

Reduction Rules For Reset WorkFlow Nets

M.T. Wynn¹, H.M.W. Verbeek², W.M.P. van der Aalst^{1,2}, A.H.M. ter Hofstede¹ and
D. Edmond¹

School of Information Systems, Queensland University of Technology

GPO Box 2434, Brisbane QLD 4001, Australia.

{m.wynn,d.edmond,a.terhofstede}@qut.edu.au

Department of Technology Management, Eindhoven University of Technology

PO Box 513, NL-5600 MB Eindhoven, The Netherlands.

{h.m.w.verbeek,w.m.p.v.d.aalst}@tm.tue.nl

Abstract. When a workflow contains a large number of tasks and involves complex control flow dependencies, verification can take too much time or it may even be impossible. Reduction rules can be used to abstract from certain transitions and places in a large net and thus could cut down the size of the net used for verification. Petri nets have been proposed to model and analyse workflows and Petri nets reduction rules have been used for efficient verification of various properties of workflows, such as liveness and boundedness. Reset nets are Petri nets with reset arcs, which can remove tokens from places when a transition fires. The nature of reset arcs closely relates to the cancellation behaviour in workflows. As a result, reset nets have been proposed to formally represent workflows with cancellation behaviour, which is not easily modelled in ordinary Petri nets. Even though reduction rules exist for Petri nets, the nature of reset arcs could invalidate the transformation rules applicable to Petri nets. This motivated us to consider possible reduction rules for reset nets. In this paper, we propose a number of reduction rules for Reset Workflow Nets (RWF-nets) that are soundness preserving. These reduction rules are based on reduction rules available for Petri nets [19] and we present the necessary conditions under which these rules hold in the context of reset nets.

Keywords: Petri nets with reset arcs, reset nets, reduction rules, workflow verification, soundness property.

1 Introduction

Some have advocated the use of Petri nets for the specification of workflows among others due to the formal foundation, their graphical nature and the presence of analysis techniques [4]. Reduction rules have also been suggested to be used together with Petri nets for verification of workflows [25]. There exists a body of work concerning the verification of workflow specifications expressed as Petri nets or expressed in languages for which mappings to Petri nets have been defined [2, 3, 25]. In either case, verification boils down to examining certain properties of Petri nets.

Unfortunately, these results are not transferable to situations where languages are involved that use concepts not easily expressed through Petri nets. One such concept that is difficult to express in terms of Petri nets is cancellation regions [6]. *Cancellation*

is used to capture the interference of one task in the execution of others. If a task is within the cancellation region of another task, it may be prevented from being started or its execution may be terminated. For example, you might want to simply cancel other order processing tasks if a customer’s credit card payment did not go through. Reset nets are Petri nets with reset arcs, which can remove tokens from places when a transition fires. The nature of reset arcs closely relates to the cancellation behaviour in workflows. As a result, reset nets have been proposed to formally represent workflows with cancellation behaviour [28]. This approach allows us to leverage existing literature and techniques in the area of Petri nets and reset nets in particular [8, 11, 14–18].

We are interested in determining whether a workflow possesses the following desirable properties. Firstly, it is important to know that a workflow, when started, can complete. Secondly, it should never have tasks still running when completion is signalled. Thirdly, the workflow should not contain tasks that can never be executed. These requirements encompass the *soundness property* of a workflow specification as expressed in [4]. In [30], we have proposed a new verification approach for the soundness property in workflows with cancellation and OR-joins using Reset Workflow Nets (RWF-nets). An RWF-net is a reset net with three structural restrictions: there is exactly one source node, one sink node and every node in the graph is on a directed path from the source node to the sink node. This is to ensure that every workflow represented by an RWF-net will have a unique start place, a unique end place and it is possible to go from the start place to the end place by following a series of transitions. Using state-based analysis, we have shown that it is possible to decide the soundness property of workflows with cancellation behaviour using reset nets. The drawback of using reset nets, however, is that there are no reduction rules defined for reset nets. As a result, the analysis is time consuming for large models. Even though reduction rules exist for Petri nets, the nature of reset arcs in an RWF-net could invalidate the transformation rules applicable to Petri nets. For example, it is possible that an incorrect net that does not satisfy proper completion criterion (i.e., tokens can be left in the net when it reaches the end) becomes sound when there is a reset arc to remove the leftover token before completion. Therefore, we propose extension to the requirements for Petri net reduction rules with additional restrictions with respect to reset arcs.

In this paper, we propose a number of reduction rules for Reset Workflow Nets (RWF-nets) that are soundness preserving¹. These reduction rules for RWF-nets are inspired by the reduction rules for Petri nets [19] and free-choice Petri nets [12]. We present the necessary conditions under which these rules hold in the context of reset workflow nets. The organisation of the paper is as follows. Section 2 provides the formal foundation by introducing reset nets and Reset Workflow Nets(RWF-nets). Section 3 describes a set of reduction rules for RWF-nets together with associated proofs. Section 4 discusses the related work and section 5 concludes the paper.

¹ The bulk of this work was done while visiting Eindhoven University of Technology in close collaboration with Dr. Eric Verbeek and Professor Wil van der Aalst.

2 Preliminaries

2.1 Petri nets and Reset nets

Petri nets were originally introduced by Carl Adam Petri [21] and since then, they are widely used as mathematical models of concurrent systems for various domains [20, 12]. Numerous analysis techniques exist to determine various properties of Petri nets and its subclasses [20, 12, 19, 22, 23].

Definition 1 (Petri net [21, 20]). A Petri net is a tuple (P, T, F) where P is a (non-empty finite) set of places, T is a set of transitions, $P \cap T = \emptyset$ and $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs.

A reset net is a Petri net with special *reset arcs*, that can clear the tokens in selected places. Graphically, reset arcs are modelled as doubled-headed arrows. Figure 1 shows a transition t with all possible combinations of input, output and reset arcs. The nature of reset arcs matches closely with the concept of cancellation in workflow modelling and reset nets are proposed as a formalism for modelling workflows with cancellation.

Definition 2 (Reset net [14]). A reset net is a tuple (P, T, F, R) where (P, T, F) is a Petri net and $R : T \rightarrow \mathbb{P}(P)$ provides the reset places for the transitions².

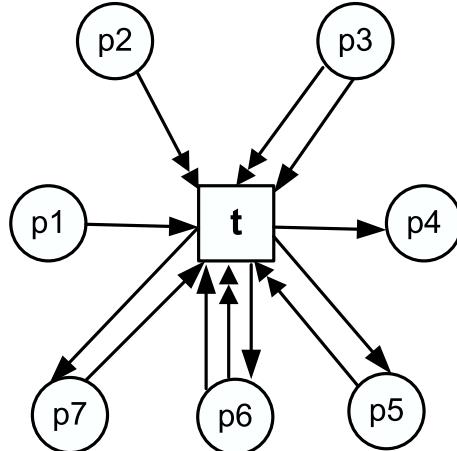


Fig. 1. An example reset net

In the remainder of the paper, when we use the function $F(x, y)$, it evaluates to 1 if $(x, y) \in F$ and 0 if $(x, y) \notin F$. We write F^+ for the transitive closure of the flow relation F and F^* for the reflexive transitive closure of F . R^{-1} is the (straightforward) inverse function of R where $R^{-1} \in P \rightarrow \mathbb{P}(T)$. The notation $R(t)$ for a transition t

² Where \mathbb{P} is a power set of P , i.e., $X \in \mathbb{P}$ if and only if $X \subseteq P$.

returns the (possibly empty) set of places that it resets. We also write $R^\leftarrow p$ for a place p , which returns the set of transitions that can reset p .

Let N be a reset net and $x \in (P \cup T)$, we use $\bullet x$ and $x\bullet$ to denote the set of inputs and outputs. If the net involved cannot be understood from the context, we explicitly include it in the notation and we write $\bullet^N x$ and $x^N\bullet$. A marking is denoted by M and, just as with ordinary Petri nets, it can be interpreted as a vector, function, and multiset over the set of places P . $M(p)$ returns the number of tokens in a place p if $p \in \text{dom}(M)$ and zero otherwise. We can use notations such as $M \leq M'$, $M + M'$, and $M \dashv M'$. $M \leq M'$ iff $\forall_{p \in P} M(p) \leq M'(p)$. $M + M'$ and $M \dashv M'$ are multisets such that $\forall_{p \in P}$: $(M + M')(p) = M(p) + M'(p)$ and $(M \dashv M')(p) = M(p) \dashv M'(p)$ ³. We represent a multiset by simply enumerating the elements, e.g., $2a + 3b + c$ is the multiset containing two a 's, three b 's and one c . If X is a set over Y , it could also be interpreted as a bag which assigns to each element a weight of 1.

The notation $\mathbf{M}(N)$ is used to represent possible markings of a reset net N .

Definition 3 ($\mathbf{M}(N)$). Let $N = (P, T, F, R)$ be a reset net, then $\mathbf{M}(N) = P \rightarrow \mathbf{N}$ is the set of possible markings.

A transition is *enabled* when there are enough tokens in its input places. Note that reset arcs do not change the requirements of enabling a transition.

Definition 4 (Enabling rule). Let N be a reset net, $t \in T$, and $M \in \mathbf{M}(N)$. Transition t is enabled at M , denoted as $M[t]$, if and only if $\forall p \in \bullet t : M(p) \geq 1$.

The concept of firing a transition t in a net N is formally defined in Definition 5 and denoted as $M \xrightarrow{N,t} M'$. If there can be no confusion regarding the net, the expression is abbreviated as $M \xrightarrow{t} M'$ and if the transition is not relevant, it is written as $M \rightarrow M'$.

Definition 5 (Forward firing). Let $N = (P, T, F, R)$ be a reset net, $t \in T$ and $M, M' \in \mathbf{M}(N)$.

$$M \xrightarrow{N,t} M' \Leftrightarrow M[t] \wedge \\ M'(p) = \begin{cases} M(p) - F(p, t) + F(t, p) & \text{if } p \in P \setminus R(t) \\ F(t, p) & \text{if } p \in R(t). \end{cases}$$

It is possible to fire a sequence of transitions from a given marking in a reset net resulting in a new marking using the forward firing rule defined above. This sequence of transitions is represented as an occurrence sequence.

Definition 6 (Occurrence sequence). Let $N = (P, T, F, R)$ be a reset net and $M, M_1, \dots, M_n \in \mathbf{M}(N)$. If $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} M_n$ are firing occurrences then $\sigma = t_1 t_2 \dots t_n$ is an occurrence sequence leading from M to M_n and it is written as $M \xrightarrow{\sigma} M_n$.

We now define the concepts of reachability and coverability of markings from a given marking in a reset net. A marking M' is reachable from another marking M in a reset net, if there is an occurrence sequence leading from M to M' .

³ For any natural numbers a, b : $a \dashv b$ is defined as $\max(a - b, 0)$.

Definition 7 (Reachability). Let $N = (P, T, F, R)$ be a reset net and $M, M' \in \mathbf{M}(N)$. M' is reachable in N from M , denoted $M \xrightarrow{N} M'$, if there exists an occurrence sequence σ such that $M \xrightarrow{\sigma} M'$.

The reachability set is the minimal set of markings that can be reached from a given marking M in a reset net after firing all possible occurrence sequences.

Definition 8 (Reachability set). Let $N = (P, T, F, R)$ be a reset net and $M \in \mathbf{M}(N)$. The reachability set of the marked net (N, M) , denoted $N[M]$, is the minimal set that satisfies the following conditions:

1. $M \in N[M]$ and
2. if transition $t \in T$ and markings $M_1, M_2 \in \mathbf{M}(N)$ exist such that $M_1 \in N[M]$ and $M_1 \xrightarrow{N, t} M_2$, then $M_2 \in N[M]$.

