Electronic Communications of the EASST
Volume 2 (2006)

Proceedings of the
Workshop on Petri Nets and Graph Transformation
(PNGT 2006)

Algebraic High-Level Nets as Weak Adhesive HLR Categories

Ulrike Prange

13 pages

Guest Editors: Paolo Baldan, Hartmut Ehrig, Julia Padberg, Grzegorz Rozenberg
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Algebraic High-Level Nets as Weak Adhesive HLR Categories

Ulrike Prange

Department of Software Engineering and Theoretical Computer Science
Technical University of Berlin, Germany
uprange@cs.tu-berlin.de

Abstract: Adhesive high-level replacement (HLR) systems have been recently introduced as a new categorical framework for double pushout transformations. Algebraic high-level nets combine algebraic specifications with Petri nets to allow the modelling of data, data flow and data changes within the net.

In this paper, we show that algebraic high-level schemas and nets fit well into the context of weak adhesive HLR categories. This allows us to apply the developed theory also to algebraic high-level net transformations.

Keywords: Algebraic High-Level Nets, Adhesive HLR Categories

1 Introduction

Adhesive high-level replacement (HLR) systems have been recently introduced as a new categorical framework for graph transformation in the double pushout approach [EHPP06, EEPT06]. They combine the well-known framework of HLR systems with the framework of adhesive categories introduced by Lack and Sobociński [LS05]. The main concept behind adhesive categories are the so-called van Kampen squares, which ensure that pushouts along monomorphisms are stable under pullbacks and, vice versa, that pullbacks are stable under combined pushouts and pullbacks. In the case of adhesive HLR categories the class of all monomorphisms is replaced by a subclass $\mathscr{M}$ of monomorphisms closed under composition and decomposition.

Algebraic high-level (AHL) nets combine algebraic specifications with Petri nets [PER95] to allow the modelling of data, data flow and data changes within the net. In general, an AHL net denotes a net based on a specification SP in combination with an SP-algebra A, in contrast a net without a specific algebra is called a schema.

While many types of graphs and graph-like structures are adhesive HLR categories, the categories of elementary nets, place/transition nets as well as AHL schemas with fixed specification only satisfy a weaker version of adhesive HLR categories [EP06] called weak adhesive HLR categories. The reason is that the category **PTNets** of place/transition nets has general pullbacks, but pullbacks in general cannot be constructed componentwise in **Sets**. However, pullbacks along monomorphisms in **PTNets** can be constructed componentwise in **Sets**. This is the key idea to weaken the concept of adhesive HLR categories using weak VK squares. In this case, van Kampen squares ensure the corresponding properties only under stricter requirements on the morphisms. Nevertheless, the framework of weak adhesive HLR categories is still sufficient to show under some additional assumptions (which are necessary also in the non-weak case) as main results the Local Church-Rosser Theorem, the Parallelism Theorem, the Concurrency Theorem, the Embedding and Extension Theorem and the Local Confluence Theorem, also called Critical Pair Lemma. Thus, underlying an adhesive HLR systems we consider either a weak or a non-weak adhesive HLR category.

Since this concept of adhesive HLR systems includes all kinds of graphs mentioned above, and also elementary nets, place/transition nets and AHL schemas with fixed specification, adhesive HLR systems can be seen as a suitable unifying framework for graph and Petri net transformations.

The question arises, if and how different types of AHL schemas and nets, where we do not fix the algebra or the specification, fit into the framework of adhesive HLR systems.

In case of AHL nets with fixed specification *SP*, this category of AHL nets can be shown to be a weak adhesive HLR category, if the underlying category of algebras $\mathbf{Algs}(\mathbf{SP})$, together with a suitable morphism class $\mathcal{M}$, is a weak adhesive HLR category. Generalized AHL schemas, where the specification may change, can be shown to be a weak adhesive HLR category using an isomorphic comma category construction. In case both specification and algebra may change, the corresponding category of generalized AHL nets is a weak adhesive HLR category, if the category of all algebras can be shown to be a weak adhesive HLR category. These three results are the main new contributions of this paper.

This paper is organized as follows:
In Section 2, we introduce weak adhesive HLR categories and systems and review in Section 3, that different kinds of Petri nets are weak adhesive HLR categories. In Section 4, AHL schemas and nets are described and shown to be weak adhesive HLR categories. In Section 5, we present generalized AHL schemas and nets and prove the properties of a weak adhesive HLR category. At last, in Section 6 we give a conclusion and identify future work.

2 Review of Weak Adhesive HLR Categories and Systems

The intuitive idea of (weak) adhesive HLR categories are categories with suitable pushouts and pullbacks which are compatible with each other. More precisely the definition is based on so-called van Kampen squares.

The idea of a van Kampen (VK) square is that of a pushout which is stable under pullbacks, and vice versa that pullbacks are stable under combined pushouts and pullbacks.

Definition 1 A pushout (1) is a *van Kampen square*, if for any commutative cube (2) with (1) in the bottom and the back faces being pullbacks holds: the top face is a pushout if and only if the front faces are pullbacks.

$$\begin{tikzcd} A \arrow[r, "m"] \arrow[d, "f"'] \arrow[dr, phantom, "(1)"] & B \arrow[d, "g"] \\ C \arrow[r, "n"'] & D \end{tikzcd}
\qquad
\begin{tikzcd} & & A' \arrow[dll, "f'"'] \arrow[dr, "m'"] \arrow[dd, "a"] & \\ C' \arrow[dr, "n'"] \arrow[dd, "c"'] & & & B' \arrow[dll, "g'"'] \arrow[dd, "b"] \\ & D' \arrow[dd, "d"] & A \arrow[dll, "f"'] \arrow[dr, "m"] & \\ C \arrow[dr, "n"'] & & & B \arrow[dll, "g"] \\ & D & & \end{tikzcd}
\quad (2)$$

Since not even in the category **Sets** of sets and functions each pushout is a van Kampen square, for (weak) adhesive HLR categories only those VK squares of Definition 1 are considered where m is a monomorphism.

The main difference between (weak) adhesive HLR categories as described in [EHPP06, EEPT06] and adhesive categories introduced in [LS05] is that a distinguished class $\mathcal{M}$ of monomorphisms is considered instead of all monomorphisms, so that only pushouts along $\mathcal{M}$-morphisms have to be VK squares. In the weak case, only special cubes are considered for the VK square property.

