

Composition Colored Petri Nets for the Refinement of Reaction-based Models

Diana-Elena Gratie¹ Cristian Gratie²

*Computational Biomodeling Laboratory, Turku Centre for Computer Science
Department of Computer Science, Åbo Akademi University
Turku, Finland*

Abstract

Model refinement is an important step in the model building process. For reaction-based models, data refinement consists in replacing one species with several of its variants in the refined model. We discuss in this paper the implementation of data refinement with Petri nets such that the size of the model (in terms of number of places and transitions) does not increase. We capture the compositional structure of species by introducing a new class of Petri nets, *composition* Petri nets (ComP-nets), and their colored counterpart, *colored composition* Petri nets (ComCP-nets). Given a reaction-based model with known compositional structure, represented as a ComP-net, we propose an algorithm for building a ComCP-net which implements the data refinement of the model and has the same network structure as the initial ComP-net.

Keywords: Composition Petri nets, composition colored Petri nets, compositional structure, reaction-based model, data refinement.

1 Introduction

Models represent abstractions of real systems, that capture some of the most important behavioral properties of the system. A biological system can be abstracted to a set of biochemical reactions, based on a system-level understanding of the interactions among species. The dynamics is captured in the kinetic rate constants of the reactions. One of the heaviest computational activities for dynamical models is parameter estimation. Usually in the model building process one starts with an abstraction of the system, which is subsequently refined in a stepwise manner so as to include more details. This refinement can be done in a quantitatively correct way, ensuring that at each step the model fit is preserved. Several approaches have

¹ Email: dgratie@abo.fi

² Email: cgratie@abo.fi

been discussed in the literature for reusing previously computed parameters, in order to obtain a more detailed model while avoiding (at least initially) the parameter estimation step for all newly introduced parameters, see [9,2,7,17]. We will consider throughout this paper the concept of data refinement as defined in [5].

Colored Petri nets have been introduced as a programmable high-level class of Petri nets that combines the modeling capabilities of Petri nets with the capabilities of a programming language. They allow the use of data types and parametrization, via the use of color sets (data types) and variables, see [11]. Colors can be used to describe a system in a more compact form, e.g. by representing two identical subsystems with different actors as a single subsystem, where each element has been assigned a color set with two colors (one color for each subsystem). They can also be used to implement refinements of systems via altering the color sets, a process called *type refinement*, see [15]. A method for implementing structural refinements of models using type refinements of colored Petri nets has been proposed in [6]. We use the framework of Petri nets for representing models, and we extend it with a passive part to encode the composition of elements acting in the modeled system. We implement structural refinements of models in our extended framework via type refinements. Our approach to refinement is thus different from the transition refinement discussed in [21] or the transition/place stepwise refinement discussed in [20].

In this paper we focus on qualitative Petri nets, as the goal is to introduce a new class of Petri nets suitable for automatable structural refinement of models. Particularities regarding the continuous and stochastic approaches are beyond the scope of this paper. We consider as a starting point the standard Petri nets, and not colored Petri nets (although a recent book on Petri nets, [18], defines markings as multisets of several types of tokens – a definition similar to that of colored Petri nets) because any colored Petri net can be unfolded to a corresponding equivalent standard Petri net.

The paper is organized as follows: in Section 2 we detail the concept of structural model refinement, as discussed in [4,5]. In Section 3 we introduce the concept of Petri nets with a compositional part, which can capture not only the functioning of a model, but also the compositional relationships between its elements. We first introduce composition Petri nets (ComP-nets), and then give a method of coloring a given ComP-net into a colored composition Petri net (ComCP-net). We discuss next in Section 4 how to implement reaction-based models as Com(C)P-nets. We detail the implementation of model refinement using ComCP-nets in Section 5, and we draw some conclusions in Section 6.

2 Model refinement

We give in this section a formal definition of reaction-based models with known composition of their species. We then introduce the data refinement of such models, in the spirit of [4], but with an explicit distinction between *atomic* and *refined* species, as first presented in [5].

Intuitively, a reaction-based model consists of a set of reactions, usually represented as rewriting rules over a given set of species. For example, consider the following chemical reaction:



We distinguish in this paper between *atomic* species, which – as far as the considered model is concerned – cannot be divided into constituent parts, and *complex* species, which consist of several atomic species. We rely on multisets for encoding the linear combinations of species on either side of a reaction, as well as for denoting the composition of complex species. We denote multiset addition by $++$, and repetitive multiset addition by $^{++}\sum$.

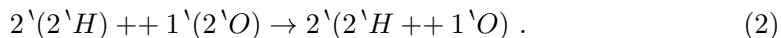
Definition 2.1 [1] Let $S = \{s_1, s_2, \dots\}$ be a set of elements. A *multiset* over S is a function $\sigma : S \rightarrow \mathbb{N}$, which maps each element s of S into a non-negative integer $\sigma(s)$ called the *multiplicity* (or *number of instances*) of s in σ . The multiset σ can also be written as:

$$\sigma = ^{++}\sum_{s \in S} \sigma(s) \setminus s = \sigma(s_1) \setminus s_1 ++ \sigma(s_2) \setminus s_2 ++ \dots ,$$

where the zero coefficient terms can be omitted on the right hand side.

Whenever $\sigma(s) > 0$ we say that σ *contains* s and we write this as $s \in \sigma$. Furthermore, for any two multisets σ, τ over S , we define their *sum* as the multiset $\sigma ++ \tau : S \rightarrow \mathbb{N}$ satisfying $(\sigma ++ \tau)(s) = \sigma(s) + \tau(s)$, for any $s \in S$. The set of all multisets over S will be denoted by S_{MS} .

We now go back to our example reaction (1) and formalize it as a reaction-based model with known composition of its species. We consider that the atomic species are the hydrogen and oxygen atoms and write this as $\Gamma = \{H, O\}$. The complex species are the hydrogen, oxygen and water molecules and are encoded as multisets over Γ to yield the set of complex species $\Delta = \{2 \setminus H, 2 \setminus O, 2 \setminus H ++ 1 \setminus O\}$. The given reaction then becomes a pair of multisets corresponding to the respective stoichiometric coefficients:



Note that atomicity is relative to the considered model, more precisely it depends on the chosen level of detail. For example, in a larger model where the focus is on macromolecules rather than atoms, the complex species from the previous reaction might be considered to be atomic.

We formalize in the following definition the intuition presented above for reaction-based models.

Definition 2.2 A reaction-based model with known composition of its species is a tuple $M = (\Gamma, \Delta, R)$, where:

- Γ is a set of *atomic species*.

- $\Delta \subseteq \Gamma_{MS}$ is a set of *complex species*, defined as multisets over the set of atomic species Γ , where the intuition is that any complex species $\sigma \in \Delta$ contains at least two instances of atomic species, i.e. $\sum_{A \in \Gamma} \sigma(A) \geq 2$.
- $R \subseteq (\Gamma \cup \Delta)_{MS} \times (\Gamma \cup \Delta)_{MS}$ is a set of *reactions* written as $\alpha \rightarrow \beta$ or, alternatively, (α, β) , where $\alpha, \beta \in (\Gamma \cup \Delta)_{MS}$ are multisets encoding the stoichiometric coefficients corresponding to the left- and right-hand sides of the reaction, respectively.

The goal of refinement is to introduce details into the model, in the form of distinguishing several subspecies or variants of a given species. The distinction between the subspecies is very often drawn by post-translational modifications such as acetylation, phosphorylation, etc., by cell differentiation, but it could also account for different possible types of a particular trait (e.g. fur color of animals in a breeding experiment). This type of refinement is called *data refinement*, because it focuses on refining the species (data) of the model.

