

Synthesis and reduction of state machine workflow nets

Citation for published version (APA):

Corro Ramos, I., Di Bucchianico, A., Hakobyan, L., & Hee, van, K. M. (2006). *Synthesis and reduction of state machine workflow nets*. (Computer science reports; Vol. 0618). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2006

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Synthesis and Reduction of State Machine Workflow Nets

Isaac Corro Ramos, Alessandro Di Bucchianico, Lusine Hakobyan, Kees van Hee

Department of Mathematics and Computer Science

Technische Universiteit Eindhoven

P.O.Box 513

5600 MB Eindhoven, The Netherlands

`{i.corro.ramos,a.d.bucchianico,l.hakobyan,k.m.v.hee}@tue.nl`

Abstract

We present two rules for generating State Machine Workflow (SMWf) nets. We prove the soundness of these rules, *i.e.*, we show that applying these rules to an SMWf net results in an SMWf net. We also prove completeness, *i.e.*, we show that any SMWf net can be obtained by applying our rules to the initial SMWf net containing one transition only. We present two proofs for completeness, one based on synthesis and one based on reduction.

Keywords: State Machine Workflow net, Petri net, soundness, completeness, construction rules, redundancy.

1 Introduction

Petri nets are known as abstract models of concurrent systems. The interest for Petri nets is caused by their multiple applications in very different areas, including workflow management, data analysis, logistics, diagnosis, reliability engineering, concurrent programming and software design. Petri nets can be synthesized out of smaller nets by applying rules and vice versa larger nets can be reduced to smaller nets by inverted rules. In both cases the rules preserve some properties of original nets like liveness or boundedness.

The specific application of Petri nets that led to this paper is the following. We are developing adaptive software release procedures based on the architecture of a software system. Since we need processes that have a clear start and end, we model software systems as State Machine Workflow (SMWf) nets. SMWf nets are a special class of Petri nets that possess one initial place and one final place where each transition has one incoming and one outgoing arc. In order to test our release procedures we need to randomly generate SMWf nets. We will show in this paper that this can be accomplished by using two construction rules. We show that these rules are sound, *i.e.*, these rules transform SMWf nets into SMWf nets. We also prove completeness, *i.e.*, we show that any SMWf net can be obtained by applying our rules to an initial SMWf net containing one transition only. Furthermore we describe reduction techniques based on the inversion of our rules, which may be used to remove redundancies in SMWf nets.

Several papers have appeared on reduction and synthesis rules of Petri nets and subclasses of Petri nets. Most papers present a set of transformation rules that preserve properties like boundedness and liveness for a specific subclass of Petri nets. For example, sets of reduction rules preserving properties for state machines and marked graphs were presented in Berthelot (1978) and extended to general Petri nets in Berthelot (1987). Desel (1990) studies sets of rules which reduce live and safe free choice nets to live and safe marked graphs or state machine graphs. Synthesis and reduction rules for well-formed free choice nets are considered in the monograph Desel and Esparza (1995). Properties of stepwise refinement, an important synthesis rule for Petri

nets, are studied in Suzuki and Murata (1983) and Valette (1979). A comprehensive overview on reduction rules for Petri nets can be found in Murata (1989). In this article we study two synthesis rules, Transition Addition and Transition Refinement, for generating SMWf nets. Similar rules can be found in Berthelot (1978), Murata (1989), Suzuki and Murata (1983) and Valette (1979). Unlike most of the papers quoted above, we are not interested in preservation of properties like boundedness and liveness, but we wish to prove our rules generate all SMWf nets in a similar way as free choice nets are generated in Desel and Esparza (1995). We present two sound and complete synthesis rules for SMWf nets. We prove that starting from an atomic SMWf net, *i.e.*, an SMWf net containing one transition only, all SMWf nets can be synthesized by repeated application of the rules, *i.e.*, we provide a method which uses these rules to construct any SMWf net. Therefore we present techniques to simplify complex processes which are modelled as SMWf nets. Furthermore, we define two reduction rules as the inverse of the synthesis rules and prove that we can remove redundancies from SMWf nets applying these reduction rules. Finally we prove that also the reduction rules are complete, *i.e.*, by repeated application of the rules any SMWf net can be reduced to an atomic SMWf net.

The rest of this paper is structured as follows. In Section 2 we introduce the notation we use throughout the paper as well as present some basic definitions and lemmas that are used in the proofs of the main results. In Section 3 we study construction rules of SMWf nets, Transition Addition and Transition Refinement, and we prove the soundness of these rules. In Section 4 and in Section 5 we prove that the rules are also complete. A completeness proof by synthesis is presented in Section 4. In Section 5 we describe reduction techniques to remove redundancies in SMWf nets. With these soundness preserving rules we can build any SMWf net, starting from an atomic SMWf net, using Transition Addition and Transition Refinement. In Section 6 we conclude discussing the obtained results and proposing directions for future work.

2 Preliminaries

As mentioned in the introduction, State Machine Workflow nets are a special class of Petri nets. The definitions and results of this section, however, refer to general Petri nets. Therefore it seems to be logical to start this section with the definition of Petri nets.

Definition 2.1 (Petri net). *A Petri net is a tuple $N = (P, T, F)$ where:*

- *P and T are disjoint non-empty finite sets, the elements of which are called places and transitions, respectively.*
- *$F \subseteq (P \times T) \cup (T \times P)$ is a non-empty finite set, the elements of which are called arcs.*

A node of N is an element of $P \cup T$.

We ignore the tokens and firing rules that are associated with Petri nets because in this paper we are only interested in the structure of Petri nets.

Definition 2.2 (Preset and postset). *Let $N = (P, T, F)$ be a Petri net and let x be a node of N . The set $\{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x (denoted by $\bullet x$). The set $\{z \in P \cup T \mid (x, z) \in F\}$ is called the postset of x (denoted by $x\bullet$).*

In the synthesis proof of completeness we need to enlarge nets. Therefore it is convenient to introduce the concepts of subnets and supernets.

Definition 2.3 (Subnet). *Let $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$ be two Petri nets. We say that N_1 is a subnet of N_2 , or equivalently N_2 is a supernet of N_1 , if and only if $P_1 \subseteq P_2$, $T_1 \subseteq T_2$ and $F_1 \subseteq F_2$. We denote this situation by $N_1 \subseteq N_2$.*

Since in the synthesis proof of completeness we make copies of SMWf nets, we need to introduce the concept of isomorphic nets.

Definition 2.4 (Isomorphic nets). *Two Petri nets $N = (P, T, F)$ and $N' = (P', T', F')$ are isomorphic, denoted by $N \cong N'$, if and only if there exists a bijection f from $P \cup T$ to $P' \cup T'$ such that*

1. $f(P) = P'$ and $f(T) = T'$
2. $\forall x, y \in P \cup T : (x, y) \in F \Leftrightarrow (f(x), f(y)) \in F'$

In our proofs we need the concept of a path in a Petri net.