Definition 2 A category $\mathbf{C}$ with a morphism class $\mathcal{M}$ is a *(weak) adhesive HLR category*, if

1. $\mathcal{M}$ is a class of monomorphisms closed under isomorphisms, composition ($f : A \to B \in \mathcal{M}, g : B \to C \in \mathcal{M} \Rightarrow g \circ f \in \mathcal{M}$) and decomposition ($g \circ f \in \mathcal{M}, g \in \mathcal{M} \Rightarrow f \in \mathcal{M}$),

2. $\mathbf{C}$ has pushouts and pullbacks along $\mathcal{M}$-morphisms and $\mathcal{M}$-morphisms are closed under pushouts and pullbacks,

3. pushouts in $\mathbf{C}$ along $\mathcal{M}$-morphisms are (weak) VK squares.

For a weak VK square, the VK square property holds for all commutative cubes with $m \in \mathcal{M}$ and ($f \in \mathcal{M}$ or $b,c,d \in \mathcal{M}$) (see Definition 1).

The categories **Sets** of sets and functions, **Graphs** of graphs and graph morphisms and $\mathbf{Graphs_{TG}}$ of typed graphs and typed graph morphisms are adhesive HLR categories for the class $\mathcal{M}$ of all monomorphisms. Moreover, an important example is the category $(\mathbf{AGraphs_{ATG}}, \mathcal{M})$ of typed attributed graphs with a type graph ATG and the class $\mathcal{M}$ of all injective morphisms with isomorphisms on the data part. The categories **ElemNets** of elementary nets and **PTNets** of place/transition nets with the class $\mathcal{M}$ of all corresponding monomorphisms fail to be adhesive HLR categories, but they are weak adhesive HLR categories (see Section 3).

Both adhesive and weak adhesive HLR categories are closed under product, slice, coslice, functor and comma category constructions. That means we can construct new (weak) adhesive HLR categories from given ones.

Theorem 1 *If $(\mathbf{C}, \mathcal{M}_1)$ and $(\mathbf{D}, \mathcal{M}_2)$ are (weak) adhesive HLR categories, then the following categories are also (weak) adhesive HLR categories:*

1. *the product category $(\mathbf{C} \times \mathbf{D}, \mathcal{M}_1 \times \mathcal{M}_2)$,*

2. *the slice category $(\mathbf{C}\backslash X, \mathcal{M}_1 \cap \mathbf{C}\backslash X)$ and the coslice category $(X\backslash\mathbf{C}, \mathcal{M}_1 \cap X\backslash\mathbf{C})$ for any object X in $\mathbf{C}$,*

3. *for every category $\mathbf{X}$ the functor category $([\mathbf{X}, \mathbf{C}], \mathcal{M}_1\text{-functor transformations})$, where an $\mathcal{M}_1$-functor transformation is a natural transformation $t : F \to G$ where all morphisms $t_X : F(X) \to G(X)$ are in $\mathcal{M}_1$,*

4. *the comma category $(ComCat(F,G;\mathcal{I}), \mathcal{M})$ with $\mathcal{M} = (\mathcal{M}_1 \times \mathcal{M}_2) \cap Mor_{ComCat(F,G;\mathcal{I})}$ and functors $F : \mathbf{C} \to \mathbf{X}$, $G : \mathbf{D} \to \mathbf{X}$, where F preserves pushouts along $\mathcal{M}_1$-morphisms and G preserves pullbacks (along $\mathcal{M}_2$-morphisms).*

Now we are able to generalize graph transformation systems, grammars and languages in the sense of [Ehr79, EEPT06].

In general, an adhesive HLR system is based on productions, also called rules, that describe in an abstract way how objects in this system can be transformed. An application of a production is called a direct transformation and describes how an object is actually changed by the production. A sequence of these applications yields a transformation.

Definition 3 Given a (weak) adhesive HLR category $(\mathbf{C}, \mathscr{M})$, a *production* $p = (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ (also called rule) consists of three objects L, K and R called left hand side, gluing object and right hand side respectively, and morphisms $l : K \to L$, $r : K \to R$ with $l, r \in \mathscr{M}$.

Given a production $p = (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ and an object G with a morphism $m : L \to G$, called match, a *direct transformation* $G \stackrel{p,m}{\Longrightarrow} H$ from G to an object H is given by the following diagram, where (1) and (2) are pushouts. A sequence $G_0 \Rightarrow G_1 \Rightarrow \ldots \Rightarrow G_n$ of direct transformations is called a *transformation* and is denoted as $G_0 \stackrel{*}{\Rightarrow} G_n$.

$$\begin{tikzcd} L \arrow[d, "m"] \arrow[dr, phantom, "(1)"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "k"] \arrow[dr, phantom, "(2)"] & R \arrow[d, "n"] \\ G & D \arrow[l, "f"] \arrow[r, "g"'] & H \end{tikzcd}$$

An *adhesive HLR system* $AHS = (\mathbf{C}, \mathscr{M}, P)$ consists of a (weak) adhesive HLR category $(\mathbf{C}, \mathscr{M})$ and a set of productions P.

3 Petri Nets as Weak Adhesive HLR Categories

Petri net transformation systems have been first introduced in [EHKP91] for the case of low-level nets and in [PER95] for high-level nets using the algebraic presentation of Petri nets as monoids introduced in [MM90]. The main idea of Petri net transformation systems is to extend the well-known theory of Petri nets based on the token game by general techniques which allow to change also the structure of the nets. In [Pad96], a systematic study of Petri net transformation systems has been presented in the categorical framework of abstract Petri nets, which can be instantiated to different kinds of low-level and high-level Petri nets.

In this section we introduce our notion of elementary nets and place/transition nets and recapitulate that the respective categories $(\mathbf{ElemNets}, \mathscr{M})$ and $(\mathbf{PTNets}, \mathscr{M})$ are weak adhesive HLR categories (see [EP06]). The corresponding instantiations of adhesive HLR systems lead to different kinds of Petri net transformation systems.

Definition 4 An *elementary net* is given by $N = (P, T, pre, post)$ with a set P of places, T of transitions, and pre- and post-domain functions $pre, post : T \to \mathscr{P}(P)$, where $\mathscr{P}$ is the power set functor.

An *elementary net morphism* $f : N \to N'$ is given by $f = (f_P : P \to P', f_T : T \to T')$ compatible with the pre- and post-domain functions, i.e. $pre' \circ f_T = \mathscr{P}(f_P) \circ pre$ and $post' \circ f_T = \mathscr{P}(f_P) \circ post$.

Elementary nets and elementary net morphisms form the category **ElemNets**.

