A Class of Composable High Level Petri Nets

Eike Best[1], Hans Fleischhack[2], Wojciech Frączak[3], Richard P. Hopkins[4], Hanna Klaudel[3] and Elisabeth Pelz[5]

Abstract

In this paper a high-level Petri net model called M-nets (for multilabeled nets) is developed. A distinctive feature of this model is that it allows not only vertical unfolding, as do most other high-level net models, but also horizontal composition - in particular, synchronisation - in a manner similar to process algebras such as CCS. This turns the set of M-nets into a domain whose composition operations satisfy various algebraic properties. The operations are shown to be consistent with unfolding in the sense that the unfolding of a composite high-level net is the composition of the unfoldings of its components. A companion paper shows how this algebra can be used to define the semantics of a concurrent programming language compositionally.

1 Introduction and Motivation

In traditional high-level Petri net models, as described for instance in [1, 10, 11, 16, 17, 20, 21], there are place/transition *annotations* which determine the transition rule of the model. Such annotations also drive the 'vertical' unfolding of a high-level net into an elementary net [10].

In this paper, we define an extended high-level net model called the *M-net model* which allows the specification of 'horizontal' annotations as well as vertical ones. Horizontal annotations, which will be called *labels*, allow the modular construction of large high-level nets from smaller components. Thus, every element (place or transition) of an M-net carries an *inscription* which is a pair of the following form:

$$\textit{inscription} \ = \ (\,\textit{label} \mid \textit{annotation}\,).$$

Using the labels of an M-net, we define a series of operations most of which are counterparts of operations known from programming languages, such as sequential composition and iteration. The most novel operation we define is *transition synchronisation* which allows transitions of a high-level net to be merged in a systematic way. This operation generalises the synchronisation operation found in process algebras such as CCS [13] in a way that serves a useful purpose for block structured languages [5].

In this paper, we demonstrate that labels (for composition) and annotations (for unfolding) can be defined in harmony with each other. More precisely, we prove that they satisfy not just a set of algebraic laws but also a property of coherence with respect to each other. Informally, this property states that the unfolding of a composite net equals the composition of the unfoldings of its components; that is, if, say, U denotes unfolding and $\odot$ denotes some binary operation on M-nets N_1, N_2 then $U(N_1 \odot N_2) = U(N_1) \odot U(N_2)$. It will be shown that in order to satisfy this property, the synchronisation operation needs rather a careful definition involving simple unification, logical conjunction and multiset addition of appropriate expressions, formulae and (multi)sets. Existing high-level net models that provide some form of compositionality understand modularity either in terms of place and/or transition fusion (e.g. [8, 15]) or in terms of synchronisation expressions (e.g. [6]). However, we know of no other approach that would turn the class of high-level nets into the kind of algebra we need, such that vertical and horizontal operations are interrelated in the coherent way we describe in this paper.

[1]Institut für Informatik, Universität Hildesheim, Marienburger Platz 22, D-31141 Hildesheim, Germany; e-mail: E.Best@informatik.uni-hildesheim.de

[2]Institut für Informatik, Universität Oldenburg; e-mail: Hans.Fleischhack@arbi.informatik.uni-oldenburg.de

[3]Université Paris Sud, L.R.I. bât. 490, 91405 Orsay Cedex, France; e-mail: {fraczak,klaudel}@lri.fr

[4]Department of Computing Science, University of Newcastle upon Tyne, UK; e-mail: R.P.Hopkins@newcastle.ac.uk

[5]UPS, L.R.I. bât. 490, F-91405 Orsay, also UPVM, Equipe d'Informatique Fondamentale, F-94010 Créteil; pelz@lri.fr

The paper is structured as follows. In section 2, we introduce basic notions needed to define the inscriptions of the model. In section 3 we introduce the M-net model and describe two basic rules associated with it, namely the transition rule and the unfolding rule, which are both quite standard. Section 4 introduces the M-net composition operations we wish to consider. They are governed by the labels rather than by the annotations of the participating M-net(s). Section 5 contains the coherence theorems mentioned earlier. Section 6 contains some concluding remarks.

The choice of M-net operations, defined in Section 4, is somewhat unusual. This choice is motivated by the background of this research. One of its origins has been the compositional elementary Petri net semantics of $B(PN)^2$, the concurrent language defined in [5]. This translation has some desirable properties; in particular, it is compositional and yields a net with a minimal number of transitions. Also, for concurrent algorithms with a comprehensible number of states, it allows the application of fast Petri net based analysis techniques such as described in [7, 19] or Esparza's model checking algorithm [9] (in fact, many such techniques have been implemented in a tool, PEP [3], which supports the elementary net semantics of $B(PN)^2$). Unfortunately, it usually yields very large nets, as a necessary consequence of the fact that elementary nets were used.

Thus, even though analysis methods are rarer for high-level nets, reasons of machine representability and human readability made it desirable to give the semantics of $B(PN)^2$ by high-level nets. It is desirable that this is done in a way which is compositional and consistent with the low-level semantics (modulo unfolding). Since the compositional elementary semantics of $B(PN)^2$ is given in terms of the Petri Box Calculus PBC [2], the same set of operations also needs to be defined for the high-level algebra. For this reason, the operations defined in Section 4 mimic the PBC operations. The actual translation from $B(PN)^2$ into the M-net algebra is not given in the present paper but in its companion paper [4]. That paper also contains the proof of the consistency between the high-level and the low-level semantics of $B(PN)^2$ defined in [5]. Even though this consistency result makes significant use of the coherence results of the present paper, the two papers can be read and understood in isolation.

2 Basic Notions

This section defines the basic concepts which are important for the inscriptions of transitions, places and arcs of an M-net.

2.1 Values, types, variables, terms, and substitutions

Let VAL be a fixed but suitably large set of *values* with at least the distinguished element $\bullet$. This set contains all the 'colours' of the high level net model we define. A *type* is a subset of VAL. The element $\bullet$ is contained in VAL for consistency with the place/transition net model; all places of P/T nets have the associated type $\{\bullet\}$, and the markings on such places are multisets of tokens $\bullet$. For examples which follow, we will assume that integers and Boolean values **true** and **false** are in VAL. The letters u, v, w almost towards the end of the alphabet will be used to denote fixed values in the set VAL. We will also need symbols to denote possibly changing values, i.e., variables. We will normally use the small letters a, b, c near the front of the alphabet to denote such variables. Let the set VAR, then, be defined as the set of (symbols for) variables $\{a, b, c, \ldots\}$. The type of a variable is the range of values it may take; if it is not explicitly given then it is implicitly the whole set VAL.

We will use standard definitions of substitutions and associated notions to describe the substitution of variables by values or, more generally, by terms. Well-formed *terms* τ, such as $5+7$ and $a \vee \mathbf{true}$, are built from values, variables and a set of operators OP inductively in a standard way. For the sake of brevity we do not include such a definition. We will assume that type errors do not occur, i.e., terms such as $a \vee \mathbf{true}$ when the type of a is not a subset of $\{\mathbf{false}, \mathbf{true}\}$, will not be considered[6]. Terms τ without variables are called *ground terms*. Every value is also a ground term. Conversely, every ground term without type errors can be evaluated and yields a value. Thus, of the above examples, only $5+7$ is a ground term; its evaluation yields 12. The set of terms will be abbreviated as $\mathbf{VT}$ (for value terms). A *substitution* θ is a function θ: $VAR \to \mathbf{VT}$ which

[6] Alternatively, one could include a special error element in the set VAL. However, type errors can be treated in various standard ways which are orthogonal to the subject of this paper, hence we ignore them.

associates a term to each variable in VAR. A substitution θ can be written (non-uniquely) in the form of a list

$$(a_1/\tau_1, \ldots, a_m/\tau_m),$$

where the a_j are mutually distinct variables, meaning that the variable a_j is substituted by the term τ_j $(1 \leq j \leq m)$. For instance, if $VAR = \{a\}$ then $\theta_0 = (a/\mathbf{false})$ is a substitution. A substitution θ can be applied to a term τ written as $\tau[\theta]$ or more simply $\tau\theta$, yielding a possibly different term when all variables a in τ are replaced by $\theta(a)$. For example, $(5+7)[\theta_0] = (5+7)$ and $(a \vee \mathbf{true})[\theta_0] = (\mathbf{false} \vee \mathbf{true})$. A substitution can also be applied to sets or multisets or vectors of terms, by applying it element-wise to every element of the set or multiset or vector, respectively. For example,

$$\{(5+7), (a \vee \mathbf{true})\}[\theta_0] = \{(5+7), (\mathbf{false} \vee \mathbf{true})\}.$$

