Transformation Systems Semantics and Compositionality of Algebraic Petri Nets

Liliana Grigoriu

3rd October 2002

Abstract

In the first part of the paper, place-transition nets and algebraic high-level nets are translated into algebra transformation systems, thus defining the net semantics. Both sequential and concurrent nets are considered. Some results concerning the relation between the net semantics resulting from a rule-based system corresponding to the net, and a transformation system built directly from the net are presented.

The second part of the paper analyzes the correspondence between transformation system composition operations applied to place-transition net semantics, and syntactical operations on place-transition nets. It is shown that no correspondence exists for the general case, and reasons for this situation are outlined. Some cases are presented, in which a correspondence can be established, the most important of which is the gluing of two nets over a common subnet. Furthermore, some elementary mechanisms are presented, showing the way in which certain characteristics of synchronized compositions correspond to syntactical composition operations on nets, that go beyond the usual gluing techniques. This corresponds very much to finding out to what extent marking graph synchronization can be reflected by operations on nets.

Contents

Chapter 1

Introduction

Aim of the paper

One of the most important paradigms of systematical software development is its model-basedness. This means that models of the system or of the software to be developed are created, which allow an abstract and formal description of main aspects of the system. The construction of the models is in most cases supported by adequate languages or techniques, such as Petri nets of different kinds or UML diagram techniques.

Transformation systems are mathematical entities that represent models of systems and components without depending on specific languages or techniques (see [Gro99a] and [Gro99b]). In this way, models constructed with different techniques can be analyzed semantically and compared with one another. For this every language has to be translated in the given domain of transformation systems, i.e., its semantics is to be reconstructed in the formalism. For a range of languages this has already been done, even if mainly on examples.

Aim of this paper is to find and formally define the mapping of Petri nets of different types to algebra transformation systems and to analyze the connection between semantical composition of net semantics (on the layer of transformation systems) and syntactical composition of nets.

First, the transformation system semantics for each net is defined. Here we consider two ways to map nets to transformation systems. One is to represent the net as a rule-based system and to consider the system's semantics, i.e. the syntactic presentation of transformation systems by rewrite rules as given in [Gro99c] is used. The other way is to directly define a transformation system for the net. In both cases the marking graph of the net coincides with the resulting transformation system. Deriving the transformation system corresponding to one net in two different ways allows comparison between the two net modelings, and having a transformation system for a net allows comparison to other systems and synchronization with them.

The second part of the overall aim concerns the composition of Petri nets. In the framework of the transformation systems' formalism composition operations are defined, that represent in a very general way overlapping, synchronization and coordination of components. These operations are defined categorically, using categorical constructs called limits. This means that their properties and compatibilities can be determined immediately from general categorical results. In this paper the parallel composition of transformation systems with synchronization and data state overlapping is transfered to Petri nets. This means that, after the transformation system semantics of Petri nets has been given, a syntactic representation of the synchronized parallel composition operation (described in [Gro01]) on the layer of Petri nets is sought.

Petri nets have been introduced by C.A. Petri in [Pet62], and since then many variants have been developed. The main extension directions have allowed representation of data types (and of firing conditions related to data objects), which led to high-level nets, and representation of the time events might take, which led to timed nets. In this paper we consider representatives of low-level nets and of high-level nets[1]. We chose to look at Place-Transition nets, since they are the most general and probably most used low-level nets, and at Algebraic High-Level nets, a net class that combines Coloured Petri Nets and Algebraic Specifications ([Vau87],[Hum89]) shown to be more general than Colored Petri Nets.

Composition of Petri nets has been mainly studied from two points of view. In papers like [Win87, Hum89, MM90, JPR95], properties of horizontal composition, i.e. the gluing of places and transitions among two or more nets (union), or within the same net (fusion), are studied. Secondly, properties of vertical composition or (rule-based) refinement of nets have been studied, ([JPR95],[Pad96]).

In this paper we analyze how much the very general composition operation of transformation systems can be mapped to operations on Petri nets. This corresponds very much to finding out to what extent marking graph synchronization can be reflected by operations on nets. Even though results

[1]Since Transformation Systems are not aimed towards considering timed applications, we do not consider this second development direction here

are shown only for Place-Transition nets, we have good reasons to believe that they can be extended to Algebraic High-Level Nets.

Structure of the paper

After this first introductory chapter, the second chapter gives a mapping from place-transition nets and from algebraic high-level nets to transformation systems. First, the necessary definitions are given. Afterwards, the nets are translated into the representation of transformation systems based on rewrite rules, and then a direct translation into the transformation system formalism is given. Finally the transformation system corresponding to the first translation is compared to the system corresponding to the second one.

In the third chapter, the synchronized composition operation for transformation systems is analysed with regard to its corresponding syntactical represenations on place-transition net. First, the fact that there is no representation for the general case is shown, and some reasons for this situation are given. Then, correspondences between the way in which the parameters of the synchronized composition are actualized and the induced syntactical operations on nets are outlined. A main result is that there is a synchronized composition operation that corresponds to gluing of two (concurrent) place-transition nets over a common subnet. For a certain type of synchronized composition, a syntactical representation we called *or-synchronization* is given, where a transition of one of the two nets to be synchronized basically can fire only at the same time with one *or* another transition from a set of transitions of the other net.

In the fourth chapter a summary of the thesis is given, conclusions are drawn, and questions that are still open are indicated.

Chapter 2

Algebra Transformation Systems for Algebraic Petri Nets

In this chapter we define a mapping from place-transition nets and from algebraic high-level nets to transformation systems. Both nets that allow parallel firing of transitions and sequential nets are considered. First, the necessary definitions and theorems from the literature are given. Afterwards, the nets are translated into the representation of transformation systems based on rewrite rules, and then a direct translation into the transformation system formalism is given. Finally the transformation system corresponding to the first translation is formally compared with the system obtained by the second one.

2.1 Algebra Rewrite Specifications and their Corresponding Transformation Systems

This section provides the definitions and theorems that are needed in this paper.

We use a slightly adapted version, suitable to our purposes of algebra rewrite specifications, however, unless otherwise specified, the following definitions are identical to those in [Gro99c]. All adaptions are discussed in the appendix (embedding in the [Gro99c]-variant will be shown). We shall also use an extension to the versions of transformation systems defined in [Gro99a] and [Gro01], in that rewrite rules with conditions are employed.

2.1.1 Algebra Rewrite Specifications

In this subsection we define algebra rewrite specifications. They are defined in Definition 2.1, however, we will also need the Definitions 2.2 of presentations and 2.4 of rules in order to specify them completely. Based on these definitions, and on Fact 2.3, algebra rewriting is defined.

Definition 2.1 (Rewrite Specification, [Gro99c])
A *rewrite specification* $R\Gamma = ((\Sigma, E), M, \mathcal{X}, R)$ is given by

- an algebraic specification $\Gamma = (\Sigma, E)$ formed by a signature $\Sigma = (S, Op)$ and a set of equations E. In [Gro99c]conditional equations are allowed, but in this thesis we need only normal equations,
- a *method signature* $M = (M_{w;w'})_{w,w' \in S^*}$, where each $M_{w,w'}$ is a set of method names indexed by method parameter sorts. A method name $m \in M_{w;w'}$ is denoted by $m\colon w; w'$ for short,
- a set $\mathcal{X}$ of variables that contains all the variables used in the named rules,
- a set of named rules $R \subseteq NRules(R\Sigma, \mathcal{X})$, included in the set of all possible named rules over the variable set $\mathcal{X}$. The set $NRules(R\Sigma, \mathcal{X})$ is defined in Definition 2.4 below.
  The pair $R\Sigma = (\Sigma, M)$ is called *rewrite signature.*

The following definitions are needed in order to define rules and rule applications.

Definition 2.2 (Presentation, [Gro99c])
Let $\Sigma = (S, Op)$ be an algebraic signature. A Σ-presentation $P = (P_S, P_E)$ is given by an S-indexed set $P_S = (P_s)_{s \in S}$, the *generators*, and a set $P_E \subseteq Eqns_\Sigma(P_S)$, the *equations.* A presentation is *functional* if P_E is a set of function entries over P_S, i.e.,

$$P_E \subseteq \{f(a) = b \mid f : w \to v \in Op, a \in P_w, b \in P_v\} \subseteq Eqns_\Sigma(P_S).$$

A *morphism* of Σ-presentations $p : (P_S, P_E) \to (P'_S, P'_E)$ is an S-indexed function $p = (p_s : P_s \to P'_s)_{s \in S}$ such that $P_E[p] \subseteq P'_E$, where $\_[p]$ denotes the substitution of generators according to p. Σ-presentations and morphisms yield the category $\mathbf{Pres}(\Sigma)$. Finally let $\mathbf{Pres}(\Gamma) = \mathbf{Pres}(\Sigma)$ for each algebraic specification $\Gamma = (\Sigma, E)$ extending Σ.

Let Γ be a algebraic specification. For each Γ-presentation $P = (P_S, P_E)$ there is a smallest partial Γ-algebra A^P that contains the generators P_S and satisfies the equations P_E. This property is formally stated as the existence of a free functor from the category of Γ-presentations to the category of partial Γ-algebras, i.e., a left adjoint to the *presentation functor* that maps partial Γ-algebras to Γ-presentations.

Fact 2.3 (Adjunction of presentations and algebras, [Gro99c])
Let $\Gamma = (\Sigma, E)$ be a partial equational specification. The *presentation functor* $Pres_\Gamma : \mathbf{PAlg}(\Gamma) \to \mathbf{Pres}(\Gamma)$ given by

$$Pres_\Gamma(A_S, A_{Op}) = (A_S, \{f(a) = b \mid f^A(a) = b\})$$

and $Pres_\Gamma(h) = h$, has a left adjoint $PAlg_\Gamma : \mathbf{Pres}(\Gamma) \to \mathbf{PAlg}(\Gamma)$. Here, $\mathbf{Palg(\Gamma)}$ denotes the category of Γ-partial algebras.

The following definition is an extended version of the definition of rules giben in [Gro99c]. The extension is discussed in the appendix.

Definition 2.4 (Rewrite Rules and Named Rules, [Gro99c])
Let $\Sigma = (S, Op)$ be a signature and $(\mathcal{X}_s)_{s\in S}$ be a set of variables. Furthermore let the set $Pres(\Sigma, \mathcal{X})$ of $(\Sigma, \mathcal{X})$-presentations be given by all Σ-presentations $P = (P_S, P_E)$ with $P_s \subseteq \mathcal{X}_s$ for each $s \in S$.

A $(\Sigma, \mathcal{X})$-*rewrite rule* $r = (P_{cond} \Rightarrow P_l \longleftrightarrow P_r)$ is given by a functional $(\Sigma, \mathcal{X})$-presentation $P_l = (X_l, E_l)$ and by two $(\Sigma, \mathcal{X})$-presentations $P_{cond} = (X_{cond}, E_{cond})$ and $P_r = (X_r, E_r)$.

A *named rule* w.r.t. $R\Sigma$ and $\mathcal{X}$ is a pair $nr = \langle m(x_1, ..., x_n), r\rangle$ given by a method name m, variables $x_1 ... x_n \in \mathcal{X}$, and a rule $r = ((X_{cond}, E_{cond}) \Rightarrow (X_l, E_l) \longleftrightarrow (X_r, E_r))$ such that $\{x_1, ..., x_n\} \subseteq X_{cond} \cup X_l \cup X_r$. The set of all named $(R\Sigma, X)$-rules is denoted by $NRules(R\Sigma, \mathcal{X})$.

In the following, We shall use strings of sort symbols for variable declaration, as in [Gro01]. Furthermore, We shall consider in the following only rules with the property $X_{cond} = X_l = X_r$ since the restriction does not interfere with our purposes and allows a significant simplification in the mechanism of rule application.

Definition 2.5 (Algebra Rewriting, cf. [Gro99c])
Let $R\Gamma = (\Gamma, M, \mathcal{X}, R)$ be a rewrite specification with $\Gamma = (S, Op, E)$. Let $r = (P_{cond} \Rightarrow P_l \longleftrightarrow P_r)$ be a $((S, Op), \mathcal{X})$-rewrite rule, with $P_i = (X, E_i)$ for some $X \subseteq \mathcal{X}$ and $i \in \{cond, l, r\}$. Let A be a partial Γ-algebra with $Pres_\Gamma(A) = (A_S, A_E)$.

1. A *match* a *for* r *in* A is given by a Γ-presentation morphism

   $$a : (X, E_{cond} \cup E_l) \to (A_S, A_E)$$

2. Let $m \in M_{w;w'}$ be a method with $\langle m(\vec{x}), r\rangle \in R$, where $\vec{x} = (x_1, \ldots, x_n)$, and let a be a match for r in A. The *rewrite step*

   $$r/a = (m(a(\vec{x})) : A \Rightarrow B)$$

   is the transition from A to B labeled by the expression $m(a(\vec{x}))$, where the partial Γ-algebra B is constructed as follows.

   $$\begin{aligned} B_S &= A_S, \\ B_E &= (A_E - (E_l - E_r)[a]) \ \cup \ (E_r - E_l)[a], \\ B &= PAlg_\Gamma(B_S, B_E). \end{aligned}$$

The postfix notation $E[a]$ denotes the substitution of variables x in E by $a(x)$.

For more details on algebra rewrite specifications, [Gro99c] can be consulted.

2.1.2 Algebra Transformation Systems

In the following we will use definitions and propositions given in [Gro01] and [Gro99a].

The semantics of a rewrite specification is defined as an algebra transformation system (ATS), i.e. a transition system whose states are labeled by algebras. In the following we define how an ATS is obtained from a rewrite specification.

An ATS is a two-layered structure. The first layer is given by a *transition graph*, that models the temporal ordering of the steps of the system. The second one is given by a *data space*, given by data states and atomic data transformations. These are associated to the states and transitions of the transition graph respectively.

Definition 2.6 (Transition Graph)
A *transition graph* $TG = (CS, T, I, id)$ is given by

- a set CS of control states,
- a family $T = (T(c,d))_{c,d \in CS}$ of sets of transitions
- a set I of designated initial states[1],
- a function $id : CS \to T$ that assigns an *idle transition* $id(c) \in T(c,c)$ to each state $c \in CS$.

A *morphism* $(h_{CS}, h_T) : TG \to TG'$ of transition graphs $TG = (CS, T, I, id)$ and $TG' = (CS', T', I', id')$ is given by

- a mapping $h_{CS} : CS \to CS'$ and
- a family of mappings $h_T = (h_{T(c,d)})_{c,d \in CS}$ with

  $$h_{T(c,d)} : T(c,d) \to T'(h_{CS}(c), h_{CS}(d)),$$

  for all $c, d \in CS$, such that
- $h_{CS}(i) \in I'$ for each $i \in I$ and $h_{T(c,c)}(id(c)) = id(h_{CS}(c))$ for each $c \in CS$.

Definition 2.7 (Data Space Specification)
A *data space specification* $D\Gamma = (\Gamma, M)$ is given by an algebraic specification $\Gamma = (S, Op, E)$, called the *data state specification* and a method signature $M = (M_{w;w'})_{w,w' \in S^*}$ (That means, a data space specification is a rewrite specification without rules.).

In the data space specifications of [Gro01] conditional equations are allowed, but in this thesis we shall use only normal equations.

Definition 2.8 (Data Space)
Let $D\Gamma = (\Gamma, M)$ be a data space specification with $\Gamma = (S, Op, E)$.

1. Given S-sorted sets A and B, the set of *actions* that can be labels of transformations from A to B is:

[1]This corresponds to the [Gro01] variant with $CS' = CS \uplus \{i\}$ and $T' = T \cup \{i \to c | c \in I\}$

$$Act_{D\Gamma}(A,B) = \{m(a,b) | m \in M_{w;w'}, a \in A_w, b \in B_{w'}\}$$

2. Given partial Σ-algebras C, D the set of transformations $Tf_{D\Gamma}(C,D)$ is defined by

$$Tf_{D\Gamma}(C,D) = \mathcal{P}(Act_{D\Gamma}(|C|,|D|))$$

where $|C|$ denotes the family of carrier sets of C and where $\mathcal{P}$ denotes the powerset[2].

3. The *data space*

$$\mathcal{D}_{D\Gamma} = (|PAlg(\Gamma)|, Tf_{D\Gamma}, |PAlg(\Gamma)|, id)$$

is given by :
– the class of partial Γ-algebras $|PAlg(\Gamma)|$ as a class of control states and of initial states,
– the set $Tf_{D\Gamma}(C,D)$ as set of transformations from C to D,
– and the function $id : |PAlg(\Gamma)| \to Tf_{D\Gamma}$ with $id(c) = \emptyset$.
Note that the data space $\mathcal{D}_{D\Gamma}$ is also a transition graph.

Definition 2.9 (Algebra Transformation System)
Let $D\Gamma = (\Gamma, M)$ be a data space specification. A *$D\Gamma$-transformation system* $\mathbf{A} = (TG_{\mathbf{A}}, \mathbf{a})$ is given by a transition graph TG_A and a transition graph morphism $\mathbf{a} : TG_A \to \mathcal{D}_{D\Gamma}$.

A $D\Gamma - morphism$ $h : \mathbf{A} \to \mathbf{B}$ of $D\Gamma$-transformation systems $\mathbf{A} = (TG_{\mathbf{A}}, \mathbf{a})$ and $\mathbf{B} = (TG_{\mathbf{B}}, \mathbf{b})$ is a transition graph morphism $h : TG_{\mathbf{A}} \to TG_{\mathbf{B}}$ such that $\mathbf{b}(h(c)) = \mathbf{a}(c)$ and $\mathbf{b}(h(t)) = \mathbf{a}(t)$ for each control state c and transition t in $TG_{\mathbf{A}}$ respectively (That means: $\mathbf{b} \circ h = \mathbf{a}$).

With composition of morphisms inherited from the category of graphs this defines the category $\mathbf{ATS(D\Gamma)}$, as shown in [Gro99a].

$$\begin{tikzcd}
TG_{\mathbf{A}} \arrow[rr, "h"] \arrow[dr, "\mathbf{a}"'] & {} \arrow[d, phantom, "="] & TG_{\mathbf{B}} \arrow[dl, "\mathbf{b}"] \\
 & TG_{D\Gamma} &
\end{tikzcd}$$

[2] Tracking relations, that appear in the [Gro01]-version, are redundant in our case, since they would only be identities.

In order to compare transformation systems of different signatures, corresponding signature (or specification) morphisms have to be introduced.

Definition 2.10 (Data Space Specification Morphism)
Given data space specifications $D\Gamma = (\Gamma, M)$ and $D\Gamma' = (\Gamma', M')$ a *data space specification morphism* $\sigma = (\sigma_\Gamma, \sigma_M) : D\Gamma \to D\Gamma'$ is given by an algebraic specification morphism $\sigma_\Gamma : \Gamma \to \Gamma'$ and a family of functions $\sigma_M = (\sigma_{M,w;w'} : M_{w;w'} \to M'_{\sigma_\Gamma(w);\sigma_\Gamma(w')})_{w,w' \in S^*}$. It induces a *data restriction functor* $V_\sigma : \mathbf{ATS(D\Gamma')} \to \mathbf{ATS(D\Gamma)}$ by

$$V_\sigma(TG_{\mathbf{A}'}, \mathbf{a}') = (TG_{\mathbf{A}'}, \mathbf{a}) \text{ with}$$

$$a(c') = V_{\sigma_\Gamma}(\mathbf{a}'(c')) \text{ and } \mathbf{a}(t') = \sigma_M^{-1}(a'(t')) \text{ for all } c', t' \text{ in } TG_{\mathbf{A}'}$$

$$\begin{tikzcd}
\Gamma \arrow[r, "\sigma_\Gamma"] & \Gamma' \\
M \arrow[r, "\sigma_M"] & M'
\end{tikzcd}
\qquad
\begin{tikzcd}
 & TG_{A'} \arrow[dl, dotted, "a"'] \arrow[d, "a'"] \\
\mathcal{D}_{D\Gamma} & \mathcal{D}_{D\Gamma'} \arrow[l, "(V_{\sigma_\Gamma}, \sigma_M^{-1})"]
\end{tikzcd}$$

Note that σ_M^{-1} maps action sets to the empty transition set, if the method names are not in the image of σ_M.

The data restriction functor does not alter the transition graph of a $D\Gamma$-system, i.e., it does not restrict the behavior. It only hides the parts of the data states and action sets that are not visible w.r.t. the smaller signature.

Definition 2.11 (Transformation System Morphism)
Let $\mathbf{A} = (TG_A, \mathbf{a})$ be a $D\Gamma$-transformation system and $\mathbf{B}' = (TG_{B'}, \mathbf{b}')$ a $D\Gamma'$-transformation system. A morphism $\mathbf{h} = (\sigma, h) : \mathbf{A} \to \mathbf{B}'$ is given by a data space specification morphism $\sigma : D\Gamma' \to D\Gamma$ and a morphism $h : V_\sigma(\mathbf{A}) \to \mathbf{B}'$ of $D\Gamma$-transformation systems according to Definition 2.9.

$$A \overset{\mathbf{h}}{\Longrightarrow} B' \ : \qquad D\Gamma \xleftarrow{\ \sigma\ } D\Gamma'$$

$$\begin{tikzcd}
TG_A \arrow[r, "h"] \arrow[d, "\mathbf{a}"'] & TG_{B'} \arrow[d, "\mathbf{b}'"] \\
\mathcal{D}_{D\Gamma} \arrow[r, "(V_{\sigma_\Sigma}, \sigma_M^{-1})"'] & \mathcal{D}'_{D\Gamma}
\end{tikzcd}$$

This means that (σ, h) is a transformation system morphism iff

$$\mathbf{b}' \circ h = (V_\sigma, \sigma_M^{-1}) \ \circ \ \mathbf{a}.$$

Theorem 2.12 (Category ATS of transformation systems)
Transformation systems and transformation system morphisms yield a category together with the composition operation $\circ$ given componentwise by:

$$(\sigma_1, h_1) \circ (\sigma_2, h_2) = (\sigma_2 \circ \sigma_1, h_1 \circ h_2).$$

Proof: This has been shown in [Gro99a].
Algebra transformation systems now can be used to define the the semantics of algebra rewrite specifications.