Definition 9 (Directed labelled graph). A directed labelled graph $G = (V, E)$ over label set \mathcal{L} consists of a set of nodes V and a set of labelled edges $E \subseteq V \times \mathcal{L} \times V$.

The reachability graph is a directed labelled graph where the elements of the reachability set form the nodes and the tuple consisting of a source marking that enables a transition, the transition and the target marking that is reached by firing the transition form the edges. The graph can be used to determine the possible states of a reset net from an initial marking.

Definition 10 (Reachability graph). Let $N = (P, T, F, R)$ be a reset net and $M \in \mathbf{M}(N)$. The directed labelled graph $G = (V, E)$ with label set $\mathcal{L} = T$ is the reachability graph of the marked net (N, M) iff

1. $V = N[M]$ and
2. for any transition $t \in T$ and markings $M_1, M_2 \in \mathbf{M}(N) : M_1 \xrightarrow{t} M_2 \Leftrightarrow (M_1, t, M_2) \in E$.

Liveness, boundedness and safeness are defined as in previous work [20, 19]. Liveness, boundedness and safeness can be determined from the reachability graph.

Definition 11 (Liveness, boundedness, safeness [20, 19]). A transition is live if it can be enabled from every reachable marking. A place is safe if it never contains more than one token at the same time. A place is k -bounded if it will never contain more than k tokens. A place is bounded if it is k -bounded for some k .

If all places in a reset net are bounded, the reset net is also bounded and hence, it is possible to generate a finite reachability set. If a place is unbounded, the reachability set contains an infinite number of states (*an infinite state space*). In such cases, reachability of a marking cannot be determined but coverability can be determined. Coverability is a relaxed notion that can handle unbounded behaviour. A marking M_2 is said to be *coverable* from another marking M_1 in a reset net if there is a reachable marking M' from M_1 such that M' is bigger than or equal to M_2 .

Definition 12 (Coverability). Let $N = (P, T, F, R)$ be a reset net and $M_1, M_2 \in \mathbf{M}(N)$. M_2 is coverable from M_1 in N , if there exists a marking M' such that $M' \in N[M_1]$ and $M' \geq M_2$.

We conclude this section with the notion of *Backward firing* that is used to generate coverable markings for a reset net by firing transitions backwards.

Definition 13 (Backward firing [29]). Let (P, T, F, R) be a reset net and $M, M' \in \mathbf{M}(N)$. $M' \dashrightarrow^t M$ iff it is possible to fire a transition t backwards starting from M and resulting in M' .

$$M' \dashrightarrow^t M \Leftrightarrow M[R(t)] \leq t \bullet [R(t)] \wedge \\ M'(p) = \begin{cases} (M(p) - F(t, p)) + F(p, t) & \text{if } p \in P \setminus R(t) \\ F(p, t) & \text{if } p \in R(t). \end{cases}$$

For places that are not reset places, the number of tokens in M' is determined by the number of tokens in M for p and the production and consumption of tokens. If a place is an output place of t and not a reset place, one token is removed from $M(p)$ if $M(p) > 0$. If a place is an input place of t and not a reset place, one token is added to $M(p)$. For any reset place p , $M(p) \leq F(t, p)$ because it is emptied when firing and then $F(t, p)$ tokens are added. We do not require $M(p) = F(t, p)$ for a reset place p because the aim is coverability and not reachability. M' , i.e., the marking before (forward) firing t , should *at least* contain the *minimal* number of tokens required for enabling t and resulting in a marking of *at least* M . Therefore, only $F(p, t)$ tokens are assumed to be present in a reset place p .

2.2 Reset WorkFlow nets (RWF-nets)

This section discusses the formalisation of workflow models using Petri nets. A WF-net is defined as a Petri net with the following structural restrictions. There is exactly one begin place and exactly one end place. Moreover, every node in the graph is on a directed path from the begin place to the end place.

Definition 14 (WF-net [3, 25]). Let $N = (P, T, F)$ be a Petri net. The net N is a WF-net iff the following three conditions hold:

1. there exists exactly one $i \in P$ such that $\bullet i = \emptyset$, and
2. there exists exactly one $o \in P$ such that $o \bullet = \emptyset$, and
3. for all $n \in P \cup T$: $(i, n) \in F^*$ and $(n, o) \in F^*$.

The notion of a Reset WorkFlow net (RWF-net) is introduced to represent workflows with cancellation features. We define Reset WorkFlow nets (RWF-nets) which are reset nets with the same structural restrictions as WF-nets.

Definition 15 (RWF-net [27]). Let $N = (P, T, F, R)$ be a reset net. The net N is an RWF-net iff (P, T, F) is a WF-net.

In an RWF-net, there is an input place i and an output place o and an initial marking M_i and an end marking M_o is defined as follows:

Definition 16 (Initial marking and End marking). Let $N = (P, T, F, R)$ be an RWF-net and i, o be the input and output places of the net. The initial marking of N is denoted as M_i and it represents a marking where there is a token in the input place i (i.e., $M_i = i$). Similarly, the end marking of N is denoted as M_o and it represents a marking where there is a token in the output place o (i.e., $M_o = o$).

A WF-net is an RWF-net iff R is empty (for all $t \in T : R(t) = \emptyset$). Thus (P, T, F) suffices (we may omit R).

The *soundness* definition for an RWF-net is based on the soundness definition from [7] for WF-nets. An RWF-net is sound if and only if it satisfies the three criteria: option to complete, proper completion and no dead transitions.

Definition 17 (Soundness [27]). Let $N = (P, T, F, R)$ be an RWF-net and M_i, M_o be the initial and end markings. N is sound iff:

1. *option to complete*: for every marking M reachable from M_i , there exists an occurrence sequence leading from M to M_o , i.e., for all $M \in N[M_i] : M_o \in N[M]$, and
2. *proper completion*: the marking M_o is the only marking reachable from M_i with at least one token in place o , i.e., for all $M \in N[M_i] : M \geq M_o \Rightarrow M = M_o$, and
3. *no dead transitions*: for every transition $t \in T$, there is a marking M reachable from M_i such that $M[t]$, i.e., for all $t \in T$ there exists an $M \in N[M_i]$ such that $M[t]$.

3 Reduction Rules for RWF-nets

Reduction rules can be used to abstract from certain transitions and places in a large net and thus could cut down the size of the net used for verification. As a result, the verification process can be performed more efficiently. Furthermore, reduction rules can highlight potential problems within a net. After applying reduction rules, a correct net can potentially be reduced to a trivial net (just a task with one input and output place) thus making the consequent verification process unnecessary. Those parts of the net that cannot be reduced could indicate problems during execution and close attention should be paid to them. When a net has reset arcs, it cannot be reduced to a trivial net even though it is correct. This is because elements with reset arcs can be combined but cannot be entirely abstracted. In any case, reduction rules enable verification to be performed on a smaller net.

The style of this section is taken from [12]. For sake of clarity, we have taken a two-step approach: first the reduction rule for WF-nets, then the extension for RWF-nets.

We will prove that reduction rules for WF-nets and RWF-nets are soundness preserving. The soundness of WF-nets has been shown to correspond to boundedness and liveness properties of the short-circuited WF-net [2]. Therefore, if a reduction rule for a WF-net preserves boundedness and liveness, then it also preserves soundness. We will show that a reduction rule for a WF-net is boundedness and liveness preserving and hence, it is also soundness preserving. However, soundness of RWF-nets does not correspond to boundedness and liveness. It is possible that an unbounded RWF-net is sound due to the presence of reset arcs. In Figure 2, place q is an unbounded place and therefore, the net is unbounded. Transition c resets both preceding places when it fires. As a result, it is not possible for tokens to be left in either p or q when the net completes. Hence, the net is sound and we cannot prove that a reduction rule for RWF-nets preserves soundness by showing that it preserves boundedness and liveness. Therefore, we will show that reduction rules for RWF-nets preserve soundness by proving that they

preserve occurrence sequences and hence, preserve the three criteria for soundness: the option to complete, proper completion, and no dead transitions.

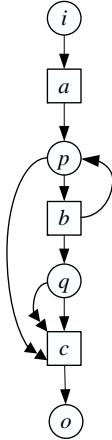


Fig. 2. An example of an unbounded RWF-net which is sound.

3.1 Fusion of series places

In this subsection, we first present the *Fusion of Series Places Rule for WF-nets* (ϕ_{FSP}) and then extend the rule for RWF-nets (ϕ_{FSP}^R) by proposing additional requirements for reset arcs. The ϕ_{FSP} rule is based on the Fusion of Series Places rule for Petri nets by Murata [19]. The rule allows for the merging of two sequential places p and q with one transition t in between them into a single place r . The rule requires that there is only one output arc from p to t , exactly one input p and one output q for t , and that there are no direct connections between inputs of p and inputs of q . The last requirement ensures that there will only be one arc connecting inputs of p in the original net to the new place r in the reduced net (no weighted arcs). Furthermore, the rule is not applicable to places that are either an input place i or an output place o of the net. See the example in Figure 3 for an application of the ϕ_{FSP} rule. The white parts in the figure are the parts being considered in the reduction step. Places p and q have been merged into a new place r in the right net.

Definition 18 (Fusion of Series Places Rule for WF-nets: ϕ_{FSP}). Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{FSP}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, two places $p, q \in P_1 \setminus \{i, o\}$, a transition $t \in T_1$, and a place $r \in P_2 \setminus P_1$ such that:

Conditions on N_1 :

1. $\bullet t = \{p\}$ (p is the only input of t)

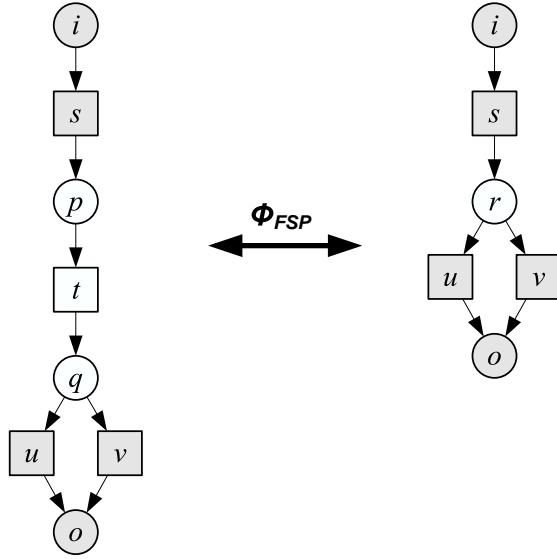


Fig. 3. Reduction of a WF-net using the ϕ_{FSP} rule

2. $t\bullet = \{q\}$ (q is the only output of t)
3. $p\bullet = \{t\}$ (t is the only output of p)
4. $\bullet p \cap \bullet q = \emptyset$ (any input of p is not an input of q and vice versa)

Construction of N_2 :

5. $P_2 = (P_1 \setminus \{p, q\}) \cup \{r\}$
6. $T_2 = T_1 \setminus \{t\}$
7. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup ((({}^N \bullet p \cup {}^N \bullet q) \setminus \{t\}) \times \{r\}) \cup (\{r\} \times q^N \bullet)$

Theorem 1 (The ϕ_{FSP} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{FSP}$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_{FSP} rule is boundedness and liveness preserving [19]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The *Fusion of Series Places Rule for RWF-nets* (ϕ_{FSP}^R) extends the ϕ_{FSP} rule by introducing reset arcs and strengthening the conditions. The rule also allows for the merging of two sequential places p and q with one transition t in between them into a single place r . Figure 4 visualises the ϕ_{FSP}^R rule. The first additional requirement is that the transition t should not have any reset arcs. See Figure 5 for a counter example where t has reset arcs. Transition t can reset place u in the left net but this behaviour is ignored in the right net. Transition sequence xt leads to a deadlock as t will remove a token from u when it fires, and u does not exist in the right net. As a result, the left net is not sound whereas the right net is. The second additional requirement is that the two places must be reset by the same set of transitions (if any). If p and q are not reset

places, then it is clear that the rule holds. If a transition resets place p , it must also resets place q as we are interested in merging these two places. See Figure 6 for a counter example: transition sequence $xtyz$ leads to an unsound net on the left (a leftover token in q), whereas the right net is sound. If all requirements for the ϕ_{FST}^R rule are satisfied, places p and q are merged into a new place r which takes on the same reset arcs as p and q .

