Refinement of Coloured Petri Nets

Eike Best[1] and Thomas Thielke[2]

[1] Carl-von-Ossietzky–Universität Oldenburg, D-26111 Oldenburg, e.best@informatik.uni-oldenburg.de
[2] Universität Hildesheim, D-31141 Hildesheim, thielke@informatik.uni-hildesheim.de

Abstract In this paper we describe and apply a multirelational approach for performing linear-algebraic refinement of coloured Petri nets.

1 Introduction

In this paper, we present a general linear-algebraic mechanism in order to describe the refinement of a transition or a place of one net by another net, where both nets can be high-level (coloured [4]). The mechanism will be based on the multirelational approach of [2, 5].

2 Motivation

2.1 A first example

Let us consider N_1 and N_2 in Figure 1 as an example. Arbitrarily consider N_1 as the 'base net' which is to be modified by 'refining N_2 into' it. As this makes no sense without further specifications, we pick some element of N_1 which is to be 'refined by N_2', say a. Inserting a net into a must affect the environment of a, that is, the place before a and the place after a. The former must be replaced by 'the beginning', and the latter by 'the end', of the net that is to be refined into a, which in our case is N_2. To specify this, we use a function f defined on the places of N_2, which yields, for a place s, either $\mathbf{e}$ (if s belongs to the 'beginning' of N_2), or $\mathbf{x}$ (if s belongs to the 'end' of N_2) or $\mathbf{i}$ (if s belongs neither to the beginning nor to the end of N_2). These letters are chosen in order to remind of 'entry', 'exit' and 'internal', respectively.

In general, let N_1, N_2, a transition t of N_1 and a function from the places of N_2 to the set $\{\mathbf{e},\mathbf{i},\mathbf{x}\}$ be given. We consider, then, the net $N = N_1[t \leftarrow N_2, f]$ to be well defined both intuitively and formally. To give a formal definition will be the objective of Section 3 of the present paper. First, however, let us check how different choices of f affect the refinement:

- Consider N_2 of Figure 1 and choose first a function f with $f(s_4)$=$\mathbf{e}$ and $f(s_5)$=$\mathbf{x}$. This means that the beginning of N_2 is $\{s_4\}$, the end of N_2 is $\{s_5\}$, and N_2 has no internal places (it corresponds, very likely, to the way N_2 is 'read' by most readers). In this case, intuitively, refinement of a by N_2 modulo f yields the third net $N_1[a \leftarrow N_2, f]$ shown in Figure 1.

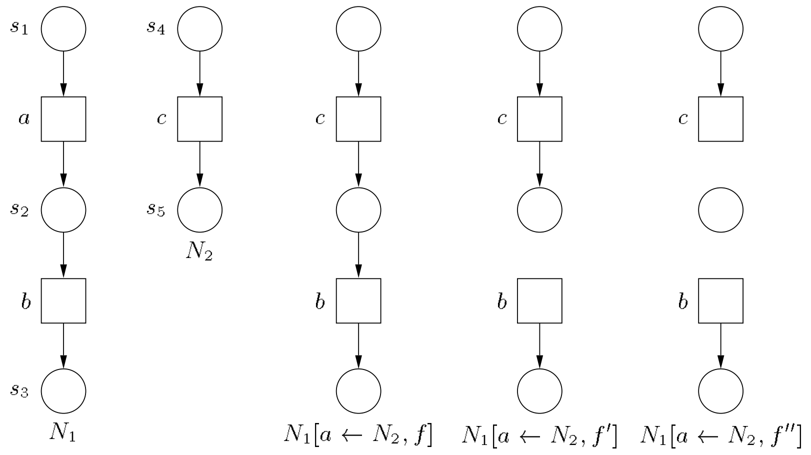

Fig. 1. Refinements using functions f, f' and f'' (Example A)

- Next, consider a function f' which specifies $f'(s_4)=\mathbf{e}$ and $f'(s_5)=\mathbf{i}$. In this case, N_2 has a beginning $\{s_4\}$ and some internal structure (only place s_5), but no end. However in N_1, it is specified that the 'end' of a (its output place) is also the 'beginning' of b. Hence, if a is replaced by something with no end, then the beginning of b vanishes as well as the end of a. Hence, we expect the refinement of a by N_2 modulo f' to be the fourth net, $N_1[a \leftarrow N_2, f']$, of Figure 1.
- Finally, consider a function f'' with $f''(s_4)=\mathbf{e}$ and $f''(s_5)=\mathbf{e}$, which specifies N_2 to have only a beginning, $\{s_4, s_5\}$, but neither internal place nor an end. This choice may make sense intuitively; the reader may think, for instance, of all $\mathbf{e}$-places receiving one token initially and letting N_2 be executed in this initial marking. However, the refinement of a by N_2 modulo f'' is not immediately clear. The most systematic choice seems to be the net shown on the right-hand side of Figure 1.