Definition 2.13 (Semantics of a rewrite specification)
Let $R\Gamma = (\Gamma, M, \mathcal{X}, R)$ with $G = (\Sigma, E)$ be a rewrite specification and let $D\Gamma = (\Gamma, M)$. The semantics $TS(R\Gamma) : (TG_{R\Gamma}, \gamma : TG_{R\Gamma} \to \mathcal{D}_{D\Gamma})$ of $R\Gamma$ is defined as follows.

$$\mathcal{D}_{D\Gamma} := (PAlg(\Gamma), Tf_{D\Gamma}, PAlg(\Gamma), id)$$

is the data space of $D\Gamma$ as given in definition 2.8.
The transition graph $TG_{R\Gamma} = (CS_\Gamma, Tr_{R\Gamma}, CS_\Gamma, id_\Gamma)$ and γ are given by:

- $CS_\Gamma := PAlg(\Gamma)$
- $$\begin{aligned} Tr_{R\Gamma}(A, B) = \ & \{m(a(\vec{x})) \mid \langle m(\vec{x}), r\rangle \in R, \text{ is a match for } r \text{ in } A \\ & \ \text{and } m(a(\vec{x})) : A \Rightarrow B \text{ is a rewrite step wrt. } R\Gamma\} \\ & \cup \quad \{id_A \mid A \in CS_\Gamma\} \end{aligned}$$
- $id_\Gamma(A) = id_A$
- $\gamma(A) = A$
- $\gamma(A \xrightarrow{m(a(\vec{x}))} B) = \{m(a(x))\} \ \ (\in Tf_{D\Gamma}(A, B))$

  $\gamma(id_{st}) = \emptyset \ \ (\in Tf_{D\Gamma}(st, st))$

Notation 2.14 (Methods)
We shall use in the following chapters the definitions as given above, with one exception: the set of methods, which is indexed in the above definitions with two words of method parameters ($M = (M_{w;w'})_{w,w' \in S^*}$, see definition 2.1),

will be indexed with only one word. The reason is that all our matches map method parameters to the algebra from which the rule application starts (and w' stands for parameters to be mapped to the algebra resulting after the application of the rule). Thus we shall have method sets given by $M = (M_w)_{w \in S^*}$.

2.2 Transformation System Semantics of Place-Transition Nets

In this section we give a transformation system that represents the behavior of a place-transition net (P/T net) with capacity. We chose to consider place-transition nets because they are the most general formalism for low-level Petri nets.

We first consider P/T nets that allow only sequential firing of transitions and afterwards the also often encountered version of P/T nets with parallel transition firing.

After defining a transformation system that has the same behavior as a given P/T net, we show the equivalence between the transformation system defined and the semantics of the rewrite specification corresponding to the net for sequentially firing nets. For nets that may fire in parallel we show isomorphism of transition graphs. We shall also show the coincidence of the defined rewrite rules and net firing steps.

Definition 2.15 (Place-Transition Net with Capacity)
A *place-transition net*, *P/T net* for short,

$$N = (P, T, Pre, Post : T \times P \to \mathbb{N}, K : P \to \mathbb{N} \cup \{\omega\})$$

is given by a set P of places, a set T of transitions, functions Pre and $Post$ specifying the pre condition and the post condition for the firing of each transition, and the capacity function K. A *marking* $m : P \to \mathbb{N}$ specifies a possible state of the net.

Definition 2.16 (Firing)
A transition t of a P/T net $N = (P, T, Pre, Post, K)$ can fire at marking m if the following conditions are satisfied:

$$
\begin{array}{ll}
m(p) \geq Pre(t,p) & \text{for all } p \in P \\
K(p) \geq (m(p) - Pre(t,p)) + Post(t,p) & \text{for all } p \in P, \text{ with } K(p) \neq \omega
\end{array}
$$

After the firing the resulting marking is

$$m'(p) = (m(p) - Pre(t,p)) + Post(t,p)$$

for each $p \in P$. A short notation for the described firing step is $m\ [t\rangle m'$.

2.2.1 Rewrite Specifications for Place-Transition Nets

In the following construction a translation of P/T nets into rewrite specifications is given. The correctness of the translation with regard to net firing is shown in Lemma 2.24. The correctness with regard to the net transformation system semantics is shown in Theorem 2.25. Net transformation system semantics is introduced in Definition 2.22. Theorem 2.25 states that the transformation system semantics of the sequential net is equivalent to the semantics of its corresponding rewrite specification.

Construction 2.17 (Rewrite Specification of a Place-Transition Net) Given a net $N = (P, T, Pre, Post, K)$ we define in the following the algebra rewrite specification

$$R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}_N, R_N),$$

which is intended to have an equivalent behavior to this net.

1. Signature and equations

   $$
   \begin{array}{llll}
   \Gamma_N = (\Sigma_N, E_N) = nat+ & & & \\
   & opns & : & m_p :\to nat \quad (p \in P) \\
   & eqns & : & [K_p] \geq m_p \quad (p \in P, K(p) \neq \omega)
   \end{array}
   $$

   where for each $k \in \mathbb{N}$ we use the notation $[k] := succ^k(0) \in T_{nat}$.

2. Method signature M_N:

   $$M_\lambda := T; \quad M_{w \neq \lambda} = \emptyset$$

3. Variables:

   $$\mathcal{X}_N = \{x_p : nat \mid p \in P\}$$

4. Rules:

$$R_N := \{\langle t, r(t)\rangle \mid t \in T\}$$

with

$$r(t) = (X, E_{condt}) \Rightarrow (X, E_{lt}) \leftarrow \rightarrow (X, E_{rt})$$

where $X := \{x_p : nat \mid p \in P\}$ and

$$\begin{aligned} E_{condt} = \quad & \{m_p \geq [Pre(t,p)] \mid p \in P\} \cup \\ & \{[K(p)] \geq (m_p - [Pre(t,p)]) + [Post(t,p)] \mid p \in P, K(p) \neq \omega\} \\ E_{lt} = \quad & \{m_p = x_p \mid p \in P\}, \\ E_{rt} = \quad & \{m_p = (x_p - [Pre(t,p)]) + [Post(t,p)] \mid p \in P\}. \end{aligned}$$

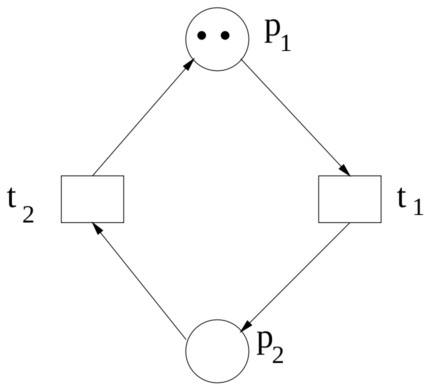
The presentation $P_{lt} = (X_t, E_{lt})$ is functional as required in the definition of rules.

Remark 2.18 (Correspondence between markings and Γ_N-Algebras)
For every marking m of N there is a corresponding Γ_N-algebra $A(m)$, given by:

$$A(m)|_{nat} = Nat \text{ and } m_p^{A(m)} = m(p),$$

where Nat is the algebra of natural numbers, as defined in the appendix. Also, for every Γ_N-algebra A that includes Nat there is a unique marking of N that corresponds to it, given for all $p \in P$ by $m(p) := m_p^A$.

Example 2.19
Let $N = (P, T, Pre, Post, K) := (\{p_1, p_2\}, \{t_1, t_2\}, \{(t_1, p_1) \mapsto 1, (t_2, p_2) \mapsto 1\}, \{(t_1, p_2) \mapsto 1, (t_2, p_1) \mapsto 1\}, \{(p_1, 3), (p_2, 3)\})$ and let the initial marking be $m = \{(p_1, 2)\}$ (we do not specify the function entries that are zero).



The rewrite specification for this net, $R\Gamma_N = (\Gamma_N, M, \mathcal{X}, R)$, is given by:

- $\Gamma = nat+$
  $$\begin{array}{ll} m_{p_1}: & \to nat \\ m_{p_2}: & \to nat \end{array}$$
- The method set is given by $M_\lambda = \{t_1, t_2\}$ and $M_w = \emptyset$ for $w \neq \lambda$
- Rules:

  $$R = \{\langle t_1, r(t_1)\rangle, \langle t_2, r(t_2)\rangle\}$$

  Let X be $\{x_{p_1} : nat, x_{p_2} : nat\}$ in

  $$r(t_1) = (X, E_{condt_1}) \Rightarrow (X, E_{lt_1}) \longleftrightarrow (X, E_{rt_1})$$

  with

  $$\begin{aligned} E_{condt_1} = \; & \{m_{p_1} \geq [1], \\ & m_{p_2} \geq 0 \;\; \text{(this condition is redundant)}, \\ & [3] \geq m_{p_1} - [1], \\ & [3] \geq m_{p_2} + [1] \; \} \end{aligned}$$

  $$\begin{aligned} E_{lt_1} = \; & \{m_{p_1} = x_{p_1}, \\ & m_{p_2} = x_{p_2}\} \\ E_{rt_1} = \; & \{m_{p_1} = x_{p_1} - [1], \\ & m_{p_2} = x_{p_2} + [1]\}. \end{aligned}$$

  The rule $r(t_2)$ is built analogously to $r(t_1)$:

  $$r(t_2) = (X, E_{condt_2}) \Rightarrow (X, E_{lt_2}) \longleftrightarrow (X, E_{rt_2})$$

  with

  $$\begin{aligned} E_{condt_2} = \; & \{m_{p_1} \geq 0 \;\; \text{(this condition is redundant)} \\ & m_{p_2} \geq [1], \\ & [3] \geq m_{p_1} + [1], \\ & [3] \geq m_{p_2} - [1] \; \} \end{aligned}$$

  $$\begin{aligned} E_{lt_2} = \; & \{m_{p_1} = x_{p_1}, \\ & m_{p_2} = x_{p_2}\} \\ E_{rt_2} = \; & \{m_{p_1} = x_{p_1} + [1], \\ & m_{p_2} = x_{p_2} - [1]\}. \end{aligned}$$

- The algebra for m is:

$$\begin{aligned} A(m) = \ & Nat+ \\ & m_{p_1}^{A(m)} = 2, \\ & m_{p_2}^{A(m)} = 0 \end{aligned}$$

The following figure shows possible rewrite steps. These correspond to possible firing steps of transitions. For the sake of briefness and clarity, in the following we use a multiset notation for markings in the figures. A marking $m : P \to \mathbb{N}$ is denoted by $\bigoplus_{p \in P} m(p)p$. The coeficient of every place occuring in the multiset denotes the number of tokens on that place, and a place that does not occur in the multiset has the marking zero.

$$\begin{tikzcd} A(2p_1) \arrow[r, Rightarrow, shift left, "r(t_1)/a_1"] & A(p_1 \oplus p_2) \arrow[l, Rightarrow, shift left, "r(t_2)/a_2"] \end{tikzcd}$$

where $a_1 = \{x_{p_1} \mapsto 2, x_{p_2} \mapsto 0\}$, and $a_2 = \{x_{p_1} \mapsto 1, x_{p_2} \mapsto 1\}$.

Allowing True Concurrency

In the above considerations the semantics of the nets did not include parallel firing of transitions. Parallel firing is an often encountered feature that allows the specification of true concurrency. This motivates us to define algebra rewrite specifications and then algebra transformation systems for nets with this feature. We shall consider both parallel firing of different transitions and of more than one instantiation of the same transition.

Nets with true concurrency allow simultaneous firing of transitions, i.e., the firing steps are determined by the number of times each transition fires in that step. In order to model parallel firing we have to introduce new methods, each representing a possible firing step. The parallel closure defined in [Gro99c] of rewrite steps w.r.t. the net specification $R\Gamma_N$ defined in Construction 2.17 does not suite this purpose because the mechanism verifies conditions separately. For example, it does not add the pre conditions of all transitions of the firing step in order to find out whether the marking is greater than this sum or not. Also, if as a result of a parallel rewrite step two rules of transitions firing in parallel conclude with $m_p = 5$ and $m_p = 3$, the resulting algebra will fulfill the equation $5 = 3$ (instead of adding both numbers of tokens the two transitions put on p to its previous marking)

In order to specify parallel firing we introduce new methods, each of which represents the firing of a combination of transitions. We will use for a net with transition set T the following set M of methods for defining the parallel rewrite specification of N:

$$M_\lambda = \mathcal{M}(T) \setminus \{\lambda\} \text{ and } \; M_w = \emptyset \text{ for } w \neq \lambda$$

where $\mathcal{M}(T)$ is the free commutative monoid over T and the coefficient of a transition t in a firing step $tr = \bigoplus_{t \in T} n_t t$ represents the number of times t is supposed to fire if tr occurs. The empty multiset λ does not represent a firing step, since a firing step includes, according to its definition, at least the firing of one transition.

Construction 2.20 (Rewrite specification of concurrent P/T-Nets) Let $N = (P, T, Pre, Post : T \times P \to \mathbb{N}, K : P \to \mathbb{N})$. The parallel rewrite specification $R\Gamma_{Np} = (\Gamma_{Np}, M_{Np}, \mathcal{X}_{Np}, R_{Np})$ of N is given by

- $\Gamma_{Np} = \Gamma_N$, the specification corresponding to N as in construction 2.17
- $M_{Np,w} = \begin{cases} \mathcal{M}(T) \setminus \{\lambda\} & \text{if } w = \lambda \\ \emptyset & \text{else,} \end{cases}$
- $\mathcal{X}_{Np} = \mathcal{X}_N$, as defined in construction 2.17,
- $R_{Np} = \{\langle tr, r(tr)\rangle \mid tr = \bigoplus_{t \in T} n_t t \in \mathcal{M}(T)\}$ with

  $$r(\bigoplus_{t \in T} n_t t) = (X, E_{condtr}) \Rightarrow (X, E_{ltr}) \leftarrow \rightarrow (X, E_{rtr})$$

  where $X = \{x_p : nat \mid p \in P\}$ and

  $$\begin{array}{ll} E_{condtr} = & \{\geq (m_p,\ [n_t Pre(t,p)])\ , \\ & \geq ([K(p)],\ (m_p - [n_t Pre(t,p)]) + [n_t Post(t,p)]) \\ & \qquad \mid p \in P, K(p) \neq \omega\} \\ E_{ltr} = & \{m_p = x_p \mid p \in P\} \\ E_{rtr} = & \{m_p = (x_p - [n_t Pre(t,p)]) + [n_t Post(t,p)] \mid p \in P\} \end{array}.$$

The correspondence between Γ-algebras and N-markings is the same as in the sequential case (see remark 2.18).

Example 2.21
For the net N in example 2.19 we have the following set of rules:

$$R_{Np} = \{\langle n_{t_1}t_1 \oplus n_{t_2}t_2, r(n_{t_1}t_1 \oplus n_{t_2}t_2)\rangle \mid n_{t_i} \in \mathbb{N}\}$$

as described above. New rewrite steps occur, for example:

$$\begin{tikzcd} A(2p_1) \arrow[r, Rightarrow, bend left=10, "r(2t_1)/a_1"] & A(2p_2) \arrow[l, Rightarrow, "r(2t_2)/a_3"] \end{tikzcd}$$

Recall that there are two denotations for method actualizations, one that gives the method and its parameter actualizations and one that gives the rule and its match. In our case this allows us to refer to rewrite steps $r(t)/a : A(m) \Rightarrow A(m')$ also with the denotation $t : A(m) \Rightarrow A(m')$, since the methods t have no parameters.

2.2.2 Transformation Systems for Place-Transition Nets

The operational semantics of Petri nets are usually defined as their marking graphs. In this subsection we rephrase these as algebra transformation systems by defining the algebra transformation system $TS(N)$ of a P/T net N. We also show how reachability graph semantics can be represented within the transformation system formalism. Finally we formally compare the defined transformation system and the transition system of the rewrite specification of the net. This comparison results in equivalence (transformation system isomorphism), in the sequential case. In the parallel case it results in behaviour equivalence (transition graph isomorphism).

In the following definition we use as a data space specification of the transformation system of a net N the data space specification corresponding to $R\Gamma_N$, the rewrite spefication of the net. This is legitimate, since there is a one-to-one correspondence between algebras the *nat*-part of which is actualized by Nat and markings.

Definition 2.22 (Algebra transformation system of a P/T-net)
Let $N = (P, T, Pre, Post, K)$ be a P/T Net and let $R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}_N, R_N)$ its rewrite specification as introduced in Construction 2.17 if the net allows only sequential transition firing or in 2.20 if it allows parallel firing. The algebra transformation system

$$TS(N) = (TG_N, n : TG_N \to \mathcal{D}_{D\Gamma_N})$$

for N is given as follows.

- $\mathcal{D}_{D\Gamma_N} := (PAlg(\Gamma_N), Tf_{D\Gamma_N}, PAlg(\Gamma_N), id)$ is the data space induced by the data space specification $D\Gamma_N = (\Gamma_N, M_N)$, as shown in Definition 2.8

- TG_N, the transition graph for N, reflects the marking graph of the net:

  $$TG_N = (CS_N, Tr_N, CS_N, id_N : CS_N \to Tr_N)$$

  where
  $*$ $CS_N = \{m : P \to N \mid m(p) \leq K(p)\}$,
  $*$ in order to give the transition set Tr_N we consider the sequential and the parallel case separately:
  – for the sequential case
  $$\begin{aligned} Tr_N(m, m') = \ & \{t \in T \mid m[t\rangle m' \text{ is a firing step in } N\} \\ & \cup \{id_{st} | st \in CS_N\} \end{aligned}$$
  – for the parallel case
  $$\begin{aligned} Tr_N(m, m') = \ & \{\bigoplus_{t\in T} k_t t \in T \mid m[\bigoplus_{t\in T} k_t t\rangle m' \text{ is a firing step in } N\} \\ & \cup \{id_{st} | st \in CS_N\} \end{aligned}$$

  Note that in both cases $Tr_N(m, m') = Tr_{R\Gamma_N}(A(m), A(m'))$ for all states $m \in CS_N$
  $*$ $id_N(st) = id_{st}$ for all $st \in CS_N$

- $n : TG_N \to \mathcal{D}_{D\Gamma_N}$
  For $m \in CS_N$

  $$n(m) = A(m).$$

  For $tr \in Tr(m, m')$ let $\bigoplus_{t\in T} k_t t = tr$ in

  $$n(tr) = \{k_t t \mid t \in T\} \in Tf_{D\Gamma_N}(n(m), n(m')).$$

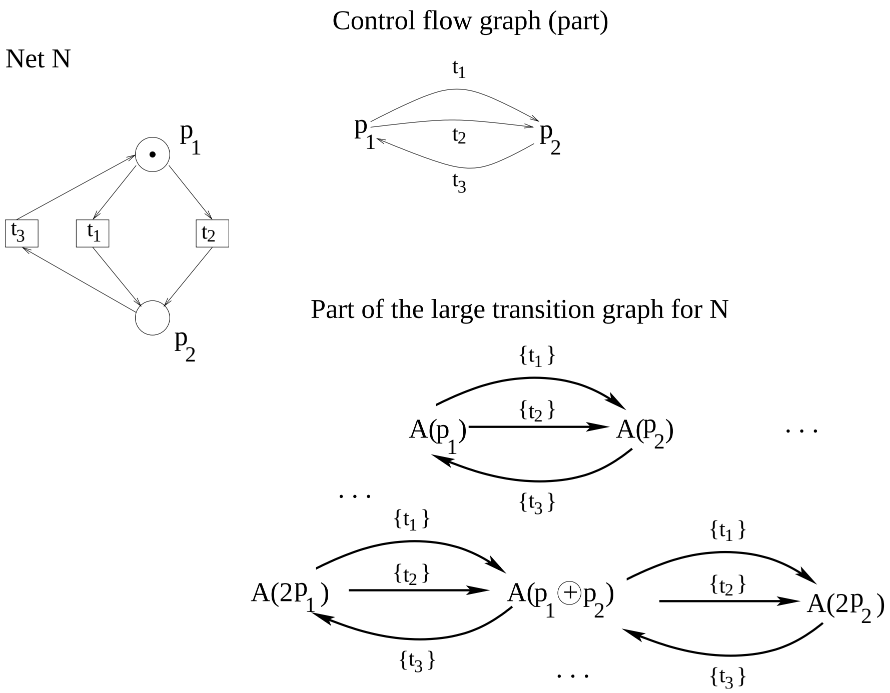
  Note that in the sequential case $tr \in T$, $\bigoplus_{t\in T} k_t t = t$, and $n(tr) = \{tr\}$

In the case of low-level nets we have only methods without parameters (see subsection 2.2.1) and $Act_{D\Gamma_N}(C, D) = M$ for any Γ-algebras C and D; if no parallel transition firing is allowed the set of methods equals the set of transitions and for any algebras C, D: $Act_{D\Gamma_N}(C, D) = M = T$ and, consequently, $\mathcal{P}(Act_{D\Gamma_N}(C, D)) = \mathcal{P}(T)$. If parallel transition firing is allowed, then $\mathcal{P}(Act_{D\Gamma_N}(C, D)) = \mathcal{P}(\mathcal{M}(T) \setminus \{\lambda\})$.

