as a multiset over the set of places. Pictorially, P -elements are represented by circles, T -elements by boxes, and the flow relation F by directed arcs. Places may carry tokens represented by filled circles. A current marking M is designated by putting $M(p)$ tokens into each place $p \in P$. Tokens residing in a place are often interpreted as resources of some type consumed or produced by a transition firing. A simple example, where tokens represent molecules of hydrogen, oxygen and water respectively is shown in Fig. 1.

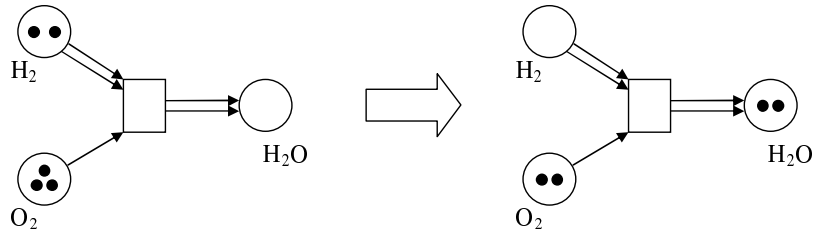


Fig. 1. A chemical reaction.

For a transition $t \in T$ an arc (x, t) is called an *input arc*, and an arc (t, x) — an *output arc*; the *preset* $\bullet t$ and the *postset* $t \bullet$ are defined as the multisets over P such that $\bullet t(p) = F(p, t)$ and $t \bullet(p) = F(t, p)$ for each $p \in P$. A transition $t \in T$ is *enabled* in a marking M iff $\forall p \in P M(p) \geq F(p, t)$. An enabled transition t may *fire* yielding a new marking $M' =_{\text{def}} M - \bullet t + t \bullet$, i. e. $M'(p) = M(p) - F(p, t) +$

$F(t, p)$ for each $p \in P$ (denoted $M \xrightarrow{t} M'$, or just $M \rightarrow M'$). We say that M' is reachable from M iff there is a sequence $M = M_1 \rightarrow M_2 \rightarrow \dots \rightarrow M_n = M'$. For a Petri net N by $\mathcal{R}(N, M_0)$ we denote the set of all markings reachable from its initial marking M_0 .

3 WF-nets with resources

In Petri nets with resources we divide Petri net places into control and resource ones.

Definition 1. A Petri net with resources is a tuple $N = (P_c, P_r, T, F_c, F_r)$, where

- P_c is a finite set of control places;
- P_r is a finite set of resource places, $P_c \cap P_r = \emptyset$;
- T is a finite set of transitions, $P_c \cap T = P_r \cap T = \emptyset$;
- $F_c : (P_c \times T) \cup (T \times P_c) \rightarrow \text{Nat}$ is a multiset of control arcs;
- $F_r : (P_r \times T) \cup (T \times P_r) \rightarrow \text{Nat}$ is a multiset of resource arcs;
- $\forall t \in T \exists p \in P_c : F_c(p, t) + F_c(t, p) > 0$ (each transition is incident to some control place).

Note that all transitions are necessarily linked to control places — this guarantees the absence of “uncontrolled” resource modifications.

A marking in a Petri net with resources is also divided into control and resource parts. For a multiset $c + r$, where $c \in \mathcal{M}(P_c)$ and $r \in \mathcal{M}(P_r)$, we write $c|r$.

Definition 2. For a net N a resource is a multiset over P_r . A controlled resource is a multiset over $P_c \cup P_r$.

Workflow nets (WF-nets) are a special subclass of Petri nets designed for modeling workflow processes. To study resource dependencies in workflow systems we consider WF-nets with resources.

Definition 3. A Petri net with resources N is called a WF-net with resources (RWF-net) iff

1. There is one source place $i \in P_c$ and one sink place $o \in P_c$ s. t. $\bullet i = o \bullet = \emptyset$;
2. Every node from $P_c \cup T$ is on a path from i to o , and this path consists of nodes from $P_c \cup T$.

