On generating *-sound nets with substitution

Jacek Sroka, Piotr Chrząstowski-Wachtel
University of Warsaw, Poland
Email: {sroka, pch}@mimuw.edu.pl

Jan Hidders
Delft University of Technology, The Netherlands
Email: a.j.h.hidders@tudelft.nl

***Abstract*—We present a method for hierarchically generating sound workflow nets by substitution of nets with multiple inputs and outputs. We show that our method is correct and generalizes the class of nets generated by other hierarchical approaches. We identify a notion of soundness that is preserved by such substitutions and correct a small omission in an earlier similar method.**

I. Introduction

Nets are used as a tool for describing complex systems. They are useful especially when several agents do parts of a complex task in parallel. Parts of their jobs are local and can be hidden from the point of view of the others, parts of them must involve some communication between them. We can design the system by just drawing the control and data flow, but when the system is too complex, it is hard to visualize the overall result and to understand its structure. This can be improved by involving the structure of a task. The net is constructed hierarchically. This means that a single node can be expanded into a bigger net. In our approach we use Petri nets, where two kinds of nodes are used: circles representing items (states) and boxes representing actions. We must define the refinement rules in such a way that the kinds of nodes must match when we replace them. This means that if a node v is replaced by a net, the external nodes of this net should be of the same type as v.

Workflows constitute an important branch in business modeling and analysis. Numerous approaches support describing and analyzing workflows. Among them nets turned out to be probably the most successful. They offer both: an intuitive design, easy to understand even for a non-mathematician and a solid mathematical background with multiple analysis techniques, like algebraic invariants, temporal logic approach and many other. Workflow nets have been considered as Petri nets with one input and one output place, representing the beginning and the end of the flow. A token is given in the input place and while the workflow is run, it follows firing rules of Petri nets. It is desired that eventually it will reach the output place, which means that the workflow execution is completed.

When studying workflows one should consider "good" scenarios, giving us some desired properties. Among others two were identified as important. Van der Aalst proposed in [10] the notion of soundness, which informally speaking means two things. First, that if we start with an initial token, then no matter how we proceed with the execution of a workflow, we can always end up in the final state. Second, that every subtask, when originated will be completed, so there will be no trash tokens representing unfinished subtasks when we come to an end.

There are many approaches to constructing correct systems based on syntactical manipulation and combination of nets [4]. In this paper, we focus on a structural approach that corresponds to a top-down methodology. We design a system specifying actions of the higher levels first and make substitutions exchanging single nodes by more complex structures of lower levels exposing more details of the actions execution. This approach provides us two advantages. Firstly, it allows us to demonstrate solutions at appropriate levels, hence hiding the details when they are not desired. Secondly, it allows us to design in a manageable way quite complex scenarios. At any point of investigation we see only a part of the whole system, and whenever it is desired, we can unfold each node to verify, what's underneath. There are also some positive side effects. Such structural approach can make it easier to match appropriate levels of design with corresponding levels of management, hence allowing us to define fixed levels of security and rights.

A technical benefit of a top-down methodology is that it allows us to ensure that nets are sound by allowing only substitutions that guarantee soundness. However, as was observed by van Hee et al. in [12], it is unfortunately not true that soundness is preserved by substitution, i.e., if we substitute a sound net in another sound net the result is not necessarily sound. This is related to the fact that although if we execute a sound workflow, starting with a single token, then we will end up with a single token in the output place and no other tokens anywhere, it could be that if we start the same workflow with 2 tokens, it does not necessarily mean that the final marking will have 2 tokens in the output place. It can therefore happen that substitution of such a workflow net will lead to an unsound net. For this reason the notion of k-soundness was introduced by van Hee et al, where k is a parameter for which whenever we start with k tokens, the net will end without deadlock having exactly k tokens in the output place, while all other places will be unmarked. It was proven that k-soundness forms a strict hierarchy, which

This work was sponsored by the polish National Science Centre.

means that for every k there exist a workflow net which is k-sound and not $(k+1)$-sound. The notion of *-soundness is reserved for nets, which are sound for every k. It is claimed by van Hee et al. in [12] that this type of soundness is preserved by substitution. In the same paper van Hee et al. defines a large class of nets by starting from very simple classes that are syntactically easy to identify and can be straightforwardly shown to be *-sound, and then generating more *-sound nets by substitution.

The idea of net refinements is quite old, and the first papers were published in the early 90's, like [2]. Methods for stepwise refinements were studied in numerous papers, including [9], [8], [7] or [6]. Usually during the refinements we ask, which net properties are preserved. Here we concentrate on the generalized notion of soundness. In many papers the approach is the following: we create a net and ask for a possibly structural method to determine if a given property is satisfied, like in [1] or [5]. However, we propose a different approach: starting with a one-node net we would rather perform a series of refinements in such a way, that the desired property will be preserved. In [3] and [11] this approach was taken where the soundness was guaranteed by limiting the nets that can be substituted to only a certain small finite set of nets but taking a slightly more general notion of substitution. Interestingly enough, this leads to a slightly different class being generated, neither strictly larger or smaller, as compared to the class generated by van Hee et al in [12]. We therefore set out in this work to combine these two approaches to generate an even larger class of nets. For this purpose we will introduce a generalized notion of substitution that for example also allows the substitution of nets with multiple input and output places, and we introduce a correspondingly generalized notion of soundness that we will call substitution soundness and which is preserved by this type of substitution.

The structure of the paper is the following. After introducing the notions of a Petri net, workflow net and soundness we propose a new class of nets, called p-WF nets and t-WF nets. Such nets have the bordering nodes being places or transitions respectively. AND-OR nets being special classes of p-WF nets and t-WF nets are introduced in Section III. We make some remarks on their properties and specify how the substitution will be performed. Next we address the problem of soundness preservation during substitution in Section IV. The main two theorems of this section say that soundness is preserved when a substitution-sound t-WF net is substituted for a transition of a substitution-sound p-WF net or t-WF net. In section V we prove that the introduced AND-OR nets are substitution-sound in general.

II. Basic Terminology

Let S be a set. A bag (multiset) m over S is a function $m : S \to \mathbb{N}$. We use $+$ and $-$ for the sum and the difference of two bags and $=$, $<$, $>$, $\leq$, $\geq$ for comparisons of bags, which are defined in a standard way. We overload the set notation, writing $\emptyset$ for the empty bag and $\in$ for the element inclusion. We list elements of bags between brackets, e.g. $m = [p^2, q]$ for a bag m with $m(p) = 2$, $m(q) = 1$, and $m(x) = 0$ for all $x \notin \{p, q\}$. The shorthand notation $k.m$ is used to denote the sum of k bags m. The size of a bag m over S is defined as $|m| = \Sigma_{s \in S} m(s)$.

Definition 1 (Petri net)**.** A *Petri net* is a tuple $N = (P, T, F)$ with P a finite set of places, T a finite set of transitions such that $P \cap T = \emptyset$ and $F \subseteq (T \times F) \cup (F \times T)$ the set of flow edges.

