Summary

Master's Thesis

Jacob Buchreitz Harbo[1]
Supervised by: Jiří Srba[1]

[1] *Department of Computer Science, Aalborg University*

June 10, 2016

Petri nets are a visual model with places and transitions, which have been extended and expanded upon ever since their introduction. One of these extensions are modalities, which allow for a single Petri net specification to model multiple different implementations and modal refinement which is a gradual process through which a modal Petri net specification is gradually refined until an implementable Petri net is obtained. A subclass of Petri nets is the workflow nets, which require the Petri net to have specific places with specific assigned properties. Workflow nets are great because they can be used to depict any number of workflows from the real world, and a property known as soundness allows us to be sure that a sound workflow can always terminate properly, and whenever it terminates it does so properly. But up until this thesis, which builds upon previous semester work, no one has attempted to combine the two to create modal workflow nets and get a model which allows a single modal workflow specification to be refined to a number of workflow implementations. We start out by introducing these workflows, and then proving that deciding soundness for these modal workflow nets is a PSPACE-complete problem.

Next, we tackle the state-space explosion problem, a problem which states that as a Petri net increases in size, the amount of reachable states, the state-space, increases exponentially, to the point that large Petri nets can take years for a computer to analyse. This is not ideal, and to combat the state-space explosion problem, we introduce five different structural reductions; the sequential place, sequential transition, parallel place, parallel transition, and looping transition reductions, and prove that these reductions, while shrinking the size of the workflow, and thus shrinking the state-space, ensure the net retains the soundness it had prior to their application, showing an example of a modal workflow being reduced down to two places and a transition, allowing the reduced net to be checked for soundness instead of the unreduced one.

Following the reductions, we introduce the concept of resources to increase the expressive power of the modal workflow modal. We change the concept of soundness to accommodate persistent resources and show that for modal workflow nets with resources the concepts of K-boundedness and soundness are not sufficient when analysing these nets. We thus introduce stricter versions of both, known as infinitary K-boundedness and infinitary soundness, which ensure that if a modal workflow net with resources is K-bounded or sound for the initial run, they will remain so for any subsequent runs. With these stronger definitions of K-boundedness and soundness, we prove that infinitary soundness for infinitary K-bounded nets is decidable in PSPACE.

Finally, we conclude upon our work and suggest a number of further extensions to the model and what their inclusions would mean for the model. We leave the research of these as open problems.

Soundness, Structural Reductions, and Resource Extensions for Modal Workflow Nets

Master's Thesis

Jacob Buchreitz Harbo[1]
Supervised by: Jiří Srba[1]

[1] *Department of Computer Science, Aalborg University*

June 10, 2016

Abstract

We study a specification formalism which combines modality, as known from modal transition systems, with Petri net-based workflows to create a new way of modelling workflows which we call modal workflow nets, and prove that deciding soundness for these modal workflow nets is PSPACE-complete. To combat the state-explosion problem in modal workflow nets we define a set of structural reductions which preserve soundness. We also extend the model with resources and show that conventional concepts of soundness and boundedness are insufficient when discussing modal workflows with resources, and we introduce stricter versions of both which we call infinitary soundness and infinitary K-boundedness. Finally, we prove that infinitary soundness is decidable in PSPACE for infinitary K-bounded nets.

1 Introduction

Workflows are used in many practical applications in today's world. From the smallest of tasks to the multi-billion dollar enterprises, workflows are used to ensure that everything being produced/done does so according to set standards or specifications [20]. Whether it is ensuring that a patient is not waiting hours for life-critical treatment or that each car rolling off the assembly is exactly the same as the next one, workflows and maintenance of such is an important factor in anything critical to ensure that nothing adverse can happen during the course of service or manufacture, this is usually referred to as *soundness*.

However, workflows themselves are not always enough; often, they need to be *extended* to accommodate analytical needs. Modalities are an extension of Petri nets, which allow for a single specification to model multiple implementations, and as such we initiated the work on adding modalities to Petri-net based Workflows in [9], in that paper, we also presented a claim that soundness was *PSPACE-complete* which we prove it is in this thesis.

A common problem to run into while working with Petri nets is the so-called *state-space explosion problem* [14], which states that as the size of a Petri net increases, the number of reachable markings increases exponentially, which, in turn, time and memory required to verify any properties of the net. This renders larger-size Petri nets immensely time-consuming to verify, some taking weeks, months or even years. To combat this for the modal Petri net model, we suggest a number of *structural reductions*, drawing inspiration from Tadao Murata's work in [13] and adapting his ideas for use in verifying soundness in modal workflows.

Another common extension to Petri nets are *resources*, which allow for a Petri net to consume and create different kinds of resources, and for transitions to be fired only if the resources are available to it. These resources can be used to model anything from wood at a lumber mill to the workforce at a company. Workflows with resources introduce new problems; it is not certain that just because a workflow is sound that it will remain so if run multiple times. Thus, we introduce the concept of *infinitary soundness*, where a modal workflow with resources is infinitary sound only if it can be run an indefinite amount of times and still be sound the next time it is run as well.

In the present paper, we introduce the modal Petri nets and workflow nets from [9] and the theorem that soundness is preserved through a special kind of refinement known as the *modal workflow refinement*. We will then prove that checking soundness for these nets can be done in a more memory-efficient way than the standard of depth-first-search, and then suggesting a way for modal workflows to combat the state-space explosion problem through reductions which preserve soundness. Finally, we combine modal workflows with resources, show how conventional concepts of soundness and boundedness are not strict enough when discussing modal workflow nets with resources, and introduce the concepts of *infinitary soundness* and *infinitary K-boundedness* while showcasing why they are needed when discussing modal workflows with resources while proving that infinitary soundness is decidable for infinitary K-bounded modal workflow nets with resources.

1.1 Related Work

Petri nets have been extended in various ways ever since their introduction by Carl Adam Petri in 1962 [16]; coloured Petri nets [10] are Petri nets which allow one to store information in the individual tokens which can be investigated and modified, timed-arc Petri nets [11] add temporal information to Petri nets, and prioritised Petri nets [8] add priorities to transitions.

Another particular extension is the modal Petri net [6], which adds modalities which allow for a transition to either be required or allowed, known as must and may respectively, in so-called refinements of the original net. Normally, every required transition is also allowed, but Mixed transition systems were introduced by Dennis Dams in [4] in the context of abstract interpretations, and the main aspect of mixed transition systems is that the must and may relation possibly does not hold, and if we, as such, end up with a transition that is required, but not allowed, we have a conflict of behaviour.

All of these extensions can be applied to Petri nets and the subset of Petri nets known as workflows [23], from this subset of Petri nets arise workflows with resources, allowing a workflow to depict consumption and creation of resources during a single pass of the workflow [22, 21]. Of course, with the creation of resources comes the problem of boundedness, and Natalia Sidorova and Christian Stahl showed in [19] that soundness for bounded modal workflows with resources was decidable through reduction to the home-space problem, which states that certain markings are so-called home markings, the set of which form the home-space, if for every reachable marking one of the home markings in the home space can always be reached. Vladimir A. Bashkin and Irina A. Lomazova takes it a step further in [1], proving that soundness for workflows with a single unbounded resource is decidable while leaving the question of whether it is decidable for multiple unbounded resources an open question, which remains unanswered to this day.

Transition Systems with Responses (TSR) is generalisation of Modal Transition Systems. While retaining the implementation-time refinement, they also introduce a *response set* instead of the usual must transitions, such that if an action belongs to the response set of a state,

it is required before termination is reached unless a state is reached in which the action no longer belongs to the response set. Thus, they combine elements of modal and mixed transition systems. They were introduced by Marco Carbone, Thomas Hildebrandt, Gian Perrone, Andrzej Wasowski in [3].

But as far as we know, while modal Petri net have been studied, and Petri net based workflows have been studied, no one has attempted to combine the two before us. This also means that there has been no work done in regards to structural reductions, nor has any extensions been suggested before.

2 Bibliographical Remarks

Chapter 3 draws heavily upon definitions from [9].

- In section 3.1:
  - Definitions 3.1 to 3.4, are identical to Definitions 2.2 to 2.5 in [9], respectively.
- In section 3.2:
  - Definition 3.5 draws heavy inspiration from Definition 3.1 in [9],
  - Definition 3.6 is almost identical to Definition 2.7 in [9], the only difference being the change from Petri net to modal Petri net,
  - Definition 3.7 and Definition 3.8 are similar to Definition 2.8 and Definition 3.3 in [9]
  - Definitions 3.9 to 3.11 are almost identical to Definitions 2.9 to 2.11 in [9], respectively. The only difference being the change from Petri net to modal Petri net and minor clarifications and remarks,
  - Definition 3.12 and Definition 3.13 are almost identical to Definitions 2.13 and 2.14 in [9], the only difference being the change from Petri net to modal Petri net and the corresponding may/must split, and
  - Definitions 3.15 to 3.17 are identical to Definitions 3.4 to 3.6 in [9], respectively.
- In section 3.3:
  - Definition 3.18 is identical to Definition 3.7 in [9] except for the difference discussed in the following remark.
  - Definition 3.19 and Definition 3.20 is identical to to Definition 2.17 and 2.18 in [9], respectively,
  - Definition 3.21 is identical to definition 3.8 in [9],
  - the text following Definition 3.21 is identical to definition 3.9 in [9],
  - Definition 3.22 is identical to Definition 3.10 in [9], and
  - any theorems in this section were proven in [9]

3 Preliminaries

3.1 Modal Transition Systems

A *modal transition system* (*MTS*) [2] is a formalism which extends the classical notion of labelled transition systems by introducing transitions of two types: must transitions that have to be present in any *implementation* of the MTS and may transitions that are allowed but not required.

Definition 3.1 (Modal Transition System)**.** A Modal transition system over an action alphabet A is a triple $T = (S, \dashrightarrow, \rightarrow)$ where

- S is the set of *states*,
- $\dashrightarrow \subseteq S \times A \times S$ is the set of *may transitions*, and
- $\rightarrow \subseteq \dashrightarrow$ is the set of *must transitions*.

We write $s \xrightarrow{a}$ if there exists some s' such that $s \xrightarrow{a} s'$ and $s \not\xrightarrow{a}$ if no such s' exists, similarilly for $\dashrightarrow$.

An example of this can be seen in Figure 1a.

3.1.1 Refinement and Implementation

A specification of an MTS includes possible may arcs, this allows a specification of an MTS to be implemented in many different ways as any number of may arcs may be included in the implementation.

Definition 3.2 (Implementation)**.** An MTS is an implementation if $\dashrightarrow = \rightarrow$.

However, implementation only represents the end product of so-called refinement, which allows gradual transformation of an MTS specification to implementation.

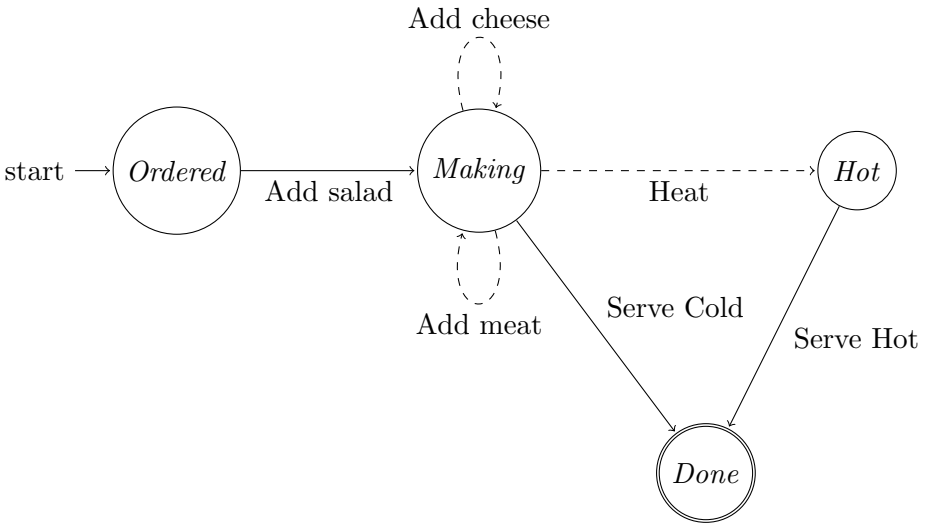
Definition 3.3 (Modal Refinement relation)**.** Let $T_1 = (S_1, \dashrightarrow_1, \rightarrow_1)$ and $T_2 = (S_2, \dashrightarrow_2, \rightarrow_2)$ be modal transition systems. A relation $R \subseteq S_1 \times S_2$ is a modal refinement relation if for all $(s_1, s_2) \in R$

a) if $s_1 \overset{a}{\dashrightarrow}_1 s_1'$ then there is a transition $s_2 \overset{a}{\dashrightarrow}_2 s_2'$ such that $(s_1', s_2') \in R$ and

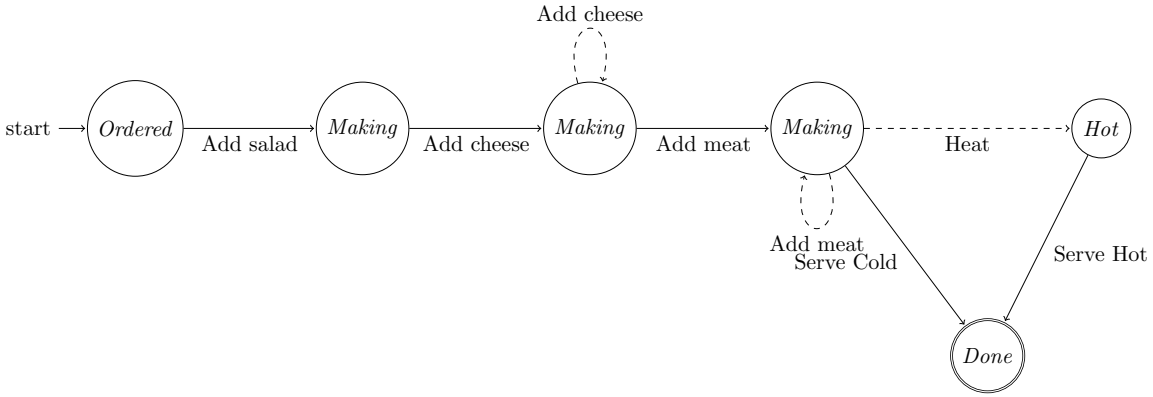
b) if $s_2 \xrightarrow{a}_2 s_2'$ then there is a transition $s_1 \xrightarrow{a}_1 s_1'$ such that $(s_1', s_2') \in R$.

Definition 3.4 (Modal Refinement)**.** Let T_1 and T_2 be MTS over the same action alphabet. We say that a state s_1 from T_1 modally refines a state s_2 from T_2, written as $s_1 \leq_m s_2$ if there is a modal refinement relation R such that $(s_1, s_2) \in R$.

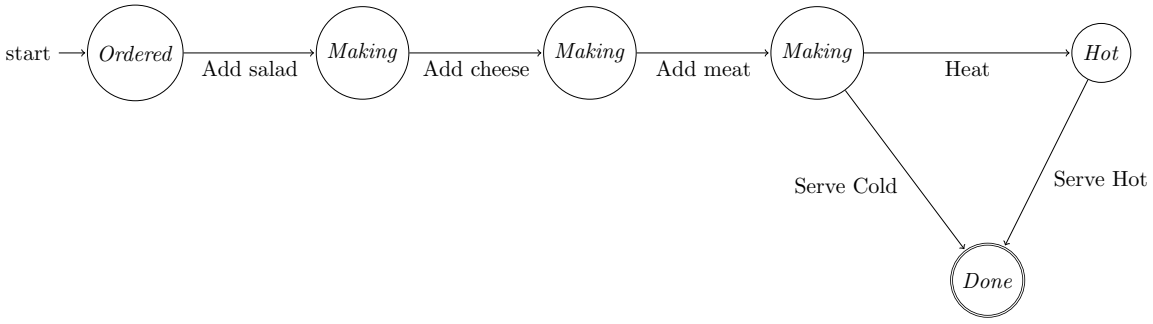
An example of a refinement of Figure 1a can be seen in Figure 1b, and this figure can be further refined to an implementation as seen in Figure 1c.