Fig. 2 represents an example of a RWF-net, where resource places r_1 and r_2 are depicted by ovals, resource arcs — by dotted arrows.

Every RWF-net $N = (P_c, P_r, T, F_c, F_r)$ contains its control subnet $N_c = (P_c, T, F_c)$, which forms a RWF-net with the empty set of resources.

A marked net is a net together with some initial marking.

Definition 4. A marked RWF-net $(N, c|r)$ is called sound iff $\forall s \in \mathcal{M}(P_r), \forall M \in \mathcal{R}(N, c|r + s)$ we have:

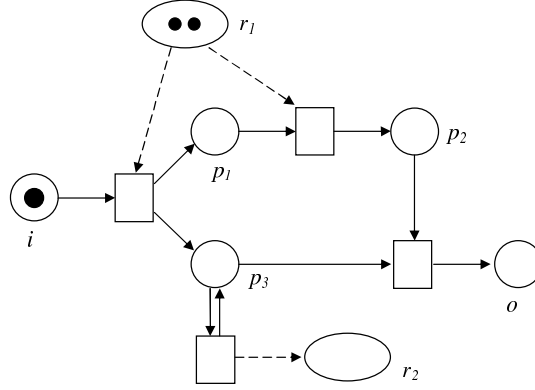


Fig. 2. WF-net with resources.

1. $\exists s' \in \mathcal{M}(P_r) : o|s' \in \mathcal{R}(N, M)$;
2. $c'|r' \in \mathcal{R}(N, M) \Rightarrow c' = o \vee c' \cap o = \emptyset$.

Thus soundness for a RWF-net means that, first, this workflow net can terminate properly from any reachable state, and, additionally, adding any extra resource does not violate the proper termination property.

Note that our definition is substantially different from the definition of sound RCWF-nets (Resource-Constrained Workflow net) in [10]. We do not forbid creating and spending of resources. Thus, in RWF-nets resources may be produced and consumed during a process execution. This implies the possible unboundness of sound RWF-nets.

The following statement is analogous to Lemma 5 in [10].

Proposition 1. [7] *If a marked RWF-net $(N, i|r)$ is sound, then its control subnet N_c with the initial marking i is also sound.*

The proof of this proposition is given in the Appendix.

The converse statement is not true: there may be RWF-nets with sound control subnets, for which sound resources do not exist. An example of such a net is given in Fig. 3.

Let N be a RWF-net. By $C(N)$ we denote the set of all control markings reachable in N_c , i. e. $C(N) = \mathcal{R}(N_c, i)$.

Proposition 2. [7] *If a marked RWF-net $(N, i|r)$ is sound, then*

1. *for any reachable control marking $c \in C(N)$ there exists a resource r' , such that $(N, c|r')$ is sound;*
2. *for any two control markings $c_1, c_2 \in C(N)$ we have $c_1 \not\subseteq c_2$ and $c_2 \not\subseteq c_1$.*

The proof of this proposition is given in the Appendix.

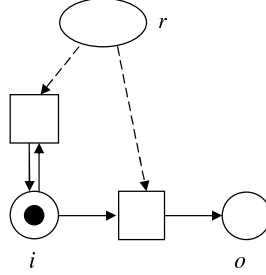


Fig. 3. RWF-net with a sound control subnet, which is not sound for any resource.

Further, we call a RWF-net N sound (without indicating any concrete resource) iff a marked RWF-net $(N, i|r)$ is sound with some initial resource r .

From the second statement of Proposition 2 and the well-known Dickson's lemma we obtain the following

Corollary 1. *For a sound RWF-net N the set $C(N)$ of all its reachable control markings is finite.*

Note 1. Since the control subnet of a sound RWF-net N is bounded, the set $C(N)$ can be effectively constructed (e. g. by constructing a coverability tree).

Definition 5. *Let N be a RWF-net, $c \in C(N)$. We define:*

1. $res(c) =_{def} \{r \in \mathcal{M}(P_r) \mid (N, c|r) \text{ is sound}\}$ — the set of all sound resources for c ;
2. $mres(c) =_{def} \{r \in res(c) \mid \nexists r' \in res(c) : r' \subset r\}$ — the set of all minimal sound resources for c .