Definition 2.3 Let Γ and Γ' be two sets of atomic species. A function $\rho : \Gamma \rightarrow 2^{\Gamma'}$ is called an *atomic refinement function* if the following conditions hold:

- (a) $\rho(A) \neq \emptyset$, for all $A \in \Gamma$;
- (b) $\rho(A_1) \cap \rho(A_2) = \emptyset$, for all $A_1, A_2 \in \Gamma$ with $A_1 \neq A_2$;
- (c) $\bigcup_{A \in \Gamma} \rho(A) = \Gamma'$.

A species A' is called an *atomic ρ -refinement* of species A if $A' \in \rho(A)$.

Note that the definition of atomic refinement is equivalent to the definition given in [4] for the species refinement relation, with the distinction that ρ is a function rather than a relation. Moreover, in this paper we also consider the composition of species and, in this context, the atomic refinement will *propagate* throughout the model and induce the refinement of all complex species and, subsequently, the refinement of reactions, following a similar intuition to that presented in [4]. Note that in this paper we prefer a formulation based on multisets rather than vectors, since the former are more common in the literature of Petri nets.

Definition 2.4 Let Γ and Γ' be two sets of atomic species and $\rho : \Gamma \rightarrow 2^{\Gamma'}$ an atomic refinement function.

- (i) A complex species $\sigma' \in \Gamma'_{MS}$ is a *ρ -refinement* of a complex species $\sigma \in \Gamma_{MS}$, written as $\sigma' \in \rho(\sigma)$, if the multiplicity of any species $A \in \Gamma$ in σ equals the sum of the multiplicities of all its ρ -refinements $A' \in \rho(A)$ in σ' , i.e.

$$\rho(\sigma) = \left\{ \sigma' \in \Gamma'_{MS} \mid \sum_{A' \in \rho(A)} \sigma'(A') = \sigma(A), \text{ for all } A \in \Gamma \right\}.$$

Given a set of complex species $\Delta \subseteq \Gamma_{MS}$, we will use $\rho(\Delta)$ to refer to the set of all ρ -refinements of complex species from Δ , i.e. $\rho(\Delta) = \bigcup_{\sigma \in \Delta} \rho(\sigma)$.

- (ii) Let $\Delta \subseteq \Gamma_{MS}$ be a set of complex species. A multiset of species $\alpha' \in (\Gamma' \cup \rho(\Delta))_{MS}$ is a *ρ -refinement* of a multiset $\alpha \in (\Gamma \cup \Delta)_{MS}$, written as $\alpha' \in \rho(\alpha)$,

if the multiplicity of any species $S \in \Gamma \cup \Delta$ in α is equal to the sum of the multiplicities of all its ρ -refinements $S' \in \rho(S)$ in α' , i.e.

$$\rho(\alpha) = \left\{ \alpha' \in (\Gamma \cup \rho(\Delta))_{MS} \mid \sum_{S' \in \rho(S)} \alpha'(S') = \alpha(S), \text{ for all } S \in \Gamma \cup \Delta \right\}.$$

- (iii) A reaction $\alpha' \rightarrow \beta'$ is a ρ -refinement of a reaction $\alpha \rightarrow \beta$ if $\alpha' \in \rho(\alpha)$ and $\beta' \in \rho(\beta)$, i.e.

$$\rho((\alpha, \beta)) = \rho(\alpha) \times \rho(\beta).$$

- (iv) Let $M = (\Gamma, \Delta, R)$ and $M' = (\Gamma', \Delta', R')$ be two reaction-based models with known composition of their species and $\rho : \Gamma \rightarrow 2^{\Gamma'}$ an atomic refinement function. We say that M' is a *structural ρ -refinement* of M if $\Delta' = \rho(\Delta)$ and $R' \subseteq \bigcup_{r \in R} \rho(r)$. If we have equality in the latter relation, we say that M' is the *full structural ρ -refinement* of M .

While the definition of atomic refinement seems to imply that all atomic species are to be refined, the refinement of an atomic species A is *nontrivial* only as long as $|\rho(A)| \geq 2$, i.e. A has at least two distinct variants in the refined model. In this context, whenever $|\rho(A)| = 1$ we will say that A undergoes a *trivial* atomic refinement (which translates to a renaming of A in the refined model).

3 Petri Nets with a Compositional Part

In this section we introduce a new class of Petri nets, *composition Petri nets*. Such nets have two parts: an *active* part, that behaves as a standard Petri net, and a *passive* part, with transitions whose role is to describe how places in the network relate to one another, i.e. how elements in some places are composed of elements in other places.

We assume the reader is familiar with the concept of Petri nets, but we recall some of the definitions and notations to make the paper self-contained. For an introduction, we refer to [19]; for more recent definitions, concepts, extensions and applications to biology we refer to [18,3,13,14].

Definition 3.1 [11] A Petri net is a tuple $N = (P, T, A, E, I)$ where P and T are disjoint sets of *places* and *transitions*, respectively; $A \subseteq P \times T \cup T \times P$ is the set of *arcs*; $E : A \rightarrow \mathbb{N}_+$ is an *arc expression* function (also called weight function); and $I : P \rightarrow \mathbb{N}$ is an *initialization* function, assigning to each place a nonnegative integer that represents the number of tokens in that place.

For a transition t , the set of its *pre-places* (places p such that there exists an arc from p to t) is denoted by $\bullet t$; the set of its *post-places* (places p such that there exists an arc from t to p) is denoted by $t \bullet$. An arc from a place p to a transition t is denoted by a pair $(p, t) \in A$, and an arc from a transition t to a place p is denoted by the pair $(t, p) \in A$.

3.1 Composition Petri nets (ComP-nets)

In this subsection we extend the definition of standard Petri nets with a *compositional part*. We do this by adding a set of *non-fireable composition transitions* and arcs connecting them with the places of the network. Their combined semantics represents the structural composition of the elements represented as places. Thus, a Petri net model describing the dynamics of a system can also include as a subnetwork the composition of the systems' entities (species). This is introduced formally in the following definition.

Definition 3.2 A *composition Petri net (ComP-net)* is a tuple $N = (P, T_c, T, A_c, A, E, I)$ with the following components:

- (i) P, T, A, I represent the set of places, set of transitions, set of arcs and the initialization function of places, respectively, as for standard Petri nets.
- (ii) T_c is a finite set of *composition transitions* such that $P \cap T_c = \emptyset$ and $T \cap T_c = \emptyset$. These transitions are used for depicting the compositional structure of places with respect to other places. Composition transitions never fire, irrespective of the marking of the network, and are also called *passive (non-active) transitions*. The regular transitions are, in contrast, called *active*.
- (iii) $A_c \subseteq P \times T_c \cup T_c \times P$ is a set of *composition arcs* such that:
 - for any place $p \in P$, there is *at most one* incoming composition arc; if there is no composition arc pointing to a place, then that place is considered *atomic*;
 - for every composition transition $t_c \in T_c$ there is *at least one* incoming composition arc connecting a place to it, and *exactly one* outgoing composition arc connecting t_c to a place;
 - the graph induced by the composition arcs and the places and transitions they connect is acyclic.
- (iv) $E : A \cup A_c \rightarrow \mathbb{N}_+$ is an *arc expression* function, such that:
 - the arc expression of a composition arc (t_c, p) (where $t_c \in T_c$ and $p \in P$) is always 1;
 - the arc expression of a composition arc from a place $p \in {}^\bullet t_c$ to a composition transition t_c has the meaning that the post-place of t_c contains $E((p, t_c))$ copies of p ;
 - the arc expression of regular arcs has the usual meaning.

We say that $(P, T_c, A_c, E|_{A_c})$ is the *compositional part* of the network, and $(P, T, A, E|_A, I)$ is the *active part* of the network. Here, for a given set S , $E|_S$ denotes the restriction of the arc expression function E to arcs in S .

