Net Refinement by Pullback Rewriting*

Renate Klempien-Hinrichs

Universität Bremen, Fachbereich 3, Postfach 33 04 40, D-28334 Bremen
email: rena@informatik.uni-bremen.de

Abstract. The theory of graph grammars is concerned with the rule-based transformation of graphs and graph-like structures. As the formalism of Petri nets is founded on a particular type of graphs, the various net refinement methods proposed for their structured design are in particular graph transformations. This paper aims at applying a recently developed technique for graph rewriting, the so-called pullback approach, to describe net refinement. The translation of this technique, which is based on (hyper)graph morphisms, into terms of net morphisms yields a well-defined mechanism closely related to pullback rewriting in hypergraphs. A variant allows to elegantly characterize a particular net refinement operation which modifies the context of the refined transition.

1 Introduction

Graph grammars have been developed as a concept to study the rule-based transformation of graphs and graph-like structures (see [Roz97] for a comprehensive overview). One can distinguish between approaches in which arbitrary subgraphs may be replaced, and approaches to rewrite elementary subgraphs, i.e. vertices, (hyper)edges, or handles. (Hyper)edge rewriting [HK87a, Hab92] is a special case of the double-pushout approach to graph rewriting [Ehr79]; it has been generalized to handle rewriting in [CER93]. With the pullback approach introduced in [Bau95a], a category theoretical framework for vertex rewriting is being developed. It is based on graph morphisms and can deal with both graphs and hypergraphs [BJ97].

A Petri net is usually defined as a bipartite graph (the underlying net structure) where a vertex is either a place or a transition, plus a marking of the places (see e.g. [Rei85]). The marking may change by the firing of transitions, thus leading to a notion of behaviour. A number of methods to refine a place or a transition – i.e. to manipulate the underlying net structure – such that the behaviour of the refined net can be inferred from the behaviour of the original and the refinement net in a compositional way may be found in the literature (for a survey see [BGV91]).

By viewing the underlying net structure of a Petri net as a hypergraph, place or transition refinement becomes the replacement of an elementary item in

* Supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems) through the University of Bordeaux I.

a hypergraph. In [HK87b] and [Vog87], it has been pointed out that hyperedge rewriting describes some types of net refinement. The operation in [GG90] modifies the context of the refined transition by multiplying the places in its pre- and postset and is thus too complex to be described by hyperedge rewriting. However, it can be seen as a special case of the vertex rewriting technique of [Kle96]. Handle rewriting has not yet been evaluated under this aspect.

Another line of research investigates rule-based refinement in the general setting of algebraic high-level nets [PER95, PGE98]. The rules which are used there have been developed from the double-pushout approach to graph rewriting of [Ehr79].

In this paper, the technique of pullback rewriting is translated into terms of net morphisms. The resulting mechanism yields a well-defined notion of net refinement and is closely related to the original pullback rewriting in hypergraphs. Furthermore, it also allows an elegant characterization of the refinement operation in [GG90]. The paper is organized as follows. Section 2 introduces the basic notions of hypergraphs and net structures. The respective categories are studied in Section 3. In Section 4, pullback rewriting in net structures is defined and compared to pullback rewriting in hypergraphs. Section 5 characterizes the net refinement technique of [GG90] in terms of pullback rewriting, and Section 6 contains some concluding remarks.

2 Hypergraphs and net structures

The basic objects considered in this paper, hypergraphs and net structures, are introduced together with the usual notions of the respective morphisms.

Definition 2.1. (Hypergraph.) A *hypergraph* $H = (V, E, src, trg)$ consists of a set V of *nodes*, a set E of *hyperedges* such that $V \cap E = \emptyset$, and two mappings $src, trg: E \to \mathcal{P}(V)$ assigning to every hyperedge $e \in E$ a set $src(e) \subseteq V$ of *source nodes* and a set $trg(e) \subseteq V$ of *target nodes*. Subscripts and superscripts carry over to the components of a hypergraph; for example, $H'_n = (V'_n, E'_n, src'_n, trg'_n)$.

Let H and H' be two hypergraphs. A *hypergraph morphism* $f: H \to H'$ is a pair of mappings $f = (f_V, f_E)$ with $f_V: V \to V'$, $f_E: E \to E'$ such that $f_V(src(e)) \subseteq src'(f_E(e))$ and $f_V(trg(e)) \subseteq trg'(f_E(e))$ for all $e \in E$. As usual, the subscripts V and E will be omitted in the sequel. If f is bijective and both f and f^{-1} are hypergraph morphisms, then f is a *hypergraph isomorphism*. In this case, H and H' are *isomorphic*.

Hypergraphs and hypergraph morphisms form a category which is denoted by $\mathcal{H}$.

In a drawing of a hypergraph H, a node v is represented by a circle and a hyperedge e by a square. There is an arrow from v to e if $v \in s(e)$ and an arrow from e to v if $v \in t(e)$. Thus, Fig. 1 shows a hypergraph.

A Petri net consists of a net structure plus a marking. As this paper concentrates on structural aspects, only the former notion is formally defined here; for other notions from net theory see e.g. [Rei85].

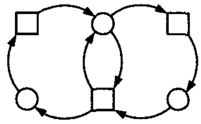

Figure 1. Drawing a hypergraph (or a net structure)

Definition 2.2. (Net structure.) A *net structure* $N = (P, T, F)$ consists of a set P of *places*, a set T of *transitions* such that $P \cap T = \emptyset$, and a *flow relation* $F \subseteq (P \times T) \cup (T \times P)$ the elements of which are called *arcs*. As for graphs, subscripts and superscripts carry over to the components of a net structure.

