HELSINKI UNIVERSITY OF TECHNOLOGY
DIGITAL SYSTEMS LABORATORY
Series **A:** Research Reports
No. **30**; September 1994

ISSN 0783–5396
ISBN 951–22–2283–3

ON THE FOLDING OF ALGEBRAIC NETS

JOHAN LILIUS

Digital Systems Laboratory
Department of Computer Science
Helsinki University of Technology
Otaniemi, FINLAND

Helsinki University of Technology
Department of Computer Science
Digital Systems Laboratory
Otaniemi, Otakaari 1
FIN–02150 ESPOO, FINLAND

HELSINKI UNIVERSITY OF TECHNOLOGY
DIGITAL SYSTEMS LABORATORY
Series **A**: Research Reports
No. **30**; September 1994

ISSN 0783–5396
ISBN 951–22–2283–3

On the folding of Algebraic Nets

JOHAN LILIUS*

Abstract: A folding in General Net theory is morphism that is surjective on places and transitions. Foldings can be used to relate CE-systems to "high-level system". In this paper we study the problem of relating Petri Nets to non-strict High-level Nets. We give a construction that given a morphism of Petri Nets produces an Algebraic Net that characterises the folding in a canonical way. We also prove that the construction is functorial. Then we show how the construction can be made to work on Algebraic Nets directly. Finally we discuss an application of the construction.

Keywords: concurrency, net theory, high-level nets.

*This research has been supported by the Finnish Academy, project number 8309.

Printing: TKK Monistamo; Otaniemi 1994

Helsinki University of Technology
Department of Computer Science
Digital Systems Laboratory
Otaniemi, Otakaari 1
FIN–02150 ESPOO, FINLAND

Phone: $\dfrac{90}{+358\text{–}0}$ 4511
Telex: 125 161 htkk fi
Telefax: +358–0–465 077
E-mail: lab@saturn.hut.fi

1 Introduction

General Net theory as initiated by C.A. Petri (1973) is the study of Nets and their morphisms. A folding is a special kind of morphism between nets. It has been used to relate a net to its causal behaviors, its processes, and to describe and motivate High-level Nets (Genrich, Lautenbach and Thiagarajan 1980). Recently the construction of strict High-level Nets from CE-System has been formalized and studied (Smith and Reisig 1987). In this paper we study the construction of non-strict High-level Nets from Petri Nets through foldings.

The setting for our study of foldings is the categorical theory of Petri Nets proposed by Meseguer and Montanari (1990). In their approach Petri nets (Place Transition nets) are presented as graphs with a monoidal structure on nodes. Contrary to General Net theory, where a morphism is defined as a function on the union of transitions and places, a morphism here is taken to be a pair of functions between the set of transitions and the monoid of places respectively, such that the underlying morphism is a graph morphism. Such a morphism is called a homomorphism in the context of General Net theory. Also, in the category of Petri Nets, processes defined through foldings are not taken as the fundamental definition of causal behavior of a net. In Meseguer and Montanaris approach this problem is addressed by a categorical notion known as a *free construction*. To a given net we can associate a new net, that has as new transitions all possible (parallel) transition sequences of the original net. Different notions of causal behavior are then obtained as suitable equivalence classes of arrows (Degano, Meseguer and Montanari 1989). However the inverse of folding, *unfolding* remains an important notion in the categorical theory of high-level nets (Dimitrovici, Hummert and Pétrucci 1991), because it relates a high-level net to a low-level net with in some sense identical behavior.

The main reason for the use of different high-level net formalism is that C/E or P/T-net are much to big when modeling realistic examples. The phenomenon is sometimes referred to as the football-field problem. The problem is attacked by folding transitions and places onto each other. Parts of the net that represent the same process with different "parameters" are folded. As some of the information is lost within this process, it is added as annotations to the new net. These annotations are used to represent the different "parameters" of the process. The original low-level net is reobtained through the inverse process of unfolding, by instantiating the transitions and places with the different possible parameters.

In this paper we define a notion of folding for Petri Nets as a special kind of morphism in the category of Petri Nets. We then show how to a given folding we can associate an Algebraic Net whose underlying net is isomorphic to the codomain of the folding and whose unfolding is isomorphic to the domain of the folding. It is at first not clear if this construction is functorial, but by analyzing the structure of the category of Petri Nets we are able to identify a preorder on nets that allows us to prove the functoriality.

The main contributions of this paper are:

- The formalization of the folding construction of non-strict High-level Nets from Petri Nets.
- The identification of a preorder on nets in the category of Petri Nets.
- A generalization of the folding construction to Algebraic nets.
- The extension of Findlows "refolding" construction to Algebraic Nets.

The paper is structured as follows. We first define the categories of Petri Nets and Algebraic nets. We also define several functors between these. Then we define the notion of folding in the category of Petri Nets and show how it gives rise to an Algebraic net. This construction is proved to be functorial. We then show how we can define a notion of folding directly on the algebraic net. Finally we discuss how the construction can be used to transform a net so that is has a deadlock preserving skeleton.

2 The Categories $\underline{\mathsf{PetriG}}$ and $\underline{\mathsf{AN}}$

The aim of this section is to briefly review the basic definitions used in section 3. Most of the definitions and results have appeared before (see eg. (Dimitrovici et al. 1991, Lilius 1991)). We will define the category of Petri Nets, the category of Algebraic Nets and some functors between these.

As the category of low-level nets we use a modified version of the category of Petri nets originally introduced by Meseguer and Montanari (1990). The difference is in the notion of morphism. Meseguer and Montanari (1990) defines the map on places as a monoid homomorphism. Below it is the free extension of a $\underline{\mathsf{Set}}$-function.

2.1 Definition
The *category of Petri nets* $\underline{\mathsf{PetriG}}$, has as objects tuples $\langle T, P, \iota, o\rangle$ where T, P are the sets of transitions and places and $\iota, o : T \to P^{\otimes}$ are the input- and output-weight functions. A morphism $h : N \to N'$ in $\underline{\mathsf{PetriG}}$ is a pair of

functions $h = \langle h_T, h_P \rangle$, such that the diagrams

$$\begin{tikzcd} T \arrow[r, "\iota"] \arrow[d, "h_T"'] & P^{\otimes} \arrow[d, "h_P^{\otimes}"] \\ T' \arrow[r, "\iota'"] & P'^{\otimes} \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "o"] \arrow[d, "h_T"'] & P^{\otimes} \arrow[d, "h_P^{\otimes}"] \\ T' \arrow[r, "o'"] & P'^{\otimes} \end{tikzcd}$$

commute. □

The category PetriG can be seen as the image of the free functor from the category of graphs to the category of Petri nets as defined by (Meseguer and Montanari 1990). The sets T and P can be infinite, but we shall assume, that the functions h_T and h_P are recursive[1].

Algebraic net specifications are obtained by annotating the net with terms from an algebraic datatype.

2.2 Definition
An *Algebraic net specification with equational signature* Σ (ANS) is a tuple

$$\langle S, \Sigma, EQ, X, T, P, \iota, o, sort \rangle$$

where:

- $\langle S, \Sigma, EQ \rangle$ is a Σ-presentation,
- X is a S-sorted set of variables,
- T is the set of transitions,
- P is the set of places,
- $\iota, o : T \to (P \times T_\Sigma)^{\otimes}$ are the input- and output-weight functions, and
- $sort : P \to S$ is the sort assignment.

□

In the sequel we will notate variables of type transition by $t, t_1, \ldots$ and variables of type term by $\hat{t}, \hat{t}_1, \ldots$.

2.3 Definition
A Σ-ANS-morphism $h : ANS \to ANS'$ is a pair $\langle h_T, h_P \rangle$ where $h_T : T \to T'$ and $h_P : P \to P'$ are functions s.t.:

$$\begin{aligned} h^{\S}(\iota(t)) &= \iota'(h_T(t)), \text{ and} \\ h^{\S}(o(t)) &= o'(h_T(t)), \end{aligned}$$

[1] We shall return to this question later on

where $h^{\S} : (P \times T_\Sigma)^{\otimes} \rightarrow (P' \times T_\Sigma)^{\otimes}$ defined by $h^{\S}(\bigotimes_{i=1}^{n} \langle p, \hat{t} \rangle) = \bigotimes_{i=1}^{n} \langle h(p), \hat{t} \rangle$, with $p \in P$, $p' \in P'$. □

2.4 Proposition
Σ-ANS's and Σ-ANS-morphisms form a category Σ-<u>ANS</u>. □

2.5 Definition
An ANS-morphism $h : ANS \rightarrow ANS'$ is tuple $h = \langle h_\Sigma, h_T, h_P \rangle$ where $h_\Sigma : \Sigma \rightarrow \Sigma'$, $h_T : T \rightarrow T'$, and $h_P : P \rightarrow P'$ s.t.:

$$\begin{array}{rcl} h_P^{\#}(\iota(t)) & = & \iota'(h_T(t)) \\ sort'(h_P(p)) & = & h_S(sort(p)) \end{array}$$

where $h_P^{\#} : (P \times T_\Sigma)^{\otimes} \rightarrow (P \times T_\Sigma)^{\otimes}$ is defined by
$h^{\#}(\bigotimes_{i=1}^{n} \langle p, \hat{t} \rangle) = (\bigotimes_{i=1}^{n} \langle h_P(p), h_\Sigma^{\#}(\hat{t}) \rangle)^2$. □

2.6 Proposition
ANS's and ANS-morphisms form the category <u>ANS</u>. □

An ANS defines a *scheme* of nets. To give an interpretation of the specification we need to add a specific interpretation of the abstract datatype to our definition. This gives us the notion of an algebraic net.