A path of a net is a non-empty sequence $(x_1, ..., x_n)$ of nodes such that for all i such that $1 \leq i \leq n-1$ it holds that $(x_i, x_{i+1}) \in F$. Markings are states (configurations) of a net and the set of markings of $N = (P, T, F)$ is the set of all bags over P and denoted as $\mathbf{M}_N$. Given a transition $t \in T$, the preset $\bullet t$ and the postset $t\bullet$ of t are the sets $\{p \mid F(p, t)\}$ and $\{p \mid F(t, p)\}$, respectively. Analogously we write $\bullet p$, $p\bullet$ for pre- and postsets of places. To emphasize the fact that the preset/postset is considered within some net N, we write $\bullet_N a$, $a\bullet_N$. We overload this notation further allowing to apply preset and postset operations to a set B of places/transitions, which is defined as the union of pre-/postsets of elements of B. A transition $t \in T$ is said to be enabled in marking m iff $\bullet t \leq m$. For a net $N = (P, T, F)$ with markings m_1 and m_2 and a transition $t \in T$ we write $m_1 \xrightarrow{t}_N m_2$ if t is enabled in m_1 and $m_2 = m_1 - \bullet t + t\bullet$. For a sequence of transitions $\sigma = \langle t_1, \ldots, t_n \rangle$ we write $m_1 \xrightarrow{\sigma}_N m_n$ if $m_1 \xrightarrow{t_1}_N m_2 \xrightarrow{t_2}_N \ldots \xrightarrow{t_n}_N m_n$ and we write $m_1 \xrightarrow{*}_N m_n$ if there exists such a sequence $\sigma \in T^*$. We will write $m_1 \xrightarrow{t} m_2$ and $m_1 \xrightarrow{\sigma} m_n$ and $m_1 \xrightarrow{*} m_n$ if N is clear from the context.

We generalize the usual notion of workflow net as introduced by van der Aalst in [10] by allowing multiple input and output places, allowing transitions as input and output nodes and also allowing input nodes to have incoming edges and output nodes to have outgoing edges.

Definition 2 (Workflow net)**.** A *place Workflow net* (pWF net) is a tuple $N = (P, T, F, I, O)$ where (P, T, F) is a Petri net with a non-empty set $I \subseteq P$ of *input places* and a non-empty set $O \subseteq P$ of *output places* such that (1) every node in $P \cup T$ is reachable by a path from at least one node in I and (2) from every node in $P \cup T$ we can reach at least one node in O. A *transition Workflow net* (tWF net) is similar to a place Workflow net except that I and O are non-empty subsets of T. A *workflow net* (WF net) is either a pWF net or tWF net.

A workflow net is called a *one-input* workflow net if I contains one element, and a *one-output* workflow net if O contains one element. In [10] workflow nets are restricted to one-input one-output place Workflow nets. We generalize

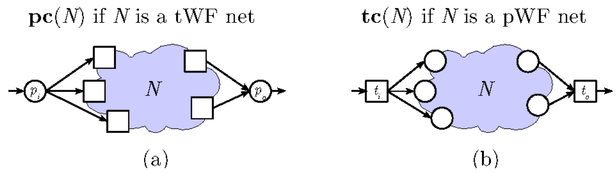


Figure 1. The place completion of a tWF net and a pWF net

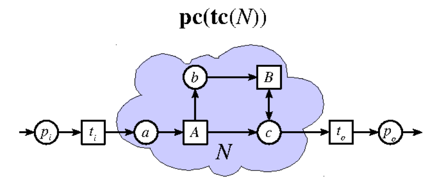


Figure 2. A counterexample showing that *-soundness is not preserved by transition completion and also not under substitution

this but define for all workflow nets the corresponding one-input one-output pWF net as follows. The *place-completion* of a tWF net $N = (P, T, F, I, O)$ is denoted as $\mathbf{pc}(N)$ and is a one-input one-output pWF net that is constructed from N by adding places p_i and p_o such that $p_i\bullet = I$ and $\bullet p_o = O$ and setting the input set and output set as $\{p_i\}$ and $\{p_o\}$ respectively. This is illustrated in Figure 1 (a). Note that we distinguish I nodes with half unconnected incoming arrows and O nodes with half unconnected outgoing arrow. The *transition-completion* of a pWF net $N = (P, T, F, I, O)$ is denoted as $\mathbf{tc}(N)$ and is a one-input one-output tWF net that is constructed from N by adding transitions t_i and t_o such that $t_i\bullet = I$ and $\bullet t_o = O$ and setting the input set and output set as $\{t_i\}$ and $\{t_o\}$, respectively. This is illustrated in Figure 1 (b).

We will focus in this paper on a particular kind of soundness, namely the soundness that guarantees the reachability of a proper final state. We generalize this for the case where there can be more than one input place and these contain one or more tokens in the initial marking. We also provide a generalization of soundness for tWF nets, which intuitively states that if there are k firings of input transitions, then the computation will end in an empty marking after k firings of the output transitions.

Definition 3 (k and *-soundness)**.** A pWF net $N = (P, T, F, I, O)$ is said to be *k-sound* if for each marking m such that $k.I \xrightarrow{*} m$ it holds that $m \xrightarrow{*} k.O$. We call N **-sound* if it is k-sound for all $k \geq 1$. We say that these properties hold for tWF net N if they hold for $\mathbf{pc}(N)$.

It would be nice if transition-completion would not affect the *-soundness of a net just like place-completion does (by definition). However this is only partially true as is shown in the following theorem.

Theorem 4. *Every pWF net N is *-sound if $\mathbf{tc}(N)$ is *-sound but not vice versa.*

Proof: Recall that by definition $\mathbf{tc}(N)$ is *-sound iff $\mathbf{pc}(\mathbf{tc}(N))$ is *-sound. Let $N = (P, T, F, I, O)$ and $N' = \mathbf{pc}(\mathbf{tc}(N)) = (P', T', F', I', O')$ with t_i and t_o being the added input and output transitions of $\mathbf{tc}(N)$, respectively. We assume that $\mathbf{tc}(N)$ is *-sound, that is N' is *-sound. Observe that $k.I' \xrightarrow{*}_{N'} k.I$ by letting input transitions t_i of $\mathbf{tc}(N)$ fire k times. Assume that $k.I \xrightarrow{*}_N m$. Since N is embedded in N' it then follows that $k.I' \xrightarrow{*}_{N'} m$. From the *-soundness of N' it follows that $m \xrightarrow{\sigma'}_{N'} k.O'$ for some $\sigma' \in \{T'\}^*$. However, we can omit the firings of t_o from σ' and obtain σ such that $m \xrightarrow{\sigma}_{N'} k.O$. Since σ cannot contain t_i it follows that $m \xrightarrow{\sigma}_N k.O$ and therefore $m \xrightarrow{*}_N k.O$.

The counterexample showing that not for every *-sound pWF net N it is true that $\mathbf{tc}(N)$ is *-sound is shown in Figure 2. Observe that N is *-sound. However the shown $\mathbf{pc}(\mathbf{tc}(N))$ is not since from the marking $[p_i]$ it can reach $[b, c]$ and therefore $[b, p_o]$ after which no transition is enabled. Since $\mathbf{pc}(\mathbf{tc}(N))$ is not 1-sound, then by definition $\mathbf{tc}(N)$ is also not 1-sound and thus not *-sound. ■

III. AND-OR NETS

To generate a large class of nets we will consider general substitutions where places and transitions are replaced with pWF nets and tWF nets, respectively. We introduce for this purpose a notion of substitution that is based on the one introduced by van Hee et al. in [12] but generalized so it can substitute nets with multiple input nodes and multiple output nodes.