Then from Dickson's Lemma we immediately obtain:

Proposition 3. [7] *For any sound RWF-net N and any control marking $c \in C(N)$ the set $mres(c)$ is finite.*

The questions of computability of $mres(c)$ and decidability of soundness for RWF-nets remain open. In the next section positive answers to these two questions are given for a restricted case — RWF-nets with a single resource place.

4 Soundness of 1-dim RWF-nets

Let $N = (P_c, P_r, T, F_c, F_r)$ be an RWF-net with $P_r = \{p_r\}$, i.e. with just one resource place. By 1-dim RWF-nets we denote the subclass of RWF-nets with single resources. An example of such a net is given in Fig. 4. In the following sections we consider only 1-dim RWF-nets.

If a control subnet of N is not sound, then N is also not sound. So, we suppose that the control subnet of N is sound, and hence bounded.

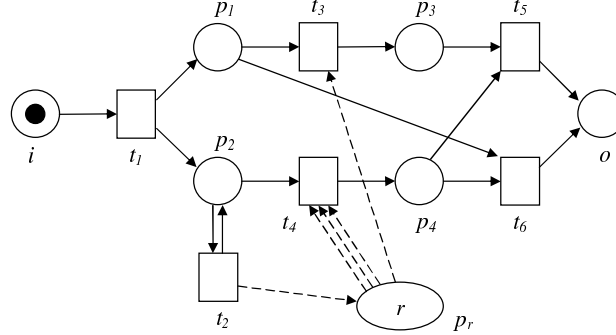


Fig. 4. 1-dim RWF-net N_1 .

4.1 Control automaton

It is easy to note that a bounded control subnet can be represented as an equivalent finite automaton (a transition system). This automaton is an oriented graph with two distinguished nodes – a source node i and a sink node o .

In this control automaton states are exactly the elements of $C(N)$, transitions — transitions of the given net N . But now it will be more convenient to consider a transition t as a pair (c_1, c_2) of control states, where $c_1 \xrightarrow{t} c_2$. Every transition t of the automaton is labeled by an integer $\delta(t)$, defining a “resource effect” of transition firing. A positive $\delta(t)$ means that the firing of t increments the marking of a (single) resource place p_r by $\delta(t)$, a negative $\delta(t)$ means that t is enabled in a state $(c|r)$ iff $r(p_r) \geq |\delta(t)|$, and that the firing of t decrements the resource by $|\delta(t)|$. Formally,

$$\delta(t) =_{\text{def}} \begin{cases} -F_r(p_r, t) & \text{for } F_r(p_r, t) > 0; \\ F_r(t, p_r) & \text{for } F_r(t, p_r) > 0. \end{cases}$$

The value $\delta(t)$ is called an *effect* of t (denoted $\text{Eff}(t)$). Note that for simplicity we exclude loops, when both $F_r(p_r, t) > 0$ and $F_r(t, p_r) > 0$; such loops can be simulated by two sequential transitions.

A *support* of t is the amount of the resource required for a firing of t . It is defined as:

$$\text{Supp}(t) =_{\text{def}} \begin{cases} 0, & \delta(t) \geq 0; \\ |\delta(t)|, & \delta(t) < 0. \end{cases}$$

Thus, a 1-dim RWF-net N can be transformed into a control automaton $\text{Aut}(N)$, which can be considered as a one-counter net (e.g. [4]) or, alternatively, a 1-dim Vector Addition System with States (VASS [12]) with a specific workflow structure: one source state, one sink state, and every state is reachable from the source state, as well as the sink is reachable from every state. Note that the control automaton $\text{Aut}(N)$ is behaviorally equivalent to N in the branching-time semantics.

Consider an example depicted in Fig. 5. This is a control automaton for the 1-dim RWF-net from Fig. 4. States are denoted by octagons, labelled with the corresponding markings of the net. Transitions are labelled with the corresponding names and effects.