Corollary 1 *The category $(\mathbf{ElemNets}, \mathscr{M})$ is a weak adhesive HLR category, where $\mathscr{M}$ is the class of all injective morphisms.*

Proof Idea. The category **ElemNets** is isomorphic to the comma category $ComCat(ID_{\mathbf{Sets}}, \mathscr{P}; \mathscr{I})$, where $\mathscr{P} : \mathbf{Sets} \to \mathbf{Sets}$ is the power set functor and $\mathscr{I} = \{1,2\}$. According to Theorem 1.4 it suffices to note that $(\mathbf{Sets}, \mathscr{M})$ is a weak adhesive HLR category and that $\mathscr{P} : \mathbf{Sets} \to \mathbf{Sets}$ preserves pullbacks along injective morphisms. $\square$

Note, that $(\mathbf{ElemNets}, \mathscr{M})$ is not an adhesive HLR category as stated in [EEPT06], since $\mathscr{P}$ only preserves pullbacks along injective morphisms, but not over general ones.

Definition 5 A *place/transition net* $N = (P, T, pre, post)$ is given by a set P of places, a set T of transitions, as well as pre- and post-domain functions $pre, post : T \to P^{\oplus}$, where $P^{\oplus}$ is the free commutative monoid over P.

A *place/transition net morphism* $f : N \to N'$ is given by $f = (f_P : P \to P', f_T : T \to T')$ compatible with the pre- and post-domain functions, i.e. $pre' \circ f_T = f_P^{\oplus} \circ pre$ and $post' \circ f_T = f_P^{\oplus} \circ post$.

Place/transition nets and place/transition net morphisms form the category **PTNets**.

Corollary 2 *The category $(\mathbf{PTNets}, \mathscr{M})$ is a weak adhesive HLR category, if $\mathscr{M}$ is the class of all injective morphisms.*

Proof Idea. The category **PTNets** is isomorphic to the comma category $ComCat(ID_{\mathbf{Sets}}, \Box^{\oplus}; \mathscr{I})$ with $\mathscr{I} = \{1,2\}$, where $\Box^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ is the free commutative monoid functor. According to Theorem 1.4 it suffices to note that $(\mathbf{Sets}, \mathscr{M})$ is a weak adhesive HLR category and that $\Box^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ preserves pullbacks along injective morphisms. $\square$

The following example shows that $(\mathbf{PTNets}, \mathscr{M})$ is not an adhesive HLR category. This is due to the fact, that $\Box^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ does not preserve general pullbacks. This would imply that pullbacks in **PTNets** are constructed componentwise for places and transitions.

Example 1 *In Figure 1, the square (1) with non-injective morphisms g_1, g_2, p_1, p_2 is a pullback in the category* **PTNets***, where the transition component is not a pullback in* **Sets***. In the cube, the bottom face is a pushout in* **PTNets** *along an injective morphism $m \in \mathscr{M}$, all front and back faces are pullbacks, but the top face is no pushout. Hence, this cube violates the VK property.*

4 AHL Schemas and Nets as Weak Adhesive HLR Categories

In this section, we combine algebraic specifications with Petri nets leading to AHL schemas and nets (see [PER95]). Intuitively, an AHL net is a Petri net, where ordinary, uniform tokens are replaced by data elements from the given algebra. Firing a transition t means to remove some data elements from the input places and add some data elements, computed by term evaluation, to the output places of t. There could be also some firing conditions to restrict the firing behaviour of a transition. In addition, a typing of the places restricts the data elements which could be put on each place to that of a certain type.

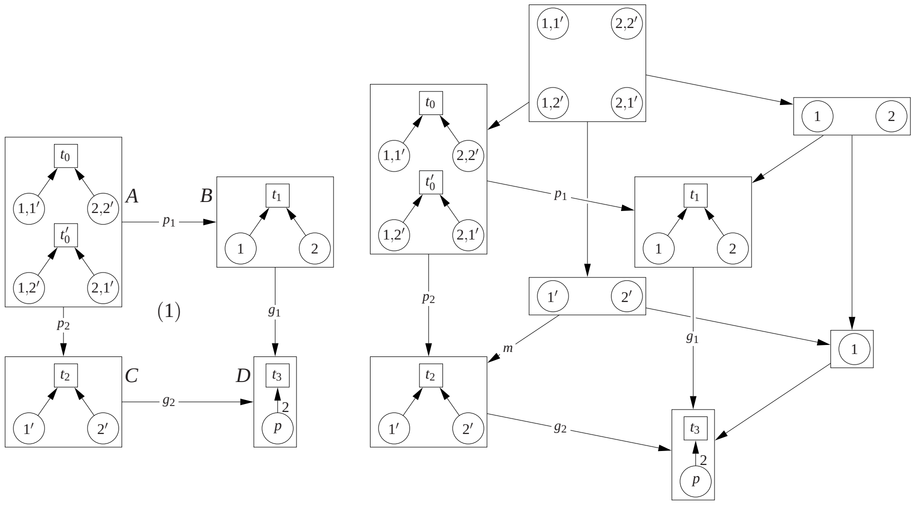


Figure 1: The pullback (1) in **PTNets** and the cube violating the VK property

Definition 6 An *AHL schema* over an algebraic specification SP, where $SP = (SIG, E, X)$ has additional variables X and $SIG = (S, OP)$, is given by $AS = (P, T, pre, post, cond, type)$ with sets P and T of places and transitions, $pre, post : T \to (T_{SIG}(X) \otimes P)^{\oplus}$ as pre- and post-domain functions, $cond : T \to \mathscr{P}_{fin}(Eqns(SIG, X))$ assigning to each $t \in T$ a finite set $cond(t)$ of equations over SIG and X, and $type : P \to S$ a type function. Note that $T_{SIG}(X)$ is the SIG-term algebra with variables X and $(T_{SIG}(X) \otimes P) = \{(term, p) \mid term \in T_{SIG}(X)_{type(p)}, p \in P\}$.