(a) A modal transition system specification of making a sandwich.



(b) A modal transition system refinement of Figure 1a.



(c) A modal transition system implementation of Figure 1a.

Figure 1: Gradual refinement of Figure 1a untill we reach an implementation allowing for the creation of a cheese-and-meat sandwich that can be served either hot or cold.

3.2 Modal Petri Net

Just as Petri nets are a graphical language which bases itself upon labelled transition systems, modal Petri nets base themselves upon Modal transition systems.

Definition 3.5 (Modal Petri Net [6])**.** Let A be an action alphabet. A modal Petri net (MPN) $\mathcal{N}$ over A is is a tuple $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ where

- P is a set of places
- $T^{\Diamond}$ is the set of *may* transitions
- $T^{\Box} \subseteq T^{\Diamond}$ is the set of *must* transitions.
- $F \subseteq (P \times T^{\Diamond}) \cup (T^{\Diamond} \times P)$ is a set of arcs
- $W : F \to \mathbb{N}$ is a weight function, and
- $L : T^{\Diamond} \to A$ is a labelling function.

We depict the transitions in $T^{\Box}$ as solid black rectangles, and transitions $t \in T^{\Diamond} - T^{\Box}$ as black-edged, white rectangles. Note that every must transition is also a may transition, but unless otherwise stated, the must attribute takes precedence when visualizing the transitions. An example of a modal Petri net which generates Figure 1a can be seen in Figure 2.

As with any Petri net, modal Petri nets have markings, and we introduce the notion of pre- and postsets for places in modal Petri nets.

Definition 3.6 (Marking)**.** A marking on a modal Petri net is a function $M : P \to \mathbb{N}^0$ that assigns a number of tokens to places. A modal Petri net without a specific initial marking is denoted by $\mathcal{N}$, and a modal Petri net with some initial marking M is called a marked modal Petri net and denoted by $(\mathcal{N}, M)$. We denote the set of all markings of a Petri net $\mathcal{N}$ as $\mathcal{M}(\mathcal{N})$.

Definition 3.7 (Pre- and postsets for places in a modal Petri net)**.** The preset of input may transitions of a place $p \in P$ is defined as ${}^{\Diamond}p = \{t \in T^{\Diamond} \mid (t, p) \in F\}$, and $p^{\Diamond} = \{t \in T^{\Diamond} \mid (p, t) \in F\}$ for the postset of may transitions. The pre- and postsets of must transitions of a place $p \in P$ is defined in the same way, with any $\Diamond$ substituted with $\Box$.

Definition 3.8 (Pre- and postsets for transitions in a modal Petri net)**.** The preset of input places of a transition $t \in T^{\Diamond}$ is defined as ${}^{\bullet}t = \{p \in P \mid (p, t) \in F\}$, and $t^{\bullet} = \{p \in P \mid (t, p) \in F\}$.

With markings and pre/postsets defined, we now define enabledness and firing of transitions.

Definition 3.9 (Enabledness)**.** A transition $t \in T^{\Diamond}$ is *enabled* in M if for all $p \in {}^{\Diamond}t$ we have $M(p) \geq W(p, t)$.

Definition 3.10 (Firing)**.** A transition $t \in T^{\Diamond}$ that is enabled may be *fired*, if so, the marking M changes to M' such that for all $p \in P$ we have $M'(p) = M(p) - W(p, t) + W(t, p)$ where we assume $W(p, t) = 0$ if $(p, t) \notin F$ and $W(t, p) = 0$ if $(t, p) \notin F$. We write this as $M \xrightarrow{t} M'$ if $t \in T^{\Box}$ and $M \overset{t}{\dashrightarrow} M'$ if $t \in T^{\Diamond}$.

Remark. *Although it is true that every* $\xrightarrow{t}$ *is also a* $\overset{t}{\dashrightarrow}$*, the must transition firing takes priority in the notation unless otherwise noted.*

Definition 3.11 (Firing Sequence)**.** A firing sequence $M \overset{w}{\dashrightarrow} M'$ where $w = t_1 \dots t_n \in T^{\Diamond^*}$ is a sequence of transition firings such that $M \overset{t_1}{\dashrightarrow} M_1 \overset{t_2}{\dashrightarrow} M_2 \dots \overset{t_n}{\dashrightarrow} M'$ where each marking enables the following transition firing.

We write $M \dashrightarrow^* M'$ if we do not care which transitions are fired and whether they are may or must transitions, only that it is possible to fire some sequence of transitions from marking M to reach marking M', likewise, we write $M \rightarrow^* M'$ if all transitions in this sequence are must transitions but we otherwise do not care about which transitions are fired. Now that we know what it means to fire a transition, we can define the concept of reachabillity, deadness and boundedness.

Definition 3.12 (Reachability)**.** We denote the set of reachable markings in $\mathcal{N}$ from a marking M as $Reach(\mathcal{N}, M) = \{M' \mid M \dashrightarrow^* M'\}$.

Definition 3.13 (Deadness)**.** A place $p \in P$ is *dead* in a marked modal Petri net $(\mathcal{N}, M)$ if $M'(p) = 0$ for all $M' \in Reach(\mathcal{N}, M)$. A transition $t \in T^{\Diamond}$ in a marked modal Petri net $(\mathcal{N}, M)$ is dead if there is no $M', M'' \in Reach(\mathcal{N}, M)$ such that $M' \overset{t}{\dashrightarrow} M''$.

Definition 3.14 (K-boundedness)**.** A modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ with a start marking M_0 is K-bounded, where $K \in \mathbb{N}$, if for every reachable marking $M \in Reach(\mathcal{N}, M_0)$, $M(p) \leq K$ for all $p \in P$ and K is the lowest natural number that allows this.

And just as a Petri net generates an LTS, so does a modal Petri net generate an MTS.

Definition 3.15 (Semantics for Modal Petri Nets)**.** A modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ generates an MTS $T(\mathcal{N}) = (\mathcal{M}(\mathcal{N}), \dashrightarrow, \rightarrow)$ where

- states are markings,
- whenever a transition $t \in T^{\Box}$ can fire from a marking M in $\mathcal{N}$ and reach M' then $T(\mathcal{N})$ can also do $M \xrightarrow{L(t)} M'$, and
- whenever a transition $t \in T^{\Diamond}$ can fire from a marking M in $\mathcal{N}$ and reach M' then $T(\mathcal{N})$ can also do $M \overset{L(t)}{\dashrightarrow} M'$.

Just like MTS, Modal Petri nets can be gradually refined into implementations.

Definition 3.16 (Modal Petri Net Refinement)**.** Let $\mathcal{N}_1 = (P_1, T_1^{\Box}, T_1^{\Diamond}, F_1, W_1, L_1, M_1)$ and $\mathcal{N}_2 = (P_2, T_2^{\Box}, T_2^{\Diamond}, F_2, W_2, L_2, M_2)$ be marked modal Petri nets. We say that $\mathcal{N}_2$ modally refines $\mathcal{N}_1$, denoted as $\mathcal{N}_2 \leq_m \mathcal{N}_1$ if $M_2 \leq_m M_1$ in MTSs $T(\mathcal{N}_2)$ and $T(\mathcal{N}_1)$, respectively.

We also define modal implementation for Petri nets.

Definition 3.17 (Modal Petri Net Implementation)**.** A MPN $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ is an implementation if $T^{\Box} = T^{\Diamond}$.

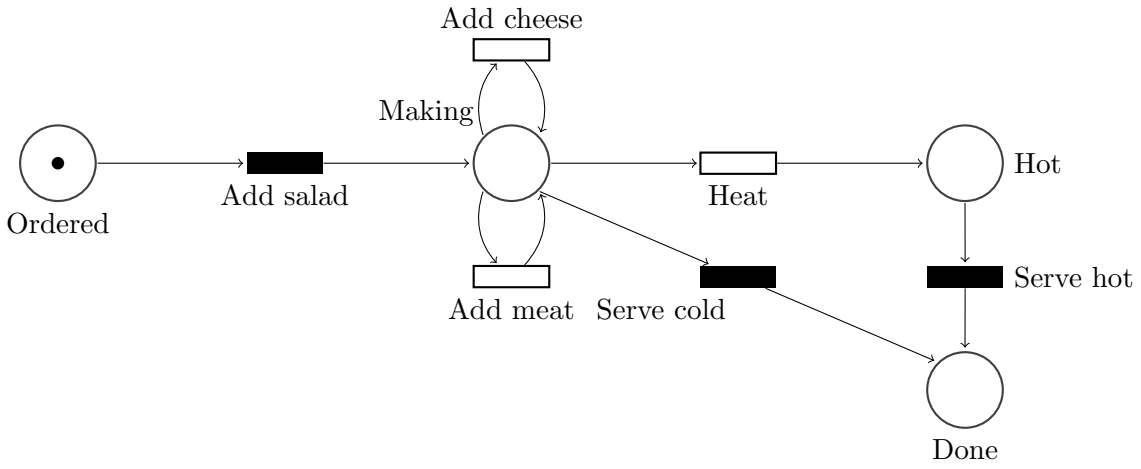


Figure 2: The modal Petri net which generates Figure 1a

3.3 Modal Workflow nets

Workflow nets are a subset of Petri nets, useful, as their name implies, for modelling workflows and procedures. The main result of [9] was the introduction of modal Workflow nets, proving that soundness for these modal workflow nets is decidable, and proving that a special refinement ensures soundness of the refined net.

Definition 3.18 (Modal Workflow Net)**.** A modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ is a modal workflow net (MWFN) if $\mathcal{N}$ has two special places, the source place $i \in P$ and the sink place $o \in P$ such that ${}^{\Diamond}i = o^{\Diamond} = \emptyset$, and ${}^{\Diamond}p \neq \emptyset$ and $p^{\Diamond} \neq \emptyset$ for all other $p \in P$.

Remark. *In other works one might find the definition of a workflow net having been appended by a second requirement, that of strong connectivity if a transition from o to i has been added. We leave this out as Theorem 3.3 presented in [9] are rendered untrue with it.*

Definition 3.19 (Start marking)**.** A marking M^{in} is the *start* marking where $M^{in}(p) = 1$ if $p = i$ and $M^{in}(p) = 0$ otherwise.

Definition 3.20 (Finish marking)**.** A marking M^{out} is the *finish* marking where $M^{out}(p) = 1$ if $p = o$ and $M^{out}(p) = 0$ otherwise.

Definition 3.21 ([9] Soundness for MWFN)**.** A modal workflow net is sound if:

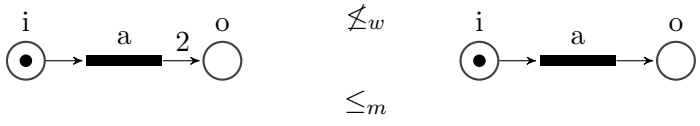
a) whenever $M^{in} \dashrightarrow^* M$ then there exists a firing sequence $M \rightarrow^* M^{out}$, and

b) if $M^{in} \dashrightarrow^* M$ such that $M(o) \geq 1$, then $M = M^{out}$.

Let $\overline{\mathcal{M}(\mathcal{N})} \subseteq \mathcal{M}(\mathcal{N})$ be the set of well-formed markings $\overline{\mathcal{M}(\mathcal{N})} \stackrel{def}{=} \{M \in \mathcal{M}(\mathcal{N}) \mid M(o) > 0$ implies that $M(p) = 0$ for all $p \in P \setminus \{o\}$ and $M(o) = 1\}$.

Definition 3.22 (Modal workflow refinement)**.** Let $\mathcal{N}_1 = (P_1, T_1^{\Box}, T_1^{\Diamond}, F_1, W_1, L_1)$ and $\mathcal{N}_2 = (P_2, T_2^{\Box}, T_2^{\Diamond}, F_2, W_2, L_2)$ be modal workflow nets with initial markings M_1^{in} and M_2^{in} respectively. We say that $\mathcal{N}_2$ modally workflow refines $\mathcal{N}_1$, written $\mathcal{N}_2 \leq_w \mathcal{N}_1$ if there is a relation $R \subseteq (\overline{\mathcal{M}(\mathcal{N}_1)} \times \overline{\mathcal{M}(\mathcal{N}_2)})$ such that:

a) R is modal refinement and



a) Unsound workflow net b) Sound workflow net

Figure 3: A workflow net in modal refinement with another workflow net, where the refined net is not sound

b) for all $(M_1, M_2) \in R$ we have $M_1(o) = M_2(o)$, and

c) $(M_1^{in}, M_2^{in}) \in R$.

An example of the difference between modal refinement and modal workflow refinement can be seen in Figure 3, while Figure 3a might be a modal refinement of Figure 3b, it does not uphold the definition of having well-formed markings and is thus not a modal workflow refinement

Theorem 3.1 (Soundness through workflow refinement [9])**.** *Let* $\mathcal{N}_1 = (P_1, T_1^{\Box}, T_1^{\Diamond}, F_1, W_1, L_1)$ *and* $\mathcal{N}_2 = (P_2, T_2^{\Box}, T_2^{\Diamond}, F_2, W_2, L_2)$ *be modal workflow nets. If* $\mathcal{N}_1$ *is sound, and* $\mathcal{N}_2 \leq_w \mathcal{N}_1$*, then* $\mathcal{N}_2$ *is sound.*

Theorem 3.2. *[9] Modal workflow refinement is undecidable.*

Theorem 3.3. *[9] If* $\mathcal{N}_1 = (P_1, T_1^{\Box}, T_1^{\Diamond}, F_1, W_1, L_1)$ *and* $\mathcal{N}_2 = (P_2, T_2^{\Box}, T_2^{\Diamond}, F_2, W_2, L_2)$ *are bounded modal workflow nets then modal workflow refinement is* EXPTIME-complete*.*

Theorem 3.4. *[9] Soundness is decideable for modal workflow nets in EXPSPACE.*

Theorem 3.5. *[9] Soundness for bounded modal workflow nets is decidable in* EXPTIME*.*

An example of a bounded Modal Workflow net representing a simplified car wash can be seen in Figure 4. Once activated, this modal workflow accepts a car inside, and then a terminal accepts a card detailing the kind of wash the car is to undergo. The modal workflow then contains a sequence detailing a standard wash of foaming, brush wash, and drying, modalities extend from this sequence to also allow for more foam, for a more thorough drying, and for an entire sequence of accessorial to the wash, such as lacquer sealing. At the end of the day, the car wash can be deactivated only if the wash is empty.