For a ComP-net, the properties of standard Petri nets (e.g. boundedness, liveness, deadlock, conflict, invariants, reachability graph) can be generalized, and they will refer only to the *active part of the network*.

The advantage of ComP-nets is that they can explicitly represent both the dynamics of a system and the composition relationships between its elements (places). Note that there may exist pairs of transitions (t_c, t) where $t_c \in T_c$ and $t \in T$ such

that $\bullet t_c = \bullet t$ and $t_c^\bullet = t^\bullet$, i.e. t and t_c have the same pre-places and the same post-places. This can happen because the semantics of such transitions are different. Note also that a ComP-net may contain places that do not take part in any active transition, but which are still compositionally important, and thus must appear in the place set for the compositional structure. Moreover, the fact that a place can have at most one incoming composition arc means that its compositional structure (if any) is unique.

Example 3.3 Consider a model M consisting of atomic species $\Gamma = \{A, B, C, D\}$, complex species $\Delta = \{P, Q, R, S\}$, and a single reversible reaction $P + Q \rightleftharpoons R + S$. Assume that the composition of the complex species is given by:

$$\begin{aligned} P &= 1 \setminus A ++ 1 \setminus B, \\ Q &= 1 \setminus C ++ 1 \setminus D, \\ R &= 1 \setminus A ++ 1 \setminus D, \\ S &= 1 \setminus B ++ 1 \setminus C. \end{aligned}$$

This model can be represented as a Petri net as shown in Figure 1a, where the atomic species are isolated places. The same model can be represented as a ComP-net, as shown in Figure 1b. From the figure it becomes clear what is the composition of species P, Q, R, S , namely that they are complexes $A:B, C:D, A:D$, and $B:C$, respectively. Moreover, from the network structure the reader can get an intuition on how atomic species are interchanged between complex species via active transitions (e.g. one molecule of A from P and one molecule of D from Q bind to form one R).

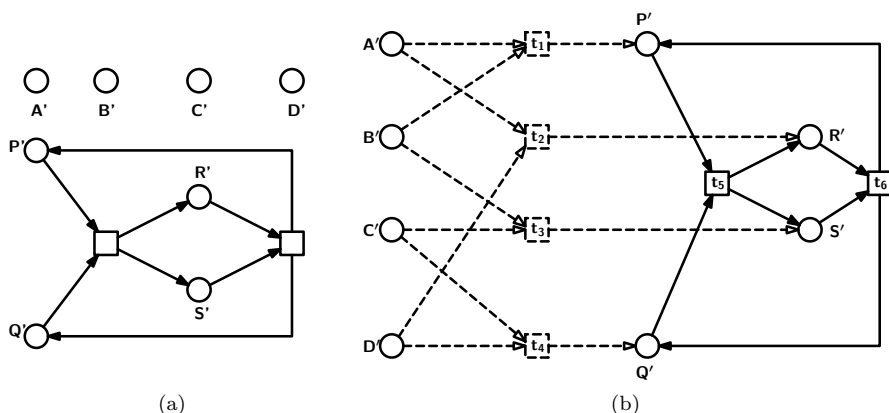


Fig. 1. The representation of a reversible reaction $P + Q \rightleftharpoons R + S$ as: a) a standard Petri net; b) a ComP-net. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are arcs connected at one end to a passive transition. Figure generated using Snoopy [8].

3.2 Composition Colored Petri nets (ComCP-nets)

Sometimes, due to the complexity of a network, it becomes unfeasible or very difficult to read the corresponding Petri net. In such cases, an extension of standard Petri nets, colored Petri nets, might help reduce the size of the model. A complete description of colored Petri nets, their properties and applications can be found in [11,12,13]. Here, we consider the definitions in [13]. We extend colored Petri nets to include a compositional part, in a similar manner as we extended standard Petri nets in the previous subsection.

Notations. We use $|S|$ to denote the cardinality of a set or multiset S . For an arc expression, $|E(a)|$ denotes the cardinality of the expression. $i \setminus S$ where i is a nonnegative integer denotes i copies of S , where S can be a species, a color or a color set. If S is a color set, then $i \setminus S$ is the set of all possible ways of choosing i (not necessarily distinct) colors from the color set S , see Table 1 for an example. In the definition of colored Petri nets, the following standard notations are used: $EXPR_V$ denotes the set of valid (under the used inscription language) expressions using variables from the typed variable set V ; $Type[e]$ denotes the type of an expression $e \in EXPR$, or that of a variable e . We recall further some notions and notations we will use in this paper. The *variables of a transition* t are the set of free variables that appear in t 's guard and in the arc expressions of arcs connected to t . This set is denoted by $Var(t) \subseteq V$ [13]. A *binding* of a transition t is a function b mapping each variable $v \in Var(t)$ into a value $b(v) \in Type[v]$. $B(t)$ denotes the set of all bindings for transition t [13]. A pair (t, b) with $t \in T$ and $b \in B(t)$ is called a *binding element* in [13], and a *transition instance* in [16]. We use here the terminology from [16]: $t(b)$ denotes the instance of transition t with binding b ; $\mathcal{I}_T(t)$ denotes the set of all transition instances of transition t , and $\mathcal{I}_T = \bigcup_{t \in T} \mathcal{I}_T(t)$ denotes the set of all transition instances for all transitions in T . A *place instance* is a pair (p, c) with $p \in P$ and $c \in C(p)$; $\mathcal{I}_P(p)$ denotes the set of all place instances of p , and $\mathcal{I}_P = \bigcup_{p \in P} \mathcal{I}_P(p)$ denotes all place instances of all places in P [16].

Definition 3.4 [13] A *colored Petri Net (CP-net)* is a tuple $N = (P, T, A, \Sigma, V, C, G, E, I)$ satisfying the requirements below:

- (i) P is a finite set of *places*.
- (ii) T is a finite set of *transitions* such that $P \cap T = \emptyset$.
- (iii) $A \subseteq P \times T \cup T \times P$ is a finite set of *arcs*.
- (iv) Σ is a finite set of non-empty types, called *color sets*.
- (v) V is a finite set of *typed variables*, where $Type[v] \in \Sigma$, for all v in V .
- (vi) $C : P \rightarrow \Sigma$ is a *color set* function. It assigns a color set to each place.
- (vii) G is a *guard* function that defines conditions for transitions. It is defined from T into expressions over the variables set V , i.e. $EXPR_V$, such that $Type[G(t)] = Bool$, for all transitions t in T .

- (viii) $E : A \rightarrow \text{EXPR}_V$ is an *arc expression* function such that $\text{Type}[E(a)] = C(p(a))_{MS}$, for all arcs $a \in A$, where $p(a)$ is the place corresponding to arc a , and $C(p(a))_{MS}$ is a multiset of elements with color set $C(p(a))$.
- (ix) I is an *initialization* function that assigns to each place p an initialization expression such that $\text{Type}[I(p)] = C(p)_{MS}$.

We want to use ComCP-nets as a means to easily model and implement the structural refinement of a system, as described in Section 2. For this, the key ingredient is the choice of color sets, especially for the complex places. The color sets should reflect the composition of places and, moreover, do it in such a way that the process of assigning color sets to complex places can be done automatically. Thus, for atomic places we propose the use of simple color sets, e.g. `int` or `Enumeration`. For complex places, the corresponding color set contains all possible multisets over the color sets of its constituent atomic places, with multiplicities dictated by the actual composition of the place. We provide in what follows a coloring example using this strategy.

Example 3.5 Let P be a molecule with two possible states, and let P_2 and P_3 denote its dimer and trimer, respectively. We will use P , P_2 and P_3 to denote both the actual molecules and the places representing them in a Petri net. We list in Table 1 a possible definition of color sets for the three entities.