An *AHL schema morphism* $f : AS \to AS'$ is given by a pair of functions $f = (f_P : P \to P', f_T : T \to T')$ which are compatible with *pre*, *post*, *cond* and *type* as shown below.

$$\begin{tikzcd}
& T \arrow[dl, "cond"'] \arrow[r, shift left, "pre"] \arrow[r, shift right, "post"'] \arrow[dd, "f_T"] & (T_{SIG}(X) \otimes P)^{\oplus} \arrow[dd, "(id \otimes f_P)^{\oplus}"] \\
\mathscr{P}_{fin}(Eqns(SIG, X)) \arrow[r, phantom, "="] & {} \arrow[r, phantom, "="] & {} \\
& T' \arrow[ul, "cond'"] \arrow[r, shift left, "pre'"] \arrow[r, shift right, "post'"'] & (T_{SIG}(X) \otimes P')^{\oplus}
\end{tikzcd}
\qquad
\begin{tikzcd}
P \arrow[dr, "type"] \arrow[dd, "f_P"'] & \\
{} \arrow[r, phantom, "="] & S \\
P' \arrow[ur, "type'"'] &
\end{tikzcd}$$

Given an algebraic specification SP, AHL schemas over SP and AHL schema morphisms form the category **AHLSchemas**$(\mathbf{SP})$.

As shown in [EEPT06], AHL schemas over a fixed algebraic specification SP are a weak adhesive HLR category.

Corollary 3 *The category* $(\mathbf{AHLSchemas(SP)}, \mathscr{M})$ *is a weak adhesive HLR category.* $\mathscr{M}$ *is the class of all injective morphisms* f*, i.e.* f_P *and* f_T *are injective.*

Proof Idea. Since SP is fixed, the construction of pushouts and pullbacks in **AHLSchemas**$(\mathbf{SP})$ is essentially the same as in **PTNets**, which is already a weak adhesive HLR category. We can apply the idea of comma categories $ComCat(F, G; \mathscr{I})$, where in our case the source functor of the operations $pre, post, cond, type$ is always the identity $ID_{\mathbf{Sets}}$, and the target functors are $(T_{SIG}(X) \otimes \_)^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ and two constant functors. In fact, $(T_{SIG}(X) \otimes \_) : \mathbf{Sets} \to \mathbf{Sets}$, the constant functors and $\square^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ preserve pullbacks along injective functions. Hence also $(T_{SIG}(X) \otimes \_)^{\oplus} : \mathbf{Sets} \to \mathbf{Sets}$ preserves pullbacks along injective functions. $\square$

To represent the actual data space, we combine AHL schemas and algebras to AHL nets.

Definition 7 An *AHL net* $AN = (S,A)$ is given by an AHL schema S over SP and an SP-algebra A.

An *AHL net morphism* $f : AN \to AN'$ is given by a pair $f = (f_S : S \to S', f_A : A \to A')$, where f_S is an AHL schema morphism and f_A an SP-homomorphism.

Given an algebraic specification SP, AHL nets over SP and AHL net morphisms form the category $\mathbf{AHLNets}(\mathbf{SP})$.

Corollary 4 *If* $(\mathbf{Algs}(\mathbf{SP}),\mathscr{M})$ *is a weak adhesive HLR category then the category* $(\mathbf{AHLNets}(\mathbf{SP}),\mathscr{M}')$ *is a weak adhesive HLR category.* $\mathscr{M}'$ *is the class of all morphisms* $f = (f_S, f_A)$*, where* f_S *is injective and* $f_A \in \mathscr{M}$.

Proof Idea. The category $\mathbf{AHLNets}(\mathbf{SP})$ is isomorphic to the product category $\mathbf{AHLSchemas}(\mathbf{SP}) \times \mathbf{Algs}(\mathbf{SP})$. According to Theorem 1.1 this implies that $(\mathbf{AHLNets}(\mathbf{SP}),\mathscr{M}')$ is a weak adhesive HLR category. □

Up to now, it is not clear whether the category $\mathbf{Algs}(\mathbf{SP})$ of algebras over an arbitrary specification SP with the class $\mathscr{M}$ of injective morphisms is a weak adhesive HLR category. This has been shown in [EEPT06] only for so-called graph structure algebras, where only unary operations are allowed. For an arbitrary specification, we can use the class $\mathscr{M}$ of isomorphisms to obtain a weak adhesive HLR category.

5 Generalized AHL Schemas and Nets as Weak Adhesive HLR Categories

We get a more powerful variant of AHL schemas, called generalized AHL schemas, if we do not fix the specification. This is especially useful for net transformations, where we can define rules based on a (small) specification SP, which represents the necessary data, that can be applied to nets over a (larger) specification SP'.

In this section, we define generalized AHL schemas and nets, and show that they form weak adhesive HLR categories under certain conditions on the data part.

Definition 8 A *generalized AHL schema* $GS = (SP,AS)$ is given by an algebraic specification SP and an AHL schema AS over SP.

A *generalized AHL schema morphism* $f : GS \to GS'$ is a tuple $f = (f_{SP} : SP \to SP', f_P : P \to P', f_T : T \to T')$, where f_{SP} is a specification morphism and f_P, f_T are compatible with *pre*, *post*, *cond* and *type*. $f_{SP}^{\#}$ is the extension of f_{SP} to terms and equations.

$$\begin{tikzcd}
\mathscr{P}_{fin}(Eqns(SIG,X)) \arrow[d, "\mathscr{P}_{fin}(f_{SP}^{\#})"'] \arrow[dr, phantom, "="] & T \arrow[l, "cond"'] \arrow[r, shift left, "pre"] \arrow[r, shift right, "post"'] \arrow[d, "f_T"] \arrow[dr, phantom, "="] & (T_{SIG}(X)\otimes P)^{\oplus} \arrow[d, "(f_{SP}^{\#}\otimes f_P)^{\oplus}"] & P \arrow[r, "type"] \arrow[d, "f_P"'] \arrow[dr, phantom, "="] & S \arrow[d, "f_{SP,S}"] \\
\mathscr{P}_{fin}(Eqns(SIG',X)) & T' \arrow[l, "cond'"] \arrow[r, shift left, "pre'"] \arrow[r, shift right, "post'"'] & (T_{SIG'}(X)\otimes P')^{\oplus} & P' \arrow[r, "type'"] & S'
\end{tikzcd}$$

Generalized AHL schemas and generalized AHL schema morphisms form the category $\mathbf{AHLSchemas}$.

To show that generalized AHL schemas form a weak adhesive HLR category, we need an extension of comma categories, where we loosen the restrictions on the domain of the functors.