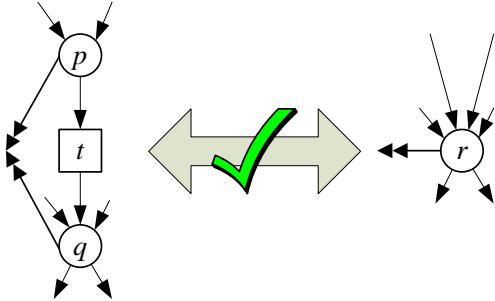


Fig. 4. Fusion of Series Places Rule for RWF-nets: ϕ_{FSP}^R

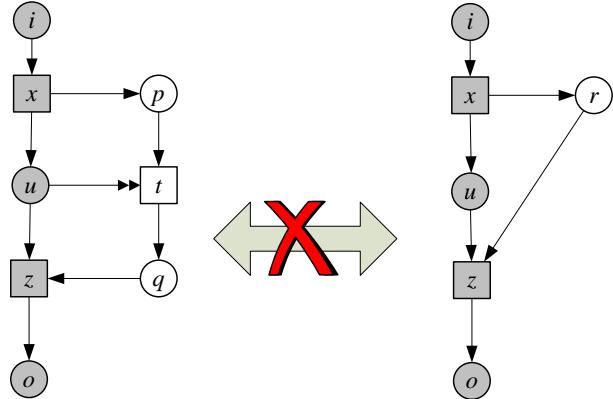


Fig. 5. Transition t resets place u . (Note that the model on the left is not sound while the one on the right is.)

Definition 19 (Fusion of Series Places Rule for RWF-nets: ϕ_{FSP}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in$

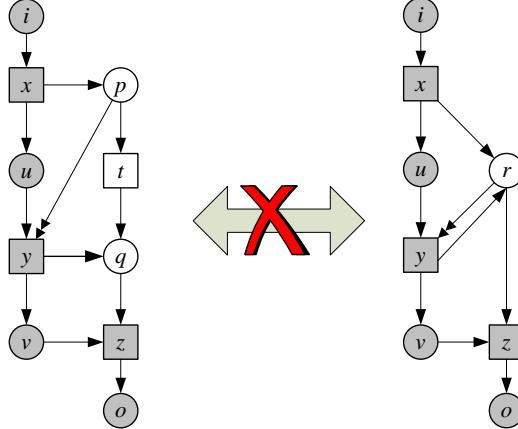


Fig. 6. Place p is a reset place and place q is not a reset place. (Note that the model on the left is not sound while the one on the right is.)

ϕ_{FSP}^R if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, two places $p, q \in P_1 \setminus \{i, o\}$, a transition $t \in T_1$, and a place $r \in P_2 \setminus P_1$ such that:

Extension of the ϕ_{FSP} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{FSP}}$ (Note that, by definition, the i, o, p, q, t , and r mentioned in this definition have to coincide with the i, o, p, q, t , and r as mentioned in the definition of ϕ_{FSP} .)

Conditions on R_1 :

2. $R_1(t) = \emptyset$ (t does not reset)
3. $R_1^\leftarrow(p) = R_1^\leftarrow(q)$ (p and q are being reset by the same transitions)

Construction of R_2 :

4. $R_2 = \{(z, R_1(z) \cap P_2) | z \in T_2 \cap T_1\} \oplus \{(z, (R_1(z) \cap P_2) \cup \{r\}) | z \in R_1^\leftarrow(p)\}$ ⁴.

Next, we show that the ϕ_{FSP}^R rule is soundness preserving. We first present two lemmas that show that occurrence sequences in N_1 and N_2 correspond to one another. These lemmas are then used to prove that the ϕ_{FSP}^R rule preserves the three criteria of soundness: the option to complete, proper completion, and dead transitions.

Lemma 1 (Under the ϕ_{FSP}^R rule, sequences in N_1 correspond to sequences in N_2). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FSP}}^R$, let $\sigma_1 \in T_1^*$ and $M_1 \in \mathbf{M}(N_1)$ be such that $i \xrightarrow{N_1, \sigma_1} M_1$, and $\sigma_2 = \alpha(\sigma_1)$, where $\alpha \in T_1^* \rightarrow T_2^*$ is defined as follows:

⁴ \oplus represents function override where $f : A \rightarrow B$, $f' = f \oplus \{(a, b)\}$ returns $f' = f \cup \{(a, b)\}$ if $a \notin \text{dom } f$ and $f' = f \setminus \{(a, f(a))\} \cup \{(a, b)\}$ if $a \in \text{dom } f$.

- $\alpha(\epsilon) = \epsilon$,
- $\alpha(t\sigma) = \alpha(\sigma)$, and
- $\alpha(x\sigma) = x\alpha(\sigma)$, where $x \in T_1 \setminus \{t\}$.

Thus, α removes every occurrence of t from the sequence. Then $i \xrightarrow{N_2, \sigma_2} M_2$, where $M_2(r) = M_1(p) + M_1(q)$ and $M_2(x) = M_1(x)$ for every $x \in P_2 \setminus \{r\}$.

Proof By induction on the length of σ_1 .

Base Assume $\sigma_1 = \epsilon$. Clearly, $i \xrightarrow{N_1, \epsilon} i$ and also $i \xrightarrow{N_2, \epsilon} i$.

Step Assume the theorem holds for some σ_1 , let M_1 be such that $i \xrightarrow{N_1, \sigma_1} M_1$, and let M_2 be such that $i \xrightarrow{N_2, \alpha(\sigma_1)} M_2$. We prove that it also holds if we extend σ_1 by one transition.

- First, assume that we extend σ by t . It is easy to see that this extension does not have any effect on $\alpha(\sigma_1)$. Therefore, we need to prove that firing t does not violate the where-clause (i.e., $M_2(r) = M_1(p) + M_1(q)$ and $M_2(x) = M_1(x)$ for every $x \in P_2 \setminus \{r\}$). As t moves only one token from p to q and does not reset any place, this is straightforward.
- Second, assume that we extend σ by an $x \in P_1 \setminus \{t\}$. First, we need to prove that $M_2[x]$ in N_2 . As r contains at least as many tokens as q , and $M_2(x) = M_1(x)$ for every $x \in P_2 \setminus \{r\}$, we conclude that this is indeed the case. Next, we need to prove that firing x in both nets does not violate the where-clause. This is straightforward as well, as any transition that adds a token to p also adds a token to r and any transition that removes a token from q also removes a token from r , and the remaining transitions are identical.

■

Lemma 2 (Under the ϕ_{FSP}^R rule, sequences in N_2 correspond to sequences in N_1). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{FSP}^R$, let $\sigma_2 \in T_2^*$ and $M_2 \in \mathbf{M}(N_2)$ be such that $i \xrightarrow{N_2, \sigma_2} M_2$, and $\sigma_1 = \beta(\sigma_2)$, where $\beta \in T_2^* \rightarrow T_1^*$ is defined as follows:

- $\beta(\epsilon) = \epsilon$,
- $\beta(x\sigma) = xt\beta(\sigma)$, if $p \in x \xrightarrow{N_1} \bullet$, and
- $\beta(x\sigma) = x\beta(\sigma)$, if $p \notin x \xrightarrow{N_1} \bullet$.

Thus, β introduces an extra t whenever place p is marked. As a result, place p is unmarked as soon as possible. Then $i \xrightarrow{N_1, \sigma_1} M_1$, where $M_1(p) = 0$, $M_1(q) = M_2(r)$ and $M_1(x) = M_2(x)$ for every $x \in P_1 \setminus \{p, q\}$.

Proof By induction on the length of σ_2 .

Base Assume $\sigma_2 = \epsilon$. Clearly, $i \xrightarrow{N_2, \epsilon} i$ and also $i \xrightarrow{N_1, \epsilon} i$.

Step Assume the theorem holds for some σ_2 , let M_2 be such that $i \xrightarrow{N_2, \sigma_2} M_2$, and let M_1 be such that $i \xrightarrow{N_1, \beta(\sigma_2)} M_1$. We prove that it also holds if we extend σ_2 by one transition.

- First, assume that we extend σ by an x such that $p \in x^{N_1}$. It is obvious that $M_1[x]$ in N_1 , and that afterwards t is also enabled. Furthermore, both x and t do not violate the where-clause (i.e., where $M_1(p) = 0$, $M_1(q) = M_2(r)$ and $M_1(x) = M_2(x)$ for every $x \in P_1 \setminus \{p, q\}$).
- Second, assume that we extend σ by an x such that $p \notin x^{N_1}$. Again it is obvious that $M_1[x]$ in N_1 , and that x does not violate the where clause.

■

Theorem 2 (The ϕ_{FSP}^R rule preserves the option to complete). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{FSP}^R$. Then N_1 has the option to complete iff N_2 has the option to complete.

Proof Let α and β be as defined in lemmas 1 and 2.

\Rightarrow Assume that N_2 does not have the option to complete, that is, there exists some $M_2 \in N_2[i]$ such that $o \notin N_2[M_2]$. Thus, there exists a $\sigma_2 \in T_2^*$ such that $i \xrightarrow{N_2, \sigma_2} M_2$ but no $\sigma'_2 \in T_2^*$ exists such that $M_2 \xrightarrow{N_2, \sigma'_2} o$. As a result, $i \xrightarrow{N_1, \beta(\sigma_2)} M_1$, for a well-defined M_1 . Now assume that N_1 does have the option to complete. As a result, there exists a σ_1 such that $i \xrightarrow{N_1, \beta(\sigma_2)\sigma_1} o$. But then $i \xrightarrow{N_2, \alpha(\beta(\sigma_2)\sigma_1)} o$, which contradicts the assumption that no $\sigma'_2 \in T_2^*$ exists such that $M_2 \xrightarrow{N_2, \sigma'_2} o$. Thus, N_1 does not have the option to complete.

\Leftarrow Similar to \Rightarrow .

■

Theorem 3 (The ϕ_{FSP}^R rule preserves proper completion). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{FSP}^R$. Then N_1 has proper completion iff N_2 has proper completion.

Proof Let α and β be as defined in lemmas 1 and 2.

\Rightarrow Assume that N_2 does not have proper completion, that is there exists some $M_2 \in N_2[i]$ such that $M_2 > o$. Thus, there exists a $\sigma_2 \in T_2^*$ such that $i \xrightarrow{N_2, \sigma_2} M_2$. Then $i \xrightarrow{N_1, \beta(\sigma_2)} M_1$ such that $M_1 > o$, and N_1 does not have proper completion.