Definition 2.5 (Path in a Petri net). *A path s in a Petri net $N = (P, T, F)$ is either an empty sequence, denoted by ε , a single node $s = (s_1)$, $s_1 \in P \cup T$, or a sequence of nodes $s = (s_1, \dots, s_n)$ such that $(s_i, s_{i+1}) \in F$ for all $1 \leq i < n$. If we want to distinguish between places and transitions we write $s = (p_1, t_1, \dots, p_n, t_n)$ where $p_i \in P$ and $t_i \in T$ for all $1 \leq i < n$. Sometimes we denote by P_s , T_s and F_s the set of places, transitions and arcs of the path s , respectively. The predicate $\text{path}_N(s)$ indicates whether s is a path in N or not. If there is no confusion possible, then we write $\text{path}(s)$ instead of $\text{path}_N(s)$. We will use the notations $\text{first}(s) = s_1$ and $\text{last}(s) = s_n$ for the first and the last node of a path, respectively. A node x belongs to the path s if and only if $x = s_i$ for some i with $1 \leq i \leq n$.*

It is convenient to denote whether two nodes in a Petri net can be joined by a path.

Definition 2.6 (Path between two nodes). *The predicate $\text{Path}_N(a, b)$ denotes whether two nodes a and b in a Petri net $N = (P, T, F)$ can be joined by a path, i.e., there is a path $s = (s_1, \dots, s_n)$ in N such that $s_1 = a$ and $s_n = b$. If there is no confusion possible, then we write $\text{Path}(a, b)$ instead of $\text{Path}_N(a, b)$.*

We now introduce a subclass of Petri nets known as State Machine Workflow (SMWf) nets.

Definition 2.7 (State Machine Workflow net). *A State Machine Workflow (SMWf) net is a Petri net $N = (P, T, F)$ with two special places i and f , which are called initial place and final place, respectively, such that*

1. $\forall t \in T : |\bullet t| = |t \bullet| = 1$
2. $\bullet i_N = \emptyset$
3. $f_N \bullet = \emptyset$
4. $\forall x \in P \cup T : \text{Path}(i_N, x) \wedge \text{Path}(x, f_N)$

The preset and postset of a transition t each contain exactly one element, which we denote by $\bullet t$ and $t \bullet$, respectively. If there is no confusion possible, then we write i and f instead of i_N and f_N . We call the nodes of an SMWf net except i and f the internal nodes of the net.

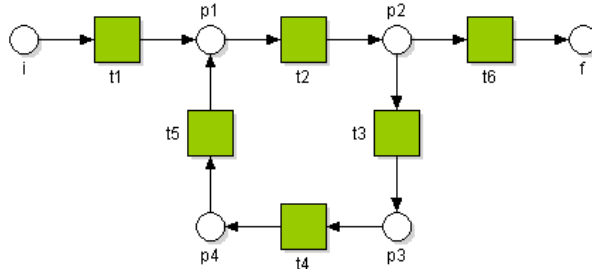


Figure 1: Example of an SMWf net containing both simple and non-simple paths.

In our proofs we sometimes need to remove redundant parts of a path like cycles. It is therefore convenient to have the notion of simple path.

Definition 2.8 (Simple path). *A path $s = (s_1, \dots, s_n)$ in a Petri net $N = (P, T, F)$ is said to be a simple path if and only if for all $1 \leq i, j \leq n$ we have that $s_i = s_j$ implies $i = j$ or $\{i, j\} = \{1, n\}$.*

Observe that a simple path is a path where all its nodes are not repeated except occasionally the first and the last. In Figure 1 consider the path $s = (i, t1, p1, t2, p2, t3, p3, t4, p4, t5, p1, t2, p2, t6, f)$. Since $p1, t2$ and $p2$ are repeated, s is not a simple path. If we consider $s' = (i, t1, p1, t2, p2, t6, f)$ then all its nodes are not repeated. If we consider $s'' = (p1, t2, p2, t3, p3, t4, p4, t5, p1)$ then only the first and the last nodes are repeated. Therefore s' and s'' are simple paths.

In every non-simple path we may find a simple (sub)path by removing internal cycles. In order to state this properly, we first introduce the concept of subpath.

Definition 2.9 (Subpath of a Petri net). *Let s be a path in a Petri net $N = (P, T, F)$. Then s' is a subpath of s if and only if s' is a path in N and $P_{s'} \subseteq P_s, T_{s'} \subseteq T_s$ and $F_{s'} \subseteq F_s$.*

The following lemma shows that in every non simple path we can always find a simple subpath. In the previous example we saw that s' is a simple subpath of s . Before proving the lemma we introduce the next two definitions.

Definition 2.10 (Length of a path). *The length of a path s in a Petri net $N = (P, T, F)$, denoted by $|s|$, is the number of nodes of s . For the empty path ε the length is defined as $|\varepsilon| = 0$.*

We now show that paths may be combined. We introduce the path gluing operation, *i.e.*, if we have two paths where the last node of the first path is the first node of the second path, then we can glue them together.

Definition 2.11 (Path gluing operation). *Let $s = (s_1, \dots, s_n)$ and $u = (u_1, \dots, u_m)$ be two paths in a Petri net $N = (P, T, F)$ such that $s_n = u_1$. We define the glued path of s and u by $s \circ u = (s_1, \dots, s_n, u_2, \dots, u_m)$. We called \circ the gluing operator.*

It is easy to see that the glued path is in fact a path. Indeed, since $(s_{n-1}, s_n) \in F$ and $(u_1, u_2) \in F$, it follows that $(s_1, \dots, s_{n-1}, s_n, u_2, u_3, \dots, u_m)$ is also a path in N . Hence, path gluing is a well-defined operation. Observe that $|s \circ u| = |s| + |u| - 1$.

Lemma 2.12 (Simple subpath). *Let $s = (s_1, s_2, \dots, s_n)$ be a path in a Petri net $N = (P, T, F)$. If s is not simple, then there exists a simple subpath \tilde{s} of s such that $first(\tilde{s}) = first(s)$ and $last(\tilde{s}) = last(s)$.*

Proof. Since s is not simple, there exists i and j such that $1 \leq i < j \leq n$ and $s_i = s_j$. Define $s' = (s_1, \dots, s_i)$ and $s'' = (s_j, \dots, s_n)$. Direct verification of Definition 2.9 shows that both s and s' are subpaths of s . Now consider $u = s' \circ s''$. Obviously, u is a subpath of s such that $first(u) = first(s)$, $last(u) = last(s)$ and $0 < |u| < |s|$. Since the length of a path is a finite nonnegative number, repeated application of this procedure yields a subpath of s without repeated nodes after a finite number of steps. \square

In Section 5 we describe reduction techniques to remove redundancies in SMWf nets. The following concepts appear in the proof of one of the most important results of that section.