Definition 9 Given index sets $\mathscr{I}$ and $\mathscr{J}$, categories $\mathbf{C}_j$ for $j \in \mathscr{J}$ and $\mathbf{X}_i$ for $i \in \mathscr{I}$, and for each $i \in \mathscr{I}$ two functors $F_i : \mathbf{C}_{k_i} \to \mathbf{X}_i$, $G_i : \mathbf{C}_{\ell_i} \to \mathbf{X}_i$ with $k_i, \ell_i \in \mathscr{J}$, then the *general comma category* $GComCat((\mathbf{C}_j)_{j\in\mathscr{J}}, (F_i, G_i)_{i\in\mathscr{I}}; \mathscr{I}, \mathscr{J})$ is defined by

- objects $((A_j \in \mathbf{C}_j)_{j\in\mathscr{J}}, (op_i)_{i\in\mathscr{I}})$, where $op_i : F_i(A_{k_i}) \to G_i(A_{\ell_i})$ is a morphism in $\mathbf{X_i}$,
- morphisms $h : ((A_j), (op_i)) \to ((A'_j), (op'_i))$ as tuples $h = ((h_j : A_j \to A'_j)_{j\in\mathscr{J}})$ such that for all $i \in \mathscr{I}$ $op'_i \circ F_i(h_{k_i}) = G_i(h_{\ell_i}) \circ op_i$.

We can extend the result from Theorem 1.4 to general comma categories, such that the general comma category is a weak adhesive HLR category under certain conditions.

Theorem 2 *A general comma category* $\mathbf{GC} = (GComCat((\mathbf{C}_j)_{j\in\mathscr{J}}, (F_i, G_i)_{i\in\mathscr{I}}; \mathscr{I}, \mathscr{J}), \mathscr{M})$ *with* $\mathscr{M} = (\times_{j\in\mathscr{J}} \mathscr{M}_j) \cap Mor_{\mathbf{GC}}$ *is a weak adhesive HLR category, if* $(\mathbf{C}_j, \mathscr{M}_j)$ *are weak adhesive HLR categories for* $j \in \mathscr{J}$*, and for all* $i \in \mathscr{I}$ F_i *preserves pushouts along* $\mathscr{M}_{k_i}$*-morphisms and* G_i *preserves pullbacks along* $\mathscr{M}_{\ell_i}$*-morphisms.*

Proof Idea. It is easy to show that $\mathscr{M}$ is a class of monomorphisms closed under isomorphisms, composition and decomposition since this holds for all components $\mathscr{M}_j$.

As in normal comma categories, pushouts along $\mathscr{M}$-morphisms are constructed componentwise in the underlying categories. The pushout object is the componentwise pushout object, where the operations are uniquely defined using the property that F_i preserves pushouts along $\mathscr{M}_{k_i}$-morphisms.

Analogously, pullbacks along $\mathscr{M}$-morphisms are constructed componentwise, where the operations of the pullback object are uniquely defined using the property that G_i preserves pullbacks along $\mathscr{M}_{\ell_i}$-morphisms.

The weak VK square property follows, since in a proper cube, all pushouts and pullbacks can be decomposed leading to proper cubes in the underlying categories, where the weak VK property holds. The subsequent recomposition yields the weak VK property for the general comma category. □

Also the restriction of a weak adhesive HLR category to a full subcategory yields a weak adhesive HLR category, if the pushouts and pullbacks over $\mathscr{M}$-morphisms are preserved.

Corollary 5 *Given a weak adhesive HLR category* $(\mathbf{C}, \mathscr{M})$*, a full subcategory* $(\mathbf{C}', \mathscr{M}')$ *of* $\mathbf{C}$ *with* $\mathscr{M}' = \mathscr{M}|_{\mathbf{C}'}$ *is a weak adhesive HLR category, if* $\mathbf{C}'$ *has pushouts and pullbacks along* $\mathscr{M}'$*-morphisms which are preserved by the inclusion functor.*

Proof Idea. By precondition, pushouts and pullbacks along $\mathscr{M}'$-morphisms in $\mathbf{C}'$ exist. Obviously, $\mathscr{M}'$ is a class of monomorphisms with the required properties. Since we only restrict the objects and morphisms, the weak VK square property is inherited from $\mathbf{C}$. □

With these results, we are now able to show that the category of generalized AHL schemas is a weak adhesive HLR category.

Theorem 3 *The category* $(\mathbf{AHLSchemas}, \mathscr{M})$ *is a weak adhesive HLR category.* $\mathscr{M}$ *is the class of all morphisms* $f = (f_{SP}, f_P, f_T)$*, where* f_{SP} *is a strict injective specification morphism and* f_P, f_T *are injective.*

Proof. The category **AHLSchemas** is isomorphic to a suitable full subcategory of the general comma category $\mathbf{GC} = GComCat(\mathbf{C}_1, \mathbf{C}_2, (F_i, G_i)_{i \in \mathscr{I}}; \mathscr{I}, \mathscr{J})$ with

- $\mathscr{I} = \{pre, post, cond, type\}$, $\mathscr{J} = \{1, 2\}$,
- $\mathbf{C}_1 = \mathbf{Specs} \times \mathbf{Sets}$, $\mathbf{C}_2 = \mathbf{Sets}$, $\mathbf{X}_i = \mathbf{Sets}$ for all $i \in \mathscr{I}$,
- $F_i : \mathbf{C}_2 \to \mathbf{X}_i$ for $i \in \{pre, post, cond\}$, $F_{type} : \mathbf{C}_1 \to \mathbf{X}_{type}$, $G_i : \mathbf{C}_1 \to \mathbf{X}_i$ for all $i \in \mathscr{I}$,

where the functors are defined by

- $F_i = Id_{\mathbf{Sets}}$, $G_i(SP, P) = (T_{SIG}(X) \times P)^{\oplus}$, $G_i(f_{SP}, f_P) = (f_{SP}^{\#} \times f_P)^{\oplus}$ for $i \in \{pre, post\}$,
- $F_{cond} = Id_{\mathbf{Sets}}$, $G_{cond}(SP, P) = \mathscr{P}_{fin}(Eqns(SIG, X))$, $G_{cond}(f_{SP}, f_P) = \mathscr{P}_{fin}(f_{SP}^{\#})$,
- $F_{type}(SP, P) = P$, $F_{type}(f_{SP}, f_P) = f_P$, $G_{type}(SP, P) = S$, $G_{type}(f_{SP}, f_P) = f_{SP,S}$.

Since $(\mathbf{Specs}, \mathscr{M}_1)$ with the class $\mathscr{M}_1$ of strict injective morphisms and $(\mathbf{Sets}, \mathscr{M}_2)$ with the class $\mathscr{M}_2$ of injective morphisms are weak adhesive HLR categories, Theorem 1.1 implies that also $(\mathbf{Specs} \times \mathbf{Sets}, \mathscr{M}_1 \times \mathscr{M}_2)$ is a weak adhesive HLR category.