\Leftarrow Similar to \Rightarrow .

■

Theorem 4 (The ϕ_{FSP}^R rule preserves dead transitions). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{FSP}^R$. Then N_1 contains dead transitions iff N_2 contains dead transitions.

Proof Let α and β be as defined in lemmas 1 and 2.

\Rightarrow Assume that N_2 contains no dead transitions, that is, for every $t_2 \in T_2$ there exists some $M_2 \in N_2[i]$ such that $M_2 \geq_{\bullet}^{N_2} t$. Let t_2 be an arbitrary transition from T_2 , and let $M_2 \in N_2[i]$ be such that $M_2 \geq_{\bullet}^{N_2} t_2$. Then there exists a $\sigma_2 \in T_2^*$ such that $i \xrightarrow{N_2, \sigma_2} M_2$. As a result, $i \xrightarrow{N_1, \beta(\sigma_2)} M_1$ and $M_1 \geq_{\bullet}^{N_1} t_2$. As $T_2 = T_1 \cup \{t\}$, only transition t can still be dead. However, t can only be dead if all transitions that mark p are dead, and these transitions exist (as $p \neq i$).

\Leftarrow Assume that N_1 contains no dead transitions, that is, for every $t_1 \in T_1$ there exists some $M_1 \in N_1[i]$ such that $M_1 \geq_{\bullet}^{N_1} t$. Let t_1 be an arbitrary transition from T_1 excluding t , and let $M_1 \in N_1[i]$ be such that $M_1 \geq_{\bullet}^{N_1} t_1$. Then there exists a $\sigma_1 \in T_1^*$ such that $i \xrightarrow{N_1, \sigma_1} M_1$. As a result, $i \xrightarrow{N_2, \alpha(\sigma_1)} M_2$ and $M_2 \geq_{\bullet}^{N_2} t_1$. Thus, N_2 contains no dead transitions.

■

Theorem 5 (The ϕ_{FSP}^R rule is soundness preserving). *Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FSP}}^R$. N_1 is sound iff N_2 is sound.*

Proof Follows from theorems 2, 3, and 4. ■

3.2 Fusion of series transitions

In this subsection, we first present *Fusion of Series Transitions Rule for WF-nets* (ϕ_{FST}) and then extend the rule for RWF-nets (ϕ_{FST}^R) by proposing additional requirements for reset arcs. The ϕ_{FST} rule is based on the Fusion of Series Transitions rule for Petri nets by Murata [19]. The rule allows for the merging of two sequential transitions t and u with one place p in between these two transitions into only one transition v . The rule requires that there is only one input t and output u for the place p , p is the only input of u , and there are no direct connections between outputs of t and outputs of u . The last requirement ensures that there will only be one arc connecting the new transition v to outputs of t in the reduced net. See the example in Figure 7 for an application of the ϕ_{FST} rule. Transitions t and u have been merged into a new transition v in the right net. Note that transitions u and x cannot be merged as x has two input places (q and r).

Definition 20 (Fusion of Series Transitions Rule for WF-nets: ϕ_{FST}).

Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{\text{FST}}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, a place $p \in P_1$, two transitions $t, u \in T_1$, and a transition $v \in T_2 \setminus T_1$ such that:

Conditions on N_1 :

1. $\bullet p = \{t\}$ (t is the only input of p)
2. $p\bullet = \{u\}$ (u is the only output of p)
3. $\bullet u = \{p\}$ (p is the only input of u)
4. $t\bullet \cap u\bullet = \emptyset$ (any output of t is not an output of u and vice versa)

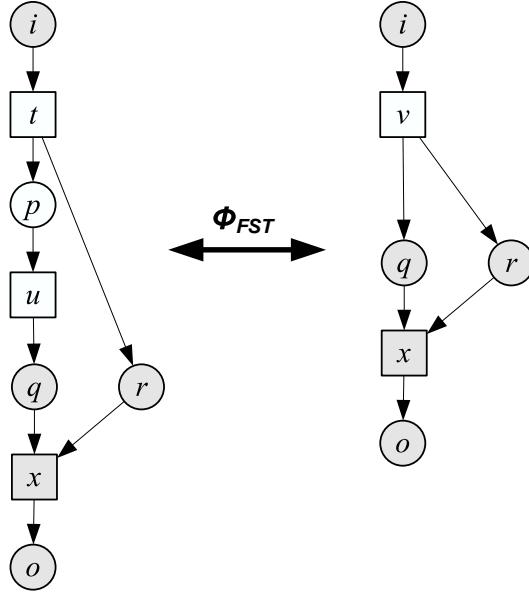


Fig. 7. Reduction of a WF-net using the ϕ_{FST} rule

Construction of N_2 :

5. $P_2 = P_1 \setminus \{p\}$
6. $T_2 = (T_1 \setminus \{t, u\}) \cup \{v\}$
7. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup (^{N_1} t \times \{v\}) \cup (\{v\} \times ((t^{N_1} \bullet \cup u^{N_1}) \setminus \{p\}))$

Theorem 6 (The ϕ_{FST} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{FST}$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_{FST} rule is boundedness and liveness preserving [19]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The *Fusion of Series Transitions Rule for RWF-nets* (ϕ_{FST}^R) extends the ϕ_{FST} rule by introducing reset arcs. The rule also allows for the merging of two sequential transitions t and u with one place p in between them into a single transition v . Figure 8 visualises the ϕ_{FST}^R rule. Additional requirements (required to allow for reset arcs) are that place p and output places of u should not be source of any reset arcs and transition u should not reset any place. The rule allows reset arcs from transition t and these arcs will be assigned to the new transition v in the reduced net. Figure 9 shows a counter example where p is a reset place: transition sequence tx leads to a deadlock, which does not exist in the other net. Figure 10 shows a counter example where transition u has reset arcs: transition sequence tu leads to a deadlock, which does not exist in the other net. Figure 11 shows a counter example where the postset of u contains a reset place: transition sequence txu results in two tokens in place r , which is not possible in the right net. As a result, the left net is not sound whereas the right net is.

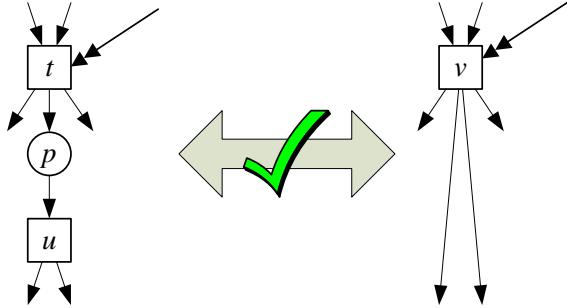


Fig. 8. Fusion of Series Transitions Rule for RWF-nets: ϕ_{FST}^R

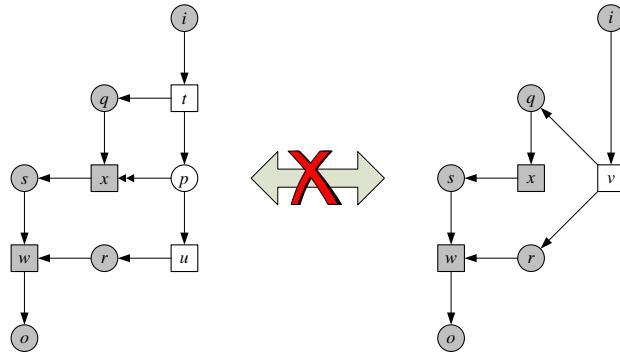


Fig. 9. Place p is a reset place. (Note that the model on the left is not sound while the one on the right is.)

Definition 21 (Fusion of Series Transitions Rule for RWF-nets: ϕ_{FST}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_{\text{FST}}^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, a place $p \in P_1$, two transitions $t, u \in T_1$, and a transition $v \in T_2 \setminus T_1$ such that:

Extension of the ϕ_{FST} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{FST}}$ (Note that, by definition, the t, u, v , and p mentioned in this definition have to coincide with the t, u, v , and p as mentioned in the definition of ϕ_{FST} .)

Conditions on R_1 :

2. $R_1^-(p) = \emptyset$ (p is not a reset place)
3. $R_1(u) = \emptyset$ (u does not reset)
4. for all $q \in u \bullet: R_1^-(q) = \emptyset$ (any output place of u is not a reset place)

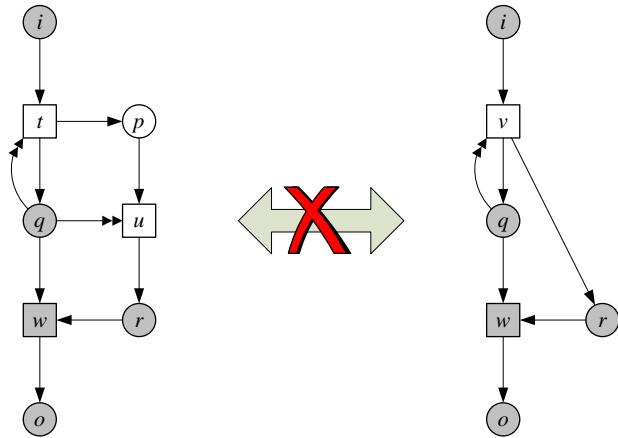


Fig. 10. Transition u resets a place that is effected by transition t . (Note that the model on the left is not sound while the one on the right is.)

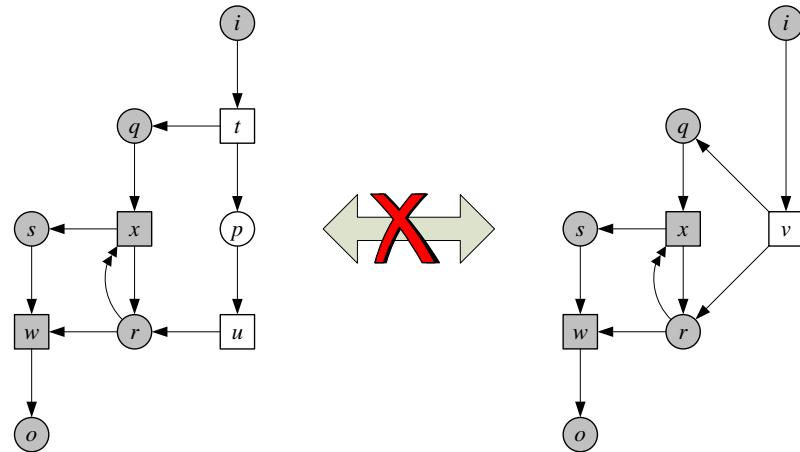


Fig. 11. The postset of transition u contains a reset place. (Note that the model on the left is not sound while the one on the right is.)

Construction of R_2 :

$$5. R_2 = \{(z, R_1(z)) | z \in T_2 \cap T_1\} \cup \{(v, R_1(t))\}$$

We now present two lemmas that show that occurrence sequences in N_1 and N_2 correspond to one another. These lemmas are then used to prove that the ϕ_{FST}^R rule preserves the three criteria of soundness: the option to complete, proper completion, and dead transitions.

Lemma 3 (Under the ϕ_{FST}^R rule, sequences in N_1 correspond to sequences in N_2). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$, let $\sigma_1 \in T_1^*$ and $M_1 \in \mathbf{M}(N_1)$ be such that $i \xrightarrow{N_1, \sigma_1} M_1$, and $\sigma_2 = \alpha(\sigma_1)$, where $\alpha \in T_1^* \rightarrow T_2^*$ is defined as follows:

- $\alpha(\epsilon) = \epsilon$,
- $\alpha(t\sigma) = v\alpha(\sigma)$,
- $\alpha(u\sigma) = \alpha(\sigma)$, and
- $\alpha(x\sigma) = x\alpha(\sigma)$, where $x \in T_1 \setminus \{t, u\}$.