We distinguish two special kinds of substitutions. Firstly, a *binding* (of $\{a_1, \ldots, a_m\}$) is a substitution $(a_1/\tau_1, \ldots, a_m/\tau_m)$ such that every τ_j is a ground term. We also say that a_j is *bound* to the ground term τ_j. If a substitution θ binds every variable of a term τ to a ground term then the term $\tau[\theta]$ is itself a ground term and can be evaluated to yield a value. For instance, $(5+7)[\theta_0]$ (still) yields 12, $(a \vee \mathbf{true})[\theta_0]$ yields **true**. The second kind of substitution is a *renaming*, that is a substitution $(a_1/\tau_1 \ldots a_m/\tau_m)$ such that the τ_j are mutually distinct variables. We will denote bindings by σ and renamings by ρ. Substitutions can be composed with each other: $\tau[\theta_1\theta_2] = (\tau[\theta_1])[\theta_2]$, and similarly for sets, multisets and vectors of terms. A term τ *matches* a term τ' if there exists a substitution θ such that $\tau'[\theta] = \tau$; in that case we also say that τ' is *more general* than τ. This relation is reflexive and transitive (i.e., a quasi-ordering) on the set $\mathbf{VT}$ of terms. The associated equivalence relation will be called *renaming equivalence*[7]. The name comes from the following lemma:

Lemma 2.1 *Let τ and τ' be two terms. Then the following are equivalent:*

(i) *τ and τ' are renaming equivalent.*

(ii) *There is a renaming ρ such that $\tau'[\rho] = \tau$.*

(iii) *There is a renaming ρ' such that $\tau[\rho'] = \tau'$.* ■

Both relations can canonically be extended to sets, multisets and vectors of terms as well. A substitution θ will be called *at least as general* as a substitution θ' if there exists a substitution θ_0 such that $\theta' = \theta\theta_0$. A substitution θ is called a *unifier* of two terms τ_1 and τ_2 if $\tau_1[\theta] = \tau_2[\theta]$. A *most general unifier* is a unifier θ_0 such that every unifier θ of τ_1 and τ_2 equals $\theta_0\theta'$ for some substitution θ'. Again, this notion can be extended canonically to sets, multisets and vectors of terms.

2.2 Action symbols and action terms

We will now define a framework which generalises in a particular way the setup found in Milner's basic CCS [13, 14]. We assume the existence of a fixed but sufficiently large set $\mathbf{A}$ of *action symbols* (sometimes also called *actions*, for short). We assume that each action symbol A from $\mathbf{A}$ has an arity $ar(A)$, i.e., by definition, a natural number, which, intuitively, describes the number of atomic values linked with it. The set $\mathbf{A}$ is, by definition, the carrier of a bijection $\overline{}\colon \mathbf{A} \to \mathbf{A}$ called *conjugation* which satisfies

$$\forall A \in \mathbf{A}\colon\ \overline{A} \neq A \text{ and } \overline{\overline{A}} = A.$$

That is, the mapping $\overline{}$ groups the elements of $\mathbf{A}$ into mutual pairwise conjugates. It is required that $ar(A) = ar(\overline{A})$ for each action symbol $A \in \mathbf{A}$, i.e., that the arities of conjugate actions are equal. We will always use capital letters to denote action symbols. An *action term* is, by definition, a construct

$$A(\tau_1, \ldots, \tau_{ar(A)}),$$

[7] Sometimes it is called *literal similarity*.

where A is an action symbol and τ_j $(1 \leq j \leq ar(A))$ are value terms as in section 2.1. Later we shall be interested only in action terms where the τ_j are distinct variables or (not necessarily distinct) values. Such action terms will be called *simple*. For instance, for $A, B, C \in \mathbf{A}$ and $ar(A) = 2$, $ar(B) = 1$, $ar(C) = 0$, the following are simple action terms: $A(a, b)$, $A(a, 7)$, $A(5, 7)$, $B(a)$, $B(7)$, and C. The action term $A(a, b + 5)$ is, by contrast, not simple. If an action term contains no variables of VAR then it will be called a *ground action term* or an *elementary action*. Of the above examples, $A(5, 7)$, $B(7)$ and C are ground action terms while the others are not. We use the symbol $\mathbf{AT}$ to denote simple action terms. Let σ be some binding which associates a ground term to each variable in an action term $A(\tau_1, \dots, \tau_{ar(A)})$. Then $A(\tau_1, \dots, \tau_{ar(A)})$ can be evaluated and yields the ground action term

$$(A(\tau_1, \dots, \tau_{ar(A)}))[\sigma] = A((\tau_1, \dots, \tau_{ar(A)})[\sigma]).$$

For instance, if σ binds a to the value 6 and b to the value 7 then $A(a, b+5)[\sigma]$ is the ground action term $A(6, 12)$. However, as we have said, we will henceforth consider only simple action terms. If all action symbol arities are 0, then the sets $\mathbf{A}$ and $\mathbf{AT}$ coincide, and we obtain the basic CCS action algebra (except for the special symbol τ which we do not need because we will represent silent actions in a different way).

3 M-nets, their Behaviour, and their Unfolding

The principal difference between M-nets and predicate/transition nets [10, 17] is that M-nets carry additional information in their place and transition inscriptions, to support composition operations which are analogous to the PBC operations. In M-nets, besides the usual annotations on places (set of allowed tokens), arcs (set of variables) and transitions (occurrence condition), we have an additional label on places, denoting their status (entry, exit or internal) and an additional label on transitions, denoting their communication interface.

3.1 M-nets and markings of M-nets

Definition 3.1 An *M-net* is a triple (S, T, ι) such that S is a set of *places*, T is a set of *transitions* with $S \cap T = \emptyset$, and ι is an inscription function with domain $S \cup (S \times T) \cup (T \times S) \cup T$ such that:

— For every place $s \in S$, $\iota(s)$ is a pair $(\lambda_s \mid \alpha_s)$, where λ_s is a *(place-)label* and α_s is a *(place-) annotation*. λ_s is an element of the set $\{e, \emptyset, x\}$ and α_s, the type of s, is a subset of VAL.

— For every arc $f \in (S \times T) \cup (T \times S)$, $\iota(f)$ is a finite multiset of variables from VAR.

— For every transition $t \in T$, $\iota(t)$ is a pair $(\lambda_t \mid \alpha_t)$, where λ_t is its *label* and α_t its *annotation*. λ_t is a finite multiset of action terms (section 2.2), and α_t is a finite set of value terms. $\diamond$

We explain the three kinds of inscriptions in turn, beginning with the arc inscriptions. In the explanation, we anticipate the intended transition rule semantics. The arc inscriptions specify the variables by which tokens flow; more formally, when tokens flow along an arc, they become bound to the variables on that arc. Note that an arc inscription may be empty (i.e., the empty multiset). This signifies that no tokens may flow along that arc and that no effective connection exists along it. Arcs with an annotation $\emptyset$ (empty multiset) will not be represented graphically.

Each place s of an M-net is annotated by a pair $(\lambda_s \mid \alpha_s)$ where $\lambda_s \in \{e, \emptyset, x\}$ describes its status (entry, internal or exit, respectively), and α_s is a subset of VAL, which delimits the set of tokens permissible on s. We impose four restrictions for places and place inscriptions:

1. there exist $s, s' \in S$ such that $\lambda_s = e$ and $\lambda_{s'} = x$;
2. if $\lambda_s \in \{e, x\}$ then α_s must be $\{\bullet\}$;
3. if $\iota_s = (e \mid \{\bullet\})$ then $\forall t \in T \colon \iota((t, s)) = \emptyset$;
4. if $\iota_s = (x \mid \{\bullet\})$ then $\forall t \in T \colon \iota((s, t)) = \emptyset$.

The first, third and fourth restrictions ensures that there is always at least one entry place and at least one exit place, entry places may have no incoming arcs and exit places may have no outgoing arcs; the second guarantees that both entry and exit interfaces have the $\{\bullet\}$ type. Both properties together ensure that entry and exit places can be composed without problems. The second restriction implies that the initial marking and also the final marking (if it can be reached) of an M-net must consist of black tokens. These restrictions are important for the results which we want to prove.