The functors F_i preserve pushouts along $\mathscr{M}_{k_i}$-morphisms, which is obvious for $F_{pre}, F_{post}, F_{cond}$ and shown in Corollary 6 for F_{type}, and the functors G_i preserve pullbacks along $\mathscr{M}_{\ell_i}$-morphisms as shown in Corollary 7, Corollary 8 and Corollary 9, therefore we can apply Theorem 2 such that **GC** is a weak adhesive HLR category.

Now we restrict the objects $((SP, P), T, pre, post, cond, type)$ in **GC** to those, where

$$(1) \quad pre(t), post(t) \in (T_{SIG}(X) \otimes P)^{\oplus} \text{ for all } t \in T.$$

The full subcategory induced by these objects is isomorphic to **AHLSchemas**. Since the condition (1) is preserved by pushout and pullback constructions in **GC**, it follows that for morphisms $f, g \in \mathbf{AHLSchemas}$ with the same (co)domain, the pushout (pullback) over f, g in **GC** is also the pushout (pullback) in **AHLSchemas**. With Corollary 5 we conclude that $(\mathbf{AHLSchemas}, \mathscr{M})$ is a weak adhesive HLR category. □

Corollary 6 *The functor* $H : \mathbf{Specs} \times \mathbf{Sets} \to \mathbf{Sets} : (SP, M) \mapsto M, (f_{SP}, f_M) \mapsto f_M$ *preserves pushouts (along* $\mathscr{M}_1 \times \mathscr{M}_2$*-morphisms).*

Proof. In a product category, a square is a pushout if and only if the componentwise squares are pushouts in the underlying categories. Thus, if (1) is a pushout in $\mathbf{Specs} \times \mathbf{Sets}$ also (2) is a pushout in **Sets**, which means that H preserves pushouts.

$$\begin{tikzcd}
(SP_0, M_0) \arrow[r, "(f_{SP}, f_M)"] \arrow[d, "(g_{SP}, g_M)"'] \arrow[dr, phantom, "(1)"] & (SP_1, M_1) \arrow[d, "(g'_{SP}, g'_M)"] & M_0 \arrow[r, "f_M"] \arrow[d, "g_M"'] \arrow[dr, phantom, "(2)"] & M_1 \arrow[d, "g'_M"] \\
(SP_2, M_2) \arrow[r, "(f'_{SP}, f'_M)"'] & (SP_3, M_3) & M_2 \arrow[r, "f'_M"'] & M_3
\end{tikzcd}$$

□

Corollary 7 *The functor H : $\mathbf{Specs} \times \mathbf{Sets} \to \mathbf{Sets}$: $(SP, M) \mapsto S, (f_{SP}, f_M) \mapsto f_{SP,S}$ preserves pullbacks (along $\mathcal{M}_1 \times \mathcal{M}_2$-morphisms).*

Proof. In a product category, a square is a pullback if and only if the componentwise squares are pullbacks in the underlying categories. Thus, if (3) is a pullback in **Specs** $\times$ **Sets** also (4) is a pullback in **Specs**. In **Specs**, pullbacks are constructed componentwise on the signature part (with some special treatment of the equations). Thus, also (5) is a pullback in **Sets**, which means that H preserves pullbacks.

$$
\begin{tikzcd}
(SP_0,M_0) \arrow[r, "(f_{SP},f_M)"] \arrow[d, "(g_{SP},g_M)"'] \arrow[dr, phantom, "(3)"] & (SP_1,M_1) \arrow[d, "(g'_{SP},g'_M)"] \\
(SP_2,M_2) \arrow[r, "(f'_{SP},f'_M)"'] & (SP_3,M_3)
\end{tikzcd}
\quad
\begin{tikzcd}
SP_0 \arrow[r, "f_{SP}"] \arrow[d, "g_{SP}"'] \arrow[dr, phantom, "(4)"] & SP_1 \arrow[d, "g'_{SP}"] \\
SP_2 \arrow[r, "f'_{SP}"'] & SP_3
\end{tikzcd}
\quad
\begin{tikzcd}
S_0 \arrow[r, "f_{SP,S}"] \arrow[d, "g_{SP,S}"'] \arrow[dr, phantom, "(5)"] & S_1 \arrow[d, "g'_{SP,S}"] \\
S_2 \arrow[r, "f'_{SP,S}"'] & S_3
\end{tikzcd}
\quad
\begin{tikzcd}
T_{SIG_0}(X_0) \arrow[r, "f^{\#}_{SP}"] \arrow[d, "g^{\#}_{SP}"'] \arrow[dr, phantom, "(6)"] & T_{SIG_1}(X_1) \arrow[d, "g'^{\#}_{SP}"] \\
T_{SIG_2}(X_2) \arrow[r, "f'^{\#}_{SP}"'] & T_{SIG_3}(X_3)
\end{tikzcd}
$$

□

Corollary 8 *The functor H : $\mathbf{Specs} \times \mathbf{Sets} \to \mathbf{Sets}$: $(SP, M) \mapsto (T_{SIG}(X) \times M)^{\oplus}, (f_{SP}, f_M) \mapsto (f^{\#}_{SP} \times f_M)^{\oplus}$ preserves pullbacks along $\mathcal{M}_1 \times \mathcal{M}_2$-morphisms.*

Proof. The product functor $\times$ preserves general pullbacks and, as shown in [EEPT06], the functor $\Box^{\oplus}$ preserves pullbacks along injective morphisms. Thus, it lasts to show that T : $\mathbf{Specs} \to \mathbf{Sets}$: $SP \mapsto T_{SIG}(X)$, where we forget the type information of the terms, preserves pullbacks.

In **Specs**, the pullback (4) is constructed componentwise on the sorts, operations and variables, which means that $S_0 = \{(s_1,s_2) \mid g'_{SP,S}(s_1) = f'_{SP,S}(s_2)\}$, $OP_0 = \{(op_1,op_2) : (s_1^1,s_2^1)...(s_1^n,s_2^n) \to (s_1,s_2) \mid g'_{SP,OP}(op_1 : s_1^1...s_1^n \to s_1) = f'_{SP,OP}(op_2 : s_2^1...s_2^n \to s_2)\}$ and $X_0 = \{(x_1,x_2) \mid g'_{SP,X}(x_1) = f'_{SP,X}(x_2)\}$. Therefore, the terms in $T_{SIG_0}(X_0)$ are defined by $T_{SIG_0,s}(X_0) = X_{0,s} \cup \{(c_1,c_2) \mid (c_1,c_2) :\to s \in OP_0\} \cup \{(op_1,op_2)(t_1,..,t_n) \mid (op_1,op_2) : s_1...s_n \to s \in OP_0, t_i \in T_{SIG_0,s_i}(X_0)\}$.