For an item $x \in P \cup T$, ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ denotes the *preset* of x, and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ its *postset*.

Let N and N' be two net structures. A *net morphism* $f: N \to N'$ is a mapping $f: P \cup T \to P' \cup T'$ satisfying $(f(x), f(y)) \in F'$ and $x \in P \Leftrightarrow f(x) \in P'$ for all $x, y \in P \cup T$ with $f(x) \neq f(y)$ and $(x, y) \in F$. If f is bijective and both f and f^{-1} are net morphisms, then f is a *net isomorphism* and N, N' are *isomorphic*.

Net structures and net morphisms form a category which is denoted by $\mathcal{N}$.

In a drawing of a net structure N, a place p is represented by a circle, a transition t by a square, and an arc (x, y) by an arrow. Thus, Fig. 1 shows a net structure.

The similar representation of hypergraphs and net structures evokes a one-to-one encoding: The hypergraph H is *associated* with the net structure N if $V = P$, $E = T$, $src(e) = {}^\bullet e$ and $trg(e) = e^\bullet$ for all $e \in E$. With respect to this encoding, every hypergraph morphism is associated with a net morphism. The opposite is not true: a net morphism may map a transition on a place (or vice versa). But if a substructure is mapped on one item, then its border has to be of the same type as the item (cf. Figs. 2 and 3, where a dashed line encircles the items the respective mapping identifies).

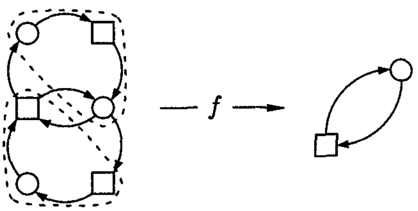

Figure 2. A net morphism without associated hypergraph morphism

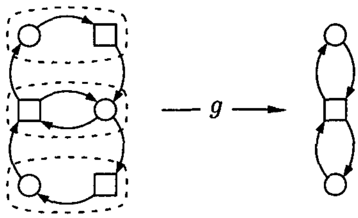

Figure 3. Neither a net morphism nor a hypergraph morphism

3 The categories of hypergraphs and net structures

In this section, the pullback construction for hypergraph morphisms is recalled. The category of hypergraphs is complete and therefore has all pullbacks. The category of net structures does not have all pullbacks, but the pairs of net morphisms for which the pullback exists are characterized, and the pullback construction is given for these cases.

As the notion of a pullback is central for pullback rewriting, the section starts with its general definition. For other concepts from category theory see e.g. [HS79].

Definition 3.1. (Pullback.) Let $\mathcal{C}$ be a category and $(f_i\colon Y_i \to Z)_{i=1,2}$ a pair of morphisms in $\mathcal{C}$. The *pullback* of $(f_i\colon Y_i \to Z)_{i=1,2}$ is another pair of morphisms $(g_i\colon X \to Y_i)_{i=1,2}$ such that $f_1 \circ g_1 = f_2 \circ g_2$, and for every pair of morphisms $(g_i'\colon X' \to Y_i)_{i=1,2}$ with $f_1 \circ g_1' = f_2 \circ g_2'$ there is a unique morphism $h\colon X' \to X$ with $g_i \circ h = g_i'$ for $i = 1, 2$.

Using a definition of hypergraphs as graphs structured by the smallest complete bipartite graph ○⇄□ (i.e. as objects in the comma category of graphs over ○⇄□) which is equivalent to the one given here, the following fact can be shown analogously to [BJ97].

Fact 3.2. *The category $\mathcal{H}$ is finitely complete and has, in particular, pullbacks. The pullback of a pair of hypergraph morphisms $(f_i\colon H_i \to H)_{i=1,2}$ consists of the projections $g_i\colon H_{pb} \to H_i$ with $g_i((x_1, x_2)) = x_i$ $(i = 1, 2)$, where H_{pb} is constructed as follows:*