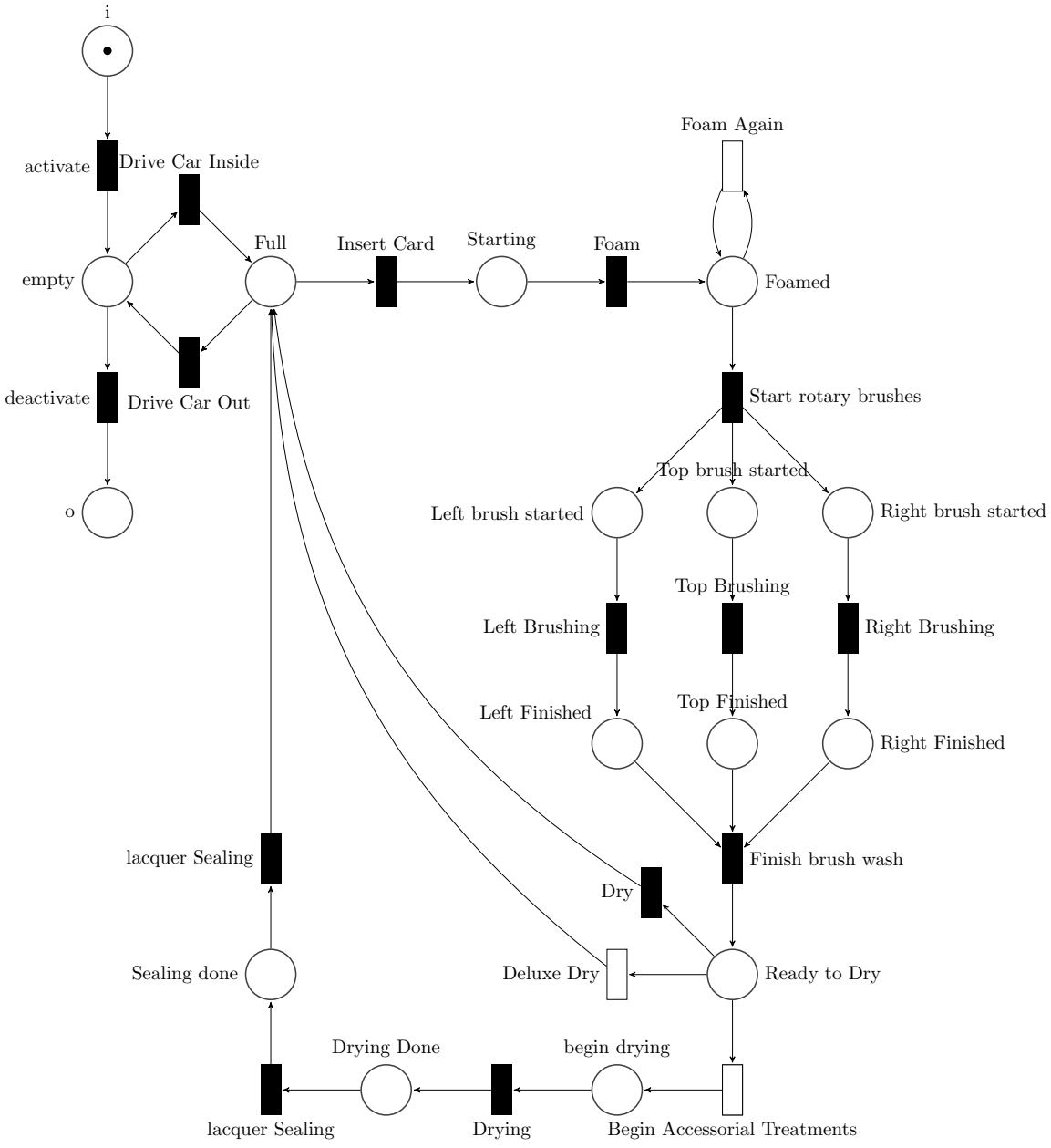


Figure 4: Modal Workflow net of a simplified car wash

4 Soundness for bounded modal workflow nets

Theorem 3.5 from [9] states that soundness for bounded modal workflow nets is decidable in *EXPTIME*, and in that paper, the claim that the problem is *PSPACE-complete* was left as an open problem, in this section we prove this claim.

Theorem 4.1. *Soundness for bounded modal workflow nets is* PSPACE-complete*.*

This claim is proven in two parts, that soundness for bounded modal workflow nets is decidable in PSPACE, and that we can reduce a PSPACE-complete problem to the soundness problem.

Lemma 4.1. *Soundness for bounded modal workflow nets is decidable in PSPACE.*

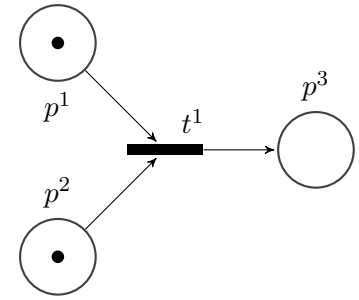
Proof. We know that PSPACE = NPSPACE from Savitch's theorem [18]. Soundness consists of two parts; the knowledge that any given reachable marking can reach termination, and that whenever termination occurs, it does so without residual tokens.

- Reachable Termination: Consider a k-bounded modal workflow net. The list of all markings is exponential sized for a k-bounded net, and thus cannot be directly enumerated as the list would use exponential space. Instead, we run through each marking with at most k tokens in sequence, resetting any space used after each marking in has been tested. For this marking, M, we nondeterministically 'guess' a path to reach M from M^{in}, which can be done in NPSPACE. If our guess does not bring us to M, then M is not reachable and we move on to the next marking in the ordered list, erase any data in the space used and repeat using the now-clean space. If our guess does bring us to the marking M, then, because reachability is decidable in PSPACE, we run reachability from M with the goal of reaching M^{out}. If we find that M^{out} is not reachable, then we know at least one of the net's reachable markings can not reach M^{out}, and the net is thus unsound, if our guess does bring us to M^{out}, then we generate the next marking in the k-bounded net, reset any space used, and repeat the procedure, reusing the same space.
- Terminating Correctly: Assume we have a bounded modal workflow net $\mathcal{N}$. We take nondeterministic guesses with the goal of ending up in a marking where $M(o) > 0$ yet $M \neq M^{out}$. If we end up in a marking where this is true, then the net is not sound, otherwise, we know that the net always terminates properly, and this is doable in NPSPACE.

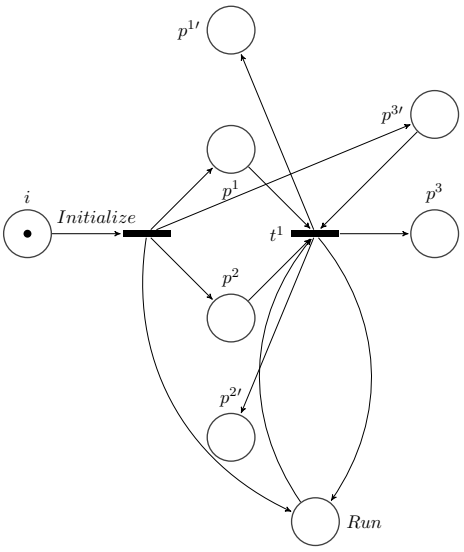
This gives us the algorithm seen in Algorithm 1, which is runnable in NPSPACE, as described above. □

Lemma 4.2. *Deciding soundness is PSPACE-hard for 1-safe modal workflow nets.*

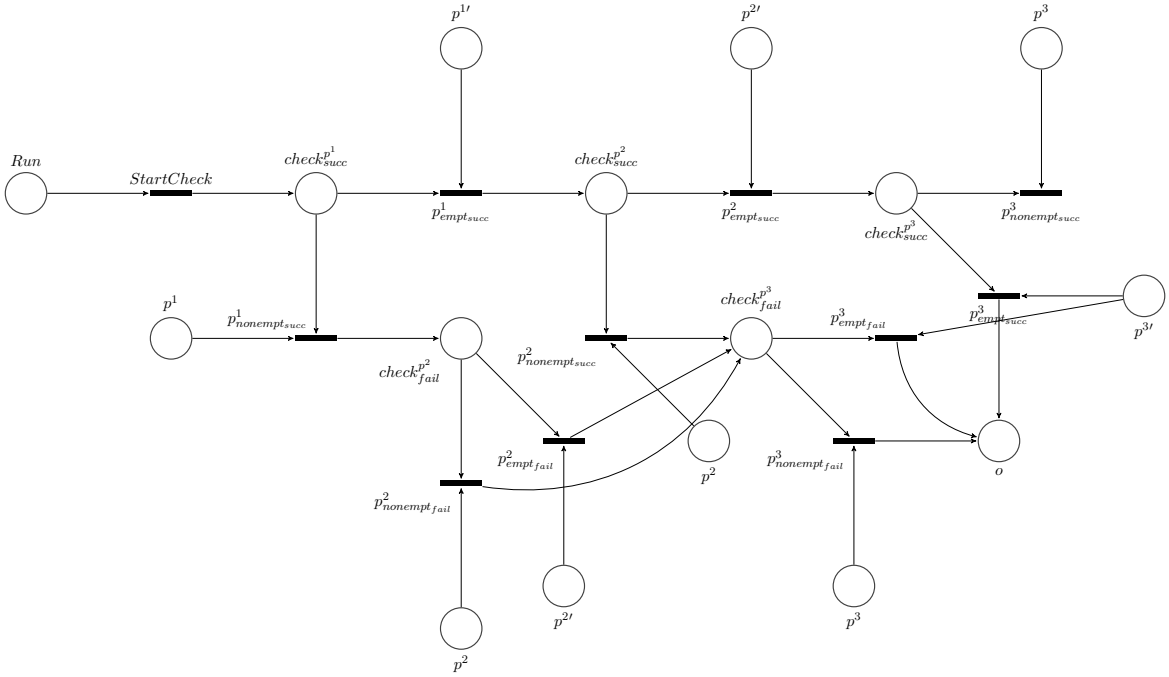
Proof. We prove this by reducing reachability for one-safe nets to soundness, as we know that reachability for 1-safe Petri Nets is *PSPACE-complete* as was proven by Esparza et. al. in [7]. Given a 1-safe net where we assume without loss of generality $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L, M_0, M_t)$ where M_0 is the initial marking, M_t is the target marking, and where we assume without loss of generality that $P = \{p^1, p^2, p^3, \ldots, p^q\}$ and $T^{\Diamond} = \{t^1, t^2, t^3, \ldots, t^m\}$ (example can be seen in Figure 5a), construct a workflow net $\mathcal{N}' = (P', T^{\Box\prime}, T^{\Diamond\prime}, F', W, L, M_0, M_t)$ as seen in



(a) Example net for Lemma 4.2



(b) First half of $\mathcal{N}'$



(c) The second part of the extension made to Figure 5a. If any transitions or places share a name in the figure, or with Figure 5b, they are the same transition or place. This is done for the sake of readability, as the net would be cluttered otherwise.

Figure 5: Lemma 4.2 visualized.

Algorithm 1 Checking termination

```
1: function TERMINATION CHECK(K-bounded net)
2:     for all markings M with at most K tokens do
3:         if M^in →* M and M ↛* M^out then return 'Not Sound'
4:         end if
5:         if M^in →* M and M(o) ≥ 1 and M ≠ M^out then return 'Not Sound'
6:         end if
7:         Clean Space
8:     end for
9:     return 'Sound'
10: end function
```

Figure 5c by adding place i and *Initialize* transitions, which adds tokens to recreate the tokens in Figure 5a's start marking and the added Run place, which only allows the original net's transitions to be fired if there is a token in it, we also add complementary places which have a token when their complementary place do not, and vice versa, and added arcs to obtain this effect, and follow this up by the second of $\mathcal{N}'$ as seen in Figure 5b which contains transitions that check whether or not M_t is reached. Thus we get that:

1. $P' = P \cup \{i\} \cup \{o\} \cup \{p' \mid p \in P\} \cup \{Run\} \cup \{check_{succ}^{p^i} \mid p \in P\} \cup \{check_{fail}^{p^i} \mid p \in P \setminus \{p^1\}\}$
2. $T^{\diamond\prime} = T^{\diamond} \cup \{initialize\} \cup \{StartCheck\} \cup \{p^i_{empt_{succ}} \mid p \in P\} \cup \{p^i_{nonempt_{succ}} \mid p \in P\} \cup \{p^i_{empt_{fail}} \mid p \in P \setminus \{p^1\}\} \cup \{p^i_{nonempt_{fail}} \mid p \in P \setminus \{p^1\}\}$
3. $F' = F \cup \{(i, initialize)\} \cup \{(initialize, p) \mid M_0(p) = 1\} \cup \{(initialize, Run\} \cup$
   $\{(initialize, p') \mid M_0(p) = 0\} \cup \{(p', t) \mid (t, p) \in F\} \cup \{(t, p') \mid (p, t) \in F\} \cup$
   $\{(Run, StartCheck)\} \cup \{(StartCheck, check_{succ}^{p1})\} \cup$
   $\{(check_{succ}^{p^i}, p^i_{empt_{succ}}) \mid p^i \in P\} \cup \{(check_{succ}^{p^i}, p^i_{nonempt_{succ}}) \mid p^i \in P\} \cup$
   $\{(check_{fail}^{p^i}, p^i_{empt_{fail}}) \mid p^i \in P \setminus \{p^1\}\} \cup \{(check_{fail}^{p^i}, p^i_{nonempt_{fail}}) \mid p^i \in P \setminus \{p^1\}\} \cup$
   $\{(p, p^i_{nonempt_{succ}}) \mid p^i \in P\} \cup \{(p', p^i_{empt_{succ}}) \mid p^i \in P\} \cup$
   $\{(p, p^i_{nonempt_{fail}}) \mid p^i \in P \setminus \{p^1\}\} \cup \{(p', p^i_{empt_{fail}}) \mid p^i \in P \setminus \{p^1\}\} \cup$
   $\{(p^i_{empt_{succ}}, check_{succ}^{p^{i+1}} \mid p^i \in P \text{ if } M_t(p) = 0 \text{ and } p^i \neq p^m\} \cup$
   $\{(p^{nonempt_{succ}}, check_{succ}^{p^{i+1}} \mid p^i \in P \text{ if } M_t(p) = 1 \text{ and } p^i \neq p^m\} \cup$
   $\{(p^i_{nonempt_{succ}}, check_{fail}^{p^{i+1}} \mid p^i \in P \text{ if } M_t(p) = 0 \text{ and } p^i \neq p^m\} \cup$
   $\{(p^{empt_{succ}}, check_{fail}^{p^{i+1}} \mid p^i \in P \text{ if } M_t(p) = 1 \text{ and } p^i \neq p^m\} \cup$
   $\{(p^i_{m_{empt_{fail}}}, o)\} \cup \{(p^i_{m_{nonempt_{fail}}, o)\}} \cup \{(p^i_{m_{empt_{succ}}}, o) \mid \text{ if } M_t(p^m) = 1\} \cup$
   $\{(p^i_{m_{nonempt_{succ}}}, o) \mid \text{ if } M_t(p^m) = 0\}$

Now we need to prove 3 things about $\mathcal{N}'$:

- $\mathcal{N}'$ is a workflow net: This is easy to prove as $\mathcal{N}'$ fulfils the condition for workflow nets.
- If $M^{in} \rightarrow^* M_t$ then $\mathcal{N}'$ is not sound: This is true as if we reach M_t, then once we do the *StartCheck* transition, there exists a transition firing sequence which consumes the tokens from both the standard and negated places, which ends ends in a transition consuming the last tokens in the net with no output, rendering us deadlocked in a

marking M_{dead} where $M_{dead}(p) = 0$ for all $p \in P$. This is depicted in Figure 5c as being the upper row of transitions.

- If $M^{in} \not\rightarrow^* M_t$ then $\mathcal{N}'$ is sound: This is true as if the marking reached is not M_t, then once *StartCheck* is fired, the only sequence of transitions that can be fired to clean out the net of any residual tokens will invariably end up in M^{out}.

Because reachability can be done in PSPACE, the class PSPACE is closed under negation, and we have just proven that the workflow net in Figure 5c is sound only if the target marking M is not reachable, and because reachability for one-safe nets is PSPACE-hard, then soundness for 1-safe workflow nets is PSPACE-hard. □

Thus we have proven that soundness can be done in PSPACE, and we have proven through reduction from reachability that it is also PSPACE-hard, which means soundness for bounded modal workflow nets is PSPACE-complete.

5 Structural Reductions

Structural reductions are a way to eliminate unneeded complexity by reducing the state space without affecting the results a given net provides. In his paper, Tadao Murata [13] describes a few such structural reductions for Petri nets, and we have used his ideas before in a previous work to make and prove structural reductions for timed-arc Petri nets [12]. We now wish to make structural reductions for modal workflow nets and prove these will preserve the soundness of the unreduced net.

For the use in this section, we will introduce a *mayst* transition, as seen in Figure 6. This transition is a visual shorthand for saying 'This transition can be either may or must for the purposes of the depicted reduction.'