2.7 Definition
An *Algebraic net* AN is a pair $\langle ANS, A \rangle$ where ANS is an algebraic net specification with signature Σ and A is a Σ-algebra. □

To define the *unfolding* of an algebraic net we need the following auxiliary definitions:

2.8 Definition
Let $\hat{t} = \hat{t}_1, \ldots, \hat{t}_n \in T_\Sigma^{\otimes}(X)$.Given $\mathrm{ass}_A^{\#} : T_\Sigma(X) \rightarrow A$, define $\mathrm{ass}_A^{*} : (P \times T_\Sigma)^{\otimes} \rightarrow (P \times A)^{\otimes}$ by

$$\mathrm{ass}_A^{*}(\bigotimes_{i=1}^{n} \langle p, \hat{t} \rangle) = \bigotimes_{i=1}^{n} \langle p, \mathrm{ass}_A^{\#}(\hat{t}) \rangle \ .$$

□

2.9 Definition
Given an algebraic $AN = \langle ANS, A \rangle$ we can define a Petri net

$$\mathsf{Unf}(\langle ANS, A \rangle) = \langle T_{\mathsf{Unf}}, P_{\mathsf{Unf}}, \iota_{\mathsf{Unf}}, o_{\mathsf{Unf}} \rangle$$

as follows:

[2] Consult the appendix for some auxiliary notations.

- $P_{\mathsf{Unf}} = \bigcup_{p \in P} p \times A_{sort(p)}$,
- $T_{\mathsf{Unf}} = \{\langle t, \mathrm{ass}_A^{\#}\rangle \,|\, \mathrm{ass}_A \in [Var(t) \to A], t \in T\}$,
- $\iota_{\mathsf{Unf}}(\langle t, \mathrm{ass}_A^{\#}\rangle) = \mathrm{ass}_A^{*}(\iota(t))$,
- $o_{\mathsf{Unf}}(\langle t, \mathrm{ass}_A^{\#}\rangle) = \mathrm{ass}_A^{*}(o(t))$.

$Var(t) = FV(\pi_2(\iota(t)))$ □

2.10 Definition
Given algebraic nets $AN = \langle ANS, A\rangle$ and $AN' = \langle ANS', A'\rangle$, an AN-morphism $h : AN \to AN'$ is a pair $\langle h_{ANS}, h_A\rangle$, where $h_{ANS} : ANS \to ANS'$ is an ANS-morphism, and $h_A : A \to A'|_{h_\Sigma}$ is a Σ-algebra morphism. □

2.11 Proposition
AN's and AN-morphisms form a category $\underline{\mathrm{AN}}$. □

2.12 Remark
$\underline{\mathrm{AN}}$ is an indexed category $\mathsf{AN} : \underline{\mathsf{AlgSig}}^{op} \to \underline{\mathrm{CAT}}$. □

2.13 Theorem
Given an AN-morphism $h : AN \to AN'$ there exists a morphism $\mathsf{Unf}(f) : \mathsf{Unf}(AN) \to \mathsf{Unf}(AN')$, ie. $\mathsf{Unf} : \underline{\mathrm{AN}} \to \underline{\mathrm{PetriG}}$ is a functor.

Proof:

1. $\mathsf{Unf}(f)$ is given by:

$$\begin{array}{rl} \mathsf{Sem}(f)_P : & P_{\mathsf{Sem}(AN)} \to P_{\mathsf{Sem}(AN')} : \\ & \langle p, a\rangle \mapsto \langle h_P(p), h_A(a)\rangle \\ \mathsf{Sem}(f)_T : & T_{\mathsf{Sem}(AN)} \to T_{\mathsf{Sem}(AN')} : \\ & \langle t, \mathrm{ass}_A^{\#}\rangle \mapsto \langle h_T(t), h_A(\mathrm{ass}_A^{\#})\rangle\ . \end{array}$$

2. That it is a $\underline{\mathrm{PetriG}}$-morphism is shown by the following tedious but straight-forward calculation:

$$\begin{array}{ll} & \mathsf{Sem}(f)_P^{\otimes}(\iota_{\mathsf{Sem}(AN)}\langle t, \mathrm{ass}_A^{\#}\rangle)) \\ = & \mathsf{Sem}(f)_P^{\otimes}(\mathrm{ass}_A^{*}(\iota(t))) \\ = & \mathsf{Sem}(f)_P^{\otimes}(\bigotimes_{i=1}^{n}\langle \pi_1(\iota(t)_i), \mathrm{ass}_A^{\#}(\pi_2(\iota(t)_i))\rangle) \\ = & \bigotimes_{i=1}^{n}\langle h_P(\pi_1(\iota(t)_i)), h_A(\mathrm{ass}_A^{\#}(\pi_2(\iota(t)_i)))\rangle \\ = & \mathrm{ass}_{A'}^{*}(\bigotimes_{i=1}^{n}\langle h_P(\pi_1(\iota(t)_i)), \pi_2(\iota(t)_i)\rangle) \end{array}$$

$$
\begin{aligned}
&= \mathrm{ass}^*_{A'}(h^{\S}(\bigotimes_{i=1}^{n}\langle \pi_1(\iota(t)_i), \pi_2(\iota(t)_i)\rangle)) \\
&= \mathrm{ass}^*_{A'}(h^{\S}(\iota(t))) \\
&= \mathrm{ass}^*_{A'}(\iota'(h_T(t))) \\
&= \iota_{\mathsf{Sem}(AN')}(\langle h_T(t), h_A(\mathrm{ass}^{\#}_A)\rangle) \\
&= \iota_{\mathsf{Sem}(AN')}(\mathsf{Sem}(f)_T(\langle t, \mathrm{ass}^{\#}_A\rangle))
\end{aligned}
$$

The above depends on the fact that $\mathrm{ass}^{\#}_{A'} = \mathrm{ass}^{\#}_{A}; h'_A$. The part for o is analogous.

□

2.14 Definition
The *underlying net* functor $\mathsf{U_N} : \underline{\mathsf{AN}} \to \underline{\mathsf{PetriG}}$ maps and algebraic net to is underlying Petri net. It is given by the assignment $\mathsf{U_N}(AN) = \langle T_U, P_U, \iota_U, o_U\rangle$ with

- $T_U = T$,
- $P_U = P$,
- $\iota_U(t) = \iota(t); \mathrm{ass}^*_{\perp}$
- $o_U(t) = o(t); \mathrm{ass}^*_{\perp}$

□

2.15 Proposition
The morphism $f : \mathsf{Unf}(AN) \to \mathsf{U_N}(AN)$ is unique.
Proof:
Follows from the fact that $\mathsf{U_N}(\langle ANS, A\rangle) \simeq \mathsf{Unf}(\langle ANS, \perp\rangle)$ and that $\perp$ is the terminal algebra. □

We can also define a functor $\mathsf{G} : \underline{\mathsf{PetriG}} \to \underline{\mathsf{AN}}$, that maps each $\underline{\mathsf{PetriG}}$-net N to the corresponding $\underline{\mathsf{AN}}$-net $G(N)$.

2.16 Definition
The functor $\mathsf{G} : \underline{\mathsf{PetriG}} \to \underline{\mathsf{AN}}$, that views each $\underline{\mathsf{PetriG}}$-net N as an $\underline{\mathsf{AN}}$-net $G(N)$. It is given by

$$
\begin{aligned}
G(N) \;=\; & \langle\langle \{*\}, \{* : * \to *\}, \phi, T_N, P_N, \iota, o, \mathrm{sort}\rangle, \\
& \{*\}, \{* : * \to *\}\rangle
\end{aligned}
$$

where $T_G = T_N$,$P_G = P_N$, the weighting functions $\iota, o : T_N \to (P_N \times \{*\})$ are by the equations

$$
\begin{aligned}
\iota_G(t) &= \pi_1(\iota(t)) \\
o_G(t) &= \pi_1(o(t)) \; .
\end{aligned}
$$

□

2.17 Lemma
$\mathsf{Unf}(\mathsf{G}(N)) \simeq N$. □

3 Foldings

As stated earlier the aim of *General Net Theory* as introduced by C.A. Petri (1973) is to study morphisms of Nets. Foldings play a central role in this theory, because they are used to relate different nets and net-classes. In this paper we are interested in the use of foldings to relate different net-classes. A study of foldings using the classical notion of net-morphism has been done by Smith and Reisig (1987). In this section we will define the notion of folding within the categories PetriG and AN.