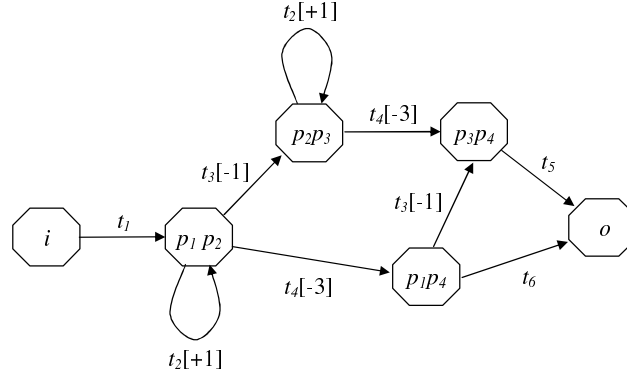


Fig. 5. Control automaton for N_1 .

A control automaton (a one-counter net) is a digraph with arcs labeled by integers. Recall some basic notion from graph theory.

A *walk* is an alternating sequence of nodes and arcs, beginning and ending with a node, where each node is incident to both the arc that precedes it and the arc that follows it in the sequence, and where the nodes that precede and follow an arc are the head and the tail of this arc.

We consider only non-empty walks, containing at least one arc.

A walk is *closed* if its first and last nodes coincide.

A *path* is a walk where no arcs are repeated (nodes may be repeated).

A *simple path* is a path where no nodes are repeated.

A *cycle* is a closed path.

A *simple cycle* is a closed path where no nodes are repeated (except the first/last one).

A walk in a control automaton corresponds to some sequence of firings in 1-dim RWF-net. Now we inductively define an effect and a support of a walk. Intuitively,

Let t be a transition and σ a walk, such that the ending node of t is the beginning node of the first transition of σ . Let $t\sigma$ denote a walk, constructed by linking t and σ . We define:

$$Eff(t\sigma) =_{\text{def}} Eff(t) + Eff(\sigma); \quad Supp(t\sigma) =_{\text{def}} Supp(t) + (Supp(\sigma) \ominus Eff(t)),$$

where \ominus denotes the truncated subtraction.

A *positive* (resp., *negative*) *walk* is a walk with a positive (resp., negative) effect. Obviously, the effect of a cycle does not depend on a choice of a starting node.

A node q is called a *positive generator*, iff there exists a simple positive path from q to q (a simple positive cycle) with a zero support.

Lemma 1. *Any simple positive cycle contains at least one generator.*

Proof. Note that without loss of generality we can consider only cycles of even lengths, having alternating positive and negative arcs. Then the proof is straightforward, by induction on the length of a cycle.

A node q is called a *negative generator*, iff there exists a simple negative path θ from q to q (a simple negative cycle), such that $Supp(\theta) = -Eff(\theta)$.

Lemma 2. *Any simple negative cycle contains at least one generator.*

Proof. Similar to the previous lemma.

4.2 Decidability of soundness for marked nets

Let $(N, i|r_0)$ be an initially marked 1-dim RWF-net. By abuse of notation we denote by N also the control automaton of N , represented as a one-counter net. Recall that $i \in C(N)$ denotes the initial control state, $r_0 \in Nat$ denotes the initial value of a counter (the single resource place), and $\mathcal{R}(N, (i|r_0))$ denotes the set of all reachable states.

Note that a marked RWF-net $(N, i|r_0)$ with a sound control subnet is not sound if and only if it does not always terminate with a final control state o for some larger initial resource $r_0 + s$:

$$\exists(c|r) \in \mathcal{R}(N, i|r_0 + s) \text{ such that } (o|s') \notin \mathcal{R}(N, c|r) \text{ for any } s' \in Nat.$$

So we consider both two kinds of possible undesirable (not properly terminating) behaviours of a Petri net, namely, deadlocks and livelocks.