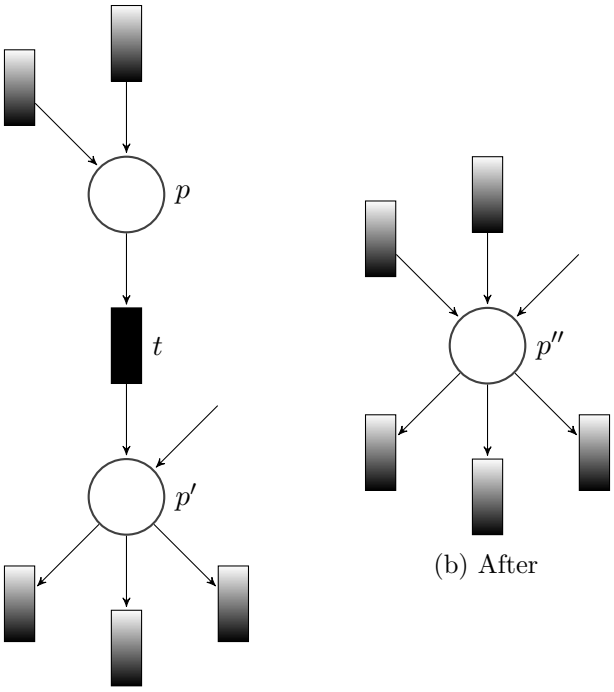
5.1 Sequential place merger

Definition 5.1 (Sequential place merger)**.** Given a modal workflow net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ with a transition $t \in T^{\Box}$ and $p, p' \in P \setminus \{i, o\}$ such that

- $p \neq p'$
- ${}^\bullet t = \{p\}$
- $t^\bullet = \{p'\}$
- $p^\Diamond = \{t\}$
- $t \in T^{\Box}$
- $W(p, t) = W(t, p') = 1$

then we can reduce the net by

Figure 6: A mayst transition



(a) Before

(b) After

Figure 7: Definition 5.1 visualised: Place p and p' can be combined without impacting the soundness of the net.

- removing t, p, and p' and any connected arcs
- adding new place p'' where:
  - ${}^{\diamond}p'' = {}^{\diamond}p \cup ({}^{\diamond}p' \setminus \{t\})$,
  - $p''^{\diamond} = p'^{\diamond}$,
  - $W(t', p'') := W(t, p) + W(t', p')$. for all $t' \in {}^{\diamond}p''$, and
  - $W(p'', t') := W(p', t')$. for all $t' \in p''^{\diamond}$.

An example can be seen in Figure 7.

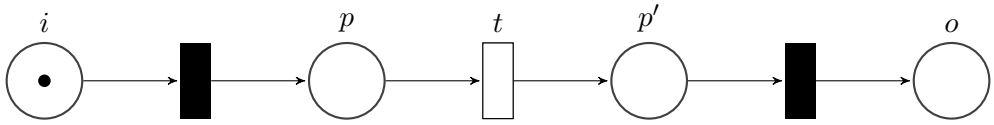
For the rest of Section 5.1 let $\mathcal{N} = (P, T^{\square}, T^{\diamond}, F, W, L)$ be a modal workflow net and $\mathcal{N}'$ the modal workflow net obtained by performing the reduction from Definition 5.1 once on $\mathcal{N}$.

Definition 5.2 (Marking equivalence for sequential place merger)**.** Let M be a marking in a modal workflow net $\mathcal{N}$ and let M' be a marking in the modal workflow net $\mathcal{N}'$. We say that marking M is equivalent to M', written $M \equiv M'$, if $M(q) = M'(q)$ for all $q \in P \setminus \{p, p'\}$ and $M(p) + M(p') = M'(p'')$.

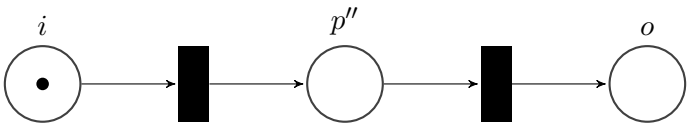
Lemma 5.1. *If M^{in} $\dashrightarrow^*$ M in $\mathcal{N}$, then M^{in} $\dashrightarrow^*$ M' in $\mathcal{N}'$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let M^{in} $\dashrightarrow^m$ M in $\mathcal{N}$ where $m \geq 0$, then we want to do M^{in} $\dashrightarrow^*$ M' in $\mathcal{N}$ such that $M \equiv M'$.

- $m = 0$ case: trivial. Because doing no transition firing from M^{in} leaves us at M^{in} in $\mathcal{N}$, we do nothing in $\mathcal{N}'$ as well. This leaves us with $M^{in} \equiv M^{in}$, which is obviously true.



(a) May transition t, net unsound



(b) Attempted Definition 5.1 on a), net is now sound

Figure 8: Why t of Definition 5.1 has to be a must transition

- $m+1$ case: assume the lemma holds for m steps, this means that $M^{in} \dashrightarrow^m M_1$ and thus $M^{in} \dashrightarrow^* M_1'$ such that $M_1 \equiv M_1'$, we now do one more step such that $M_1 \overset{t_1}{\dashrightarrow} M$.
  - if $t_1 \neq t$, this is trivial, as M_1' can also fire t_1, thus $M_1' \overset{t_1}{\dashrightarrow} M'$ and clearly $M \equiv M'$.
  - if $t_1 = t$, then we simply do nothing in $\mathcal{N}'$, thus $M \equiv M_1'$.

□

Lemma 5.2. *If $M^{in} \dashrightarrow^* M'$ in $\mathcal{N}'$, then $M^{in} \dashrightarrow^* M$ in $\mathcal{N}$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let $M^{in} \dashrightarrow^m M'$ in $\mathcal{N}'$ where $m \geq 0$, then $M^{in} \dashrightarrow^* M$ such that $M \equiv M'$

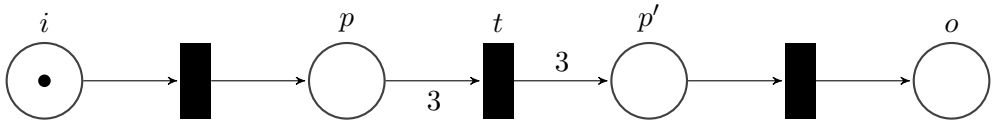
- $m = 0$ case: like in Lemma 5.1.
- $m+1$ case: assume the lemma holds for m steps, this means that $M^{in} \dashrightarrow^m M_1'$ and thus $M^{in} \dashrightarrow^* M_1$ such that $M_1 \equiv M_1'$, we now do one more step such that $M_1' \overset{t_1}{\dashrightarrow} M'$.
  - if $t_1 \notin p''^{\diamond}$, it is trivial; we fire the same transition in $\mathcal{N}$ such that $M_1 \overset{t_1}{\dashrightarrow} M$ and $M' \equiv M$.
  - if $t_1 \in p''^{\diamond}$, it is possible we cannot fire the same transition right away in $\mathcal{N}$ because the tokens required to do so might not be available in p', we thus fire t as many times as possible to move all tokens from p to p' such that $M_1 \overset{t}{\to} M_2 \overset{t}{\to} M_3 \overset{t}{\to} \dots M_m$ where $M_m \equiv M_1'$. We can now do the transition firing $M_m \overset{t_1}{\dashrightarrow} M$, and from this we get that $M \equiv M'$.

□

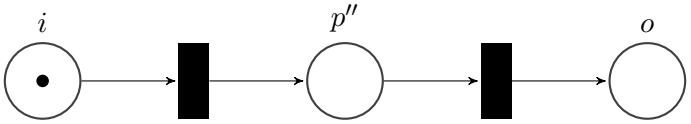
Theorem 5.1. *The modal workflow net $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound.*

Proof. We will split this proof up in two; the 'if' and the 'only if' halves.

⇒ Assume that $\mathcal{N}$ is sound, for $\mathcal{N}'$ to be sound, it has to fullfill the two conditions of Definition 3.21.



(a) Net is not sound.



(b) Attempted Definition 5.1 on a), net is now sound

Figure 9: Why we must ensure the weights connecting to t is equal 1 for Theorem 5.1 to hold.

Condition a) Let M' be a reachable marking in $\mathcal{N}'$. For $\mathcal{N}'$ to fulfill this case, it must be possible to reach M^{out}. Because of Lemma 5.2, this means that an equivalent marking M is reachable in $\mathcal{N}$. Since $\mathcal{N}$ is sound, this means a transition firing sequence exists such that $M \rightarrow^* M^{out}$. Because of Lemma 5.1, this means an equivalent transition firing sequence exists in $\mathcal{N}'$ such that $M' \rightarrow M^{out\prime}$ and since $M^{out} \equiv M^{out\prime}$, this implies $M^{out} = M^{out\prime}$.

Condition b) Let M' be a reachable marking in $\mathcal{N}'$. We know from Lemma 5.2 that any reachable marking M' in $\mathcal{N}'$ can be matched by an equivalent marking M in $\mathcal{N}$. As $\mathcal{N}$ is sound, this means that if $M(o) \geq 1$ then $M = M^{out}$, thus $M' \equiv M$ and this implies $M' = M^{out}$.

$\Leftarrow$ Assume that $\mathcal{N}'$ is sound, for $\mathcal{N}$ to be sound, it has to fullfill the two conditions of Definition 3.21.

Condition a) In much the same way as before, Lemma 5.1 proves that any marking in the unreduced net can be matched by an equivalent marking in the reduced net, because $\mathcal{N}'$ is sound, we know that from any reachable marking, $\mathcal{N}'$ can reach M^{out} by firing a sequence of transitions. Following Lemma 5.2 we will thus naturally reach M^{out} in N also.

Condition b) Let M be a reachable marking in $\mathcal{N}$. As Lemma 5.1 proves any reachable marking M in $\mathcal{N}$ can be matched by an equivalent marking M' in $\mathcal{N}'$. As $\mathcal{N}'$ is sound, this means that if $M'(o) \geq 1$ then $M' = M^{out}$, thus $M \equiv M' = M^{out}$, hence $M = M^{out}$

Thus we have proven that $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound. □

Remark. *It is important that the transition t in Definition 5.1 is a must transition, as if it is not, the unreduced net gains soundness as defined in Definition 3.21 through the refinement, an example of which can be seen in Figure 8. Likewise, it is important that the weights of the arcs connecting t to p and p′ are equal to 1, as otherwise we cannot prove that an unsound net stays unsound, as seen in Figure 9.*

5.2 Sequential transition merger

Definition 5.3 (Sequential transitions merger)**.** Given a modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ with two transitions t, t' and a place p, such that

- $t \neq t'$
- ${}^\bullet t' = \{p\}$
- $p^\diamond = \{t\}$
- ${}^\diamond p = \{t'\}$
- $W(t,p) = W(p,t')$
- $t' \in T^\square$

then

a) if $t \in T^\square$, we can remove t, p, and t', and replace them with $t'' \in T^\square$ where

- $t''^\bullet = t^\bullet \cup t'^\bullet \setminus \{p\}$,
- ${}^\bullet t'' = {}^\bullet t \cup {}^\bullet t' \setminus \{p\}$,
- $W(p', t'') = W(p', t)$ for all $p' \in {}^\bullet t''$, and
- $W(t'', p') = W(t, p') + W(t', p')$ for all $p' \in t^\bullet$.

b) if $t \in T^\diamond \setminus T^\square$ we can remove t, p, and t', and replace them with $t'' \in T^\diamond \setminus T^\square$ where

- $t''^\bullet = t^\bullet \cup t'^\bullet \setminus \{p\}$,
- ${}^\bullet t'' = {}^\bullet t \cup {}^\bullet t' \setminus \{p\}$,
- $W(p', t'') = W(p', t)$ for all $p' \in {}^\bullet t''$, and
- $W(t'', p') = W(t, p') + W(t', p')$ for all $p' \in t^\bullet$.

An example of case a) can be seen in Figure 10a and Figure 10b, and likewise with case b) in Figure 10c and Figure 10d.

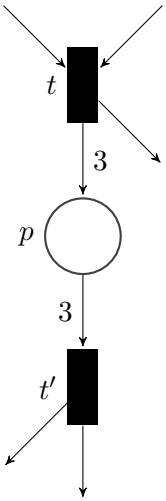
For the rest of Section 5.2 let $\mathcal{N} = (P, T^\square, T^\diamond, F, W, L)$ be a modal workflow net. Then $\mathcal{N}'$ is the modal workflow net obtained by performing the reduction from Definition 5.3 once on $\mathcal{N}$

Definition 5.4 (Marking equivalence for sequential transition merger)**.** Let M be a marking in a modal workflow net $\mathcal{N}$ and let M' be a marking in the modal workflow net $\mathcal{N}'$. We say that marking M is equivalent to M', written $M \equiv M'$, if $M(q) = M'(q)$ for all $q \in P \setminus \{p\}$ and $M(p) = 0$.

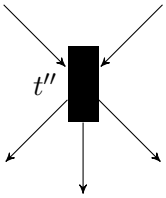
Lemma 5.3. *If $M^{in} \dashrightarrow^* M$ in $\mathcal{N}$ such that $M(p) = 0$, then $M^{in} \dashrightarrow^* M'$ in $\mathcal{N}'$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let $M^{in} \dashrightarrow^m M$ in $\mathcal{N}$ where $m \geq 0$, then $M^{in} \dashrightarrow^* M'$ in $\mathcal{N}$ such that $M \equiv M'$.

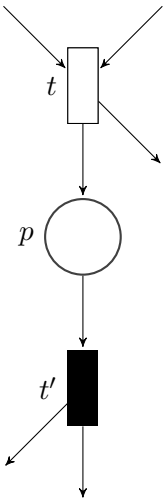
- $m = 0$ case: trivial. Because doing no transition firing from M^{in} leaves us at M^{in} in $\mathcal{N}$, we do nothing in $\mathcal{N}'$ as well. This leaves us with $M^{in} \equiv M^{in}$, which is obviously true.
- $m + 1$ case: assume the lemma holds for m steps such that $M^{in} \dashrightarrow^m M_1$ in $\mathcal{N}$. There will be two possible cases depending on the marking M_1
  1. Assume that $M_1(p) = 0$, this means that $M^{in} \dashrightarrow^* M_1'$ in $\mathcal{N}'$ such that $M_1 \equiv M_1'$, we now do one more step such that $M_1 \overset{t_1}{\dashrightarrow} M$.



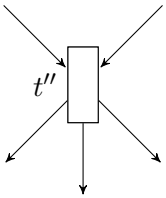
(a) Before



(b) After



(c) Before



(d) After

Figure 10: Definition 5.3 visualised: transition t and t' can be combined without impacting the soundness of the net.

- if $t_1 \neq t$, this is trivial, as M_1' can also fire t_1, thus $M_1' \xrightarrow{t_1} M'$ and clearly $M \equiv M'$.
- if $t_1 = t$, then we immediately see that M is not equivalent with any other marking in $\mathcal{N}'$ as $M(p) \neq 0$, and the lemma thus holds as it only accounts for markings where $M(p) = 0$.
- Note that it is impossible for $t_1 = t'$ in this case, as t' is not enabled from the marking where $M_1(p) = 0$

2. Assume that $M_1(p) \neq 0$. This means there is no equivalent marking in $\mathcal{N}'$. We solve this problem is five steps:
   (a) First, we revert all the transition firings that has come before in both nets untill we reach a reverted marking M_{r_1} where $M_{r_1} \xrightarrow{t} M_{r_2}$ was fired in $\mathcal{N}$ and $M_{r_1} \equiv M'_{r_1}$.
   (b) We know the transition firing $M_{r_1} \xrightarrow{t} M_{r_2}$ has taken place as t is the only transition that has p in its postset.
   (c) From this reverted marking M_{r_1} we fire t, and from the following marking M_{r_2} we immediately fire $M_{r_2} \xrightarrow{t'} M_{r_3}$ which we know we can because $W(t,p) = W(p,t')$.
   (d) We thus ensure that t is immediately followed by t', and we match this by firing the merged transition $M'_{r_1} \xrightarrow{t''} M'_{r_2}$ in $\mathcal{N}'$ such that $M_{r_3} \equiv M'_{r_2}$
   (e) We then retrace the steps that led us to M_1, but now $M_1(p) = 0$ and we can use the above case.