Definition 2.13 (Circuit). *A circuit c in a Petri net $N = (P, T, F)$ is a pair of simple paths $s = (s_1, \dots, s_n)$ and $u = (u_1, \dots, u_m)$ such that $s_1 = u_1, s_n = u_m, s_1 \neq s_n$ and $s_i \neq u_j$ for all $1 < i < n, 1 < j < m$. We will use the notations $b = s_1$ and $e = s_n$ for the first and the last node of a circuit, respectively. We call the nodes of the circuit except b and e internal nodes of the circuit.*

Definition 2.14 (Cycle). *A non-empty path $s = (s_1, \dots, s_n)$ in a Petri net $N = (P, T, F)$ is a cycle if and only if $s_1 = s_n$.*

In the next definition we introduce the concept of self loop which is the simplest case of a cycle.

Definition 2.15 (Self loop). *A Petri net $N = (P, T, F)$ contains a self loop if and only if there is a node x in N such that $x \bullet \cap \bullet x \neq \emptyset$.*

Observe that if we consider an SMWf net $N = (P, T, F)$ then a self loop is of the form (p, t, p) where $p \in P$ and $t \in T$.

Depending on whether a Petri net contains cycles or not we can classify it as a cyclic or an acyclic Petri net.

Definition 2.16 (Acyclic Petri net). *A Petri net $N = (P, T, F)$ is acyclic if it does not contain cycles.*

In order to compute the number of incoming and outgoing arcs of a node in a Petri net we introduce the concept of degree of a node.

Definition 2.17 (Degree of a node). *A node x of a Petri net $N = (P, T, F)$ has a degree k if and only if $|\bullet x| + |x \bullet| = k$. We denote the degree of the node x in a Petri net N by $\deg_N(x)$. If there is no confusion possible, we write $\deg(x)$ instead of $\deg_N(x)$.*

Note that by Definition 2.7 the degree of any transition in an SMWf net is two. We finish this section by introducing a reduction technique that allows us to remove cycles from an SMWf net.

Definition 2.18 (Cycle Reduction). *Let $N = (P, T, F)$ be an SMWf net and $N' = (P', T', F')$ be a Petri net. If $s = (p_1, t_1, \dots, p_n, t_n, p_1)$ is a cycle in N , then N' is said to be the cycle reduction of N at s with a place \mathbf{p} if*

1. $P' = (P \setminus \{p_1, \dots, p_n\}) \cup \{\mathbf{p}\}$
2. $T' = T \setminus \{t_1, \dots, t_n\}$
3. $F' = (F \setminus \{(p_1, t_1), (t_1, p_2), \dots, (t_n, p_1)\}) \cup \left\{ (\mathbf{p}, t) \mid t \in \left(\bigcup_{i=1}^n p_i \bullet \right) \setminus \{t_1, \dots, t_n\} \right\} \\ \cup \left\{ (t, \mathbf{p}) \mid t \in \left(\bigcup_{i=1}^n \bullet p_i \right) \setminus \{t_1, \dots, t_n\} \right\}$

Let us remark that N' is indeed an SMWf net. Condition 1) of Definition 2.7 is satisfied since all the transitions in N' are transitions in N which is an SMWf net. Conditions 2) and 3) are also satisfied since it is not possible to have cycles containing i or f and therefore i and f remain the same for N and N' . In order to verify condition 4) let us consider a node x in N that does not belong to the cycle. Then there is a path in N from i to f through x . Either this path passes through the cycle or not. If the path does not pass through the cycle this is also a path in N' . If the path passes through the cycle in N then it enters the cycle in some node p_i and leaves it again say in node p_j . Therefore there is also a path from i to f through x via \mathbf{p} in N' . Now suppose that x belongs to the cycle. It means that x is one of the nodes reduced to \mathbf{p} . Then there is a path in N from i to f through x . Since the path passes through the cycle in N we already mentioned that there is a path in N' from i to f through \mathbf{p} . Hence, condition 4) holds. Figure 2 shows the cycle $c = (p_1, t_1, p_2, t_2, p_3, t_3, p_4, t_4)$ reduced to the place \mathbf{p} .

3 Construction rules and their soundness

Using the notation and the basic definitions of the previous section we are now ready to introduce the two main concepts of this paper: the transformation rules. We will show that these rules are sound, *i.e.*, these rules transform SMWf nets into SMWf nets.

We start with the Transition Refinement rule. The idea is to refine a transition by adding a simple path that does not belong to the SMWf net.

Definition 3.1 (Transition Refinement (TR)). *Let $N = (P, T, F)$ and $N' = (P', T', F')$ be two SMWf nets. If t is a transition of N and $s = (p_1, t_1, \dots, p_n, t_n)$ is a simple path in N' such that p_1, \dots, p_n and t_1, \dots, t_n are not in N , then N' is said to be the refinement of N at t with s if*

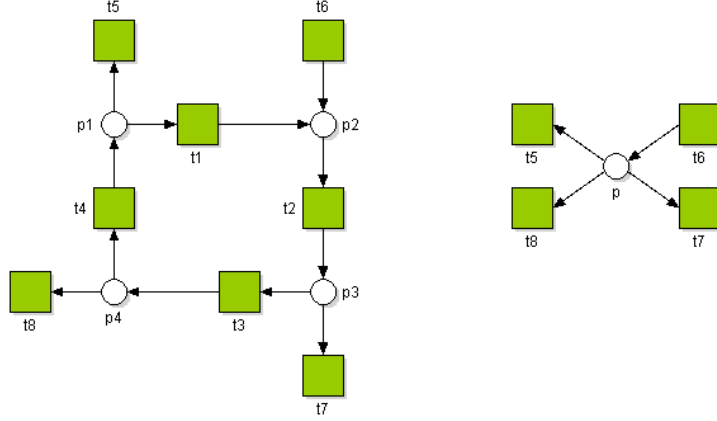


Figure 2: Cycle reduction at $s = (p_1, t_1, p_2, t_2, p_3, t_3, p_4, t_4, p_1)$ with p .

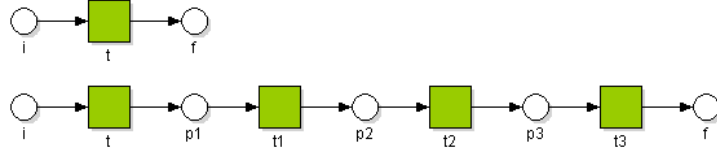


Figure 3: Refinement at t with $(p_1, t_1, p_2, t_2, p_3, t_3)$.

1. $P' = P \cup \{p_1, \dots, p_n\}$
2. $T' = T \cup \{t_1, \dots, t_n\}$
3. $F' = (F \setminus \{(t, \underline{t})\}) \cup \{(t, p_1), (p_1, t_1), \dots, (p_n, t_n), (t_n, \underline{t})\}$

We denote the resulting SMWf net by $N' = TR(N, t, s)$.