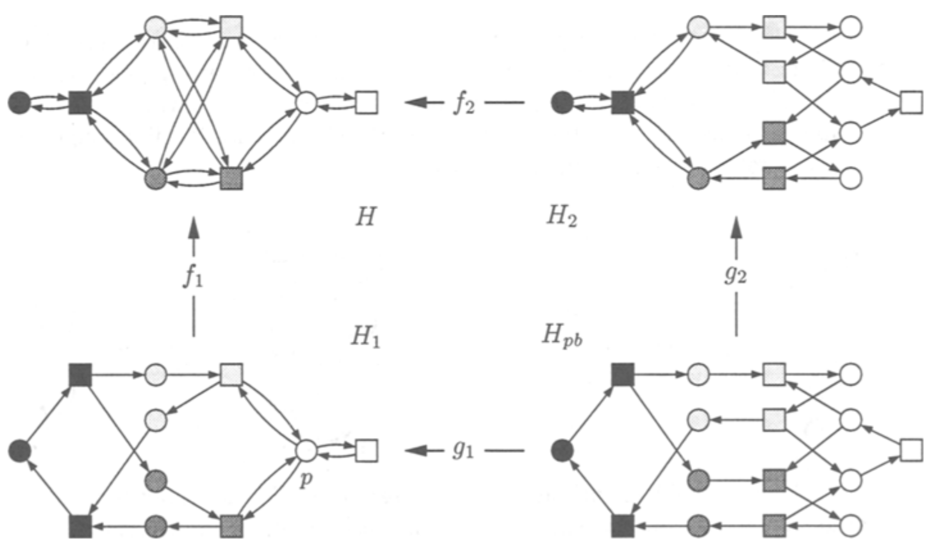

Figure 4. A pullback in $\mathcal{H}$

- $V_{pb} = \{(v_1, v_2) \in V_1 \times V_2 \mid f_1(v_1) = f_2(v_2)\}$,
- $E_{pb} = \{(e_1, e_2) \in E_1 \times E_2 \mid f_1(e_1) = f_2(e_2)\}$,
- $src_{pb}((e_1, e_2)) = \{(v_1, v_2) \in V_{pb} \mid v_1 \in src_1(e_1),\ v_2 \in src_2(e_2)\}$ *and*
- $trg_{pb}((e_1, e_2)) = \{(v_1, v_2) \in V_{pb} \mid v_1 \in trg_1(e_1),\ v_2 \in trg_2(e_2)\}$ *for all* $(e_1, e_2) \in E_{pb}$. ◇

An example for a pullback of hypergraph morphisms f_1, f_2 is given in Fig. 4. The morphisms are indicated by the relative arrangement of the items and their shading. As explained in the next section, this pullback can be interpreted as deriving H_2 from H_1 by rewriting the node p.

Unlike $\mathcal{H}$, the category of net structures is not finitely complete, but the characterization of Theorem 3.3 allows to easily verify that pullbacks do exist in the cases which will be interpreted as net rewriting in the following section.

Theorem 3.3. *For $i = 1, 2$, let N_i and N be net structures and $f_i: N_i \to N$ net morphisms. The pullback of (f_1, f_2) exists if and only if for every item $z \in P \cup T$ of N, at most one of the sets $f_1^{-1}(z)$, $f_2^{-1}(z)$ contains distinct items x and y such that (x, y) belongs to the flow relation of the corresponding net structure.*

Proof. "$\Rightarrow$": Let $z \in P \cup T$ and $p_i, t_i \in f_i^{-1}(z)$ with $(p_i, t_i) \in F_i$ or $(t_i, p_i) \in F_i$, for $i = 1, 2$. Moreover, let $N^\star$ be a net structure and $g_1: N^\star \to N_1$, $g_2: N^\star \to N_2$ net morphisms with $g_1 \circ f_1 = g_2 \circ f_2$.

Now let N' be the net structure with places p'_1, p'_2, transitions t'_1, t'_2, and an arc between p'_i and t'_i mirroring (one of) the arc(s) between p_i and t_i $(i = 1, 2)$. Consider the two net morphisms $g'_1: N' \to N_1$, $g'_2: N' \to N_2$ with $g'_1(p'_1) = p_1$, $g'_1(\{t'_1, p'_2, t'_2\}) = \{t_1\}$, $g'_2(\{p'_1, t'_1, p'_2\}) = \{p_2\}$, and $g'_2(t'_2) = t_2$; clearly, $g'_1 \circ f_1 = g'_2 \circ f_2$. Finally, let $h: N' \to N^\star$ be a net morphism such that $g_i \circ h = g'_i$. The situation is depicted in Fig. 5.

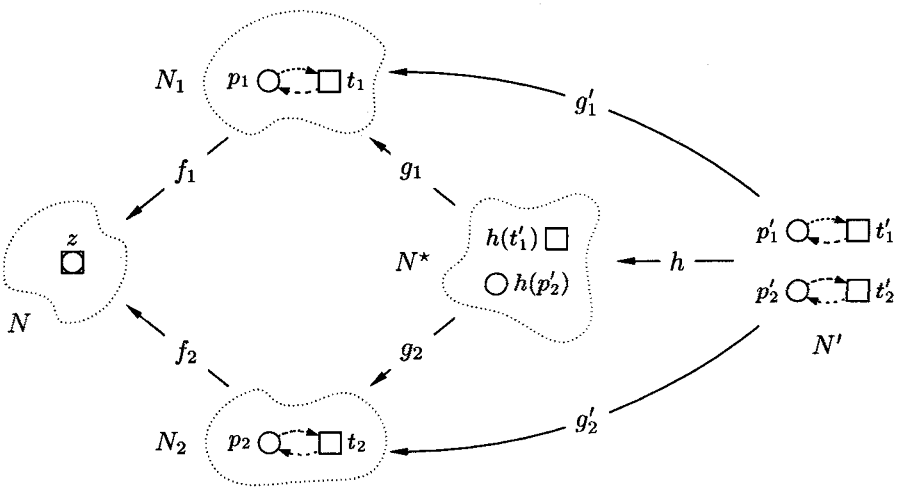

Figure 5. Illustrating the proof of Theorem 3.3

As $g_i \circ h = g'_i$ maps p'_i to p_i and t'_i to t_i, h does not identify p'_i and t'_i $(i = 1, 2)$. Therefore, the arc between p'_1 and t'_1 resp. p'_2 and t'_2 implies that $h(t'_1)$ is a transition and $h(p'_2)$ a place. Moreover, $g_i \circ h = g'_i$ identifying t'_1 and p'_2 means that g_i identifies $h(t'_1)$ and $h(p'_2)$ $(i = 1, 2)$. Hence, for net morphisms $g''_i: N' \to N_i$ with $g''_1(P' \cup T') = \{t_1\}$ and $g''_2(P' \cup T') = \{p_2\}$, the two distinct net morphisms $h_1, h_2: N' \to N^\star$ with $h_1(P' \cup T') = h(t'_1)$ and $h_2(P' \cup T') = h(p'_2)$ fulfil $g_i \circ h_j = g''_i$ $(i, j \in \{1, 2\})$. Thus, (g_1, g_2) cannot be the pullback of (f_1, f_2).