□

Lemma 5.4. *If $M^{in} \dashrightarrow^* M'$ in $\mathcal{N}'$, then $M^{in} \dashrightarrow^* M$ in $\mathcal{N}$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let $M^{in} \dashrightarrow^m M'$ in $\mathcal{N}'$ where $m \geq 0$, then $M^{in} \dashrightarrow^* M$ such that $M \equiv M'$.

- $m = 0$ case: trivial for the same reason that the same case in Lemma 5.1 is trivial.
- $m + 1$ case: assume the lemma holds for m steps, this means that $M^{in} \dashrightarrow^m M'_1$ and thus $M^{in} \dashrightarrow^* M_1$ such that $M_1 \equiv M'_1$, we now do one more step such that $M'_1 \xrightarrow{t_1} M'$.
  - if $t_1 \neq t''$, it is trivial; we fire the same transition in $\mathcal{N}$ such that $M_1 \xrightarrow{t_1} M$ and $M' \equiv M$.
  - if $t_1 = t''$, we know that ${}^\diamond t'' =^\diamond t$ and $t''^\diamond = t^\diamond \cup t'^\diamond \setminus \{p\}$, as such, we know that t is enabled in M_1, and firing it will bring us to a marking where t' is enabled. By firing t' immediately afterward, we thus end up in a marking that has consumed all tokens from p and has outputted tokens in the same places as t'', thus by doing the transition firing $M_1 \xrightarrow{t} M_2 \xrightarrow{t'} M$, we have ensured that $M \equiv M'$.

□

Theorem 5.2. *The modal workflow net $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound.*

Proof. As with Theorem 5.1

$\Rightarrow$ Assume that $\mathcal{N}$ is sound, for $\mathcal{N}'$ to be sound, it has to fullfill the two conditions of Definition 3.21.

Condition a) Let M' be a reachable marking in $\mathcal{N}'$. For $\mathcal{N}'$ to fulfill this condition, it must be possible to perform a transition firing sequence such that $M' \rightarrow^* M^{out}$. We know from Lemma 5.4 that any reachable marking in $\mathcal{N}'$ has a corresponding marking in $\mathcal{N}$. Since we know $\mathcal{N}$ is sound, this means that $\mathcal{N}$ can always reach M^{out}, and because of Lemma 5.3 we know that even though not every marking in $\mathcal{N}$ has an equivalent marking in $\mathcal{N}'$, we can always manipulate the transition firings such that whenever t is fired, t' is fired immediately afterward which is matched by firing t'' in $\mathcal{N}'$. Thus $\mathcal{N}'$ always able to end up in $M^{out\prime}$, and since $M^{out} \equiv M^{out\prime}$ this implies that $M^{out} = M^{out\prime}$.

Condition b) Let M' be a reachable marking in $\mathcal{N}'$ where $M'(o) \geq 1$. Because M' is reachable, this means it has an equivalent marking M in $\mathcal{N}$ because of Lemma 5.4, such that $M \equiv M'$, because $\mathcal{N}$ is sound, this means $M = M^{out}$, and thus $M^{out} \equiv M'$, which implies $M' = M^{out}$.

$\Leftarrow$ Assume that $\mathcal{N}'$ is sound, for $\mathcal{N}$ to be sound, it has to fullfill the two conditions of Definition 3.21.

Condition a) Let M be a reachable marking in $\mathcal{N}$, for $\mathcal{N}$ to fulfill this condition, it must be possible to perform a transition firing sequence such that $M \rightarrow^* M^{out}$. We know from Lemma 5.3 that any reachable marking in $\mathcal{N}$ has a corresponding marking in $\mathcal{N}'$. Since we know $\mathcal{N}'$ is sound, this means that $\mathcal{N}'$ can always reach M^{out}, and because of Lemma 5.4, we know that $\mathcal{N}$ can do an equivalent transition firing and end up in M^{out}.

Condition b) Let M be a reachable marking in $\mathcal{N}$ where $M(o) \geq 1$. Because M is reachable, this means if $M(p) = 0$ it has an equivalent marking M' in $\mathcal{N}'$ because of Lemma 5.3, such that $M \equiv M'$, because $\mathcal{N}'$ is sound, this means $M' = M^{out}$, and thus $M^{out} \equiv M$, which implies $M = M^{out}$. if $M(p) \neq 0$ then Lemma 5.3 still shows a way to work around this by reverting transition firings untill t was fired an immediately firing t' afterward, such that a firing of t is always succeeded by a firing of t' to reach a marking where $M(p) = 0$ and the lemma, as such, holds.

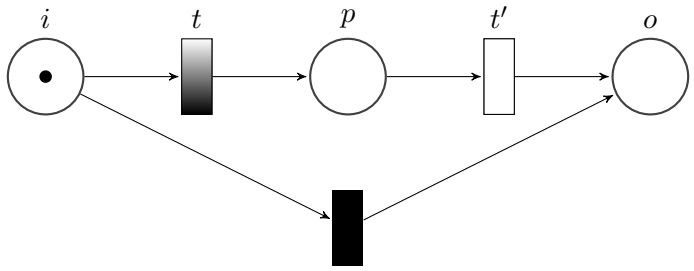
Thus we have proven that $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound. □

Remark. *It is important that t' is not a may transition, as this renders us unable to prove soundness is preserved, a counterexample is given in Figure 11.*

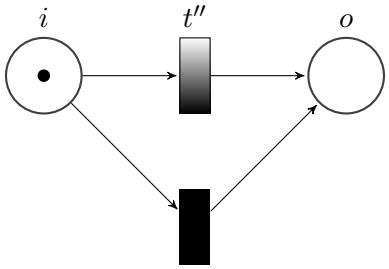
5.3 Parallel place merger

Definition 5.5 (Parallel place merger)**.** Given a modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$, if for two places p and $p' \in P \setminus \{i, o\}$

- $p \neq p'$
- ${}^{\Diamond}p = {}^{\Diamond}p'$
- $p^{\Diamond} = p'^{\Diamond}$



(a) May transition t', net unsound.



(b) Attempted Definition 5.3 on a), net is now sound

Figure 11: Why t' of Definition 5.3 has to be a must transition

- for all $t \in {}^{\diamond}p, W(t,p) = W(t,p')$, and
- for all $t \in p^{\diamond} W(p,t) = W(p',t)$

then one of the two places, and any arcs connected to it, can be removed without affecting the soundness of the net, as seen in Figure 12.

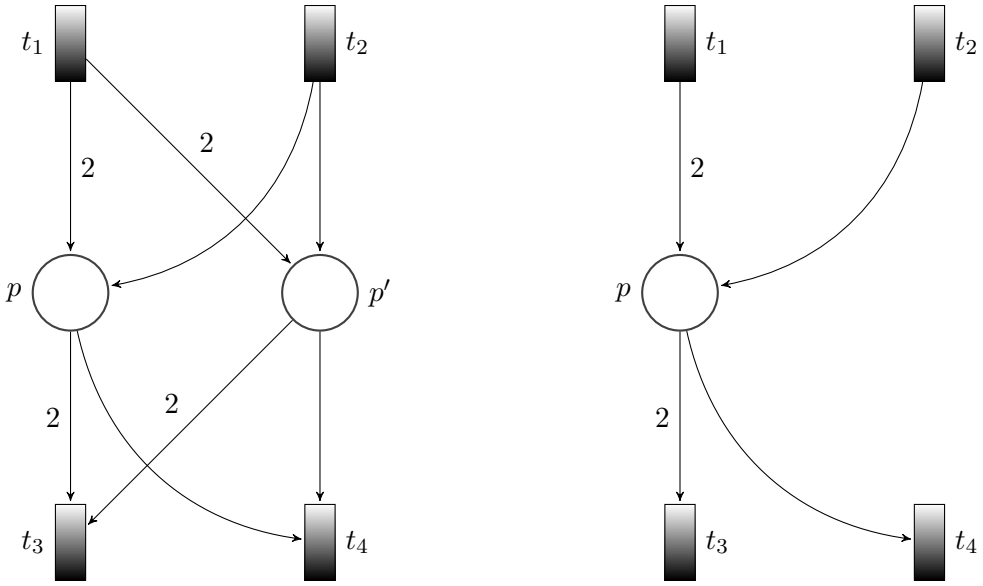
For the rest of Section 5.3 let $\mathcal{N} = (P, T^{\square}, T^{\diamond}, F, W, L)$ be a modal workflow net. Then $\mathcal{N}'$ is the modal workflow net obtained by performing the reduction from Definition 5.5 once on $\mathcal{N}$

Definition 5.6 (Marking equivalence for parallel place merger)**.** Let M be a marking in a modal workflow net $\mathcal{N}$ and let M' be a marking in the modal workflow net $\mathcal{N}'$. We say that marking M is equivalent to M', written $M \equiv M'$ if $M(q) = M'(q)$ for all $q \in P \setminus \{p'\}$ and $M(p') = M(p)$.

Lemma 5.5. *If M^{in} $\dashrightarrow^{*}$ M in $\mathcal{N}$, then M^{in} $\dashrightarrow^{*}$ M' in $\mathcal{N}'$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let M^{in} $\dashrightarrow^{m}$ M in $\mathcal{N}$ where $m \geq 0$, then M^{in} $\dashrightarrow^{*}$ M' in $\mathcal{N}$ such that $M \equiv M'$.

- $m = 0$ case: Trivial. Because doing no transition firing from M^{in} leaves us at M^{in} in $\mathcal{N}$, we do nothing in $\mathcal{N}'$ as well. This leaves us with $M^{in} \equiv M^{in}$, which is obviously true.
- $m + 1$ case: Assume the lemma holds for m steps, this means that M^{in} $\dashrightarrow^{m}$ M_1 and thus M^{in} $\dashrightarrow^{*}$ M_1' such that $M_1 \equiv M_1'$, we now do one more step such that $M_1 \overset{t_1}{\dashrightarrow} M$.
  - if $t_1 \notin^{\diamond} p$, this is trivial, as M_1' can also fire t_1, thus $M_1' \overset{t_1}{\dashrightarrow} M'$ and clearly $M \equiv M'$.



(a) Before (b) After

Figure 12: Definition 5.5 visualized: one place can be combined without impacting the computations of the net.

- if $t_1 \in^{\diamond} p$, we fire the same transition in $\mathcal{N}'$ and note that from this new marking M' in $\mathcal{N}'$ we can fire the same transitions we can in $\mathcal{N}$, and as $M(p') = M(p)$, then $M \equiv M'$.

□

Lemma 5.6. *If $M^{in} \dashrightarrow^* M'$ in $\mathcal{N}'$, then $M^{in} \dashrightarrow^* M$ in $\mathcal{N}$ such that $M \equiv M'$.*

Proof. We prove this by induction on the number of transition firings. Let $M^{in} \dashrightarrow^m M'$ in $\mathcal{N}'$ where $m \geq 0$, then $M^{in} \dashrightarrow^* M$ such that $M \equiv M'$

- $m = 0$ case: trivial. Because doing no transition firing from M^{in} leaves us at M^{in} in $\mathcal{N}$, we do nothing in $\mathcal{N}'$ as well. This leaves us with $M^{in} \equiv M^{in}$, which is obviously true.
- $m + 1$ case: assume the lemma holds for m steps, this means that $M^{in} \dashrightarrow^m M_1'$ and thus $M^{in} \dashrightarrow^* M_1$ such that $M_1 \equiv M_1'$, we now do one more step such that $M_1' \xrightarrow{t_1} M'$.
  - if $t_1 \notin^{\diamond} p$, this is trivial, as M_1 can also fire t_1, thus $M_1 \overset{t_1}{\dashrightarrow} M$ and clearly $M \equiv M'$.
  - if $t_1 \in^{\diamond} p$, we fire the same transition in $\mathcal{N}$ and note that from this new marking M in $\mathcal{N}$ we can fire the same transitions we can in $\mathcal{N}'$, and as $M(p') = M(p)$, then $M \equiv M'$.

□

Theorem 5.3. *The modal workflow net $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound.*

Proof. As Lemma 5.5 and Lemma 5.6 prove, $\mathcal{N}$ can fire a transition if and only if $\mathcal{N}'$ can fire the same transition. It is thus impossible for the two nets to end up in markings that are not equivalent, and hence $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound. □

5.4 Parallel transition merger

Definition 5.7 (Parallel transition merger)**.** Given a modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ with two transitions t, t', such that

- $t \neq t'$
- ${}^{\bullet}t = {}^{\bullet}t'$
- $t^{\bullet} = t'^{\bullet}$
- $W(p,t) = W(p,t')$ for all $p \in P$, and
- $W(t,p) = W(t',p)$ for all $p \in P$

then:

a) if $t \in T^{\Box}$ and $t' \in T^{\Diamond}$, or $t, t' \in T^{\Box}$ then we can remove t' and any arcs connected to t' without affecting soundness as seen in Figure 13a and Figure 13b, or

b) if $t, t' \in T^{\Diamond} \setminus T^{\Box}$ then we can remove either t or t' (but not both) and any arcs connected to it without affecting soundness as seen in Figure 13c and Figure 13d.

For the rest of Section 5.4 let $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ be a modal workflow net. Then $\mathcal{N}'$ is the modal workflow net obtained by performing the reduction from Definition 5.7 once on $\mathcal{N}$.

Theorem 5.4. *The modal workflow net $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound.*

Proof. Observe that any marking M which allows the transition firing $M \xrightarrow{t} M_1$ also allows the transition firing $M \xrightarrow{t'} M_2$, and observe that $M_1 = M_2$. Thus one transition is redundant and can be removed without impacting the soundness of the net as long as we always preserve a *must* transition if available, as otherwise we could reach a marking which violates Definition 3.21 Item a by being reachable, yet unable to reach M^{out} by only firing must transitions. □

5.5 Transition loop

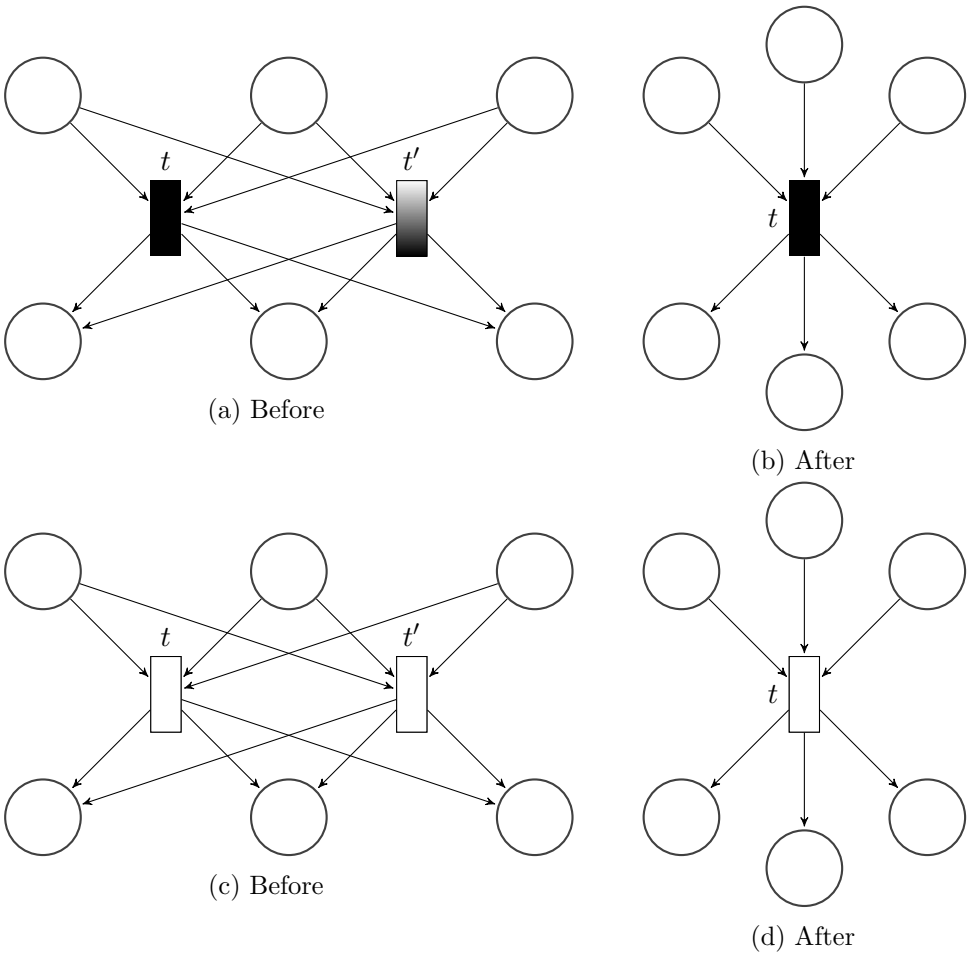
Definition 5.8 (Looping transition removal)**.** Given a modal Petri net $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ with a transition t, such that

- $t^{\Diamond} = {}^{\Diamond}t$, and
- $W(p,t) = W(t,p)$ for all $p \in t^{\Diamond}$

then remove t and any arcs connected to it as seen in Figure 14.