Unlike in other definitions of refinement (see Suzuki and Murata (1983)) we do not delete the transition to be refined; in fact we do arc refinement.

It is easy to see that for the empty path ε we have $TR(N, t, \varepsilon) = N$. Figure 3 shows an example of a small SMWf net before (top) and after (bottom) applying the Transition Refinement rule.

The following lemma states the refinement of a transition t recursively.

Lemma 3.2 (Recursion for Transition Refinement). *Let N' be a transition refinement of a Petri net $N = (P, T, F)$ at t with $s = (p_1, t_1, \dots, p_n, t_n)$. Then we have*

$$TR(N, t, s) = TR(TR(N, t, (p_1, t_1)), t_1, (p_2, \dots, t_n))$$

Proof. Let $N' = (P', T', F')$ be the transition refinement of N at t with $s = (p_1, t_1, \dots, p_n, t_n)$. First note that $TR(N, t, (p_1, t_1))$ is an SMWf net $N^* = (P^*, T^*, F^*)$, where $P^* = P \cup \{p_1\}$, $T^* = T \cup \{t_1\}$, $t \in T$ and $F^* = (F \setminus \{(t, \underline{t})\}) \cup \{(t, p_1), (p_1, t_1), (t_1, \underline{t})\}$. Similarly it follows that $TR(N^*, t_1, (p_2, t_2, \dots, p_n, t_n))$ is an SMWf net $N'' = (P'', T'', F'')$ where

$$\begin{aligned} P'' &= P^* \cup \{p_2, \dots, p_n\} = P \cup \{p_1, p_2, \dots, p_n\}, \\ T'' &= T^* \cup \{t_2, t_3, \dots, t_n\} = T \cup \{t_1, t_2, \dots, t_n\}, \\ F'' &= (F^* \setminus \{(t_1, \underline{t_1})\}) \cup \{(t_1, p_2), (p_2, t_2), \dots, (t_n, \underline{t_1})\} \\ &= (F \setminus \{(t, \underline{t})\}) \cup \{(t, p_1), (p_1, t_1), (t_1, \underline{t})\} \setminus \{(t_1, \underline{t})\} \cup \{(t_1, p_2), (p_2, t_2), \dots, (t_n, \underline{t_1})\} \\ &= (F \setminus \{(t, \underline{t})\}) \cup \{(t, p_1), (p_1, t_1), (t_1, p_2), (p_2, t_2), \dots, (p_n, t_n), (t_n, \underline{t})\}. \end{aligned}$$

Hence, by Definition 3.1 we have that $N'' = N'$. \square

We now introduce the second rule which is Transition Addition. The idea is to add a “parallel” transition between two places of an SMWf net. The initial and final places are partially excluded because that would violate the SMWf property.

Definition 3.3 (Transition Addition (TA)). *Let $N = (P, T, F)$ and $N' = (P', T', F')$ be two SMWf nets. If $b \neq f$ and $e \neq i$ are places in N , then N' is said to be the Transition Addition of N at (b, e) with t if*

1. $P' = P$
2. $T' = T \cup \{t\}$
3. $F' = F \cup \{(b, t), (t, e)\}$

We write $N' = TA(N, b, t, e)$.

Figure 4 shows an example of the application of the Transition Addition rule (bottom) at the pairs of places (i, p_1) , (p_2, p_1) and (p_3, p_3) with the transitions t_5 , t_6 and t_7 , respectively.

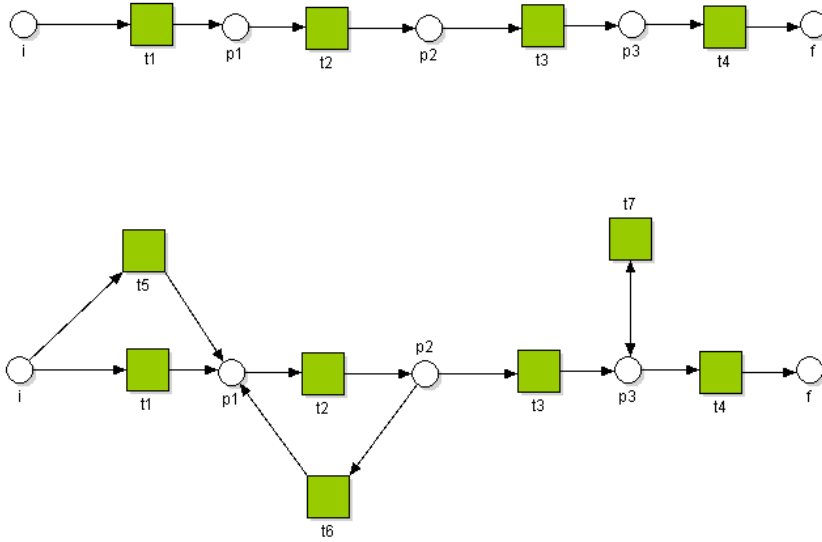


Figure 4: Transition Addition at (i, p_1) , (p_2, p_1) and (p_3, p_3) with t_5 , t_6 and t_7 , respectively.

We now prove the main result of this section: the soundness of the rules.

Theorem 3.4 (Soundness of construction rules). *Let $N = (P, T, F)$ be an SMWf net. If we apply either Transition Addition or Transition Reduction to N , then the resulting net is again an SMWf net.*

Proof. Suppose we apply Transition Refinement to N at t with $s = (p_1, t_1, \dots, p_n, t_n)$. Then $|\bullet t| = 1$ and $|t \bullet| = 1$. Since s is a simple path, the corresponding property also holds for all transitions on s . Hence, condition 1) of Definition 2.7 is satisfied for $TR(N, t, s)$. Conditions 2) and 3) are also satisfied because no incoming arcs were added to i and no outgoing arcs to f . In order to verify condition 4) note that, by the definition of Transition Refinement, it follows that $(t, p_1, t_1, \dots, p_n, t_n, t \bullet)$ is also a path in N . Hence, the existence of a path from i to f through t implies that all places and transitions of $TR(N, t, s)$ lie on a path from i to f .

Suppose we apply Transition Addition to N at (b, e) with t . Condition 1) is satisfied, because we added no arcs from or to transitions in N and by construction we have $|\bullet t| = 1$ and $|t \bullet| = 1$.

Conditions 2) and 3) are also satisfied because no incoming arcs were added to i and no outgoing arcs were added to f . Finally, note that since we are not removing arcs, we only need to verify condition 4) for t . By gluing a path from i to b with the path (b, t) there is a path from i to t . In the same way, we obtain a path from t to f by gluing the path (t, e) with a path from e to f . \square

4 Completeness proof by synthesis

In the previous section we introduced two transformation rules and proved that the class of SMWf nets is closed with respect to these rules. In this section we show by synthesis that these rules are also complete, *i.e.*, any SMWf net can be obtained by repeated application of the rules to an initial SMWf net containing one transition only.