Species	Color set	Colors
P	$\text{CS}_P = \text{enum with } a, b$	$\{a, b\}$
P_2	$\text{CS}_{P_2} = \text{bag } 2 \setminus \text{CS}_P$	$\{2 \setminus a, 1 \setminus a ++ 1 \setminus b, 2 \setminus b\}$
P_3	$\text{CS}_{P_2} = \text{bag } 3 \setminus \text{CS}_P$	$\{3 \setminus a, 2 \setminus a ++ 1 \setminus b, 1 \setminus a ++ 2 \setminus b, 3 \setminus b\}$

Table 1
Coloring strategy for the dimer and trimer of a molecule

Definition 3.6 A *composition colored Petri net (ComCP-net)* is a tuple $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ that satisfies the following requirements:

- (i) P, T_c, T, A_c, A satisfy the constraints of Definition 3.2.
- (ii) Σ, V, I have the usual meaning, namely the set of color sets, the set of variables, and the initialization function, respectively.
- (iii) $C : P \rightarrow \Sigma$ is the *color function* assigning color sets to places such that:
 - all atomic places have disjoint color sets, and
 - for all complex places $p \in P$, $C(p) = {}^{++}\sum_{p' \in \bullet t_c} |E(p', t_c)| \setminus C(p')$, where t_c stands for the composition transition encoding the composition of p , i.e. $t_c^\bullet = \{p\}$, ;
- (iv) $G : T_c \cup T \rightarrow \text{EXPR}_V$ is the *guard function*, such that for each composition transition $t_c \in T_c$ with $t_c^\bullet = \{p\}$ there exists exactly one binding for which the guard is true for each color in $C(p)$.

- (v) $E : A \cup A_c \rightarrow \text{EXPR}_V$ is the *arc expression function*, defined such that for every composition transition $t_c \in T_c$ with $t_c^\bullet = \{p\}$: $E(t_c, p) = {}^{++}\sum_{p' \in \bullet t_c} E(p', t_c)$.

We say that $(P, T_c, A_c, \Sigma, V, C, G|_{T_c}, E|_{A_c})$ is the *compositional part* of the network, and $(P, T, A, \Sigma, V, C, G|_T, E|_A, I)$ is the *active part* of the network.

For a ComCP-net, the properties of colored Petri nets (e.g. boundedness, liveness, deadlock, conflict, invariants, reachability graph) can be generalized, and they will refer only to the *active part of the network*. We extend the notion of transition instance (binding element) to cover the composition and the active transitions of a ComCP-net respectively. We denote by \mathcal{I}_T the set of active transition instances, i.e. pairs (active transition, binding of variables): $\mathcal{I}_T = \{(t, b) \mid t \in T, b \in B(t)\}$ where $B(t)$ is the set of all possible bindings for a transition t . We use $\mathcal{I}_T\langle \text{true} \rangle$ to denote the set of active transition instances whose guard evaluates to *true*: $\mathcal{I}_T\langle \text{true} \rangle = \{(t, b) \in \mathcal{I}_T \mid G(t)\langle b \rangle = \text{true}\}$. Similarly, we use \mathcal{I}_{T_c} and $\mathcal{I}_{T_c}\langle \text{true} \rangle$ for the composition transition counterparts of these sets.

Example 3.7 Let us consider the example net in Figure 2a. This is not a ComCP-net because of several violations of the ComCP-net definition. The network suggests that E, S, P are atomic places, and ES is a complex place. But the atomic places E and S have the same color set, while the definition requires that atomic places are assigned disjoint color sets (to allow for the identification of colors that come from different places in the color of a complex place). Moreover, the color set of ES is independent of the color sets of E and S, which compose it, and the arc expressions of the composition arcs do not capture any kind of composition.

4 Implementing models as composition (colored) Petri nets

In this section we introduce a way of modeling with composition (colored) Petri nets. We consider as input models of the form $M = (\Gamma, \Delta, R)$ as discussed in Section 2. Every species in M is represented by a place, and each reaction is represented by a transition. The compositional structure of each complex species is represented as the *compositional part* of the composition Petri net model. We give examples for both ComP-nets and ComCP-nets.

4.1 Implementing models as ComP-nets

Definition 4.1 Let $M = (\Gamma, \Delta, R)$ be a reaction-based model with known composition of its species. We say that a ComP-net $N = (P, T_c, T, A_c, A, E, I)$ *structurally implements* the reaction-based model M if there are a bijection $f_P : P \rightarrow \Gamma \cup \Delta$ between places in P and species in $\Gamma \cup \Delta$, a bijection $f_T : T \rightarrow R$ between transitions in T and reactions in R , and a bijection between composition transitions and complex species $f_c : T_c \rightarrow \Delta$ such that:

- (i) for every place $p \in P$ and every composition transition $t \in T_c$ the following conditions regarding the composition transitions hold:

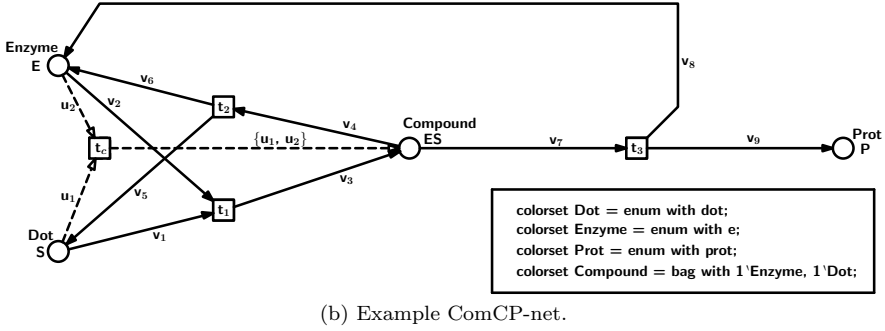
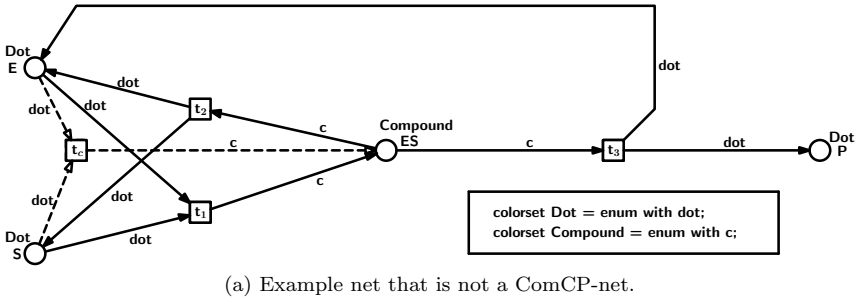


Fig. 2. Example of (a) an ill-defined and (b) a properly defined ComCP-net. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are composition arcs. The text on top of arcs is the arc expression. Places are labeled with a name and their corresponding color set, and the color set definition is given in the inset. Arc expressions in (a) are values, and in (b) are typed variables with the type given by the color set of the place connected to the arc. Figure generated using Snoopy [8].

- $(p, t) \in A_c \Leftrightarrow f_c(t)(f_P(p)) \geq 1$ and, moreover, $E(p, t) = f_c(t)(f_P(p))$,
- $(t, p) \in A_c \Leftrightarrow f_c(t) = f_P(p)$ and, moreover, $E(t, p) = 1$;

(ii) for every place $p \in P$ and every transition $t \in T$ with $f_T(t) = \alpha \rightarrow \beta$ the following conditions hold:

- $(p, t) \in A \Leftrightarrow \alpha(f_P(p)) \geq 1$ and, moreover, $E(p, t) = \alpha(f_P(p))$,
- $(t, p) \in A \Leftrightarrow \beta(f_P(p)) \geq 1$ and, moreover, $E(t, p) = \beta(f_P(p))$.