For the rest of Section 5.5 let $\mathcal{N} = (P, T^{\Box}, T^{\Diamond}, F, W, L)$ be a modal workflow net. Then $\mathcal{N}'$ is the modal workflow net obtained by performing the reduction from Definition 5.8 once on $\mathcal{N}$.

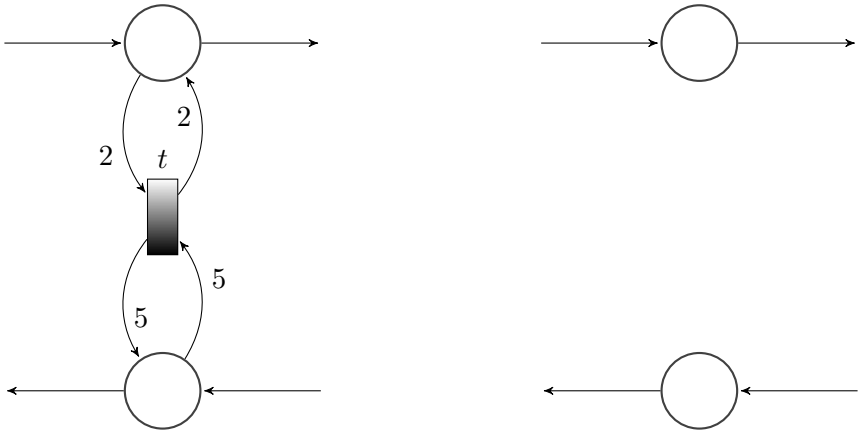
Theorem 5.5. *The modal workflow net $\mathcal{N}$ is sound if and only if $\mathcal{N}'$ is sound.*



(a) Before (b) After

(c) Before (d) After

Figure 13: Definition 5.7 visualized: One transition is superflous and can be removed.



(a) Before (b) After

Figure 14: Definition 5.8 visualized: The transition changes nothing when fired, it can thus be removed.

Proof. $\Rightarrow$ Observe two markings in $\mathcal{N}$, M and M' such that $M \xrightarrow{t} M'$ is a possible transition firing. We now observe that $M = M'$ in every aspect. Thus firing the transition t changes nothing about the marking, and thus $\mathcal{N}'$ is sound if $\mathcal{N}$ is sound.

$\Leftarrow$ Observe a marking M in $\mathcal{N}$. If this marking is sound, then adding a transition which can be fired to reach it, again and again, does not alter the soundness. Thus $\mathcal{N}$ is sound if $\mathcal{N}'$ is sound.

□

5.6 Reduction example

Recall Figure 4. Through repeated applications of the reductions introduced in this chapter, we can reduce it down to a much simpler and easy to analyse net:

1. Observe that the entire chain of transitions starting with *Begin Accessorial Treatments* and ending in the place *Full* is a series of transitions and places, and can as such be reduced to a single may transition through repeated applications of the sequential transition and place reductions.

2. This, in turn, leaves three parallel transitions from *Ready to Dry* to *Full*, which can be reduced to a single must transition through repeated applications of parallel transition merger.

3. Now Observe the parallel tasks of the rotary brushes can be reduced to singular places through sequential place reductions, and then these singular places can be reduced to a single place through repeated applications of the parallel place reduction rule.

4. Next, the *Foam Again* transition is a looping transition, and can be removed per Definition 5.8

5. This leaves us with a sequence of must transitions and places starting at *Full* and ending at *Full*. Through repeated applications of the sequential place and transition reductions, we can reduce this sequence down to a single looping must transition with input and output to *Full*, we name this transition *Wash Car*.

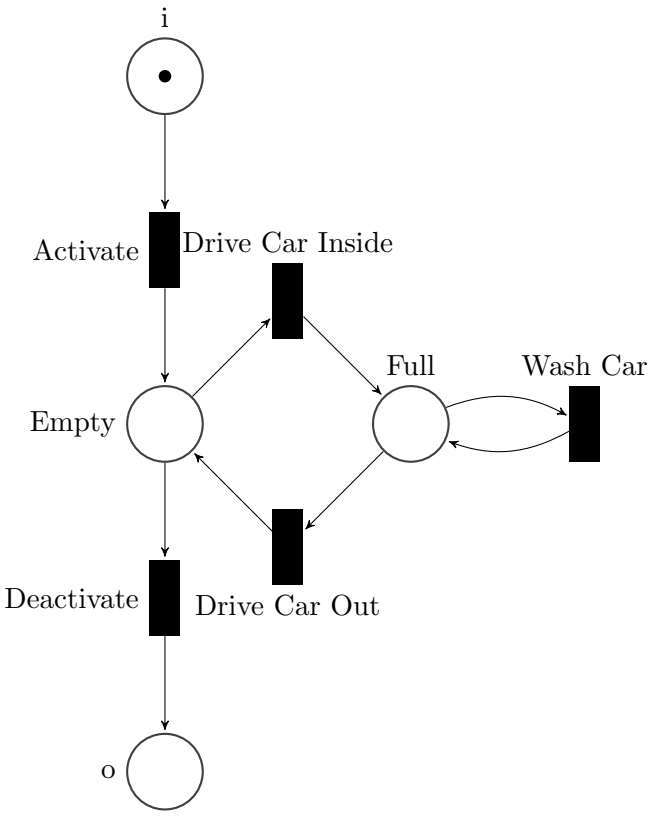
Through repeated applications of the reductions introduced in this paper, we have reduced away the entire wash part of the net, to only a single transition as seen in Figure 15a. A quick look at the figure will reveal that this net is obviously sound, but even if it was not immediately apparent, our reductions allow us to continue reducing this particular net until there is nothing left except for the places i and o, and a transition connecting them:

1. *Wash Car* is a looping transition, and can be removed.

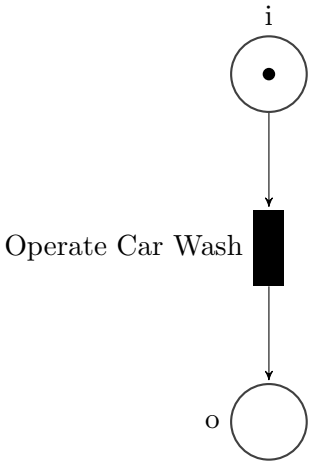
2. We can then use the sequential transition merger case a) on the transitions *Drive Car Inside* and *Drive Car Out*, along with the place *Full*.

3. This leaves us with a looping transition originating from *Empty*, and we can remove it just the same as we have done with other looping transitions.

4. Finally, we are left with a sequence of three places and two transitions. We do a final sequential transition merger the transitions *Activate* and *Deactivate* and name the resultant transition *Operate Car Wash*.



(a) The partway reduced Figure 4.



(b) The fully reduced Figure 4, obviously sound.

Figure 15: The reduction of Figure 4.

This leaves us with Figure 15b and because this reduced net is sound obviously sound, we know from the main result given in this section that the unreduced net in Figure 4 is sound as well.

Remark. *Not every net can be reduced down to just i, o, and the transition connecting them. An example of an unreducible, yet sound, net can be seen in Figure 16.*

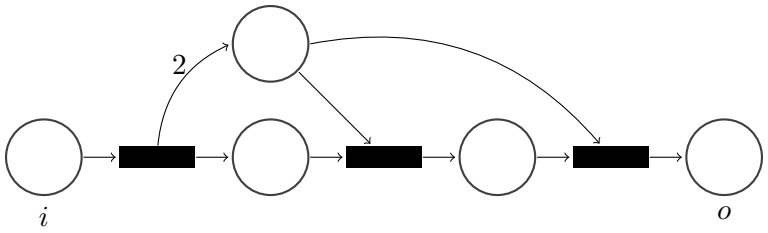


Figure 16: Example of a sound workflow unreducible by the reductions given in sections 5.1-5.5.

6 Modal workflow nets with resources

Sometimes having a modal Petri net with tokens is not enough, we also need to consider a finite amount of different resources; for instance, the amount of wood at a lumber mill, or the amount of foam/soap in a carwash. In this case, resources can be added to a Petri net, allowing for the creation and consumption of resources during the course of the Petri net [1].

Definition 6.1 (Modal Petri net with resources)**.** A Modal Petri net with resources is a tuple $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ where.

- $(P, F, T^{\Diamond}, T^{\Box}, W, L)$ is modal Petri net,
- P_c is the set of control places,
- P_r is the set of resource places such that $P_c \cap P_r = \emptyset$ and $P_c \cup P_r = P$, and
- for every transition $t \in T^{\Diamond}$ we require that $t^{\Diamond} \cup {}^{\Diamond}t \cap P_c \neq \emptyset$.

We depict any $p \in P_r$ as diamonds while retaining the circular shape for control places, see Figure 17 for an example. As before, we define the workflow as follows:

Definition 6.2 (Modal workflow net with resources)**.** A modal Petri net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ is a modal workflow net with resources (MWFNR) if

- $\mathcal{R}$ has two special places, the source place $i \in P_c$ and the sink place $o \in P_c$ such that ${}^{\Diamond}i = o^{\Diamond} = \emptyset$, ${}^{\Diamond}p \neq \emptyset$, and $p^{\Diamond} \neq \emptyset$ for all other $p \in P_c$.

We also redefine what M^{in} and M^{out} means for MWFNR.

Definition 6.3 (Start marking)**.** A marking M^{in} is a *start* marking if $M^{in}(p) = 1$ if $p = i$ and $M^{in}(p) = 0$ for all other $p \in P_c$. The set of all M^{in} is denoted by $\mathcal{M}^{in}$.

Definition 6.4 (Finish marking)**.** A marking M^{out} is a *finish* marking if $M^{out}(p) = 1$ if $p = o$ and $M^{out}(p) = 0$ for all other $p \in P_c$. The set of all M^{out} is denoted by $\mathcal{M}^{out}$.

Remark. *Note we do not care about the amount of resources at point of initialisation or termination.*

An example of a MWFNR in one of many possible M^{in} markings can be seen in Figure 17.

Because soundness prior to this chapter was determined from a single, static M^{in} and M^{out}, we cannot merely say that a given MWFNR is sound without providing an explicit M^{in} for which is it sound. Hence, we redefine what soundness means for MWFNR.

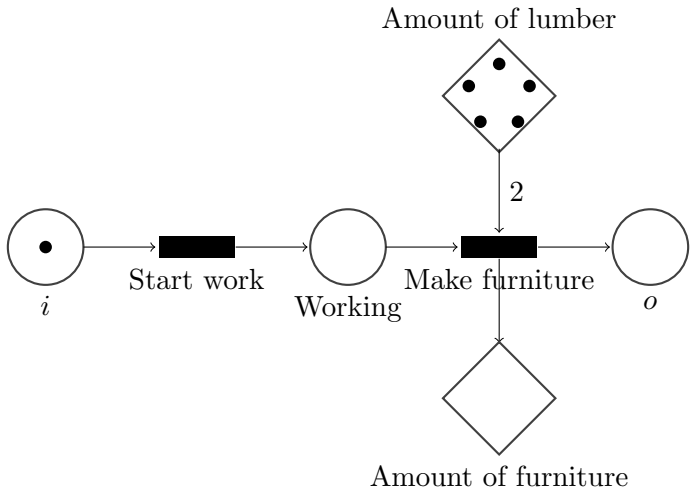


Figure 17: A modal workflow with resources.

Definition 6.5 (Sound for a given M^{in})**.** A MWFNR is sound for a given $M^{in} \in \mathcal{M}^{in}$ if:

a) whenever $M^{in} \dashrightarrow^* M$ then there exists a firing sequence $M \rightarrow^* M^{out}$ for some $M^{out} \in \mathcal{M}^{out}$.

b) if $M^{in} \dashrightarrow^* M$ such that $M(o) \geq 1$, then $M \in \mathcal{M}^{out}$.

But by redefining soundness for modal workflow nets with resources, we now run into a different problem, that of repeatability. Imagine the MWFNR seen in Figure 17 as being a real world workflow that is done once each day; a worker signs in, takes two pieces of wood from the storage and makes a piece of furniture. As it is, while the workflow may be sound now, it will not continue to be so; lumber will be used each day and there are no means depicted by which we can restock the supply of lumber. In the case of Figure 17, this means that by the third day there will be insufficient lumber to make furniture, and the workflow thus enters a deadlocked state. Thus, standard soundness is no longer sufficient, and we need a stricter version that ensures a net is both sound now and sound no matter how many times we run it without adjusting resources left over from a previous run. We call this *Infinitary soundness*

Definition 6.6 (Infinitary soundness for a given M^{in})**.** A modal workflow net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ is infinitary sound for a given $M^{in} \in \mathcal{M}^{in}$ if

a) $\mathcal{R}$ is sound for the given M^{in}, and

b) for every $M^{out} \in \mathcal{M}^{out}$ such that $M^{in} \dashrightarrow^* M^{out}$, $\mathcal{R}$ is infinitary sound for the marking M' where $M'(p) = M(p)$ for all $p \in P_c \cup P_r \setminus \{i, o\}$, $M'(i) = 1$, and $M'(o) = 0$.

Remark. *Note that Definition 6.6 is a recursive definition. A net is infinitary sound if it can be run an unbounded number of times without adjusting the number of resources left over from the preceding run.*

Let us return to the example seen in Figure 17 - this net is sound as defined in Definition 6.5, but it is not infinitary sound as we have just defined; eventually the net will reach a point where insufficient lumber is available to produce a piece of furniture and will thus be stuck.

Adding a transition like in Figure 18 will make the net infinitary sound, but it will also make it unbounded for the given M^{in}.

Definition 6.7 (K-bounded for the given M^{in})**.** A modal workflow net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ is K-bounded for the given $M^{in} \in \mathcal{M}^{in}$ if there exists a $K \in \mathbb{N}$ such that for every reachable marking M in $\mathcal{R}$, $M(p) \leq K$ for all $p \in P$.

A way to make Figure 18 K-bounded could be to make an alteration to the net as seen in Figure 19, but in doing this, we are merely postponing the problem; while Figure 19 may be infinitary sound and K-bounded, if one was to run this net an unbounded number of times as described in Definition 6.6b), one would see that the amount of furniture would become unbounded in the process, our definition of boundedness for MWFNR is inadequate when discussing these infinitary sound nets, and thus we introduce the concept of *Infinitary K-Boundedness*.