Thus the material in this section differs from the material presented by Smith and Reisig (1987) in two respects. First, the notion of morphism is different. The classical net-morphism (Petri 1973) allows a map to take transitions to places, while morphisms in PetriG are essentially graph-morphism, and thus respects places and transitions. Although Smith and Reisig (1987) use place and transition respecting morphisms, our morphisms is required to preserve arc-weights in addition. Second, Smith and Reisig (1987) discuss foldings of CE-systems into strict High-level Nets. In this paper we take a more concrete view of the process and give a construction to a specific class of High-level Nets. Also the construction is given for non-strict nets, thus answering a question left open by Smith and Reisig (1987).

The first thing we need to do is define the notion of folding. Since the intuition in the formation of a High-level Net is that we are describing several identical processes with the same underlying net through one net it seems plausible to take the following as the definition.

3.1 Definition
A morphism $f : N_1 \to N_2$ in PetriG is a *folding* iff both f_T and f_P are surjective. □

3.2 Lemma
If $f : N_1 \to N_2$ is a folding then

$$N_1/\ker f \simeq N_2 \ .$$

□

One of the advantages of working within the category PetriG is that the notion of surjection can readily be transfered from the category Set.

3.3 Proposition
Given an AHL-net AN the assignment

$$\begin{aligned} f_P(\langle p, a\rangle) &= \langle p, \perp\rangle \\ f_T(\langle t, \mathrm{ass}^{\#}_A\rangle) &= \langle t, \mathrm{ass}^{\#}_{\perp}\rangle \end{aligned}$$

defines a folding $f : \mathsf{Unf}(AN) \to \mathsf{U}(AN)$.

Proof:
The fact that f_P and f_T are surjective follows directly from the definition. That f is a <u>PetriG</u> morphism is also clear. □

3.4 Proposition
Foldings define a natural transformation $\phi : \mathsf{Unf} \to \mathsf{U}$.

Proof:
Evident from the fact that the diagram

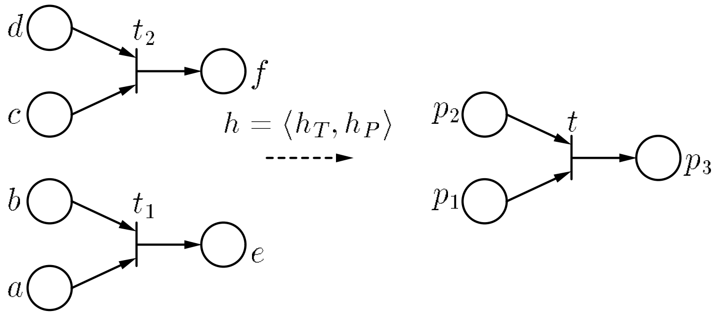
$$\begin{tikzcd} \mathsf{Unf}(AN) \arrow[r, "\phi(AN)"] \arrow[d, "\mathsf{Unf}(h)"'] & \mathsf{U}(AN) \arrow[d, "\mathsf{U}(h)"] \\ \mathsf{Unf}(AN') \arrow[r, "\phi(AN')"] & \mathsf{U}(AN') \end{tikzcd}$$

commutes. □

Thus each Algebraic Net defines a unique folding. It is now natural to ask the converse question: Given a folding $f : N_1 \to N_2$ in <u>PetriG</u> does there exists an AHL-net AN s.t. $\mathsf{Unf}(AN) \simeq N_1$ and $\mathsf{U}(AN) \simeq N_2$? The answer to this question is positive; the construction however is rather non-trivial.

Let us approach the question with an example. Consider the nets in figures 1 and 2. The map in figure 1 folds two independent transitions onto the same transition. The folded net can simulate the behavior of the original net, by having multiple markings on the places p_1, and p_2, but the information of which specific transition (t_1 or t_2) is fired is lost.

This information can be recovered by coding it into the annotation functions. Figure 2 gives one possible annotation for N_2. Its main virtue is that $\mathsf{Unf}(AN) \simeq N_1$. The idea is that for each equivalence class in $[p] \in P/\ker h$ and $[t] \in T/\ker h$ we construct an operation $f_{[t],[p]} : [t] \to [p]$ whose definition is such that the transition firing in $\mathsf{Unf}(AN)$ corresponding to $t_1 : a \otimes b \to e$ is fired with mode $\{x \leftarrow t_1\}$. The original behavior is thus recoverable. This suggests that there is a simple and straightforward way of coding the annotation directly from a given folding $f : N_1 \to N_2$. This is essentially the construction proposed by Smith and Reisig (1987).



$$
\begin{array}{lccc}
h_T: & t_1 & \mapsto & t \\
 & t_2 & \mapsto & t \\
h_P: & a & \mapsto & p_1 \\
 & b & \mapsto & p_2 \\
 & c & \mapsto & p_1 \\
 & d & \mapsto & p_2 \\
 & e & \mapsto & p_3 \\
 & f & \mapsto & p_3
\end{array}
$$

Figure 1: A folding $h : N_1 \to N_2$ in <u>PetriG</u>.

S : $\{p_1, p_2, p_3, t\}$

Σ :

$a :\to p_1$
$b :\to p_2$
$c :\to p_1$
$d :\to p_2$
$e :\to p_3$
$f :\to p_3$
$t_1 :\to t$
$t_2 :\to t$
$f_{t,p_1} : t \to p_1$
$f_{t,p_2} : t \to p_2$
$f_{t,p_3} : t \to p_3$

EQ :

$f_{t,p_1}(t_1) = a$
$f_{t,p_1}(t_2) = c$
$f_{t,p_2}(t_1) = b$
$f_{t,p_2}(t_2) = d$
$f_{t,p_3}(t_1) = e$
$f_{t,p_3}(t_2) = f$

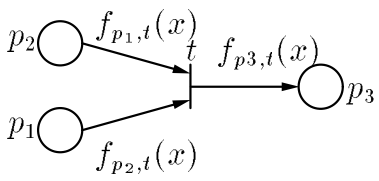
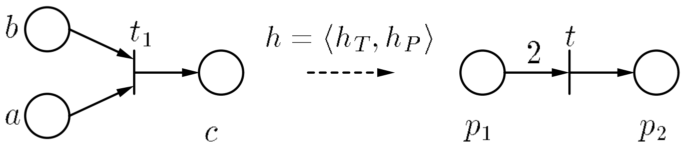


Figure 2: The AN-net corresponding to the folding in figure 1.



$$\begin{array}{llcl} h_T: & t_1 & \mapsto & t \\ h_P: & a & \mapsto & p_1 \\ & b & \mapsto & p_1 \\ & c & \mapsto & p_2 \end{array}$$

Figure 3: A folding $h' : N_1 \to N_2$ in PetriG.

However, the problem is more complicated. The construction as outlined above does not work if we fold several input-places of a transition into one. Consider the folding in figure 3. If we want to construct the annotation of the corresponding AN-net, we cannot simply take $f_{[t],[p_1]} : [t] \to [p_1]$ with $f_{[t],[p_1]}(t) = a \otimes b$, because $a \otimes b$ is not a well formed term in the domain $[p_1]$. But if we construct two operations $f_{[t],a}$ and $f_{[t],b}$ with $f_{[t],a}(t) = a$ and $f_{[t],b}(t) = b$ then the input-weight function ι has the definition $\iota(t)(p_1) = f_{[t],a}(x) \otimes f_{[t],b}(x)$. This construction is formalized in the following theorem.

3.5 Theorem
Given nets $N_1 = \langle P_1, T_1, \iota_1, o_1 \rangle$, $N_2 = \langle P_2, T_2, \iota_2, o_2 \rangle$ and a folding $f : N_1 \to N_2$ in PetriG there exists an AHL-net AN s.t. $\mathsf{Unf}(AN) \simeq N_1$ and $\mathsf{U}(AN) \simeq N_2$.

Proof:
The proof is done in three stages. First we construct AN, then the isomorphism $\mathsf{Unf}(AN) \simeq N_1$, and finally the isomorphism $\mathsf{U}(AN) \simeq N_2$.

1. Let $AN = \langle ANS, A \rangle$ where $ANS = \langle S, \Sigma, EQ, X, T, P, \iota, o, \mathrm{sort} \rangle$ with S, P, T, X, Σ, EQ, ι, o and sort constructed as follows.
   (a) $P = P_2$, $T = T_2$, $X = T_1$, and $S = P_2 \bigcup T_2$, where each sort $[t], [p]$ is inhabited by the elements of the equivalence relation.
   (b) sort is $\mathbf{id}_S \restriction P_2$.
   (c) To define the set of operator symbols, we first need to establish the domains and codomains of the operators. To this end we define functions $\delta_\iota : P_2 \times T_1 \to P_1$ and $\delta_o : P_2 \times T_1 \to P_1$ given by
   $$\delta_\iota([p], t) = \gamma(\iota(t)) \cap \gamma(f_P^{-1}([p]))$$
   and
   $$\delta_o([p], t) = \gamma(o(t)) \cap \gamma(f_P^{-1}([p])) \ .$$
   The functions give the set of input(output)-places of t in N_1 that are mapped to $[p]$ in N_2. For example in figure 1 $\delta_\iota(p_1, t_1) = \{a\}$, while in example 3 $\delta_\iota(p_1, t_1) = \{a, b\}$.
   Then let $n^i_{[t],[p]} = \max\{|\delta_i([p], t)| \mid t \in [t]\}$ for $i \in \{\iota, o\}$. $n^i_{[t],[p]} > 1$ if we fold several places of a transition onto one.