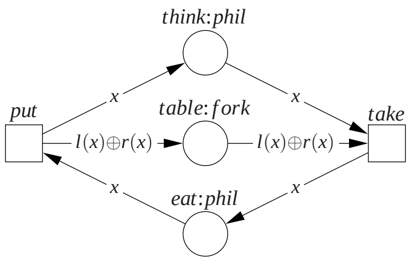
We have to show that $T_{SIG_0}(X_0)$ is isomorphic to the pullback object P over $f'^{\#}_{SP}$ and $g'^{\#}_{SP}$ with $P = \{(t_1,t_2) \mid g'^{\#}_{SP}(t_1) = f'^{\#}_{SP}(t_2)\}$. Since P is a pullback, with $f'^{\#}_{SP} \circ g^{\#}_{SP} = g'^{\#}_{SP} \circ f^{\#}_{SP}$ we get an induced morphism $i : T_{SIG_0}(X_0) \to P$ with $i(t) = (f^{\#}_{SP}(t), g^{\#}_{SP}(t))$, which means that i is inductively defined by $i(c_1,c_2) = (c_1,c_2)$ for constants, $i(x_1,x_2) = (x_1,x_2)$ for variables and $i((op_1,op_2)(t_1,...,t_n)) = (op_1(i(t_1)_1,...,i(t_n)_1), op_2(i(t_1)_2,...,i(t_n)_2))$ for complex terms.

f'_{SP}, g'_{SP} are specification morphisms and $f'^{\#}_{SP}, g'^{\#}_{SP}$ are inductively defined on terms. This means, for a pair $(t_1,t_2) \in P$, the terms t_1 and t_2 have to have the same structure. Define $j : P \to T_{SIG_0}(X_0)$ inductively by $j(c_1,c_2) = (c_1,c_2)$ for constants, $j(x_1,x_2) = (x_1,x_2)$ for variables and $j(op_1(t_1^1,...,t_1^n), op_2(t_2^1,...,t_2^n)) = (op_1,op_2)(j(t_1^1,t_2^1),...,j(t_1^n,t_2^n))$ for complex terms.

By induction, it can be shown that $i \circ j = id_P$ and $j \circ i = id_{T_{SIG_0}(X_0)}$. This means that i and j are isomorphisms and (6) is a pullback in **Sets**. □

Corollary 9 *The functor H : $\mathbf{Specs} \times \mathbf{Sets} \to \mathbf{Sets}$: $(SP, M) \mapsto \mathcal{P}_{fin}(Eqns(SIG,X)), (f_{SP}, f_M) \mapsto \mathcal{P}_{fin}(f^{\#}_{SP})$ preserves pullbacks along $\mathcal{M}_1 \times \mathcal{M}_2$-morphisms.*

Proof. In [EEPT06], it is shown that $\mathcal{P}$ preserves pullbacks along injective morphisms. Analogously, this can be shown for $\mathcal{P}_{fin}$, since if we start the construction for finite sets, this property is preserved. Thus, it lasts to show that $Eqns$ preserves pullbacks, which can be proven similar to the proof for sets of terms in Corollary 8 above. □



$$\begin{array}{ll} sorts: & phil, fork \\ opns: & p_1, \ldots, p_n :\to phil \\ & f_1, \ldots, f_n :\to fork \\ & l, r : phil \to fork \\ eqns: & l(p_i) = f_i \quad \forall i = 1, \ldots, n \\ & r(p_i) = f_{i+1} \quad \forall i = 1, \ldots, n-1 \\ & r(p_n) = f_1 \end{array}$$

Figure 2: The AHL net for *n* dining philosophers

As previously, we combine generalized AHL schemas and algebras to generalized AHL nets.

Definition 10 A *generalized AHL net* $GN = (GS, A)$ is given by a generalized AHL schema GN over the algebraic specification SP and an SP-algebra A.

A *generalized AHL net morphism* $f : GN \to GN'$ is a tuple $f = (f_{GS} : GS \to GS', f_A : A \to V_{f_{SP}}(A'))$, where f_{GS} is a generalized AHL schema morphism and f_A an algebra homomorphism. $V_{f_{SP}} : \mathbf{Algs}(\mathbf{SP}') \to \mathbf{Algs}(\mathbf{SP})$ is the forgetful functor induced by f_{SP}.

Generalized AHL nets and generalized AHL net morphisms form the category **AHLNets**.

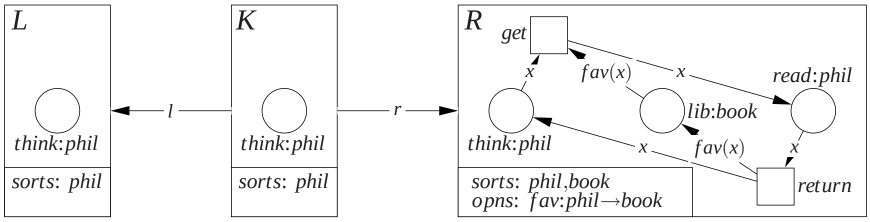
Corollary 10 *If the category* $(\mathbf{Algs}, \mathcal{M}_1)$ *of all algebras and generalized homomorphisms is a weak adhesive HLR category, then also the category* $(\mathbf{AHLNets}, \mathcal{M})$ *is a weak adhesive HLR category.* $\mathcal{M}$ *is the class of all injective AHL net morphisms* f *with* $f_A \in \mathcal{M}_1$.