Remark 2.23 (Reachability)
A rewrite specification does not specify an initial state, and a P/T net does not necessarily have an initial marking assigned to it. The marking graph of a net includes all possible reachability graphs of it. By reducing the number of control states and by giving only one initial state, the transformation system $TS^r(N) = (TG^r_N, n^r : TG^r_N \to \mathcal{D}_{D\Gamma_N})$ reflecting the reachability semantics of a net N with start marking m_0 can be constructed as follows:

- $TG^r_N = (Reach(N), Tr_N|_{Reach_N \times Reach_N}, \{m_0\}, id_N|_{Reach_N})$
  where $Reach(N)$ is the set of reachable (N, m_0)-markings,
- $n^r = n|_{TG^r_N}$.

The figure shows an example.



The following lemma shows that there is a one-to-one correspondence between firings of a sequential net and rewritings of the corresponding rewrite specification, which was defined in Construction 2.17. It will be used in the proof of Theorem 2.25.

Lemma 2.24 (Behaviour equivalence for sequential nets)
A firing step $m[t\rangle m'$ of a transition t of a net $N = (P, T, Pre, Post, K)$ is possible if and only if there is a rewrite step $r(t)/a : A(m) \Rightarrow A(m')$ with regard to the rewrite specification $R\Gamma_N$ of the net N.

Proof: Let $m[t\rangle m'$ be a firing step in N. According to Definition 2.16: $Pre(t,p) \geq m(p)$ and $K(p) \geq (m - Pre(t,p)) + Post(t,p)$ for each $p \in P$. Thus, according to the definition of $A(m)$ the equations E_{condt} are satisfied in $A(m)$. For the match $a : \{x_p \mid p \in P\} \to \mathbb{N}$ given by $a(x_p) = m(p)$, the equations in E_{lt} are also satisfied, so there is a rewrite step: $r(t)/a : A(m) \Rightarrow A'$ w.r.t. $R\Gamma_N$. According to Definition 2.5 of rewriting we have:

$$A'_S = A_S = A(m')_S$$
$$A'_E = (A_E - \{m_p = m(p) \mid p \in P\}) + \{m_p = m(p) - Pre(t,p) + Post(t,p) \mid p \in P\}$$

Thus $A' = PAlg_{\Gamma_N}((A'_S, A'_E)) = A(m')$.
Suppose $r(t)/a : A(m) \Rightarrow A'$. Then the equations E_{condt} are satisfied by $A(m)$ and, consequently the firing of transition t is possible in N at marking m. Let m' be the marking resulting from the firing step $m[t\rangle$. According to the first part of this proof we have $A' = A(m')$.

An important result of this section, the semantical equivalence of transformation systems of sequential P/T-nets and of their corresponding rewrite specifications, can now be derived from Lemma 2.24.

Theorem 2.25 (Isomorphism of transformation systems for sequential nets)
The transformation system of net N is isomorphic to the transformation system of the rewrite specification of net N in the category of ATS:

$$TS(N) \cong TS(R\Gamma_N).$$

Proof: Let $(TG_N, n : TG_N \to \mathcal{D}_{D\Gamma_N}) := TS(N)$ and $(TG_{R\Gamma_N}, \gamma : TG_{R\Gamma_N} \to D_{D\Gamma(R\Gamma_N)}) := TS(R\Gamma_N)$. Since $D\Gamma(R\Gamma_N) = D\Gamma_N$ it is sufficient to find a transition graph isomorphism $h : TG_{R\Gamma_N} \to TG_N$, that is compatible with n and γ. In order to show isomorphism we will also define its inverse h^{-1}. According to Lemma 2.24 for the sequential case and because $Tr_N(m, m') = Tr_{R\Gamma_N}(A(m), A(m'))$ for all $m, m' \in CS_N$, we have

$$TS(N) = TS(R\Gamma_N)[TG_{R\Gamma_N}/(TG_{R\Gamma_N}[A(m)/m])].$$

Let $h(A(m)) := m$ and $h^{-1}(m) := A(m)$. Let the transition components of h and h^{-1} be identities on M_N.
Now we show that h and h^{-1} are $D\Gamma_N$ morphisms:

$$n \circ h(A(m)) = n(m) = A(m) = \gamma(A(m))$$
$$n \circ h(t) = n(t) = \{t\} = \gamma(t)$$

$$\gamma \circ h^{-1}(m) = \gamma(A(m)) = A(m) = n(m)$$
$$\gamma \circ h^{-1}(t) = \gamma(t) = \{t\} = n(t)$$

Since $h \circ h^{-1} = id_{TG_N}$ and $h^{-1} \circ h = id_{TG_{R\Gamma_N}}$ the transformation systems $TS(N)$ and $TS(R\Gamma_N)$ are isomorphic.

The following lemma shows that there is a one-to one correspondence between firings of a parallel net and rewritings of its corresponding rewrite specification. It will be used in the proof of Theorem 2.27

Lemma 2.26 (Behaviour Equivalence for parallel nets)
A firing step $m[\bigoplus_{t\in T} n_t t > m'$ is possible in a net N if and only if there is a rewrite step $r(\bigoplus_{t\in T} n_t)t/a : A(m) \Rightarrow A(m')$ in the algebra rewrite specification $R\Gamma_{Np}$ corresponding to the net N with true concurrency.

Proof: The proof is analogous to the proof of Theorem 2.24.

Theorem 2.27 (Transition Graph Equivalence for parallel nets)
The transition graph of the transformation system of net N is isomorphic to the transition graph of the semantics of $R\Gamma_N$.

Proof:
We find an isomorphism $h : TG_{R\Gamma_N} \to TG_N$ between the transition graphs $TG_{R\Gamma_N}$ and TG_N. To show that it's indeed an isomorphism, we also define its inverse h^{-1}. According to Lemma 2.26 and because $Tr_N(m, m') = Tr_{R\Gamma_N}(A(m), A(m'))$ for all $m, m' \in CS_N$ we have:

$$TG_N = TG_{R\Gamma_N}[A(m)/m].$$

Let $h(A(m)) := m$ and $h^{-1}(m) := A(m)$. Let the transition components of h and h^{-1} be identities on M_N. Obviously, h is an isomorphism.

Discussion 2.28 (Transformation System Morphism)
We could not find a transformation system morphism in the parallel case from $TS(R\Gamma_N)$ to $TS(N)$ or in the opposite direction because the same firing steps in the transition graph correspond to different method sets in the data space. For example,

$$\gamma(m \xrightarrow{\oplus k_t t} m') = A(m) \xRightarrow{\oplus k_t t} A(m'), \text{ and}$$

$$\mathbf{n}(m \xrightarrow{\oplus k_t t} m') = A(m) \xRightarrow{\{k_t t \ | \ k_t t \in tr\}} A(m'), \text{ where } tr = \oplus k_t t.$$

It is impossible to have a (σ, h) to be compatible with n and γ. This is implied by the characteristics σ_M needs to have such that σ_M^{-1} maps $\{\oplus k_t t\}$ to $\{k_t t\}$ or $\{k_t t\}$ to $\{\oplus k_t t\}$ (i.e., if σ_M goes in the other direction)[3].

Discussion 2.29 (Transformation System of a Place-Transition Net)
As a data space specification for a net N we took the two components of its rewrite specification (Γ_N, M_N), where the algebraic specification Γ_N contains a specification *nat* of natural numbers as subspecification. For the purpose of defining a transformation system it is not absolutely necessary to maintain the operations from the specification of *nat*, which would make our signature smaller and our data space larger.

This observation is not surprising, since the aim of a rewrite specification is to allow the formal description of rules, and the aim of a data space specification is only to give the possibility of addressing the right data states (the right algebras) by the morphisms from the transition graph. In the case of transformation systems, the transition graph contains all necessary control flow information, which the rewrite specification must specify by its rules. Thus, a rewrite specification is very likely to need more operations in its signature than the signature of a data space specification.

This leads to the conclusion that mapping methods to *unsuitable* method sets in the defined semantics $TS(S)$ of a system S, as we did for parallel place-transition nets, is not the only reason for which the semantics of a rewrite specification of S might not be isomorphic to the semantics of S. Another reason is that the semantics of a *smallest* possible rewrite specification for

[3]In the first case this would imply that all $k_t t$-transitions would be mapped by σ to $\oplus k_t t$, thus $\sigma(5t1) = 5t1 = 5t1 \oplus 5t2 = 5t1 \oplus 4t2 = \ldots$, which is impossible, and in the second case the only preimage of $k_t t$ needs to be $\oplus k_t t$, thus $V(\{2t_1\}) = \{2t_1\}$, $V(\{3t_2\}) = \{3t_2\}$, which implies $V(\{2t_1, 3t2\}) = \{2t_1, 3t2\} \neq \{2t_1 \oplus 3t_2\}$, contradiction to the assumption that there was an adequate V.

a system does not need to be isomorphic to the transformation system with the shortest possible data state specification in the category of transformation systems. In our case, this would follow from the fact that there is no specification morphism from the specification of the rewrite specification to the data state specification described in this discussion. For example, there is no way to map the operations $+$ and $-$.

However, isomorphism of transition graphs and of the method sets would be provided, and there would be a bijective mapping between the algebras addressed by the two transformation systems.

We maintained entire nat-specification since this does not imply significant disadvantages, while at the same time allowing proof of isomorphism for the sequential case.

Remark 2.30 (Faithfullness)
Although the formalism of transformation systems is very general, a sequential net can be entirely reconstructed (up to isomorphism) from the transformation system $TS_N = (TG_N, n : TG_N \to D_{D\Gamma_N})$ representing it.

The set of places is isomorphic to the set of nat-constants $P := \Gamma_N \setminus nat$. The capacity of a place $p \in P$ is $max(p^A)$ where $A \in n(CS_N)$, and CS_N is the set of states of TS_N.

The transition set is the set of methods. The pre and post conditions can be determined from the transition graph: $Pre_N(t, p)$ is the lowest p^A with $A \in n(CS_N)$ with the property that there is a state $B \in n(CS_N)$ such that $t : A \Rightarrow B$ belongs to the image of n. The post condition can be derived immediately from that: $Post_N(t, p) = p^B - p^A$.

Exceptions are the cases in which there is no firing of one transtion t in TG_N (because of the capacity condition, that can be lower than the pre condition of t for one place).

A parallel net can also be reconstructed, if the condition that the method naming allows identification of the monoidal structure is satisfied. This means that, if the name of a method that represents the firing step $t_1 \oplus t_2$ is, for example τ, this would make determination of the number of transitions and their identification among the methods impossible.

2.3 Transformation System Semantics of Algebraic High-Level nets

This section gives a representation of the semantics of typed algebraic high-level nets (AHL nets) through rewrite specifications and transformation systems. We show that the transformation system defined as net semantics is equivalent to the semantics of the rewrite specification of the net.

Once more, we first consider nets with sequential firing of transitions and afterwards look at the more general case of nets with true concurrency.

We chose to consider nets without capacity because nets with capacity can be simulated by nets without capacity [4].

We consider *typed* AHL-Nets because this allows the inclusion of the net signature in the rewrite specification and in the transformation system. Markings of untyped AHL-nets may include tokens of different sorts within one term, in fact they are multisets over the union of the carrier algebra sets. Since we have to specify firing conditions through rules, there must be one sort including the constants representing place markings, which results in the necessity of a sort including terms of all sorts of the signature of the AHL net. Since in algebraic signatures terms belong to only one sort, there is no way of doing this while at the same time including the signature of the net (with its sorts) in the rewrite specification.

Definition 2.31 (Typed AHL Net)
An AHL-net $N(A)$ is given by an AHL-net scheme

$$N = (\Sigma, P = \bigcup_{s \in S} P_s, T, Pre, Post : T \times P \to \mathcal{M}(T_\Sigma(X)), Cond : T \to Eqns_\Sigma(X))$$

and by a $\Sigma-$Algebra A, with $\Sigma = (S, Op, E)$ an algebraic specification with a finite set of sorts, X a set of variables, P_s $(s \in S)$ finite pairwise disjoint sets of places, T a finite set of transitions. The functions Pre, $Post$ and $Cond$ specify the firing conditions for each transition. A place $p \in P_s$ can hold only token belonging to A_s.

[4]For example by adding to every place p that can contain tokens of sort s a complementary twin place p' containing dummy s–terms the number of which equals the desired capacity $K(p)$ minus the number of tokens on p and by adding to $Pre(t)$ the mapping $p' \mapsto |Post(t,p)|dummy$ and to $Post(t)$ the mapping $p' \mapsto |Pre(t,p)|dummy$ (here $|mset|$ denotes the number of elements of the multiset $mset$; the latter variant also holds for concurrent nets.)

A marking of an AHL-Net $N(A)$ is a function $m : P \to \bigcup_{s \in S} \mathcal{M}(A_s)$ with the property that $m(p) \in \mathcal{M}(A_s)$ if $p \in P_s$. Furthermore for all transitions t and places p each term τ in $Pre(t,p)$ and $Post(t,p)$ has the same sort as p, i.e., $\tau \in Terms_{\Sigma,s}(X)$ if $p \in P_s$.

Notation 2.32 (sort(p))
Since the sets P_s for $s \in S$ are disjoint, each place p can be assigned to only one sort $s =: sort(p)$, which has the property $p \in P_s$.

Definition 2.33 (Firing of AHL-Nets)
A transition t can fire at marking m in the algebraic high-level net N if its pre condition is satisfied, i.e. if there is an assignment $a : Vars(t) \to A_{sorts(Vars(t))}$ where $Vars(t)$ is an ordered set containing the variables occuring in $Pre(t,p)$, $Cond(t)$ and $Post(t,p)$, such that

$$Pre(t,p)^A/a \leq m(p) \text{ and}$$
$$Cond(t)^A/a.$$

The follow-up marking m' that occurs in N after the firing of transition t with the assignment a at m, is given for all $p \in P$ by:

$$m'(p) = (m(p) \ominus Pre(t,p)^A/a) \oplus Post(t,p)^A/a$$

Here, $\oplus$ and $\ominus$ are the usual operations on multisets. The above firing step can also be refered to as $m[t/a\rangle m'$.

2.3.1 Rewrite Specifications for AHL-Nets

In this subsection we give rewrite specifications for AHL nets.

Construction 2.34 (Algebra Rewrite Specification of an AHL Net)
Given an AHL net scheme $N = (\Sigma = (S, Op, E), P, T, Pre, Post : T \times P \to \mathcal{M}(T_\Sigma(X)), Cond : T \to Eqns_\Sigma(X))$ in the following we will define an algebra rewrite specification

$$R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}_N, R_N)$$

which will have an equivalent behaviour to this net.

- Signature and equations:

$$\begin{aligned}\Gamma_N = (\Sigma_N, E_N) = \quad & (S, Op, E)+ \\ & \underset{s \in S}{+}\ mset(s)\ + \\ & opns\ :\ m_p :\to mset(s)\ \forall p \in P_s,\ \forall s \in S\end{aligned}$$

We assume without loss of generality that the specifications added above are disjoint. If this is not the case the disjoint union should be used.

- Method signature M_N

$$M_N = (M_w)_{w \in S^*},\ M_w = \{t \in T \mid sorts(Vars(t)) = w\}$$

- Variables

$$\mathcal{X} = X \cup \{x_p :\ mset(sort(p)) \mid p \in P\}$$

We assume that the above sets are disjoint, otherwise the disjoint union should be used.

- Rules

$$R = \{(t, r(t)) \text{ for } t \in T\}$$

with

$$r(t) = (X_t, E_{condt})\ \Rightarrow\ (X_t, E_{lt}) \leftarrow\rightarrow (X_t, E_{rt})$$

where $X_t = Vars(t) \cup \{x_p \mid p \in P\}$,

$$\begin{aligned} E_{condt} = \quad & \{Cond(t)\} \cup \\ & \{\geq (m_p, Pre(t,p)) \mid \forall p \in P\}, \\ E_{lt} = \quad & \{m_p = x_p \mid p \in P\} \\ E_{rt} = \quad & \{m_p = (x_p \ominus Pre(t,p)) \oplus Post(t,p) \mid p \in P\}. \end{aligned}$$

Definition 2.35 (Γ_N-algebra corresponding to a marking and a $\Sigma-$Algebra) For a $\Sigma-$Algebra A and a marking $m : P \to \bigcup_{s \in S} \mathcal{M}(A_s)$ the construction of the rewrite specification corresponding to $N(A)$ can be completed in the following way:

The Σ_N-Algebra $A(m)$ is given by

$$
\begin{array}{lll}
A(m)_s & = A_s & (s \in S) \\
A(m)_{mset(s)} & = \mathcal{M}(A_s) & (s \in S) \\
op^{A(m)} & = op^A & (op \in OP) \\
m_p^{A(m)} & = M(p) & (p \in P) \\
\geq, \oplus, \ominus & & \text{usual operations on multisets}
\end{array}
$$

The operators $\geq, \oplus, \ominus$ are overloaded, as they are used for multisets on different sorts, and both in the specification and in the algebras. It will always be clear from the context which operation is meant.

The correctness of the above translation of AHL-nets into rewrite specifications with regard to the equivalence to firing steps is shown in Lemma 2.40.

The following example illustrates the way a rewrite specification represents the behaviour of a net.

Example 2.36 (Umbrellas)
In the following we define a simple AHL net N, give the rewrite specification for it and show how firing is reflected in rewrite steps. The transition lh(leave home) describes how a person leaves home and takes with him his umbrella, but not the umbrella of another person. The places h_m, a_m contain token representing man at home and man away from home. Also, umbrellas can be in the house (on the place h_u), or away (on the place a_u).

Let the AHL net $N(A)$ be given by the net scheme $N = (\Sigma, P, T, Pre, Post, Cond)$ with

$$
\begin{array}{lll}
\Sigma = & sorts: & man, umbrella \\
 & opns: & belongs\_to : umbrella \to man
\end{array}
$$

$P_{umbrella} = \{h_u, a_u\}$
$P_{man} = \{h_m, a_m\}$
$T = \{lh\}$
$Pre(lh) = \{h_u \mapsto y, h_m \mapsto x, a_u \mapsto \lambda, a_m \mapsto \lambda\}$
$Post(lh) = \{h_u \mapsto \lambda, a_u \mapsto y, a_m \mapsto x\}$
$Cond(lh) = \{belongs\_to(y) = x\}$
and by the algebra A with
$A_{man} = \{Peter, Joe\}$
$A_{umbrella} = \{u_P, u_J\}$
$belongs\_to = \{u_P \mapsto Peter, u_J \mapsto Joe\}$.

The rewrite specification for N is given by $R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}, R)$ where:

- Signature and equations

$$
\begin{array}{lll}
\Gamma_N = & mset(man) + & mset(umbrella) \\
& opns: & belongs\_to : umbrella \to man \\
& & m_{h_m} :\to mset(man) \\
& & m_{a_m} :\to mset(man) \\
& & m_{h_u} :\to mset(umbrella) \\
& & m_{a_u} :\to mset(umbrella)
\end{array}
$$

Note that the specifications $mset(man)$ and $mset(umbrella)$ include the sorts man and $umbrella$, and multiset operations as described in the appendix.

- Method signature
$M_{N,man\ umbrella} = \{lh\}$
$M_{N,w} = \emptyset$ for $w \neq man\ umbrella$

- Variables
$\mathcal{X}_N = \{x : man, y : umbrella, x_{h_m} : mset(man),$
$x_{a_m} : mset(man), x_{h_u} : mset(umbrella), x_{a_u} : mset(umbrella)\}$

- Rules
$R_N = \{\langle lh, r(lh)\rangle\}$ with
$r(lh) = (X_{lh}, E_{condlh}) \Rightarrow (X_{lh}, E_{llh}) \longleftrightarrow (X_{lh}, E_{rlh})$ where:$X_{lh} = \mathcal{X}$ and

$$
\begin{array}{ll}
E_{condlh} = \{ & belongs\_to(x) = y, \\
& m_{h_m} \geq x, \\
& m_{h_u} \geq y, \\
& m_{a_m} \geq \lambda, \\
& m_{a_u} \geq \lambda \ \}
\end{array}
$$

$$
\begin{array}{ll}
E_{llh} = \{ & m_{h_m} = x_{h_m}, \\
& m_{h_u} = x_{h_u}, \\
& m_{a_m} = x_{a_m}, \\
& m_{a_u} = x_{a_u} \}
\end{array}
$$

$$
\begin{array}{ll}
E_{rlh} = \{ & m_{h_m} = x_{h_m} \ominus x, \\
& m_{h_u} = x_{h_u} \ominus y, \\
& m_{a_m} = x_{a_m} \oplus x, \\
& m_{a_u} = x_{a_u} \oplus y \ \}
\end{array}
$$

Let $m = \{h_m \mapsto Peter \oplus Joe, h_u \mapsto u_h \oplus u_j\}$ be a marking. The places that do not appear in the marking are understood to have the marking λ.