Definition 6. *A state $(c|r) \in C(N) \times Nat$ is a deadlock iff $c \neq o$ and there is no transition $t \in T$ s.t. $(c|r) \xrightarrow{t} (c'|r')$ for some c', r' .*

A finite set $L \subset C(N) \times Nat$ of states is a livelock iff

1. $|L| > 1$;
2. for any $(c|r), (c'|r') \in L$ there is a finite transition sequence $\sigma \in T^*$ s.t. $(c|r) \xrightarrow{\sigma} (c'|r')$;
3. for any $(c|r) \in L$ and $t \in T$ s.t. $(c|r) \xrightarrow{t} (c''|r'')$ we have $(c''|r'') \in L$.

A livelock state is a state that belongs to some livelock.

Note that by definition $(o|r) \notin L$ for any r .

Proposition 4. *If a state $(c|r)$ is a deadlock then for any $t \in T$ s.t. $c \xrightarrow{t} c'$ we have $\text{Supp}(t) > r$.*

Proof. Straightforward.

Note that Prop. 4 implies $\delta(t) < 0$ and hence we can reformulate it:

Corollary 2. *If a state $(c|r)$ is a deadlock then:*

1. $\forall t \in T$ s.t. $c \xrightarrow{t} c'$ for some c' we have $\delta(t) < 0$;
2. $r < \min\{|\delta(t)| : c \xrightarrow{t} c' \text{ for some } c'\}$.

So deadlocks can occur (1) just for control states with only negative outgoing transitions; (2) only for a finite number of different resources – when there are not enough resources for firing any of the successor transitions.

Proposition 5. *The set of deadlock states is finite.*

Proof. The set of “potential deadlock” control states (nodes with only negative outgoing transitions) is finite. For a given “potential deadlock” control state the set of applicable deadlock states (natural numbers smaller than the smallest required resource for a successor transition) is also finite.

Hence, all deadlocks can be detected by checking control states with only negative outgoing transitions.

Now let us consider livelocks.

Proposition 6. *If $L \subset C(N) \times \text{Nat}$ is a livelock then there is a state $(c|r) \in L$ and a negative transition $t \in T$ with $c \xrightarrow{t} c'$, such that $\delta(t) < -r$.*

Proof. Straightforward, since the control subnet of RWF-net N is sound.

Proposition 7. *The set of livelocks is finite.*

Proof. First note that if $(c|r), (c|r+x) \in L$ with $x > 0$ then L is not a livelock. Indeed, in this case the transition sequence $(c|r) \xrightarrow{\sigma} (c|r+x)$ corresponds to a positive cycle, that can generate an infinite number of states — a contradiction to the finiteness of livelocks. So, every control state can occur in a given livelock at most once.

Now assume the converse: there are infinitely many livelocks. Then there are infinitely many livelocks with the same set of control states, which differ only in their resource value. Hence, this set includes a livelock with an arbitrarily large resource, and we can take a livelock with a resource big enough to reach the final state o . This implies that o belongs to the livelock — a contradiction with the definition of a livelock.

Thus, all livelocks can be easily detected by checking finite systems of states, closed under transition firings (strongly connected components) and satisfying the property from Prop. 6.

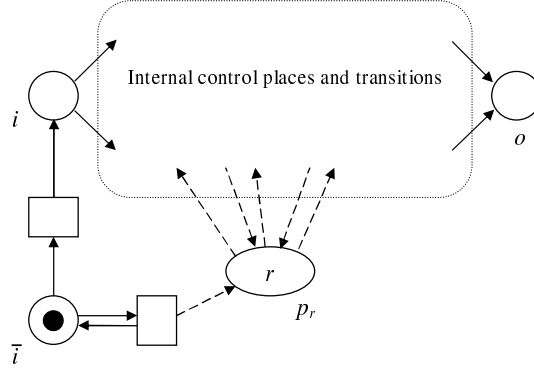


Fig. 6. Modified RWF-net \bar{N} .

Theorem 1. *Soundness is decidable for marked 1-dim RWF-nets.*

Proof. The following proof is similar to the proof of decidability of structural soundness in [14].