The main objective of this paper is to provide a general definition resolving the above question – what it means to refine a transition of a net N_1 by a net N_2 modulo f – in a very general setting, namely in the setting of coloured nets as defined in [4], or more precisely, in the setting of the underlying semantic model of coloured nets: colour sets, multisets thereof, and functions thereon. We will not just give this definition, but also explore some of its properties and discuss some of its uses. We intend to show that, not only is such a general definition possible in what seems a fairly clean way, but also does it give an intuitively satisfactory answer. All proofs can be found in the longer version of the paper, obtainable from one of the authors.

2.2 The linear algebraic setup

In our definition of refinement, we use the linear-algebraic formalism defined in [2]. This allows one to manipulate a coloured net by splitting and/or combining places and/or transitions in such a way that all nets so obtained are still equivalent representations of each other (or, if the reader prefers, of the underlying elementary net). We use this formalism both on N_1, the net of which transition t is to be refined, and on N_2, the net which is provided with a function f and which is to be inserted into t, modulo f. We use it on N_1 in order to isolate t and its surrounding places from the rest of the net. And we use it on N_2 in order to isolate the **e**/**x**-places (collectively also called the interface places) and their immediately neighbouring transitions from the rest of the net. Once this isolation has been carried out, there is then a formula which allows the two nets to be combined. Let us consider this mechanism on an example, N_1 and N_2 of Figure 2.

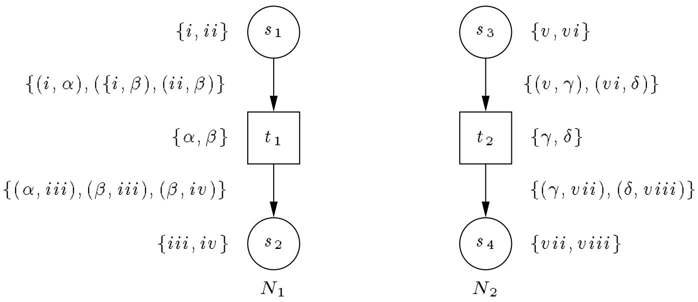

Fig. 2. A first coloured net example (Example B)

Before considering refinement in this example, we explain it. The type of s_1, $\{i, ii\}$, means that s_1 may hold (any multiset of) tokens i or ii. Similarly, s_2 has type $\{iii, iv\}$, which means that it may hold tokens – or, as they are also called, values – iii or iv. The type of t_1, $\{\alpha, \beta\}$, means that t_1 may occur in one of two modes, α or β. Similarly, the type of t_2 means that t_2 can occur in mode either γ or δ.

Notice that both nets have the property that not only are all place types (value sets) disjoint from all transitions types (mode sets), but that this is the case also mutually: types of different places are disjoint and types of different transitions are also disjoint. This is a pleasant property which makes life (in terms of formalisation) easy. Fortunately, it is not a restrictive property because it can always be made true by simple renaming [2]. We will call it the 'disjointness property'. Notice that it implies that we can, in principle, neglect the names of places and transitions and identify them instead by their types. For instance, in Figure 2, instead of saying 'the place s_1', we could equally well (exploiting disjointness) say 'the place $\{i, ii\}$'.

We continue to explain the formalism used in Figure 2. The first arrow, $\{(i, \alpha), (i, \beta), (ii, \beta)\}$, means that for t_1 to occur in mode α, there needs to be at least one token i on s_1 (such a token could be provided by the 'current marking', a notion that is usual in net theory, but plays only a background role in the present paper), and that if t_1 occurs in mode α, one such token i will be removed from s_1, as well as that for t_1 to occur in mode β, s_1 needs to hold at least one token i and one token ii, and that if t_1 occurs in mode β, a token i and a token ii will be removed from s_1. The arc $\{(\alpha, iii), (\beta, iii), (\beta, iv)\}$ means that if t_1 occurs in mode α, a token iii is added to place s_2, and that if t_1 occurs in mode β, two tokens (one being iii, the other being iv) are added to s_2.

Thus, consider, for example, the marking M_0 which puts two tokens i and two tokens ii on s_1, and no tokens on s_2. In this marking, t_1 can occur in mode α or in mode β. In the first case, the resulting marking is: one token i on s_1, two tokens ii on s_1 and one token iii on s_2, and in the second case the resulting marking is: one token i on s_1, one token ii on s_1 and two tokens, iii and iv, on place s_2.

2.3 What is to be refined by what?

After explaining thus the formalism of [2], which coincides with the usual coloured net formalism [4] after a little reformulation, we now turn to the question of refining N_2 (of Figure 2) into N_1 (again of Figure 2). As in the previous example, this can be done only when the part of N_1 that is to be refined, as well as the interface of N_2 that is to be refined into it, have been specified. What should these be, however?