Definition 6.8 (infinitary K-boundedness for the given M^{in})**.** A modal workflow net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ is infinitary bounded for a given M^{in} if

- $\mathcal{R}$ is K-bounded for the given M^{in}, and
- for every $M^{out} \in \mathcal{M}^{out}$ such that $M^{in} \dashrightarrow^* M^{out}$, $\mathcal{R}$ is infinitary K-bounded for the marking M' where $M'(p) = M(p)$ for all $p \in P_c \cup P_r \setminus \{i, o\}$, $M'(i) = 1$, and $M'(o) = 0$.

This infinitary K-boundedness definition is reduceable to K-boundedness through the algorithm seen in Algorithm 2.

Algorithm 2 Reducing infinitary K-boundedness to K-boundedness

A modal workflow net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ is infinitary bounded for a given M^{in} if the modal Petri net with resources $\mathcal{R}' = (P', P'_c, P'_r, T^{\Box\prime}, T^{\Diamond\prime}, F', W', L')$ where:

- $P' = P$
- $P'_c = P_c$
- $P'_r = P_r$
- $T^{\Box\prime} = T^{\Box} \cup \{again\}$
- $T^{\Diamond\prime} = T^{\Diamond} \cup \{again\}$
- $F' = F \cup \{(o, again)\} \cup \{(again, i)\}$
- $W'(o, again) = 1, W'(again, i) = 1$, $W'(p, t) = W(p, t)$ for all $(p, t) \in F$, and $W'(t, p) = W(t, p)$ for all $(t, p) \in F$
- $L' = L$

is K-bounded for the given M^{in}.

An example of Algorithm 2 performed on Figure 19 can be seen in Figure 20, and as we can now see, Figure 19 may be infinitary sound and bounded for the given M^{in}, but it is not infinitary bounded for the given M^{in}. A net which is both infinitary sound and infinitary bounded for the M^{in} depicted can be seen in Figure 21.

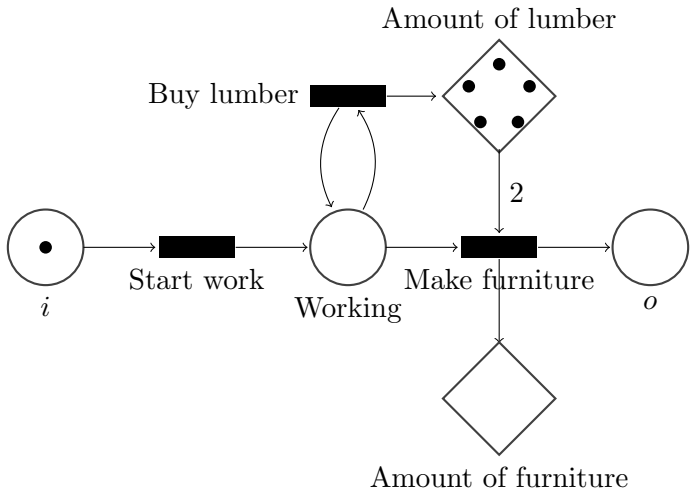


Figure 18: An infinitary sound, unbounded modal workflow with resources.

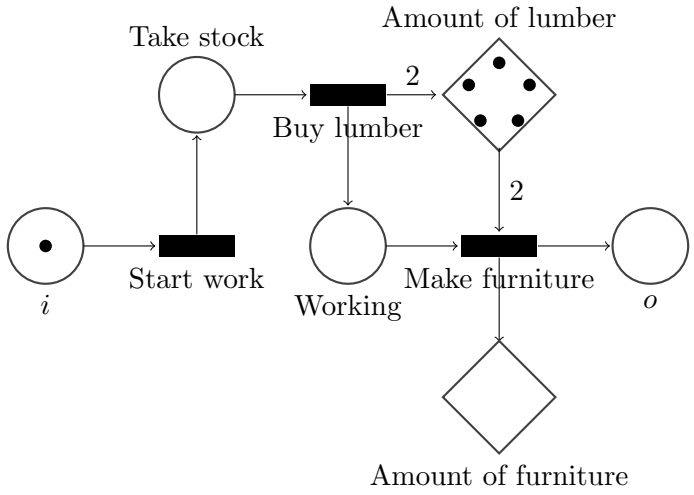


Figure 19: An infinitary sound, 7-bounded modal workflow with resources.

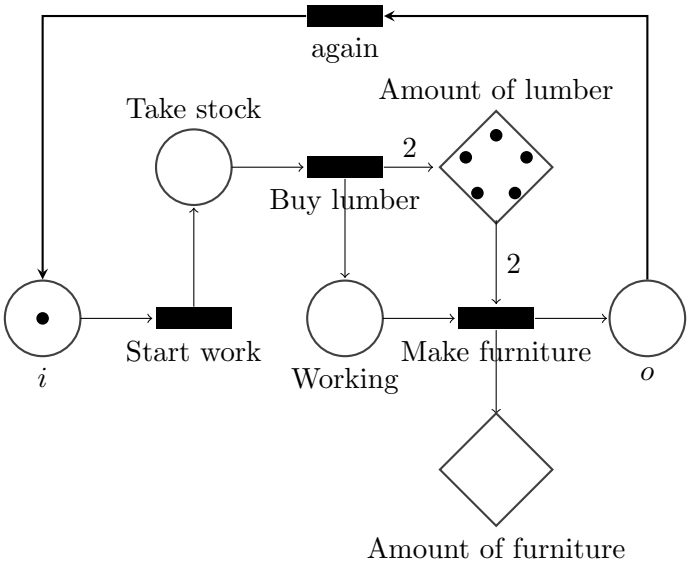


Figure 20: Algorithm 2 applied to Figure 19. As we can see, this Petri net is not bounded, and therefore Figure 19 is not infinitary bounded.

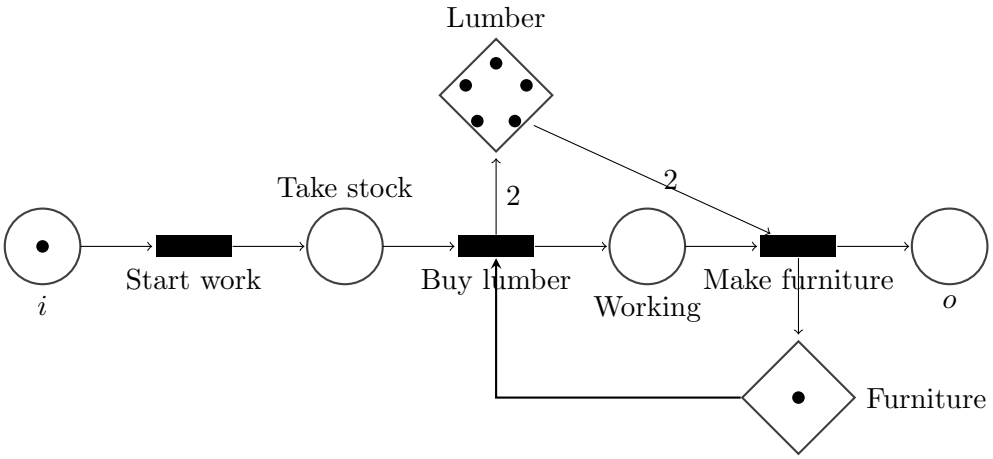


Figure 21: An infinitary sound, infinitary 7-bounded modal workflow with resources.

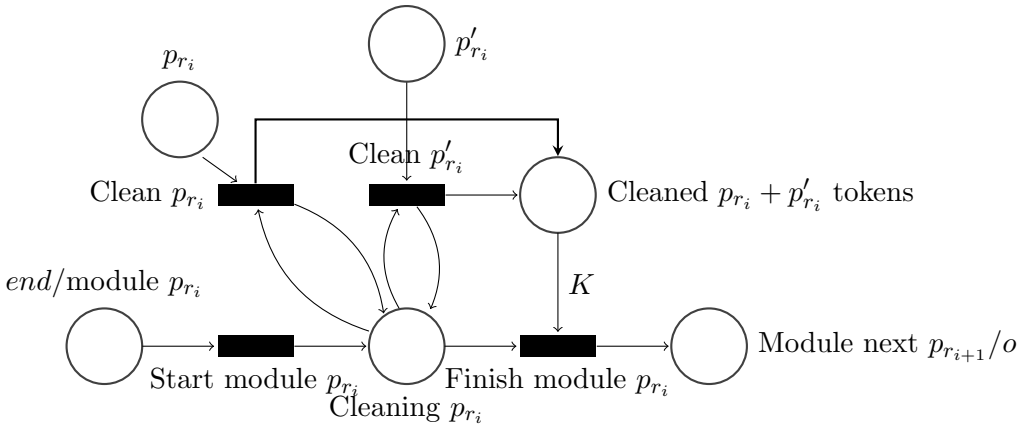


Figure 22: An example of a cleanup module for a K-bounded net.

6.1 Results

Lemma 6.1. *Infinitary soundness for infinitary bounded modal workflow nets with resources is reducable to standard soundness for bounded modal workflow nets in polynomial time.*

Proof. Given an infinitary K-bounded modal workflow net with resources $\mathcal{R} = (P, P_c, P_r, T^{\Box}, T^{\Diamond}, F, W, L)$ where, without loss of generality we assume $P_r = \{p_{r_1}, p_{r_2} \dots p_{r_m}\}$ we construct the modal workflow net $\mathcal{N} = (P, T^{\Box\prime}, T^{\Diamond\prime}, F', W', L')$ such that $\mathcal{R}$ is infinitary sound if and only if $\mathcal{N}$ is sound. We do this by renaming the original i and o place to *start* and *end* respectively, changing every resource place to a standard place, adding complementary places to every former resource place, adding new i and o places, adding a number of 'clean up modules', each connecting to a specific former resource place and said former resource place's complementary place as seen in Figure 22, and, finally, adding two new transitions, *Initialize* and *Again*. *Initialize* takes a token from i and outputs a token to *start* and to every former resource place such that they have the same amount of tokens as they had in M^{in}, it then adds tokens to their complementary places such that the original place's tokens in addition to the complementary place's tokens equals the K-bound. *Again* takes a token from *end* and outputs a token in *start*, allowing the part of $\mathcal{N}$ which that corresponds to $\mathcal{R}$ to be run again. Thus we get:

- P $= P_c \cup P_r \cup \{Start\} \cup \{End\} \cup \{p' \mid p_{r_i} \in P_r\} \cup \{\text{Cleaning } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Cleaned } p_{r_i} + p'_{r_i} \text{ tokens} \mid p_{r_i} \in P_r\} \cup \{\text{module } p_{r_i} \mid p_{r_i} \in P_r \setminus \{p_{r_1}\}\}$
- $T^{\Box\prime} = T^{\Box} \cup \{again\} \cup \{\text{Start module } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Finish module } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Clean } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Clean } p'_{r_i} \mid p_{r_i} \in P_r\}$
- $T^{\Diamond\prime} = T^{\Box} \cup \{again\} \cup \{\text{Start module } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Finish module } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Clean } p_{r_i} \mid p_{r_i} \in P_r\} \cup \{\text{Clean } p'_{r_i} \mid p_{r_i} \in P_r\}$
- $F' = F + \{(End, Again)\} \cup \{(Again, Start)\} \cup \{(Initialize, Start)\} \cup \{(Initialize, p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{(Initialize, p'_{r_i}) \mid p_{r_i} \in P_r\} \cup \{(p'_{r_i}, t) \mid p_{r_i} \in P_r \text{ where } (t, p_{r_i}) \in F\} \cup \{(t, p'_{r_i}) \mid p_{r_i} \in P_r \text{ where } (p_{r_i}, t) \in F\} \cup \{(End, \text{Start module } p_{r_1})\} \cup \{(\text{module } p_{r_i}, \text{ Start module } p_{r_i}) \mid p_{r_i} \in P_r \setminus \{p_{r_1}\}\} \cup \{(\text{Start module } p_{r_i}, \text{ Cleaning } p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{(\text{Cleaning } p_{r_i}, \text{Finish module } p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{(\text{Finish module } p_{r_i}, \text{module } p_{r_{i+1}}) \mid p_{r_i} \in P_r \setminus p_{r_m}\} \cup$

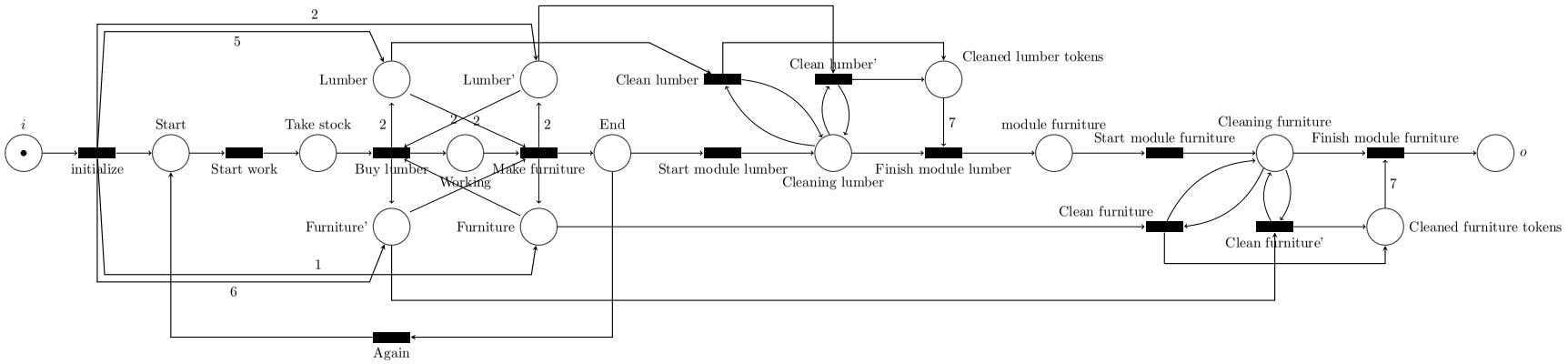


Figure 23: The modal workflow which is sound if and only if Figure 21 is infinitary sound.

$\{$(Finish module p_{r_m}, o)$\} \cup \{(p_{r_i}$, clean $p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{(p'_{r_i}$, clean $p'_{r_i}) \mid p_{r_i} \in P_r\} \cup$ $\{$(cleaning p_{r_i}, clean $p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{$(clean p_{r_i},cleaning $p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{$(cleaning p_{r_i}, clean $p'_{r_i} \mid p_{r_i} \in P_r\} \cup \{$(clean p'_{r_i},cleaning $p_{r_i}) \mid p_{r_i} \in P_r\} \cup \{$(clean p_{r_i},cleaned p_{r_i} tokens$) \mid p_{r_i} \in P_r\} \cup \{$(clean p'_{r_i},cleaned p_{r_i} tokens$) \mid p_{r_i} \in P_r\}$

- $W'(Initialize, p_{r_i}) = M^{in}(p_{r_i}) \; \forall p_{r_i} \in P_r$, $W'(Initialize, p'_{r_i}) = K - M^{in}(p_{r_i}) \; \forall p_{r_i} \in P_{r_i}$, $W'(t, p'_{r_i}) = W(p_{r_i}, t) \; \forall p_{r_i} \in P_r$, $W'(p'_{r_i}, t) = W(t, p_{r_i}) \; \forall p_{r_i} \in P_r$, W'(cleaned p_{r_i}, finish module p_{r_i}) $= K \; \forall p_{r_i} \in P_{r_i}$, $W'(p, t) = W(p, t) \; \forall (p, t) \in F$, $W'(t, p) = W'(t, p)$ $\forall (t, p) \in F$ and every single other new arc has a weight of 1.

An example of this performed on Figure 21 can be seen in Figure 23, which is sound if Figure 21 is infinitary sound. As this transformation creates four new places and four new transitions per resource place, this means it can be done in polynomial space. With this transformation done, we now wish to prove that $\mathcal{R}$ is infinitary sound if and only if $\mathcal{N}$ is sound. This will be done in two parts

$\Rightarrow$ If $\mathcal{R}$ is infinitary sound, this means it fulfils the soundness criteria of Definition 6.5 and, and from the marking described in Definition 6.6b) it remains infinitary sound. For $\mathcal{N}$ to be sound, it must fulfil the soundness criteria of Definition 3.21:

a) Let M be a reachable marking in $\mathcal{N}$. For $\mathcal{N}$ to fulfil this case, it must be possible to reach M^{out} from this marking.