Thus, α removes every occurrence of u from the sequence, and replaces every occurrence of t with v . Then $i \xrightarrow{N_2, \sigma_2} M_2$, where $M_2(x) = M_1(x) + M_1(p)$ for every $x \in v^{N_2}$ and $M_2(x) = M_1(x)$ for every $x \notin v^{N_2}$.

Proof By induction on the length of σ_1 .

Base Assume $\sigma_1 = \epsilon$. Clearly, $i \xrightarrow{N_1, \epsilon} i$ and also $i \xrightarrow{N_2, \epsilon} i$.

Step Assume the theorem holds for some σ_1 , let M_1 be such that $i \xrightarrow{N_1, \sigma_1} M_1$, and let M_2 be such that $i \xrightarrow{N_2, \alpha(\sigma_1)} M_2$. We prove that it also holds if we extend σ_1 by one transition.

- First, assume that we extend σ by t . t and v have the same preset, thus we can extend $\alpha(\sigma)$ by v . t adds a token to place p , whereas v adds tokens to its postset, which does not violate the where-clause.
- Second, assume that we extend σ by u . It is obvious that v does not violate the where-clause.
- Third, assume that we extend σ by x , where $x \in P_1 \setminus \{t, u\}$. As all places in N_2 contains at least as many tokens as their counterparts in N_1 (the where-clause), we know that x is enabled in N_2 as well. Furthermore, x does not violate the where-clause.

■

Lemma 4 (Under the ϕ_{FST}^R rule, sequences in N_2 correspond to sequences in N_1). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$, let $\sigma_2 \in T_2^*$ and $M_2 \in \mathbf{M}(N_2)$ be such that $i \xrightarrow{N_2, \sigma_2} M_2$, and $\sigma_1 = \beta(\sigma_2)$, where $\beta \in T_2^* \rightarrow T_1^*$ is defined as follows:

- $\beta(\epsilon) = \epsilon$,

- $\beta(v\sigma) = tu\beta(\sigma)$, and
- $\beta(x\sigma) = x\beta(\sigma)$, if $x \in T_2 \setminus \{v\}$.

Thus, β replaces every occurrence of v with tu . Then $i \xrightarrow{N_1, \sigma_1} M_1$, where $M_1(p) = 0$ and $M_1(x) = M_2(x)$ for every $x \in P_1 \setminus \{p\}$.

Proof By induction on the length of σ_2 .

Base Assume $\sigma_2 = \epsilon$. Clearly, $i \xrightarrow{N_2, \epsilon} i$ and also $i \xrightarrow{N_1, \epsilon} i$.

Step Assume the theorem holds for some σ_2 , let M_2 be such that $i \xrightarrow{N_2, \sigma_2} M_2$, and let M_1 be such that $i \xrightarrow{N_1, \beta(\sigma_2)} M_1$. We prove that it also holds if we extend σ_2 by one transition.

- First, assume that we extend σ by v . It is obvious that $M_1[t]$ in N_1 , and that afterwards u is also enabled. Furthermore, the combination tu and v does not violate the where-clause.
- Second, assume that we extend σ by x such that $x \in T_2 \setminus \{v\}$. Again it is obvious that $M_1[x]$ in N_1 , and that x does not violate the where clause.

■

Theorem 7 (The ϕ_{FST}^R rule preserves the option to complete). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$. Then N_1 has the option to complete iff N_2 has the option to complete.

Proof Let α and β be as defined in lemmas 3 and 4. The proof is similar to the proof of Theorem 2, but with different α and β . ■

Theorem 8 (The ϕ_{FST}^R rule preserves proper completion). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$. Then N_1 has proper completion iff N_2 has proper completion.

Proof Let α and β be as defined in lemmas 3 and 4. The proof is similar to the proof of Theorem 3, but with different α and β . ■

Theorem 9 (The ϕ_{FST}^R rule preserves dead transitions). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$. Then N_1 has proper completion iff N_2 has proper completion.

Proof Let α and β be as defined in lemmas 3 and 4. The proof is similar to the proof of Theorem 4, but with different α and β . ■

Theorem 10 (The ϕ_{FST}^R rule is soundness preserving). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FST}}^R$. N_1 is sound iff N_2 is sound.

Proof Follows from theorems 7, 8, and 9. ■

3.3 Fusion of parallel places

In this subsection, we first present *Fusion of Parallel Places Rule for WF-nets* (ϕ_{FPP}) and then extend the rule for RWF-nets (ϕ_{FPP}^R) by proposing additional requirements for reset arcs. The ϕ_{FPP} rule is a generalization of the Fusion of Parallel Places rule for Petri nets by Murata [19]. The rule allows for the merging of multiple places (at least two) with the same inputs and outputs into a single place q . See the example in Figure 12 for an application of the ϕ_{FPP} rule. Places p_1 and p_2 have the same input set $\{t_1, t_2, t_3\}$ and the same output set $\{x_1, x_2\}$. The reduced net contains a new place q that has the same input and output sets as places p_1 and p_2 .

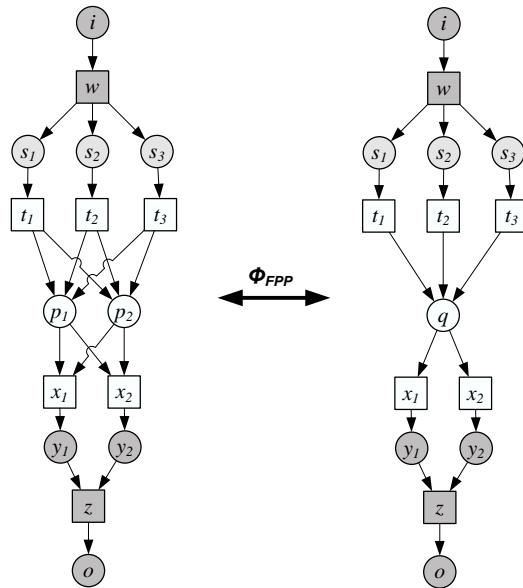


Fig. 12. Reduction of a WF-net using the ϕ_{FPP} rule

Definition 22 (Fusion of Parallel Places Rule for WF-nets: ϕ_{FPP}).

Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{FPP}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q \subseteq P_1$ where $|Q| \geq 2$ and a place $q \in P_2 \setminus P_1$ such that:

Conditions on N_1 :

1. for all $px, py \in Q : \bullet px = \bullet py$ (input transitions for all places in Q are identical)
2. for all $px, py \in Q : px\bullet = py\bullet$ (output transitions for all places in Q are identical)

Construction of N_2 :

3. $P_2 = (P_1 \setminus Q) \cup \{q\}$
4. $T_2 = T_1$
5. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup (^{N_1} p \times \{q\}) \cup (\{q\} \times p^{N_1})$ where $p \in Q$

Theorem 11 (The ϕ_{FPP} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{\text{FPP}}$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_{FPP} rule is boundedness and liveness preserving [19]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The *Fusion of Parallel Places Rule for RWF-nets* (ϕ_{FPP}^R) extends the ϕ_{FPP} rule by introducing reset arcs. The rule also allows for the merging of places in Q (i.e., p_1 to p_L) that have the same inputs and outputs into a single place q . The additional requirement is that these places are reset by the same set of transitions. If none of the places are reset places, then it is obvious that the rule holds. If one is a reset place, then other places should also be reset by the same set of transitions. Figure 13 visualises the ϕ_{FPP}^R rule. As all places in $Q = \{p_1, \dots, p_L\}$ have the same input, output and reset arcs, these identical places can be merged into a single place while preserving the soundness property. Place q in the reduced net has the same input, output and reset arcs as any place in Q .

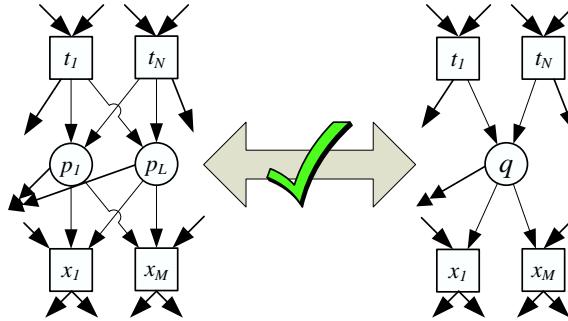


Fig. 13. Fusion of Parallel Places Rule for RWF-nets: ϕ_{FPP}^R

Definition 23 (Fusion of Parallel Places Rule for RWF-nets: ϕ_{FPP}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_{\text{FPP}}^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q \subseteq P_1$ where $|Q| \geq 2$ and a place $q \in P_2 \setminus P_1$ such that:

Extension of the ϕ_{FPP} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{FPP}}$ (Note that, by definition, the i , o , Q , and q mentioned in this definition have to coincide with the i , o , Q , and q as mentioned in the definition of ϕ_{FPP} .)

Condition on R_1 :

2. for all $px, py \in Q : R_1^{\leftarrow}(px) = R_1^{\leftarrow}(py)$ (all places in Q are being reset by the same transitions)

Construction of R_2 :

3. $R_2 = \{(z, R_1(z) \cap P_2) | z \in T_2 \cap T_1\} \oplus \{(z, (R_1(z) \cap P_2) \cup \{q\}) | z \in R_1^{\leftarrow}(p) \wedge p \in Q\}$

Theorem 12 (The ϕ_{FPP}^R rule is soundness preserving). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{FPP}^R$. N_1 is sound iff N_2 is sound.

Proof It is easy to see that the state spaces of both nets are identical, except that the markings differ: A marking in the state space of N_1 contains places Q , and every one of them contains n tokens, whereas a marking in the state space of N_2 contains one place q which contains n tokens. ■

3.4 Fusion of parallel transitions

In this subsection, we first present *Fusion of Parallel Transitions Rule for WF-nets* (ϕ_{FPT}) and then extend the rule for RWF-nets (ϕ_{FPT}^R) by proposing additional requirements for reset arcs. The ϕ_{FPT} rule is a generalization of the Fusion of Parallel Transitions rule for Petri nets by Murata [19]. The rule allows for the merging of multiple transitions (at least two) that have the same inputs and outputs into a single transition. See the example in Figure 14 for an application of the ϕ_{FPT} rule. Transitions t_1 and t_2 have the same input set $\{p_1, p_2, p_3\}$ and the same output set $\{x_1, x_2\}$. The reduced net contains a new transition v that has the same input and output sets as t_1 and t_2 .