A first attempt could be to identify a transition, say t_1, of N_1, to attach elements of $\{\mathbf{e}, \mathbf{i}, \mathbf{x}\}$ to the places of N_2, say $\mathbf{e}$ to s_3 and $\mathbf{x}$ to s_4, and then try to define the resulting net. However, we consider this to be too coarse. The reason is that we want refinement to be stable over equivalent transformations, that is, if N_1' is equivalent with N_1 and N_2' is equivalent with N_2, then 'the same' refinement should be defined on N_1' and N_2', and lead to an equivalent result. Unfolding a net into its elementary net is a particular equivalence in this sense. Just to make this point clear, Figure 3 shows the unfoldings of the two nets of Figure 2, N_1' being the unfolding of N_1 and N_2' being the unfolding of N_2.

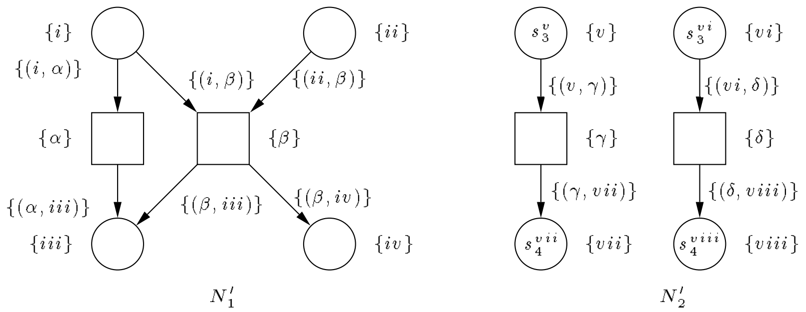

Fig. 3. Unfolding of the two nets of Figure 2 (Example B ctd.)

Unfoldings in this sense are characterised by the fact that all types are singleton sets. Consult [2] to see how N_1 and N_1', and N_2 and N_2', are connected in terms of transformation matrices. For the present context, it is important to note that the 'single transition', t_1, of N_1, has disappeared in N_1', and yet, N_1 and N_1' are equivalent. Thus, if we tried to model the refinement 'refine t_1, in N_1, by N_2' in terms of N_1' and N_2', we would be bound to fail because there is no transition t_1.

A similar effect is that the characterisation of s_3 as an **e**-place and s_4 as an **x**-place in Figure 2 leads canonically to a characterisation of s_3^v and s_3^{vi} (in N_2') as **e**-places and s_4^{vii} and s_4^{viii} as **x**-places, but this does not work the other way round. If, for instance, we characterised place s_3^v as **e**, places s_3^{vi} and s_4^{vii} as **i**, and place s_4^{viii} as **x** in N_2' (Figure 3), how then should we characterise s_3 and s_4 in N_2?

It follows from the above considerations that, in order to get a good formal – in particular, a stable – definition of the refinement of N_2 into N_1 (both coloured nets), we should select a mode m_0 – rather than a transition – in N_1, to be refined, and we should let the function f to be defined on the values – rather than the places – of N_2. Let us, for example, in Figure 2, choose the mode $m_0{=}\alpha$ of N_1, and for N_2 choose the function f defined by $f(v){=}f(vi){=}\mathbf{e}$ and $f(vii){=}f(viii){=}\mathbf{x}$. Notice that the same choice of m_0 and f are also possible in N_1' and N_2' (and in fact in any other nets equivalent with these two), so that such this choice is stable over equivalent transformations. It remains to define refinement in the same stable (and, of course, intuitively 'correct') way.

2.4 Isolating refinement interfaces

Now let us return to the two nets of Figure 2 and assume that a mode m_0, say $m_0{=}\alpha$, has been selected for N_1 and a function f, say the above one, for N_2. We now describe informally (part of) the definition of

$$N = N_1[m_0 \leftarrow N_2, f], \quad \text{the 'refinement of } m_0 \text{ in } N_1 \text{ by } N_2 \text{ modulo } f\text{'.}$$

Our first aim will be to isolate mode $m_0{=}\alpha$ of N_1 into a single transition and to isolate the input/output values of this mode from the rest of the values. Our second aim is to isolate the **e**/**x**-values of N_2 from the rest of N_2's values and to isolate the modes connected to those **e**/**x**-values from the rest of the modes. Both aims can be achieved by transformations as described in [2], one applied to N_1 and the other applied to N_2. Figure 4 shows the results, N_1'' and N_2''. Notice that N_1'' is again equivalent with N_1 (and with N_1') and that N_2'' is equivalent with N_2 (and with N_2'). Notice also how values i and ii (and also iii and iv) have been split in N_1'', since the former (but not the latter) are connected with mode $m_0{=}\alpha$. And notice that values v–$viii$ of N_2 have been combined into a single place, since all of them are **e**- or **x**-values (on the right-hand side of Figure 4, we have indicated function f by using superscripts **e** and **x** attached to values).