- Once *Initialize* is fired, $\mathcal{N}$ will be in a marking corresponding to the given $M^{in} \in \mathcal{M}^{in}$ of $\mathcal{R}$, from this marking $\mathcal{N}$ can fire any transition that $\mathcal{R}$ can fire, and any marking that requires $\mathcal{R}$ to be 'reset' per Definition 6.6b) can be reached by firing *Again*. Thus, any marking in $\mathcal{R}$ can be reached in $\mathcal{N}$. We know that $\mathcal{R}$ can always reach a $M^{out} \in \mathcal{M}^{out}$, and thus since we renamed o to *end* in $\mathcal{N}$, the only markings reachable in $\mathcal{N}$ where $M(end) \geq 1$ are markings where $M(end) = 1$, from any of these markings, we can initiate a cleanup of the net that allows us to reach $\mathcal{N}$'s M^{out}. Thus $\mathcal{N}$ can always reach M^{out} from any reachable marking.

b) Let M be a reachable marking in $\mathcal{N}$ where $M(o) \geq 1$, we need to show that $M = M^{out}$.

- We know from the above case that if $\mathcal{R}$ is infinitary sound, the only markings where $M(end) \geq 1$ are markings where $M(end) = 1$. Since the cleanup of $\mathcal{N}$ can only be initiated from *end*, and the entire sequence of clean-up modules clean up every token in the net before outputting a token in o, this means that since the only markings where $M(end) \geq 1$ are markings where $M(end) = 1$, the only marking where $M(o) \geq 1$ is M^{out}

$\Leftarrow$ If $\mathcal{N}$ is sound, this means it fulfils the soundness criteria of Definition 3.21. To prove $\mathcal{R}$ is infinitary sound, we need to show it fulfils the criteria of Definition 6.6:

1. $\mathcal{R}$ is sound. This means it must fulfil the two criteria of Definition 3.21
   (a) Let M be a reachable marking in $\mathcal{R}$. We can reach an equivalent marking in $\mathcal{N}$ by firing the *initialize* transition followed by the same transitions used to reach M. Because we know $\mathcal{N}$ is sound, this means that any marking reached in $\mathcal{N}$ can reach M^{out}, and any transition firing sequence $M^{in} \rightarrow M^{out}$ in $\mathcal{N}$ must go through a marking M' where $M'(end) = 1$. Because *end* is the sink place o

in $\mathcal{R}$, we thus know that any marking reached in $\mathcal{R}$ can reach a marking M_f where $M_f(o) = 1$, and we know that any other $p \in P$ where $M'(p) \geq 1$ in $\mathcal{N}$ have been converted to resource places, as otherwise $\mathcal{N}$ would not be sound. But soundness for *MWFNR* does not care about tokens in resource places, and thus $M_f = M^{out}$

(b) Let M be a reachable marking in $\mathcal{R}$ where $M(o) \geq 1$, we know from the above that the only such markings are markings where $M(o) = 1$ and $M(p_c) = 0$ for all $p_c \in P_c$. this means $M = M^{out}$.

2. $\mathcal{R}$ is infinitary sound
   - We can see that $\mathcal{N}$ has a transition *Again* which allows it to mimic the marking of Definition 6.6b). Thus, if $\mathcal{N}$ is sound, it can do the part of the net that $\mathcal{R}$ embodies an unbounded number of times and remain sound. Thus $\mathcal{R}$ is infinitary sound as $\mathcal{R}$ does not need to clean up the tokens in the resource places to retain this property.

Thus $\mathcal{R}$ is infinitary sound if and only if $\mathcal{N}$ is sound. □

Remark. *It is important to find the infinitary K-bound for the MWFNR before performing this transformation, as it is required to know the weights of some of the arcs.*

Theorem 6.1. *Infinitary soundness for infinitary bounded modal workflow nets with resources is decidable in* PSPACE*.*

Proof. Through Lemma 6.1, we know that infinitary soundness for infinitary bounded modal workflow nets with resources can be reduced to soundness for bounded modal workflow nets. Since we in Theorem 4.1 proved that soundness for bounded modal workflow nets is *PSPACE-complete*, and because our reduction in Lemma 6.1 is done in polynomial time, this means infinitary soundness for bounded modal workflow nets with resources is decidable in *PSPACE*. □

Remark. *We have shown that infinitary soundness for infinitary bounded modal workflow nets with resources is decidable in* PSPACE*, but whether or not infinitary soundness for standard bounded modal workflow nets with resources is decidable at all will be left as an open problem for the reader to research. Our approach will not work to this problem as the reduction to standard soundness is dependent on knowing the infinitary K-bound beforehand.*

7 Conclusion

Throughout this paper, we have reintroduced the modal workflow formula, and answered an open problem leftover from [9], proving that deciding soundness for bounded modal workflow nets is *PSPACE-complete*. To combat the state-explosion problem in order to ease the verifiability of soundness, we have introduced five different reductions and proven they preserve the soundness of any net upon which they are performed. Realising that the modal workflow model is not the end-all-be-all, we have introduced resources to the model to create modal workflow nets with resources, we have shown that the standard concepts of soundness and K-boundedness are not strict enough to be useful when considering modal workflow nets with resources. To alleviate this, we have introduced the concepts of *infinitary soundness* and *infinitary K-boundedness* which ensure that no matter how many times a net is run, it remains sound or bounded, respectively. We have then proven that for infinitary K-bounded

nets with resources infinitary soundness is decidable in PSPACE.
In summary the modal workflow net model allows for a model that is much easier to change and adapt according to the demands of the real world because of our result from [9] that if the specification of a net is sound, then as long as the specification is modal workflow refined, any refinement will also be sound, and with the structural reductions presented in this paper proven to preserve soundness, along with a more memory-efficient way of deciding soundness, it should theoretically be relatively easy to decide if a given specification is sound, even if it is large. Finally, The addition of resources allows the model to depict the consumption and creation of resources, opening it up to being more easily used for production-line workflows, which are widely used across the globe.

7.1 Future Work

Resources are but one extension of the modal workflow model. Drawing inspiration from non-modal Petri nets, we can find a number of additional extensions which would prove interesting to research.

7.1.1 Inhibitor Arcs

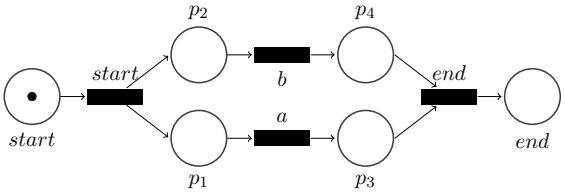
Inhibitor arcs [13] are a kind of arcs, usually depicted by a line with a circular tip, which prevents the enabling, and thus firing, of any transition if a place connected to the transition by an inhibitor arc is non-empty. Inhibitor arcs do not affect the marking reached by firing a transition to which they are connected. This means inhibitor arcs increase the modelling power of a Petri net to Turing completeness by adding the ability to test for zero in any place [15].

7.1.2 Reset Arcs

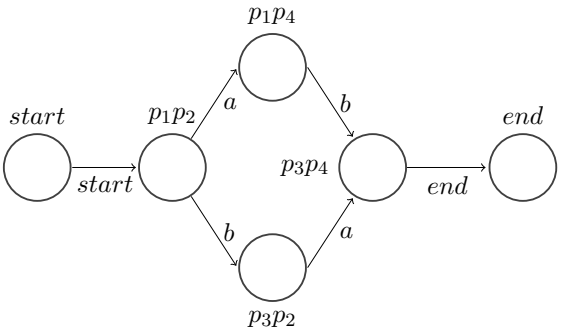
A reset arc [5] is a type of arc which connects a place to a transition, but which has no influence on the enabledness of the transition in question. When the transition is fired, the place(s) connected to the transition by reset arcs immediately have all their tokens, if any, consumed. If the places in question have no tokens, no change occurs in them and the transition fires as normal. This allows a Petri net with reset arcs to have 'localised' unboundedness in otherwise bounded nets which normal modal Petri nets do not. Reset arcs are, per definition, syntactic sugar - they do not add anything in terms of computability power that inhibitor arcs do not already add to an unextended net, and can be simulated by inhibitor arcs.

7.1.3 Asynchrony

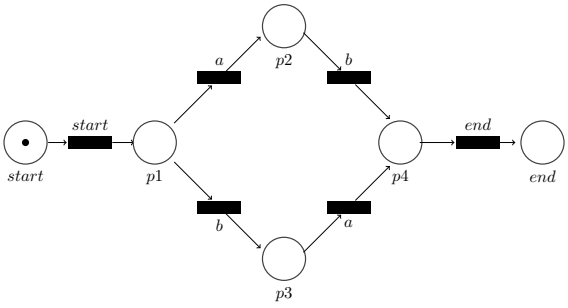
Asynchronous systems are systems which model asynchrony, which means the system has no global clock [17]. One might conclude that standard, untimed, Petri Nets fit this mold, but there is still an inferred clock in the sequencing of transition firings. To model concurrency, a Petri net thus allows for a splitting of tasks and transitions. The difference between concurrent and serial tasks can be seen in Figure 24a and Figure 24c respective, but this is still just a visual shorthand, if one is to look at the generated LTS, in Figure 24b and Figure 24d respectively, one would see that the two only differ in the names of the states. Asynchronous systems avoid this by depicting the concurrency in the generated LTS, and this could maybe be expanded upon further to allow for asynchronous modal refinement.



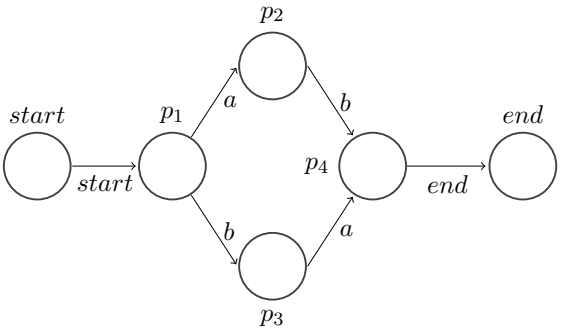
(a) How Petri nets model concurrency. a and b can be fired in whatever order desired, giving us the Petri net $start, a \mid b, end$



(b) The LTS generated by Figure 24a, concurrency is lost and this LTS is the one for $start, ab+ba, end$



(c) How Petri nets model choice. This net allows for a to be followed by b or vice-versa, giving us the Petri net $start, ab + ba, end$



(d) The LTS generated by Figure 24c, note how the actions are identical to Figure 24a

Figure 24: Two Petri nets modelling concurrency and choice, and the two generated LTS.

Acknowledgements

While this thesis was written in its entirety by me, it would not have been possible without the continued support of a great many people, without whom I would never have made it this far. I wish to extend my thanks to those people.

To Jiří Srba for his continuous support and supervision throughout the thesis project this past year. While the thesis is written by me, there is no doubt it would never have reached the level of standard and competence it has without his aid. Thank you for showing me that researching can be fun.

To Hans Hüttel, who helped me through a crisis a year ago and without whom I likely would have given up on university as a whole before the thesis had even been started. Thank you from the bottom of my heart for your support and vigour.

To Jytte Magdalene Severinsen, one elementary school teacher amongst many at Farsø Skole, but the one who, above all else, managed to take charge and ensure that the school's 'no bullying' policies were upheld. You believed in me back then, and you have continued to believe throughout my education. To this day, I am still thankful for what you did back then. Thank you for showing me the world is not apathetic.

To my family for their support throughout the education as a whole. Regardless of circumstance, through thick and thin, they have always been there to encourage me onward. This extends not just to aunts, uncles, grandparents and their families, but also to those whom I, while not family by blood, still consider family by relation. Thank you for your kind and supportive words and actions throughout my entire education.

And finally, to my friends. You are a small group, but it is a tightly-knit one I would not trade for anything. Thank you for countless great moments.

References

[1] Vladimir A Bashkin and Irina A Lomazova. Soundness of workflow nets with an unbounded resource is decidable. In *PNSE+ ModPE*, pages 61–75. Citeseer, 2013. 2, 28

[2] Nikola Beneš, Jan Křetínskỳ, Kim Guldstrand Larsen, and Jiri Srba. On determinism in modal transition systems. *Theoretical Computer Science*, 410(41):4026–4043, 2009. 4

[3] Marco Carbone, Thomas Hildebrandt, Gian Perrone, and Andrzej Wasowski. Refinement for transition systems with responses. *arXiv preprint arXiv:1207.4270*, 2012. 3

[4] Dennis Dams. abstract interpretation and partition refinement for model checking. 1996. URL `http://alexandria.tue.nl/extra3/proefschrift/PRF12B/9602314.pdf`. 2

[5] Catherine Dufourd, Alain Finkel, and Ph Schnoebelen. Reset nets between decidability and undecidability. In *Automata, Languages and Programming*, pages 103–115. Springer, 1998. 37

[6] Dorsaf Elhog-Benzina, Serge Haddad, and Rolf Hennicker. Refinement and asynchronous composition of modal petri nets. In *Transactions on Petri Nets and Other Models of Concurrency V*, pages 96–120. Springer, 2012. 2, 6

[7] Javier Esparza and Mogens Nielsen. Decidability issues for petri nets. *BRICS Report Series*, 1(8), 1994. 11

[8] Sheng-Uei Guan, Hsiao-Yeh Yu, and Jen-Shun Yang. A prioritized petri net model and its application in distributed multimedia systems. *Computers, IEEE Transactions on*, 47 (4):477–481, 1998. 2

[9] Jacob Buchreitz Harbo. Modal reasoning in petri net based workflows. 2016. URL `http://projekter.aau.dk/projekter/files/225447316/Afl.pdf`. 1, 2, 3, 8, 9, 11, 36, 37

[10] Kurt Jensen. *Coloured Petri nets*. Springer, 1987. 2

[11] J.A. Mateo, J. Srba, and M.G. Sørensen. Soundness of timed-arc workflow nets in discrete and continuous-time semantics. *Fundamenta Informaticae*, 140(1):89–121, 2015. doi: 10.3233/FI-2015-1246. 2

[12] Nichlas Korgaard Møller, Jacob Buchreitz Harbo, and Martin Christensen. Work smarter, not harder — basic structural reductions for timed-arc petri nets. 2014. URL `http://projekter.aau.dk/projekter/files/198206191/D603f14_Paper.pdf`. 14

[13] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989. 1, 14, 37

[14] Radek Pelánek. Fighting state space explosion: Review and evaluation. In *Formal Methods for Industrial Critical Systems*, pages 37–52. Springer, 2008. 1

[15] James L Peterson. Petri net theory and the modeling of systems. 1981. 37

[16] Carl Adam Petri. Kommunikation mit automaten. 1962. 2

[17] Chander Ramchandani. Analysis of asynchronous concurrent systems by petri nets. Technical report, DTIC Document, 1974. 37

[18] Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970. 11

[19] Natalia Sidorova and Christian Stahl. Soundness for resource-constrained workflow nets is decidable. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(3): 724–729, 2013. 2

[20] Wil MP van Der Aalst, Arthur HM Ter Hofstede, Bartek Kiepuszewski, and Alistair P Barros. Workflow patterns. *Distributed and parallel databases*, 14(1):5–51, 2003. 1

[21] Huaiqing Wang and Qingtian Zeng. Modeling and analysis for workflow constrained by resources and nondetermined time: An approach based on petri nets. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(4):802–817, 2008. 2

[22] MengChu Zhou and Frank DiCesare. Parallel and sequential mutual exclusions for petri net modeling of manufacturing systems with shared resources. *Robotics and Automation, IEEE Transactions on*, 7(4):515–527, 1991. 2

[23] MD Zisman. Representation, specification and automation of office procedures. *University of Pennsylvania (January 1977)*, 1977. 2