For a given 1-dim RWF-net N construct the modified RWF-net \bar{N} by adding a new initial place \bar{i} and two new transitions, as depicted in Fig. 6. The original 1-dim RWF-net $(N, i|r)$ is sound iff neither deadlocks nor livelocks are reachable in 1-dim RWF-net $(\bar{N}, \bar{i}|r)$ (otherwise some large enough initial resource would produce the same undesirable situation in the given net N).

Since the sets of deadlocks and livelocks are finite and computable, the problem of soundness of a marked 1-dim RWF-net can be reduced to a finite number of instances of a reachability problem for a 1-counter Petri net. This reachability problem is decidable.

4.3 Decidability of soundness for unmarked nets

Theorem 1 gives us only a semidecision procedure for soundness of a net. One can check the soundness of a given initial marking, but if the answer is negative, it is not known whether there exists a larger sound marking.

Definition 7. *An unmarked RWF-net N is called sound iff $(N, i|r)$ is sound for some resource r .*

Corollary 2 gives us only a necessary condition of a deadlock, reachable from some initial marking. Now we prove a stronger theorem, which gives a sufficient and necessary condition for existence of a soundness-violating deadlock (i.e. a deadlock that is reachable from an infinite number of different initial markings).

Theorem 2. *An unmarked 1-dim RWF-net is not sound with deadlocks iff there exist a deadlock state $(c|r)$, a negative generator q and a simple path $q \xrightarrow{\sigma} c$ such that $\text{Eff}(\sigma) \ominus \text{Supp}(\sigma) \leq r$.*

Proof. (\Leftarrow) It is sufficient to show that for any (large enough) initial resource r_0 there exists a larger initial resource $r_0 + x$, such that a deadlock is reachable from $(i|r_0 + x)$.

Consider an arbitrary (large enough) initial resource r_0 s.t.

$$(i|r_0) \xrightarrow{\tau} (q|s)$$

for some path τ and resource s (it is always possible to find such a resource since the control net is sound, and therefore any control state is reachable for some sufficiently large initial resource). Let $\theta = qc_1 \dots c_j q$ be a simple negative cycle with generator q , i.e. $Supp(\theta) = -Eff(\theta)$. Denote $z = s \bmod Supp(\theta)$ and consider a larger initial resource $r_0 + z + Supp(\sigma)$.

We have

$$\begin{aligned} & (i|r_0 + z + Supp(\sigma)) \\ & \quad \downarrow \tau \\ & (q|s + z + Supp(\sigma)) \\ & \quad \downarrow \theta^{((s+z)/Supp(\theta))} \\ & (q|Supp(\sigma)) \\ & \quad \downarrow \sigma \\ & (c|Eff(\sigma) \ominus Supp(\sigma)) \end{aligned}$$

and hence a deadlock.

(\Rightarrow) Assume the converse: the net is unsound with a deadlock, but for any given deadlock state, it is impossible to find a negative generator that satisfies the conditions in the theorem.

The number of deadlock states is finite, hence some deadlock state $(c|r)$ is reachable from an infinite number of different initial states (initial resource values).

Every transition sequence $\sigma = t_1.t_2.\dots.t_n$ from $(i|r_0)$ to $(c|r)$ corresponds to a walk σ in the control automaton graph. Since there are infinitely many deadlock-generating initial states, the set of corresponding walks is also infinite. Each of these walks can be decomposed into a sequence of alternating simple cycles and acyclic simple paths:

$$\sigma = \tau_1(\theta_1)^{k_1}\tau_2(\theta_2)^{k_2} \dots \tau_{n-1}(\theta_{n-1})^{k_{n-1}}\tau_n.$$

Note that this decomposition is not unique: $ababa$ can be considered both as $(ab)^2a$ and $a(ba)^2$. To fix ideas, we only consider “decomposition from the right to the left”, so $a(ba)^2$.