In N_1'', the mode to be refined, α, is in a side condition loop with a place describing all of (and only) its input/output values, i.e. i and iii. Such a

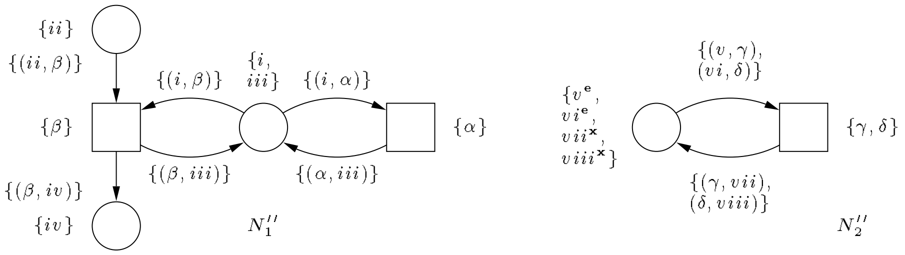

Fig. 4. Isolation of mode to be refined and interface places (Example B ctd.)

transformation can always be achieved, even if some or even all arcs of the loop are missing (which is equivalent with there being a $\emptyset$ arc). The rest of N_1'' is connected only to the place of that loop. In N_2'', the interface values, which happen to be all values, are collected in a single place, with all modes connected to them, which again happen to be all modes, forming a side transition loop. In the general case, when there are internal values and/or modes of N_2 not connected to $\mathbf{e}/\mathbf{x}$-values, the rest of N_2'' will be connected to the transition of that side loop, but not to its place.

Once both nets have been brought into this form, we will define an operation which will take them and essentially add them together, such that the two side loops, the one on the left and the one on the right, are superimposed and new types are formed by adding old types in a special way. The general scheme of this approach is shown in Figure 5. The place $s_1 \oplus s_2$ in the new side loop in $N_1[m_0 \leftarrow N_2, f]$ arises from superimposing the old places s_1 and s_2. Similarly, the transition $t_1 \oplus t_2$ arises from superimposing the transitions t_1 and t_2. Those are the only elements of the two nets where a type change occurs, and the arcs surrounding those two elements are the only ones where these type changes have any effect, i.e. which change as a result.

2.5 Discussing the commutativity of refinement

It is still too early, at this point, to explain the exact nature of the type changes from s_1, s_2 to $s_1 \oplus s_2$ and from t_1, t_2 to $t_1 \oplus t_2$. We first need to discuss the desired property of 'commutativity of refinement'. Fix any N_1. Choose any two modes in N_1, m_0 and m_1, the latter different from the former. Further, choose N_2 with function f, and N_3 with function g, disjoint from each other. Then, after refining m_0 by N_2 modulo f, the mode m_1 still exists and can be further refined by N_3 modulo g. The reverse order is, however, also possible, and it is highly desirable that no two different nets result from this. That is, we wish

$$(N_1[m_0 \leftarrow N_2, f])[m_1 \leftarrow N_3, g] \quad = \quad (N_1[m_1 \leftarrow N_3, g])[m_0 \leftarrow N_2, f]$$

to be true, at least in the sense that the left-hand side is transformation (and hence unfolding) equivalent with the right-hand side. To see the problems in getting this equality, consider the (fragment of an) example shown in Figure 6, with $f(ii)=f(iii)=\mathbf{e}=g(iv)=g(v)$.

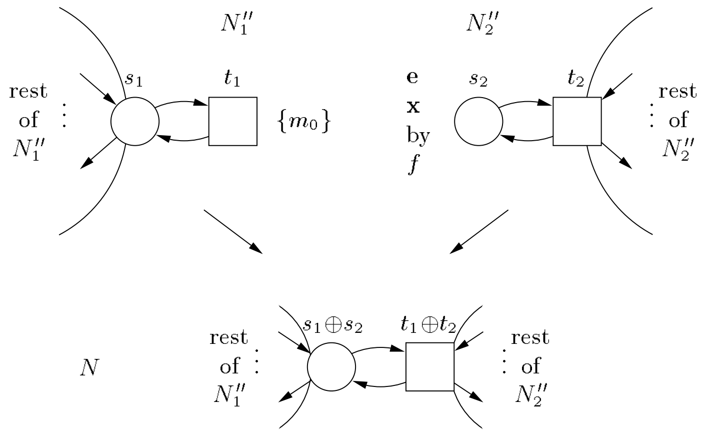

Fig. 5. Schematic representation of $N = N_1[m_0 \leftarrow N_2, f]$

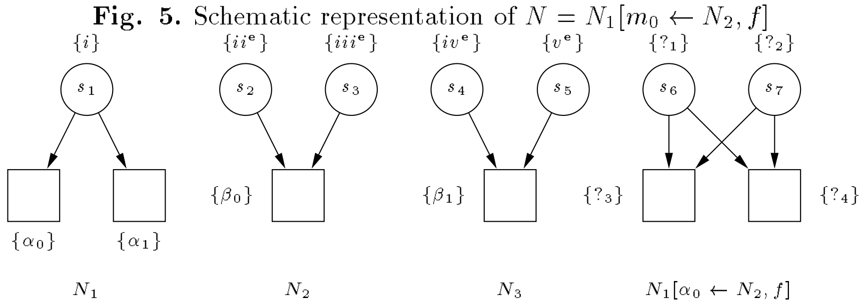

Fig. 6. Obtaining commutativity of refinement? (Example C)

The question is: what should be the types of s_6, s_7 and the two transitions connected to them? What is clear, first of all, is that all four types should be singletons, and thus, $?_i$ should be single values ($i = 1, 2$) and single modes ($i = 3, 4$). In defining these values and modes, it is also clear (except for a symmetric exchange of the places) that $?_1$ comes from i, ii, $?_2$ comes from i, iii, $?_3$ comes from α_0, β_0 and $?_4$ comes from α_1.