The Γ_N-algebra $A(m)$ is given by:

$A(m) = A \cup \{m_{h_m}^{A(m)} = Joe \oplus Peter, m_{h_u}^{A(m)} = u_h \oplus u_j,$
$m_{a_u}^{A(m)} = \lambda, m_{a_m}^{A(m)} = \lambda\}$

The following figure shows two possible rewrite steps:

$$\begin{tikzcd}
 & A(h_u \mapsto u_J, h_m \mapsto Joe, a_u \mapsto u_P, a_m \mapsto Peter) \arrow[d, Rightarrow, "r(lh)/a_2"] \\
A(h_u \mapsto u_P \oplus u_J, h_m \mapsto Joe \oplus Peter) \arrow[ur, Rightarrow, "r(lh)/a_1"] & A(a_u \mapsto u_P \oplus u_J, h_m \mapsto Joe \oplus Peter)
\end{tikzcd}$$

Here, $a_1 = \{x \mapsto Peter, y \mapsto u_P\} \cup \{x_{h_m} \mapsto Joe \oplus Peter, x_{h_u} \mapsto u_P \oplus u_J, x_{a_u} \mapsto \lambda, x_{a_m} \mapsto \lambda\}$ and $a_2 = \{x \mapsto Joe, y \mapsto u_J\} \cup \{x_{h_m} \mapsto Joe, x_{h_u} \mapsto u_J, x_{a_u} \mapsto u_P, x_{a_m} \mapsto Peter\}$

Allowing true concurrency

Analoguously to the sequential case we now define AHL-nets with true concurrency. Similar to the low-level nets we use multisets of transitions to describe the new method set.

Note that, unlike the low-level nets, the transitions that fire more than once in a firing step may not be identical to each other with regard to their effect, since they can have different matchings for their variables. Let in the following for $tr = \bigoplus_{i=1}^{k} n_i t_i \in \mathcal{M}(T) \setminus \{\lambda\}$ the variable set for tr be given by

$$Vars(tr) = \biguplus_{i=1}^{k} (\biguplus_{j=1}^{n_i} Vars(t_i)).$$

Furthermore, let

$$sorts(Vars(tr)) = Vars(t_1)^{n_1} \ldots Vars(t_k)^{n_k}.$$

Construction 2.37 (Rewrite specification for parallel AHL-nets)
Let $N = \{\Sigma, P = \bigcup_{s \in S} P_s, T, Pre, Post : T \times P \to \mathcal{M}(T_\Sigma(X)), Cond : T \to Eqns_\Sigma(X)\}$ and $R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}_N, R_N)$ its sequential rewrite specification, as given in construction 2.34. The parallel rewrite specification $R\Gamma_{Np} = (\Gamma_{Np}, M_{Np},, \mathcal{X}_{Np}, R_{Np})$ is given by:

- $\Gamma_{Np} = \Gamma_N$
- $M_{Np,w} = \{tr \mid tr \in \mathcal{M}(T) \setminus \{\lambda\}, w = sorts(Vars(tr))\}$
- $\mathcal{X}_{Np} = \{x_i \mid x \in X,\ i \in \mathbb{N}\} \ \cup\ \{x_p \mid p \in P\}$
- $R_{Np} = \{(tr, r(tr)) \mid tr \in \mathcal{M}(T)\}$ where:

$$r(tr) = (X_t, E_{condtr}) \ \Rightarrow\ (X_t, E_{ltr}) \leftarrow \rightarrow (X_t, E_{rtr})$$

with $X_{tr} = Vars(tr) \cup \{x_p \mid p \in P\}$,

$$\begin{aligned} E_{condtr} = \ & \{ \bigcup_{t \in tr} Cond(t), \\ & \geq (m_p, \bigoplus_{t \in tr} Pre(t,p))\ \forall p \in P\ \}, \\ E_{lt} = \ & \{m_p = x_p \mid p \in P\} \\ E_{rt} = \ & \{m_p = x_p \ominus (\bigoplus_{t \in tr} Pre(t,p)) \oplus \bigoplus_{t \in tr} Post(t,p) \mid p \in P\}. \end{aligned}$$

The correctness of this construction with regard to the behaviour of the net is shown in Lemma 2.41

2.3.2 Transformation systems for AHL-Nets

In this subsection we shall give the algebra transformation system $TS(N)$ of an AHL-net N based on the usual marking graph semantics of Petri nets. We shall also show how reachability graph semantics can be represented within the transformation system formalism. Finally we shall show that the defined transition system and the transition system of the rewrite specification of the net are equivalent.

Definition 2.38 (Algebra transformation system of an AHL-net) Let $N(A) = \{\Sigma, P, T, Pre, Post, Cond\}(A)$ be an AHL-net and let $R\Gamma_N = (\Gamma_N, M_N, \mathcal{X}_N, R_N)$ be its rewrite specification as shown in construction 2.34 if the net allows only sequential transition firing and in construction 2.37 if it allows parallel firing. The transition system

$$TS(N) = (TG_N, n : TG_N \rightarrow \mathcal{D}_{D\Gamma_N})$$

for N is defined as follows:

- $\mathcal{D}_{D\Gamma_N} := (PAlg(\Gamma_N), Tf_{D\Gamma_N}, PAlg(\Gamma_N), id)$ is the data space induced by the data space specification $D\Gamma_N = (\Gamma_N, M_N)$, as shown in definition 2.8

- TG_N, the transition graph for N, reflects the marking graph of the net:

$$TG_N = (CS_N, Tr_N, CS_N, id_N : CS_N \to Tr_N)$$

where
$*$ $CS_N = [P \to \bigcup\limits_{s\in S} \mathcal{M}(A_s)]_S = \{m : P \to \bigcup\limits_{s\in S} \mathcal{M}(A_s) \mid m(p) \in A_s \text{ if } p \in P_s\}$,
$*$ For the transition set Tr_N we consider the sequential and the parallel case separately:
– for the sequential case
$Tr_N(m, m') = \{t(a(\vec{x}))| \; m[t/a\rangle \text{ is a sequential firing step in } N(A)\} \cup \{id_{st} \mid st \in CS_N\}$
where $\vec{x_t} = Vars(t)$.
– For the parallel case
$Tr_N(m, m') = \{tr(a(\vec{x}_{tr}))| \; m[tr/a\rangle \text{ is a parallel firing step in } N(A)\} \cup \{id_{st} \mid st \in CS_N\}$
where $\vec{x_{tr}} = \biguplus\limits_{t\in tr} Vars(t)$
Note that $Tr_N(m, m') = Tr_{R\Gamma_N}(A(m), A(m'))$ for all states $m \in CS_N$
$*$ $id_N(st) = id_{st}$ for all $st \in CS_N$

- $n : TG_N \to \mathcal{D}_{D\Gamma_N}$
For $m \in CS_N$ let

$$n(m) = A(m).$$

For $t \in Tr(m, m')$ let :

$$n(t(a(\vec{x}))) = \{t(a(\vec{x}))\} \in Tf_{D\Gamma_N}(n(m), n(m')).$$

The addressed method set, $\{t(a(\vec{x}))\}$, exists in $Tf_{D\Gamma_N}(A(m), A(m'))$ since $a(\vec{x_t}) \in A(m)_{sorts(Vars(t))}$ and since, according to definition 2.8 and notation 2.14,

$$Tf_{D\Gamma_N}(A(m), A(m')) = \{tr(\overline{a}) | tr \in M_N, \overline{a} \in A(m)_{sorts(Vars(tr))}\}$$

Remark 2.39 (Reachability)
We now give the transformation system reflecting the reachability semantics of net $N(A)$ with a start marking m_0

- $TG^r_N = (Reach(N), Tr_N|_{Reach_N \times Reach_N}, \{m_0\}, id_N|_{Reach_N})$
where $Reach(N)$ is the set of reachable $(N(A), m_0)$-markings

- $n^r = n|_{TG_N^r}$

The following lemma states the equivalence between the rewrite steps specified in Construction 2.34 and the firing of transitions.

Lemma 2.40 (Behaviour equivalence for sequential nets)
The firing step $m[t/a\rangle m'$ is possible in net $N = (\Sigma, P, T, Pre, Post : T \times P \to \mathcal{M}(T_\Sigma(X)), Cond : T \to Eqns_\Sigma(X))$ with $\Sigma = (S, OP, E)$ if and only if there is a rewrite step $r(t)/a' : A(m) \Rightarrow A(m')$ w.r.t. the algebra rewrite specification $R\Gamma_N$ corresponding to the net N.

Proof: Let $a : Vars(t) \to A_{sorts(Vars(t))}$ such that $m[t/a\rangle m'$ is a firing step in $N(A)$. Then the firing conditions of t are fulfilled at marking m with match a, and since $A \subseteq A(m)$ and $m_p^{A(m)} = m(p)$ the conditions E_{condt} are fulfilled for any match $a' : X_t \to A(m)_{sorts(X_t)}$ that includes a. With the match $a' = a \cup \{x_p \mapsto m_p\}$ the equations E_{lt} are also satisfied, so there is a rewrite step $r(t)/a' : A(m) \Rightarrow A'$ w.r.t. $R\Gamma_N$. We have according to definition 2.5 of rewriting

$$\begin{aligned} A'_S &= A_S = A(m')_S \\ A'_E &= (A_E - \{m_p = m(p) \mid p \in P\}) + \\ &\quad \{m_p = m(p) \ominus Pre(t,p) \oplus Post(t,p) \mid p \in P\} \end{aligned}$$

Thus $A' = PAlg(A'_S, A'_E) = A(m')$.
Suppose $A(m') \overset{r(t)/a'}{\Rightarrow} A'$. Then the equations E_{Condt} are satisfied by $A(m)$ with match a' and consequently the firing of transition t is possible in net N at marking m with match $a := a'|_{Vars(t)}$. This results from the equivalence of E_{Condt}/a' and the firing conditions of t at marking m with match a'. Let m' be the marking resulting from the firing step $m[a/t\rangle$. According to the first part of the proof we have $A' = A(m')$.

The following Lemma is needed in order to proove Theorem 2.42

Lemma 2.41 (Behaviour equivalence for parallel nets)
The firing step $m[t/a\rangle m'$ is possible in the net $N = (\Sigma, P, T, Pre, Post : T \times P \to \mathcal{M}(T_\Sigma(X)), Cond : T \to Eqns_\Sigma(X))$ with $\Sigma = (S, Op, E)$ where parallel firing of transitions is allowed if and only if there is a rewrite step $r(t)/a' : A(m) \Rightarrow A(m')$ in the algebra rewrite system $R\Gamma_{Np}$ corresponding to the net N.

Proof: The proof is analogous to the proof for the sequential case (of lemma 2.40).

The following theorem and its proof applies to both parallel and sequential nets.

Theorem 2.42 (Isomorphism of transformation systems)
Let N be an AHL-net and let $TS(R\Gamma_N)$ be the transformation system induced by its rewrite specification and $TS(N)$ its transformation system as defined in Definition 2.38. Then the two transformation systems are isomprphic in the category of transformation systems:

$$TS(N) \cong TS(R\Gamma_N)$$

Proof: Let $(TG_N, n : TG_N \to \mathcal{D}_{D\Gamma_N}) := TS(N)$ and $(TG_{R\Gamma_N}, \gamma : TG_{R\Gamma_N} \to D_{D\Gamma(R\Gamma_N)}) := TS(R\Gamma_N)$. Since $D\Gamma(R\Gamma_N) = D\Gamma_N$ it is sufficient to find a transition graph isomorphism $h : TG_{R\Gamma_N} \to TG_N$ that is compatible with n and γ. In order to show the isomorphism we will also define its inverse h^{-1}. According to lemma 2.40 for the sequential case and to lemma 2.41 for the parallel case and because $Tr_N(m, m') = Tr_{R\Gamma_N}(A(m), A(m'))$ for all $m, m' \in CS_N$ we have:

$$TS(N) = TS(R\Gamma_N)[TG_{R\Gamma_N}/(TG_{R\Gamma_N}[A(m)/m])]$$

Let $h(A(m)) := m$ and $h^{-1}(m) := A(m)$. Let the transition components of h and h^{-1} be identities on M_N.
Now we show that h and h^{-1} are $D\Gamma_N$ morphisms:

$$n \circ h(A(m)) = n(m) = A(m) = \gamma(A(m))$$
$$n \circ h(t) = n(t) = \{t\} = \gamma(t)$$

$$\gamma \circ h^{-1}(m) = \gamma(A(m)) = A(m) = n(m)$$
$$\gamma \circ h^{-1}(t) = \gamma(t) = \{t\} = n(t)$$

Since $h \circ h^{-1} = id_{TG_N}$ and $h^{-1} \circ h = id_{TG_{R\Gamma_N}}$ the transformation systems $TS(N)$ and $TS(R\Gamma_N)$ are equivalent.

For AHL-nets it was possible to derive equivalence of transformation systems for both the parallel and the sequential case because the adequate definition of n in Definition 2.38.

Chapter 3

Composition of Net Algebra Transformation Systems

In this chapter we analyse the composition of the transformation system semantics of nets as defined in the previous chapter, in the sense that we try to give a representation of the synchronized composition of transformation systems on the syntactical layer of place-transition nets. In the first section, we provide the necessary definitions and theorems. Secondly we make some considerations on the specific properties of the synchronized composition operation if it is applied to semantics of place-transition nets. Then we show that no syntactical representation of the synchronized composition exists for the general case. Afterwards we give some cases in which the composition of net semantics does correspond to an operation on nets and prove this correspondence. A section with informal observations on correspondencies between properties of the synchronized composition and possible syntactical representations on the layer of nets concludes the chapter.

3.1 Composition of Transformation Systems

The composition operations on algebra transformation systems are defined in [Gro99a].

Definition 3.1 (Synchronized Composition)
Let $\mathbf{A}_i = (TG_i, \mathbf{a}_i : TG_i \to \mathcal{D}_{D\Gamma_i})$ for $i \in \{0, 1, 2\}$ be three transformation systems and $(\sigma_1, h_1) : \mathbf{A}_1 \to \mathbf{A}_0$ and $(\sigma_2, h_2) : \mathbf{A}_2 \to \mathbf{A}_0$.

The *synchronized composition* of $\mathbf{A}_1$ and $\mathbf{A}_2$ w.r.t. $\mathbf{A}_0$ is the algebra transformation system $\mathbf{A}_3 = (TG_3, \mathbf{a}_3 : TG_3 \to \mathcal{D}_{D\Gamma_3})$ given by the following construction:

1. The data space specification
   Let $(D\Gamma_3, \overline{\sigma_1} : D\Gamma_1 \to D\Gamma_3, \overline{\sigma_2} : D\Gamma_2 \to D\Gamma_3)$ be the pushout of σ_1 and σ_2 in the category of data space specifications. The data space specification of $\mathbf{A}_3$ is $D\Gamma_3$, defined by

   $$D\Gamma_3 = D\Gamma_1 +_{(\sigma_1, \sigma_2)} D\Gamma_2$$

   The existence of this pushout is shown in [Gro99a].

2. The transition graph $TG_3 = (CS_3, T_3, I_3, id_3)$ is the synchronized composition (pullback) of $h_1 : TG_1 \to TG_0$ and $h_2 : TG_2 \to TG_0$ and is given by

   - $CS_3 = \{\langle c_1, c_2\rangle \mid c_1 \in CS_1, c_2 \in CS_2, h_{1S}(c_1) = h_{2S}(c_2)\}$,
   - $T_3 = \{\langle t_1, t_2\rangle \in T_1 \times T_2 \mid h_1(t_1) = h_2(t_2)\}$,
   - $I_3 = \{\langle i_1, i_2\rangle \mid i_1 \in I_1, i_2 \in I_2, h_{1S}(i_1) = h_{2S}(i_2)\}$,
   - $id(\langle c_1, c_2\rangle) = \langle id_{c_1}, id_{c_2}\rangle$
     the addressed transition exists in T_3 because of its construction and because h_1 and h_2 are transition graph morphisms.

3. The data space
   $\mathcal{D}_{D\Gamma_3}$ is uniquely determined by $D\Gamma_3$ as shown in definition 2.8.

4. The induced morphism $\mathbf{a}_3$:

   - For $h_1(c_1) = c_0 = h_2(c_2)$ (the only case in which $\langle c_1, c_2\rangle$ is a state in TG_3 let

     $$\mathbf{a}_3(\langle c_1, c_2\rangle) = \mathbf{a}_1(c_1) +_{\mathbf{a}_0(c_0)} \mathbf{a}_2(c_2).$$

     The amalgamation above is possible since:

     $$V_{\sigma_1}(\mathbf{a}_1(c_1)) = \mathbf{a}_0(h_1(c1)) = \mathbf{a}_0(h2(c_2)) = V_{\sigma_2}(\mathbf{a}_2(c2))$$

     analogous $V_{\sigma_2}(\mathbf{a}_2(c_2)) = \mathbf{a}_0(c_0) = V_{\sigma_1}(\mathbf{a}_1(c_1))$
   - For $\langle t_1, t_2\rangle \in T_3$ we define $\mathbf{a}_3(\langle t_1, t_2\rangle)$. The amalgamation construction is extended to method sets as follows:
     For $T_1' \subseteq Act_{D\Gamma_1}(A_1, B_1)$ and $T_2' \subseteq Act_{D\Gamma_2}(A_2, B_2)$ such that

$\sigma_1^{-1}(T_1') = \sigma_2^{-1}(T_2') =: T_0' \in Act_{D\Gamma_0}(A_0, B_0)$ the amalgamation is

$$\overline{\sigma_1}(T_1') \cup \overline{\sigma_2}(T_2') \in Act_{D\Gamma_1 +_{\langle \sigma_1, \sigma_2 \rangle} D\Gamma_2}(A_1 +_{A_0} A_2, B_1 +_{B_0} B_2).$$

This property is used in the following construction:

$$\mathbf{a}_3(\langle t_1, t_2 \rangle) = \overline{\sigma}_{1M}(\mathbf{a}_1(t_1)) \cup \overline{\sigma}_2(\mathbf{a}_2(t_2))$$

Fact 3.2 (The Synchronized Composition is a Pullback [Gro99a]) The synchronised composition of two transformation systems $\mathbf{A}_1$ and $\mathbf{A}_2$ over the cospan

$$\mathbf{A}_1 \xRightarrow{(\sigma_1, h_1)} \mathbf{A}_0 \xLeftarrow{(\sigma_2, h_2)} \mathbf{A}_2$$

yields a pullback in the category of transformation systems, i.e. $(\mathbf{A}_3, (\overline{\sigma_1}, \overline{h_1}), (\overline{\sigma_1}, \overline{h_1}))$ is a pullback of the morphisms (σ_1, h_1) and (σ_2, h_2) in **ATS**.

3.2 Composition of Place-Transition Net Semantics — General Considerations

In this section we try to give an overview on the specific properties of synchronized composition for the case that it is applied to place-transition net semantics. First we give some specific properties the synchronized composition has if it is applied to place-transition net transformation systems. Then we give two examples which show that synchronized composition does not always result in the transformation system of a place-transition net, and to familiarize the reader with specific reasons for such situations. We also give an example where the synchronized composition is equivalent to the transition system of a net in the parallel case and can be embedded in the transformation system of a place-transition net with equivalent behaviour in the sequential case.

The notations regarding the synchronized composition introduced in this section will be used in the whole chapter. Also, since we speak only about place-transition nets we shall often call them nets.

For $i \in \{1, 2\}$ let N_i be place-transition nets and $TS(N_i) = (TG_i, n_i : TG_i \to \mathcal{D}_{D\Gamma_i})$ be their transformation systems. As described in the previous section, the cospan

$$TS(N_1) \xRightarrow{(\sigma_1, h_1)} TS_0 \xLeftarrow{(\sigma_2, h_2)} TS(N_2)$$

induces a synchronized composition operation on the transformation systems. A more detailed figure showing the cospan is given below:

$$\begin{tikzcd} M(T_1) & M(T_0) \arrow[l, "\sigma_{1M}"'] \arrow[r, "\sigma_{2M}"'] & M(T_2) \\ \Sigma_{N_1} & \Sigma_{N_0} \arrow[l, "\sigma_{1\Sigma}"'] \arrow[r, "\sigma_{2\Sigma}"'] & \Sigma_{N_2} \end{tikzcd}$$

$$\begin{tikzcd} TG_1 \arrow[r, "h_1"'] & TG_0 & TG_2 \arrow[l, "h_2"'] \end{tikzcd}$$

Within the process of defining the cospan there are several degrees of freedom:

- The transformation system $TS_0 = (TG_0, u_0 : TG_0 \to \mathcal{D}_{D\Gamma_0})$,
- The data space specification morphisms with two components each:
  – the specification components $\sigma_{1\Gamma}$ and $\sigma_{2\Gamma}$
  – the method components σ_{1M} and σ_{2M}
- The transition graph morphisms h_1 and h_2 that have to satisfy:

  $$u_0 \circ h_i = V_{\sigma_i} \circ n_i$$

  Essentially, h_1 and h_2 map the marking graphs of the two nets to a common transition graph obeying this property. The figure below illustrates the fact that (σ_i, h_i) are transition system morphisms.

$$\begin{tikzcd} & TG_0 \arrow[dd, "u_0"] & \\ TG_1 \arrow[ur, "h_1"] \arrow[dd, "n_1"'] & & TG_2 \arrow[ul, "h_2"'] \arrow[dd, "n_2"] \\ & \mathcal{D}_{D\Gamma_0} & \\ \mathcal{D}_{D\Gamma_1} \arrow[ur, "V_{\sigma_1}"'] & & \mathcal{D}_{D\Gamma_2} \arrow[ul, "V_{\sigma_2}"] \end{tikzcd}$$

Let for $i \in \{1, 2\}$ $TG_i = (CS_i, Tr_i, CS_i, id_i)$. The cospan properties described by the following two facts result from $TS(N_1)$ and $TS(N_2)$ being transformation systems of nets.