The set of function symbols is given by[3]

$$\begin{aligned}\Sigma_f &= \{f^i_{[t],[p],x} \mid [t] \in T, [p] \in P, \\ &\quad 1 \leq x \leq n^i_{[t],[p]}\} \\ \Sigma_p &= \{p :\to [p] \mid \forall p \in P_1\} \\ \Sigma_t &= \{t :\to [t] \mid \forall t \in T_1\} \\ \Sigma &= \Sigma_f \cup \Sigma_p \cup \Sigma_t\end{aligned}$$

If $n^i_{[t],[p]} = 1$ we omit the x. So in figure 1 we have $\Sigma = \{f^\iota_{t,p_1}, f^\iota_{t,p_2}, f^o_{t,p_3}, a, b, c, d, t_1, t_2\}$, while in figure 3 we have $\Sigma = \{f^\iota_{t,p_1,1}, f^\iota_{t,p_1,2}, f^o_{t,p_2}, a, b, c, t_1\}$.

(d) The construction of the equations is best explained by cases. If $n^i_{[t],[p]} = 0$ there is nothing to do. If $n^i_{[t],[p]} \geq 1$ the equations are defined by

$$\begin{aligned}EQ^i &= \{f^\iota_{[t],x}(t) = y \mid \forall t, y : t \in [t] \\ &\quad \wedge y \in \delta_i([p], t))\}\ .\end{aligned}$$

In figure 3 we have that $f^\iota_{t,p_1,1}(t) = a \in EQ$.

(e) Then we have

$$\iota([t]) = \bigotimes_{1 \leq x \leq n^\iota_{[t],[p]}} \langle [p], f^\iota_{[t],[p],x}\rangle,$$

and

$$o([t] = \bigotimes_{1 \leq x \leq n^o_{[t],[p]}} \langle [p], f^o_{[t],[p],x}\rangle\ .$$

(f) Finally let $A = T_\Sigma(X)$.

2. The sets of places and transitions in $\mathsf{Unf}(AN)$ are given by

$$\begin{aligned}P' &= \bigcup_{p \in P} p \times [p]\ . \\ T' &= \{\langle [t], \mathrm{ass}^\#_A\rangle \mid [t] \in T, \\ &\quad \mathrm{ass}^\#_A \in [Var([t]) \to A]\}\ .\end{aligned}$$

Now because all function symbols are of the form $f : [t] \to [p]$ and by construction $Var([t]) = h_T^{-1}([t])$, the only valid assignments in this context are of the form $\{z \leftarrow t\}$ with $t \in [t]$. The set of transitions is thus isomorphic to T_{N_1}. Analogously for places.

The isomorphism $\mathsf{Unf}(AN) \simeq N_1$ is now obtained as follows. Because $P' \simeq P$ and $T' \simeq P$ we have the two morphisms $m_P : P' \to P : \langle p, q\rangle \mapsto$

[3] In the sequel i ranges over $\{\iota, o\}$

q and $m_T : T' \to T : \langle [t], \{z \leftarrow t\}\rangle \mapsto t$ which form the morphism $\langle m_T, m_P\rangle : \mathsf{Unf}(AN) \to N_1$. It remains to show, that $\iota_{\mathsf{Sem}(AN)}; m_P = m_T; \iota_{N_1}$. As a first step we have:

$$\begin{aligned}
& m_P(\iota_{\mathsf{Sem}(AN)}(\langle [t], \mathrm{ass}^{\#}_{T_\Sigma(X)}\rangle)) \\
&= m_P(\mathrm{ass}^{*}_{T_\Sigma(X)}(\iota([t]))) \\
&= m_P^{\otimes}(\bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \langle [p], \mathrm{ass}^{\#}_{T_\Sigma(X)}(f^{\iota}_{[t],[p],x})\rangle)
\end{aligned}$$

Now, because each $\mathrm{ass}^{\#}_{T_\Sigma(X)}$ is of the form $\{z \leftarrow t\}$, and by definition each $f_{[t],[p],x}$ is a function, we have that $\mathrm{ass}^{\#}_{T_\Sigma(X)}(f_{[t],[p],x}) = y$ with $y \in \delta_\iota([p], t))$, where t is the t in $\{z \leftarrow t\}$. So we get:

$$\begin{aligned}
& m_P^{\otimes}(\bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \langle [p], \mathrm{ass}^{\#}_{T_\Sigma(X)}(f^{\iota}_{[t],[p],x})\rangle) \\
&= \bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \mathrm{ass}^{\#}_{T_\Sigma(X)}(f^{\iota}_{[t],[p],x}) \\
&= \bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} y_x \\
&= \iota_{N_1}(t) \\
&= \iota_{N_1}(t)(m_T(\langle [t], \mathrm{ass}^{\#}_{T_\Sigma(X)}\rangle))
\end{aligned}$$

The proof for o is analogous

3. To obtain the isomorphism $\mathsf{U}(AN) \simeq N_2$ observe that by construction we have for $\mathsf{U}(AN) = \langle T_U, P_U, \iota_U, o_U\rangle$ that $T_U = T_2$, and $P_U = P_2$.

   It remains to check that $\iota = \iota_2$ and $o = o_2$. By definition $\iota(t)_U = \iota; \mathrm{ass}^{*}_{\perp} = \bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \langle [p]\rangle = \iota_2(t)$.

$\square$

The construction codes the folding f into the annotation functions. Although we allow the sets T and P to be infinite, we have assumed that the functions f_T, f_P must be recursive. Thus according to a result by Bergstra and Tucker (1987) there exists a specification of the functions f_T, f_P

3.6 Example
Given the nets in figure 1 the folded net is given in figure 2. $\square$

To make the folding construction into a functor $\mathsf{Fold} : \underline{\mathrm{PetriG}} \to \mathrm{AN}$, we need to define maps $\mathsf{Fold}_{\mathrm{obj}} : |\underline{\mathrm{PetriG}}| \to |\mathrm{AN}|$ and $\mathsf{Fold}_{\mathrm{mor}} : \mathbf{Mor}_{\underline{\mathrm{PetriG}}} \to \mathbf{Mor}_{\mathrm{AN}}$. However the construction is defined by a morphism, so it is not quite clear how to define the map on objects. An obvious solution would be to define $\mathsf{Fold}_{\mathrm{obj}}(N)$ as the folding of $f : N \to *$, where $*$ is the terminal object. But unfortunately PetriG has no terminal object. The terminal object of Petri is not usable, because the terminal maps are not maps in PetriG. The reason why PetriG fails to have a terminal object is that since the arrows are free extensions of Set-morphisms, the total weight of the arc-annotation is always preserved.

However in PetriG for each net N we can find and object N_M, such that the morphism $f : N \to N_M$ is unique. It has just one place, but for each possible combination of input and output weights a transition. Given a transition $t : a_1 \otimes \ldots \otimes a_n \to b_1 \otimes \ldots \otimes b_m$ let the type$(t) = \langle n, m \rangle$. Thus to each transition we can attach a pair of integers that denote its *type*.

3.7 Proposition
Given a net $N = \langle T, P, \iota, o \rangle$ there exists a net $N_M = \langle P_M, T_M, \iota_M, o_M \rangle$ with one place, $P_M = \{*\}$, and for each $t \in T$ there exist $t' \in T_M$ such that type$(t') =$ type(t), and the morphism $m_N : N \to N_M$ is unique.

Proof:
Clearly the function type partitions the set of transitions of N into equivalence classes. Thus f_T is uniquely defined. The map f_P is just the terminal map in Set. □

3.8 Lemma
In PetriG, the nets N_M with one place form a partial order without bottom.

Proof:
Follows directly from the fact that the collection of nets N_M is isomorphic to $\mathcal{P}(\mathbb{N}_1 \times \mathbb{N}_1)$ ordered by inclusion, where $\mathbb{N}_1 = \{n \mid n \geq 1, n \in \mathbb{N}\}$. □

3.9 Proposition
Given a PetriG-net N denote the algebraic net obtained through proposition 3.7 by $\mathsf{Fold}(N)$. Given a PetriG-morphism $f : N \to N'$, there exists an AN-morphism $\mathsf{Fold}(f) : \mathsf{Fold}(N) \to \mathsf{Fold}(N')$ defining a functor $\mathsf{Fold} :$ PetriG $\to$ AN.

Proof:
The net-part of the map $\langle h_T, h_P, h_\Sigma, h_A \rangle : \mathsf{Fold}(N) \to \mathsf{Fold}(N')$ is obtained from the inclusion $i : N_M \to N'_M$.

The map h_Σ is constructed by "simulating" the map f by a signature morphism. This is possible, because in the process of folding, all the information of the net is retained in the equational signature.