The first part λ_t of the inscription of a transition t is a multiset of simple action terms from **AT**. A transition t can occur only under some binding which allows every term in that multiset to be evaluated; then λ_t yields a multiset of elementary actions. The second part, $\alpha_t = \{\tau_1, \ldots, \tau_n\}$, is a set of value terms from **VT**. It will play the rôle of an *occurrence condition* or a *guard*, in the sense that t may occur only if the binding in question yields the evaluation **true** for all of the terms τ_i; thus we will also denote $\{\tau_1, \ldots, \tau_n\}$ as a logical conjunction of its elements (for instance, $\{\tau_1, \tau_2\}$ will be written $\tau_1 \wedge \tau_2$ or $\tau_2 \wedge \tau_1$; by this convention, the empty set will be represented by an empty conjunction, i.e., by **true**). In Figure 1 we present two examples of M-nets in accordance with the formal definition. Given a transition $t \in T$, the immediate environment of t in N will be called the

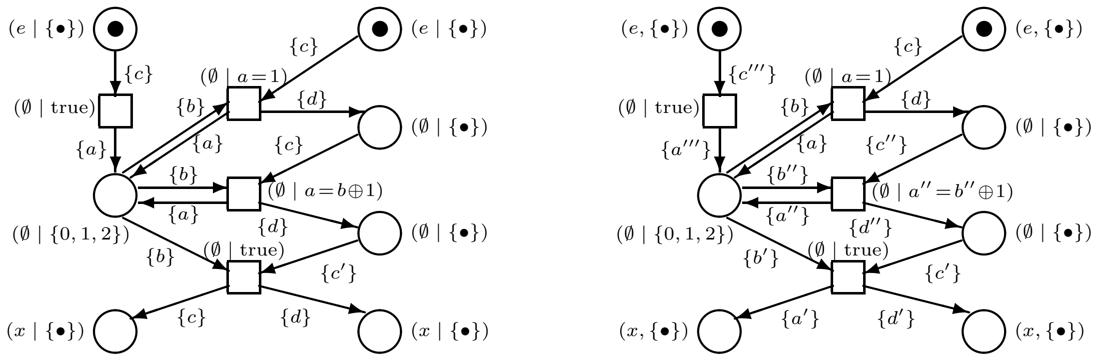

Figure 1: Two M-nets

area of t:

$$area(t) \quad = \quad \{(s,t) \mid s \in S \wedge \iota((s,t)) \neq \emptyset\} \cup \{t\} \cup \{(t,s) \mid s \in S \wedge \iota((t,s)) \neq \emptyset\}.$$

Note that areas of different transitions are always disjoint. The set of all variables occurring in the annotations of an area $area(t)$ will be denoted by $VAR(t)$.

Definition 3.2 Let $N = (S, T, \iota)$ be an M-net. Two transitions $t, t' \in T$ are said to *duplicate* each other (or to be *renaming equivalent*) in N iff their areas are renaming equivalent, more precisely (with a view to Lemma 2.1), iff there exists a renaming ρ such that the following holds:

(i) $\forall s \in S \colon ((s,t') \in area(t') \Leftrightarrow (s,t) \in area(t)) \quad \wedge \quad \iota((s,t'))[\rho] = \iota((s,t))$.

(ii) $\iota(t')[\rho] = \iota(t)$.

(iii) $\forall s \in S \colon ((t',s) \in area(t') \Leftrightarrow (t,s) \in area(t)) \quad \wedge \quad \iota((t',s))[\rho] = \iota((t,s))$. $\diamond$

That is, in words, t and t' are duplicates if they are connected to the rest of the net in the same ways and carry the same labels, up to some renaming ρ which applies uniformly to the entire area of the two transitions. The area of a transition is the domain of relevance of variables occurring in annotations. This means that the name of a variable from the annotation of a transition t or an arc bordering on t is relevant only inside the area of t. We will see that this has two consequences. First, every M-net can be transformed, without changing its semantics, by renaming of variables local to transition areas, into an M-net where variables of different areas are disjoint:

$$\forall t_1, t_2 \in T \colon t_1 \neq t_2 \Rightarrow VAR(t_1) \cap VAR(t_2) = \emptyset.$$

The justification for this claim will be given at the end of section 3.2. The left hand side of Figure 1 shows an M-net which violates this property while the right hand side of the same figure shows an M-net which has this property. Secondly, if two transitions t and t' duplicate each other then one of them is redundant; omitting it will not change the behaviour of the net. The justification of this second claim will be given at the end of sections 3.2 and 3.3.

Definition 3.3 We call an M-net (S, T, ι) *elementary* if and only if

$$\forall s \in S\colon \quad \iota(s) = (\lambda_s \mid \alpha_s) \ \Rightarrow \ \alpha_s \text{ is a singleton set}$$

$$\forall t \in T\colon \quad \iota(t) = (\lambda_t \mid \alpha_t) \ \Rightarrow \ \lambda_t \text{ is a multiset over elementary actions derived from } \mathbf{A} \text{ and } \alpha_t = \mathbf{true}. \quad \diamond$$

Definition 3.4 A *marking* M of an M-net (S, T, ι) is a function $M\colon S \to \mathcal{M}_f(VAL)$, where $\mathcal{M}_f(VAL)$ denotes the set of finite multisets over VAL, satisfying:

$$\forall v \in VAL\colon v \notin \alpha_s \Rightarrow (M(s))(v) = 0. \quad \diamond$$

An equivalent reformulation of this condition is that for every $s \in S$, $M(s)$ is a finite multiset over the type α_s of s. The number $(M(s))(v)$ describes the number of occurrences of the value v in the place s at the marking M. Note that M may be assign s the empty multiset, indicating no marking at all on s.

3.2 The transition rule

In accordance with the usual definition (e.g. [11]), the transition rule specifies the circumstances under which a marking M_2 is reachable from a marking M_1. For the remainder of this section, let $N = (S, T, \iota)$ be a fixed M-net. Let t be a transition of N. A binding σ will be called *legal for* t, if it satisfies the following properties:

— σ binds every variable in $VAR(t)$ to some value;

— $\forall s \in S\colon a \in (\iota((s,t)) \cup \iota((t,s))) \Rightarrow \sigma(a) \in \alpha_s$.

The first requirement means that σ associates values to all variables that occur freely in the area of t. The second requirement means that for all variables that occur on arcs between some place s and t, σ has to respect the type α_s of s.

Let σ be a binding which is legal for t. Then σ associates a value $\sigma(a)$ to every variable a in $VAR(t)$. Since σ can be lifted to vectors and (multi-)sets of terms in the manner specified in sections 2.1 and 2.2, it follows that if t is a transition and τ is a term in α_t, then σ defines - via an evaluation of that term - a value $\tau[\sigma]$ in VAL. Let μ be a multiset of value terms; then $\mu[\sigma]$ defines - via an evaluation of every term in the multiset - a multiset of values in VAL. Similarly, let $A(\tau_1, \ldots, \tau_n)$ be an action term; then σ determines an elementary action $A(\tau_1, \ldots, \tau_n)[\sigma]$. Finally, if μ is a finite multiset of action terms, then $\mu[\sigma]$ is a multiset of elementary actions. It follows that if t is a transition of N and σ is a legal binding of t then inside the area of t we have:

— For every place s and arc (s,t), $\iota((s,t))[\sigma]$ is a well-defined multiset of values.

— For every place s and arc (t,s), $\iota((t,s))[\sigma]$ is a well-defined multiset of values.

— $\lambda_t[\sigma]$ is a well-defined multiset of elementary actions.

— $\alpha_t[\sigma]$ is a well-defined set of values.

Definition 3.5 The *transition relation* is a quaternary relation $\to$, defined with respect to a value $\mathbf{true} \in VAL$, with the following relationality:

$$\to \ \subseteq \ \mathcal{M} \times T \times \mathcal{B} \times \mathcal{M},$$

where $\mathcal{M}$ denotes the set of all markings of N and $\mathcal{B}$ denotes the set of all bindings. A quadruple (M_1, t, σ, M_2) is defined to be in $\to$ if and only if:

(a) σ is a legal binding of t.

(b) $\alpha_t[\sigma] = \mathbf{true}$[8].

[8]I.e., every term τ in α_t satisfies $\tau[\sigma] = \mathbf{true}$. In particular, if α_t includes a term such as $b{+}5$ then the transition t can never occur, no matter what markings or bindings are given.

(c) There is a marking $M \in \mathcal{M}$ such that:

$$\forall s \in S\colon \quad M_1(s) \quad = \quad \iota((s,t))[\sigma] + M(s)$$
$$\forall s \in S\colon \quad M_2(s) \quad = \quad M(s) + \iota((t,s))[\sigma],$$

where + denotes multiset sum (see Appendix A.1). $\diamond$

In part (c) of this definition, the last part of the legality of σ is used, because this is what guarantees that the multiset sum is well-defined. The marking M that must exist by (c) can be understood as the part of the global marking that remains unchanged by the occurrence of t. If (M_1, t, σ, M_2) is in $\rightarrow$ then we will also write

$$M_1 \xrightarrow{t:\sigma} M_2$$

and say that the occurrence of t under the binding σ leads from M_1 to M_2. All the usual basic dynamic concepts (occurrence sequences, set of reachable markings, etc.) follow from the transition rule we have just defined.