Definition 5 (Place substitution, Transition substitution)**.** Consider two *disjoint* WF nets N and M, i.e., if $N = (P, T, F, I, O)$ and $M = (P', T', F', I', O')$ then $(P \cup T) \cap (P' \cup T') = \emptyset$.

Place substitution: If p is a place in N and M a pWF net, then we define the result of substituting p in N with M, denoted as $N \otimes_p M$, as the net that is obtained if in N we replace p with M, i.e., add edges such that $\bullet p' = \bullet p$ for each input place p' of M and $p'\bullet = p\bullet$ for each output place p' of M and remove p together with its edges from $\bullet p$ and to $p\bullet$. If $p \in I$ then p is replaced in the set of input nodes with I', and if $p \in O$ then p is replaced in the set of output nodes with O'.

Transition substitution: Likewise, if t is a transition in N and M a tWF net then we define the result of substituting t in N with M, denoted as $N \otimes_t M$, as the net that is obtained if in N we replace t with M, i.e., add edges such that $\bullet t' = \bullet t$ for each input transition t' of M and $t'\bullet = t\bullet$ for each output transition t' of M and remove t together with its edges from $\bullet t$ and to $t\bullet$. If $t \in I$ then t is replaced in the set of input nodes with I', and if $t \in O$ then t is replaced in the set of output nodes with O'.

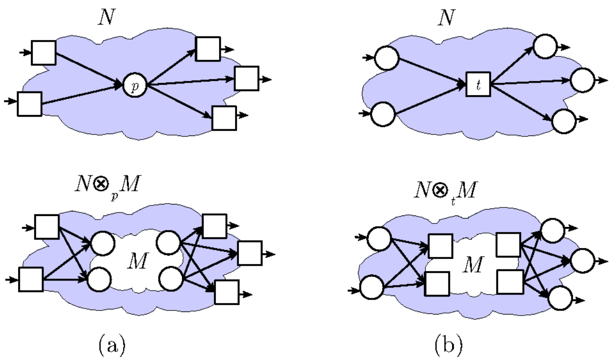


Figure 3. Illustration of place substitution and transition substitution

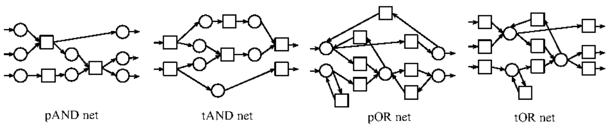


Figure 4. Examples of a pAND, tAND, pOR and tOR nets

The results of a place substitution and transition substitution are illustrated in Figure 3 (a) and (b), respectively. It is not hard to see that if N and M are WF nets and n a node in N then $N \otimes_n M$ is again a WF net. It also holds for all WF nets A, B and C that $(A \otimes_a B) \otimes_b C = A \otimes_a (B \otimes_b C)$ if b is a node in B, and $(A \otimes_a B) \otimes_b C = (A \otimes_a C) \otimes_b B$ if a and b are nodes in A.

As the basic nets with which we will start the generation process we will consider the nets that we call pAND nets, tAND nets, pOR nets and tOR nets, which are all illustrated in Figure 4 with input and output nodes on the left-hand side and right-hand side, respectively. Informally we can describe AND nets as acyclic nets that consist only of AND splits and AND joins, and OR nets can be described as possibly cyclic nets consisting of only OR splits and OR joins. AND and OR nets are generalizations of state machines/S-nets and marked graph/T-nets [4], respectively, which have one input and output node. More formally, these are defined as follows.

Definition 6 (AND net)**.** An *AND net* is an acyclic WF net (P, T, F, I, O) such that for every place $p \in P$ it holds that (1) $p \in I \wedge |\bullet\, p| = 0$ or $p \notin I \wedge |\bullet\, p| = 1$ and (2) $p \in O \wedge |p\, \bullet| = 0$ or $p \notin O \wedge |p\, \bullet| = 1$. An AND net that is a pWF net is called a pAND net, and if it is a tWF net it is called a tAND net.

OR nets are the counterpart of AND nets and are defined as follows.

Definition 7 (OR net)**.** An *OR net* is a WF net (P, T, F, I, O) such that for every transition $t \in T$ it holds that (1) $t \in I \wedge |\bullet\, t| = 0$ or $t \notin I \wedge |\bullet\, t| = 1$ and (2) $t \in O \wedge |t\, \bullet| = 0$ or $t \notin O \wedge |t\, \bullet| = 1$. An OR net that is a pWF net is called a pOR net, and if it is a tWF net it is called a tOR net.

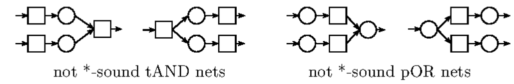


Figure 5. Examples of tAND and pOR nets that are not *-sound

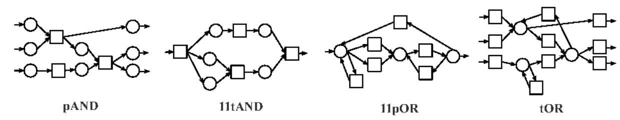


Figure 6. Example nets from classes **pAND**, **11tAND**, **11pOR** and **tOR**

Note that OR nets can contain cycles and AND nets by definition cannot, but otherwise they are each others dual. For these nets there are some straightforward soundness results in that all pAND and tOR nets are *-sound, and that for tAND and pOR nets this is the case if they are one-input one-output nets. To understand the restriction to one-input one-output nets consider the examples tAND and pOR nets in Figure 5 which are all nets with either multiple input nodes or multiple output nodes and which are all not *-sound. This is why, while generating nets with place and transition substitution, we limit ourselves to the following classes of nets: the class of pAND nets represented by $\mathbf{pAND}$, the class of one-input one-output tAND nets represented by $\mathbf{11tAND}$, the class of one-input one-output pOR nets represented by $\mathbf{11pOR}$**,** and the class of tOR nets represented by $\mathbf{tOR}$ (see Figure 6 for examples).

We will generate nets by allowing substitutions of places with pWF nets and transitions with tWF nets.

Definition 8 (Substitution closure, AND-OR net)**.** Given a class C of nets we defined the substitution closure of C, denoted as $\mathbf{S}(C)$, as the smallest superclass of C that is closed under transition substitution and place substitution, i.e., the following two rules hold: if N and M are disjoint nets in $\mathbf{S}(C)$ then (1) if M is a pWF net and p a place in N then $N \otimes_p M$ is a net in $\mathbf{S}(C)$ and (2) if M is a tWF net and t a transition in N then $N \otimes_t M$ is a net in $\mathbf{S}(C)$. We call the class $\mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{11pOR} \cup \mathbf{tOR})$ the class of AND-OR nets.

An example of the generation of an AND-OR net is shown in Figure 7, with on the left-hand side the hierarchical decomposition and on the right-hand side the resulting net.

It can be shown that the tAND nets are not needed, i.e.,

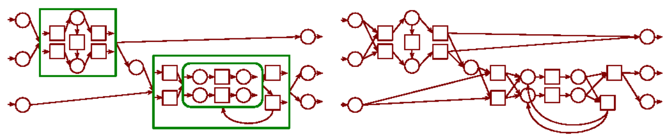
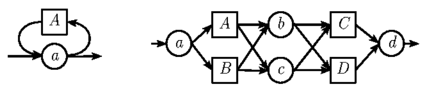


Figure 7. An example of the generation of an AND-OR net



(a) (b)

Figure 8. Examples showing the expressive power of certain classes

we can remove them from the initial class without changing the set of nets that can be generated.