We call the ComP-net N the (f_P, f_T, f_c) -implementation of model M .

Example 4.2 The ComP-net N represented in Figure 1b is an implementation of the model $M = \{\{A, B, C, D\}, \{P, Q, R, S\}, \{P + Q \rightleftharpoons R + S\}\}$ presented in Example 3.3. There exist bijections f_P , f_T and f_c that satisfy the conditions in Definition 4.1. We provide the definitions of these functions in what follows. The place to species function f_P is defined as $f_P(X') = X$, where X' is a place of N and X is the species with the same name that it represents, i.e. place A' represents species A of model M and so on. The composition transition to complex species function is defined as $f_c(t_1) = P$, $f_c(t_2) = R$, $f_c(t_3) = S$, $f_c(t_4) = Q$. It is easy to notice that the requirements for f_c are fulfilled. The transition to reaction function is defined as $f_T(t_5) = P + Q \rightarrow R + S$, $f_T(t_6) = R + S \rightarrow P + Q$. Again, the conditions on existence of arcs and their expressions are fulfilled.

4.2 Implementing models as ComCP-nets

In the colored setting, there are several aspects that one has to be very careful about. For example, consider a reaction that uses multiple instances of some species, e.g. $2A + B \rightarrow C$. If the color set of the place representing A contains more than one color, then the arc expression of the arc connecting the place that denotes A with the transition that encodes the mentioned reaction should contain variables. Moreover, the transition should have a guard so that it would not allow for two bindings that evaluate to the same multiset of colors. This can easily be implemented by considering an ordering of the elements of each color set, and a guard that tests that the values that the variables on adjacent arcs evaluate to are ordered, with a non-strict ordering. So the guards should be of the form $[!(v_i < v_j)], \forall i > j$, or equivalently $(v_i \geq v_j)$, for all i, j such that $i > j$.

Definition 4.3 Let $M = (\Gamma, \Delta, R)$ be a reaction-based model with known composition of its species. We say that a ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ *structurally implements* the reaction-based model M if there are a bijection $f_P : \mathcal{I}_P \rightarrow \Gamma \cup \Delta$ mapping place instances $(p, c) \in \mathcal{I}_P$ to species in $\Gamma \cup \Delta$, a bijection $f_T : \mathcal{I}_T\langle true \rangle \rightarrow R$ mapping active transition instances to reactions in R , and a bijection $f_c : \mathcal{I}_{T_c}\langle true \rangle \rightarrow \Delta$ mapping composition transition instances to complex species such that:

- (i) for every composition transition instance $(t_c, b) \in \mathcal{I}_{T_c}\langle true \rangle$ and every place instance $(p, c) \in \mathcal{I}_P$ such that c appears in the binding b the following conditions hold:
 - $(p, t_c) \in A_c \Leftrightarrow f_c(t_c, b)(f_P(p, c)) \geq 1$ and, moreover, the corresponding arc expression satisfies $E(p, t_c)\langle b \rangle(c) = f_c(t_c, b)(f_P(p, c))$;
 - $(t_c, p) \in A_c \Leftrightarrow f_c(t_c, b) = f_P(p, c)$ and, moreover, $E(t_c, p)\langle b \rangle = 1 \setminus c$;
- (ii) for every active transition instance $(t, b) \in \mathcal{I}_T\langle true \rangle$ with $f_T(t, b) = \alpha \rightarrow \beta$ and every place instance $(p, c) \in \mathcal{I}_P$ such that c appears in the binding b the following conditions hold:
 - $(p, t) \in A \Leftrightarrow \alpha(f_P(p, c)) \geq 1$ and, moreover, $E(p, t)\langle b \rangle(c) = \alpha(f_P(p, c))$;
 - $(t, p) \in A \Leftrightarrow \beta(f_P(p, c)) \geq 1$ and, moreover, $E(t, p)\langle b \rangle(c) = \alpha(f_P(p, c))$.

We call the ComCP-net N the (f_P, f_T, f_c) -*implementation* of model M .

Note that there are multiple ways of representing a model M as a ComCP-net, depending on the color sets one chooses, and on the bijections f_P , f_T and f_c . Note also that, because of the bijectivity of the functions characterizing the model implementation, for every active transition it holds that each of its instances with true guards stands for a reaction in the model; moreover, a place can encode more than one species if and only if all species that it encodes take part in similar reactions, in all possible combinations. One could also formulate the previous definition to say that a ComCP-net implements a model if its unfolding implements that model.

Example 4.4 Consider a model $M = (\Gamma, \Delta, R)$ with $\Gamma = \{E, S, P\}$, $\Delta = \{ES\}$

such that $ES = 1'E ++ 1'S$, and R containing the reactions:

$$\begin{aligned} r_1 &: E + S \rightarrow ES; \\ r_2 &: ES \rightarrow E + S; \\ r_3 &: ES \rightarrow E + P. \end{aligned}$$

M is a model for an enzymatic reaction, and we show next that the ComCP-net N represented in Figure 2b implements it. The place instances of N are

$$\mathcal{I}_P = \{(E, e), (S, \text{dot}), (P, \text{prot}), (ES, 1'e ++ 1'\text{dot})\}.$$

The possible active transition instances are:

$$\begin{aligned} \mathcal{I}_T = \{ & (t_1, \langle v_1 = \text{dot}, v_2 = e, v_3 = 1'e ++ 1'\text{dot} \rangle), \\ & (t_2, \langle v_4 = 1'e ++ 1'\text{dot}, v_5 = \text{dot}, v_6 = e \rangle), \\ & (t_3, \langle v_7 = 1'e ++ 1'\text{dot}, v_8 = e, v_9 = \text{prot} \rangle) \}. \end{aligned}$$

The only passive transition instance is $\mathcal{I}_{T_c} = \{(t_c, \langle u_1 = \text{dot}, u_2 = e \rangle)\}$.

There exist bijections f_P , f_T and f_c that satisfy the conditions in Definition 4.3. We detail here the definition of these functions.

The place to species function f_P is defined as $f_P(X, \text{col}(X)) = X$, where X is a place of N , $\text{col}(X)$ is its color (note that every color set has only one color), and X is the species with the same name that it represents.

The composition transition to complex species function is defined as

$$f_c(t_c, \langle u_1 = \text{dot}, u_2 = e \rangle) = ES.$$

It is easy to notice that the requirements for f_c are fulfilled.

The transition to reaction function is defined as:

$$\begin{aligned} f_T(t_1, \langle v_1 = \text{dot}, v_2 = e, v_3 = 1'e ++ 1'\text{dot} \rangle) &= r_1; \\ f_T(t_2, \langle v_4 = 1'\text{dot} ++ 1'e, v_5 = \text{dot}, v_6 = e \rangle) &= r_2; \\ f_T(t_3, \langle v_7 = 1'\text{dot} ++ 1'e, v_8 = e, v_9 = \text{prot} \rangle) &= r_3. \end{aligned}$$

Again, the conditions on existence of arcs and their expressions are fulfilled.

Example 4.5 Consider a model M consisting of atomic species $\Gamma = \{A, B\}$, complex species $\Delta = \{C, D, E\}$, with

$$\begin{aligned} C &= 2'A; \\ D &= 1'A ++ 1'B; \\ E &= 2'B. \end{aligned}$$

and the set of reactions $R = \{2A \rightarrow C, A + B \rightarrow D, 2B \rightarrow E, C + E \rightarrow 2D\}$. This model can be implemented with the ComCP-net from Figure 3.