In general, if $\rho\colon \mathit{VAR} \rightarrow \mathit{VAR}$ is a renaming bijection for variables and $\sigma\colon V \rightarrow \mathit{VAL}$ is a binding of $V \subseteq \mathit{VAR}$ then $\rho^{-1} \circ \sigma$ is a corresponding binding of $\rho(V)$; conversely, if $\sigma'\colon \rho(V) \rightarrow \mathit{VAL}$ is a binding of $\rho(V)$ then $\rho \circ \sigma'$ is an inversely corresponding binding of V. In particular, by this inverse transformation, if the variables in areas of transitions other than t are renamed, this has no influence on the transition rule for t; every binding that was legal for t before the renaming is also legal after the renaming, and conversely. Similarly, any legal binding for a transition t is also legal (modulo the corresponding renaming) for any transition that is renaming equivalent with t.

The transition rule (and also the unfolding rule below) is hence invariant over renamings, as long as they are consistently done within the area of a transition. For this reason, every M-net can be transformed into one in which the variable sets in different areas are disjoint. Another possible modification is to replace a term τ occurring in an action term $A(\ldots, \tau, \ldots)$ belonging to λ_t by a fresh variable a and adding the equation $a = \tau$ to the multiset α_t. Both the transition rule and the unfolding rule are invariant against this modification.

3.3 The unfolding rule

In this section we specify the unfolding rule, which associates an elementary M-net to every M-net in a canonical way, as well as a marking of the former to every marking of the latter. Quite in accordance with traditional ideas (for instance, [18]) we associate with $N = (S, T, \iota)$ an elementary M-net

$$U(N) = (U(S), U(T), U(\iota))$$

which - in a sense made precise by means of Lemma 3.6 - is a model of N:

- For every place $s \in S$ and every value $v \in \alpha_s$ there is a place s_v in $U(S)$. No other places are in $U(S)$. The inscription $\iota(s_v)$ is defined by $\lambda_{s_v} = \lambda_s$ and $\alpha_{s_v} = \{v\}$.
- For every transition $t \in T$ and every legal binding σ which satisfies both conditions (a) and condition (b) of the transition rule, there is a transition t_σ in $U(T)$. No other transitions are in $U(T)$. The inscription $\iota(t_\sigma)$ is defined by $\lambda_{t_\sigma} = \lambda_t[\sigma]$ and $\alpha_{t_\sigma} = \mathbf{true}$.
- The inscription $(U(\iota))((s_v, t_\sigma))$ of an arc in $U(S) \times U(T)$ is defined for each value variable $a \in \mathit{VAR}$, as:

  $$((U(\iota))((s_v, t_\sigma)))(a) = \sum_{\sigma(a)=v} \iota((s,t)))(a)$$

  An arc (s_v, t_σ) is inscribed by the multiset of variables coming from the inscription of the arc (s,t) and such that each of them evaluates through σ to v.
- The inscription $(U(\iota))((t_\sigma, s_v))$ of an arc in $U(T) \times U(S)$ is defined symmetrically.

It is not hard to see that $U(N)$ is indeed an elementary M-net. Next, let M be any marking of N. We define a marking $U(M)$ as follows: For every place $s_v \in U(S)$, and for each value $v' \in VAL$:

$$((U(M))(s_v))(v') = \sum_{v'=v} (M(s))(v')$$

Thus, each elementary place $s_v \in U(S)$ contains as many values v as the number of times this value occurs in the marking $M(s)$.

Lemma 3.6 *Unfolding and transition rules commute*
With notations as above and M_1, M_2 two markings of N, the following are equivalent:

$$\begin{array}{lll} \text{(i)} & M_1 \xrightarrow{t:\sigma} M_2 & \textit{in } N \\ \text{(ii)} & U(M_1) \xrightarrow{t_\sigma:\sigma} U(M_2) & \textit{in } U(N). \end{array}$$

Proof: First, we prove (i)⇒(ii).
Suppose that conditions (a)-(c) of the transition rule are valid for the quadruple (M_1, t, σ, M_2) in N, it has to be shown that the same conditions are valid for the quadruple $(U(M_1), t_\sigma, \sigma, U(M_2))$ in the unfolded net $U(N)$. Conditions (a) and (b) are clearly valid.
To check condition (c), we use the fact that by condition (c) in N, there exists a suitable marking $M$of N which makes the two equations of (c) hold true (in N). We show that the marking $U(M)$ plays the same rôle in $U(N)$; we check the first equation of condition (c):

$$\forall s_v \in U(S)\colon \underbrace{(U(M_1))(s_v)}_{(A)} = \underbrace{(U(\iota))((s_v, t_\sigma))[\sigma]}_{(B)} + \underbrace{(U(M))(s_v)}_{(C)}.$$

Part (A) is, by the definition of the transformation U, a multiset containing as many values v as the number of times the value v occurs in $M_1(s)$. Similarly, part (C) is a multiset containing as many values v as the number of times the value v occurs in $M(s)$. Part (B) is a multiset containing as many values v as the number of times the value v occurs in $\iota((s,t))[\sigma]$. The whole equation then follows from the first equation of (c) in N, valid, in particular, for the value v. The second part of condition (c) is symmetrical.
(ii)⇒(i): The above calculation can be reversed and done for every value v; this shows that the condition (c) on the level of N follows from its validity at the level of $U(N)$. The other two conditions (a) and (b) are satisfied by the definition of σ. ■
The following lemma follows directly from the observation in the previous section, that the set of legal bindings of two duplicating transitions are the same modulo renaming.

Lemma 3.7 *Renaming equivalence is a congruence w.r.t. unfolding.*
Two duplicating transitions unfold into sets of mutually duplicating transitions. ■

4 Operations on M-nets

The previous section has defined unfolding, which is a vertical operation in the sense that it creates an elementary object (M-net or marking) from a not necessarily elementary object. Unfolding was driven by the annotations, namely the types of places and the multisets of value terms in transition inscriptions. In this section we define composition operations corresponding to the operators of the PBC [2]. They are horizontal in the sense that they create a new, not necessarily elementary, M-net out of one, two, or three given M-nets. The correspondence between vertical and horizontal operations is the subject of the next section.
For the purpose of this section we define ${}^\bullet N$ and $N^\bullet$ as the sets of, respectively, e-places and x-places of an M-net N. We will define the net compositions for choice, sequence, concurrent composition and iteration via some auxiliary net manipulation operators (see Appendix A.2 for their definitions): iN for making a copy of N tagged by the integer i; $\otimes\{S_1, \ldots, S_n\}$ for multiplying n sets of places, $N \oplus S'$ for adding a new set of places S' to N and at the same time removing the old places from which S' has been constructed, and $N_1 \cup N_2$ for the union of N_1 and N_2.
We concentrate in this section on the synchronisation operation which is the main and novel construct. Unlike all preceeding operations, its definition does not refer to any auxiliary net operations from the Appendix A.2, and can be read independently.
In the following, N_1, N_2 and N_3, are all M-nets.

4.1 Sequence

$N_1; N_2 \;=\; (1N_1 \cup 2N_2) \;\oplus\; \otimes\{(1N_1)^\bullet, {}^\bullet(2N_2)\}.$

This should be read as follows. To obtain the net $N_1; N_2$, first create two independent copies $1N_1$ of N_1 and $2N_2$ of N_2. Then multiply ($\otimes$) the x-places of $1N_1$ pairwise with the e-places of $2N_2$. Finally, ($\oplus$) connects those new places into the net like the old ones, gives them the label $\emptyset$ and removes the old ones.

The other constructions below should be read in similar ways, but we will not describe them in detail.

4.2 Choice

$N_1 \,\square\, N_2 \;=\; (1N_1 \cup 2N_2) \;\oplus\; \otimes\{{}^\bullet(1N_1), {}^\bullet(2N_2)\} \;\oplus\; \otimes\{(1N_1)^\bullet, (2N_2)^\bullet\}.$

4.3 Parallel composition

$N_1 \| N_2 \;=\; 1N_1 \cup 2N_2.$

4.4 Iteration

For the iteration construct $[N_1 * N_3 * N_5]$, we use the construction shown in Figure 2. In this construction, 2 is a copy of 1, 4 is a copy of 3 and 6 is a copy of 5:

$$
\begin{aligned}
[N_1 * N_3 * N_5] \;=\;\; & (1N_1 \cup 2N_1 \cup 3N_3 \cup 4N_3 \cup 5N_5 \cup 6N_5) \\
& \oplus\; \otimes\,\{{}^\bullet(1N_1), {}^\bullet(2N_1)\} \\
& \oplus\; \otimes\,\{(1N_1)^\bullet, {}^\bullet(3N_3), (4N_3)^\bullet, {}^\bullet(5N_5)\} \\
& \oplus\; \otimes\,\{(2N_1)^\bullet, {}^\bullet(4N_3), (3N_3)^\bullet, {}^\bullet(6N_5)\} \\
& \oplus\; \otimes\,\{(5N_5)^\bullet, (6N_6)^\bullet\}.
\end{aligned}
$$

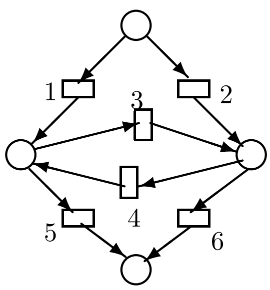

Figure 2: Schematic construction for the iteration

4.5 Synchronisation

The intuitive idea behind the synchronisation operation of an M-net consists of a repetition of certain basic synchronisations. The operation is defined as follows:

$$N \;\textbf{sy}\; A,$$

where $N = (S, T, \iota)$ is a given M-net and A is a given action symbol (section 2.2). We first describe informally the basic synchronisation, then we give the full definition of the repetition of this basic synchronisation, and finally we define basic synchronisations formally.