Theorem 9. *The tAND nets are redundant for generating AND-OR nets, i.e.,* $\mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{11pOR} \cup \mathbf{tOR}) = \mathbf{S}(\mathbf{pAND} \cup \mathbf{11pOR} \cup \mathbf{tOR})$.

Proof: Recall that tAND nets do not contain cycles. Also note that if we take a one-input one-output tAND net with input transition t_i and output transition t_o and we remove the begin and end transition, then we are left with a pAND net with $I = t_i\bullet$ and $O = \bullet t_o$. So every one-input one-output tAND net can be generated by starting with an tOR net consisting of a transition followed by a place which is again followed by a transition, and then substituting the previously mentioned pAND net for the place in the middle. ■

However, the pOR nets are not redundant, because a cycle containing the input and output nodes can not be obtained in any other way.

Theorem 10. *The pOR nets are not redundant for generating all AND-OR nets, i.e.,* $\mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{11pOR} \cup \mathbf{tOR}) \supsetneq \mathbf{S}(\mathbf{pAND} \cup \mathbf{11tAND} \cup \mathbf{tOR})$.

Proof: See the counterexample in Figure 8 (a). This one-input one-output pOR net cannot be generated by using pAND, one-input one-output tAND and tOR nets. ■

Of course pAND nets and tOR nets are not redundant either, since they allow for multiple input and output nodes.

The AND-OR nets are very similar to the ST nets defined in [12] by van Hee et al. In fact, the ST nets are exactly $\mathbf{S}(\mathbf{11tAND} \cup \mathbf{11pOR})$. This is a proper subclass of the AND-OR nets since it only contains one-input one-output WF nets. However there are in addition also one-input one-output AND-OR nets that are not in $\mathbf{S}(\mathbf{11tAND} \cup \mathbf{11pOR})$ as is shown by the following theorem.

Theorem 11. *The class* $\mathbf{S}(\mathbf{11tAND} \cup \mathbf{11pOR})$ *does not contain all one-input one-output AND-OR nets.*

Proof: The counterexample is in Figure 8 (b). To show that it is an AND-OR net we consider its generation in reverse. The transitions A and B form an tOR net and can be contracted into a single transition. The same for the transitions C and D. The places b and c form a pAND net and can be contracted into a single place. The result will be a linear net that is in fact both a pAND net and a one-input one-output pOR net. To see that the example net is not in $\mathbf{S}(\mathbf{11tAND} \cup \mathbf{11pOR})$ it can be verified that there is no proper subnet that is either in $\mathbf{11tAND}$ or $\mathbf{11pOR}$ and can be contracted into a single transition or place, respectively. ■

IV. Substitution Soundness

It is unfortunately not true that *-soundness is preserved by substitutions as defined in this paper, because we allow the output nodes to have outgoing edges. A counterexample is shown in Figure 2 where the shown pWF net can be thought of as being constructed by substituting a *-sound net N, with input place a and output place c, into an also *-sound pWF net. As was discussed in the proof of Theorem 4 the resulting net is not 1-sound and therefore also not *-sound. Therefore, we introduce a new notion of soundness called *substitution soundness* and study its properties[1].

The intuition underlying substitution soundness is that it should not matter that during a run of a workflow net we remove tokens from the output places that seem to be ready. In other words, it should hold that if the net started with k tokens in the input places and at one point we have in each output place at least $k' \leq k$ tokens, and we remove these k' tokens from each output place, then the net can still finish with $k - k'$ tokens in each output place.

Definition 12 (Substitution soundness). Let $N = (P, T, F, I, O)$ be a pWF net. We say that N is *substitution sound* (or simply *sub-sound)* if for all $k \geq k' \geq 0$ and every marking m' it holds that if $k.I \xrightarrow{*} (m' + k'.O)$ then $m' \xrightarrow{*} (k - k').O$. We generalize this property for tWF nets and say that a tWF net N is sub-sound if $\mathbf{pc}(N)$ is sub-sound.

We already know that sub-soundness is a necessary condition for constructing *-sound nets by our notion of substitution. Now we prove that sub-soundness is sufficient for constructing *-sound nets by substitution. First, note that the case where $k' = 0$ describes *-soundness and so sub-soundness implies *-soundness. Furthermore, on many classes of nets the two notions of soundness coincide, as is shown by the following two lemmas.

Lemma 13. *For every pWF net N such that all output places have no outgoing edges it holds that N is *-sound iff N is sub-sound.*

[1]Some proofs in this section are long and technical and had to be summarized because of the page limit.

Proof: As already argued it holds that sub-soundness implies *-soundness, so the converse remains to be shown. Let $N = (P, T, F, I, O)$. Assume that $k.I \xrightarrow{*} (m + k'.O)$ for some k' such that $k \geq k' \geq 0$. By *-soundness it holds for some σ that $(m + k'.O) \xrightarrow{\sigma} k.O$. However, since the places in O have no outgoing edges none of the transitions in σ consumes any of their tokens and so $m \xrightarrow{\sigma} (k - k').O$. ■

Note that the restriction mentioned in Lemma 13 is included in the classical definition of WF net by van der Aalst [10]. However, with this restriction we would not be able to generate all AND-OR nets, not even all those that satisfy this restriction. In particular we would not be able to do arbitrary loop additions. As an example consider Figure 8 (b) where we would not be able to add a loop to place b. Note that a similar restriction is not necessary for tWF nets because the soundness properties are defined by place completion for them. Recall also that for tOR nets the output transitions cannot have outgoing edges by definition and for one-input one-output tAND nets this follows from the facts that AND nets are acyclic and that in a tWF nets it is possible from every place and transition to reach one of the output transitions.

Lemma 14. *For every tWF net N it holds that N is *-sound iff N is sub-sound.*

Proof: As already argued, it is enough to show that *-soundness implies sub-soundness. A tWF net N is by definition sub-sound iff $\mathbf{pc}(N)$ is sub-sound. Since in $\mathbf{pc}(N)$ the output place has no outgoing edges it follows from Lemma 13 that $\mathbf{pc}(N)$ is sub-sound iff it is *-sound. Finally, by definition it holds that $\mathbf{pc}(N)$ is *-sound iff N is *-sound.

We claim this is the weakest condition that is necessary to construct 1-sound nets by substitution of nodes in 1-sound nets. This implies that it is a necessary condition in the sense that there is no weaker condition that is preserved by substitution and implies 1-soundness. Now we proceed with proving that place substitution preserves sub-soundness. We start with showing this for pWF nets, and then show that it also holds for tWF nets. ■

Theorem 15. *If a pWF net $N = (P_N, T_N, F_N, I_N, O_N)$ and a disjoint pWF net $M = (P_M, T_M, F_M, I_M, O_M)$ are sub-sound, then for any $p \in P_N$ the net $N \otimes_p M$ is also sub-sound.*

Proof: (Sketch) Let $N_{NM} = N \otimes_p M = (P_{NM}, T_{NM}, F_{NM}, I_{NM}, O_{NM})$. We define $\mathbf{S}(M, k)$ as the set of markings m of M that represent the fact that there are still k "threads" active in M after having started possibly with more threads but some of them were ended by the removal of tokens from O', i.e., all markings m_M such that for some $k' \geq k$ it holds that $k'.I' \xrightarrow{*}_M m_M + (k' - k).O_M$. We define a simulation relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_{NM}$ such that $m_N \sim m_{NM}$ represents the fact that m_N is the same as m_{NM} except that all (say k) tokens are removed from p and some marking in $\mathbf{S}(M, k)$ is added, i.e., $m_{NM} = m_N - [p^k] + m_M^k$ for some $m_M^k \in \mathbf{S}(M, k)$ with $k = m_N(p)$.