Should we perhaps let $?_1$ be a pair (i, ii) or a tree with root i and leaf ii? Such an approach is indeed possible and has been used successfully, for example in [1], and especially by Raymond Devillers to obtain deep results about S-invariant and S-component coverings in [3]. Unfortunately, pairs, trees, or any other device of this sort, will destroy the simple multirelational setup we obtained in [2], and which we intend to keep as much as possible.

Therefore, we have explored a way of defining types such as $?_1, \ldots, ?_4$ set theoretically. It turns out that only one further level of sets leads to a solution of the problem: we will allow values such as i, ii etc. in the above examples, as well as modes such as α, β etc., to be sets rather than unstructured entities. Thus, assuming some basic set $\{0, 1, 2, \ldots\}$ of 'numbers' and some other basic set $\{a, b, c, \ldots\}$ (both assumed infinite) of 'letters', disjoint from the numbers, we will allow values i, ii etc., to be sets of numbers, and modes α, β etc., to be sets of letters. In this spirit, Figure 7 rewrites Figure 6 by replacing i by $\{1\}$, ii by $\{2\}$, α_0 by $\{x\}$, etc.; let us also consider functions f and g with $f(\{2\}){=}f(\{3\}){=}\mathrm{e}{=}g(\{4\}){=}g(\{5\})$, i.e. the same f and g as for Figure 6.

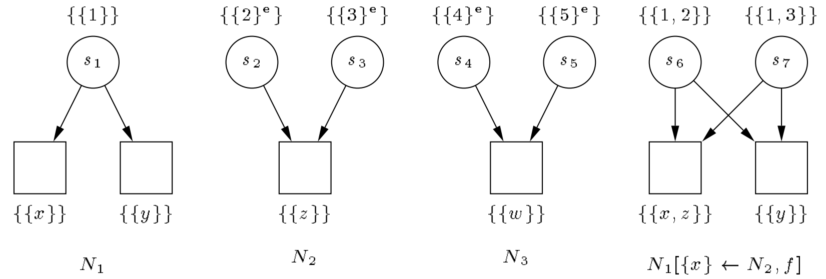

Fig. 7. Achieving commutativity of refinement! (Example C ctd.)

Note that, as it should be, the types of all elements of $N_1[\{x\} \leftarrow N_2, f]$ in Figure 7 are singletons, even though the values themselves are not singleton sets. Obtaining the type of s_6 from the types of s_1 and s_2 (from which s_6 originates) is very simple: take the type A of s_1 ($A{=}\{\{1\}\}$) and the type B of s_2 ($B{=}\{\{2\}\}$) and form the 'sum' $A{\oplus}B = \{\, a{+}b \mid a{\in}A, b{\in}B \,\}$ (which equals $\{\{1\}{+}\{2\}\}{=}\{\{1,2\}\}$ in our case, the $+$ being multiset sum). Note that if A and B are sets of multisets, then $A{\oplus}B$ is always well defined (and, in passing, satisfies $A{\oplus}\emptyset = \emptyset = \emptyset{\oplus}A$ and $A{\oplus}\{\emptyset\} = A = \{\emptyset\}{\oplus}A$). The type of the new transition is obtained by the same rule. If the right-hand net of Figure 7 is now further modified by refining mode $\{y\}$ of $N_1[\{x\} \leftarrow N_2, f]$ by N_3 modulo g, using the same rule again both for the new types of places and for the new types of the transitions, the net shown in Figure 8 is obtained.

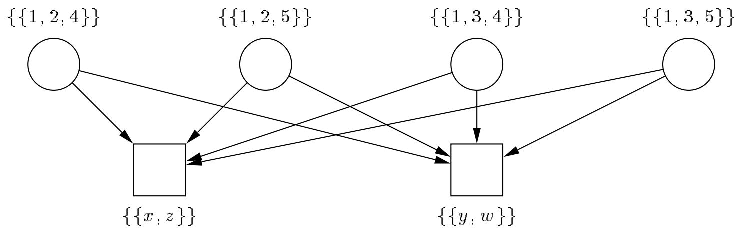

Fig. 8. $(N_1[\{x\}{\leftarrow}N_2, f])[\{y\}{\leftarrow}N_3, g]{=}(N_1[\{y\}{\leftarrow}N_3, g])[\{x\}{\leftarrow}N_2, f]$ (Example C ctd.)

It can be seen from the symmetry that exactly the same net would have been obtained if first $\{y\}$, then $\{x\}$, would have been refined by N_3, g and N_2, f, respectively. Thus, we get isomorphic nets if refinement is done in either order. We will generalise this approach, and in general, the two nets obtained will be unfolding-equivalent, but not necessarily isomorphic.