On sorts it is given by $h_S = i_T \cup i_P$. For the operators the situation is more complicated. Essentially the map is given by $h_\sigma : f_{[t],[p],x} \mapsto f_{i_t([t]),i_p([p]),y}$, but is unclear what the x should map to. The problem arises because we must somehow recover the choice we made in step 1.(d) in theorem 3.5. However, we

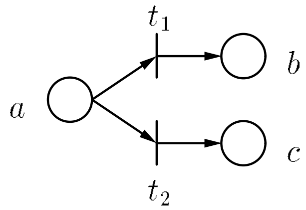
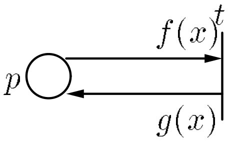


Figure 4: The net N.



$$\begin{array}{llll}
S: & \{p,t\} & & \\
\Sigma: & a :\to p & EQ & f(t_1) = a \\
 & b :\to p & & f(t_2) = a \\
 & c :\to p & & g(t_1) = b \\
 & t_1 :\to t & & g(t_2) = c \\
 & t_2 :\to t & & \\
 & f : t \to p & & \\
 & g : t \to p & &
\end{array}$$

Figure 5: The folding of N.

can recover the choice, because the set of equations of $\mathsf{Fold}(N)$ and $\mathsf{Fold}(N')$ have this mapping encoded. That is given the set of equations defining the behavior of an operator $f_{[t],[p],x}$ we can construct a map $\phi_{f_{[t],[p]}} : [p] \to \mathbb{N}$ given by $\phi_{f_{[t],[p]}}(p) = x$ iff $f_{[t],[p],x} = p \in EQ$. Now given $f_P : P \to P'$ we can construct a map $\Phi : \mathbb{N} \to \mathbb{N}$ such that the diagram commutes:

$$\begin{tikzcd}
P \arrow[r, "f_P"] \arrow[d, "\phi_{f_{[t],[p]}}"'] & P' \arrow[d, "\phi_{f_{f_T([t]),f_P([p])}}"] \\
\mathbb{N} \arrow[r, "\Phi"'] & \mathbb{N}
\end{tikzcd}$$

The map h_σ is then defined by:

$$h_\sigma : f_{[t],[p],x} \mapsto f_{i_t([t]),i_p([p]),\Phi(x)} \; .$$

$\square$

3.10 Corollary
$\mathsf{Unf}(\mathsf{Fold}(N)) \simeq N$

Proof:
Follows directly from theorem 3.5. $\square$

$$
\begin{array}{lll}
\texttt{sorts} & : & \mathit{phil}, \mathit{fork} \\
\texttt{opns} & : & ph_1, ph_2, ph_3 \quad :\to \mathit{phil} \\
 & & f_1, f_2, f_3 \qquad\; :\to \mathit{fork} \\
 & & l \qquad\qquad\qquad\; : \mathit{phil} \to \mathit{fork} \\
 & & r \qquad\qquad\qquad\; : \mathit{phil} \to \mathit{fork} \\
\texttt{eqns} & : & l(ph_i) = f_i \\
 & & r(ph_i) = f_i \qquad \text{for } i = 1, 2 \\
 & & r(ph_3) = f_1
\end{array}
$$

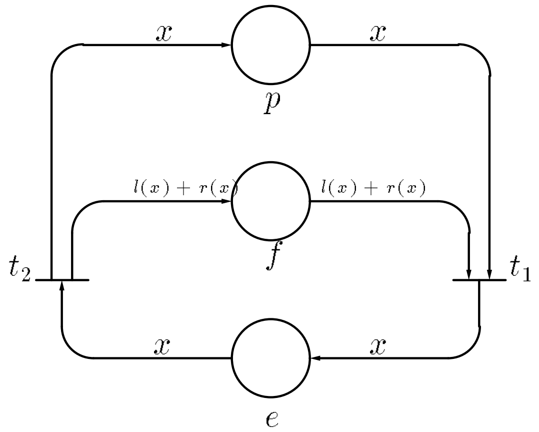


Figure 6: The specification AN of the dining philosophers problem.

3.11 Example
The net in figure 5 is the net corresponding to the folding $f : N \to N_T$ in figure 4. □

Observe that choice is mapped into surjective annotation functions.

It is naturally interesting to ask whether $\mathsf{Fold}(\mathsf{Unf}(AN)) \simeq AN$? However it is rather easy to see, that this is not the case. We do not even have a map $\mathsf{Fold}(\mathsf{Unf}(AN)) \to AN$. Figure 8 is the result of folding the net in figure 7, which is the unfolding of the net in figure 6. It is clear that since all information of the specification in the net AN is lost during unfolding, there is no way to automatically recover this.

Here a folding has been defined as a surjective mapping between nets. Since a surjective map defines an equivalence on its domain, it would be interesting to look at the reverse question: When does an equivalence on places define a folding? Clearly the equivalence must be such that it respects the flow of the net.

3.12 Definition
Given a net $N = \langle T, P, \iota, o \rangle$ an equivalence relation $\equiv$ is said to be *compatible* iff

$$\forall\, [p], \forall\, t, t' \in i^{-1}([p]) : |\delta_i([p], t)| = |\delta_i([p], t')|,$$

where $i^{-1}([p]) = \{t \mid \exists\, p \in [p] : p \in \gamma(i(t))\}$ and $i \in \{\iota, o\}$. □

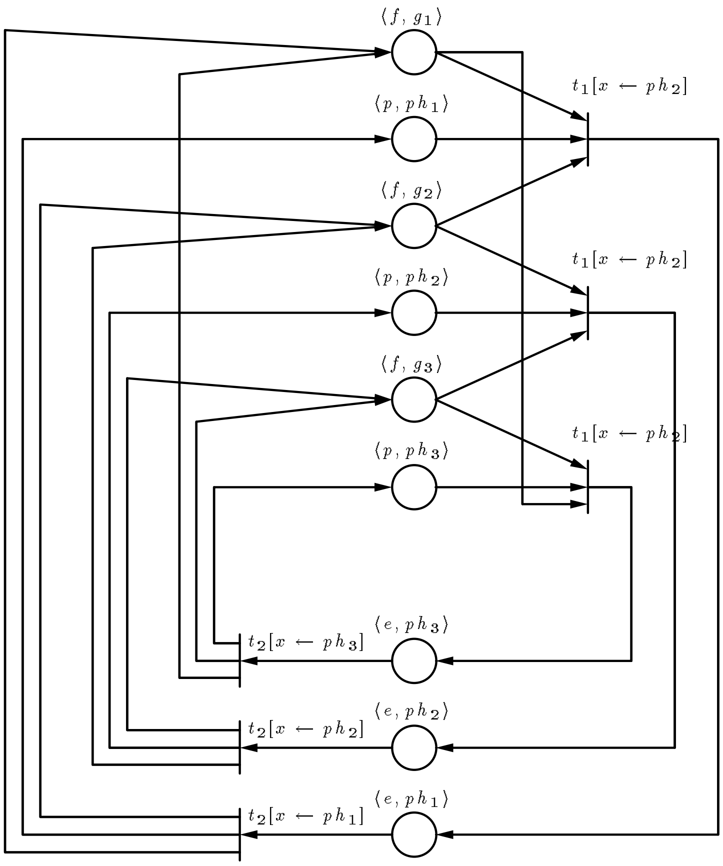


Figure 7: The net $\mathsf{Unf}(AN)$.

sorts : p, t_1, t_2

opns :
$$\langle f, g_1\rangle, \langle f, g_2\rangle, \ldots : \to p$$
$$\langle t_1, [x \leftarrow ph_1]\rangle, \ldots : \to t_1$$
$$\langle t_2, [x \leftarrow ph_1]\rangle, \ldots : \to t_2$$
$$f^{\iota}_{t_1,p,1} f^{\iota}_{t_1,p,2}, f^{\iota}_{t_1,p,3} : t_1 \to p$$
$$f^{o}_{t_1,p} : t_1 \to p$$
$$f^{\iota}_{t_2,p} : t_2 \to p$$
$$f^{o}_{t_2,p,1}, f^{o}_{t_2,p,2}, f^{o}_{t_2,p,3} : t_2 \to p$$

eqns :
$$f^{\iota}_{t_1,p,1}(\langle t_1, [x \leftarrow ph_1]\rangle) = \langle f, g_1\rangle$$
$$f^{\iota}_{t_1,p,2}(\langle t_1, [x \leftarrow ph_1]\rangle) = \langle f, ph_1\rangle$$
$$f^{\iota}_{t_1,p,3}(\langle t_1, [x \leftarrow ph_1]\rangle) = \langle f, g_2\rangle$$
$$\vdots$$

$$\iota(t_1) = f^{\iota}_{t_1,p,1}(x) \otimes f^{\iota}_{t_1,p,2}(x) \otimes f^{\iota}_{t_1,p,3}(x)$$

$$o(t_2) = f^{o}_{t_2,p,1}(x) \otimes f^{o}_{t_2,p,2}(x) \otimes f^{o}_{t_2,p,3}(x)$$

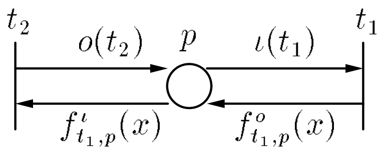


Figure 8: The result of $\mathsf{Fold}(\mathsf{Unf}(AN))$.