Fact 3.3 (Induced cospan restriction)
The transition system TS_0 has the following property:

$M_{0w} = \emptyset$ if $w \neq \lambda$

This means that $M_0 = M_{0\lambda}$.

Proof: Suppose it were otherwise, we would have for $t \in M_{0w}$: $\sigma_{1M}(t) \in M_{1w}$: contradiction to $M_{1w} = \emptyset$.

Fact 3.4 (Result properties)
Let $TS_3 = (TG_3, u_3 : TG_3 \to \mathcal{D}_{D\Gamma_3})$ be the result of the synchronized composition of the cospan above. Let $(CS_3, Tr_3, I_3, id_3) := TG_3$. Let $D\Gamma_3 = (\Gamma_3, M_3)$. The fact that we are synchronising place-transition net semantics induces some result properties:

1. The resulting specification has exactly one sort if Γ_0 is not empty. This happens because every sort in Γ_0 has to be mapped by $\sigma_{i\Gamma}$ to the only sort in Γ_i ($i \in \{1, 2\}$), and thus the pushout of $\sigma_{1\Gamma}$ and $\sigma_{2\Gamma}$ contains only one sort. We shall call in the following this sort *nat*.
   (If Γ_0 is empty Γ_3 has two sorts, yielding the pushout of the empty morphisms.)

2. All operations in Γ_3 that do not belong to the *nat* specification are *nat*-constants. Their maximum number is $|P_1| + |P_2|$, the sum of the cardinalities of the sets of places of the nets N_1 and N_2. This results immediately from the properties of the pushout construction.

3. If we consider sequential nets, for $tr_3 \in Tr_3$ there are two possibilities w.r.t. the set $n_3(tr_3)$. Let $(tr_1, tr_2) = tr_3$:

   - $n_3(tr_3) = \{tr_1\}$, if $\exists t \in M_0$ such that $\sigma_1(t) = tr_1$ and $\sigma_2(t) = tr_2$
   - $n_3(tr_3) = \{tr_1, tr_2\}$ otherwise

   This follows from the fact that $n_1(tr_1) = \{tr_1\}$ and that $n_2(tr_2) = \{tr_2\}$ and from the construction of n_3 which follows from the synchronized composition operation (see definition 3.1).

Note that, as far as the data space specification is concerned, the result is suitable for the semantics of a place-transition net. It almost always has exactly one sort, *nat*, the operations not belonging to the *nat* specification are *nat*-constants, and all actions are included in $M_{3\lambda}$. If we had defined the data space specification of a net ommitting the operations from the *nat*-specification, and excluded the case $\Gamma_0 \neq \emptyset$, every resulting specification $D\Gamma_3$ would correspond to a net.

In the following we show that there is no general representation of the synchronized composition of net semantics in the formalism of place-transition nets. This means that, given two nets there is a synchronized composition operation of their semantics, that does not have as a result the semantics of a net.

Example 3.5 (Empty data space specification)
Without loss of generality we suppose for this example that the sets P_1 and P_2 and the sets T_1 and T_2 are disjoint. Let $D\Gamma_0 = (\emptyset, \emptyset)$ and $TG_0 = (\{*\}, \{id_*\}, \{*\}, id(*) = id_*)$. Let σ_1 and σ_2 be the empty data space specification morphisms. Let $h_i(m) = *$ for all $m \in CS_i$ and let $h_i(tr) = id_*$ for $tr \in Tr_i$.

The resulting specification has two sorts which would not allow the construction of a specification isomorphism between a data space specification of a net and $D\Gamma_3$. This implies that no transition system isomorphism exists between TS_3 and the transition system of a place-transition net.

However, in this case the resulting transition graph TG_3 is isomorphic to the transition graph of a net that can be obtained syntactically from N_1 and N_2, both in the parallel and in the sequential case. This we prove in the following.

Proof: Transition graph equivalence
In the following let $TG_N = (CS_N, Tr_N, CS_N, id_N)$ be the transition graph of the net N, which we are defining gradually. Also gradually we define a bijective mapping $h : TG_N \to TG_3$ and it inverse $h^{-1} : TG_3 \to TG_N$.

The transition graph morphisms map all states to $*$. This results in the fact that all combinations of states are encountered in CS_3, i.e. $CS_3 = CS_1 \times CS_2$. This transition graph state space corresponds to a net N with the set of places

$$P_N = P_1 \cup P_2$$

(recall the disjointness of P_1 and P_2) and with the capacity function given by

$$K_N = K_1 \cup K_2.$$

This is justified by the existence of a bijective mapping between CS_N and CS_3 given by $m_N \mapsto (m_N|_{P_1}, m_N|_{P_2})$ for $m_N \in CS_N$ and $(m_1, m_2) \mapsto m_1 \cup m_2$ for $(m_1, m_2) \in CS_3$.

Also, all transitions to are mapped to id_*. Thus all combinations of transitions $tr_1 \in Tr_1(m_1, m_1')$ and $tr_2 \in Tr_2(m_2, m_2')$ have the property $(t_1, t_2) \in Tr_3((m_1, m_2), (m_1', m_2'))$.

For the parallel case, this yields the transition graph of a net with

$$
\begin{aligned}
&T_N = T_1 \cup T_2 \\
&Pre_N(t,p) = Pre_i(t,p), \text{ for } t \in T_i, p \in P_i \\
&Post_N(t,p) = Post_i(t,p), \text{ for } t \in T_i, p \in P_i
\end{aligned}
$$

since there is a bijective mapping between $Tr_N(m_N, m_N')$ and $Tr_3((m_1, m_2), (m_1', m_2'))$ where $(m_1, m_2) = (m_N|_{P_1}, m_N|_{P_2})$, and $(m_1', m_2') = (m_N'|_{P_1}, m_N'|_{P_2})$ This mapping is represented in the following table by $\longleftrightarrow$:

$$
\begin{array}{cccl}
Tr_N((m_1, m_1'), (m_2, m_2')) & \underset{h^{-1}}{\overset{h}{\rightleftarrows}} & Tr_3((m_1, m_2), (m_1', m_2')) & \\
id_{m_1 \cup m_2} & \longleftrightarrow & (id_{m_1}, id_{m_2}) & \\
tr_1 & \longleftrightarrow & (tr_1, id_{m_2}) & \text{if } tr_1 \in M_{1\lambda} \\
tr_2 & \longleftrightarrow & (id_{m_1}, tr_2) & \text{if } tr_2 \in M_{2\lambda} \\
tr_1 \oplus tr_2 & \longleftrightarrow & (tr_1, tr_2) & \text{if } tr_1 \in M_{1\lambda} \\
 & & & \text{and } tr_2 \in M_{2\lambda}
\end{array}
$$

In the following let $tr_1 = tr_N|_{T_1}$ and $tr_2 = tr_N|_{T_2}$ for $tr_N \in Tr_N$.

According to the definitions of Pre_N, $Post_N$, and K_N, the existence of the firing step $m_N[tr_N\rangle m_N'$ in N implies the existence of firing steps $m_i[tr_i\rangle m_i'$ in N_i if $tr_i \neq \lambda$ for $i \in \{1,2\}$. Thus, according to the definitions of h_1 and h_2 this results in the existence of $h(tr_N) \in Tr_3((m_1, m_2), (m_1', m_2'))$.
For $(tr_1, tr_2) \in Tr_3((m_1, m_2), (m_1', m_2'))$ the existence of the firing steps $m_1[tr_1\rangle m_1'$ and $m_2[tr_2\rangle m_2'$ implies according to the definitions of Pre_N, $Post_N$, and K_N, the existence of the firing step $m_1 \cup m_2[tr_1 \oplus tr_2\rangle m_1' \cup m_2'$ in N. If both tr_1 and tr_2 are identities, there is nothing to be shown, since the transition $id_{m_1 \cup m_2}$ exists anyway in $Tr_3((m_1, m_2), (m_1', m_2'))$ if $m_1 = m_1'$ and $m_2 = m_2'$.

For the sequential case, the definition of N can be completed as follows:

$$
\begin{aligned}
&T_N = (T_1 \times T_2) \cup T_1 \cup T_2 \\
&Pre_N(t_i, p_i) = Pre_i(t_i, p_i)
\end{aligned}
$$

$Post_N(t_i, p_i) = Post_i(t_i, p_i)$
$Pre_N(t_i, p_j) = 0 \quad (i \neq j)$
$Post_N(t_i, p_j) = 0 \quad (i \neq j)$
$Pre_N((t_1, t_2), p) = Pre_i(t_i, p)$ for $p \in P_i$
$Post_N((t_1, t_2), p) = Post_i(t_i, p)$ for $p \in P_i$

The mapping is:

$$Tr_N(m_1 \cup m_2, m'_1 \cup m'_2) \quad \underset{h^{-1}}{\overset{h}{\rightleftarrows}} \quad Tr_3((m_1, m_2), (m'_1, m'_2))$$

$$\begin{array}{cccl}
id_{m_1 \cup m_2} & \longleftrightarrow & (id_{m_1}, id_{m_2}) & \\
t_1 & \longleftrightarrow & (t_1, id_{m_2}) & \text{if } t_1 \in T_1 \\
t_2 & \longleftrightarrow & (id_{m_1}, t_2) & \text{if } t_2 \in T_2 \\
(t_1, t_2) & \longleftrightarrow & (t_1, t_2) & \text{if } t_1 \in T_1, t_2 \in T_2
\end{array}$$

Let $m_N = m_1 \cup m_2$, $m'_N = m'_1 \cup m'_2$ and let $m_3 = (m_1, m_2)$, $m'_3 = (m'_1, m'_2)$. The fact that $id_{m_1 \cup m_2} \in Tr_N(m_1 \cup m_2, m'_1 \cup m'_2) \Leftrightarrow (id_{m_1}, id_{m_2}) \in Tr_3((m_1, m_2), (m'_1, m'_2))$ is obvious. Also, according to the definitions of $Pre_N, Post_N, K_N$ and that of h_1 and h_2 we also have the following equivalences:

$$t_1 \in Tr_N(m_N, m'_N) \text{ and } t_1 \in T_1 \quad \Leftrightarrow \quad (t_1, id_{m_2}) \in Tr_3(m_3, m'_3)$$

since $Pre_N(T_1, p)) = Post_N(t_1, p) = 0$ for each $p \in P_2$.

$$\begin{array}{lcl}
t_2 \in Tr_N(m_N, m'_N) \text{ and } t_2 \in T_2 & \Leftrightarrow & (id_{m_1}, t_2) \in Tr_3(m_3, m'_3) \\
(t_1, t_2) \in Tr_N(m_N, m'_N) & \Leftrightarrow & (t_1, t_2) \in Tr_3(m_3, m'_3)
\end{array}$$

Thus isomorphism of the transition graphs TG_3 and TG_N holds both for nets with parallel firing and for nets with sequential firing.

Fact 3.6 (Minimal Net Yielding Synchronization)
Let the cospan $((\sigma_1, h_1), TS_0, (\sigma_2, h_2))$ be given by

- $TS_0 = (TG_0, u_0 : TG_0 \to \mathcal{D}_{D\Gamma_0})$ with the data space signature $D\Gamma_0 = (nat, \emptyset)$, the transition graph TG_0 from example 3.5, and with $u_0(*) = Nat$ and $u_0(id_*) = \emptyset \in Tf_{D\Gamma_0}(Nat, Nat)$,
- $\sigma_{1\Gamma}$ and $\sigma_{2\Gamma}$ are inclusions of the nat specification in $D\Gamma_1$ and $D\Gamma_2$, σ_{1M} and σ_{2M} are the empty functions,
- h_1 and h_2 are given as in example 3.5.

The result of the synchronized composition of the above cospan is equivalent in the parallel case to the transformation system of the net N, as it is given in example 3.5.

In the sequential case, there is a transformation system morphism $(\sigma', h') : TS_3 \to TS(N)$, that embeddes the transformation system resulting from the mentioned synchronization into the transformation system of net N from the sequential case of example 3.5.

Proof: We use the notations from example 3.5, and this proof relies on the transition graph isomorphisms constructed there. Let the resulting transformation system be $TS_3 = (TG_3, u_3 : TG_3 \to \mathcal{D}_{D\Gamma_3})$. Let $h : TG_N \to TG_3$ and $h^{-1} : TG_3 \to TG_N$ denote mappings we defined there.

The parallel case
Since $D\Gamma_3 = D\Gamma_N$ we only have to show that h and h^{-1} are $D\Gamma$-morphisms. First, we shall introduce a notation that suits this purpose:
Notation
For the multiset $tr_N \in Tr_N(m_N, m'_N)$, $tr_N = \bigoplus\limits_{t \in T_N} k_t t$ we address the restriction of tr_N to T_1 or T_2 in the following way:

$$tr_N|_{T_i} = \begin{cases} \bigoplus\limits_{t \in T_i} k_t t & \text{if } \bigoplus\limits_{t \in T_i} k_t t \neq \lambda \\ id_{m_N|_{P_i}} & \text{if } \forall t \in T_i : \; k_t = 0 \end{cases}$$

Furthermore, let id_m represent the empty multiset if it occurs in expressions like $id_m \oplus tr$, i.e. let $id_m \oplus tr = tr = tr \oplus id_m$.

The picture below illustrates the situation:

$$\begin{tikzcd} TG_3 \arrow[rr, shift right, "h^{-1}"'] \arrow[dr, "u_3"'] & \arrow[d, phantom, "="] & TG_N \arrow[ll, shift right, "h"'] \arrow[dl, "n"] \\ & TG_{D\Gamma_N} & \end{tikzcd}$$

We have:

$$\begin{aligned} & n \circ h^{-1}((m_1, m_2)) = n(m_1 \cup m_2) = A(m_1 \cup m_2) = \\ & = u_1(m_1) +_{\mathbb{N}} u_2(m_2) = u_3((m_1, m_2)) = \end{aligned}$$

The last equality holds by construction of u_3, from Definition 3.1.

$$u_3 \circ h(m_N) = A((m_N|_{P_1}, m_N|_{P_2})) = n(m_N)$$

$$n \circ h^{-1}((tr_1, tr_2)) = n(tr_1 \oplus tr_2) \stackrel{(*)}{=} n(tr_1) \cup n(tr_2) = \\ = n_1(tr_1) \cup n_2(tr_2) = u_3((tr_1, tr_2))$$

Here, $(*)$ holds since tr_1 and tr_2 are disjoint.

$$u_3 \circ h(tr_N) = u_3((tr_N|_{T_1}, tr_N|_{T_2})) = n_1(tr_N|_{T_1}) \cup n_2(tr_N|_{T_2}) = \\ = n(tr_N)$$

The sequential case

We define a data space specification morphism $\sigma = (\sigma_\Gamma, \sigma_M) : D\Gamma_N \to D\Gamma_3$ such that (σ, h) is a transformation system morphism. Let σ_Γ be the identity on $\Gamma_N = nat + (m_p)_{p \in P_1 \cup P_2}$. Let σ_M be given by $\sigma_M(t) := t$. We have

$u_3 \circ h(m_N) = u_3(m_N|_{P_1}, m_N|_{P_2}) = A(m_N|_{P_1}) +_{Nat} A(m_N|_{P_2}) = A(m_N) = V_\sigma(A(m_N))$

$u_3 \circ h(tr_N) = u_3((tr_N|_{T_1}, tr_N|_{T_2})) = n_1(tr_N|_{T_1}) \cup n_2(tr_N|_{T_2}) = n(tr_N) = V_\sigma(n(tr_N))$

Remark 3.7 (The sequential case)

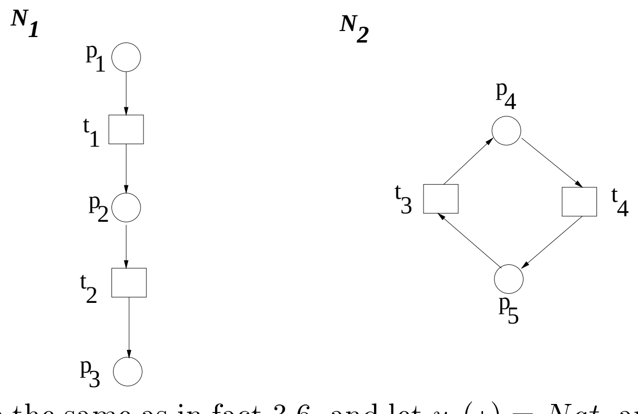
We could not define a transformation system morphism from $(\sigma', h^{-1}) : TS_3 \to TS(N)$ because M_N contains more transitions than M_3 (like for example (t_1, t_2)) and these can't be mapped properly by a data space specification morphism σ' to the set $T_1 \cup T_2$. This means that no mapping can be found such that $n \circ h^{-1}((t_1, t_2)) = {\sigma'_M}^{-1} \circ u_3((t_1, t_2))$, which is required if (σ', h^{-1}) is to be a transition system isomorphism, since the transition component of $V_{\sigma'}$ is ${\sigma'_M}^{-1}$. The following figure illustrates this situation.

$$
\begin{tikzcd}
\underrightarrow{(t_1,t_2)} \arrow[d, "n"'] & & \underrightarrow{(t_1,t_2)} \arrow[ll, "h^{-1}"'] \arrow[d, "u_3"] \\
\underrightarrow{\{(t_1,t_2)\}} & & \underrightarrow{\{t_1,t_2\}} \arrow[dll, "{\sigma'_M}^{-1}"] \\
\underrightarrow{{\sigma'_M}^{-1}(t_1) \cup {\sigma'_M}^{-1}(t_2)} & &
\end{tikzcd}
$$

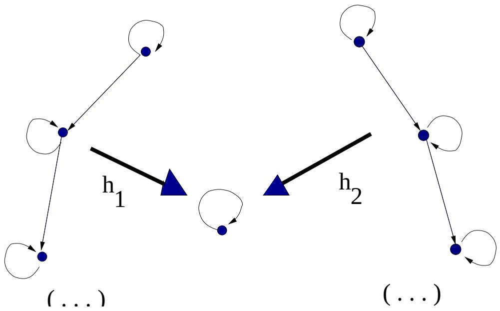
Note that the set ${\sigma'_M}^{-1}(t_1) \cup {\sigma'_M}^{-1}(t_2)$ has at least two elements and that the set $\{(t_1, t_2)\}$ has only one

In order to illumine one other specific problem that might appear even if the gluing specification Γ_0 is not empty we shall give another example.

Example 3.8 (Too much Gluing of Transitions)
Let the net N_1 and N_2 in the figure below, from a cospan, using the notations above:



Let TG_0 be the same as in fact 3.6, and let $u_0(*) = Nat$, and $u_0(id_*) = \emptyset$. Let h_1 and h_2 map all states of TG_1 and TG_2 to $*$, and all transitions to id_*, as suggested in the following figure:



Let $\Gamma_0 = nat$ in $D\Gamma_0 = (\Gamma_0, M_0)$ and $M_0 = \{a, b, c, d\}$ for the sequential case and $M_0 = \mathcal{M}(\{a, b, c, d\}) \setminus \{\lambda\}$ for the parallel case. Let the morphisms $\sigma_{1\Gamma}$ and $\sigma_{2\Gamma}$ be inclusions, and let **for the sequential case:**

$$\sigma_{1M} = \{a \mapsto t_1, b \mapsto t_2, c \mapsto t_2\}$$
$$\sigma_{2M} = \{a \mapsto t_3, b \mapsto t_3, c \mapsto t_4\}$$

This yields according to the standard pushout construction as a resulting set of actions $M_3 = \{[t_1]\}$, where $[t_1]$ is the equivalence class of t_1 of $T_1 \cup T_2$ w.r.t. the equivalence relation $\{(f_1(x), f_2(x)) \mid x \in M_0\}$. Thus M_3 contains only one element, and transitions in the data space can thus be labeled either with the empty set or with the set containing this element.
For the parallel case let σ_{1M} and σ_{2M} be the extensions of the sequential case morphisms to monoid morphisms. In this case $M_3 = \mathcal{M}(\{[t_1]\})$.

Proposition 3.9 (Too much Transition Gluing)
No net transformation system is equivalent to TS_3, the transformation system resulting from the synchronized composition presented in example 3.8.

Proof: Suppose there is a net $N = (P, T, Pre, Post, K)$ such that an isomorphism $(\sigma, h) : TS(N) \to TS_3$ and its inverse $(\sigma^{-1}, h^{-1}) : TS_3 \to TS(N)$ exist. This morphism includes a specification isomorphism component, thus the number of nat-constants of Γ_N must equal the number of nat-constants of Γ_3. Thus $|P| = |P_1 \cup P_2| = 5$ Without loss of generality we can call the places of N $p'_1, p'_2, p'_3, p'_4, p'_5$ and suppose that $\sigma(m_{p_i}) = m_{p'_i}$ for $i \in \{1, 2, 3, 4, 5\}$.