Theorem 4.1 (Completeness of construction rules). *Any SMWf net can be obtained by repeatedly applying the Transition Addition and Transition Refinement rules of Definitions 3.3 and 3.1 to an initial SMWf net containing one transition only.*

Proof. Let $N = (P, T, F)$ be an SMWf net with initial place i and final place f . We construct an SMWf net N' that is isomorphic with N . For simplicity we use the same notations for places and transitions in N and N' .

Initial Step Take any transition $t \in T$ in the postset of i , so that $\bullet t = i$. Consider the SMWf net $N' := (P', T', F')$ where $P' = \{i, f\}$, $T' = t$ and $F' = \{(i, t), (t, f)\}$. If $N' = N$, then we are done. If $N' \neq N$, then we first have to enlarge N' so that it becomes a subnet of N . This can be achieved using Transition Refinement in the following way. Since N is an SMWf net and by assumption $(i, t) \in F$, there exists a path s in N from i to f through t . If s is not simple then by Lemma 2.12 there exists a simple subpath u . In this case, since $i = \bullet t$, we have that t is the first transition in u and we write $u = (i, t, p_1, t_1, \dots, p_n, t_n, f)$. Then we apply Transition Refinement at t with $(p_1, t_1, \dots, p_n, t_n)$. By construction N' is a subnet of N . Hence, we proved starting from an initial SMWf net with one transition from N only, applying the Transition Refinement rule we obtain a subnet $N' = (P', T', F')$ of N . If $N = N'$, then we are done. Otherwise, we move to the following iterative step.

Iterative Step Take any transition $t \in T \setminus T'$. Since N is an SMWf net, there exists a path s in N from i to f through t . Let t_0 be the left-most transition on this path that is not in T' . We now show that $b := \bullet t_0$ has to be in P' . If $b = i$ then there is nothing to prove. If $b \neq i$, then let \tilde{t} be a transition in s such that $\tilde{t} \bullet = b$. Since $\tilde{t} \in N'$ and N' is an SMWf net, there exists a path in N' from i to f through \tilde{t} . This path must also pass through b , since the postset of \tilde{t} is a singleton. Hence, $b \in P'$. Similar arguments yield that if t_n is the right-most transition on this path that is not in T' , then $e := t_n \bullet$ has to be in P' . We now consider the path u in N from b to e . If u is not simple then by Lemma 2.12 there exists a simple subpath w . Since $b = \bullet t_0$, we have that t_0 is the first transition in w and we write $w = (b, t_0, p_1, t_1, \dots, p_n, t_n, e)$. Note that $b, e \in P'$, but $p_i \in P \setminus P'$, ($i = 1, \dots, n$) and $t_j \in T \setminus T'$, ($j = 0, \dots, n$). If $(p_1, t_1, \dots, p_n, t_n) = \varepsilon$, then we enlarge N' by applying Transition Addition at (b, e) with t_0 . If $N' \neq N$, then we apply the iterative step again. If $(p_1, t_1, \dots, p_n, t_n) \neq \varepsilon$, then we enlarge N' by applying Transition Refinement at t_0 with $(p_1, t_1, \dots, p_n, t_n)$. By construction N' is a subnet of N . If $N' = N$, then we are done. If $N' \neq N$, then we apply the iterative step again.

In each step we apply Transition Addition and/or Transition Refinement and therefore the number of elements of $T \setminus T'$ decreases in every iteration. Since we proved that as long as $T' \neq T$ we can select a transition $t \in T \setminus T'$, after a finite number of steps we arrive at $T' = T$.

We also proved that in each iteration N' is a subnet of N . We do not introduce any new places and arcs because we always use Transition Addition, Transition Refinement and simple paths. It thus remains to prove that when we arrive at $T' = T$, it cannot happen that $P \setminus P' \neq \emptyset$ or $F \setminus F' \neq \emptyset$. Let p be an arbitrary place in N which is not i or f . Let t be any element of the

preset of p . Since $t \in T'$ and N' is an SMWf net, there exists a path in N' from i to f through t . Since p is the unique element of the postset of t , it follows that $p \in P'$. Since $T' = T$ and $P' = P$, any arc in F must be in F' because N' is an SMWf net and the presets and postsets of transitions in an SMWf net are singletons. \square

5 Completeness proof by reduction

In this section we present two *reduction* rules. Reduction means that we can reduce the number of nodes of an SMWf net while the resulting net is again an SMWf net. We show that these rules are complete, *i.e.*, we use the rules to verify whether a Petri net is an SMWf net. We apply the rules as long as possible and then we check if the final result is an SMWf net containing exactly one transition. Simple reduction rules for state machines and for general Petri nets can be found in Berthelot (1978) and Murata (1989), respectively.

We start with the definition of Transition Reduction, which can be viewed as the inverse version of the Transition Refinement rule of Definition 3.1.

Definition 5.1 (Transition Reduction (TRD)). *Let $N = (P, T, F)$ and $N' = (P', T', F')$ be two SMWf nets. If p is a place and t_1 and t_2 are transitions of N such that $\bullet p = \{t_1\}$ and $p\bullet = \{t_2\}$, then N' is said to be the Transition Reduction of N at (t_1, t_2) with t if*

1. $P' = P \setminus \{p\}$
2. $T' = (T \setminus \{t_1, t_2\}) \cup \{t\}$
3. $F' = (F \setminus \{(\bullet t_1, t_1), (t_1, p), (p, t_2), (t_2, t_2\bullet)\}) \cup \{(\bullet t_1, t), (t, t_2\bullet)\}$

We write $TRD(N, t_1, t_2, t)$.

Notice that the definition states that the place p between t_1 and t_2 must have exactly one incoming and one outgoing arc otherwise we cannot apply the rule. Figure 5 shows an example of a small SMWf net before (top) and after (bottom) applying the Transition Reduction rule. We observe that it is possible to apply the rule at (t_1, t_2) because p_1 has exactly one incoming and one outgoing arc whereas it is not possible to apply it at (t_2, t_3) or (t_2, t_4) because p_2 has two outgoing arcs.

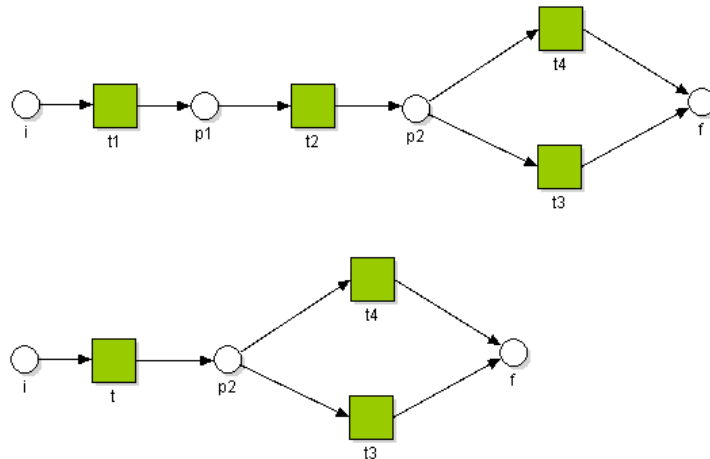


Figure 5: Transition Reduction at (t_1, t_2) with t .