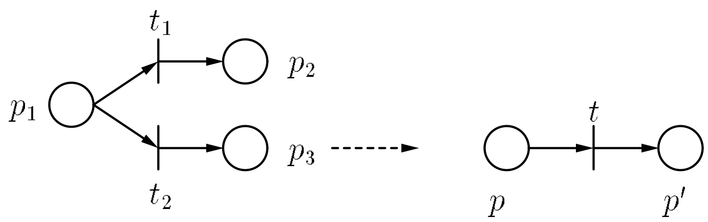


Figure 9: The net AN.

The intuition of this definition is that the equivalence relation should be compatible with the arc-weights of the original net. We can now state the following proposition.

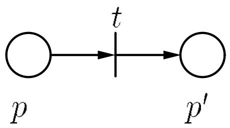
3.13 Proposition
Given a net $N = \langle T, P, \iota, o \rangle$ and a compatible equivalence relation $\equiv$, we can construct a folding $\langle f_T, f_P \rangle$, such that $\equiv = P/\ker f_P$. □

4 Foldings in AN

It is interesting also to look also at foldings in the category AN. That is given an Algebraic net AN a folding $f : \mathsf{U}(AN) \to N$, can we construct an algebraic net AN' s.th $\mathsf{Unf}(AN) \simeq \mathsf{Unf}(AN')$ and $U(AN') \simeq N$? This time it is not always possible to fold or to obtain a net with isomorphic semantics.

Consider the algebraic net AN in figure 9. Suppose we want to fold the transitions t_1 and t_2 into one, ie onto transition t of N. First of all we must make sure that the sort of places p_2 and p_3 are the same, otherwise it is not possible to construct an output-arc annotation for t. Let us suppose, that this is the case in this example. To construct the annotation functions of AN' we can use the idea of adding the set of transitions of AN as a new sort to the signature of AN'. One can the construct a new function k_t for each transition t of AN' that is defined by and equation $k^i_t(\overline{x}, t') = \pi_2(i(t'))$. So in our example we get the net in figure 10 (we have only written down the equations for the k^i_t, the rest of the specification is the same as for AN). However, now $\mathsf{Unf}(AN) \not\simeq \mathsf{Unf}(AN')$. To see why, notice that we have into k_t collected all the variables in $f(x)$ and $k(z, y)$. Thus we will for the case of t_1 have $Z \times Y$ assignments, where Z and Y are the sizes of the domains of the corresponding variables, which the implies that for each $t_1[x \leftarrow c]$ in the net $\mathsf{Unf}(AN)$ we have $Z \times Y$ transitions in $\mathsf{Unf}(AN')$. To get rid of this problem, we need to require that $Var(t) = Var(t')$ $\forall t, t' \in [t]$.

4.1 Definition
Given an Algebraic net AN, a folding $f : \mathsf{U}(AN) \to N$ is an *AN-folding* iff:



$$
\begin{array}{rcl}
k_t^{\iota}(x,y,z,t_1) & = & f(x) \\
k_t^{\iota}(x,y,z,t_2) & = & g(y,z) \\
k_t^{o}(x,y,z,t_1) & = & x \\
k_t^{o}(x,y,z,t_2) & = & h(y,z)
\end{array}
$$

Figure 10: The net AN'.

1. $Var(t) = Var(t') \;\; \forall t, t' \in [t]$, and
2. $\text{sort}(p) = \text{sort}(p') \; \forall p, p' \in [p]$.

$\square$

Then we can give the following theorem. Again, care must be taken in the case where two places are folded onto the same place.

4.2 Theorem
Given an Algebraic net AN and an AN-folding $f : \mathsf{U}(AN) \to N$, we can construct and algebraic net AN' s.th. $\mathsf{Unf}(AN) \simeq \mathsf{Unf}(AN')$ and $U(AN') \simeq N$.
Proof:
Let $AN' = \langle P, T, X, \Sigma_{AN'}, EQ_{AN'} \rangle$ be given by:

1. $P = P_N$,
2. $T = T_N$,
3. $\text{sort}([p]) = \text{sort}_{AN}(p)$,
4. $X = X_{AN} \bigcup T_N$,
5. $S = S_{AN} \bigcup T_N$,
6. The set of operator symbols consists of the operator symbols of AN, together with a new constant operator for each transition in AN and the new operators that express the folding. Again we must define these with care.

   The set of transition symbols is

   $$\Sigma_T = \{t : \to [t] \mid \forall t \in T_N\} \; .$$

To define the operators k we need the following auxiliary functions: $\delta_\iota : P_2 \times T_1 \to P_1$ and $\delta_o : P_2 \times T_1 \to P_1$ given by

$$\delta_\iota([p], t) = \gamma(\iota(t)) \cap \gamma(\mathsf{U}(f_P^{-1}([p])))$$

and

$$\delta_o([p], t) = \gamma(o(t)) \cap \gamma(\mathsf{U}(f_P^{-1}([p]))) \; .$$

The functions give the set of input(output)-places of t in AN that are mapped to $[p]$ in N.

Then let $n^i_{[t],[p]} = \max\{|\delta_i([p], t)| \mid t \in [t]\}$ for $i \in \{\iota, o\}$. $n^i_{[t],[p]} > 1$ if we fold several places of a transition onto one.

The set Σ_k is given by[4]

$$\begin{aligned}\Sigma_k \quad = \quad & \{k^i_{[t],[p],n} : \\ & \mathrm{sort}(Var(f^{-1}([t]))) \times [t] \to \mathrm{sort}([p]) \mid \\ & [t] \in T, [p] \in P, 1 \leq n \leq n^i_{[t],[p]}\},\end{aligned}$$

So we have

$$\Sigma_{AN'} = \Sigma_{AN} \bigcup \Sigma_T \bigcup \Sigma_k \; .$$

7. The set of new equations is given by

$$\begin{aligned}EQ^i \quad = \quad & \{k^i_{[t],[p],n}(\overline{x}, t) = \pi_1(i_{AN}(t)) \mid \\ & \forall t \in [t], [p] \in P, [t] \in T\}\end{aligned}$$

where $\overline{x}$ is the set of all variables $Var(f^{-1}(t))$ seen as a vector. Thus

$$EQ_{AN'} = EQ_{AN} \bigcup EQ^i \; .$$

8. Then we have

$$\iota([t]) = \bigotimes_{1 \leq x \leq n^\iota_{[t],[p]}} \langle [p], f^\iota_{[t],[p],x} \rangle,$$

and

$$o([t] = \bigotimes_{1 \leq x \leq n^o_{[t],[p]}} \langle [p], f^o_{[t],[p],x} \rangle \; .$$

9. Finally $A' = T_{\Sigma_{AN'}}$.

[4] In the sequel i ranges over $\{\iota, o\}$

The fact that $N \simeq \mathsf{U}(AN')$ is evident. It remains to prove, that $\mathsf{Unf}(AN) \simeq \mathsf{Unf}(AN')$. We have:

$$\begin{aligned} P_{\mathsf{Unf}(AN')} &= \bigcup_{p \in P} p \times A_{\mathrm{sort}(p)}, \\ T_{\mathsf{Unf}(AN')} &= \{\langle [t], \mathrm{ass}^{\#}_{A_{AN'}} \rangle\} \end{aligned}$$

Now $P_{\mathsf{Unf}(AN')} \simeq P_{\mathsf{Unf}(AN)}$, as by construction $T_{\Sigma_{AN'}} \restriction \mathrm{sort}_{P_{AN}} \simeq T_{\Sigma_{AN}} \restriction \mathrm{sort}_P$. The map $m_P : P_{\mathsf{Unf}(AN)} \to P_{\mathsf{Unf}(AN')}$ is given by $m_P(\langle p, \hat{t} \rangle) = \langle [p], \hat{t} \rangle$ because of the definition of $k_{[t],[p],n}$ by the equations in EQ^i.

We also have that $T_{\mathsf{Unf}(AN')} \simeq T_{\mathsf{Unf}(AN)}$, because $Var(t) = Var(t') \quad \forall t, t' \in f^{-1}([t])$ and through the definition of EQ^i it is the case that $\mathrm{ass}^{\#}_{A_{AN'}} = \mathrm{ass}^{\#}_{A_{AN}} \bigcup \{t \leftarrow t_1\}$, where $\mathrm{ass}^{\#}_{A_{AN}} \in [Var(t_1) \to A_{AN}]$. The map $m_T : T_{\mathsf{Unf}(AN)} \simeq T_{\mathsf{Unf}(AN')}$ is now given by $m_T(\langle t, \mathrm{ass}^{\#}_{A_{AN}} \rangle) = \langle [t], \mathrm{ass}^{\#}_{A_{AN}} \bigcup \{[t] \leftarrow t\} \rangle$.