The transformation system TS_3 contains the following transition steps, which are mapped to the data space transitions drawn below them.

$$(p_1, p_4) \xrightarrow{(t_1, t_4)} (p_2, p_5) \xrightarrow{(t_2, t_3)} (p_3, p_4)$$

$$A(p_1) \cup A(p_4) \xrightarrow{\{[t_1]\}} A(p_2) \cup A(p_5) \xrightarrow{\{[t_1]\}} A(p_3) \cup A(p_4)$$

Note that $[t_1] = [t_2] = [t_3] = [t_4] = [t_5] = \{t_1, t_2, t_3, t_4, t_5\}$ which implies $\overline{\sigma}_{jM}(t_i) = [t_1]$ for $j \in \{1, 2\}$ and $i \in \{1, 2, 3, 4, 5\}$. Let

$$t := \sigma([t_1])$$

We have:

$$\begin{array}{l}
n \circ h^{-1}((t_1, t_4)) = V_{\sigma^{-1}} \circ u_3((t_1, t_4)) = \{t\} = n(t) \\
n \circ h^{-1}((t_2, t_3)) = V_{\sigma^{-1}} \circ u_3((t_2, t_3)) = \{t\} = n(t) \\
n \circ h^{-1}((p_1, p_4)) = V_{\sigma^{-1}} \circ u_3((p_1, p_4)) = A(p'_1 \oplus p'_4) = n(p'_1 \oplus p'_4) \\
n \circ h^{-1}((p_2, p_5)) = V_{\sigma^{-1}} \circ u_3((p_2, p_5)) = A(p'_2 \oplus p'_5) = n(p'_2 \oplus p'_5) \\
n \circ h^{-1}((p_3, p_4)) = V_{\sigma^{-1}} \circ u_3((p_3, p_4)) = A(p'_3 \oplus p'_4) = n(p'_3 \oplus p'_4)
\end{array}$$

Both in the parallel and in the sequential case the preimage with regard to n of a method set with one element $\{t\} \in Tf_{D\Gamma_N}(A(m_1), A(m_2))$ can be only $t \in Tr_N(m_1, m_2)$. Thus the transition graph TG_N includes the following graph:

$$(p'_1 \oplus p'_4) \xrightarrow{\quad t \quad} (p'_2 \oplus p'_5) \xrightarrow{\quad t \quad} (p'_3 \oplus p'_4)$$

In the sequential case, where $t \in T$, firing rules imply:

$$\begin{array}{l}
1 - Pre(t, p_1) + Post(t, p_1) = 0 \text{ and} \\
0 - Pre(t, p_1) + Post(t, p_1) = 0
\end{array}$$

which is a contradiction, since it implies $1 = 0$.
In the parallel case, since $t \in M_N$ and $n(t) = \{t\}$ there must be a number $k \in \mathbb{N}$ and a transition $t' \in T$ such that $t = kt'$. This implies according to firing rules:

$$\begin{array}{l}
1 - kPre(t', p_1) + kPost(t', p_1) = 0 \text{ and} \\
0 - kPre(t', p_1) + kPost(t', p_1) = 0
\end{array}$$

which is a contradiction, too.

Discussion 3.10 (Method Morphism Components)
The example 3.8 and the proof of proposition 3.9 indicate that it is only possible to glue transitions of the same net by defining non-injective morphism components σ_{1M} and/or σ_{2M} if the firing effects of these transitions are exactly the same.

3.3 Gluing of Places and Transitions

The syntactical synchronization of two parallel nets done by gluing of places and transitions reflects on the layer of place-transition nets the synchronized composition of the semantics of these nets by a cospan that has certain properties. In the sequential case we are able to define a sequential net, the semantics of which is behaviour equivalent to the synchronization result, and allows an embedding of the synchronized composition result in it.

First, we give some restrictions for the cospan that yields the described effect. Then we define the net N that corresponds to the result TS_3 of synchronized composition in the parallel case, and prove the equivalence of TS_3 and $TS(N)$. Afterwards we discuss the sequential case.

Definition 3.11 (Gluing cospan)
Let $N_i = (P_i, T_i, Pre_i, Post_i, K_i)$ for $i \in \{1, 2\}$ be two nets and let $TS(N_i) = (TG_i, n_i : TG_i \to \mathcal{D}_{D\Gamma_i})$ be their transformation systems.

$$TS(N_1) \xRightarrow{(\sigma_1,h_1)} TS_0 \xLeftarrow{(\sigma_2,h_2)} TS(N_2)$$

is called a cospan for the gluing synchronized composition if the following holds:

1. $TS_0 = (TG_0, u_0 : TG_0 \to \mathcal{D}_{D\Gamma_0})$ is the transformation system $TS(N_0) = (TG_{N_0}, n_0 : TG_{N_0} \to \mathcal{D}_{D\Gamma_{N_0}})$ of some net $N_0 = (P_0, T_0, Pre_0, Post_0, K_0)$. This means that $D\Gamma_0 = D\Gamma_{N_0}$, $TG_0 = TG_{N_0}$ and $u_0 = n_0$.

2. The specification morphisms σ_1 and σ_2 are injective.

3. For $i \in \{1, 2\}$, the specification morphisms σ_i correspond to net morphisms as follows. Let $f_i : P_0 \to P_i$ be given by $f_i = \{p \mapsto p' \mid m_{p'} = \sigma_{i\Gamma}(m_p)\}$. The following properties must hold:

$$Pre_i \circ (\sigma_{iM}|_{T_i}, f_i) = Pre_0 \qquad (a)$$
$$Post_i \circ (\sigma_{iM}|_{T_i}, f_i) = Post_0 \qquad (b)$$
$$K_i \circ f_i = K_0 \qquad (c)$$

   and

$$\sigma_{iM}(k_t t) = k_t \sigma_{iM}(t) \qquad (d)$$

The following picture refers to property (a) for σ_i.

$$\begin{tikzcd}
T_i \times P_i \arrow[dd, "(\sigma_{iM}|_{T_i}, f_i)"'] \arrow[dr, "Pre_i"] & \\
& \mathbb{N} \\
T_0 \times P_0 \arrow[ur, "Pre_0"'] &
\end{tikzcd}$$

4. Only transitions to be synchronized can put token on or take token from places to be synchronized. For $i \in \{1, 2\}$, for all $p \in f_{iP}(p_0)$, and for all $t \in T_i \backslash \sigma_{iM}(T_0)$, we must have

   $pre(t, p) = 0$ and $post(t, p) = 0$.

   This condition ensures that every state change in the subnets to be glued corresponds to a state change in the net N_0.

5. For $tr \in Tr_i(m, m')$ with $tr = \bigoplus\limits_{t \in T_i} k_t t$ the following correspondence between h_i and σ_i holds:

   $$h_i(tr) = \bigoplus_{t \in \sigma_{iM}(T_0)} k_t \sigma_{iM}^{-1}(t) \in Tr_0(h_i(m), h_i(m'))$$

Notation 3.12
We define for a net $N = (P, T, Pre, Post, K)$ extensions of $Pre, Post : T \times P \to \mathbb{N}$ to firing steps. For $tr \in \mathcal{M}(T)$ whith $tr = \bigoplus\limits_{t \in T} k_t t$ let:

$$Pre^*(tr) = \sum_{t \in T} k_t Pre(t) \text{ and } Post^*(tr) = \sum_{t \in T} k_t Post(t).$$

Notation 3.13
Let T_1', T_2', P_1' and P_2' be given as follows: $T_1' := T_1 \setminus \sigma_{1M}(T_0)$, $P_1' := P_1 \setminus f_1(P_0)$, $T_2' := T_2 \setminus \sigma_{2M}(T_0)$ and $P_2' := P_2 \setminus f_2(P_0)$. Also, let $T_2^0 := \sigma_{2M}(T_0)$, $T_1^0 := \sigma_{1M}(T_0)$ and $P_1^0 := f_1(P_0)$ and $P_2^0 = f_2(P_0)$

Let $TS_3 = (TG_3, n_3 : TG_3 \to TG_{D\Gamma_3})$ be the transformation system resulting from the pullback operation on the above defined cospan. Let $D\Gamma_3 = (\Gamma_3, M_3)$ and let $TG_3 = (CS_3, Tr_3, CS_3, id_3)$. The set of initial states equals the whole set of states CS_3 because this is also the case in TG_0, TG_1 and TG_2.

For the following proof we suppose without loss of generality that there are no name clashes, i.e. $T_1 \cap T_2' = \emptyset$ and $P_1 \cap P_2' = \emptyset$. Also without loss of generality we suppose that $T_0 \subseteq T_1$ such that $f_1(t) = t$, which can be achieved by adequate renaming of the transitions in T_0. (Then $T_1 \cup T_2' = T_1 \cup T_2$, thus $\mathcal{M}(T_1 \cup T_2') = \mathcal{M}(T_1 \cup T_2)$.)

According to synchronized composition building mechanisms we have:

$$\begin{array}{ll} \Gamma_3 = & nat+ \\ & m_p :\to nat \quad (p \in P_1 \cup P_2') \end{array}$$

and $M_3 = \{k_t t \mid t \in T_1 \cup T_2'\}$.

The following fact states some properties of the cospan defined above, which are useful for proving theorem 3.15, which states the equivalence between the semantics of a net N we shall define and TS_3. These properties are proved directly after being stated.

Lemma 3.14 (Cospan properties)
The cospan defined above has the following properties:

1. We have $h_1(m_1) = m_1 \circ f_1$ and $h_2(m_2) = m_2 \circ f_2$.
   for $m_1 \in CS_1$ and $m_2 \in CS_2$.
2. If $(m_1, m_2) \in CS_3$ then $m_1 \circ f_1 = m_2 \circ f_2$

Proof: For the first claim, for a function $g : A \to B$ the notation $[g]$ occuring after an entity means that all elements e of the preimage of g are replaced in the entity with $g(e)$.
Let $m_0 = h_1(m_1)$. Then $V_{\sigma_1}(A(m_1)) = n_0(m_0) = A(m_0)$, which is equivalent to $m_{0p}^{A(m_0)} = V_{\sigma_1}(m_{\sigma_1(p)}^{n_1(m_1)}) = m_1(f_1(p))$ Analogous $h_2(m_2) = m_2 \circ f_2$

The second point follows immediately from the first, as $(m_1, m_2) \in CS_3$ implies $h_1(m_1) = h_2(m_2)$.

Theorem 3.15 (Net corresponding to TS_3)
The transformation system TS_3 is equivalent to the transformation system of the net $N = (P_N, T_N, Pre_N, Post_N, K_N)$ given by:

$$\begin{array}{ll} P_N & = P_1 \cup P_2', \\ T_N & = T_1 \cup T_2', \\ Pre_N & = Pre_1 \cup Pre_2|_{T_2' \times P_2'} \cup \{(t,p) \mapsto 0 \mid (t,p) \in T_1 \times P_2' \cup T_2' \times P_1\}, \\ Post_N & = Post_1 \cup Post_2|_{T_2' \times P_2'} \cup \{(t,p) \mapsto 0 \mid (t,p) \in T_1 \times P_2' \cup T_2' \times P_1\}, \\ K_N & = K_1 \cup K_2|_{P_2'} \end{array}$$

in the sense that TS_3 is isomorphic to $TS(N)$ in the category of transformation systems.

Proof:

Let $TS(N) = (TG_N, n : TG_N \to TG_{D\Gamma_N})$. We have $D\Gamma_N = D\Gamma_3$, and thus it is enough to find a transition graph isomorphism $h : TG_N \to TG_3$ which is also a $D\Gamma_N$-transformation system isomorphism in order to prove equivalence of the transformation systems TS_3 and $TS(N)$.

In order to do this, we first define a transition graph isomorphism $h : TG_N \to TG_3$ and its inverse $h^{-1} : TG_3 \to TG_N$, and afterwards we show that h and h^{-1} are $D\Gamma_N$-morphisms.

Transition graph (iso)morphism

We define a transition graph morphism $h : TG_N \to TG_3$ such that (σ, h) becomes a transformation system isomorphism.

1. **State component of *h***
   For a state $m_N \in TG_N$ let
   $$h(m_N) := (m_N|_{P_1}, m_N|_{P_2'} \cup \{f_2(p_0) \mapsto m_N(f_1(p_0)) \mid p_0 \in P_0\}).$$
   We show the existence of $(m_1, m_2) := h(m_N) \in CS_3$. Since for all $p \in P_1$ we have $m_1(p) = m_N(p) \leq K_N(p) = K_1(p)$, $m_1 \in CS_1$ is true. For $p \in P_2'$ we have: $m_2(p) = M_N(p) \leq K_N(p) = K_2(p)$. For $p \in f_2(P_0)$ let $p_0 = f_2^{-1}(p)$. then $m_2(f_2(p_0)) = m_N(f_1(p_0)) \leq K_1(f_1(p_0)) = K_2(f_2(p_0))$, and thus $m_2 \in CS_2$. Furthermore, according to Lemma 3.14 and the definitions of m_1 and m_2 we have:
   $h_1(m_1) = \{p \mapsto m_1(f_1(p)) = m_N(f_1(p)) = m_2(f_2(p)) \mid p \in P_0\} = h_2(m_2)$

2. **State component of *h* *-1***
   For $(m_2, m_2) \in CS_3$ we define
   $$h^{-1}((m_1, m_2)) := m_1 \cup m_2|_{P_2'}.$$
   The existence of the state $m_1 \cup m_2|_{P_2'}$ in TG_N results from the fact that $m_1 \in CS_1$ and $m_2 \in CS_2$:
   $$m_1 \leq K_1,\ m_2 \leq K_2$$
   $$\Rightarrow m_1 \cup m_2|_{P_2'} \leq K_1 \cup K_2|_{P_2'}$$

3. **Transition component of h^{-1}**
   For a transition $tr_3 = (tr_1, tr_2) \in Tr_3(m_3, m_3')$ we define $h^{-1}(tr_3)$

   $$h^{-1}(tr_3) = tr_1 \oplus tr_2|_{T_2'} =: tr_N$$

   Recall that we are using the denotation $\bigoplus\limits_{t\in T} k_t t|_{T_0\subseteq T} := \bigoplus\limits_{t\in T_0} k_t t$.
   We show that $tr_N \in Tr_N(h(m_3), h(m_3'))$. Let $(m_1, m_2) := m_3$ and $(m_1', m_2') := m_3'$. We then have: $h^{-1}(m_3) = m_1 \cup m_2|_{P_2'}$ and $h^{-1}(m_3') = m_1' \cup m_2'|_{P_2'}$
   According to the definition of Pre_N and to the cospan property 3.11.4 we have:

   $$Pre_N^*(tr_N, p) = \begin{cases} Pre_1(tr_N|_{T_1}, p) = Pre_1(tr_1, p) & \\ \quad \leq m_1(p) = m_N(p) & p \in P_1' \\ Pre_1(tr_N|_{\sigma_1(T_0)}, p) = Pre_1(tr_1, p) & \\ \quad \leq m_1(p) = m_N(p) & p \in f_1(P_0) \\ Pre_2(tr_N|_{T_2'\cup f_1(T_1)}[\sigma_{1M}^{-1}][\sigma_{2M}], p) = & \\ \quad = Pre_2(tr_2, p) \leq m_2(p) = m_N(p) & p \in P_2', t \in f_1(T_0) \end{cases}$$

   Thus the first pre condition for the firing of the transition vector tr_N in N is satisfied. Analogously

   $$Post_N(tr_N, p) = \begin{cases} Post_1(tr_1, p) & p \in P_1 \\ Post_2(tr_2, p) & p \in P_2' \end{cases}$$

   Let $m_N' : P_N \to \mathbb{N}$, $m_N' = m_N - Pre_N(tr_N) + Post_N(tr_N)$. Thus

   $$m_N'(p) = \begin{cases} (m_1 - Pre_1(tr_1) + Post_1(tr_1))(p) \leq K_1(p) & p \in P_1 \\ (m_2 - Pre_2(tr_2) + Post_2(tr_2))(p) \leq K_2(p) & p \in P_2' \end{cases}.$$

   The inequalities result from the fact that tr_3 is a firing step in TS_3. Consequently $m_N' \leq K_N$ (we had $K_N = K_1 \cup K_2|_{P_2'}$) and there is a firing step $m_N[tr_N\rangle m_N'$ in N and a transition $tr_N \in Tr_N(m_N, m_N')$. Furthermore, since $m_N' = h^{-1}(m_3')$,

   $$tr_N \in Tr_N(h^{-1}(m_3), h^{-1}(m_3')).$$

4. **Transition component of h**
   For a transition $tr_N \in Tr_N(m_N, m_N')$ we define $h(tr_N)$ by

   $$h(tr_N) := (tr_N|_{T_1}, tr_N|_{\sigma_1(T_0)}[\sigma_{1M}^{-1}][\sigma_{2M}] \oplus tr_N|_{T_2'}) =: (tr_1, tr_2).$$

Let $h(m_N) =: (m_1, m_2)$ and $h(m'_N) =: (m'_1, m'_2)$. Thus $m'_1 \leq K_1$ and $m'_2 \leq K_2$. We show that there are transitions $tr_1 \in Tr_1(m_1, m'_1)$ and $tr_2 \in Tr_2(m_2, m'_2)$ by showing that corresponding firing steps exist for N_1 and respectively N_2.

According to the definition of Pre_N and to cospan property 3.11.4 we have for all $p \in P_1$:
$m_1(p) = m_N(p) \geq Pre^*_N(tr_N, p) = Pre^*_1(tr_1, p)$
$m_1(p) - Pre^*_1(tr_1, p) + Post^*_1(tr_1, p) =$
$\quad = m_N(p) - Pre^*_N(tr_N, p) + Post^*_N(tr_N, p) =$
$\quad = m'_N(p) = m'_1(p) \leq K_1(p)$
Thus $tr_1 \in Tr_1(m_1, m'_1)$

For $p \in f_1(P_0)$ we show that $Pre^*_2(tr_2, p) = Pre^*_N(tr_N, f_1 \circ f_2^{-1}(p))$. According to the cospan properties 3.11.4 and 3.11.3.a, and according to the definitions of Pre_N, tr_1 and tr_2, we have:

$$\begin{aligned} Pre^*_2(tr_2, p) &= Pre^*_2(\sigma_{2M} \circ \sigma_{1M}^{-1}(tr_N|_{T_1^0}), p) \\ &= Pre^*_1(tr_1, f_1 \circ f_2^{-1}(p)) \\ &= Pre^*_N(tr_N, f_1 \circ f_2^{-1}(p) \end{aligned}$$

Analogously, we have:

$$Post^*_2(tr_2, p) = Pre^*_N(tr_N, f_1 \circ f_2^{-1}(p)).$$

The pre condition of the firing step $m_2[tr_2\rangle m'_2$ in N_2 is satisfied, which results as follows:

$$m_2(p) = \begin{cases} m_N(p) \geq Pre^*_N(tr_N, p) = Pre^*_2(tr_2, p) & p \in P'_2 \\ m_N \circ f_1 \circ f_2^{-1}(p) \geq Pre^*_N(tr_N, f_1 \circ f_2^{-1}(p)) & \\ \quad = Pre^*_2(tr_2, p) & p \in f_2(P_0) \end{cases}$$

The capacity condition for the firing of tr_2 in N_2 at m_2 follows from the existence of m'_2 as a state in TG_2 (recall $(m'_1, m'_2) \in CS_3$).
For $p \in f_2(P_0)$ let $p' = f_2^{-1}(p)$. We have:
$m_2(p) - Pre_2(tr_2, p) + Post_2(tr_2, p) =$
$\quad = m_N(f_1(p')) - Pre_N(tr_N, f_1(p')) + Post_N(tr_N, f_1(p'))$
$\quad = m'_N(f_1(p')) = m'_1(f_1(p'))$
$\quad = m'_2(f_2(p')) = m'_2(p)$.
For $p \in P'_2$ we have

$m_2(p) - Pre_2(tr_2, p) + Post_2(tr_2, p) = m_N(p) - Pre_N(tr_N, p) + Post_N(tr_N, p)$
$= m_N(p) - Pre_N(tr_N, p) + Post_N(tr_N, p)$
$= m'_2(p)$.

Thus, there is a transition $tr_2 \in Tr_2(m_2, m'_2)$.
Now we show that $h_1(tr_1) = h_2(tr_2)$ According to the cospan property 3.11.5 and to the definition of (tr_1, tr_2) we have:

$$h_1(tr_1) = tr_1|_{T_1^0}[\sigma_{1M}^{-1}] = tr_N|_{T_1^0}[\sigma_{1M}^{-1}] = tr_2|_{T_2^0}[\sigma_{2M}^{-1}] = h_2(tr_2)$$

Thus:

$$tr_3 \in Tr_3(m_3, m'_3).$$

5. **$h : TS(N) \to TS_3$ and $h^{-1} : TS_3 \to TS(N)$ are $D\Gamma_N$-transformation system morphisms:**

For $m_3 = (m_1, m_2) \in CS_3$ and $tr_3 = (tr_1, tr_2) \in Tr_3$, let $tr_0 = h_1(tr_1)$ in:

$$u_3(m_3) = A(m_1) +_{(A(m_0), \sigma_{1\Gamma}, \sigma_{1\Gamma})} = A(m_1 \cup m_2|_{P'_2}) = n(h^{-1}(m_3))$$

$$\begin{aligned}
&u_3(tr_3) = n_1(tr_1) +_{n_0(tr_0), \sigma_{1M}, \sigma_{1M})} n_2(tr_2) \\
&= n_1(tr_1) \cup (n_2(tr_2) \setminus \sigma_{2M}(n_0(tr_0))) \\
&= n_1(tr_1) \cup n_2(tr_2|_{T'_2}) = n(tr_1 \oplus tr_2|_{T'_2}) \\
&= n \circ h^{-1}(tr_3)
\end{aligned}$$

Thus h^{-1} is a $D\Gamma_N$-transformation system morphism. That h is also a $D\Gamma_N$-transformation system morphism follows immediately from that:

$$n(tr_N) = n(h^{-1}(h(tr_N))) = u_3(h(tr_N))$$

$$n(m_N) = n(h^{-1}(h(m_N))) = u_3(h(m_N))$$

Discussion 3.16 (Glued places)
The condition 3.11.4 is necessary because otherwise transitions not to be glued would cause a state change that does not correspond to a statechange in TG_0. For example, if a transition $t_1 \in T'_1$ fires and puts a token on $p_1 \in P_1^0$, the following happens. Let for the transition $\{p_1 \mapsto 0\}[t_1\rangle\{p_1 \mapsto 1\}$. Let $tr_0 := h_1(t_1)$. If $h_2^{-1}(tr_0) = \emptyset$ then the transition $\{p_1 \mapsto 0\}[t_1\rangle\{p_1 \mapsto 1\}$ is deleted, it does not have any correspondent transition in TG_3. Otherwise, if

there is a $tr_2 = h_2^{-1}(tr_0)$, this means that there is a transition in TG_2 that has the same firing effects on places in TG_0 as tr_1. So in TG_3 the following step occurs: $(m_1, m_2) \stackrel{(t_1,tr_2)}{\longrightarrow} (m_1', m_2')$, where two different firing steps (not to be glued), each one of which should put a token on p_1, fire in parallel and put only one token on p_1. With other words the effect of the parallel firing step is reduced to its half.