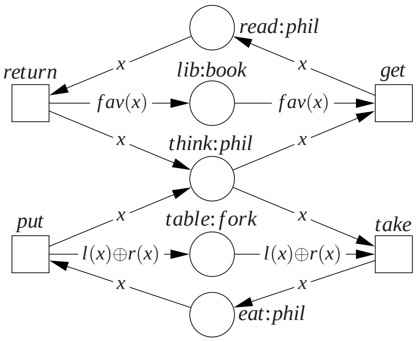
Proof Idea. The category **AHLNets** is isomorphic to the full subcategory $(\mathbf{AHLSchemas} \times \mathbf{Algs})|_{Ob'}$, where $Ob' = \{((SP, P, T, pre, post, cond, type), A) \mid A \in \mathbf{Algs}(\mathbf{SP})\}$. In this subcategory, the pushout and pullback objects over $\mathcal{M}$-morphisms are the same as in $\mathbf{AHLSchemas} \times \mathbf{Algs}$. According to Theorem 1.1 and Corollary 5 this implies that $(\mathbf{AHLNets}, \mathcal{M})$ is a weak adhesive HLR category. □

Up to now we do not know whether the category $(\mathbf{Algs}, \mathcal{M}_1)$ with the class $\mathcal{M}_1$ of injective morphisms is a weak adhesive HLR category. But if we restrict $\mathcal{M}_1$ to isomorphisms, $(\mathbf{Algs}, \mathcal{M}_1)$ is a weak adhesive HLR category and $\mathcal{M}_1$ is already a useful class for rules in net transformation systems. In many cases, one does not want to change the specification and algebra within the rule (where $\mathcal{M}_1$-morphisms are necessary). But for the match, general morphisms are allowed, thus we can apply such a rule to nets over different specifications and with different algebras. Another possibility is to restrict the algebra part to quotient term algebras leading to the category $\mathbf{Algs}|_{\mathbf{QTA}}$ with objects (SP, T_{SP}) and morphisms $f = (f_{SP}, f_T) : (SP, T_{SP}) \to (SP', T_{SP'})$ with $f_{SP} : SP \to SP'$ and $f_T : T_{SP} \to V_{f_{SP}}(T_{SP'})$ uniquely determined. $\mathbf{Algs}|_{\mathbf{QTA}}$ is isomorphic to the category of specifications and thus, together with strict injective morphisms, a weak adhesive HLR category.

In the following, we present an example based on the well-known Dining Philosophers Problem (see [BEE01]), where the behaviour of the philosophers is modeled by a net, while the philosophers themselves are modeled within the data structure.

Example 2 *For n philosophers, the AHL net with its specification is given in Figure 2. For the*





$$
\begin{array}{ll}
\textit{sorts}: & \textit{phil}, \textit{fork}, \textit{book} \\
\textit{opns}: & p_1, \ldots, p_n :\rightarrow \textit{phil} \\
 & f_1, \ldots, f_n :\rightarrow \textit{fork} \\
 & l, r : \textit{phil} \rightarrow \textit{fork} \\
 & \textit{fav} : \textit{phil} \rightarrow \textit{book} \\
\textit{eqns}: & l(p_i) = f_i \quad \forall i = 1, \ldots, n \\
 & r(p_i) = f_{i+1} \quad \forall i = 1, \ldots, n-1 \\
 & r(p_n) = f_1
\end{array}
$$

Figure 3: The production and the result of the direct transformation

data part, we use the quotient term algebra. Each philosopher p_i has a left fork f_i and a right fork f_{i+1}, except p_n with the right fork f_1, and needs these two forks to eat. In the AHL net, this condition is assured by the pre- and post-domain functions.

In the top of Figure 3 an examplary production is shown, where we extend the possible behaviour of the philosophers. We introduce a library, where a philosopher p_i may go to and get his favourite book $fav(p_i)$ to read. Due to our developed theory, this very simple rule can be applied to all kinds of nets, independent from the number of philosophers. In the bottom, the application of this rule to the AHL net in Figure 2 is shown, where the library has been introduced. Note that also the specification has changed, since the new sort book and the operation fav have been added. Now a thinking philosophers may go to the library by firing the new transition get.

6 Conclusion and Future Work

In this paper we have shown that all kinds of algebraic high-level schemas and nets are weak adhesive HLR categories. This means, that we can apply the theory for graph transformations developed in [EEPT06] also to different kinds of net transformations based on AHL schemas and nets.

At the moment, the available data structure underlying the AHL nets is restricted to a few, but still interesting cases. More work is needed in the area of algebras, where the categories **Algs**(**SP**) of algebras over a certain specification *SP* and **Algs** of generalized algebras and homomorphisms should be verified to be weak adhesive HLR categories, likely under some restrictions on the specification or $\mathscr{M}$-morphisms. The category **Algs** is equivalent to a Grothendieck category (see [TBG91]) indexed over the category **Specs**. Grothendieck categories have general pushouts and pullbacks, if so have the underlying categories, but they have not been shown to be weak adhesive HLR categories. A step towards this has been made in [EOP06], where also some restrictions to the morphism class $\mathcal{M}$ are discussed which could lead to a suitable weak adhesive HLR category.

Bibliography

[BEE01] R. Bardohl, C. Ermel, H. Ehrig. Generic Description of Syntax, Behavior and Animation of Visual Models. Technical report 19, TU Berlin, 2001.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs. Springer, 2006.

[EHKP91] H. Ehrig, A. Habel, H. Kreowski, F. Parisi-Presicce. Parallelism and Concurrency in High-Level Replacement Systems. *Mathematical Structures in Computer Science* 1(3):361–404, 1991.

[EHPP06] H. Ehrig, A. Habel, J. Padberg, U. Prange. Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation. *Fundamenta Informaticae* 74(1):1–29, 2006.

[Ehr79] H. Ehrig. Introduction to the Algebraic Theory of Graph Grammars (A Survey). In Claus et al. (eds.), *Graph Grammars and their Application to Computer Science and Biology*. LNCS 73, pp. 1–69. Springer, 1979.

[EOP06] H. Ehrig, F. Orejas, U. Prange. Categorical Foundations of Distributed Graph Transformation. In Corradini et al. (eds.), *Graph Transformations. Proceedings of ICGT 2006*. LNCS 4178, pp. 215–229. Springer, 2006.

[EP06] H. Ehrig, U. Prange. Weak Adhesive High-Level Replacement Categories and Systems: A Unifying Framework for Graph and Petri Net Transformations. In Futatsugi et al. (eds.), *Algebra, Meaning and Computation. Essays Dedicated to J.A. Goguen*. LNCS 4060, pp. 235–251. Springer, 2006.

[LS05] S. Lack, P. Sobociński. Adhesive and Quasiadhesive Categories. *Theoretical Informatics and Applications* 39(3):511–546, 2005.

[MM90] J. Meseguer, 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*. PhD thesis, TU Berlin, 1996.

[PER95] J. Padberg, H. Ehrig, L. Ribeiro. Algebraic High-Level Net Transformation Systems. *Mathematical Structures in Computer Science* 5(2):217–256, 1995.

[TBG91] A. Tarlecki, R. Burstall, J. Goguen. Some Fundamental Algebraic Tools for the Semantics of Computation: Part 3: Indexed Categories. *Theoretical Computer Science* 91(2):239–264, 1991.