It can then be shown that $\sim$ indeed defines a kind of bisimilarity, i.e., it holds that (1) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_{NM}$, then there is a marking m'_{NM} such that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ and $m'_N \sim m'_{NM}$ and (2) if $m_{NM} \xrightarrow{\sigma}_{NM} m'_{NM}$ and $m_N \sim m_{NM}$, then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_{NM}$. This can be shown with induction on the length of σ. For each transition t in σ we then distinguish for (1) the cases where $p \in \bullet_N t$ or not and $p \in t\bullet_N$ or not. Likewise for (2) we distinguish these cases when t is a transition in N.

We then can show the sub-soundness of $N \otimes_p M$ using (1) and (2). Assume that $k.I_{NM} \xrightarrow{*}_{NM} (m_{NM} + k'.O_{NM})$ with $k \geq k' \geq 0$. By (2) it then follows that $k.I_N \xrightarrow{*}_N m_N$ such that $m_N \sim (m_{NM} + k'.O_{NM})$. We can show that we can assume that $m_N = m'_N + k'.O_N$ with m'_N a marking of N. By the sub-soundness of N it holds that $m'_N \xrightarrow{*}_N (k - k').O_N$. We can show that $m'_N \sim m_{NM}$ and so from (1) it then follows that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ such that $(k - k').O_N \sim m'_{NM}$. Finally, by using the sub-soundness of M it can be shown that $m'{}_{NM} \xrightarrow{*}_{NM} (k - k').O_{NM}$.

We proceed with the full version of the last part, i.e., prove the sub-soundness of $N \otimes_p M$ using (1) and (2). Assume that $k.I_{NM} \xrightarrow{*}_{NM} (m_{NM} + k'.O_{NM})$ with $k \geq k' \geq 0$. Since $I_{NM} = I_N$ if $p \notin I_N$ and $I_{NM} = I_N - [p] + I_M$ if$p \in I_N$, it holds that $k.I_N \sim k.I_{NM}$. By (2) it then follows that $k.I_N \xrightarrow{*}_N m_N$ such that $m_N \sim (m_{NM} + k'.O_{NM})$.

We now construct $m'_N = m_N - k'.O_N$ and show that $m'_N \sim m_{NM}$ regardless of $p \notin O_N$ or $p \in O_N$. We start with showing the fact that m'_N is a valid state, i.e., m_N includes the tokens we are subtracting from it. Since $m_N \sim (m_{NM} + k'.O_{NM})$ there is $m_M^{k''} \in \mathbf{S}(M, k'')$ such that $m_{NM} + k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''}$with $k'' = m_N(p)$. This gives $m_N = m_{NM} + k'.O_{NM} + [p^{k''}] - m_M^{k''}$. For $p \notin O_N$, in which case $O_{NM} = O_N$, this gives $m_N = m_{NM} + k'.O_N + [p^{k''}] - m_M^{k''}$. It remains to observe that the $m_M^{k''}$ component does not remove any tokens from O_N because from disjointness of N and M we have $O_N \cap P_M = \emptyset$. For $p \in O_N$, in which case $O_{NM} = O_N - [p^k] + O_M$, we get $m_N = m_{NM} + k'.O_N - [p^{k'}] + k'.O_M + [p^{k''}] - m_M^{k''} = m_{NM} + k'.O_N + [p^{k''-k'}] + k'.O_M - m_M^{k''}$. The right hand side of the equality has to include the same number of tokens in p. Since m_{NM} marks only places from $P_{NM} = (P_N \setminus \{p\}) \cup P_M$, $k'.O_M - m_M^{k''}$ only places from P_M, and $p \notin P_M$, all the tokens in p are given by $k'.O_N + [p^{k''-k'}]$. It remains to show that $k'' \geq k'$. This follows from further examination of the equality $m_N = m_{NM} + k'.O_N + [p^{k''-k'}] + k'.O_M - m_M^{k''}$. This time we look at the number of tokens in O_M. On the left-hand there are clearly none. On the right hand side there are k' introduced by $k'.O_M$, possibly yet some introduced by O_{NM}, and the only negative component $-m_M^{k''}$ subtracts no more that k'' of such tokens. Now we continue with showing that $m'_N \sim m_{NM}$. This time from $m_{NM} + k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''}$ we conclude $m_{NM} = m_N - [p^{k''}] + m_M^{k''} - k'.O_{NM}$ and again consider the two cases for $p \notin O_N$ or $p \in O_N$. If $p \notin O_N$, then $O_{NM} = O_N$ and so $m_N - [p^{k''}] + m_M^{k''} - k'.O_{NM} = m_N - k'.O_N - [p^{k''}] + m_M^{k''} = m'_N - [p^{k''}] + m_M^{k''}$ so $m'_N \sim m_{NM}$. If $p \in O_N$, then $O_{NM} = O_N - [p] + O_M$ and so $m_N - [p^{k''}] + m_M^{k''} - k'.O_{NM} = m_N - [p^{k''}] + m_M^{k''} - k'.O_N + [p^{k'}] - k'.O_M = m_N - k'.O_N - [p^{k''-k'}] + m_M^{k''} - k'.O_M = m'_N - [p^{k''-k'}] + m_M^{k''} - k'.O_M$, so also then we can conclude that $m'_N \sim m_{NM}$ because $m_M^{k''} - k'.O_M \in \mathbf{S}(M, k'' - k')$ and $k'' - k' = (m_N - k'.O_N)(p) = m'_N(p)$. By the sub-soundness of N it then holds that $m'_N \xrightarrow{*}_N (k - k').O_N$. From (1) it then follows that $m_{NM} \xrightarrow{*}_{NM} m'_{NM}$ such that $(k - k').O_N \sim m'_{NM}$, that is $m'_{NM} = (k - k').O_N - [p^x] + m_M^x$ with $m_M^x \in \mathbf{S}(M, x)$ and $x = (k - k').O_N(p)$. If $p \notin O_N$, then $O_N = O_{NM}$ and $x = 0$, and therefore $m'_{NM} = (k - k').O_{NM}$. If $p \in O_N$, then $O_{NM} = O_N - [p] + O_M$ and $x = k - k'$, and therefore $m'_{NM} = (k - k').O_N - [p^{k-k'}] + m_M^{k-k'}$. Because M is sub-sound, it holds that $m_M^{k-k'} \xrightarrow{*}_M (k - k').O_M$, and since M is embedded in N, it follows that $m'_{NM} = (k - k').O_N - [p^{k-k'}] + m_M^{k-k'} \xrightarrow{*}_{NM} (k - k').O_N - [p^{k-k'}] + (k - k').O_M = (k - k').O_{NM}$.

■

We now proceed with the case for place substitution in tWF nets. For that we will use the following lemma.