Discussion 3.17 (Gluing in the Sequential Case)
Let $N_i = (P_i, T_i, Pre_i, Post_i, K_i)$ for $i \in \{1,2\}$ be two nets and let $TS(N_i) = (TG_i, n_i : TG_i \to \mathcal{D}_{D\Gamma_i})$ be their transformation systems. Let

$$TS(N_1) \xRightarrow{(\sigma_1,h_1)} TS_0 \xLeftarrow{(\sigma_2,h_2)} TS(N_2)$$

be the restriction of a cospan for the gluing synchronization, as defined in 3.11 to sequential nets. This means that the domains and codomains of the method components of the data space specification morphisms and the transition graph morphisms are restricted. The sequential net corresponding to the TS_3 is:

$$N = (P_N, T_N, Pre_N, Post_N, K_N)$$

Here, P_N and K_N are defined like in the sequential case, the transition set is

$$T_N = (T_1' \times T_2') \cup T_1 \cup T_2',$$

and for $Op \in \{Pre, Post\}$ let:

$$Op(t,p) = \begin{cases} Pre_1(t,p) & \text{if } t \in T_1, p \in P_1 \\ Pre_i(t_i,p) & \text{if } t = (t_1,t_2) \in T_1' \times T_2', p \in P_i \\ Pre_2(t,p) & \text{if } t \in T_2', p \in P_2 \\ 0 & \text{else} \end{cases}$$

Because in the synchronized composition operation transitions not to be glued within the nets are synchronized by the transition graph pullback, we needed a set $T_1' \times T_2'$ of new transitions in order to define a net that has equivalent behaviour to TS_3. This means that TG_3 and TG_N are isomorphic. Furthermore, there is an embedding transformation system morphism from TS_3 to $TS(N)$. However, the transformation systems are not isomorphic, which is due to their different method sets.

3.4 Or-Synchronization

In the following we show a synchronization operation in which each transition t that belongs to a designated subset T_1^0 of T_1, is synchronized with one of the transitions of a specified subset of T_2 or is deleted. Since every transition of T_1^0 is synchronized, if it occures, with one *or* another transition of the specified subset of T_2 we call this type of synchronization *Or-Synchronization*. The cospan of the operation is determined by two subsets T_1^0 of T_1 and T_2^0 of T_2, by a set T_0 and by two functions:

$$g_1 : T_1^0 \to T_0 \text{ and } g_2 : T_2^0 \to T_0$$

where g_1 must be injective (see discussion 3.22). Every transition of T_1^0 will be synchronized with one transition of from the set $g_2^{-1}(g_1(t_1))$.

In the following, we define an *or-cospan*, the synchronized composition of which has the described property, then we show behaviour equivalence to a transformation system of a Petri net N and the existence of a transformation system morphism, that embedds TS_3 into $TS(N)$. These properties we shall show first for the parallel and then for the sequential case.

Definition 3.18 (Or-cospan)
Let N_1, N_2 be two place-transition nets with disjoint sets of places and transitions, with $N_i = (P_i, T_i, Pre_i, Post_i, K_i)$ and let $TS(N_i) = (TG_i, n_i : TG_1 \to \mathcal{D}_{D\Gamma_i})$ be their transformation systems.

$$TS(N_1) \xRightarrow{(\sigma_1,h_1)} TS_0 \xLeftarrow{(\sigma_2,h_2)} TS(N_2)$$

is a cospan for the *or*-synchronized composition iff:

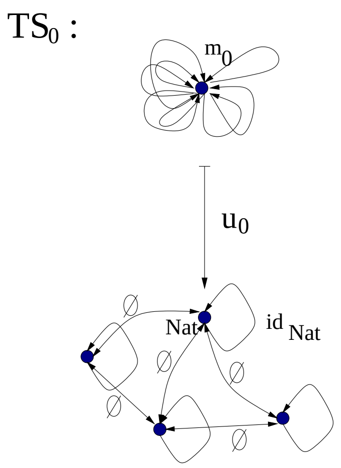
1. $TS_0 = (TG_0, u_0 : TG_0 \to \mathcal{D}_{D\Gamma_0})$ is given by:
$$\begin{array}{rl} D\Gamma_0 = & \Gamma_0 = nat \\ & M_0 = \emptyset \\ TG_0 = & (CS_0, Tr_0, CS_0, id) \\ & CS_0 = \{*\} \\ & Tr_0(*,*) = \mathcal{M}(T_0) \setminus \{\lambda\} \cup \{id_*\} \\ & id(*) = id_* \\ u_0 = \{ & * \mapsto Nat \\ & tr \mapsto \emptyset\} \end{array}$$

2. and the morphisms (σ_i, h_i) for $i \in \{1, 2\}$ have the following properties:

(a) $h_i(m_i) = *$ and
$h_1(\bigoplus_{t_1 \in T_1} k_{t_1} t_1) = \bigoplus_{t \in T_0} k_{g_1^{-1}(t)} t$
$h_2(\bigoplus_{t_2 \in T_2} k_{t_2} t_2) = \bigoplus_{t \in T_0} (\sum_{t_2 \in g_2^{-1}(t)} k_t) t$
Note that the cofficient of t in $h_2(tr_2)$ is the sum of the coefficients mapped to t by g_2.

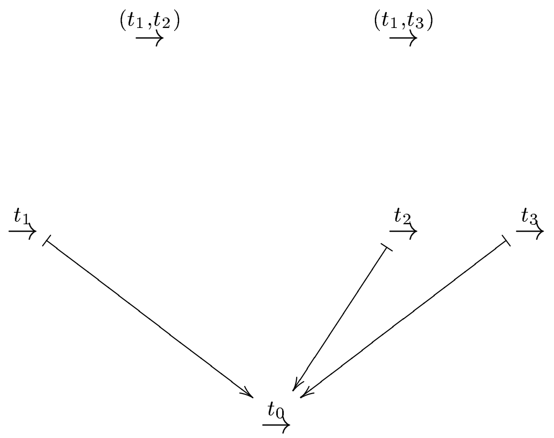
(b) $\sigma_i|_{nat}$ is the identity on nat

The following figure illustrates the gluing transformation system TS_0



Let $T_i' := T_i \setminus T_i^0$ for $i \in \{1, 2\}$. (Recall thet T_i^0 where the domains of g_i).

The following figure shows a part of the transition graphs of TG_3 (top), of TG_1 (left), of TG_2 (right) and of TG_0 (bottom), together with mappings that are part of h_1 and h_2. In TG_3 we can see the effect of mapping t_1, t_2 and t_3 to t_0: t_1 appears in TG_0 either synchronized to t_2 or to t_3.



We consider in the following subsection the parallel case, and in subsection 3.4.2 the sequential case.

3.4.1 Parallel Or-Synchronization

In the following fact we enumerate some properties of the transformation system TS_3 that results from the synchronized composition of $TS(N_1)$ and $TS(N_2)$ with the given cospan.

Fact 3.19 (Properties of the Result for Concurrent Nets)
Let $TS_3 = (TG_3, u_3 : TG_3 \to \mathcal{D}_{D\Gamma_3})$ be the synchronized composition of $TS(N_1)$ and $TS(N_2)$ over (σ_1, h_1) and (σ_2, h_2). The resulting data space specification is the union of the net data space specifications $D\Gamma_1$ and $D\Gamma_2$. The resulting transition graph $TG_3 = (CS_3, Tr_3, CS_3, id)$ is characterized by:

- $CS_3 = CS_1 \times CS_2$
- $Tr_3((m_1, m_2), (m_1', m_2')) = (\bigoplus_{t_1 \in T_1} k_{t_1} t_1, \bigoplus_{t_2 \in T_2} k_{t_2} t_2)$ where : $\sum_{t_2 \in g_2^{-1}(t)} k_{t_2} = k_{g_1^{-1}(t)}$ for all $t \in T_0$, with $\bigoplus_{t_1 \in T_1} k_{t_1} t_1 \in Tr_1(m_1, m_1')$, and $\bigoplus_{t_2 \in T_2} k_{t_2} t_2 \in Tr_2(m_2, m_2')$. This results from the definitions of h_1 and h_2.

We also have:

$u_3((m_1, m_2)) = A(m_1) +_{Nat} A(m_2)$ and

$$u((tr_1, tr_2)) = \{tr_1, tr_2\}$$

for $(tr_1, tr_2) \in Tr_3((m_1, m_2), (m'_1, m'_2))$

Proposition 3.20 (Behaviour equivalence: Net corresponding to TS_3)
Let the net

$$N = (P_1 \cup P_2, T_N, Pre_N, Post_N, K_1 \cup K_2)$$

be defined by

$$T_N = T'_1 \cup T'_2 \cup \{(t_1, t_2) \mid g_1(t_1) = g_2(t_2)\}$$

and

$$Pre_N(t,p) := \begin{cases} Pre_i(t,p) & t \in T'_i, p \in P_i \\ Pre_i(t_i,p) & p \in P_i, t = (t_1, t_2) \\ 0 & \text{else} \end{cases}$$

and analoguous for $Post_N$. Then the transition graphs TG_N and TG_3 are isomorphic.

Proof:
1. Isomorphism of transition graphs

We define $h : TG_3 \to TG_N$ and $h^{-1} : TG_N \to TG_3$ in order to show the isomorphism of the transition graphs.

- The state components

Let $m_3 = (m_1, m_2) \in CS_3$. Thus $m_i(p) \leq K_i(p)$ for all $p \in P_i, i \in \{1, 2\}$. Consequently $(m_1 \cup m_2)(p) \leq (K_1 \cup K_2)(p)$ for $p \in P_1 \cup P_2$ and we can define:

$$h(m_3) := m_1 \cup m_2$$

For $m_N \in CS_N$ we define:

$$h(m_N) := (m_N|_{P_1}, m_N|_{P_2})$$

The state $(m_N|_{P_1}, m_N|_{P_2})$ exists in CS_3 since $K_N = K_1 \cup K_2$, thus $m_i := m_N|_{P_i} \in CS_i$ for $i \in \{1, 2\}$ and since $h_1(m_i) = h_2(m_i) = *$.

- The transition components

Let $tr_3 \in Tr_3(m_3, m'_3)$ and $(tr_1, tr_2) := tr_3$, $(m_1, m_2) := m_3$ and $(m'_1, m'_2) = m'_3$. This implies:

$$Pre_i^*(tr_1, p) \leq m_1(p) \text{ for all } p \in P_1 \text{ and}$$
$$Pre_2^*(tr_2) \leq m_2(p) \text{ for all } p \in P_2 \tag{1}$$

where $Pre_i^*(\bigoplus_{t \in T_i} k_t t, p) = \sum_{t \in T_i} (k_t Pre_i(t,p))$. Let $\bigoplus_{t \in T_i} k_t t := tr_i$. Since $h_1(tr_1) = h_2(tr_2)$ we have for all $t_1 \in T_1^0$:

$$k_{t_1} = \sum_{t_2 \in g_2^{-1}(g_1(t_1))} k_{t_2} \tag{2}$$

Finally, according to the firing characteristics of N_1 and N_2 we have for all $p_i \in P_i$ and for $i \in \{1,2\}$:

$$m_i'(p_i) = m_i(p_i) - Pre_i^*(tr_i, p_i) + Post_i^*(tr_i, p_i) \leq K_i(p_i) \tag{3}$$

We define:

$$h(tr_3) := \bigoplus_{t \in T_1' \cup T_2'} k_t t \oplus \bigoplus_{(t_1,t_2) \in T_N} k_{t_2}(t_1, t_2) =: tr_N$$

We have according to (1) and (2):

$$Pre_N^*(tr_N, p) = \begin{cases} Pre_1^*((\bigoplus_{t \in T_1'} k_t t \oplus \bigoplus_{t \in T_1^0} (\sum_{(t_1,t_2) \in T_N} k_{t_2}) t_1), p) \leq m_1(p) & \text{if } p \in P_1 \\ Pre_2^*(\bigoplus_{t \in T_2} k_t t, p) \leq m_2(p) & \text{if } p \in P_2 \end{cases}$$

Thus $Pre_N^*(tr_N, p) \leq h(m_3)(p)$ for all $p \in P_N$. Analoguous to $Pre_N(tr_N, p)$, $Post_N(tr_N, p)$ can be calculated for all $p \in P_N$, implying with (3):

$$(h(m_3)(p) - Pre_N^*(tr_N, p)) + Post_N^*(tr_N, p) = h(m_3')(p)$$

Accordingly $tr_N \in Tr_N(h(m_3), h(m_3'))$.

Let $tr_N \in Tr_N(m_N, m_N')$ and $\bigoplus_{t \in T_N} k_t t := tr_N$. We define:

$$h^{-1}(tr_N) := (tr_1, tr_2)$$

where $tr_1 = tr_N|_{T_1'} \oplus \bigoplus_{(t_1,t_2) \in T_N} k_{(t_1,t_2)} t_1$ and $tr_2 = tr_N|_{T_2'} \oplus \bigoplus_{(t_1,t_2) \in T_N} k_{(t_1,t_2)} t_2$. We have: $Pre_N^*(tr_N, p) \leq m_N(p)$ for $p \in P_1 \cup P_2$ and

$$(m_N(p) - Pre_N^*(tr_N, p)) + Post_N^*(tr_N, p) = m_N'(p)$$

Thus:

$\forall p \in P_1 : m_1(p) \geq Pre^*_N(tr_N, p)$ and since
$Pre^*_N(tr_N, p) = Pre^*_1((\bigoplus_{t\in T'_1} k_t t \oplus \bigoplus_{t\in T^0_1} (\sum_{(t_1,t_2)\in T_N} k_{t_2}) t_1), p) \Rightarrow$
$m_1(p) \geq Pre^*_1(tr_1, p)$

similar, for $p \in P_2$:

$m_2(p) \geq Pre^*_2(tr_2, p)$

and, since $Post^*_i$ relates to $Post^*_N$ analogously to the relation between Pre^*_i and Pre^*_N we have for $p \in P_i$:

$(m_N(p) - Pre^*_N(tr_N, p)) + Post^*_N(tr_N, p) = m_i(p) - Pre^*_i(tr_i, p) + Post^*_i(tr_i, p)$
and, in consequence: $tr_1 \in Tr_1(m_1, m'_1)$ and $tr_2 \in Tr_2(m_2, m'_2)$

where $(m'_1, m'_2) = h^{-1}(m'_N)$. We still have to show that $h_1(tr_1) = h_2(tr_2)$. For $t \in T_0$ and for $tr \in \mathcal{M}(T_0)$ let $tr(t)$ be the coefficient of t in tr (i.e. $\bigoplus_{t\in T_0} k_t t(x) = k_x$ and a coefficient that does not exist in the sum is considered to be zero). We have for $t \in T_0$:

$$h_1(tr_1)(t) = k_{g_1^{-1}(t)} = \sum_{t_2 \in g_2^{-1} \circ g_1(g_1^{-1}(t))} k_{(t_1,t_2)}$$
$$= \sum_{t_2 \in g_2^{-1}(t)} k_{(g_1^{-1}(t), t_2)} = h_2(tr_2)(t)$$

Thus $h_1(tr_1) = h_2(tr_2)$ and $(tr_1, tr_2) \in Tr_3(h^{-1}(m_N), h^{-1}(m'_N))$. h^{-1} is the inverse function of h, and the transition graphs TG_3 and TG_N are isomorphic.

2. Transformation system We define $\sigma : (\Gamma_N, M_N) \to (\Gamma_3, M_3)$ such that $(\sigma, h) : TS_3 \to TS(N)$ is a transformation system morphism.
Let $\sigma_\Gamma(m_p) := m_p$ and let $\sigma|_{nat}$ be an inclusion. We define $\sigma_M(kt) = kt$ for $t \in T'_1 \cup T'_2$ and $k \in \mathbb{N}$ and
$\sigma_M(k(t_1, t_2)) = kt_2$ for $(t_1, t_2) \in T_N$.
Note that, since g_1 is injective and the transition t_2 appears only if it is synchronized with t_1 in TG_N there is a bijective mapping between $M_3 \setminus \{k_t t | t \in T^0_1, k_t \in \mathbb{N}\}$ and M_N. No method t_1 is addressed by σ_M, which has as a effect that, provided $(t_1, t_2) \in T_N$ and $k \in \mathbb{N}$:

$V_\sigma(u_3(kt_1 \oplus kt_2)) = V_\sigma(\{kt_1, kt_2\}) = \{k(t_1, t_2)\} = n(k(t_1, t_2)) = n \circ h(kt_1 \oplus kt_2)$

We define $\sigma(\bigoplus_{t \in T} k_t t) := \bigoplus_{t \in T} \sigma_M(k_t t)$. Then we have for $m_3 = (m_1, m_2) \in CS_3$ and $tr_3 = (tr_1, tr_2) \in Tr_3(m_3, m_3')$ with $(\bigoplus_{t \in T_1} k_t t, \bigoplus_{t \in T_1} k_t t) = tr_3$.

$V_\sigma(u_3(m_3)) = V_\sigma(A(m_1) +_{nat} A(m_2)) = A(m_1 \cup m_2) = n(m_1 \cup m_2) = n \circ h(m_3)$
$V_\sigma(u_3(tr_3)) = V_\sigma(\{k_t t \mid t \in T_1 \cup T_2\})$
$= \{k_{t'} t', k_t t \mid t' = (t_1, t_2) \in T_1^0 \times T_2^0, k_{t'} = k_{t_2}, t \in T_1' \cup T_2'\}$
$= n(\bigoplus_{t \in T_1^0 \times T_2^0} k_{t'} t' \oplus \bigoplus_{t \in T_1' \cup T_2'} k_t t) = n(h(tr_3))$

Remark 3.21 (Side effect: Deletion of transitions)
In the case that the set $C := g_1(T_1) \setminus g_2(T_2)$ is not empty, the transitions $t \in C$ do not appear in the resulting net, they are deleted. Also, the transitions $t \in g_2(T_2) \setminus g_1(T_1)$ are deletd.

Discussion 3.22 (Injectivness of g_1)
This condition is needed in order to provide an isomorphism between the transtition graph of the resulting transition system and the place-transition net transition system. Without this condition being satisfied we can have more transitions in the transition graph of the net for one transition in TG_3. For example, if $g_1(t_1) = g_1(t_1') = g_2(t_2) = g_2(t_2')$ we can define $h(t_1 \oplus t_1', t_2 \oplus t_2')$ both as $tr_N = (t_1, t_2) \oplus (t_2, t_2')$ and as $tr_N' = (t_1, t_2') \oplus (t_1', t_2)$. Since $h^{-1}(tr_N) = h^{-1}(tr_N')$ there is no way to define a graph isomorphism between the transition graph of the net transformation system (of N) and TS_3.

3.4.2 Sequential Or-Synchronization

The cospan for sequential or-synchronization is the same as in the parallel case, except for the transition graphs and method sets of $TS(N_1)$ and $TS(N_2)$ and for $Tr_0(*, *) = T_0 \cup id_*$.

The resulting transformation system has equivalent behaviour to the net:

$$N = (P_1 \cup P_2, T_N, Pre_N, Post_N, K_1 \cup K_2)$$

with Pre_N and $Post_N$ defined like in the parallel case and with

$$T_N = (T_1' \times T_2') \cup T_1' \cup T_2' \cup \{(t_1, t_2) \mid g(t_1) = g(t_2)\}.$$

The resulting transformation system

Let TS_3 be the resulting transformation system then we have:
$TS_3 = (CS_3, Tr_3, CS_3, id)$ where

$$CS_3 = CS_1 \times CS_2$$

$$Tr_3 = ((m_1, m_2), (m_1', m_2')) = \{(t_1, t_2) \in Tr_1(m_1, m_1') \times Tr_2(m_2, m_2') \mid h_1(t_1) = h_2(t_2)\}$$

$h_1(t_1) = h_2(t_2)$ happens exactly if either:

- $h_1(t_1) = h_2(t_2) = id(*)$ with the following possibilities
  - $t_1 \in T_1', t_2 \in T_2'$
  - $t_1 = id_{m_1}, t_2 = id_{m_2}$ $(m_1 = m_1', m_2 = m_2')$
  - $t_1 = id_{m_1}, t_2 \in T_2'$
  - $t_1 \in T_1', t_2 = id_{m_2}$
- $h_1(t_1) = h_2(t_2) \in T_0$ and
  - $g_1(t_1) = g_2(t_2)$

Proposition 3.23 (Behaviour equivalence: Net corresponding to TS_3)
Let the net

$$N = (P_1 \cup P_2, T_N, Pre_N, Post_N, K_1 \cup K_2)$$

be defined by

$$T_N = T_1' \cup T_2' \cup \{(t_1, t_2) \mid g_1(t_1) = g_2(t_2)\} \cup T_1' \cup T_2'$$

and

$$Pre_N(t, p) := \begin{cases} Pre_i(t, p) & t \in T_i', p \in P_i \\ Pre_i(t_i, p) & p \in P_i, t = (t_1, t_2) \\ 0 & \text{else} \end{cases}$$

and analoguous for $Post_N$. Then the transition graphs TG_N and TG_3 are isomorphic.