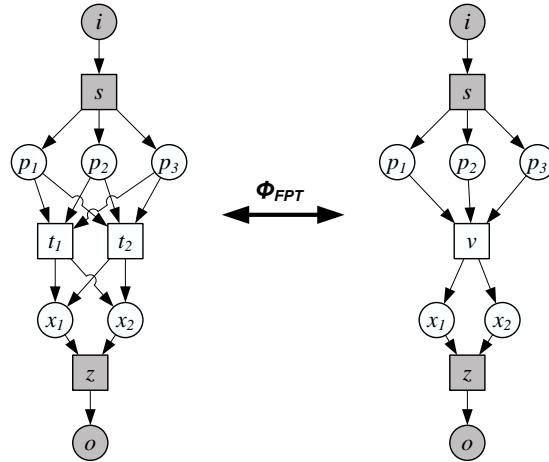


Fig. 14. Reduction of a WF-net using the ϕ_{FPT} rule

Definition 24 (Fusion of Parallel Transitions Rule for WF-nets: ϕ_{FPT}). Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{\text{FPT}}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, transitions $V \subseteq T_1$ where $|V| \geq 2$, and a transition $v \in T_2 \setminus T_1$ such that:

Conditions on N_1 :

1. for all $tx, ty \in V : \bullet tx = \bullet ty$ (input places for all transitions in V are identical)
2. for all $tx, ty \in V : tx\bullet = ty\bullet$ (output places for all transitions in V are identical)

Construction of N_2 :

3. $P_2 = P_1$
4. $T_2 = (T_1 \setminus V) \cup \{v\}$
5. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup (\{v\} \times {}^{N_1} t) \cup (t {}^{N_1} \times \{v\})$ where $t \in V$

Theorem 13 (The ϕ_{FPT} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{\text{FPT}}$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_{FPT} rule is boundedness and liveness preserving [19]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The *Fusion of Parallel Transitions Rule for RWF-nets* (ϕ_{FPT}^R) extends the ϕ_{FPT} rule by introducing reset arcs. The rule allows for the merging of transitions V (i.e., t_1 to t_L) that have the same inputs and outputs into a single transition v . The additional requirement is that these transitions should reset the same set of places (if any). If no transition has reset arcs, then it is obvious that the rule holds. If one transition resets a place, then other transitions must also reset the same place. Figure 15 visualises the ϕ_{FPT}^R rule. As all transitions in $V = \{t_1, \dots, t_L\}$ now have the same input, output and reset arcs, these identical transitions could be merged into a single transition while preserving the soundness property. Transition v in the reduced net has the same input, output and reset arcs as any transition $t \in V$.

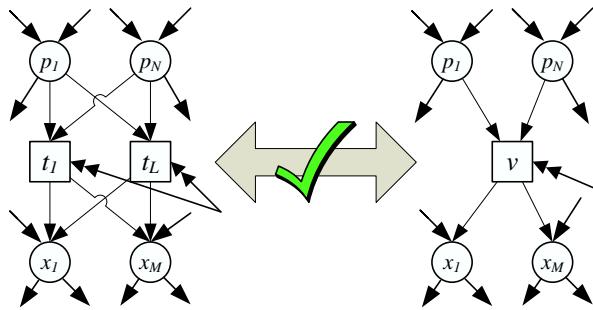


Fig. 15. Fusion of Parallel Transitions Rule for RWF-nets: ϕ_{FPT}^R

Definition 25 (Fusion of Parallel Transitions Rule for RWF-nets: ϕ_{FPT}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_{\text{FPT}}^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, transitions $V \subseteq T_1$ where $|V| \geq 2$, and a transition $v \in T_2 \setminus T_1$ such that:

Extension of the ϕ_{FPT} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{FPT}}$ (Note that, by definition, the i , o , V , and v mentioned in this definition have to coincide with the i , o , V , and v as mentioned in the definition of ϕ_{FPT} .)

Condition on R_1 :

2. for all $tx, ty \in V : R_1(tx) = R_1(ty)$ (all transitions in V reset the same places)

Construction of R_2 :

3. $R_2 = \{(z, R_1(z)) | z \in T_2 \cap T_1\} \cup \{(v, R_1(z)) | z \in V\}$

Theorem 14 (The ϕ_{FPT}^R rule is soundness preserving.). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FPT}}^R$. N_1 is sound iff N_2 is sound.

Proof It is obvious that the state spaces of both nets are identical, except that some edges differ: where the state space of N_1 contains edges for transitions t_1 up to t_L , the state space of N_2 only contains one edge for transition v . ■

3.5 Elimination of self-loop transitions

In this subsection, we first present *Elimination of Self-Loop Transitions Rule for WF-nets* (ϕ_{ELT}) and then extend the rule for RWF-nets (ϕ_{ELT}^R) by proposing additional requirements for reset arcs. The ϕ_{ELT} rule is based on the Elimination of Self-Loop Transitions rule for Petri nets by Murata [19]. The rule allows the removal of a self-loop transition. A self-loop transition is one that has one input place which is also the only output place of the transition. See the example in Figure 16 for an application of the ϕ_{ELT} rule. Transition t has been abstracted from in the reduced net as p is the only input place and the only output place of t .

Definition 26 (Elimination of Self-Loop Transitions for WF-nets: ϕ_{ELT}). Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{\text{ELT}}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, a place $p \in P_1 \cap P_2$, and a transition $t \in T_1$ such that:

Conditions on N_1 :

1. $\bullet t = \{p\}$ (p is the only input place of t)
2. $t\bullet = \{p\}$ (p is the only output place of t)

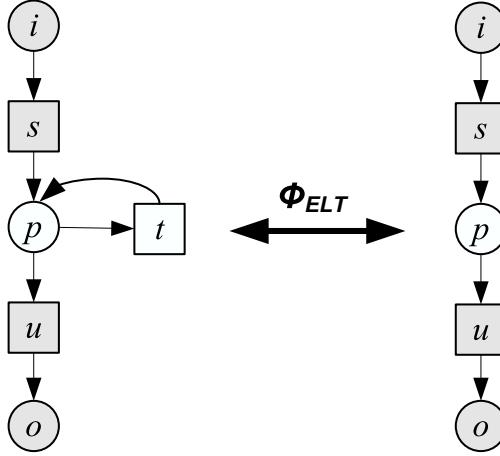


Fig. 16. Reduction of a WF-net using the ϕ_{ELT} rule

Construction of N_2 :

3. $P_2 = P_1$
4. $T_2 = T_1 \setminus \{t\}$
5. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2)))$

Theorem 15 (The ϕ_{ELT} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{ELT}$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_{ELT} rule is boundedness and liveness preserving [19]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The *Elimination of Self-Loop Transitions Rule for RWF-nets* (ϕ_{ELT}^R) extends the ϕ_{ELT} rule by introducing reset arcs. The rule also allows removal of a transition t which has a single place as its input and its output. The additional requirement is that transition t has no reset arcs. Figure 17 visualises the ϕ_{ELT}^R rule.



Fig. 17. Elimination of Self-Loop Transitions Rule for RWF-nets: ϕ_{ELT}^R

Definition 27 (Elimination of Self-Loop Transitions Rule for RWF-nets: ϕ_{ELT}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_{\text{ELT}}^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, a place $p \in P_1 \cap P_2$, and a transition $t \in T_1$ such that:

Extension of the ϕ_{ELT} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{ELT}}$ (Note that, by definition, the i , o , t , and p mentioned in this definition have to coincide with the i , o , t , and p as mentioned in the definition of ϕ_{ELT} .)

Condition on R_1 :

2. $R_1(t) = \emptyset$ (t does not reset)

Construction of R_2 :

3. $R_2 = \{(z, R_1(z)) | z \in T_2 \cap T_1\}$

Theorem 16 (The ϕ_{ELT}^R rule is soundness preserving). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{ELT}}^R$. N_1 is sound iff N_2 is sound.

Proof It is obvious that the state spaces of both nets are identical, except that the state space of N_1 contains additional self-edges. Furthermore, it is clear that t can only be dead if every transition that marks p is dead. Therefore, removing t preserves dead transitions. ■

We have presented five reduction rules for RWF-nets based on the reduction rules defined by Murata [19]. We have omitted the sixth rule, “Elimination of Self-Loop Places” as this rule requires a place to be marked in an initial marking of a net. For WF-nets and RWF-nets, this is not possible as the input place i is the only place that could be marked in an initial marking. By definition, i cannot be a self-loop (i.e., it cannot have any incoming arcs $\bullet i = \emptyset$) and therefore, this rule is not applicable to WF-nets and RWF-nets. In addition to the “Murata rules” we also present some additional rules. These rules turn out to be particularly useful when reducing YAWL models.

3.6 Fusion of equivalent subnets

In this subsection, we first present *Fusion of Equivalent Subnets Rule for WF-nets* (ϕ_{FES}) and then extend the rule for RWF-nets (ϕ_{FES}^R) by proposing additional requirements for reset arcs. The ϕ_{FES} rule allows removal of multiple identical subnets by replacing them with only one subnet. The rule requires that pairs of transitions have the same input and output places. See the example in Figure 18 for an application of the ϕ_{FES} rule. The set of transitions V_1 has been merged into V_3 . The set of transitions V_2 has been merged into v_4 , and places in Q_2 have been merged into one place r . Note that the name of the rule may be a bit misleading. This rule only applies to subnets having the structure shown in Figure 18. The reason that this rule has been added is that it is very effective in reducing YAWL models (cf. [31]).

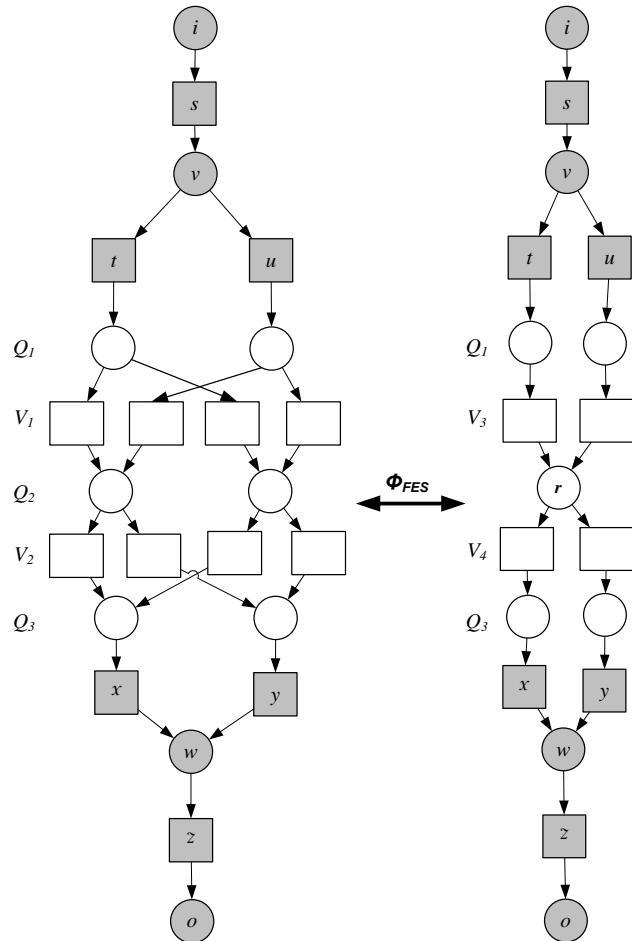


Fig. 18. Reduction of a WF-net using the ϕ_{FES} rule

Definition 28 (Fusion of Equivalent Subnets Rule for WF-nets: ϕ_{FES}). Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_{\text{FES}}$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q_1, Q_3 \subseteq P_1 \cap P_2$, $Q_2 \subseteq P_1$ where $|Q_2| \geq 2$, $r \in P_2 \setminus P_1$, transitions $V_1, V_2 \subseteq T_1$, and $V_3, V_4 \subseteq T_2 \setminus T_1$ such that:

Conditions on N_1 :