Lemma 16. *For every tWF net N with a place p and a disjoint pWF net M it holds that $\mathbf{pc}(N \otimes_p M) = \mathbf{pc}(N) \otimes_p M$.*

Proof: Let $N = (P_N, T_N, F_N, I_N, O_N)$ with $p \in P_N$ and $M = (P_M, T_M, F_M, I_M, O_M)$. In both cases the same nodes are added, viz., those of M and p_i and p_o. Clearly the edges F_M are added in the same way. Also in both cases afterward $p_i\bullet = I_N$ and $\bullet p_o = O_N$ because N is a tWF net and $p \notin I_N$ and $p \notin O_N$. For nodes $p' \in I_M$ it holds in both cases that afterward $\bullet p' = \bullet_N p$ if $p \notin I_N$ and $\bullet p' = \{p_i\}$ if otherwise. Similarly for nodes $p' \in O_M$ afterward $p'\bullet = p\bullet_N$ if $p \notin O_N$ and $p'\bullet = \{p_o\}$. Finally, in both cases the final input set is $\{p_i\}$ and the final output set is $\{p_o\}$. ■

Theorem 17. *If a tWF net N is sub-sound and a disjoint pWF net M is sub-sound and p is a place in N then $N \otimes_p M$ is sub-sound.*

Proof: Assume that a tWF net N is sub-sound and a pWF net M is sub-sound. By definition of sub-soundness for tWF nets it follows that $\mathbf{pc}(N)$ is sub-sound. By Theorem 15 it follows that $\mathbf{pc}(N) \otimes_p M$ is sub-sound. By Lemma 16 it then holds that $\mathbf{pc}(N \otimes_p M)$ is sub-sound.

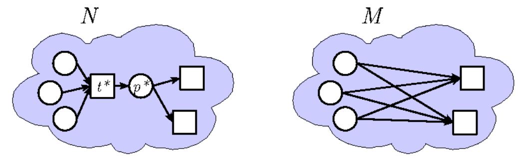


Figure 10. Transition-place pair removal

Finally, by definition of sub-soundness for tWF nets, it follows that $N \otimes_p M$ is sub-sound. ■

We now proceed with showing that also transition substitution preserves sub-soundness. The proof strategy will be to show that this substitution is equivalent to a place substitution as is illustrated in Figure 9. In the left-top corner we see the original net N with transition t that is to be replaced with net M, the result of which, i.e., $N \otimes_t M$, is shown in the left-bottom corner. The equivalent place substitution is shown in the right column. Here we see at the top $N \otimes_t \mathbf{tc}(N^*)$ which is equivalent to N in the sense that it is sub-sound iff N is sub-sound. In the right-bottom corner we see the result of substituting the place p^* in $N \otimes_t \mathbf{tc}(N^*)$ with the pWF net $\mathbf{pc}(M)$. As will be shown this net is sub-sound iff $N \otimes_t M$, the net in the left-bottom corner, is sub-sound. It then follows that the transition substitution in the left column preserves sub-soundness if the one in the right column also does this, which, as was just shown it indeed does.

We begin now with the lemmas that show that the two WF nets in the bottom row of Figure 9 are equivalent for sub-soundness. These results are similar to those of the abstraction rule of [4].

Lemma 18. *Let N be a pWF net with transition t^* and place p^* such that $t^*\bullet_N = \bullet_N p^*$ and p^* is not an input nor output place and there are no edges between $\bullet_N t^*$ and $p^*\bullet_N$. Furthermore, let M be the pWF net that is obtained from N if we remove t^* and p^* and add all the edges in $\bullet_N t^* \times p^*\bullet_N$ as illustrated in Figure 10. Then N is sub-sound iff M is sub-sound.*

Proof: (Sketch) We define a similarity relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ such that $m_N \sim m_M$ represents the fact that m_M is the same as m_N except that all (say k) tokens are removed from p^* and k tokens are added to each of $\bullet t^*$, or in other words, fire k times in reverse t^*. More formally: $m_N \sim m_M$ holds iff $m_M = m_N - [p^{*k}] + k.(\bullet_N t^*)$ where $k = m_N(p^*)$.

It can then be shown that $\sim$ indeed defines a kind of bisimilarity, i.e., it holds that (1) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$, then there is a marking m'_M such that $m_M \xrightarrow{*}_M m'_M$ and $m'_N \sim m'_M$ and (2) if $m_M \xrightarrow{\sigma}_M m'_M$ and $m_N \sim m_M$, then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_M$. This can be shown with induction on the length of σ. For each transition t in σ we then distinguish for (1) the cases where $t = t^*$ and if not

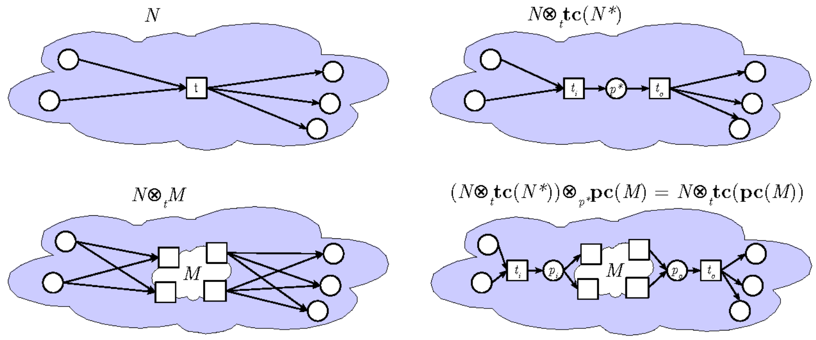


Figure 9. Transforming transition substitution to place substitution

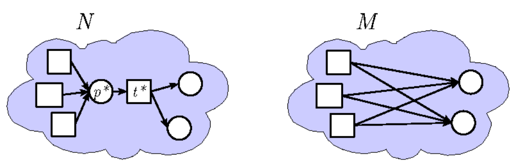


Figure 11. Place-transition pair removal

then $p^* \in \bullet_N t$ or not. Likewise for (2) we distinguish the cases where $p^* \in \bullet_N t$ or not.

We now show that N is sub-sound if M is sub-sound. Note that by construction N and M have the same input set I and output set O, and that $I \sim I$ and $O \sim O$. Assume that $k.I \xrightarrow{*}_N (m_N + k'.O)$. By (1) it follows that $k.I \xrightarrow{*}_M m_M$ such that $(m_N + k'.O) \sim m_M$. Since p^* is not an output node we can assume that $m_M = m'_M + k'.O$ with $m_N \sim m'_M$. From the sub-soundness of M it follows that at $m'_M \xrightarrow{*}_M (k-k').O$. By (2) it follows that $m_N \xrightarrow{*}_N m'_N$ such $m'_N \sim (k-k').O$. Since p^* is not an output node it follows that $m'_N = (k-k').O$.

The proof for the fact that M is sub-sound if N is sub-sound proceeds analogously. ■

Lemma 19. *Let N be a pWF net with place p^* and transition t^* such that $p^* \bullet_N = \bullet_N t^*$ and p^* not an input nor output place and there are no edges between $\bullet_N p^*$ and $t^* \bullet_N$. Furthermore, let M be the pWF net that is obtained from N if we remove p^* and t^* and add all the edges in $\bullet_N p^* \times t^* \bullet_N$ as illustrated in Figure 11. Then N is sub-sound iff M is sub-sound.*

Proof: The proof proceeds analogously to that of the preceding Lemma 18 with the relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ redefined such that $m \sim m'$ iff $m' = m - [p^{*k}] + k.(t^* \bullet_N)$ where $k = m(p^*)$. ■

We now proceed with the lemma that shows that in Figure 9 the bottom nets are equivalent in the sense that one is sub-sound iff the other is.