"$\Leftarrow$" (Outline): Let $Z_i := \{z \in P \cup T \mid \exists x, y \in f_i^{-1}(z) \text{ with } (x, y) \in F_i\}$ for $i = 1, 2$. By assumption, Z_1 and Z_2 are disjoint. Let N_{pb} be as follows:

- $P_{pb} = \{(x_1, x_2) \in f_1^{-1}(z) \times f_2^{-1}(z) \mid$
  $(z \in P \setminus (Z_1 \cup Z_2))$ or $(z \in Z_1$ and $x_1 \in P_1)$ or $(z \in Z_2$ and $x_2 \in P_2)\}$,
- $T_{pb} = \{(x_1, x_2) \in f_1^{-1}(z) \times f_2^{-1}(z) \mid$
  $(z \in T \setminus (Z_1 \cup Z_2))$ or $(z \in Z_1$ and $x_1 \in T_1)$ or $(z \in Z_2$ and $x_2 \in T_2)\}$,
- $F_{pb} = \{((x_1, x_2), (y_1, y_2)) \in (P_{pb} \times T_{pb}) \cup (T_{pb} \times P_{pb}) \mid$
  $(x_i, y_i) \in F_i$ and $(x_j = y_j$ or $(x_j, y_j) \in F_j)$ for $i, j \in \{1, 2\}, i \neq j\}$.

Clearly, N_{pb} is a net structure, and it is not difficult to verify that the projections $g_i: N_{pb} \to N_i$ with $g_i((x_1, x_2)) = x_i$ form the pullback of (f_1, f_2) in $\mathcal{N}$. □

4 Net rewriting by pullbacks

In this section, pullback rewriting is defined directly in the category $\mathcal{N}$ of net structures. The basic idea is to achieve the partition of a net structure into three parts – the item to be rewritten, its immediate neighbourhood, and the context of the item – by a net morphism (an *unknown*) to a special net structure (the *alphabet*). Another kind of net morphism to the alphabet (a *rule*) specifies the net structure replacing the item, and its *application* is modelled by the pullback of the two net morphisms. Thus, pullback rewriting yields a notion of net refinement where items in the pre- and postsets of the refined item can be multiplied. Example 4.1, a place refinement, illustrates the usefulness of such an operation and will be formalized as both net and hypergraph rewriting in this section.

The close relationship between pullback rewriting in net structures and in hypergraphs allows to transfer the formalism presented in [BJ97] for an arbitrary number of items to be rewritten – possibly of different types – to net structures, too. The same holds for the notion of parallel rewriting as proposed in [Bau95b].

Example 4.1. (Cf. the reduction example of [GF95].) The (marked) Petri net PN in Fig. 6 models a situation of mutual exclusion, with p as a semaphore. Its refinement to PN' explicitly represents the critical sections and the initialization of their common resources. Moreover, each transition connected with p is split in two to express the entrance into and exit from its associated critical section. ◇

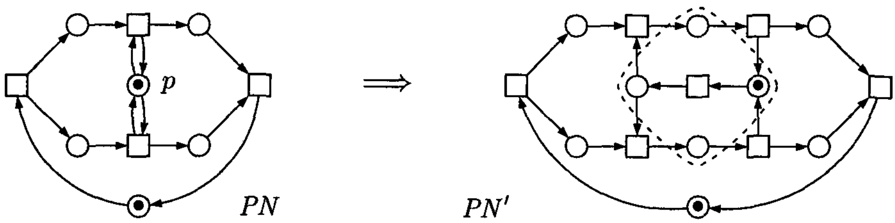

Figure 6. Refining a Petri net

Notation 4.2. For a relation $X \subseteq S \times S$ on a set S, $X^\sigma = X \cup \{(y,x) \mid (x,y) \in X\}$ denotes the symmetric hull. The set of all positive integers is denoted by $\mathbb{N}_+$.

The first mechanism to be presented is place rewriting in net structures. The place rewriting alphabet contains a place p_{-1} (for the place to be rewritten), transitions t_j linking it to neighbour places p_i, and a farther context t_0.

Definition 4.3. (Alphabet.) The *place rewriting alphabet* is the net structure N_A with $P_A = \{p_{-1}\} \cup \{p_i \mid i \in \mathbb{N}_+\}$, $T_A = \{t_0\} \cup \{t_j \mid j \in \mathbb{N}_+\}$, and

$$F_A = \bigcup_{i,j \in \mathbb{N}_+} \{(t_0, p_i), (p_i, t_j), (t_j, p_{-1})\}^\sigma.$$

A substructure $N_{A(m,n)}$ of N_A with $m+1$ places and $n+1$ transitions with $m, n \in \mathbb{N}_+$ "as required" will be used for finite examples; cf. Fig. 7 for $N_{A(2,3)}$.

A place rewriting unknown maps the place to be rewritten on p_{-1} and identifies those linking transitions resp. neighbour places which will be treated equally during a rewriting step.

Definition 4.4. (Unknown.) Let N be a net structure and $p \in P$. A *place rewriting unknown* on p is a net morphism $u_p \colon N \to N_A$ such that

- $u_p^{-1}(p_{-1}) = \{p\}$,
- for every $j \in \mathbb{N}_+$, $x \in u_p^{-1}(t_j)$ implies $\{(x,p)\}^\sigma \cap F \neq \emptyset$, and
- for every $i \in \mathbb{N}_+$, $y \in u_p^{-1}(p_i)$ implies that $j \in \mathbb{N}_+$ and $t \in u_p^{-1}(t_j)$ exist with $\{(y,t)\}^\sigma \cap F \neq \emptyset$.