1. $V_1 = \{v_1^{q_1, q_2} \mid q_1 \in Q_1 \wedge q_2 \in Q_2\}$ (every transition of V_1 is of the form $v_1^{q_1, q_2}$)
2. $V_2 = \{v_2^{q_2, q_3} \mid q_2 \in Q_2 \wedge q_3 \in Q_3\}$ (every transition of V_2 is of the form $v_2^{q_2, q_3}$)
3. for all $p \in Q_2 : \bullet^{\overset{N_1}{\bullet}} p \subseteq V_1 \wedge p \bullet^{\overset{N_1}{\bullet}} \subseteq V_2$ (preset and postset of all places in Q_2 are from V_1 and V_2 respectively)
4. for all $v_1^{q_1, q_2} \in V_1 : \overset{N_1}{\bullet} v_1^{q_1, q_2} = \{q_1\} \wedge v_1^{q_1, q_2} \bullet^{\overset{N_1}{\bullet}} = \{q_2\}$ (preset of $v_1^{q_1, q_2}$ is q_1 and postset is q_2)
5. for all $v_2^{q_2, q_3} \in V_2 : \overset{N_1}{\bullet} v_2^{q_2, q_3} = \{q_2\} \wedge v_2^{q_2, q_3} \bullet^{\overset{N_1}{\bullet}} = \{q_3\}$ (preset of $v_2^{q_2, q_3}$ is q_2 and postset is q_3)

Construction of N_2 :

6. $P_2 = (P_1 \setminus Q_2) \cup \{r\}$
7. $T_2 = (T_1 \setminus (V_1 \cup V_2)) \cup (V_3 \cup V_4)$ where $V_3 = \{v_3^{q_1, r} \mid q_1 \in Q_1\}$ and $V_4 = \{v_4^{r, q_3} \mid q_3 \in Q_3\}$
8. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup (V_3 \times \{r\}) \cup (\{r\} \times V_4) \cup \{(q_1, v_3^{q_1, r}) \mid q_1 \in Q_1 \wedge v_3^{q_1, r} \in V_3\} \cup \{(v_4^{r, q_3}, q_3) \mid q_3 \in Q_3 \wedge v_4^{r, q_3} \in V_4\}$

Theorem 17 (The ϕ_{FES} rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_{\text{FES}}$. N_1 is sound iff N_2 is sound.

Proof The state spaces of both nets are comparable, such that where the state space of N_1 contains edges for transitions in V_1 , the state space of N_2 only contains edges for transitions in V_3 . Similarly, the set of transitions V_2 in N_1 is now V_4 in N_2 . The set of places Q_2 has been replaced with r . ■

The *Fusion of Equivalent Subnets Rule for RWF-nets* (ϕ_{FES}^R) extends the ϕ_{FES} rule by introducing reset arcs. The rule allows the removal of multiple identical subnets by replacing them with only one subnet. Additional requirements are that all places in Q_2 are reset by the same set of transitions and all transition pairs in V_1 and V_3 also reset the same places. Figure 19 visualises the ϕ_{FES}^R rule.

Definition 29 (Fusion of Equivalent Subnets Rule for RWF-nets: ϕ_{FES}^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_{\text{FES}}^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q_1, Q_3 \subseteq P_1 \cap P_2$, $Q_2 \subseteq P_1$ where $|Q_2| \geq 2$, $r \in P_2 \setminus P_1$, transitions $V_1, V_2 \subseteq T_1$, and $V_3, V_4 \subseteq T_2 \setminus T_1$ such that:

Extension of the ϕ_{FES} rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_{\text{FES}}$ (Note that, by definition, the $i, o, Q_1, Q_2, Q_3, V_1, V_2, V_3$, and V_4 mentioned in this definition have to coincide with the $i, o, Q_1, Q_2, Q_3, V_1, V_2, V_3$, and V_4 as mentioned in the definition of ϕ_{FES} .)

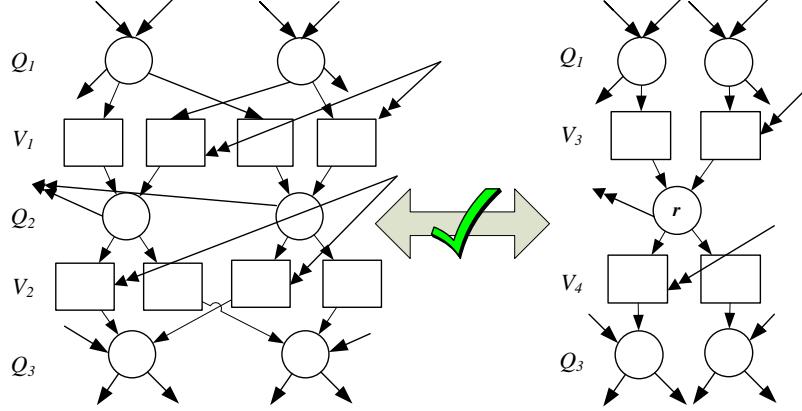


Fig. 19. Fusion of Equivalent Subnets Rule for RWF-nets: ϕ_{FES}^R

Condition on R_1 :

2. for all $q_1 \in Q_1$ and $q_2, q'_2 \in Q_2 : R(v_1^{q_1, q_2}) = R(v_1^{q_1, q'_2})$ (transitions in V_1 that have the same input set should also have the same reset arcs)
3. for all $q_3 \in Q_3$ and $q_2, q'_2 \in Q_2 : R(v_1^{q_2, q_3}) = R(v_1^{q'_2, q_3})$ (transitions in V_2 that have the same output set should also have the same reset arcs)
4. for all $q_2, q'_2 \in Q_2, R_1^\leftarrow(q_2) = R_1^\leftarrow(q'_2)$ (places in Q_2 are reset by the same set of transitions)

Construction of R_2 :

5. $R_2 = \{(z, R_1(z) \cap P_2) | z \in T_2 \cap T_1\}$
 $\oplus \{(z, (R_1(z) \cap P_2 \cup \{r\})) | z \in R_1^\leftarrow(q_2) \wedge q_2 \in Q_2\}$
 $\cup \{(v_3^{q_1, r}, R_1(v_1^{q_1, q_2}) \cap P_2) | q_1 \in Q_1 \wedge q_2 \in Q_2\}$
 $\cup \{(v_4^{r, q_3}, R_1(v_2^{q_2, q_3}) \cap P_2) | q_2 \in Q_2 \wedge q_3 \in Q_3\}$

Theorem 18 (The ϕ_{FES}^R rule is soundness preserving). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_{\text{FES}}^R$. N_1 is sound iff N_2 is sound.

Proof The proof is similar to the one for the ϕ_{FES} rule. The state spaces of both nets are comparable, such that where the state space of N_1 contains edges for transitions in V_1 , the state space of N_2 only contains edges for transitions in V_3 . Similarly, the set of transitions V_2 in N_1 is now V_4 in N_2 . The set of places Q_2 has been replaced with r . Additional requirements for reset arcs ensure that the transitions can be abstracted. ■

3.7 Abstraction

In this subsection, we first present *Abstraction Rule for WF-nets* (ϕ_A) and then extend the rule for RWF-nets (ϕ_A^R) by proposing additional requirements for reset arcs. The

ϕ_A rule is based on the Abstraction rule for Petri nets from Desel and Esparza [12]. The rule allows the removal of a place s and a transition t , where s is the only input of t , t is the only output of s and there is no direct connection between the inputs of s with the outputs of t . See the example in Figure 20 for an application of the ϕ_A rule. The reduced net on the right abstracts from place s and transition t and provides direct connections between the inputs of s and the outputs of t .

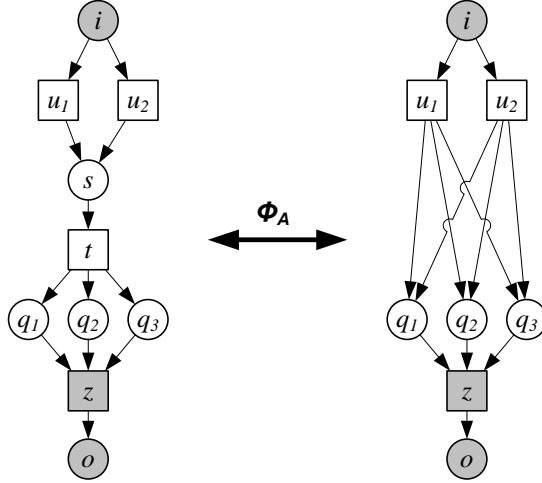


Fig. 20. Reduction of a WF-net using the ϕ_A rule

Definition 30 (Abstraction Rule for WF-nets: ϕ_A). Let N_1 and N_2 be two WF-nets, where $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$. $(N_1, N_2) \in \phi_A$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q \subseteq P_1 \cap P_2$, a place $s \in P_1 \setminus Q$, transitions $U \subseteq T_1 \cap T_2$, and a transition $t \in T_1 \setminus U$ such that:

Conditions on N_1 :

1. $\bullet t = \{s\}$ (s is the only input of t)
2. $s \bullet = \{t\}$ (t is the only output of s)
3. $\bullet s = U$ (transitions in U are input transitions for s)
4. $t \bullet = Q$ (transitions in Q are output transitions for t)
5. $(\bullet s \times t \bullet) \cap F = \emptyset$ (any input of s is not connected to an output of t and vice versa)

Construction of N_2 :

6. $P_2 = P_1 \setminus \{s\}$
7. $T_2 = T_1 \setminus \{t\}$
8. $F_2 = (F_1 \cap ((P_2 \times T_2) \cup (T_2 \times P_2))) \cup (\overset{N_1}{\bullet} s \times t \overset{N_2}{\bullet})$

Theorem 19 (The ϕ_A rule is soundness preserving). Let N_1 and N_2 be two WF-nets such that $(N_1, N_2) \in \phi_A$. Then N_1 is sound iff N_2 is sound.

Proof The ϕ_A rule is boundedness and liveness preserving as shown by Desel and Esparza [12]. Soundness of a WF-net corresponds to boundedness and liveness of the short-circuited WF-net [2]. ■

The Abstraction Rule for RWF-nets (ϕ_A^R) extends the ϕ_A rule by introducing reset arcs. The rule allows for the removal of a place s and a transition t , where s is the only input of t , t is the only output of s and there is no direct connection between the inputs for s with the outputs for t . Additional requirements are that transition t does not reset any place, place s is not reset by any transition, and outputs for t are not reset by any transition. Input transitions for place s can have reset arcs. Figure 21 visualises the ϕ_A^R rule.

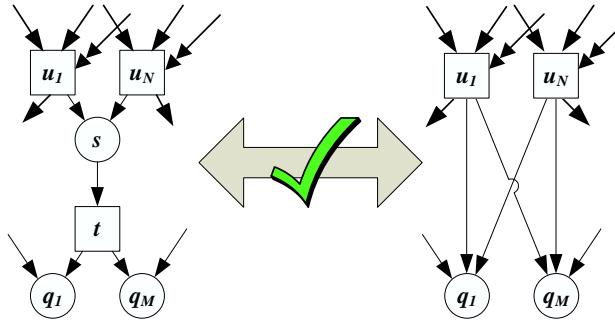


Fig. 21. Abstraction Rule for RWF-nets: ϕ_A^R

Definition 31 (Abstraction Rule for RWF-nets: ϕ_A^R). Let N_1 and N_2 be two RWF-nets, where $N_1 = (P_1, T_1, F_1, R_1)$ and $N_2 = (P_2, T_2, F_2, R_2)$. $(N_1, N_2) \in \phi_A^R$ if there exists an input place $i \in P_1 \cap P_2$, an output place $o \in P_1 \cap P_2$, places $Q \subseteq P_1 \cap P_2$, a place $s \in P_1 \setminus Q$, transitions $U \subseteq T_1 \cap T_2$, and a transition $t \in T_1 \setminus U$ such that:

Extension of the ϕ_A^R rule:

1. $((P_1, T_1, F_1), (P_2, T_2, F_2)) \in \phi_A$ (Note that, by definition, the i , o , s , t , Q , and U mentioned in this definition have to coincide with i , o , s , t , Q , and U as mentioned in the definition of ϕ_A .)