Lemma 20. *If N is a pWF net with a transition t^* and N^* a pWF net that consists of only a single place p^*, then $N \otimes_{t^*} \mathbf{tc}(N^*)$ is sub-sound iff N is sub-sound.*

Proof: (Sketch) Let $M = N \otimes_t \mathbf{tc}(N^*)$. We define a relation $\sim \subseteq \mathbf{M}_N \times \mathbf{M}_M$ such that $m_N \sim m_M$ represents the fact that m_N is the same as m_M except that all (say k) tokens are removed from p^* and k tokens are added to each of $t_o \bullet_M$, or in other words, fire k times in t_o where t_o is the output transition added in $\mathbf{tc}(N^*)$. More formally: at $m_N \sim m_M$ iff $m_N = m_M - [p^{*k}] + k.(t_o \bullet_M)$ where $k = m_M(p^*)$.

It can then be shown that $\sim$ indeed defines a kind of bisimilarity, i.e., it holds that (1) if $m_N \xrightarrow{\sigma}_N m'_N$ and $m_N \sim m_M$, then there is a marking m'_M such that $m_M \xrightarrow{*}_M m'_M$ and $m'_N \sim m'_M$ and (2) if $m_M \xrightarrow{\sigma}_M m'_M$ and $m_N \sim m_M$, then there is a marking m'_N such that $m_N \xrightarrow{*}_N m'_N$ and $m'_N \sim m'_M$. This can be shown with induction on the length of σ. For each transition s in σ we then distinguish for (1) the cases where $s = t$ or not. Likewise for (2) we distinguish the cases where $s = t_i$ or $s = t_o$ or a transition in N not equal to t.

We now show by using the preceding facts that M is sub-sound if N is sub-sound. Note that by construction N and M have the same input set I and output set O, and that $I \sim I$ and $O \sim O$. Assume that $k.I \xrightarrow{*}_M (m_M + k'.O)$. By (2) it follows that $k.I \xrightarrow{*}_N m_N$ such that $m_N \sim (m_M + k'.O)$. Since $p^* \notin O$ we can assume that $k'.O \leq m_N$ and $(m_N - k'.O) \sim m_M$. From the sub-soundness of N it follows that at $(m_N - k'.O) \xrightarrow{*}_N (k-k').O$. By (1) it follows that $m_M \xrightarrow{*}_M m'_M$ such that $(k-k').O \sim m'_M$. Since $p^* \notin O$ it follows that $m'_M = (k-k').O$.

The proof for the fact that N is sub-sound if M is sub-sound proceeds analogously. ■

We are now ready to prove that sub-soundness is preserved by transition substitution.

Theorem 21. *If a pWF net N is sub-sound and a disjoint tWF net M is sub-sound and t is a transition in N, then $N \otimes_t M$ is sub-sound.*

Proof: (Sketch) Let $N = (P, T, F, I, O)$ be a sub-sound pWF net containing a transition t, and $M =$

(P', T', F', I', O') a sub-sound tWF net. Furthermore, let N^* be a pWF net consisting of a single new place p^*. We then show the following two facts:

(A) $(N \otimes_t (\mathbf{tc}(N^*))) \otimes_{p^*} \mathbf{pc}(M) = N \otimes_t (\mathbf{tc}(\mathbf{pc}(M)))$. *proof*: To see this consider the illustration in Figure 9. It is not hard to see that taking the top-left net (N) and substituting $\mathbf{tc}(\mathbf{pc}(M))$ for t gives the bottom-right net. However, if we first substitute in N the transition t with $\mathbf{tc}(N^*)$ we get the top-right net, and this leads also to the bottom-right net if we subsequently substitute $\mathbf{pc}(M)$ for the place p^*.

(B) $N \otimes_t M$ is sub-sound iff $(N \otimes_t \mathbf{tc}(\mathbf{pc}(M)))$ is sub-sound. *proof*: Consider the bottom-right net in Figure 9 which represents $(N \otimes_t \mathbf{tc}(\mathbf{pc}(M)))$. This can be transformed into the bottom-left net, which represents $N \otimes_t M$, by (i) removing t_i and p_i and connecting all nodes in $\bullet t_i = \bullet_N t$ with those in $p_i \bullet = I'$, and (ii) removing p_o and t_o and connecting all nodes in $\bullet p_o = O'$ with those in $t_o \bullet = t \bullet_N$. By construction there are in the bottom-right net no edges between nodes in $\bullet t_i$ and in $p_i \bullet$, and between nodes in $\bullet p_o$ and in $t_o \bullet$. It follows that operations (i) and (ii) are equivalent with those in Lemma 18 and Lemma 19, and by these lemmas it then follows that the right-bottom net for $(N \otimes_t \mathbf{tc}(\mathbf{pc}(M)))$ is sub-sound iff the left-bottom net for $N \otimes_t M$ is.

Then we argue as follows. Since M is sub-sound it follows by definition that $\mathbf{pc}(M)$ is sub-sound. By Lemma 20 it holds that $N \otimes_t \mathbf{tc}(N^*)$ is sub-sound. By Theorem 15 it follows that $(N \otimes_t (\mathbf{tc}(N^*))) \otimes_{p^*} \mathbf{pc}(M)$ is sub-sound. By (A) it then follows that $N \otimes_t (\mathbf{tc}(\mathbf{pc}(M))$ is sub-sound. By (B) it then follows that $N \otimes_t M$ is sub-sound. ∎

Theorem 22. *If a tWF net N is sub-sound and a disjoint tWF net M is sub-sound and t is a transition in N then $N \otimes_t M$ is sub-sound.*

Proof: Assume that N is sub-sound tWF net with a transition t and M a sub-sound tWF net. By Lemma 16 it holds that $\mathbf{pc}(N \otimes_t M) = \mathbf{pc}(N) \otimes_t M$. By Theorem 21 it follows that $\mathbf{pc}(N) \otimes_t M$ is sub-sound. Since by Lemma 16 it holds that $\mathbf{pc}(N \otimes_t M) = \mathbf{pc}(N) \otimes_t M$, it follows that $\mathbf{pc}(N \otimes_t M)$ is sub-sound. By definition of sub-soundness of tWF nets it then holds that $N \otimes_t M$ is sub-sound. ∎

Corollary 23. *If N and M are disjoint sub-sound WF nets and n is a node in N then $N \otimes_n M$ (if defined) is a sub-sound WF net.*

Proof: This follows from the fact that Theorem 15, Theorem 17, Theorem 21 and Theorem 22 cover all possible combinations of N and M being pWF nets or tWF nets. ∎

V. Sub-soundness of AND-OR nets

In this section we show that all AND-OR nets are sub-sound. First we show that the AND and OR nets from which AND-OR nets are generated are sound.