We now introduce the second reduction rule. The idea is to remove “redundant” transitions of an SMWf net. The definition of redundant transition is stated as follows.

Definition 5.2 (Redundant Transition). *Let $N = (P, T, F)$ be an SMWf net. A transition $t \in T$ is said to be redundant if the net $N' = (P', T', F')$, where $P' = P$, $T' = T \setminus \{t\}$ and $F' = F \setminus \{(\bullet \underline{t}, t), (t, \underline{t} \bullet)\}$, remains an SMWf net.*

Note that the second rule, *i.e.*, removing redundant transitions, can be viewed as the inverse version of the Transition Addition rule of Definition 3.3. Let us remark that a self loop always contains a redundant transition.

The next lemma is a direct consequence of the definition of Transition Reduction.

Lemma 5.3. *Let $N = (P, T, F)$ be an SMWf net. By repeated application of the Transition Reduction rule we can reduce N to an SMWf net $N' = (P', T', F')$ in such a way that either $P' = \{i, f\}$, $T' = t$ and $F' = \{(i, t), (t, f)\}$, or for any place $p \in P' \setminus \{i, f\}$ we have $\deg(p) \geq 3$.*

Proof. Let us first assume that all places in N have exactly one incoming and one outgoing arc. Applying the Transition Reduction rule at $(\bullet p, p \bullet)$ for every place $p \in P \setminus \{i, f\}$ we end up with an SMWf net $N' = (P', T', F')$ such that $P' = \{i, f\}$, $T' = t$, and $F' = \{(i, t), (t, f)\}$.

Let us assume now that there exists at least one place $p_0 \in P \setminus \{i, f\}$ with exactly one incoming and one outgoing arc. Otherwise we had $\deg(p) \geq 3$ for all places $p \in P \setminus \{i, f\}$, and there is nothing to prove. Since p_0 has exactly one incoming and one outgoing arc, by applying Transition Reduction at $(\bullet p_0, p_0 \bullet)$ with t we remove p_0 . We can apply Transition Reduction as long as there are places in N with the same property as p_0 . Note that every time we apply the Transition Reduction rule we remove the places with exactly one incoming and one outgoing arc. Since N has a finite number of nodes, we end up with an SMWf net $N' = (P', T', F')$ such that $\deg(p) \geq 3$ for all places $p \in P' \setminus \{i, f\}$. \square

Lemma 5.3 is used as initial step for the next theorem. Since by repeated application of the Transition Reduction rule we reduce the number of places and transitions of an SMWf net, the next step is to check whether it is possible to continue reducing the SMWf net. We accomplish this by looking for redundant transitions.

Theorem 5.4. *Let $N = (P, T, F)$ be an SMWf net with at least 3 places. If $\deg(p) \geq 3$ for all $p \in P \setminus \{i, f\}$, then there exists at least one redundant transition in T .*

Before proving the theorem for any SMWf net we state and prove the following two lemmas for acyclic SMWf nets.

Lemma 5.5. *Let $N = (P, T, F)$ be an acyclic SMWf net satisfying the conditions of Theorem 5.4. Then there exists at least one circuit in N .*

Proof. Suppose the initial place i of N has only one outgoing arc, *i.e.*, we have $\deg(i) = 1$. Then the first place after i , p say, has exactly one incoming arc (and therefore at least two outgoing arcs). Otherwise p had at least two incoming arcs however since the net is acyclic we had $\deg(i) \geq 2$ which contradicts the fact that $\deg(i) = 1$. Therefore i or p have at least two outgoing arcs. Since N is an SMWf net it follows that i or p are the initial places of at least two paths to f . Any pair of these paths are either disjoint (except for i or p and f) or there is a first common place p' . Therefore by Definition 2.13 there exists a circuit with i or p as initial places and p' or f as final places. \square

Lemma 5.6. *Let $N = (P, T, F)$ be an acyclic SMWf net satisfying the conditions of Theorem 5.4. Then there exists at least one redundant transition in N .*

Proof. By Lemma 5.5 there exists a circuit c in N . Call the initial and final nodes of the circuit b and e , respectively. Suppose c has exactly two places then there are two paths s and u such that $s = (b, t_1, e)$ and $u = (b, t_2, e)$. By Definition 5.2 we have that both t_1 and t_2 are redundant. If c has three places then without loss of generality we may consider $s = (b, t_1, p_2, t_2, e)$ and $u = (b, t_3, e)$. Then by Definition 5.2 we have that t_3 is redundant. Now let us assume that c contains at least four places. Let the internal places belonging to s be p_1, p_2, \dots, p_n , $n \geq 1$, and the internal places belonging to u be q_1, q_2, \dots, q_m , $m \geq 1$. Since N is an SMWf net there exists a path from i to b and a path from e to f that are disjoint from the circuit. Let us now distinguish two cases.

Case 1 All places p_1, \dots, p_n have exactly one incoming arc, therefore they have at least two outgoing arcs. In this case we claim that the transition t between p_n and e on s is redundant. To prove this let us consider N' the net after deletion of t . We have to show that N' remains SMWf net. Since we only deleted one transition we only have to verify condition 4) of Definition 2.7. Let us consider an arbitrary node x from N' . Since x is also a node in N then there is a path v_N from i to x and from x to f . If t is not on v_N then v_N is also a path in N' and therefore condition 4) holds. Suppose now that t is on v_N . Either t is on the path from i to x or on the path from x to f . Since p_1, \dots, p_n have exactly one incoming arc the path from i to x or the path from x to f enters c via b . Thus there is one other path w_N from i to x or from x to f via q_1, q_2, \dots, q_m (that does not contain t). Therefore w_N is also a path in N' . Hence, condition 4) is satisfied and N' is an SMWf net.