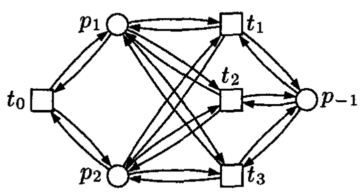

Figure 7. The place rewriting alphabet $N_{A(2,3)}$

A place rewriting rule maps what would classically be called the right-hand side of a production on p_{-1} and fixes its possible connexions to a context through the inverse images of the t_j.

Definition 4.5. (Rule.) A net morphism $r: N_R \to N_A$ is a *place rewriting rule* if

- for every item $x \in \{t_0\} \cup \{p_i \mid i \in \mathbb{N}_+\}$, $r^{-1}(x)$ contains exactly one element,
- $\{(r^{-1}(t_0), r^{-1}(p_i)) \mid i \in \mathbb{N}_+\}^\sigma \subseteq F_R$, and
- for every $j \in \mathbb{N}_+$, $r^{-1}(t_j)$ contains only transitions.

The notions of a rule application and a rewriting step are defined uniformly for all the concrete rewriting mechanisms studied in this and the next section.

Definition 4.6. (Rule application, rewriting step.) Let $\mathcal{C}$ be a category with an alphabet object A, an unknown morphism $u_x: Y \to A$, and a rule morphism $r: R \to A$ such that A, u_x, and r belong to the same rewriting mechanism (e.g. place rewriting in $\mathcal{N}$). The *application* of r at u_x is the pullback of (u_x, r) in $\mathcal{C}$. If Y' is the object constructed by the application of r at u_x (the *derived* object), then $Y \Longrightarrow_{(u_x, r)} Y'$ denotes a *rewriting step.*

Figure 8 formalizes the refinement $PN \Longrightarrow PN'$ of Example 4.1 as the place rewriting step $N \Longrightarrow_{(u_p, r)} N'$. The unknown u_p distinguishes the "upper" from the "lower" context of p, and the rule r specifies the net structure replacing p as well as the splitting of the transitions connected with p. Note that there are alternative choices for u_p and r to derive N' from N.

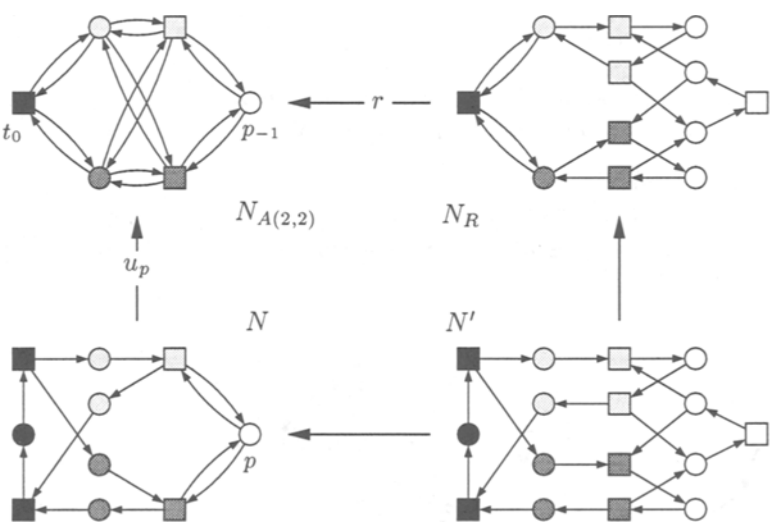

Figure 8. Formalizing Example 4.1 as pullback rewriting

In general, the application of a place rewriting rule r at a place rewriting unknown u_p produces in the derived net structure exactly one copy of the context $u_p^{-1}(t_0)$ of p. Similarly, the $u_p^{-1}(p_i)$ are reproduced, as is the right-hand side of the rule. Only the linking transitions may be multiplied (the factors being the size of the respective inverse images) and have their arcs of the flow relation altered.

Corollary 4.7. *For every place rewriting rule r and unknown u_p, the application of r at u_p is defined.* $\diamond$

Proof. Of N_A, only the item t_0 (resp. p_{-1}) may contain an arc in its inverse image under u_p (resp. r). As $t_0 \neq p_{-1}$, Theorem 3.3 implies the assertion. $\square$

There is a close relationship between place rewriting in $\mathcal{N}$ and node rewriting in $\mathcal{H}$, which differs from that introduced in [BJ97] only in that it deals with directed instead of undirected hypergraphs. Thus, the notions of an alphabet, an unknown, and a rule can be gained from those for place rewriting in $\mathcal{N}$ by changing the (terminal) substructures t_0, p_{-1} of N_A and their inverse images $r^{-1}(t_0), u_p^{-1}(p_{-1})$ into copies of the (terminal) hypergraph , and adjusting the involved net morphisms u_p and r accordingly to hypergraph morphisms $\langle u_p \rangle$ and $\langle r \rangle$. Figure 4 shows how the place rewriting step $N \Longrightarrow_{(u_p,r)} N'$ of Fig. 8 is transformed into the node rewriting step $H \Longrightarrow_{(\langle u_p \rangle, \langle r \rangle)} H'$, where $H = H_1$, $H' = H_{pb}$, $\langle u_p \rangle = f_1$, and $\langle r \rangle = f_2$. The example may be explicit enough so that the formal definitions can be omitted. It also illustrates that for the formalization of net refinement, pullback rewriting in net structures is more adequate than pullback rewriting in hypergraphs: In the latter case, one cannot directly take the hypergraph associated with the net to be refined, but has to alter it in order to get the desired result.