3 Formal definition of refinement

3.1 Sets and nets

We assume the following (infinite) sets, where $\mathcal{M}$ denotes 'multiset of', $\mathcal{M}_+$ denotes 'nonnegative multiset of' and $\mathcal{P}$ denotes 'powerset of':

$$\begin{aligned}&\mathit{Numbers} = \{0, 1, \dots\}, \mathit{Letters} = \{a, b, \dots\},\\ &\mathit{Values} = \mathcal{M}_+(\mathit{Numbers}), \mathit{Modes} = \mathcal{M}_+(\mathit{Letters}),\\ &\mathit{PlaceTypes} = \mathcal{P}(\mathit{Values}), \mathit{TransitionTypes} = \mathcal{P}(\mathit{Modes}),\end{aligned}$$

where $\mathit{Numbers} \cap \mathit{Letters} = \emptyset$. As in [2], a net is a triple $N = (S, T, \omega)$ where S is a set of place types, T is a set of transition types (together with S satisfying, without loss of generality, the disjointness property) and $\omega \in \mathcal{M}_+(S{\times}T{+}T{\times}S)$. By the disjointness property, we can identify places and transitions with their types. Hence, if we, for instance, say 'mode m_0 in net N', then this already uniquely identifies the transition in whose mode set m_0 occurs, since m_0 is guaranteed not to occur in any other transition's mode set.

3.2 Refinement

Let N_1 and N_2 be two nets. Let m_0 be a mode (not a transition) of N_1. Let f be a function from the values occurring in N_2 (not its places) to the set $\{\mathbf{e}, \mathbf{i}, \mathbf{x}\}$. Now we define the net $N = N_1[m_0 \leftarrow N_2, f]$. We do so in two steps:

Step 1: Isolate m_0 and its surrounding values in N_1 (**Step 1a**), and isolate $\mathbf{e}/\mathbf{x}$-places and its surrounding modes in N_2 (**Step 1b**). This can always be done by subjecting N_1 and N_2 to suitable transformation matrices as described in [2]. By $\mathcal{T}^{m_0}_{N_1}$ and $\mathcal{U}^{m_0}_{N_1}$ we denote the transformation matrices which achieve Step 1a and similarly by $\mathcal{T}^{f}_{N_2}$ and $\mathcal{U}^{f}_{N_2}$ we denote the transformation matrices which achieve Step 1b. We will use the following shorthands: $N_1' = \mathcal{T}^{m_0}_{N_1} \cdot N_1 \cdot \mathcal{U}^{m_0}_{N_1}$ and $N_2' = \mathcal{T}^{f}_{N_2} \cdot N_2 \cdot \mathcal{U}^{f}_{N_2}$, to denote the transformed nets. If one substitutes the incidence matrices (or the forward/backward flow relations) in place of N_1 and N_2 in these shorthand formulae, the incidence matrices (the flow relations) of N_1' and N_2' are, respectively, obtained by matrix multiplication [2]. Thus, N_1' and N_2' are transformation-equivalent with N_1 and N_2, respectively, and, additionally, have the following special forms:

- N_1' has a transition $t_1 = \{m_0\}$ and a place s_1 as shown in Figure 9.
- N_2' has a transition t_2 and a place s_2 as shown in Figure 10.

Thus, the incidence matrix of N_1' has a column where all entries but – at most – one are $\emptyset$, and the incidence matrix of N_2' has a row where all entries but – at most – one are $\emptyset$. This simplifies (somewhat) the definition in Step 2, which follows.

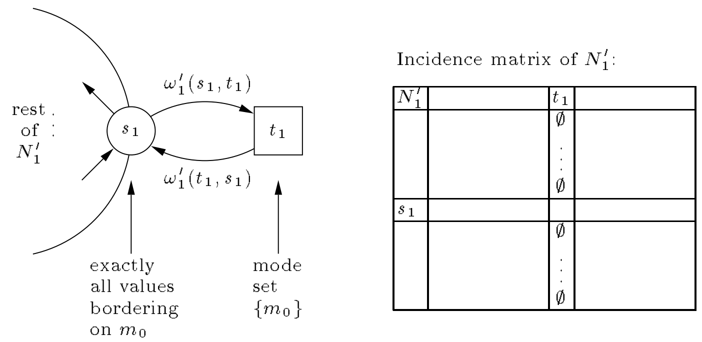

Fig. 9. Special form of N_1' after Step 1a (m_0-isolation)