Let us show that among these walks there is a walk with a negative last cycle θ_{n-1} . Indeed, if the last cycle is positive (or neutral) with an effect x , we can consider a larger initial resource $r_0 + x * k_{n-1}$ and a shorter walk

$$\sigma' = \tau_1(\theta_1)^{k_1}\tau_2(\theta_2)^{k_2} \dots \tau_{n-1}\tau_n,$$

having the same ending — a deadlock. Now, the new walk σ' can be decomposed into simple cycles and simple paths, then the last cycle, if it is positive, can be removed by increasing the initial resource, and so on. At the end of this process we will obtain either a walk with a negative “last cycle” or a completely acyclic walk (simple path from i to c). There are only finitely many acyclic paths in the graph, but infinitely many deadlock-generating initial markings (and hence deadlock-generating walks from i to c), so we necessarily obtain a walk with a negative last cycle.

Consider such a deadlock-walk σ'' , ending with a suffix $\theta^k\tau$, where θ is a negative cycle and τ is acyclic. Let $\theta = c_1c_2\dots c_i\dots c_m c_1$, where c_i is a negative generator (from Lemma 2 such c_i always exists). The path $((c_i\dots c_m c_1)\tau)$ is simple (remember that we decompose “from the right to the left” and hence $\theta\tau$ cannot contain cycles other than θ). Since the final state of the whole walk σ'' is $(c|r)$, for any suffix ϕ of σ'' we have

$$Eff(\phi) \ominus Supp(\phi) \leq r.$$

It holds for $((c_i\dots c_m c_1)\tau)$ as well. But this is a simple path that leads from a negative generator to a deadlock control state – Q.E.D.

A result similar to Th. 2 is valid for livelocks:

Theorem 3. *An unmarked 1-dim RWF-net is not sound with livelocks iff there exist a livelock state $(c|r)$, a negative generator q and a simple path $q \xrightarrow{\sigma} c$ such that $Eff(\sigma) \ominus Supp(\sigma) \leq r$.*

Proof. Similar to Th. 2.

Corollary 3. *Soundness is decidable for unmarked 1-dim RWF-nets.*

Proof. All simple (negative) cycles can be found by Tarjan algorithm, deadlock and livelock states — by searching for states, satisfying Prop. 4 and Prop. 6 respectively. The set of simple paths is finite (and easily computable).

4.4 Computability of a minimal sound resource

Now we propose a plain (and hence, may be, not the most effective) algorithm for computing the minimal resource r such that $(N, i|r)$ is sound: one tests soundness for incremented values of r until success. Note that this method can be applied only to sound nets, while soundness of the unmarked net can be checked with the algorithm given in Cor. 3.

5 Conclusion

In this paper we have investigated the soundness property for workflow nets with one (unbounded) resource place. We have proved that soundness is decidable

for marked and unmarked nets, and that the minimal sound resource can be effectively computed.

Our decision algorithms use the reduction to the reachability problem for unbounded Petri nets and hence cannot be considered efficient. However, the inefficiency could be unavoidable, since RWF-nets are expressively rather close to ordinary Petri nets (VASS).

Further research will concern decidability of soundness for the general case of RWF-nets. It is also quite interesting to apply some alternative notions of soundness to our infinite-state workflow nets. The so-called *relaxed soundness* is of a particular interest. Relaxed soundness has been proposed as a weaker than soundness property.

Some other interesting variants of soundness property are k-soundness, generalized and structural soundness [3].

The authors would like to thank the anonymous reviewers for their helpful comments.