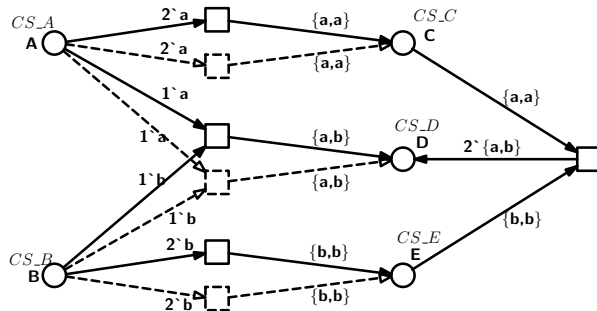


Fig. 3. The representation of an example model as a ComCP-net. The model consists of reactions $\{2A \rightarrow C, A + B \rightarrow D, 2B \rightarrow E, C + E \rightarrow 2D\}$. Circles represent places; solid squares represent active transitions; dashed squares represent passive transitions; solid arrows represent arcs; dashed arrows are arcs connected at one end to a passive transition. The name of the color set of a place is the italic text next to a place. The text on top of arcs is the arc expression. All color sets have only one color, which appears on the arc expressions. Figure generated using Snoopy [8].

4.3 From ComP-nets to ComCP-nets

In the following we give an algorithm for coloring a ComP-net to get a corresponding ComCP-net, Algorithm 1. We call the resulting ComCP-net the *natural coloring* of the given ComP-net. The ComCP-net in Figure 2b is an example of a natural coloring. Every place corresponding to an atomic species gets as color set an enumeration color set with only one element, and every place corresponding to a complex species gets as color set the set of possible multiset of all its compositional constituents' colors, each as many times as dictated by compositional arcs. The arc expressions in the built ComCP-net use a variable for each token traversing the arc, so that we don't restrict the natural coloring. All guards are set to *true*, as all color sets have exactly one color and thus there cannot exist several bindings that evaluate to the same multiset of colored tokens.

Algorithm 1 ComP_to_ComCP**Input:** a ComP-net $N = (P, T_c, T, A_c, A, E, I)$;**Output:** a ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma, V, C, G, E', I')$;

```

1: procedure ASSIGN_CS( $p$ )                                 $\triangleright$  assign the color set of place  $p$ 
2:    $T_p \leftarrow \bullet p \cap T_c$ ;
3:   if  $T_p = \emptyset$  then                                 $\triangleright p$  is an atomic place
4:      $C(p) \leftarrow$  new distinct color set with one element;
5:     return
6:   end if
7:    $t_c \leftarrow$  the one value in  $T_p$ ;
8:   for all  $q \in \bullet t_c$  do
9:     if  $C(q) = \text{NIL}$  then ASSIGN_CS( $q$ );
10:    end if
11:  end for
12:   $CS_p \leftarrow \sum_{q \in \bullet t_c}^{++} E(q, t_c) \cdot C(q)$ ;
13:   $\Sigma \leftarrow \Sigma \cup CS_p$ ;
14:   $C(p) \leftarrow CS_p$ ;
15: end procedure
16:
17:  $\Sigma \leftarrow \emptyset$ ;
18:  $V \leftarrow \emptyset$ ;
19:
20: for all  $p \in P$  do
21:    $C(p) \leftarrow \text{NIL}$ ;
22: end for
23: for all  $p \in P$  do                                        $\triangleright$  assign color sets
24:   ASSIGN_CS( $p$ );
25: end for
26:
27: for all  $a \in A \cup A_c$  do                                 $\triangleright$  assign arc expressions
28:    $p \leftarrow$  the place connected with arc  $a$ ;
29:    $V' \leftarrow \emptyset$                                     $\triangleright V'$  stores the variables used in the arc expression of  $a$ 
30:   for all  $i \leftarrow 1$  to  $E(a)$  do
31:     create a variable  $v_{a,i} : C(p)$ ;
32:      $V' \leftarrow V' \cup \{v_{a,i}\}$ ;
33:   end for
34:    $E'(a) \leftarrow \sum_{v \in V'}^{++} v$ ;
35:    $V \leftarrow V \cup V'$ ;
36: end for

```

Algorithm 1 (continued) ComP.to.ComCP

```

37: for all  $t \in T_c \cup T$  do                                 $\triangleright$  transition guards are all set to true
38:    $G(t) \leftarrow true$ ;
39: end for
40:
41: for all  $p \in P$  do                                        $\triangleright$  assign initial markings
42:    $I'(p) \leftarrow I(p) \setminus C(p)[0]$ ;                 $\triangleright$  the one color in the color set  $C(p)$ 
43: end for
44:
45:  $N' = (P, T_c, T, A_c, A, \Sigma, V, C, G, E', I')$ ;
return  $N'$ ;

```

There are of course multiple ways of coloring the given network N . We chose here different color sets for each atomic element, so that each such element can be identified by its color set. Furthermore, we chose a representation of complex elements based on multisets, as this allows for the implementation of the refinement of a network without changing the structure of the network's implementation.

In the algorithm we assume that the network is well-defined and composition places are post-places of exactly one composition transition. The sets of places, composition transitions, transitions, composition arcs and arcs are the same in the initial and final networks.

The set of color sets contains, for places with no incoming composition arc, a color set with one color, and for places p with an incoming composition arc (t_c, p) , a color set that is the set of all multisets of colors from the color sets of the pre-places of t_c , as many times as the value of the arc expression of the arc from the pre-place to t_c .

Arc expressions use a distinct variable for each colored token. We do this in order to not restrict the natural coloring, and allow for further extensions of it.

Transition guards are all set to *true*; no ordering is needed because each color set has only one element.

It is not difficult to see that, if the input of the algorithm is a ComP-net that is a (f_P, f_T, f_c) -implementation of a reaction network $M = (\Gamma, \Delta, R)$, then the output is a ComCP-net that structurally implements M .

5 Implementing Data Refinement with ComCP-nets

Colored Petri nets can be used to implement refinements of a model in a compact way, as discussed in [15,5,6]. We present here an algorithmic method for implementing the structural refinement of a model using its ComCP-net representation. Our approach differs from that of [6] via the automatic propagation of refinement from one atomic place to all places connected to it by compositional transitions. We consider the *type refinement* of colored Petri nets, namely a refinement of the color sets of some of the places in the network, see [15]. A morphism between two colored Petri nets captures a type refinement if it induces no change in the structure

of the network, and the colors in the resulting network are consistently subtyped. Namely, the refinement adds details to the color sets, such that the resulting color sets can be projected onto the initial color sets.

Definition 5.1 We say that a ComCP-net CP' is a *type refinement* of a ComCP-net CP if the compositional parts of the two networks are isomorphic and there exists a type refinement morphism between their active parts.

For a given model $M = (\Gamma, \Delta, R)$ represented as a ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ using the natural coloring, assume that one of the atomic species, $S \in \Gamma$, is to be refined (i.e. replaced throughout the model with several of its variants). Let γ be the number of such variants that S can be replaced with. We build a ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma', V, C', G', E', I')$ to be the *type (color) refinement* of N . The sequence of steps required to implement the refinement is presented in Algorithm 2 and briefly explained here.

Reflecting the change for place q in the ComCP-net N' is done by adding more colors to the color set of q . This can be done by either adding an attribute with γ possible values, or by altering the enumeration color set such that instead of one element it has γ elements (colors). From the definition of the ComCP-net N as the natural coloring of the model M it follows that all color sets of places corresponding to complex species containing the refined species will automatically reflect the refinement (as they contain the refined color set $C(q)$). For species that are not refined, the corresponding places get as initial marking $I'(p) = I(p)$. For the refined species, there are multiple ways of choosing the initial marking for each of the newly introduced subspecies. The condition they must obey is $|I'(p)| = |I(p)|$.