Case 2 The place $p_k, 1 \leq k \leq n$, is the first place in s with at least two incoming arcs. Then we claim that the transition \hat{t} from p_{k-1} to p_k on s , where $p_0 = b$, is redundant. As in the previous case let us consider N' the net after deletion of \hat{t} and show that N' remains SMWf net. To verify condition 4) of Definition 2.7 let us consider an arbitrary node x from N' . Since x is also a node in N then there is a path \hat{v}_N from i to x and from x to f . If \hat{t} is not on \hat{v}_N then \hat{v}_N is also a path in N' and therefore condition 4) holds. Suppose now that \hat{t} is on \hat{v}_N on the subpath from i to x . Since p_k has at least one incoming arc that does not belong to c there exists at least one other path \hat{w}_N from i to p_k which does not contain \hat{t} (otherwise we have a cycle). Therefore \hat{w}_N is also a path in N' . Hence, condition 4) is satisfied and N' is an SMWf net. Finally suppose that \hat{t} is on \hat{v}_N on the subpath from x to f . Since p_1, \dots, p_{k-1} have exactly one incoming arc the path from x to f enters c via b . Thus there is one other path \tilde{w}_N from x to f via q_1, q_2, \dots, q_m (that does not contain \hat{t}). Therefore \tilde{w}_N is also a path in N' . Hence, condition 4) is satisfied and N' is an SMWf net. \square

By Lemma 5.6 it follows that Theorem 5.4 is already proved for acyclic SMWf nets. For general (cyclic) SMWf nets we prove it considering cycle reductions and the following remark. Given a cyclic SMWf net N with a cycle s , by Definition 2.18 the degree of the reduced place \mathbf{p} satisfies

$\deg(\mathbf{p}) = \sum_{i=1}^k (\deg(p_i) - 2)$, where $k \geq 1$ is the number of places of s . If N satisfies the conditions of the theorem (all its internal places have degree at least three) and $k \geq 3$ then the following inequality holds:

$$\deg(\mathbf{p}) = \sum_{i=1}^k (\deg(p_i) - 2) = \sum_{i=1}^k \deg(p_i) - 2k \geq 3k - 2k = k \geq 3 \quad (1)$$

Observe that in general the inequality is not true for $k = 2$. If we consider a cycle with two places p_1 and p_2 with $\deg(p_1) = \deg(p_2) = 3$ then $\deg(\mathbf{p}) = \deg(p_1) + \deg(p_2) - 4 = 2$. This case is illustrated in Figure 6. However if we consider a cycle with two places p_1 and p_2 such that $\deg(p_1) > 3$ or $\deg(p_2) > 3$ then the previous inequality holds. This case is illustrated in Figure 7.

Now we continue with the proof of Theorem 5.4.

Proof of Theorem 5.4:

Proof. Given any SMWf net $N = (P, T, F)$ satisfying the conditions of the theorem, we prove by induction on the number of internal places n that N has at least one redundant transition.

Basic step Suppose $n = 1$. Assume that N is an SMWf net such that $P = \{i, p_1, f\}$, and $\deg(p_1) \geq 3$. Now assume that $|p_1 \bullet| \geq 2$. The case $|\bullet p_1| \geq 2$ can be treated similarly. Then there exists at least two transitions t and t' in T with $t \neq t'$, such that (p_1, t) and (p_1, t') are in F . Since N is an SMWf net with only one internal place either both t and t' are directly connected to f , *i.e.*, (t, f) and (t', f) are in F , or one of them, say t , is in a self loop with p_1 , *i.e.*, (p_1, t)

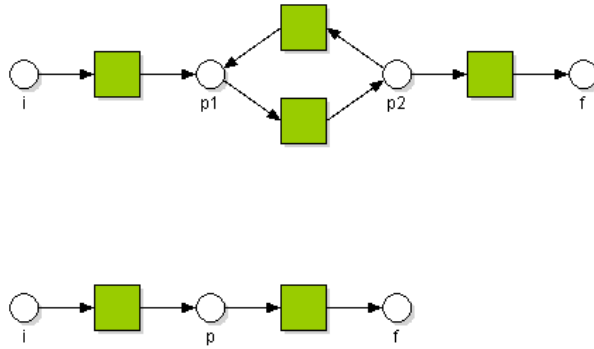


Figure 6: Cycle reduction with two places having degree three.

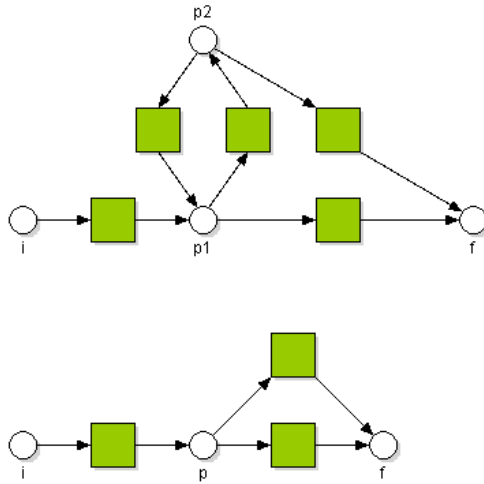


Figure 7: Cycle reduction with two places having p_1 degree four and p_2 degree three.

and (t, p_1) are in F . In the first case both t and t' are redundant and in the second case only t is redundant.

Induction hypothesis Assume that any SMWf net with n internal places such that all of them have degree greater or equal than three contains at least one redundant transition.

Statement to show in induction hypothesis Let $N = (P, T, F)$ be an SMWf net with $n + 1$ internal places such that all of them have at least degree three. We want to show that N has at least one redundant transition.

Proof of induction step If N is acyclic then by Lemma 5.6 there exists a redundant transition in N . Suppose that N is a cyclic SMWf net with a cycle s and let the places in s be $p_1, p_2, \dots, p_k, p_{k+1}$, where $1 \leq k \leq n + 1$ and $p_{k+1} = p_1$. If $k = 1$ we have a self loop and the corresponding transition is redundant. If $k = 2$ we distinguish two cases. First consider p_1 and p_2 such that $\deg(p_1) = \deg(p_2) = 3$. Take a path from i to f through s . This path either enters s via p_1 and leaves it via p_2 or enters s via p_2 and leaves it via p_1 . Then by Definition 5.2 we have that t_1 or t_2 is redundant. Suppose now that p_1 and p_2 are such that $\deg(p_1) > 3$ or $\deg(p_2) > 3$.

For this case and $k \geq 3$ we proceed as follows. Let us consider N' as the cycle reduction of N at s with \mathbf{p} . Then by Definition 2.18 it follows that N' is SMWf net and by inequality (1) we have that $\deg(\mathbf{p}) \geq 3$ and then N' also satisfies the conditions of the theorem. Therefore by the induction hypothesis N' has a redundant transition t . Let \tilde{N} and \tilde{N}' be the nets after deletion of t . Note that since t is a redundant transition of N' we have that \tilde{N}' is also an SMWf net. We now show that \tilde{N} is an SMWf net and thus t is also redundant in N . Since we only deleted one transition we only have to check whether condition 4) of Definition 2.7 is satisfied or not. Let us consider an arbitrary node x in \tilde{N} that does not belong to s . Since x is also a node in \tilde{N}' then there exists a path in \tilde{N}' from i to f through x . Either this path passes through \mathbf{p} or not. If the path does not pass through \mathbf{p} in \tilde{N}' then this is also a path in \tilde{N} and therefore condition 4) holds. If the path passes through \mathbf{p} in \tilde{N}' then it enters s in some node p_i and leaves it say in node p_j in \tilde{N} . Note that p_i and p_j can be the same place. Therefore there is also a path from i to f through x in \tilde{N} and thus condition 4) is satisfied. Suppose now that x belongs to s . It means that x is one of the nodes reduced to \mathbf{p} in \tilde{N}' . Then there is a path in \tilde{N}' from i to f through \mathbf{p} . The path in \tilde{N}' from i to f enters s in some node p_i passes through x and leaves it say in node p_j . Note that x can be p_i or p_j . Then there is also a path from i to f through x in \tilde{N} and thus condition 4) is again satisfied. Therefore we have shown that each node of \tilde{N} is on a path from i to f . Hence, t is a redundant transition of N . \square

Figure 8 shows three examples of cycles with two places where the first (top) and the second (middle) cannot occur in an SWMf net. In the first case it is impossible to leave the cycle and in the second case it is impossible to enter. Only the third case (bottom) can occur and then t_1 or t_2 is redundant.