References

1. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
2. W.M.P. van der Aalst, K.M. van Hee. *Workflow Management: Models, Methods and Systems*, MIT Press, 2002.
3. W.M.P. van der Aalst, K.M. van Hee, A.H.M. Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, M.T. Wynn. Soundness of workflow nets: classification, decidability, and analysis, *Formal Aspects of Computing*, 23(3):333–363, Springer, 2011.
4. P. A. Abdulla, K. Čerans. Simulation is decidable for one-counter nets. In *Proc. CONCUR'98*, vol. 1466 of *Lecture Notes in Computer Science*, pages 253–268, Springer, 1998.
5. K. Barkaoui, L. Petrucci. Structural Analysis of Workflow Nets with Shared Resources. In *Proc. Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM98)*, volume 98/7 of *Computing Science Reports*, pages 82–95, Eindhoven University of Technology, 1998.
6. K. Barkaoui, R. Ben Ayed, Z. Sbaï. Workflow Soundness Verification based on Structure Theory of Petri Nets. *International Journal of Computing and Information Sciences*, Vol. 5, No. 1, 2007. P.51–61.
7. V. A. Bashkin, I. A. Lomazova. Petri Nets and resource bisimulation. *Fundamenta Informaticae*, Vol. 55, No. 2, 2003. P.101–114,
8. V. A. Bashkin, I. A. Lomazova. Resource equivalence in workflow nets. In *Proc. Concurrency, Specification and Programming, 2006*, volume 1, pages 80–91. Berlin, Humboldt Universitat zu Berlin, 2006.
9. K. van Hee, N. Sidorova, M. Voorhoeve. Generalized Soundness of Workflow Nets is Decidable. In *Proc. ICATPN 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 197–216. Springer, 2004.
10. K. van Hee, A. Serebrenik, N. Sidorova, M. Voorhoeve. Soundness of Resource-Constrained Workflow Nets. In *Proc. ICATPN 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 250–267. Springer, 2005.

11. K. van Hee, O. Oanea, A. Serebrenik, N. Sidorova, M. Voorhoeve, I.A. Lomazova. Checking Properties of Adaptive Workflow Nets. *Fundamenta Informaticae*, Vol. 79, No. 3, 2007. P.347–362.
12. J. Hopcroft, J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, **8**(2), 1979, pages 135–159.
13. N. Sidorova, C. Stahl. Soundness for Resource-Constrained Workflow Nets is Decidable. In *BPM Center Report BPM-12-09*, BPMcenter.org, 2012.
14. F. L. Tiplea, D. C. Marinescu. Structural soundness of workflow nets is decidable. *Information Processing Letters*, Vol. 96, pages 54–58. Elsevier, 2005.

6 Appendix

Proof of Proposition 1.

The proof is by contradiction. Let $(N, i|r)$ be a sound RWF-net and let the net (N_c, i) be not sound. Then there exists a marking $c \in \mathcal{R}(N_c, i)$, such that either the final marking o is not reachable from c , or $o \in c$ and $c \neq \{o\}$.

Since for the control subnet the control marking c is reachable from the initial marking i via some sequence of firings, we can always take a resource s , sufficiently large to support the same sequence of firings for $(N, i|r+s)$ and to reach the same control state c . If for the control subnet the final state was not reachable from c , then adding resource places can't make it reachable for the net with resources, i. e. for $(N, i|r+s)$, in contradiction with the soundness of $(N, i|r)$. If, otherwise, $o \in c$ and $c \neq \{o\}$, then we also obtain a contradiction with the soundness of $(N, i|r)$, since the control state c is reachable for $(N, i|r+s)$.

Proof of Proposition 2.

(1) Similarly to the proof of the Proposition 1 we can always take a sufficiently large initial resource $r+s$.

(2) Suppose this is not true. Assume that for some $c_1, c_2 \in C(N)$ we have $c_2 = c_1 + c'$ for some $c' \neq \emptyset$. From the first statement of this proposition it follows that there exist resources r_1 and r_2 s. t. RWF-nets $(N, c_1|r_1)$ and $(N, c_2|r_2)$ are sound. Then nets $(N, c_1|r_1+r_2)$ and $(N, c_2|r_1+r_2)$ are also sound. Thus the final marking $o|r'$ is reachable from the marking $c_1|r_1+r_2$, and (due to monotonicity property of Petri nets) the marking $o+c'|r'$ is reachable from the larger marking $c_2|r_1+r_2$ — contradiction with the soundness for RWF-net $(N, c_2|r_1+r_2)$.