Proof: Let the transition graph isomorphisms for the parallel case be h_p and and h_p^{-1}. We define h and h^{-1}. Let their state components be the same as in the parallel case.

Let $tr_3 \in Tr_3((m_1, m_2), (m_1', m_2'))$ and let $(t_1, t_2) := tr_3$. Let:

$$h(tr_3) = \begin{cases} t_1 & t_2 = id(m_1) \\ t_2 & t_1 = id(m_2) \\ (t_1, t_2) & (t_1, t_2) \in (T_1' \times T_2') \cup (T_1^0 \times T_2^0) \end{cases}$$

For $tr_N \in Tr_N(m_N, m_N')$ let

$$h^{-1}(tr_N) = h_p^{-1}(tr_N) = \begin{cases} (t_1, t_2) & tr_N \in (T_1' \times T_2') \cup (T_1^0 \times T_2^0) \\ (t_1, id(m_N|_{P_2})) & tr_N \in T_1' \\ (id(m_N|_{P_1}), t_2) & tr_N \in T_2' \end{cases}$$

We showed in the proof for the sequential case that the resulting transitions fulfill the pre and capacity conditions, respectively that the components have the same image in TG_0 for a more general case.

The transition system morphism (σ, h) has $\sigma = (\sigma_\Gamma^p, \sigma_M^p|_{T_N})$ where σ^p is the data space specification morphism for the parallel case.

Discussion 3.24 (Or-Synchronization and Gluing)
We consider in the following parallel nets. If both the morphism g_1 and the morphism g_2 were injective, the resulting net (in the semantics of which TS_3 can be embedded) would be the same as the net resulting from a gluing operation, up to transition renaming. The gluing operation would have to obey $\Gamma_0 = nat$, $M_0 = T_0$, and that $\sigma_{iM} = g_i^{-1}$ and $\sigma_{i\Gamma}$ are inclusions.

3.5 Observations

In this section we make some informal observations on the effect of cospan properties we did not consider in the previous chapters.

We consider non-injective $\sigma_{i\Gamma}$ cospan morphism components, the representation of which includes gluing of places of the same net.

We also show how certain transition graph morphism properties result in a synchronized composition that can be interpreted as the behaviour restriction of a net.

3.5.1 State Transition Graph Morphism Components: Marking Sum Restriction

Analoguous to the mechanism to generate the or-synchronization we now present a case in which certain state transition graph morphism components induce a synchronization operation that can be reflected on the layer of nets. The resulting transition system can be, we believe, embedded in that of a net where gluing of places takes place in the sense that it forces a place of one net to always have as a marking the sum of the markings of other one, two or more places. Also, only firing steps remain, that maintain this sum.

Let N_1 and N_2 be two nets with $N_i = (P_i, T_i, Pre_i, Post_i, K_i)$. Let in the following

$$TS(N_1) \xRightarrow{(\sigma_1,h_1)} TS_0 \xLeftarrow{(\sigma_2,h_2)} TS(N_2)$$

be a cospan.

Similar to the or-synchronization, the operation is based on two functions $q_1 : P_1^0 \to P_0$ and $q_2 : P_2^0 \to P_0$, where $P_1^0 \subseteq P_1$, $P_2^0 \subseteq P_2$ and q_1 is injective. Moreover, $\Gamma_0 = nat$ and the method set M_0 is empty. The transition graph TG_0 has as state set $CS_0 = \mathcal{M}(P_0)$, which can be interpreted as all possible markings for a net with the place set P_0, and as transitions $Tr_0(m_0, m_0') = \{\tau\}$ if $m_0 \neq m_0'$, and $Tr_0(m, m) = id_m$. Since q_1 is injective, we work with it as if it were an inclusion. We define $h_1(m_1) = m_1|_{P_0}$ and $h_2(m_2)(p) = \sum_{p' \in q_2^{-1}(p)} m_2(p')$. We also define for $tr_1 \in Tr_1(m_1, m_1')$ and $tr_2 \in Tr_2(m_2, m_2')$ the transition components $h_1(tr_1) = \tau$ and $h_1(tr_2) = \tau$. Note that two transitions $m_1[tr_1\rangle m_1'$ and $m_2[tr_2\rangle m_2'$ are mapped to the same transition exactly if for all p in P_0

$$\sum_{p' \in q_2^{-1}(p)} m_2(p) = m_1(p) \text{ and}$$
$$\sum_{p' \in q_2^{-1}(p)} m'_2(p) = m'_1(p).$$

This means that a firing step that appears in TG_3 is bound to maintain state consistency with regard to the condition that the sum of markings of the places $p \in q_2^{-1}(p_0)$ equals the marking of p_0 in N_1 (Recall that q_1 is an inclusion):

$$\sum_{p' \in q_2^{-1}(p)} m_2(p) = m_1(p) \qquad (1).$$

Transition steps that do not maintain this condition are deleted. Transitions $m[t\rangle m' \in Tr_i$ that do not change the markings of any place in P_i^0 are mapped to identities on $h_i(m) = h_i(m')$ for $i \in \{1, 2\}$

We suppose that TG_3 is isomorphic in the parallel case to the restriction of the transtion graph of the net $N = (P_1 \uplus P_2, T_1 \uplus T_2, Pre_1 \uplus Pre_2, Post_1 \uplus Post_2, K_1 \uplus K_2)$ w.r.t condition (1). Here, $Op_1 \uplus Op_2$ means that pairs of places and transitions from different nets (i.e. $p \in P_1$, $t \in T_2$, etc.)are mapped to zero for $Op \in \{Pre, Post\}$. The following figure shows an example:

$$\begin{tikzcd}
 & & 3p_1 \arrow[r, "\tau"] & p_1 \oplus 2p_2 & \\
3p_1 \arrow[urr, "h_1"] \arrow[r, "tr_1"] & p_1 \oplus 2p_2 & & 2p'_1 \oplus p''_1 \arrow[ul, "h_2"'] \arrow[r, "tr_2"] & p'_1 \oplus p'_2 \oplus p''_2 \\
 & (3p_1, 2p'_1 \oplus p''_1) \arrow[ul, "\overline{h}_1"] \arrow[urr, "\overline{h}_2"'] \arrow[rrr, "(tr_1, tr_2)"] \arrow[d, "u_3"'] & & & (p_1 \oplus 2p_2, p'_1 \oplus p'_2 \oplus p''_2) \arrow[d, "u_3"] \\
 & A(3p_1) +_{Nat} A(2p'_1 \oplus p''_1) \arrow[rrr, "n_1(tr_1) \cup n_2(tr_2)"'] & & & A(p_1 \oplus 2p_2) +_{Nat} A(p'_1 \oplus p'_2 \oplus p''_2)
\end{tikzcd}$$

Here, $q_2(p'_i) = p_i$, $q_2(p''_i) = p_i$ for $i \in \{1, 2\}$.

The figure shows two data states and a transition resulting from the synchronized composition operation, while at the same time specifying some mappings of the morphisms h_1, h_2 and u_3. Note that there is a transformation $n(tr_1 \oplus tr_2) : A(p_1 \oplus 2p'_1 \oplus p_1) \to A(p_1 \oplus 2p_2 \oplus p'_1 \oplus p'_2 \oplus p_2)$ in N. That makes us suppose that there is an embedding morphism from TS_3 to $TS(N)$.

3.5.2 Non-Injective Place Mappings: Extended Place Gluing

While considering gluing of places in section 3.3 we considered only injective σ_Γ morphisms. We now consider the case in which this is not the case. Let in the cospan $TS(N_1) \xRightarrow{(\sigma_1,h_1)} TS_0 \xLeftarrow{(\sigma_2,h_2)} TS(N_2)$

$\sigma_{i\Gamma}|_{nat}$ be inclusions on nat. Let $f_i : P_0 \to P_i$ be defined along $\sigma_{i\Gamma}$ as follows: $f_i = \{p \mapsto p' \mid \sigma_{i\Gamma}(m_p) = m_{p'}\}$ The following figure shows mappings included in f_1 and f_2.

$$\begin{tikzcd}
 & p_1,\ p_2 \arrow[dl, mapsto, "f_1"'] \arrow[dr, mapsto, shift left, "f_2"] \arrow[dr, mapsto, shift right] & \\
p_{12} \arrow[dr, mapsto] & & p_1\ p_2 \arrow[dl, mapsto, shift left] \arrow[dl, mapsto, shift right] \\
 & {[p_{10}] = \{p_{12}, p_1, p_2\}} &
\end{tikzcd}$$

Note that the fact that (h_i, σ_i) have to be transformation system morphisms restricts here in a very useful way the possibility of defining the transition graph morphisms, once $\sigma_{1\Gamma}$ and $\sigma_{2\Gamma}$ are defined. It does allow mapping of states in CS_1 and CS_2 to states corresponding to marking restrictions. In fact, since

$$A(u_0(h(m_1))) = V_{\sigma_1}(n_1(m_1))$$

and there is a one to one correspondence between $A(m_0)$ and $m_0 \in CS_0$, the state to which m_1 has to be mapped is uniquely determined by this equation. Let the transitions of TG_0 be $T_1 \cup T_2$, and let them represent the effects of the transitions of N_1 and N_2 to P_0. Also, let $CS_0 = \mathcal{M}(P_0)$ Thus, in the example, $h(3p'_1) = 3p_1 \oplus 3p_2$, since $\sigma_{1\Gamma}(m_{p_1}) = \sigma_{1\Gamma}(m_{p_2}) = m_{p'_1}$. The effect is that that the only states in TG_0 that belong to the image of h_1 are characterized by equal markings of p_1 and p_2. Thus, if $(m_1, m_2) \in CS_3$, we must have $m_1(p'_1) = m_2(p_1) = m_2(p_2)$.

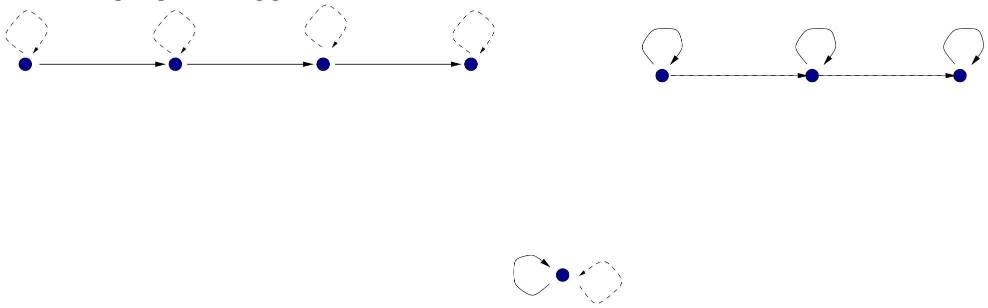
The gluing effect reached here synchronizes transitions (of TG_1 and TG_2) with equal effects on the places to glue. Transitions that do not have this equal effect are deleted. In the sequential case we think that the net corresponding to TS_3 can be constructed by a syntactical operation starting with N_1 and N_2 by using this remark (deleting *inconsistent* net transitions). In the parallel case, this would be difficult, because net transitions could belong to both *consistent* and *inconsistent* firing steps.

3.5.3 Mutual Exclusion

In the following we discuss whether the coproduct of sequential nets can be modelled by a composition of transformation systems.

We believe that it cannot be represented, because there is no way to define transition graph morphisms such that the resulting transition graph has subgraphs isomorphic to the transition graphs of the two nets, while at the same time assuring mutual exclusion of transitions coming from different subgraphs. Let $h_1 : TG_1 \to TG_0$ and $h_2 : TG_2 \to TG_0$ be the transition graph morphisms we are looking for, and let TG_3 be the object of the pullback of h_1 and h_2. Here, $TG_i = (CS_i, Tr_i, CS_i, id_i)$ for $i \in \{0, 1, 2, 3\}$.

In order that all combinations of states of the two net transition graphs occure in TG_3, all states of TG_1 and TG_2 must be mapped to the same state in TG_0. We name this state $*$. Now, in order not to delete the transitions of TG_1, all transitions of these graphs must be mapped to transitions in TG_0 where ONLY state identities from TG_2 are mapped. And here is the problem: all state identities of TG_2 are mapped, according to transition graph morphism definition to the identity of $*$. Thus all transitions of TG_1 should be mapped to $id_0(*)$ and no transition of TG_2 should be mapped to $id_0(*)$. Analogous, we can deduce that all transitions of TG_2 should be mapped to id_* and the transitions of TG_1 not. If the condition that identities have to be mapped to identities would not exist, mutual exclusion with maintanance of behaviour could be represented by synchronization as the example in the following figure suggests:



Chapter 4

Conclusion and Outlook

4.1 Summary

In this thesis we have translated place-transition nets and abgebraic high-level nets into algebra rewrite systems and gave a transformation system semantics for this formalisms. While doing this, we considered both sequential and concurrent nets. After defining the algebra rewrite specifications and the semantics of nets, we formally compared the semantics of the rewrite specifications and the net semantics on the layer of transformation systems. We showed equivalence of the semantics for sequential nets and for parallel AHL nets and isomorphism of transition graphs for parallel P/T nets.

In order to find out what representation the synchronized composition of place-transition net transformation systems has on the syntactical layer of P/T nets, we first made some considerations on the specific properties of the synchronized composition operation if it is applied to semantics of place-transition nets. We showed that no syntactical representation of the synchronized composition exists for the general case, and identified some reasons for this situation. This was to be expected, since the formalism of transformation systems is more general then the net formalism, and, accordingly, also its synchronization mechanism. However, the fact that we were synchronizing net transformation systems made restrictions to the synchronization morphisms, that reduced considerably the number of cases in which no net transformation system can be obtained. For example, in the case of place gluing caused by the gluing of place markings through pushout of specifications, the transition graph morphisms could only be defined such that only *consistent* states resulted, i.e. all states in the resulting transition graph had to be pairs of markings (m_1, m_2) where $\sigma_{1\Gamma}(m_p) = \sigma_{2\Gamma}(m_{p'})$ implied $m_1(p) = m_2(p')$.

Finally, we gave some cases in which the synchronized composition operation on net semantics does correspond to an operation on nets and we showed that correspondence. We gave the synchronized composition operations that correspond to gluing places and transitions of two nets, and found out that a certain synchronized composition type has a syntactical representation that we called or-synchronization. Furthermore we commented the syntactical representation of synchronized compositions with other cospan properties, trying to describe the way in which every degree of liberty in the definition of the cospan affects a possible syntactical representation on net layer.

Alltogether, we considered injective and non-injective mappings of place marking constants, injective and non-injective mappings of transitions (reflected in method components of data space specification morphisms), and state and transition components of transition graphs defined in different ways.

We were often able to show only isomorphism of transition graphs and/or embedding of the transformation system resulting from the synchronized composition into the semantics of a net delivering the resulting transition graph. The main reason for this is in the parallel case that in the net semantics there are a lot of method sets corresponding to the same net firing step, and the pushout on method sets often results in a set that can not be mapped by a forgetful functor to the method set of the same net firing step. If this is the case, usually it is impossible to define the inverse data space specification morphism such that isomorphism is achieved.

In order to establish precise correspondence in some cases it would be necessary to establish identification of union of sets and monoidal products.

In the sequential case, isomorphism could not be shown in most cases because we often had to introduce new transitions in order to represent synchronization on the net layer, which prevented isomorphism of the method sets and of the transition systems.

However, the result that gluing of places and transitions of two nets over a common subnet corresponds completely to a synchronized composition operation for parallel nets is a remarkable result, since this is probably the most used syntactical operation on nets. Also, that we were able to show isomorphism of transition graphs and embedding of the synchronized composition in the semantics of a net in the case of *or*-synchronization is remarkable, especially considering the fact that composition of monoidal structures is not well-supported within the synchronized composition of transformation systems.

4.2 Open Questions

We focused in our analysis of the correspondence between synchronized composition and syntactical net operations on elementary mechanisms, usually keeping all aspects except the one to be analyzed as simple as possible.

The question how these mechanisms would interact if more than one would occur in a synchronized composition operation has not been answered, and some of the aspects have not been analyzed formally.

Besides this version of ATS, there is a more general one in which the categories of partial algebras as data states and of sets as transformations can be replaced by other categories. Using multisets as transformations would better suite the formalism of Petri nets both with regard to the definition of semantics and with regard to the synchronization operation.

Another open question is what new aspects occure if algebraic high-level nets are considered instead of place-transition nets with regard to the synchronized composition operation. We suppose that sharing of data types together with gluing of places and transitions can be done in a similar way to the low-level case. Equivalence of transition firing conditions on the net parts to be glued together would have to be provided in addition to the conditions for the low-level case. Also, we see no reason for which or-synchronization and other mechanisms presented in this thesis would be prevented from having a generalized version suiting AHL nets.

Appendix

Specification and Algebra of Natural Numbers

Specification of natural numbers

$$
\begin{array}{lll}
nat & = & sorts: & nat \\
 & & opns: & 0 & :\rightarrow nat \\
 & & & succ & : nat \rightarrow nat \\
 & & & + & : nat\ nat \rightarrow nat \\
 & & & - & : nat\ nat \rightarrow nat \\
 & & & \geq & : nat\ nat \\
 & & eqns: & \multicolumn{2}{l}{add(n, 0) = n} \\
 & & & \multicolumn{2}{l}{add(n, succ(m)) = succ(add(n, m))}
\end{array}
$$

Algebra of natural numbers

$Nat = \quad (\mathbb{N}, 0, +1, +, -, \geq)$
Here, $+$ and $-$ are the normal operations on natural numbers, while $\geq$ is the predicate meaning greater or equal.

Specification of multisets

$$
\begin{array}{lcll}
mset(data) & = & sorts: & data, \quad mset \\
 & & opns: & \lambda_{data} \quad :\rightarrow mset \\
 & & & lift \quad : data \rightarrow mset \\
 & & & \oplus \quad : mset\ mset \rightarrow mset \\
 & & & \ominus \quad : mset\ mset \rightarrow mset \\
 & & & \geq \quad : mset\ mset \\
 & & eqns: & a \oplus b = b \oplus a \\
 & & & a \oplus b \ominus b = a \\
 & & & a \oplus \lambda = a \\
 & & & a \ominus b \oplus c = (a \ominus b) \oplus c \\
 & & & a \oplus b \ominus c = (a \oplus b) \ominus c \\
 & & & a \oplus b \geq a
\end{array}
$$

$\mathcal{M}(A)$ is a denotation for the set of multisets over A.

Extension to rules with conditions

In [Gro99c]rules are defined with two components, $r : P_l \longleftrightarrow P_r$, and a match satisfying the equations of P_l allowed the application of the rule. One rule with conditions as we will use is a rule as defined in [Gro99c], that can fire only if the match satisfies not only the equations of P_l and P_r, but also the conditions expressed in P_{cond}

Bibliography

[Gro99a] M. Grosse-Rhode. A compositional comparison of specifications of the alternating bit protocol in CCS and UNITY based on Algebra Transformation Systems. In A. Galloway and K. Taguchi, editors, *Proceedings of the 1st International Conference on Integrated Formal Methods*, pages 253–272. Springer, June 1999.

[Gro99b] M. Grosse-Rhode. On a reference model for the formalization and integration of software specification languages. *Bulletin of EATCS* , 68:81–89, 1999. The Formal Specification Column, Part 8, by Ehrig, H.

[Gro99c] M. Grosse-Rhode. Specification of state-based systems by Algebra Rewrite systems and Refinements. Technical Report 99-04, Technische Universität Berlin, 1999.

[Gro01] M. Grosse-Rhode. Semantic integration of specification formalisms via transformation systems. Technical Report 2001-13, Technische Universität Berlin, 2001.

[Hum89] U. Hummert. *Algebraische High-Level-Netze*. Ph.D. thesis, Technical University of Berlin, 1989.

[JPR95] H. Ehrig J. Padberg and L. Ribiero. Algebraic High-Level Net Transformation Systems. *Mathematical Structures in Computer Science* , 5:217–256, 1995.

[MM90] J. Meseguer and U. Montanari. Petri Nets are Monoids. *Information and Computation* , 88(2):105–155, 1990.

[Pad96] J. Padberg. *Abstract Petri Nets: A Uniform Approach and Rule-Based Refinement.* Shaker Verlag, 1996. Ph.D. thesis, Technical University of Berlin.

[Pet62] C. A. Petri. *Kommunikation mit Automaten.* Ph.D. thesis, Schriften des Instituts f"ur Instrumentelle Mathematik, Bonn, 1962.

[Vau87] J. Vautherin. Parallel System Specification with Coloured Petri Nets. *Advances in Petri Nets 87*, pages 293–308, 1987. Lecture Notes in Computer Science 266.

[Win87] G. Winskel. Petri nets, algebras, morphisms, and compositionality. *Information and Computation*, 72:197–238, 1987.