To explain the basic synchronisation, consider Figure 3. The upper part of the figure shows two transitions, t_1 and t_2, one of which contains a term mentioning the action variable A in its label and the other one contains a term including $\overline{A}$ in its label. They can synchronise (as we shall say, *over* A) using these two terms, yielding an additional new transition. This new transition can be thought of being constructed in several steps.

First of all, the label of t_1 contains two action terms mentioning A, but only one of them may be chosen to be matched with the term $\overline{A}(a,d)$ of the label of t_2; let us choose the first of the two, i.e., $A(a,b)$. Now, we try to unify the two chosen matching terms, such that they have the same arguments. The variables in corresponding areas must be substituted in order to make $VAR(t_1)$ and $VAR(t_2)$ disjoint except in the arguments of the chosen terms. This substitution is necessary because by synchronisation, the two areas of the transitions t_1 and t_2 are combined to result in a new single area. When the two areas are combined, it is necessary to make explicit the previously implicit distinction between the d belonging to t_1 and the d belonging to t_2.

To this end we consider two substitutions θ_1, θ_2: $VAR \to VAR$. The first substitution pertains to the area of t_1, the second to the area of t_2. On the left hand side of figure 3, b is substituted through θ_1 by itself, b, and on the right hand side of the figure, pertaining to t_2, b is substituted through θ_2 by b_1. The analogous happens with d: on the left it is substituted by d and on the right it is substituted by b (since this is the variable matching d as the second argument of A). The middle row of Figure 3 shows that we have resorted to a new name b_1 in order to get the variable d of t_2 (indirectly) out of the way of the variable d of t_1.

In the final step, a new transition (called t_{12}) is created. It has the same arcs with the same labels as t_1 and t_2 (after the substitutions, of course - note that the substitutions have also modified the arc annotations). The label of t_{12} is simply the multiset sum of the two constituent labels, minus the matching pair of terms. The annotation of t_{12} is also the set sum of the two constituent annotations, written in our example as a logical conjunction. The last row of the figure shows t_{12} with its full annotation, but, for brevity, the annotations of t_1 and t_2 have been omitted (they are the same as in the top row of the figure).

Had we initially chosen the term $A(c,d)$ to synchronise with the term $\overline{A}(a,d)$ of t_2, then this would also have been possible. But there would have had to be different substitutions, and a different (not even renaming equivalent) transition t'_{12} would have resulted.

The transition t_{12} of the last part of Figure 3 contains a term with A, and hence, it can again synchronise with the transition t_2 which contains a term with $\overline{A}$ (this is what was meant by the *repetition* of the basic synchronisation). Since the transitions t_2 and t_{12} share an input place, the resulting arc inscription would then be a proper multiset of variable names.

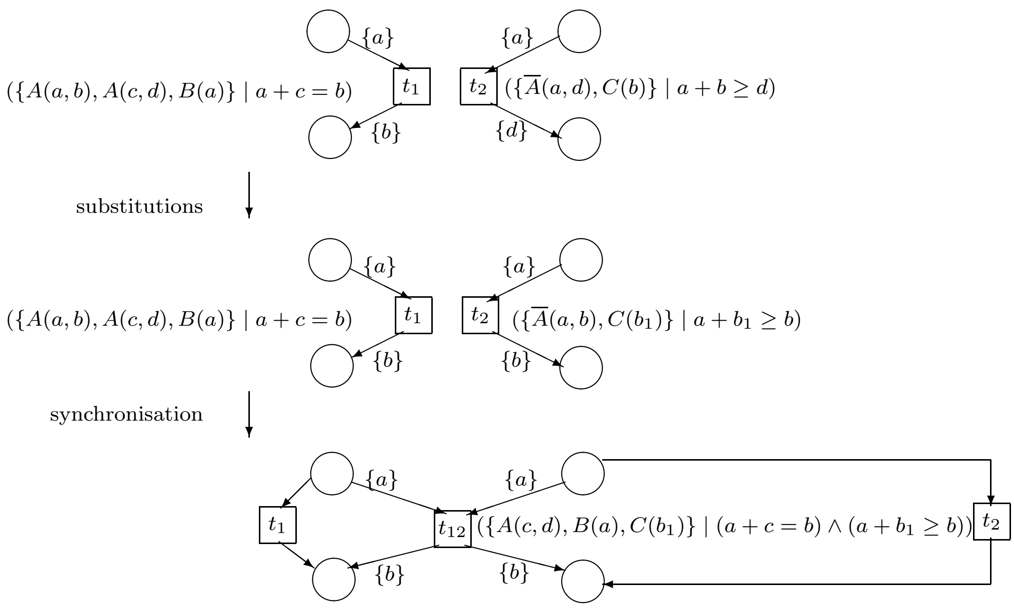

Figure 3: Explanation of the basic synchronisation in creating N **sy** A

Definition 4.1 (Synchronisation)
Let $N = (S, T, \iota)$ be an M-net and A an action symbol. The net

$$N' = N \textbf{ sy } A = (S', T', \iota')$$

is defined as the smallest M-net satisfying:

(a) The set of places of N and N' (and their inscriptions) are exactly the same: $S = S'$.

(b) Every transition of N (and its set of surrounding arcs) are also in N', all with the same inscriptions as in N.

(c) If t_1 is a transition of N and t_2 a transition of N', such that one of them contains an action term with A in its label and the other one with $\overline{A}$, then any transition t arising through a *basic synchronisation* out of t_1 and t_2 over A (and its surrounding arcs) is also in N'. $\diamond$

Note the asymmetry between t_1 and t_2 in step (c). This makes the definition more amenable to an induction argument. Later, at the end of this section, we show that the definition could have been phrased symmetrically without change of generality. The symmetric definition will be essential for the proof of commutativity while the asymmetrical one is useful in the inductive proof of coherence of unfolding. Let us also remark that a transition arising out of two transitions t_1, t_2 is not unique; each conjugate pair of action terms can define one transition (as for instance t_{12} and t'_{12} in the example). Note also that in step (c) we may even have the case that $t_1 = t_2$, namely when the label of this transition contains both a term with A and a term with $\overline{A}$. The definition is completed by defining the basic synchronisation formally.

Definition 4.2 (Basic synchronisation of two transitions)
Let t_1 and t_2 be transitions of N and N', respectively, as above. A transition t (and its surrounding arcs) *arises out of* t_1 *and* t_2 *through a basic synchronisation over* A, if there is an action term $A(a_1, \ldots, a_m)$ in the label of t_1 and an action term $\overline{A}(b_1, \ldots, b_m)$ in the label of t_2 ($ar(A) = m = ar(\overline{A})$), as well as two substitutions θ_1 and θ_2 which are decomposable with two renamings ρ_1, ρ_2 and a substitution θ as follows: $\theta_1 = \rho_1\theta$ and $\theta_2 = \rho_2\theta$, such that the five following conditions hold:

1. ρ_1, ρ_2 are separating substitutions for the areas of t_1 and t_2, i.e.:

$$VAR(t_1)[\rho_1] \cap VAR(t_2)[\rho_2] = \emptyset.$$

2. θ is the most general unifier of terms[9] (parameters) of the two parameterised actions in question:

$$(a_1, \ldots, a_m)[\rho_1\theta] = (b_1, \ldots, b_m)[\rho_2\theta].$$

3. The arc inscriptions around t are computed as follows, using $\oplus$ for multiset sum:

$$\begin{array}{lcl} \forall s \in S: \ \ \iota((s,t)) & = & \iota((s,t_1))[\theta_1] \oplus \iota'((s,t_2))[\theta_2] \\ \forall s \in S: \ \ \iota((t,s)) & = & \iota((t_1,s))[\theta_1] \oplus \iota'((t_2,s))[\theta_2]. \end{array}$$

4. The label of t contains all parameterised actions from $\lambda_{t_1}[\theta_1]$ and from $\lambda_{t_2}[\theta_2]$ less the conjugate pair $A((a_1, \ldots, a_m)[\theta_1]), \overline{A}((b_1, \ldots, b_m)[\theta_2])$:

$$\lambda_t = (\lambda_{t_1}[\theta_1] \oplus \lambda_{t_2}[\theta_2]) \ominus \{A(\vec{u_1}[\theta_1]), \overline{A}(\vec{u_2}[\theta_2])\},$$

where $\ominus$ denotes multiset difference.