The chosen method of implementing the refinement conserves the structure of the network and is thus the most compact with respect to the initial network. Moreover, based on the compositional part of the network, a simulation software that would support composition Petri nets could automatically generate the color sets for complex species based on the color sets of atomic species that are input by the modeler. This would give a significant speedup in the refinement process.

For the construction detailed above, the size of the model is the same as that of the initial model in terms of number of places and transitions. However, the increase in model size is encapsulated in the number of colors used in each color set, and the possible binding elements for each transition.

Theorem 5.2 Let $M = (\Gamma, \Delta, R)$ and $M' = (\Gamma', \Delta', R')$ be two reaction-based models with known composition of their species, and $\rho : \Gamma \rightarrow 2^{\Gamma'}$ an atomic refinement function, such that M' is the full ρ -refinement of M . Let $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$ be a naturally colored ComCP-net that is a (f_P, f_T, f_c) -implementation of M (e.g. the natural coloring of a ComP net that implements M). Then the ComCP-net $N' = (P', T'_c, T', A'_c, A', \Sigma', V', C', G', E', I')$, obtained by repeatedly running Algorithm 2 to compute a color refinement of each atomic place, structurally implements the refined model M' .

Proof. We only consider the refinement of a single atomic species and prove the

claim of the theorem for this case. The result can then be easily extended for the repeated application of Algorithm 2. Let $A \in \Gamma$ be the atomic species that is refined, γ the number of variants it is refined to, and let p_A denote the place that stands for A in N .

The ComCP-net N has exactly one place for each species from model M . The set of places is the same for N' , but the place instances mirror the refined model M' as follows: for each atomic place $p \in P$ except for p_A , there is only one place instance, $(p, C(p))$; for p_A there are γ instances, one for each color in the refined color set. The places corresponding to complex species have as color sets a multiset containing the color sets of the constituent places and are thus automatically updated to reflect the refinement of p_A . Each place instance will thus correspond to a refined complex species in M' .

The set of arcs is the same in the two networks, and the set of arc expressions differs only in the typing of variables.

Each active transition that is not connected to p_A or to a place that is connected to p_A via a composition transition has its guard set to true, and only one possible binding. Such transitions correspond to those reactions that refine to a singleton set in M' . Transitions connected to p_A or to a place that is connected to p_A via a composition transition have a guard that allows exactly one binding for each possible multiset of tokens. A binding (t, b) of such a transition will thus encode precisely the refinements $\rho(f_T(t, b))$ of $f_T(t, b)$.

□

Example 5.3 For the Example 4.4, let us consider a refinement where the enzyme E can be in two different conformations, E^1 and E^2 , both of which can catalyze the production of P . Moreover, consider that the environment can induce the transformation of one conformation into the other, but this is not explicitly modeled in the system. In order to reflect this change, the complex species ES refines to

Algorithm 2 ComCP_refinement

Input: a naturally colored ComCP-net $N = (P, T_c, T, A_c, A, \Sigma, V, C, G, E, I)$; an atomic place $q \in P$ to be refined, and γ , the number of colors that q 's color set refines to.

Output: the corresponding color refinement ComCP-net $N' = (P, T_c, T, A_c, A, \Sigma', V', C', G', E', I')$;

- 1: $C' \leftarrow C$; ▷ start with the color function of N
 - 2: $\Sigma' \leftarrow \Sigma \setminus \{C(q)\}$; ▷ color sets from N , except the color set of place q
 - 3: $CS \leftarrow \text{enumeration with } \gamma \text{ elements}$;
 - 4: $C'(q) \leftarrow CS$; ▷ modify the color set of the place to be refined
 - 5: $\Sigma' \leftarrow \Sigma' \cup \{CS\}$; ▷ add q 's new color set to the set of color sets
 - 6:
 - 7: $E' \leftarrow E$;
 - 8: **for all** $\{a \in A \cup A_c \mid a = (q, t) \text{ OR } a = (t, q)\}$ **do**
 - 9: $V_a \leftarrow$ all variables appearing in $E(a)$;
-

Algorithm 2 (continued) ComCP_refinement

```

10:   $V' \leftarrow V' \setminus V_a$ ;
11:   $V'_a \leftarrow \emptyset$ ;
12:  for all  $v_{a,i} \in V_a$  do                                 $\triangleright$  re-type the variables of arcs connected to  $q$ 
13:      define  $v'_{a,i} : C'(q)$ ;
14:       $V'_a \leftarrow V'_a \cup \{v'_{a,i}\}$ ;
15:  end for
16:   $E'(a) \leftarrow {}^{++}\sum_{v \in V'_a} v$ ;
17:   $V' \leftarrow V' \cup V'_a$ ;
18: end for
19:
20:  $P_q \leftarrow \{q\}$ ;                                 $\triangleright$  set of places affected by the color refinement
21: for all  $t \in T_c$  do
22:     if  $(q, t) \in A_c$  then
23:          $P_q = P_q \cup t^\bullet$ ;
24:     end if
25: end for
26:
27: for all  $t \in T \cup T_c$  do                                 $\triangleright$  change guards where needed
28:     for all  $p \in t^\bullet \cup {}^\bullet t$  do
29:         if  $p \in P_q$  then
30:              $G(t) \leftarrow$  new guard such that no two bindings evaluate to the same
             multiset of tokens;
31:             break;
32:         end if
33:     end for
34: end for
35: for all  $p \in P$  do     $\triangleright$  change the initial marking for places affected by the color
    refinement
36:     if  $p \in P_q$  then
37:          $I'(p) \leftarrow$  assign initial marking;
38:     else
39:          $I'(p) \leftarrow I(p)$ ;
40:     end if
41: end for
42:
43:  $N' = (P, T_c, T, A_c, A, \Sigma', V', C', G', E', I')$ ;
return  $N'$ ;

```

$E^1S = 1 \setminus E^1 {}^{++} 1 \setminus S$ and $E^2S = 1 \setminus E^2 {}^{++} 1 \setminus S$. The new set of reactions is listed in Table 2. In order to implement the refinement for the ComCP-net in Figure 2b, we only change the color set of place E to be **Enzyme** = enum e1, e2. The mapping from place instances to species is straightforward, and the mapping from transition instances to reactions is captured in Table 2.

Initial reaction	Refined reaction	Transition instance
$E + S \rightarrow ES$	$E^1 + S \rightarrow E^1 S$	$(t_1, \langle v_1 = \text{dot}, v_2 = e1, v_3 = 1'e1 ++ 1'dot \rangle)$
	$E^1 + S \rightarrow E^2 S$	$(t_1, \langle v_1 = \text{dot}, v_2 = e1, v_3 = 1'e1 ++ 1'dot \rangle)$
	$E^2 + S \rightarrow E^1 S$	$(t_1, \langle v_1 = \text{dot}, v_2 = e2, v_3 = 1'e1 ++ 1'dot \rangle)$
	$E^2 + S \rightarrow E^2 S$	$(t_1, \langle v_1 = \text{dot}, v_2 = e2, v_3 = 1'e1 ++ 1'dot \rangle)$
$ES \rightarrow E + S$	$E^1 S \rightarrow E^1 + S$	$(t_2, \langle v_4 = 1'e1 ++ 1'dot, v_5 = \text{dot}, v_6 = e1, \rangle)$
	$E^1 S \rightarrow E^2 + S$	$(t_2, \langle v_4 = 1'e1 ++ 1'dot, v_5 = \text{dot}, v_6 = e2, \rangle)$
	$E^2 S \rightarrow E^1 + S$	$(t_2, \langle v_4 = 1'e2 ++ 1'dot, v_5 = \text{dot}, v_6 = e1, \rangle)$
	$E^2 S \rightarrow E^2 + S$	$(t_2, \langle v_4 = 1'e2 ++ 1'dot, v_5 = \text{dot}, v_6 = e2, \rangle)$
$ES \rightarrow E + P$	$E^1 S \rightarrow E^1 + P$	$(t_3, \langle v_7 = 1'e1 ++ 1'dot, v_8 = e1, v_9 = \text{prot}, \rangle)$
	$E^1 S \rightarrow E^2 + P$	$(t_3, \langle v_7 = 1'e1 ++ 1'dot, v_8 = e2, v_9 = \text{prot}, \rangle)$
	$E^2 S \rightarrow E^1 + P$	$(t_3, \langle v_7 = 1'e2 ++ 1'dot, v_8 = e1, v_9 = \text{prot}, \rangle)$
	$E^2 S \rightarrow E^2 + P$	$(t_3, \langle v_7 = 1'e2 ++ 1'dot, v_8 = e2, v_9 = \text{prot}, \rangle)$

Table 2

Full structural refinement of an enzymatic model to consider two variants of an enzyme.