Conditions on R_1 :

2. $R_1^\leftarrow(s) = \emptyset$ (s is not a reset place)
3. $R_1(t) = \emptyset$ (t does not reset)
4. for all $q \in t\bullet : R_1^\leftarrow(q) = \emptyset$ (all output places for t are not reset places)

Construction of R_2 :

$$5. R_2 = \{(z, R_1(z) \cap P_2) | z \in T_2 \cap T_1\}$$

Theorem 20 (The ϕ_A^R rule is soundness preserving). Let N_1 and N_2 be two RWF-nets such that $(N_1, N_2) \in \phi_A^R$. N_1 is sound iff N_2 is sound.

Proof This rule is quite close to the ϕ_{FST}^R rule (i.e., the fusion of two subsequent transitions), except that it rule allows for s (p for the ϕ_{FST}^R rule) to have multiple inputs. Using the ϕ_{FST}^R rule, the proof is quite simple. It is obvious that we can replace s and t by s_1, \dots, s_N and t_1, \dots, t_N in such a way that $\bullet s_i = \{u_i\}$, $s_i \bullet = \{t_i\}$, $\bullet t_i = \{s_i\}$, and $t_i \bullet = Q$ while preserving soundness. Next, we can use the ϕ_{FST}^R rule to reduce every s_i and t_i . Figure 22 visualises the proof of the soundness preserving property of the ϕ_A^R rule. ■

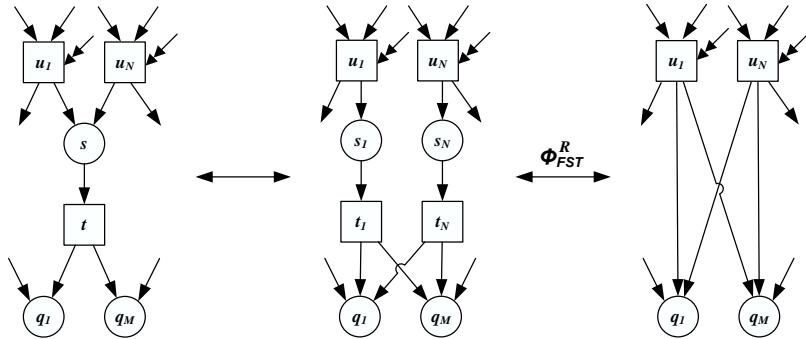


Fig. 22. Proof sketch for the ϕ_A^R rule

The other two linear dependency rules described by Desel and Esparza [12] to remove nonnegative linearly dependent places and nonnegative linearly dependent transitions are only applicable to free-choice nets. The rules are said to be not strongly sound for arbitrary nets (i.e., N is well-formed if and only if N' is well-formed) [12]. Hence, they cannot be used for WF-nets and RWF-nets.

4 Related work

A number of authors have investigated reduction rules for Petri nets and for various subclasses of Petri nets. In [9] and [10], Berthelot presents a set of reduction rules for general Petri nets. He proposes transformations on places and transitions that preserve language, deadlock-freeness, 1-liveness and liveness for place/transition systems. They include transformation on places such as structurally redundant places, double places

and equivalent places and fusion of transitions such as post-fusion, pre-fusion and lateral fusion. In [19], six reduction rules are presented for Petri nets. In [12], a set of reduction rules are proposed for free-choice Petri nets while preserving well-formedness. They include the abstraction rule, linear dependency rules for a non-negative linearly dependent place and for a non-negative linearly dependent transition. In [24], authors extends the reduction rules given by Berthelot for Time Petri nets. Six reduction rules that preserve correctness for EPCs including reduction rules for trivial constructs, simple splits and joins, similar splits and joins, XOR loop and optional OR-loop are proposed [13]. Some reduction rules presented for EPCs such as reduction rules for simple splits and joins and reduction rules for similar splits and joins are related to reduction rules that we have defined for WF-nets. However, these reduction rules do not take cancellation into account.

Reduction rules have been suggested to be used together with Petri nets for the verification of workflows (cf. Chapter 4 of the book by van der Aalst and van Hee [5]). In [26], the authors present how to decide relaxed soundness property of workflows with cancellation and OR-joins using invariants. We follow a similar approach with a set of reduction rules for workflow nets with cancellation regions and OR-joins using reset nets.

5 Conclusion

An important correctness notion for a workflow net is the soundness property. A workflow net is sound if it has the option to complete, proper completion, and no dead transitions. Verification can be used to detect whether a net satisfies the soundness property. When a workflow language supports cancellation behaviour, verification becomes time consuming, challenging and sometimes not even possible. In our previous work [30], we proposed a new verification technique for workflows with cancellation and OR-joins using reset nets and reachability analysis. We found that state based analysis for large nets can be time consuming and this has motivated us to consider possible reduction rules for such nets while preserving the soundness property.

A reduction rule can transform a large net into a smaller and simple net while preserving certain interesting properties and it is usually applied before verification to reduce the complexity and to prevent state space explosion. There are no reduction rules defined for reset nets in the literature. In this paper, we continue our work on verification of workflows with cancellation by exploring possible reduction rules for RWF-nets. We have presented a set of reduction rules for WF-nets and RWF-nets that are soundness preserving. These rules are based on existing reduction rules for Petri nets and free-choice nets [19, 12] and they have been extended and generalised as necessary. We have also provided detailed proofs for these reduction rules. We have also realised these reduction rules as part of the verification feature in the workflow language YAWL [1].

References

1. Yawl home page. <http://www.yawl.fit.qut.edu.au/> accessed on 21 May 2006.

2. W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azéma and G. Balbo, editors, *Proceedings of Application and Theory of Petri Nets*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426, Toulouse, France, 1997. Springer-Verlag.
3. 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.
4. W.M.P. van der Aalst. Workflow Verification: Finding Control-Flow Errors using Petri Net-Based Techniques. In W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors, *Proceedings of Business Process Management: Models, Techniques and Empirical Studies*, volume 1806 of *Lecture Notes in Computer Science*, page 161. Springer-Verlag, 2000.
5. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods and Systems*. MIT press, Cambridge, MA, 2004.
6. W.M.P. van der Aalst and A.H.M. ter Hofstede. Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. In Kurt Jensen, editor, *Proceedings of the Fourth International Workshop on Practical Use of Coloured Petri Nets and the CPN Tools*, volume 560 of DAIMI, pages 1–20, Aarhus, Denmark, August 2002. University of Aarhus.
7. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, June 2005.
8. P.A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 313–321, New Brunswick, NJ, July 1996. IEEE Computer Society.
9. G. Berthelot. Checking properties of nets using transformation. In *Advances in Petri Nets 1985, covers the 6th European Workshop on Applications and Theory in Petri Nets-selected papers*, pages 19–40, London, UK, 1986. Springer-Verlag.
10. G. Berthelot. Transformations and Decompositions of Nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets, Proceedings of an Advanced Course, Part 1*, volume 254 of *Lecture Notes in Computer Science*, pages 359–376, Bad Honnef, September 1986. Springer-Verlag.
11. P. Darondeau. Unbounded Petri net Synthesis. In J. Desel, W. Reisig, and G. Rozenberg, editors, *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 413–428, Eichstätt, Germany, 2003. Springer-Verlag.
12. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, United Kingdom, 1995.
13. B.F. van Dongen, W.M.P. van der Aalst, and H.M.W. Verbeek. Verification of EPCs: Using Reduction rules and Petri Nets. In O. Pastor and J. Falcão e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE 2005)*, volume 3520 of *Lecture Notes in Computer Science*, pages 372–386, Porto, Portugal, June 2005. Springer-Verlag.
14. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset Nets Between Decidability and Undecidability. In K. Larsen, S. Skjum, and G. Winskel, editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115, Aalborg, Denmark, July 1998. Springer-Verlag.
15. C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of Reset P/T Nets. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Lectures on Concurrency and Petri Nets*, volume 1644 of *Lecture Notes in Computer Science*, pages 301–310, Prague, Czech Republic, July 1999. Springer-Verlag.
16. A. Finkel, J.-F. Raskin, M. Samuelides, and L. van Begin. Monotonic Extensions of Petri Nets: Forward and Backward Search Revisited. *Electronic Notes in Theoretical Computer Science*, 68(6):1–22, 2002.

17. A. Finkel and Ph. Schnoebelen. Fundamental Structures in Well-Structured Infinite Transition Systems. In C.L. Lucchesi and A.V. Moura, editors, *Theoretical Informatics: Third Latin American Symposium, Campinas, LATIN'98*, volume 1380 of *Lecture Notes in Computer Science*, pages 102–118, Campinas, Brazil, 1998. Springer-Verlag.
18. A. Finkel and Ph. Schnoebelen. Well-structured Transition Systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, April 2001.
19. T. Murata. Petri nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
20. J.L. Peterson. *Petri net theory and the modeling of systems*. Prentice-Hall, Englewood Cliffs, USA, 1981.
21. C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für instrumentelle Mathematik, Bonn, Germany, 1962. In German.
22. W. Reisig and G. Rozenberg. Lectures on Petri Nets I: Basic Models. In W. Reisig and G. Rozenberg, editors, *Advances in Petri Nets*, volume 1491 of *Lecture Notes in Computer Science*, Berlin, Germany, 1998. Springer-Verlag.
23. W. Reisig and G. Rozenberg. Lectures on Petri Nets II: Basic Models. In W. Reisig and G. Rozenberg, editors, *Advances in Petri Nets*, volume 1492 of *Lecture Notes in Computer Science*, Berlin, Germany, 1998. Springer-Verlag.
24. R.H. Sloan and U.A. Buy. Reduction Rules for Time Petri Nets. *Acta Informatica*, 33(7):687–706, 1996.
25. H.M.W. Verbeek. *Verification of WF-nets*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, June 2004.
26. H.M.W. Verbeek, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Relaxed Soundness and Invariants. *The Computer Journal*, 2006. to appear.
27. M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Verifying workflows with Cancellation Regions and OR-joins: An Approach Based on Reset nets and Reachability Analysis. Technical report BPM-06-12, BPM Center (bpmcenter.org), 2006.
28. M.T. Wynn, D. Edmond, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Achieving a General, Formal and Decidable Approach to the OR-join in Workflow using Reset nets. In G. Ciardo and P. Darondeau, editors, *Proceedings of the 26th International conference on Application and Theory of Petri nets and Other Models of Concurrency*, volume 3536 of *Lecture Notes in Computer Science*, pages 423–443, Miami, USA, June 2005. Springer-Verlag.
29. M.T. Wynn, D. Edmond, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Synchronisation and Cancellation in Workflows based on Reset Nets. Technical report, submitted to an international journal, jan 2006, Queensland University of Technology, 2006.
30. M.T. Wynn, D. Edmond, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Verifying workflows with Cancellation Regions and OR-joins: An Approach Based on Reset nets and Reachability Analysis. In S. Dustdar, J. Fiadeiro, and A. Sheth, editors, *Proceedings of 4th International Conference of Business Process Management*, volume 4102 of *Lecture Notes in Computer Science*, pages 389–394, Vienna, Austria, Sep 2006. Springer-Verlag.
31. M.T. Wynn, H.M.W. Verbeek, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Reduction rules for Workflows With Cancellation Regions and OR-joins. Technical report BPM-06-24, BPM Center (bpmcenter.org), 2006.