5. The annotation of t is the union of annotations of t_1 and t_2 where variables are substituted:

$$\alpha_t = \alpha_{t_1}[\theta_1] \cup \alpha_{t_2}[\theta_2].$$

$\diamond$

[9] Recall that terms can be only variables or constants.

Note that the pair of substitutions θ_1, θ_2 is unique up to renaming. The choice of the most general unifier makes the number of created transitions as small as possible. However, this choice is not necessary; if it is not obeyed, then simply more duplicating transitions are created.

As mentioned before, definition 4.1 can be rephrased by changing item (c) into:

> (c)$'$. If t_1 and t_2 are transitions of N', such that one of them contains an action term with A and the other one with $\overline{A}$, then any transition t arising through a basic synchronisation out of t_1 and t_2 over A (and its surrounding arcs) is also in N'.

To switch from one definition to the other is possible by the following proposition.

Proposition 4.3 *The definitions* 4.1*, and* 4.1 *with* (c) *replaced by* (c)$'$*, are equivalent.*

Proof: *(Sketch)* Let us call (a)-(c) the asymmetrical definition, (a)-(c)$'$ the symmetrical definition. It is clear that every transition created by the asymmetrical definition can also be created by the symmetrical definition. Conversely, let t be created by the symmetrical definition. This means that there are two N'-transitions t' and t'' such that t arises by a basic synchronisation step out of t' and t''. By an induction hypothesis, we may assume that t' and t'' satisfy the asymmetrical definition, i.e., that there are transitions t_1' and t_2' such that t_1' is in N (not N'!) and t_2' is in N' and t' arises from a basic synchronisation out of t_1' and t_2'; and similarly that t'' arises from a basic synchronisation out of t_1'' and t_2'', for some transitions t_1'' in N and t_2'' in N'.

In the language of [2], any conglomerate of such synchronisations forms a finite spanning tree in the graph of possible conjugate pair connections generated by the transition labels involved, counting multiplicities properly. Any nontrivial spanning tree (i.e., one not just consisting of a single node) has at least two endpoint nodes that border on exactly one edge because otherwise it would contain a cycle. This means that t' and t'' have at least four such nodes, and no matter how t' and t'' are synchronised together by a basic step, at most two of them can disappear and in the resulting synchronised conglomerate still at least two such endpoints are available. Any one of these endpoints will do as a transition t_1 in N, and the rest as a transition t_2 in N' which, when synchronised together, yield t.

A detailed proof is given in [12] for A-nets which are generalisation of M-nets, supporting algebraic data representation. ■

4.6 Restriction

Intuitively, the restriction operation N **rs** A removes from N all transitions that mention either A or $\overline{A}$, and hence all synchronisation capabilities for A.

Definition 4.4 (Restriction)
Given an M-net $N = (S, T, \iota)$, and a pair of conjugate action symbols A, we define N **rs** A as the M-net $N' = (S', T', \iota')$ where

— $S' = S$.

— $T' = \{t \in T \mid \iota(t) = (\lambda_t \mid \alpha_t)$ and λ_t contains no action term with either A or $\overline{A}\}$.

— $\iota' = \iota$ restricted to S' and T'. ◇

In Figure 4(lhs) we show an M-net consisting of two disjoint parts. In Figure 4(rhs) we show the M-net resulting from the synchronisations and restrictions over Y and Y_t.

5 Properties of the M-net algebra

This section mainly states coherence results of the form: the unfolding of a composite object is the composition of its unfoldings. However, for synchronisation and restriction we also state some algebraic properties that do not pertain to unfolding.

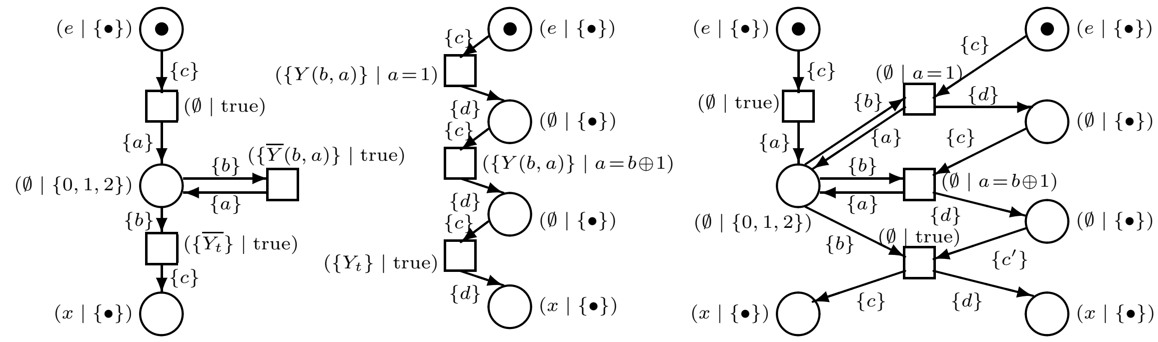

Figure 4: An M-net before (lhs) and after (rhs) synchronisation and restriction over Y and Y_t.

5.1 Coherence of sequence, concurrent composition, choice and iteration

The next result states that vertical unfolding commutes with the compositions of sequence, concurrent composition, choice and iteration introduced in the previous section.

Proposition 5.1 *(Coherence of unfolding and composition)*
Let N_1, N_2 and N_3 be M-nets. Then

$$\begin{array}{rcl} U(N_1;N_2) & = & U(N_1);U(N_2) \\ U(N_1 \| N_2) & = & U(N_1) \| U(N_2) \\ U(N_1 \,\square\, N_2) & = & U(N_1) \,\square\, U(N_2) \\ U(\,[N_1 * N_2 * N_3]\,) & = & [\,U(N_1) * U(N_2) * U(N_3)\,] \end{array}$$

Proof: More or less directly (modulo technical details) from the definitions. ■

5.2 Coherence and algebraic properties of synchronisation

In this section we state a result for synchronisation that is similar to proposition 5.1. First, let us remark that $\{A, \overline{A}\}$, the considered pair of conjugate action symbols, is the same in an M-net and in its unfolding.

Proposition 5.2 *(Coherence of unfolding and synchronisation)*
Let N be an M-net and A a pair of conjugate action symbols. Then

$$U(N \textbf{ sy } A) \;=\; U(N) \textbf{ sy } A.$$

In order to make the proof of Proposition 5.2 easier, the recursive definition of synchronisation 4.1 given in the Section 4.5 can be viewed in an iterative form as the union of successive synchronisation steps, seen as binary operations on M-nets:

$$N \textbf{ sy } A = \bigcup_{k \geq 0} N_A^k \quad \text{where} \quad N_A^0 = N \quad \text{and} \quad N_A^{k+1} = N \;\circ_A\; N_A^k$$

and $\circ_A$ is a synchronisation step, i.e., an operation which, given two M-nets N and N' with the same sets of places, yields an M-net (also with the same set of places) which contains exactly the transitions (and their surrounding arcs) which arise, for each $t \in T_N$ and $t' \in T_{N'}$, through a basic synchronisation over A, out of t and t'.

Thus, for all $k \geq 0$ the M-net N_A^k contains exactly the transitions resulting from exactly k basic synchronisations, and it is an M-net which has the same set of places as the original net N, as is the union $N \textbf{ sy } A = \bigcup_{k \geq 0} N_A^k$.

Lemma 5.3 *Let N and N' be two M-nets and A an action symbol. Then*

$$U(N \;\circ_A\; N') = U(N) \;\circ_A\; U(N').$$

Proof: The sets of places of $U(N \ \circ_A \ N')$ and $U(N) \ \circ_A \ U(N')$ are equal by definition. To prove that $U(N \ \circ_A \ N') \equiv U(N) \ \circ_A \ U(N')$ it is sufficient to show that each transition t from $U(N \ \circ_A \ N')$ belongs also to $U(N) \ \circ_A \ U(N')$ and vice versa. By definition of synchronisation, the surrounding arcs will be correct.