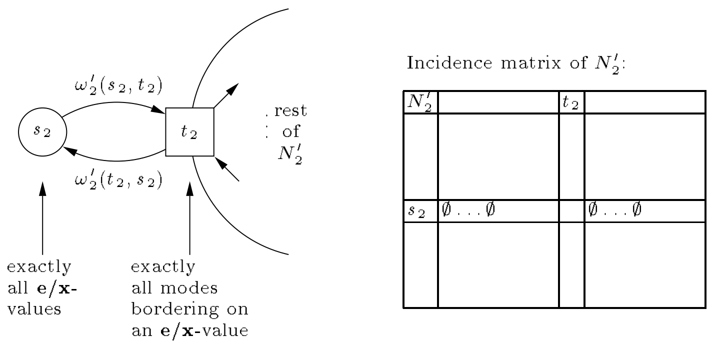

Fig. 10. Special form of N_2' after Step 1b (f-isolation)

Step 2: The net $N = N_1[m_0 \leftarrow N_2, f]$ will be defined using a binary 'sum' operation ${}_{m_0}\!\oplus_f$ whose left-hand argument is a net N_1' where m_0 is isolated and whose right-hand argument is a net N_2' where f is isolated. Thus,

$$N \;=\; N_1[m_0 \leftarrow N_2, f] \;=\; \underbrace{(\mathcal{T}_{N_1}^{m_0} \cdot N_1 \cdot \mathcal{U}_{N_1}^{m_0})}_{N_1'} {}_{m_0}\!\oplus_f \; \underbrace{(\mathcal{T}_{N_2}^{f} \cdot N_2 \cdot \mathcal{U}_{N_2}^{f})}_{N_2'}.$$

The net $N = N_1' {}_{m_0}\!\oplus_f N_2'$ will be defined by giving its sets of places and transitions and its flow relation explicitly, using the corresponding sets from N_1' and N_2':

$S = (S_1' \backslash \{s_1\}) \cup \{s_1 {}_{m_0}\!\oplus_f s_2\} \cup (S_2' \backslash \{s_2\})$, where $s_1 {}_{m_0}\!\oplus_f s_2$ is defined below.
$T = (T_1' \backslash \{t_1\}) \cup \{t_1 \oplus t_2\} \cup (T_2' \backslash \{t_2\})$, where $t_1 \oplus t_2$ is defined below.
ω = derived from the flow relations ω_1', ω_2' of N_1', N_2' as defined below.

Step 2a: (The type of) $s_1 {}_{m_0}\!\oplus_f s_2$ is the following set:

$\{\ a + k_1 \cdot b_1 + k_2 \cdot b_2 \ \mid\ a \in s_1,\ b_1 \in s_2,\ b_2 \in s_2,\ f(b_1) = \mathbf{e},\ f(b_2) = \mathbf{x},$
a occurs k_1 times as a first element of an element of $\omega_1'(s_1, t_1)$
and k_2 times as a second element of an element of $\omega_1'(t_1, s_1)$
$\}$.

Note that this is a good definition: a, b_1 and b_2 are multisets of numbers, k_1 and k_2 are natural numbers, so that $k_1{\cdot}b_1$, $k_2{\cdot}b_2$ and $a + k_1{\cdot}b_1 + k_2{\cdot}b_2$ are well defined. Moreover, k_1 is well defined, since $\omega_1'(s_1, t_1)$ is a finite multiset of pairs (w, m) where w could be a; and k_2 is well defined for analogous reasons.

Step 2b: (The type of) $t_1{\oplus}t_2$ is the following set: $\{\, a + b \mid a{\in}t_1,\ b{\in}t_2 \,\}$.

Step 2c: For the definition of ω, let $s{\in}S$ and $t{\in}T$. We need to define two multirelations, $\omega(s, t)$ and $\omega(t, s)$, but we define only the former because the latter is analogous. Figure 11 shows how the matrix $\omega(s, t)$ is built from the corresponding matrices of N_1' and N_2'.

$\omega(s,t)$	$T_1' \setminus \{t_1\}$	$t_1{\oplus}t_2$	$T_2' \setminus \{t_2\}$
$S_1' \setminus \{s_1\}$			
$s_1\,{}_{m_0}{\oplus}_f\, s_2$			
$S_2' \setminus \{s_2\}$			

Fig. 11. How the matrix $\omega(s, t)$ is divided up

Thus, we need to distinguish nine cases for s and t, corresponding to the nine fields in this matrix. We give these nine cases explicitly for reasons of clarity, even though some of them can be combined into single cases. The full definition of $\omega(s, t)$ is given by the **if**–**fi** case distinction in Table 1. In the definition, w denotes a value, m denotes a mode of N and '$\leq$' denotes multiset inclusion.