It remains to show that $\iota_{\mathsf{Unf}(AN)}; m_P = m_T; \iota_{\mathsf{Unf}(AN')}$:

$$\begin{aligned} & m_P(\iota_{\mathsf{Sem}(AN)}(\langle [t], \mathrm{ass}^{\#}_{T_{\Sigma_{AN'}}} \rangle)) \\ &= m_P(\mathrm{ass}^{*}_{T_{\Sigma_{AN'}}}(\iota([t]))) \\ &= m_P^{\otimes}(\bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \langle [p], \mathrm{ass}^{\#}_{T_{\Sigma_{AN'}}}(k^{\iota}_{[t],[p],x}) \rangle) \end{aligned}$$

Now, because each $\mathrm{ass}^{\#}_{T_{\Sigma_{AN'}}(X)}$ is of the form $\mathrm{ass}^{\#}_{A_{AN}} \bigcup \{z \leftarrow t\}$, and by definition each $k_{[t],[p],x}$ is a function, we have that $\mathrm{ass}^{\#}_{T_{\Sigma_{AN}}}(f_{[t],[p],x}) = \iota(t)$, where t is the t in $\{z \leftarrow t\}$. So we get:

$$\begin{aligned} & m_P^{\otimes}(\bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \langle [p], \mathrm{ass}^{\#}_{T_{\Sigma_{AN}}}(k^{\iota}_{[t],[p],x}) \rangle) \\ &= \bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} \mathrm{ass}^{\#}_{T_{\Sigma_{AN}}}(k^{\iota}_{[t],[p],x}) \\ &= \bigotimes_{1 \le x \le n^{\iota}_{[t],[p]}} y_x \\ &= \iota_N(t) \\ &= \iota_N(t)(m_T(\langle [t], \mathrm{ass}^{\#}_{T_{\Sigma_{AN}}} \rangle)) \end{aligned}$$

$\square$

It turns out, that since every Petri net is an Algebraic net through the functor G, the construction of theorem 2.13 is a special case of the construction presented above.

5 Deadlock preserving skeletons

The idea of forgetting the annotations on the algebraic net and investigating the resulting *skeleton* for deadlocks was first proposed by Vautherin (1987). As pointed out by Findlow (1991), the result is of limited usefulness, as even if the dead skeletal marking is reachable its pre-image may not be. Vautherin (1987) gave a sufficient condition on the algebraic net to have the property that a marking is dead iff the image of the marking in the skeleton is dead, but this condition is very restrictive. This observation later led to a set of necessary and sufficient conditions for a net to have a deadlock preserving skeleton (Findlow 1992), together with a transformation algorithm that unfolds and then partially refolds the net until it satisfies the conditions. In this section we will transfer Findlows' result to algebraic nets to give an application for the construction in section 3.

The problem with the original idea of analyzing skeletons, as pointed out by Findlow (1991), is that since it is defined without reference to an initial marking it does not distinguish between reachable dead markings and non-reachable dead markings. Thus if we look at the skeleton and find a deadlock, it may well be, that a corresponding marking in the net $\langle AN, T_\Sigma \rangle$ is not reachable. To overcome this problem Vautherin (1987) proposed the following condition.

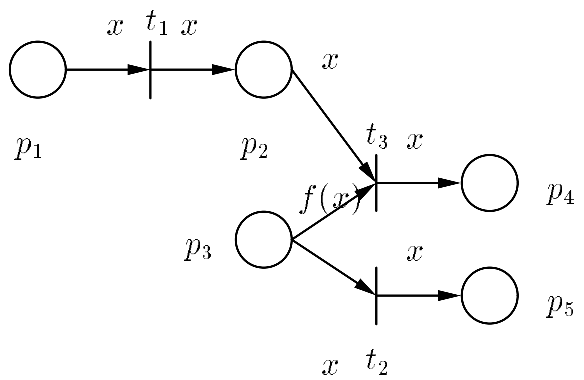
5.1 Proposition
Given an Algebraic Net $\langle AN, T_\Sigma \rangle$ if for each transition t and each place p in ${}^\bullet t$, with $\iota(t) = \langle p, f_1 \rangle \otimes \ldots \otimes \langle p, f_n \rangle$ we have:

1. $(p, i) \neq (p', j) \Rightarrow Var(f_i) \bigcap Var(f_j) = \varnothing$, and
2. $\alpha(f_i)$ is surjective in $T_\Sigma(X)/EQ$,

then each marking m is a deadlock in $\mathsf{Unf}(AN, T_\Sigma)$ iff m is a deadlock in $\mathsf{Unf}(AN, \bot)$ (the skeleton). □

The condition is sufficient to ensure that a marking is enabled irrespective of the identity of the tokens. However this condition is very restrictive and there are many examples where the net has a deadlock-preserving skeleton although it does not satisfy the condition. For example the net N in figure 11 has a skeleton whose every deadlock is also a deadlock in N although it does not satisfy Vautherins condition. The fact that transition t_3 does not satisfy the condition is circumvented by transition t_2 which fires in any mode.

The basic idea in Findlows methods is to test whether a place is live at a marking regardless of its color. In our case this amounts to the following definitions.



$$\begin{array}{rlll} sort: & s & & \\ ops: & a: & \to s & \\ & b: & \to s & \\ & f: & s \to s & \\ eqns: & f(a) & = a & \\ & f(b) & = a & \end{array}$$

Figure 11: A net with DPS that does not satisfy Vautherins condition

5.2 Definition
For any marking M of $\mathsf{Unf}(AN)$, with a token in place p, let $M(\langle p, \hat{t}\rangle \leftarrow \langle p, \hat{t}'\rangle)$ be the marking obtained by replacing the token $\hat{t}$ in p with $\hat{t}'$. Now define $\langle p, \hat{t}\rangle \leftarrow \langle p, \hat{t}'\rangle$ iff for any marking M of $\mathsf{Unf}(AN)$ M is live and $\hat{t} \in M(p) \Rightarrow M(\langle p, \hat{t}\rangle \leftarrow \langle p, \hat{t}'\rangle)$ is live. We will say that $\langle p, \hat{t}\rangle$ is interchangeable with $\langle p, \hat{t}'\rangle$ and write $\langle p, \hat{t}\rangle \leftrightarrow \langle p, \hat{t}'\rangle$ iff $\langle p, \hat{t}\rangle \leftarrow \langle p, \hat{t}'\rangle$ and $\langle p, \hat{t}'\rangle \leftarrow \langle p, \hat{t}\rangle$. □

5.3 Proposition
The relation $\langle p, \hat{t}\rangle \leftrightarrow \langle p, \hat{t}'\rangle$ defines an equivalence relation on the places of $\mathsf{Unf}(AN)$, denoted by $P_{\mathsf{Unf}(AN)}/\leftrightarrow$. □

The following lemma is now easy to see, because only places that are instances of the same high-level places are in the relation $\leftrightarrow$.

5.4 Lemma
$\leftrightarrow$ is a compatible equivalence relation. □

The necessary and sufficient condition is now given by the following proposition.

5.5 Proposition
Let N be and algebraic net. The following is a necessary and sufficient condition for every dead marking of N to have a dead skeletal image:

- For any place p of N, and any terms $\hat{t}, \hat{t}'$ of $sort(p)$

$$\langle p, \hat{t} \rangle \leftrightarrow \langle p, \hat{t}' \rangle$$

□

Findlow (1992) proposes an algorithm for calculating the equivalence relation $\leftrightarrow$.

We can now obtain a version of the net AN that has a deadlock-preserving skeleton through the following.

5.6 Theorem
Let $AN = \langle S, \Sigma, EQ, X, T, P, \iota, o, \mathrm{sort}, T_\Sigma \rangle$ be an arbitrary algebraic net. Define the net AN' as the net obtained by folding along $\langle f_P, f_T \rangle : \mathsf{Unf}(AN) \rightarrow \mathsf{Unf}(AN)/\leftrightarrow$. Then AN' has a deadlock-preserving skeleton.

Proof:
Because of lemma 5.4 we know that AN' is well defined and its unfolding is isomorphic to $\mathsf{Unf}(AN)$. Then the result follows because of proposition 5.5.

□

It should also be noted that since we are only interested in the skeleton, it is actually unnecessary to construct the specification for AN.

6 Conclusions

In this paper we have presented a construction that given a folding returns an Algebraic Net whose underlying net is isomorphic to the codomain of the folding and whose unfolding is isomorphic to the domain of the folding. We have also shown that this construction gives rise to a functor from the category PetriG to the category AN, and presented and application for the construction.

The work solves the problem of describing foldings of non-strict nets left open by Smith and Reisig (1987), although in a very concrete manner. Due to the non-strictness of our nets and the algebraic nature of the morphism in PetriG we are unable to come up with a classification scheme of morphisms and the corresponding classification of high-level systems as proposed by Smith and Reisig (1987). Our construction gives just one type of annotation functions (unary-functions). This is also the reason why the folding and unfolding functors are not adjoint. There is no way to relate the specification obtained by folding to the original specification since all information about it is lost while unfolding.

A construction similar to the one in section 4 has recently also been proposed by Battiston, Cindio, Mauri and Rapanotti (1991) in the context of so called minimal models of OBJSA Nets. They present a construction that maps an OBJSA net into an equivalent net that is minimal in the sense that it has only one place for each net-component. This mapping allows them to define an equivalence on OBJSA Nets that describes the level of abstraction of the net. Our idea of partitioning PetriG according to the types of the transitions was inspired by the minimal model construction. However our folding construction is different because we fold transitions. The construction presented by Battiston et al. (1991) leaves the transitions alone and only folds the places of each transition. However the construction has recently been extended to the folding of places (de Cindio 1994).