Proposition 4.8. *Let u_p be a place rewriting unknown and r a place rewriting rule. If $N \Longrightarrow_{(u_p,r)} N'$ and $H \Longrightarrow_{(\langle u_p \rangle, \langle r \rangle)} H'$, then H' is isomorphic to the hypergraph associated with N'.*

To end this section, consider briefly a variant of place rewriting allowing a rule $r: N_R \to N_A$ to map places as well as transitions of N_R on the transitions t_i of N_A. (With the concepts of [Bau95b], this can be interpreted as a parallel rewriting step.) The application of such a rule to an unknown is still defined and results in the multiplication of the induced substructures of N_R. The idea is illustrated in Fig. 9 by an adaptation of Example 4.1; note how much the rule and its application gain in clarity. Moreover, the same rule can be applied to a net modelling an arbitrary number of processes which share a common resource.

5 A particular net refinement technique

By the symmetry of net structures, pullback rewriting of places immediately implies a notion of transition rewriting. In this section, a slightly different instance

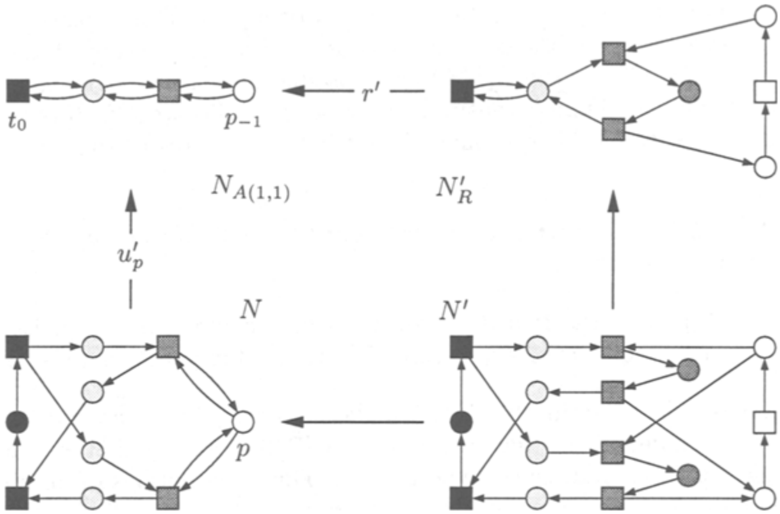

Figure 9. Application of a more general rewriting rule

of pullback rewriting is used to characterize the transition refinement operation introduced in [GG90] for one-safe nets. Their operation allows to infer the behaviour (in particular liveness properties) of a refined net compositionally from the behaviours of the original and the refinement net, and in contrast to previous studies their refinement nets may display initial or terminal concurrency.

Markings and behavioural aspects are not formally considered here; this concerns in particular some additional restrictions for refinement structures.

Notation 5.1. Let N be a net structure. The set ${}^{\circ}N = \{x \in P \mid {}^{\bullet}x = \emptyset\}$ contains the *initial places* of N, and $N^{\circ} = \{x \in P \mid x^{\bullet} = \emptyset\}$ its *terminal places.*

General assumption [GG90]. In this section, all net structures N are assumed to have arcs $(p,t), (t,p') \in F$ and ${}^{\bullet}t \cap t^{\bullet} = \emptyset$ for every $t \in T$.

Definition 5.2. (Refinement structure, cf. [GG90].) A net structure N_R is a *refinement structure* if ${}^{\circ}N_R \neq \emptyset \neq N_R{}^{\circ}$ and ${}^{\circ}N_R \cap N_R{}^{\circ} = \emptyset$.

Figure 10 shows a refinement structure N_R with initial places $(a), (b)$ and terminal place (e).

Definition 5.3. (Net refinement [GG90].) Let N_1 be a net structure and $t \in T_1$. Moreover, let N_R be a refinement structure (disjoint from N_1). Then the refined net structure $N_2 = N_1[N_R/t]$ is defined by

- $P_2 := (P_1 \setminus ({}^{\bullet}t \cup t^{\bullet})) \cup (P_R \setminus ({}^{\circ}N_R \cup N_R{}^{\circ})) \cup \mathit{Int}$,
  where $\mathit{Int} := ({}^{\bullet}t \times {}^{\circ}N_R) \cup (t^{\bullet} \times N_R{}^{\circ})$,
- $T_2 := (T_1 \setminus \{t\}) \cup T_R$, and

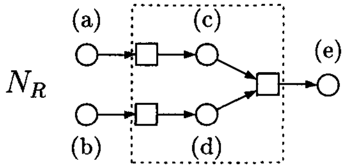

Figure 10. A refinement structure [GG90]

– $F_2 := ((F_1 \cup F_R) \cap (P_2 \times T_2 \cup T_2 \times P_2))$
$\cup \{((p_1,p_2),t_1) \mid (p_1,p_2) \in \mathit{Int},\ t_1 \in T_1 \setminus \{t\},\ (p_1,t_1) \in F_1\}$
$\cup \{(t_1,(p_1,p_2)) \mid (p_1,p_2) \in \mathit{Int},\ t_1 \in T_1 \setminus \{t\},\ (t_1,p_1) \in F_1\}$
$\cup \{((p_1,p_2),t_2) \mid (p_1,p_2) \in \mathit{Int},\ t_2 \in T_R,\ (p_2,t_2) \in F_R\}$
$\cup \{(t_2,(p_1,p_2)) \mid (p_1,p_2) \in \mathit{Int},\ t_2 \in T_R,\ (t_2,p_2) \in F_R\}$.