$$
\begin{array}{lll}
& (\omega(s,t))(w,m) \quad = & \\
1.\ \mathbf{if} & s{\in}S_1'\backslash\{s_1\}, t{\in}T_1'\backslash\{t_1\} & \to (\omega_1'(s,t))(w,m) \\
2.\ \square & s{\in}S_1'\backslash\{s_1\}, t = t_1{\oplus}t_2 & \to \emptyset \\
3.\ \square & s{\in}S_1'\backslash\{s_1\}, t{\in}T_2'\backslash\{t_2\} & \to \emptyset \\
4.\ \square & s = s_1\,{}_{m_0}{\oplus}_f\, s_2, t{\in}T_1'\backslash\{t_1\} & \to \sum\limits_{v_1 \leq w, v_1 \in s_1} (\omega_1'(s_1,t))(v_1,m) \\
5.\ \square & s = s_1\,{}_{m_0}{\oplus}_f\, s_2, t = t_1{\oplus}t_2 & \to \sum\limits_{\substack{v_1 \leq w,\, v_1 \in s_1 \\ n_1 \leq m,\, n_1 \in t_1 \\ v_2 \leq w,\, v_2 \in s_2 \\ n_2 \leq m,\, n_2 \in t_2}} (\omega_1'(s_1,t_1))(v_1,n_1){\cdot}(\omega_2'(s_2,t_2))(v_2,n_2) \\
6.\ \square & s = s_1\,{}_{m_0}{\oplus}_f\, s_2, t{\in}T_2' \setminus \{t_2\} & \to \emptyset \\
7.\ \square & s{\in}S_2'\backslash\{s_2\}, t{\in}T_1'\backslash\{t_1\} & \to \emptyset \\
8.\ \square & s{\in}S_2'\backslash\{s_2\}, t = t_1{\oplus}t_2 & \to \sum\limits_{n_2 \leq m, n_2 \in t_2} (\omega_2'(s,t_2))(w,n_2) \\
9.\ \square & s{\in}S_2'\backslash\{s_2\}, t{\in}T_2'\backslash\{t_2\} & \to (\omega_2'(s,t))(w,m) \\
\mathbf{fi}. & &
\end{array}
$$

Table 1. Definition of $\omega(s, t)$

In the long version of the paper, we show that this definition leads to the results given by our examples, and in particular, to the three nets shown on the right-hand side of Figure 1 (Example A), depending on the particular choices of

f, f', f''. Note also that, due to its symmetrical form, the definition can easily be rewritten for place/value refinement.

3.3 Properties of refinement

The two results to be stated in this section are: first, the definition of refinement given above is stable with respect to equivalence transformations, and second, the results of successive refinements carried out in different orders are equivalent. Formally, this is stated in Propositions 3.1 and 3.2.

Proposition 3.1. *Stability of refinement w.r.t. equivalence transformations*
Let N_1 be a net with mode m_0 and N_2 be a net with function f as above. Let K_1 and K_2 be transformation-equivalent with N_1 and N_2, i.e., using the shorthand introduced above, $K_1 = \mathcal{T}_1 \cdot N_1 \cdot \mathcal{U}_1$ and $K_2 = \mathcal{T}_2 \cdot N_2 \cdot \mathcal{U}_2$ with certain transformation matrices $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{U}_1, \mathcal{U}_2$. Then there exist transformation matrices $\mathcal{T}$ and $\mathcal{U}$ such that
$K_1[m_0 \leftarrow K_2, f] \;=\; \mathcal{T} \cdot (N_1[m_0 \leftarrow N_2, f]) \cdot \mathcal{U}$. ■ 3.1

Note that the left-hand side of the equation is well defined, because mode m_0 occurs in K_1 as well as in N_1 and function f applies to K_2 as well as to N_2. The proof of Proposition 3.1 works by a direct construction of the transformation matrices $\mathcal{T}$ and $\mathcal{U}$.

Proposition 3.2. *Commutativity of refinement modulo equivalence*
Let N_1 be a net with modes m_1 and m_2. Let N_2 and N_3 be nets with functions f (for N_2) and g (for N_3), as above. Then there exist transformation matrices $\mathcal{T}$ and $\mathcal{U}$ such that
$(N_1[m_2 \leftarrow N_3, g])[m_1 \leftarrow N_2, f] = \mathcal{T} \cdot (N_1[m_1 \leftarrow N_2, f])[m_2 \leftarrow N_3, g]) \cdot \mathcal{U}$.
■ 3.2

The proof of Proposition 3.2 works by generalising the observations made in Section 2.5.

Acknowledgment We thank the reviewers for their comments.

References

1. E. Best, R. Devillers, J. Esparza: General Refinement and Recursion Operators for the Petri Box Calculus. Proc. STACS-93, Springer-Verlag, LNCS Vol.665 (P. Enjalbert, A. Finkel, K.W. Wagner, eds), 130-140 (1993).
2. E. Best, Th. Thielke: Coloured Nets with Curry: a multirelation-based linear algebra for high-level Petri nets. Petri Net Newsletters No. 50, 27–44 (April 1996). Slightly abrigded version: Orthogonal Transformations for Coloured Petri Nets. To appear in Proc. of Petri Net Conference'97 (Toulouse), Springer-Verlag, LNCS (P. Azéma, G. Balbo, eds) (1997).
3. R. Devillers: S-invariant Analysis of General Recursive Petri Boxes. *Acta Informatica* 32, 313-345 (1995).

4. K. Jensen: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1992).
5. W. Reisig: *Petri Nets. An Introduction.* EATCS Monographs on Theoretical Computer Science Vol. 4, Springer-Verlag (1985).