Theorem 24. *Every one-input one-output pOR net is sub-sound.*

Proof: Let $I_N = \{p_i\}$ and $O_N = \{p_o\}$. For each place p in a pOR net N it holds that $[p_i] \xrightarrow{*} [p]$ and $[p] \xrightarrow{*} [p_o]$ since there must be paths from p_i to p and from p to p_o and each transition in those paths has one input edge and one output edge. It follows that (A) if $|m| = k$, then $k.[p_i] \xrightarrow{*} m$ and $m \xrightarrow{*} k.[p_o]$. It can also be shown by induction on the length of σ that (B) if $|m| = k$ and $m \xrightarrow{\sigma} m'$ then $|m'| = k$.

We now show the sub-soundness requirement. Assume that $k.I_N \xrightarrow{*} (m + k'.O_N)$. Since $|k.I_N| = |k.[p_i]| = k.|[p_i]| = k$ it follows by (B) that $|m + k'.O_N| = k$. Since $|m + k'.O_N| = |m| + |k'.O_N|$ and $|k.O_N| = |k.[p_o]| = k.|[p_o]| = k$ it follows that $|m| = k - k'$. By (A) it then follows that $m \xrightarrow{*} (k - k').[p_o] = (k - k').O_N$. ∎

Theorem 25. *Every tOR net is sub-sound.*

Proof: Consider a tOR net N. Then the place-completion of N, $\mathbf{pc}(N)$, is a one-input one-output pOR net. Moreover, by the definition of sub-soundness for tWF nets it holds that N is sub-sound iff $\mathbf{pc}(N)$ is sub-sound, and in Theorem 24 it is shown that all one-input pOR nets are *-sound. ∎

Theorem 26. *Every one-input one-output tAND net is sub-sound.*

Proof: Theorem 17 in [12] says that all one-input one-output tAND nets are *-sound, which holds since they are acyclic. By Lemma 14 it follows that they are therefore sub-sound. ∎

Theorem 27. *Every pAND net is sub-sound.*

Proof: Consider a pAND net N. Then the transition-completion of N, $\mathbf{tc}(N)$, is a one-input one-output tAND net. By Theorem 17 in [12] it follows that $\mathbf{tc}(N)$ is *-sound. By Theorem 4 it follows that N is *-sound. It holds by the definition of pAND net that the places in O_N have no outgoing edges. Therefore by Lemma 13 it follows that N is sub-sound. ∎

Corollary 28. *All AND-OR nets are sub-sound.*

Proof: By the Theorem 24, Theorem 25, Theorem 26 and Theorem 27 the initial nets are all sub-sound, and by Corollary 23 substitution preserves sub-soundness. ∎

VI. Future Research

The class of AND-OR nets can be researched further in several ways. One direction could be to attempt to characterize the class in terms of syntactic and semantic properties. As was shown all the nets in it are sound, even sub-sound, and it is also not hard to see that they are all free-choice nets, but it certainly not true that the class contains

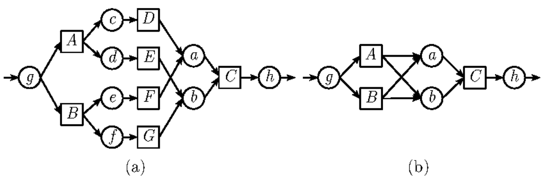


Figure 12. Counterexample for the completeness of AND-OR nets

all sub-sound free-choice nets as is show in Theorem 29. So it remains open which semantic property characterizes the AND-OR nets.

Theorem 29. *Not all free-choice sub-sound workflow nets are AND-OR nets.*

Proof: The counterexample is given in Figure 12 (a) (taken from [12]). ■

Another potential research direction is the extension of the class by introducing new forms of substitution that still can be considered hierarchical. For example, it might be allowed that not only substitute nodes but also edges: an edge from a place to a transition could be replaced with a workflow net starting with a single place and ending with a single transition. In general such substitutions do not preserve sub-soundness, but they can be syntactically restricted such that they do. To illustrate, such substitutions could be used to generate Figure 12 (a) from the AND-OR net in Figure 12 (b) by substituting the edges (A, a), (A, b), (B, a) and (B, b).

Yet another possible generalization can be achieved by weakening the requirement that a substitution links all the input and output nodes in the same way. For example, it could be allowed that a transition is replaced with a tAND net with a single input transition and several output transitions such that (1) each output transition in the tAND net is linked to at least one place in the postset of the replaced transition and (2) each place in the postset of the replaced transition is linked with exactly one output transition in the tAND net. Also this would allow us to generate Figure 12 (a) from the AND-OR net in Figure 12 (b) by substituting the transitions A and B.

VII. Conclusions

We have presented an approach for designing sound workflow nets in a structured way. This method is based on the notion of a substitution of one node by a workflow net with input and output nodes being of the same type as the substituted node. The substituted nets can have multiple inputs and outputs, which is an extension to the previously considered substitutions as it allows to generate more general class of nets. We have identified a notion of soundness that is preserved by such substitutions and corrected a small omission in an earlier similar method.

Acknowledgements We would like to thank the reviewers for their very constructive and detailed comments which have helped to improve this paper.

References

[1] Barkaoui A., R Ben Ayed, and Sbai Z. Workflow soundness verification based on structure theory of petri nets. *International Journal of Computing & Information Sciences*, 5(1):51–61, 2007.

[2] Wilfried Brauer, Robert Gold, and Walter Vogler. A survey of behaviour and equivalence preserving refinements of petri nets. *Advances in Petri Nets 1990*, pages 1–46, 1991.

[3] Piotr Chrzastowski-Wachtel, Boualem Benatallah, Rachid Hamadi, Milton O'Dell, and Adi Susanto. A top-down petri net-based approach for dynamic workflow modeling. In *Proceedings of BPM'03*, pages 336–353, Berlin, Heidelberg, 2003. Springer-Verlag.

[4] Jorg Desel and Javier Esparza. *Free Choice Petri Nets (Cambridge Tracts in Theoretical Computer Science)*. Cambridge University Press, New York, NY, USA, 2005.

[5] Javier Esparza and Manuel Silva. On the analysis and synthesis of free choice systems. In Grzegorz Rozenberg, editor, *Applications and Theory of Petri Nets*, volume 483 of *Lecture Notes in Computer Science*, pages 243–286. Springer, 1989.

[6] Hejiao Huang, To Y. Cheung, and Wai M. Mak. Structure and behavior preservation by petri-net-based refinements in system design. *Theoretical Computer Science*, 328(3):245–269, December 2004.

[7] J. Padberg, M. Gajewsky, and C. Ermel. Rule-based refinement of high-level nets preserving safety properties. *Science of Computer Programming*, 40(1):97–118, 2001.

[8] Zhang Peng, Qi Mei, and Liu Xikui. The behavior properties of refinement of petri nets and application in parallel programming. *Computer Science and Information Engineering, World Congress on*, 3:330–333, 2009.

[9] Ichiro Suzuki and Tadao Murata. A method for stepwise refinement and abstraction of petri nets. *Journal of Computer and System Sciences*, 27(1):51–76, 1983.

[10] Wil M. P. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[11] Kees van Hee, Jan Hidders, Geert-Jan Houben, Jan Paredaens, and Philippe Thiran. On the relationship between workflow models and document types. *Inf. Syst.*, 34:178–208, March 2009.

[12] Kees M. van Hee, Natalia Sidorova, and Marc Voorhoeve. Soundness and separability of workflow nets in the stepwise refinement approach. In Wil M. P. van der Aalst and Eike Best, editors, *ICATPN*, volume 2679 of *Lecture Notes in Computer Science*, pages 337–356. Springer, 2003.