It should be noted that there are other ways to define the notion of AN-folding in section 4. The requirement that the sorts should be preserved could be circumvented by introducing new "union" sorts for each place in the folded net. Then the equations for the operators $k^i_{[t],[p],x}$ would have to be modified to include suitable "injection" operators.

Our construction also has relied heavily on the fact that the morphisms in PetriG are free extensions of functions. It would be interesting to look at the construction when the morphisms are allowed to be Mon-morphisms. However we restricted ourselves to PetriG, since it is not quite clear what the right notion of folding is in Petri.

References

Battiston, E., Cindio, F. D., Mauri, G. and Rapanotti, L.: 1991, Morphisms and minimal models for OBJSA nets, *Proc. 12th International Conference on Application and Theory of Petri Nets*, Gjern, pp. 455–475.

Bergstra, J. and Tucker, J.: 1987, Algebraic specifications of computable and semicomputable data types, *Theoretical Computer Science* **50**, 137–151.

de Cindio, F.: 1994, Private Communication.

Degano, P., Meseguer, J. and Montanari, U.: 1989, Axiomatizing net computations and processes, *4th Symposium on Logic in Computer Science*, IEEE Computer Press.

Dimitrovici, C., Hummert, U. and Pétrucci, L.: 1991, Semantics, composition, and properties of algebraic high-level nets, *in* G. Rozenberg (ed.), *Advances in Petri Nets 1991*, Vol. 524 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, pp. 93–117.

Findlow, G.: 1991, Can skeletons really be used to detect deadlocks of Nets, *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, pp. 198–203.

Findlow, G.: 1992, Obtaining deadlock-preserving skeletons for Coloured Nets, *in* K. Jensen (ed.), *Application and Theory of Petri Nets*, Vol. 616 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, pp. 173–192.

Genrich, H., Lautenbach, K. and Thiagarajan, P.: 1980, Elements of general net theory, *in* W. Brauer (ed.), *Net Theory and Applications*, Vol. 84 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin.

Goguen, J. A. and Burstall, R.: 1984, Some fundamental algebraic tools for the semantics of computation. Part1: Comma categories, colimits, signatures and theories, *Theoretical Computer Science* **31**, 175–209.

Goguen, J. A. and Burstall, R.: 1990, Institutions: Abstract model theory for specification and programming, *Research Report ECS-LFCS-90-106*, Laboratory for Foundations of Computer Science, University of Edinburgh, Scotland.

Lilius, J.: 1991, On the compositionality and analysis of algebraic high-level nets, *Research Report A16*, Digital Systems Laboratory.

Meseguer, J. and Montanari, U.: 1990, Petri nets are monoids, *Information and Computation* **88**, 105–155.

Petri, C. A.: 1973, Concepts of Net Theory, *Mathematical Foundations of Computer Science*, Math. Inst. of the Slovak Acad. of Science, pp. 137–146.

Smith, E. and Reisig, W.: 1987, The semantics of a net is a net: An exercise in general net theory, *in* K. Voss, H. Genrich and G. Rozenberg (eds), *Concurrency and Nets*, Springer Verlag, Berlin, pp. 461–497.

Vautherin, J.: 1987, Parallel systems specification with Coloured Petri Nets and algebraic specifications, *in* G. Rozenberg (ed.), *Advances in Petri Nets*, Vol. 266 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, pp. 293–308.

Appendix

This appendix contains some auxiliary definitions that would have cluttered the exposition.

A Many-sorted algebra

The theory of abstract datatypes is based on many-sorted algebra. A category-theoretic formulation is presented by Goguen and Burstall in (Goguen and Burstall 1984). In the following we shall follow the more concise formulation of (Goguen and Burstall 1990).

The fundamental intuition is, that a datatype consists of a set of elements of different sorts, and a set of operations making the datatype into an algebra. Each algebra has a signature that names the sorts and the operators of the algebra.

A.1 Definition
Given a set of *sorts* S an *equational signature* is a pair $\langle S, \Sigma \rangle$, where Σ is a family of sets (of *operator* names) indexed by $S^* \times S$, where S^* denotes the set of strings with alphabet S. The operator $\sigma \in \Sigma_{w,s}$ has *arity* w, *sort* s, and *rank* or *type* $\langle w, s \rangle$. It is customary to just write Σ instead of $\langle S, \Sigma \rangle$. □

A.2 Definition
A *equational signature morphism* $\Phi : \langle S, \Sigma \rangle \to \langle S', \Sigma' \rangle$ is pair $\langle h_S, h_\Sigma \rangle$ where $h_S : S \to S'$ is the sort map and h_Σ is a $S^* \times S$-indexed family of maps $h_{u,s} : \Sigma_{u,s} \to \Sigma'_{f^*(u),f(s)}$. □

A.3 Definition
Equational signatures and equational signature morphisms for the *category of equational signatures* $\underline{\mathsf{Sig}}$. □

A.4 Definition
Given a signature Σ, a Σ-algebra A is a S-indexed family of sets $|A| = \langle A_s | s \in S \rangle$ called the carries of A together with an $S^* \times S$ indexed family α of maps $\alpha_{u,s} : \Sigma_{u,s} \to [A_u \to A_s]$ for u in S^* and s in S, where $A_{s_1 \ldots s_n} = A_{s_1} \times \cdots \times A_{s_n}$ and $[A \to B]$ denotes the set of functions from A to B. □

A.5 Definition
A Σ-homomorphism is a S-indexed map $h : A \to A'$ that satisfies the equation

$$h(\sigma(x_1, \ldots, x_n) = h(\sigma)(h(x_1), \ldots, h(x_n)) \ .$$

□

Σ-algebras and Σ-homomorphism form a category Σ-$\underline{\mathsf{Alg}}$. There exists a *forgetful* functor $U : \Sigma\text{-}\underline{\mathsf{Alg}} \to \underline{\mathsf{Set}}_S$ which sends a Σ-algebra to its S-indexed set of carriers.

Given a S-indexed set of variables X we can form the set of all terms that are constructed with operator symbols from Σ and variables from X. This set forms a Σ-algebra in a natural way. It is called the *free* algebra over the set of variables X and denoted by $T_\Sigma(X)$. We first define a special case, defining $(T_\Sigma)_s$ to be the smallest set of strings of symbols such that:

- $\Sigma_{\lambda,s} \subseteq T_{\Sigma,s}$,
- $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and $t_i \in T_{\Sigma,s_i}$ imply that the string $\sigma(t_1, \ldots, t_n)$ is in $T_{\Sigma,s}$

If we now define α by

- for $\sigma \in \Sigma_{\lambda,s}$ let $\alpha(\sigma)$ be the string σ of length 1 in $T_{\Sigma,s}$,
- for $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and $t_i \in T_{\Sigma,s_i}$ let $\alpha(\sigma(t_1, \ldots, t_n))$ be the string $\sigma(t_1, \ldots, t_n)$ in $T_{\Sigma,s}$.

then α defines the Σ-structure on T_Σ. Let now define $\Sigma(X)$ be the signature with $(\Sigma(X))_{\lambda,s} = (\Sigma)_{\lambda,s} \cup X_s$ and $(\Sigma(X))_{u,s} = (\Sigma)_{u,s}$ if $u \neq \lambda$. Then the free algebra $T_\Sigma(X)$ is just $T_{\Sigma(X)}$ seen as a Σ-algebra.

The free algebra enjoys the following universal property. For any Σ-algebra, every map $f : X \to U(B)$ has a unique extension to a Σ-homomorphism $f^\# : T_\Sigma(X) \to B$. The map $f : X \to U(B)$ is called *variable assignment* and its extension $f^\#$ *assignment*.

The construction of a free Σ-algebra on a set of variables X extends to a functor $F : \underline{\mathsf{Set}} \to \Sigma\text{-}\underline{\mathsf{Alg}}$ that is left adjoint to the forgetful functor $U : \Sigma\text{-}\underline{\mathsf{Alg}} \to \underline{\mathsf{Set}}$. Together the composition $F; U$ is the functor part of a monad on $\underline{\mathsf{Set}}$ that is denoted by T_Σ.

B Monoids

B.1 Definition
Given a set S the free commutative monoid generated by S is denoted by $S^\otimes$. If $f : S \to S'$ is a $\underline{\mathsf{Set}}$-function, then the free extension along the functor $F : \underline{\mathsf{Set}} \to \underline{\mathsf{Mon}}$ is denoted by $f^\otimes$. $\square$

B.2 Definition
Given a monoid $A^\otimes$ with a set of generators A, define a function $\gamma_A : A^\otimes \to \mathcal{P}(A)$ by the assignment $\gamma_A(a_1 \otimes \cdots \otimes a_n) = \{a_1, \ldots, a_n\}$. γ maps a term of the monoid into its set of generators. If the context is clear the subscript will be omitted. $\square$