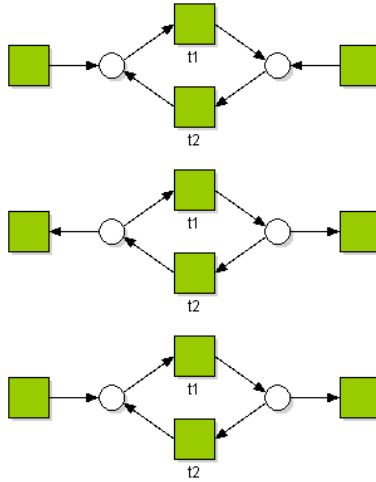


Figure 8: Example of cycles with two places.

The proof of Theorem 5.4 is based on reduction arguments although there is a different and shorter proof based on synthesis arguments. We accomplish this following the steps of the proof of Theorem 4.1.

Lemma 5.7. *Construction proof of Theorem 5.4.*

Proof. We show in Theorem 4.1 that we can construct every SMWf net by applying Transition Addition and Transition Refinement rules in a certain sequence. Applying Transition Refinement gives at least one place with degree two. Since according to the conditions of Theorem 5.4 all the places have degree greater or equal than three the last rule to be applied should be Transition Addition. Since in every step of the construction procedure (see proof of Theorem 4.1) we have an SMWf net the last application of Transition Addition gives a redundant transition. \square

To conclude this section let us remark that we showed in Lemma 5.3 that, given an SMWf net, we can apply TRD rule to obtain a new SMWf net with less nodes. Directly from the application of Transition Reduction rule, we proved in Theorem 5.4 that we can always find at least one transition that is redundant. Therefore, we have introduced two different methods to reduce the number of nodes of an SMWf net. Let us illustrate this with an example.

Example 5.8. *This is an example of an application of both Lemma 5.3 and Theorem 5.4. Consider the SMWf net from Figure 9 (top). Places p_2 and p_4 have one incoming and one outgoing arc. Then, according to Lemma 5.3, we can apply Transition Reduction at (t_2, t_4) with t and at (t_6, t_7) with t^* .*

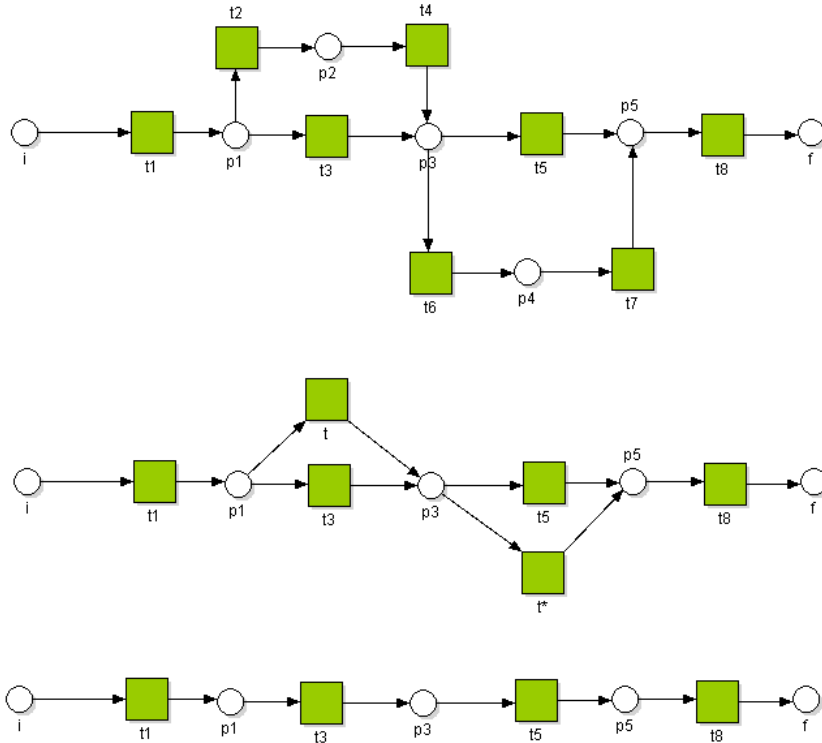


Figure 9: Transition Reduction at (t_2, t_4) with t and at (t_6, t_7) with t^* (middle) and the resulting SMWf net after removing the redundant transitions (bottom).

The resulting SMWf net appears in Figure 9 (middle). All the places of this net, except the initial and the final, have three or more arcs.

Since all the places of this net have three or more arcs, according to Theorem 5.4, there exists at least one redundant transition. It is obvious that in this case t , t_3 , t^ and t_5 are redundant. Figure 9 (bottom) shows the SMWf resulting after removing the redundant transitions t and t^* .*

6 Conclusion

In this report we have shown some results on State Machine Workflow nets. First we have introduced basic background in Petri nets and we have established the notations later used throughout the text. Afterwards we focused on synthesis of State Machine Workflow nets, defining two main construction rules, which are Transition Refinement and Transition Addition and proving that these rules are sound and complete. Finally we went on to State Machine Workflow nets reduction

properties. We presented two rules to reduce the number of nodes of an SMWf net. Finally we also showed the completeness of these rules.

Future work leads us to study similar synthesis and reduction properties on more complicated nets. The first approach is to extend the rules for other classes of Petri nets such as SMWf nets with multiple initial and final places.

References

- G. Berthelot. *Verification de Réseaux de Petri*. PhD thesis, Université Pierre et Marie Curie, Paris, 1978.
- G. Berthelot. Transformations and decompositions of nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri nets 1986, part I on Petri nets: Central models and their properties*, pages 359–376, London, 1987. Springer.
- J. Desel. Reduction and design of well-behaved concurrent systems. In J.C.M. Baeten et al., editors, *CONCUR '90: Proceedings on Theories of concurrency : unification and extension*, pages 166–181, New York, 1990. Springer.
- J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, New York, 1995.
- T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, volume 77, pages 541–580, 1989.
- I. Suzuki and T. Murata. A method for stepwise refinement and abstraction of Petri nets. *Journal of Computer and System Science*, 27(1):51–76, 1983.
- R. Valette. Analysis of Petri nets by stepwise refinements. *Journal of Computer and System Sciences*, 18:35–46, 1979.