Figure 11 illustrates the refinement of a transition t with the refinement structure N_R of Fig. 10: For every preplace p of t in N_1 and every initial place p' in N_R, there is a new place (p,p') in N_2 with ingoing arcs from each transition in the preset of p and outgoing arcs to each transition in the postsets of p and p', and analogously for the postplaces of t and the terminal places in N_R.

This refinement technique can be characterized by pullback rewriting as follows.

Definition 5.4. (Refinement alphabet, unknown, and rule.) The *refinement alphabet* is the net structure N_α with $P_\alpha = \{p_1,p_2\}$, $T_\alpha = \{t_0,t_{-1}\}$, and $F_\alpha = \{(t_0,p_1),(t_0,p_2)\}^\sigma \cup \{(p_1,t_{-1}),(t_{-1},p_2)\}$.

Let N_1 be a net structure and $t \in T_1$. The *refinement unknown* on t is the net morphism $u_t: N_1 \to N_\alpha$ with $u_t^{-1}(t_{-1}) = \{t\}$, $u_t^{-1}(p_1) = {}^\bullet t$, and $u_t^{-1}(p_2) = t^\bullet$.

Let N_R be a refinement structure and N'_R a net structure with $P'_R = P_R$, $T'_R = T_R \dot{\cup} \{t'\}$, and $F'_R = F_R \cup \{(p,t') \mid p \in {}^\circ N_R \cup N_R{}^\circ\}^\sigma$. The *refinement rule* induced by N_R is the net morphism $r: N'_R \to N_\alpha$ with $r^{-1}(t_0) = \{t'\}$, $r^{-1}(p_1) = {}^\circ N_R$, and $r^{-1}(p_2) = N_R{}^\circ$.

The conversion of the example above into terms of pullback rewriting is depicted in Fig. 12. Note that the flow relation of N_α is not symmetric. Moreover,

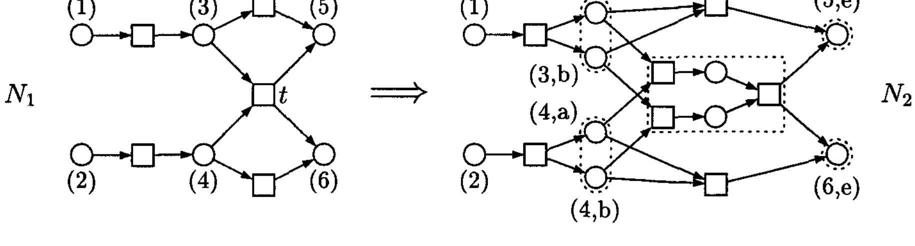

Figure 11. Transition refinement [GG90]

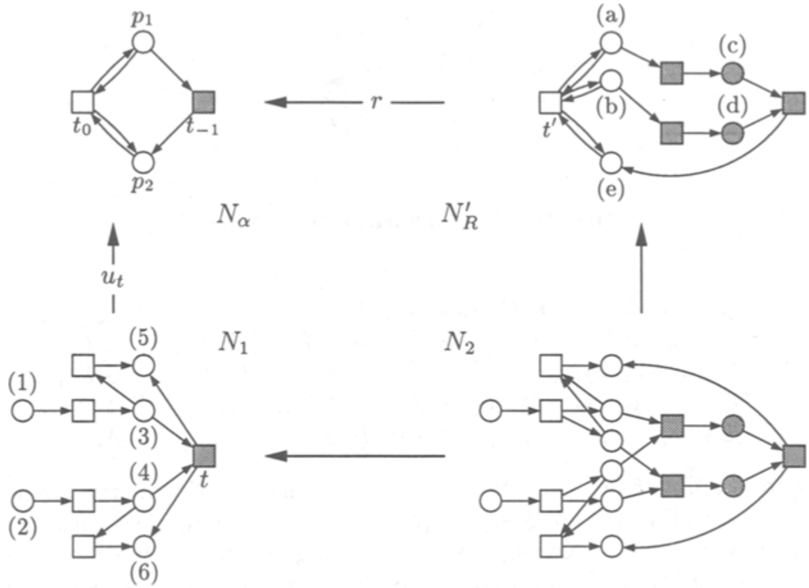

Figure 12. Transition refinement as pullback rewriting

the refinement unknown u_t is a mapping (by the assumption above), and unique for every transition t of a net structure N_1.

Theorem 5.5. *Let N_R be a refinement structure, r the induced refinement rule, N_1 a net structure with $t \in T_1$, and $u_t: N_1 \to N_\alpha$ the refinement unknown on t. If $N_1 \Rightarrow_{(u_t,r)} N_2$, then N_2 and $N_1[N_R/t]$ are isomorphic.*

Proof. By construction, N_2 and $N_1[N_R/t]$ only differ in that N_2 contains an item (x, t') for each $x \in (P_1 \cup T_1) \setminus ({}^\bullet t \cup \{t\} \cup t^\bullet)$ and an item (t, y) for each $y \in (P_R \setminus ({}^\circ N_R \cup N_R{}^\circ)) \cup T_R$. □

Note that the canonical vicinity respecting morphism $f: N_1[N_R/t] \to N_1$ of [GG90] is (modulo isomorphism) exactly the morphism $f: N_2 \to N_1$ generated by the pullback construction.