- $t_0 \in T_{U(N \ \circ_A \ N')} \Rightarrow t_0 \in T_{U(N) \ \circ_A \ U(N')}$:
  - By unfolding, there exists a legal binding σ and a transition $t \in T_{N \ \circ_A \ N'}$ such that $U(t) = t_0$ under σ.
  - By the definition of a synchronisation step $\circ_A$, there exist two transitions $t' \in T_N$ and $t'' \in T_{N'}$, and two substitutions $\theta_1 = \rho_1\theta$ and $\theta_2 = \rho_2\theta$, such that t arises out of t' and t'' through a basic synchronisation (definition 4.2).
  - By unfolding with the legal bindings $\sigma' = \theta_1\sigma$ and $\sigma'' = \theta_2\sigma$, there exist transitions $t'_{\sigma'} \in T_{U(N)}$ and $t''_{\sigma''} \in T_{U(N')}$ such that $U(t') = t'_{\sigma'}$ under σ', and $U(t'') = t''_{\sigma''}$ under σ'', and such that t_0 arises out of $t'_{\sigma'}$ and $t''_{\sigma''}$ through a basic synchronisation in the unfolded net, hence $t_0 \in U(N) \ \circ_A \ U(N')$.
- $t_0 \in T_{U(N) \ \circ_A \ U(N')} \Rightarrow t_0 \in T_{U(N \ \circ_A \ N')}$:
  - By the definition of a synchronisation step $\circ_A$, there exist two transitions $t_1 \in T_{U(N)}$ and $t_2 \in T_{U(N')}$, such that t_0 arises out of t_1 and t_2 through a basic synchronisation over A.
  - By unfolding, there exist two legal bindings σ_1 and σ_2, and two transitions $t_1^h \in T_N$ and $t_2^h \in T_{N'}$, such that $U(t_1^h) = t_1$ under σ_1, and $U(t_2^h) = t_2$ under σ_2. We assume σ_1 and σ_2 chosen in such a way that the sets of variables on which σ_1 and σ_2 differ from identity are disjoint, which is always possible.
  - By definition 4.2, there exists a transition $t \in T_{N \ \circ_A \ N'}$ which arises out of t_1^h and t_2^h through the basic synchronisation with some substitutions $\theta_1 = \rho_1\theta$ and $\theta_2 = \rho_2\theta$, where ρ_1 and ρ_2 are renamings, and θ a substitution, as in definition 4.2. Because of the particular choice of σ_1 and σ_2 we can take ρ_1 and ρ_2 equal to *id*.
  - By unfolding with the legal binding $\sigma = \theta\sigma_1\sigma_2$, there exists a transition $t_\sigma \in T_{U(N \ \circ_A \ N')}$, such that $t_\sigma = U(t)$ under σ. But t_σ is exactly t_0, hence $t_0 \in T_{U(N \ \circ_A \ N')}$. ■

Lemma 5.4 *Let N be a M-net and A an action symbol. Then*

$$\forall k \ \ U(N_A^k) = (U(N))_A^k.$$

Proof: By induction on the number of steps of $\circ_A$, k.

- In case $k = 0$ (induction base) we have to show that $U(N_A^0) = (U(N))_A^0$.

  By definition of the initial steps $U(N_A^0) = U(N)$ and $(U(N))_A^0 = U(N)$. Then the equality $U(N_A^k) = (U(N))_A^k$ is true for $k = 0$.
- If $\forall n \leq k$ we have $U(N_A^n) = (U(N))_A^n$ (induction hypothesis), then $k \rightsquigarrow k+1$ as follows:

$$\begin{array}{rcll} U(N_A^{k+1}) & = & U(N \ \circ_A \ N_A^k) & \text{by the definition of } \circ_A, \\ & = & U(N) \ \circ_A \ U(N_A^k) & \text{by lemma 5.3,} \\ & = & (U(N))_A^{k+1} & \text{by the definition of } \circ_A. \end{array}$$

■

Proof: (of Proposition 5.2)
By lemma 5.4 we obtain $\forall k \ \ U(N_A^k) = (U(N))_A^k$. Then also $\bigcup_{k\geq 0} U(N_A^k) = \bigcup_{k\geq 0}(U(N))_A^k$.
By definition of the synchronization operation this is equivalent to $U(N \ \mathbf{sy} \ A) = U(N) \ \mathbf{sy} \ A$ which is what we want to show. ■

Theorem 5.5 *(Commutativity and idempotence of synchronisation)*
Let N be an M-net and A and B two action symbols. Then

$$\begin{array}{rcl} (N \ \mathbf{sy} \ A) \ \mathbf{sy} \ B & = & (N \ \mathbf{sy} \ B) \ \mathbf{sy} \ A. \\ (N \ \mathbf{sy} \ A) \ \mathbf{sy} \ A & = & N \ \mathbf{sy} \ A. \end{array}$$

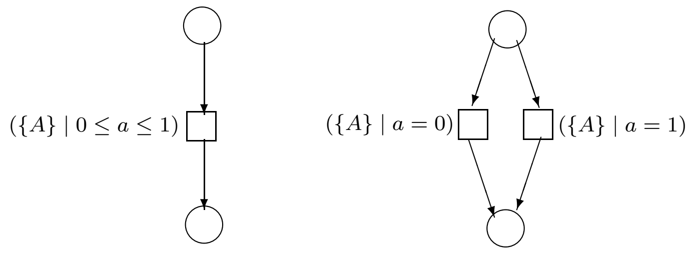

Figure 5: Two non-isomorphic M-nets whose unfoldings are isomorphic

One might be tempted to think that the commutativity of the synchronisation stated in Theorem 5.5 can be reduced, via unfolding, to the commutativity of synchronisation on the elementary level, which has already been proved in [2]. However, this is only possible if there is some connection between two M-nets whose unfoldings are isomorphic (or at least renaming equivalent, in the terms of [2]). Figure 5 shows two M-nets whose unfoldings - assuming a to be integer valued - are isomorphic, but which are not themselves isomorphic (nor renaming equivalent).

The example suggests investigating a notion of duplication equivalence on the M-net level, which might be connected to the equivalence defined in [2] on the elementary level. However, for the time being we have chosen not to pursue this approach, and instead, to prove Theorem 5.5 directly.

Proof: (of Theorem 5.5)

This result follows directly from the commutativity of synchronisation of A-nets [12].

The idempotence of synchronisation is an immediate consequence of the definitions. ∎

5.3 Coherence and algebraic properties of restriction

An analogous set of results can be proved with less effort for the restriction operation.

Proposition 5.6 *(Coherence of unfolding and restriction)*
Let N be an M-net and A an action symbol. Then

$$U(N \textbf{ rs } A) \;=\; U(N) \textbf{ rs } A.$$

Proof: Directly from the definitions of the restriction and the unfolding of an M-net. ∎

Proposition 5.7 *(Commutativity and idempotence of restriction)*
Let N be an M-net and A, A' two action symbols. Then

$$\begin{array}{lcl} (N \textbf{ rs } A) \textbf{ rs } A' & = & (N \textbf{ rs } A') \textbf{ rs } A. \\ (N \textbf{ rs } A) \textbf{ rs } A & = & N \textbf{ rs } A. \end{array}$$

Proof: Immediate from the definition of restriction. ∎

6 Concluding Remarks

In this paper, an algebra on the domain of high level nets has been proposed. The class of high level nets defined here is unique in having labels for transitions and places on which are defined the composition operations. The most novel and interesting is the synchronisation operation. This operation involves a unification step on labels, a logical conjunction step on annotations and a multiset addition step on surroundings arcs. Moreover this operation is commutative, an important and non-trivial property to ensure coherence with the elementary level box algebra.

Thus, the main results of this paper are of the sort that in category theory would be exhibited by means of commutative diagrams. We have two types of such diagrams: coherence of horizontal and vertical operations, and algebraic properties of the operations.

As well as the theoretical work on Box Calculi, a tool, called PEP, has been developed. The system presently allows a $B(PN)^2$ program to be translated into an elementary net, a temporal logic formula to be specified, and a fast model checker to be called. Our plans for future work include the intention to incorporate into the tool a high level net semantics given by M-nets as defined in the companion paper [4].

Acknowledgements

This work is partly supported by ESPRIT WG CALIBAN, no 6067. We are grateful to Raymond Devillers for very helpful comments. The first author would also like to thank Alexander Lavrov for commenting on a version of the manuscript.

References

[1] Bernd Baumgarten: Petri-Netze. Grundlagen und Anwendungen. BI-Wissenschaftsverlag, 1990.

[2] Eike Best, Raymond Devillers, and Jon Hall: The Box Calculus: a New Causal Algebra with Multi-label Communication. In Vol. 609 of LNCS, 21-69. Springer Verlag, 1992.

[3] Eike Best and Hans Fleischhack: PEP - Programming Environment Based on Petri Nets (project supported by the Deutsche Forschungsgemeinschaft, 1993-1994).

[4] Eike Best, Hans Fleischhack, Wojciech Frączak, Richard P. Hopkins, Hanna Klaudel and Elisabeth Pelz: An M-net Semantics of $B(PN)^2$. To appear in STRICT, Berlin, May 1995.

[5] Eike Best and Richard P. Hopkins: $B(PN)^2$ - a Basic Petri Net Programming Notation. In Proc. of PARLE, Springer LNCS, June 1993.