If a refinement where transitions from one conformation of the enzyme to the other is preferred to the full structural refinement (e.g. $E^1 + S \rightarrow E^2 S$ is not a valid reaction), the ComCP-net can be further restricted with guards by not allowing certain bindings (e.g. $[(v_1 = \text{dot} \ \& \ v_2 = e1 \ \& \ v_3 = 1'e1 ++ 1'dot)]$).

6 Conclusions

We introduced in this paper a new class of Petri nets that has capabilities for fast model refinement, when the compositional structure of the elements is known. Such Petri nets have a passive compositional part and an active part. The passive part encodes the compositional structure of the elements (species, encoded as places in the network), and all transitions in this part never fire. The active part encodes the behavior of the model.

Model refinement in some formalisms (e.g. ODE models) requires explicitly writing all possible combinations of reactions induced by replacing some species with several of their variants. With colored Petri nets, this can be done without changing the structure of the network. Internally, all these combinations are generated when binding the variables on arcs to values. Moreover, considering the compositional structure of species and choosing the colors in the manner we propose means that all species containing some atomic species that needs to be refined are automatically refined at once.

There exist also modeling frameworks that allow for a compact characterization

of models and are good at handling model explosion upon refinement of a model. For example the Kappa language, see [2,7,17], allows compactness via explicitly mentioning an attribute only when its value is important, and omitting it whenever the actual value is not important, with the understanding that a reaction happens regardless of the value of that particular attribute. Refinement could resume then to adding attributes to a species, as presented e.g. for a case study of the heat shock response in [10]. The framework we are proposing allows for a similar approach of modeling, where attributes and internal states of species can be represented in the color set. Variables can be used whenever the explicit value of some attribute is not important, and actual colors should be used in arc expressions and guards when the particular value of an attribute is important. ComCP-nets have the advantage of being graphical and adding structural information in a formalized manner to the widely used framework of Petri nets.

Some of the combinations of species that are generated when refining some species may be biologically impossible. The formalism of Petri nets is suitable for dealing with such reactions by adding guards to the “parent” reaction (the reaction that was replaced with some biologically impossible reactions). Also, if additional information is known about the model, e.g. on the way atomic species are transferred from some complex species to other complex species, it can easily be implemented by manipulating arc expressions. This will be in the scope of a future paper.

Acknowledgement

The authors gratefully acknowledge support from Academy of Finland through project 267915.

References

- [1] Calude, C. S., P. Gheorghe, G. Rozenberg and A. Salomaa, editors, “Multiset Processing,” Lecture Notes in Computer Science **2235**, Springer-Verlag Berlin Heidelberg, 2001.
- [2] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Rule-based modelling and model perturbation*, in: C. Priami, R.-J. Back and I. Petre, editors, *Transactions on Computational Systems Biology XI*, Lecture Notes in Computer Science **5750**, Springer Berlin Heidelberg, 2009 pp. 116–137.
- [3] David, R. and H. Alla, “Discrete, continuous, and hybrid Petri nets,” Springer Berlin Heidelberg, 2010.
- [4] Gratie, C. and I. Petre, *Fit-preserving data refinement of mass-action reaction networks*, in: A. Beckmann, E. Csuhaj-Varjú and K. Meer, editors, *Language, Life, Limits*, Lecture Notes in Computer Science **8493**, Springer International Publishing, 2014 pp. 204–213.
- [5] Gratie, D.-E., B. Iancu, S. Azimi and I. Petre, *Quantitative model refinement in four different frameworks, with applications to the heat shock response*, Technical Report 1067, TUCS (2013).
- [6] Gratie, D.-E. and I. Petre, *Full structural model refinement as type refinement of colored Petri nets*, in: M. Heiner and A. K. Wagler, editors, *Proceedings of the 6th International Workshop on Biological Processes and Petri Nets*, CEUR Workshop Proceedings **1373** (2015), pp. 70–84.
- [7] Harmer, R., *Rule-based modelling and tunable resolution*, EPTCS **9** (2009), pp. 65–72.
- [8] Heiner, M., M. Herajy, F. Liu, C. Rohr and M. Schwarick, *Snoopy - a unifying Petri net tool*, in: S. Haddad and L. Pomello, editors, *Application and Theory of Petri Nets*, Lecture Notes in Computer Science **7347**, Springer Berlin Heidelberg, 2012 pp. 398–407.

- [9] Iancu, B., E. Czeizler, E. Czeizler and I. Petre, *Quantitative refinement of reaction models*, International Journal of Unconventional Computing **8** (2012), pp. 529–550.
- [10] Iancu, B., D.-E. Gratie, S. Azimi and I. Petre, *On the implementation of quantitative model refinement*, in: A.-H. Dediu, C. Martn-Vide and B. Truthe, editors, *Algorithms for Computational Biology*, Lecture Notes in Computer Science **8542**, Springer International Publishing, 2014 pp. 95–106.
URL http://dx.doi.org/10.1007/978-3-319-07953-0_8
- [11] Jensen, K., “Coloured Petri Nets,” EATCS Monographs on Theoretical Computer Science, Springer-Verlag Berlin Heidelberg, 1992.
- [12] Jensen, K., “Coloured Petri Nets,” Monographs in Theoretical Computer Science. An EATCS Series, Springer-Verlag Berlin Heidelberg, 1995.
- [13] Jensen, K. and L. M. Kristensen, “Coloured Petri Nets,” Springer-Verlag Berlin Heidelberg, 2009.
- [14] Koch, I., W. Reisig and F. Schreiber, editors, “Modeling in systems biology: the Petri net approach,” Computational Biology **16**, Springer-Verlag London, 2011.
- [15] Lakos, C., *Composing abstractions of coloured Petri nets*, in: *Application and Theory of Petri Nets 2000*, Springer, 2000 pp. 323–342.
- [16] Liu, F., “Colored Petri nets for systems biology,” Ph.D. thesis, Universitätsbibliothek (2012).
- [17] Murphy, E., V. Danos, J. Feret, J. Krivine and R. Harmer, *Rule-based modeling and model refinement*, Elements of Computational Systems Biology (2009), pp. 83–114.
- [18] Reisig, W., “Understanding Petri Nets,” Springer-Verlag Berlin Heidelberg, 2013.
- [19] Reisig, W. and G. Rozenberg, editors, “Lectures on Petri Nets I: Basic Models,” Lecture Notes in Computer Science **1491**, Springer-Verlag Berlin Heidelberg, 1998.
- [20] Suzuki, I. and T. Murata, *A method for stepwise refinement and abstraction of Petri nets*, Journal of Computer and System Sciences **27** (1983), pp. 51 – 76.
- [21] Valette, R., *Analysis of Petri nets by stepwise refinements*, Journal of Computer and System Sciences **18** (1979), pp. 35 – 46.