6 Conclusion

The aim of this work was to investigate an application of the pullback approach to hypergraph transformation by translating the notion of pullback rewriting from terms of hypergraph morphisms into terms of net morphisms. It turned out that unlike the category of hypergraphs, the category of net structures is not complete; in particular, it does not have all pullbacks. Nevertheless, there is an easily verified criterion to determine whether the pullback of two given net morphisms exists. This criterion ensures that net rewriting by pullbacks is indeed well-defined. Moreover, the net refinement operation of [GG90] has a concise characterization in the pullback rewriting approach.

There are two main areas for future research on the issues presented here.

On the one hand, pullback rewriting has been introduced but quite recently as a hypergraph rewriting approach. It already appears to be promising as an abstract framework for the known hypergraph transformation techniques. Moreover, this paper shows that the idea of pullback rewriting in net structures has a meaningful interpretation as net refinement. So, the pullback rewriting approach needs further development.

On the other hand, the relationship between hypergraph transformations and net refinements (or, conversely, net reductions) should be investigated:

As a number of refinement operations correspond to rather restricted types of context-free hypergraph rewriting mechanisms, interpreting more general types of hypergraph rewriting as net refinement will probably lead to new net refinements. Moreover, the well-known results on compatible properties may lead to similar results for net refinement, i.e. to results on the compositionality of net properties. In the setting of high-level nets and refinements based on double-pushout rules, similar ideas have already been investigated in [PER95, PGE98]; the link to the work presented here remains to be established.

Vice versa, finding adequate descriptions of particular types of net refinement as hypergraph rewriting may also lead to extensions of the latter.

Acknowledgement. I thank Annegret Habel and two anonymous referees for their valuable comments on previous versions of this paper. The pictures have been concocted with Frank Drewes's $\LaTeX 2_\varepsilon$ package for typesetting graphs. Special thanks go to Anne Bottreau for her timely email.

References

Bau95a. Michel Bauderon. A uniform approach to graph rewriting: the pullback approach. In *Graph-Theoretic Concepts in Computer Science*, volume 1017 of *Lecture Notes in Computer Science*, 101–115, 1995.

Bau95b. Michel Bauderon. Parallel rewriting of graphs through the pullback approach. In *Proc. SEGRAGRA'95*, volume 2 of *Electronic Notes in Theoretical Computer Science*, 8 pages, 1995.

BGV91. Wilfried Brauer, Robert Gold, and Walter Vogler. A survey of behaviour and equivalence preserving refinements of Petri nets. In *Advances in Petri Nets*, volume 483 of *Lecture Notes in Computer Science*, 1–46, 1991.

BJ97. Michel Bauderon and Hélène Jacquet. Node rewriting in hypergraphs. In *Graph-Theoretic Concepts in Computer Science*, volume 1197 of *Lecture Notes in Computer Science*, 31–43, 1997.

CER93. Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46:218–270, 1993.

Ehr79. Hartmut Ehrig. Introduction to the algebraic theory of graph grammars. In *Graph-Grammars and Their Application to Computer Science and Biology*, volume 73 of *Lecture Notes in Computer Science*, 1–69, 1979.

GF95. Anja Gronewold and Hans Fleischhack. Computing Petri net languages by reductions. In *Fundamentals of Computation Theory*, volume 965 of *Lecture Notes in Computer Science*, 253–262, 1995.

GG90. Rob van Glabbeek and Ursula Goltz. Refinement of actions in causality based models. In *Stepwise Refinement of Distributed Systems*, volume 430 of *Lecture Notes in Computer Science*, 267–300, 1990.

Hab92. Annegret Habel. Hypergraph grammars: Transformational and algorithmic aspects. *Journal of Information Processing and Cybernetics EIK*, 28:241–277, 1992.

HK87a. Annegret Habel and Hans-Jörg Kreowski. Characteristics of graph languages generated by edge replacement. *Theoretical Computer Science*, 51:81–115, 1987.

HK87b. Annegret Habel and Hans-Jörg Kreowski. May we introduce to you: Hyperedge replacement. In *Graph Grammars and Their Application to Computer Science*, volume 291 of *Lecture Notes in Computer Science*, 15–26, 1987.

HS79. Horst Herrlich and George E. Strecker. *Category Theory*. Sigma Series in Pure Mathematics. Heldermann Verlag, Berlin, 2nd edition, 1979.

Kle96. Renate Klempien-Hinrichs. Node replacement in hypergraphs: Simulation of hyperedge replacement, and decidability of confluence. In *Graph Grammars and Their Application to Computer Science*, volume 1073 of *Lecture Notes in Computer Science*, 397–411, 1996.

PER95. Julia Padberg, Hartmut Ehrig, and Leila Ribeiro. Algebraic high-level net transformation systems. *Math. Struct. in Comp. Science*, 5:217–256, 1995.

PGE98. Julia Padberg, Magdalena Gajewsky, and Claudia Ermel. Rule-based refinement of high-level nets preserving safety properties. To appear in *Proc. FASE*, *Lecture Notes in Computer Science*, 1998.

Rei85. Wolfgang Reisig. *Petri Nets*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin Heidelberg, 1985.

Roz97. Grzegorz Rozenberg, ed. *Handbook of Graph Transformations*, volume I: Foundations. World Scientific, Singapore, 1997.

Vog87. Walter Vogler. Behaviour preserving refinements of Petri nets. In *Graph-Theoretic Concepts in Computer Science*, volume 246 of *Lecture Notes in Computer Science*, 82–93, 1987.