[6] Didier Buchs and Nicolas Guelfi: CO-OPN: A Concurrent Object-Oriented Petri Nets Approach for System Specification. In Proc. of 12th ICPN, Springer LNCS, 1991, also LRI Technical Report 616, Orsay, 1990.

[7] Jörg Desel and Javier Esparza: *Free Choice Petri Nets*. Cambridge University Press, Tracts in Theoretical Computer Science, 1994.

[8] Soren Christensen and Laure Petrucci: Towards a Modular Analysis of Coloured Petri Nets. In Vol. 609 of LNCS, 113-133. Springer Verlag, 1992.

[9] Javier Esparza: Model Checking based on Branching Processes. Hildesheimer Informatik-Bericht (December 1992). Published as: J. Esparza: Model Checking Using Net Unfoldings. Proceedings of TAPSOFT, M.C. Gaudel, J.P. Jouannaud (ed.), In Vol. 668 of LNCS, 613-628, 1993. To appear in Science of Computer Programming.

[10] Hartmann Genrich: Predicate-transition Nets. In Petri Nets: Central Models and their Properties, Advances in Petri Nets 1986, Part I, LNCS Vol.254, Springer-Verlag, 207-247, 1987.

[11] Kurt Jensen: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1. EATCS Monographs on Theoretical Computer Science, 1992.

[12] Hanna Klaudel and Elisabeth Pelz: Communication as Unification in the Petri Box Calculus. To appear in FCT'95, Dresden, August 1995.

[13] Robin Milner: A Calculus of Communicating Systems. Vol. 92 of LNCS. Springer Verlag, 1980.

[14] Robin Milner: Communication and Concurrency. Prentice Hall, 1989.

[15] Giuseppe Pinna: Petri Nets and Their Composition Problem. PhD Thesis: TD-2/90, Dipartimento di Informatica, Universita degli Studi di Pisa, March 1990.

[16] Wolfgang Reisig: Petri Nets and Algebraic Specifications. SFB-Bericht 342/1/90/B (March 90). Theoretical Computer Science Vol.80, 1-34, 1991.

[17] Einar Smith: A Primer on High-Level Petri-Net Theory, Informatik-Bericht Nr. 25 der Humboldt-Universität zu Berlin, 1993.

[18] Einar Smith and Wolfgang Reisig: The Semantics of a Net is a Net - an Exercise in General Net Theory. Advances in Petri Nets 87, LNCS, Springer-Verlag, 461-480, 1987.

[19] Peter H. Starke: Analyse von Petrinetz-Modellen. Verlag Teubner, 1990.

[20] Jacques Vautherin: Un modèle algebrique, basé sur les réseaux de Petri, pour l'étude des systèmes parallèles; Thèse de 3ème cycle, LRI, Orsay, 1985.

[21] Jacques Vautherin: Parallel systems specification with colored Petri nets and algebraic specification, In Vol. 266 of LNCS, 1987.

A Appendix

A.1 Operations on Multisets

Let Z be a set. The set of multisets over Z, denoted by $\mathcal{M}(Z)$, is defined as

$$\mathcal{M}(Z) = \{\mu \mid \mu\colon Z \to \mathbf{N}\}.$$

A multiset μ is finite if $\{a \in Z \mid \mu(a) > 0\}$ is finite. The set of finite multisets over Z is denoted by $\mathcal{M}_f(Z)$. The usual set enumeration notation can readily be extended. For instance, $\{a, a, b, c, c, c\}$ denotes the multiset μ with $\mu(a) = 2$, $\mu(b) = 1$ and $\mu(c) = 3$. A multiset μ is a set (a subset of Z) iff $\mu(a) \leq 1$ for all $a \in Z$. The set theoretical operations are extended as follows:

$$\begin{array}{lrcl}
\text{Union:} & (\mu_1 \cup \mu_2)(a) & = & \max(\mu_1(a), \mu_2(a)) \\
\text{Intersection:} & (\mu_1 \cap \mu_2)(a) & = & \min(\mu_1(a), \mu_2(a)) \\
\text{Sum:} & (\mu_1 + \mu_2)(a) & = & \mu_1(a) + \mu_2(a) \\
\text{Difference:} & (\mu_1 - \mu_2)(a) & = & \begin{cases} \mu_1(a) - \mu_2(a) & \text{if } \mu_1(a) \geq \mu_2(a) \\ 0 & \text{otherwise.} \end{cases}
\end{array}$$

Multiset inclusion is defined as $\mu_1 \subseteq \mu_2$ iff $\forall a \in Z\colon \mu_1(a) \leq \mu_2(a)$.

A.2 Auxiliary net operators

Definition A.1 (Making disjoint copies of a net)
For an integer i and an M-net $N = (S, T, \iota)$ we define $iN = (\{(s, i) \mid s \in S\}, \{(t, i) \mid t \in T\}, \iota')$ where

$$\forall x \in S \cup T\colon \iota'(x, i) = \iota(x) \quad \text{and} \quad \forall (x, y) \in (S \times T) \cup (T \times S)\colon \iota'(((x, i), (y, i))) = \iota((x, y)). \ \diamond$$

Definition A.2 (Place multiplication)
For a set $\mathcal{S} = \{S_1, \ldots, S_n\}$ of mutually disjoint sets (of places) $\iota_1, \ldots, \iota_n$, we define

$$\otimes\mathcal{S} \ = \ \{\pi \subseteq \bigcup_{i=1}^{n} S_i \mid |\pi \cap S_i| = 1 \text{ for each } i \in \{1..n\}\}.$$

Each of the S_i is a set of places which participates in the place multiplication. A new place $\pi = \{s_1, \ldots, s_n\}$ with $s_i \in S_i$ for each $i \in \{1..n\}$ is obtained for every possible way of taking one place from each participating set. The inscription $\iota(\pi)$ is obtained as follows:

$$\iota(\pi) = \begin{cases} (\lambda \mid \alpha) & \text{if } \forall i \in \{1..n\}\colon \ \iota(s_i) = (\lambda \mid \alpha) \\ (\emptyset, \{\bullet\}) & \text{otherwise} \end{cases}$$

$\diamond$

Note that place multiplication is symmetrical; e.g., the multiplication of two sets of places $\{r_1, r_2\}$ and $\{s_1, s_2\}$ is

$$\otimes\{\{r_1, r_2\}, \{s_1, s_2\}\} \quad = \quad \{\{r_1, s_1\}, \{r_1, s_2\}, \{r_2, s_1\}, \{r_2, s_2\}\}.$$

The next operation is used to add into a net a set of new places S', normally created by a place multiplication on sets of places of the net, and delete the original places used in the multiplication.

Definition A.3 (Place substitution)
Let $N = (S, T, \iota)$ be an M-net and $S' = \{\pi_1, \ldots, \pi_n\}$ such that for all $\pi_i \in S'$:

1. $\pi_i \subseteq S$
2. $\pi_i \neq \emptyset$

Taking $\Pi = \pi_1 \cup \ldots \cup \pi_n$, we define $N \oplus S' = ((S \backslash \Pi) \cup S', T, \iota')$ where ι' is defined as follows:

$$\begin{array}{l}
\forall x \in (S\backslash\Pi) \cup T \colon \iota'(x) = \iota(x) \\
\forall \pi \in S' \colon \iota'(\pi) = \left\{ \begin{array}{ll} (\lambda \mid \alpha) & \text{if } \forall s \in \pi \colon \iota(s) = (\lambda \mid \alpha) \\ (\emptyset, \{\bullet\}) & \text{otherwise} \end{array} \right. \\
\forall x \in ((S\backslash\Pi) \times T) \cup (T \times (S\backslash\Pi)) \colon \iota'(x) = \iota(x) \\
\forall (\pi, t) \in (S' \times T) \colon \iota'((\pi, t)) = \sum_{s \in \pi} \iota((s, t)) \\
\forall (t, \pi) \in (T \times S') \colon \iota'((t, \pi)) = \sum_{s \in \pi} \iota((t, s)).
\end{array}$$

The sum in the last two lines denotes multiset sum (of multisets of variables). $\diamond$

As it happens, we shall use this construction in such a way that the following holds in addition to the properties (1.) and (2.):

$$\forall t \in T \colon |{}^\bullet t \cap \pi_i| \leq 1 \geq |t^\bullet \cap \pi_i|,$$

for all elements π_i of S'. Here, ${}^\bullet t$ and $t^\bullet$ denote the set of predecessor and successor places of t, respectively.

Definition A.4 (Net union)
Let $N = (S, T, \iota)$ and $N' = (S', T', \iota')$ be two M-nets. Their union is defined as

$$N \cup N' \quad = \quad (S'', T'', \iota''),$$

where $S'' = S \cup S'$, $T'' = T \cup T''$, and all inscriptions of ι'' are carried over from ι or ι', respectively, whatever applies. $\diamond$