

Parallelism and Concurrency in Adhesive High-Level Replacement Systems with Negative Application Conditions

Leen Lambers ^{a,1} Hartmut Ehrig ^{a,2} Ulrike Prange ^{a,3}
Fernando Orejas ^{b,4}

^a *Institute for Software Engineering and Theoretical Informatics
Technical University
Berlin, Germany*

^b *Department of Software
Technical University of Catalonia
Barcelona, Spain*

Abstract

The goal of this paper is the generalization of parallelism and concurrency results for adhesive High-Level Replacement (HLR) systems to adhesive HLR systems with negative application conditions. These conditions restrict the application of a rule by expressing that a specific structure should not be present before or after applying the rule to a certain context. Such a condition influences thus each rule application or transformation and therefore changes significantly the properties of the replacement system. The effect of negative application conditions on parallelism and concurrency in the replacement system is described in the generalization of the following results, formulated already for adhesive HLR systems without negative application conditions: Local Church-Rosser Theorem, Parallelism Theorem and Concurrency Theorem. These important generalized results will support the development of formal analysis techniques for adhesive HLR systems with negative application conditions.

Keywords: Negative Application Conditions, Adhesive High-Level Replacement Categories, Parallelism, Concurrency

1 Introduction

Adhesive High-Level Replacement (HLR) categories as introduced in [3] provide a formal method to describe transformation systems. The resulting framework is called adhesive HLR systems. These systems are based on rules that describe in an

¹ Email: leen@cs.tu-berlin.de

² Email: ehrig@cs.tu-berlin.de

³ Email: uprange@cs.tu-berlin.de

⁴ Email: orejas@lsi.upc.edu

abstract way how objects in adhesive HLR categories can be transformed. In [3], it is explained moreover how to define application conditions for rules that restrict the application of a rule. Most of the theoretical results in [3] though have been formulated for adhesive HLR systems based on rules without application conditions. These results should thus be generalized to adhesive HLR systems based on rules holding application conditions. The most frequently used kind of application condition is the so-called negative application condition (NAC) as introduced in [4] and used e.g. in [1,5,6,12,13]. It forbids a certain structure to be present before or after applying a rule. Therefore at first we concentrate on generalizing the theoretical results formulated for adhesive HLR systems based on rules without application conditions to adhesive HLR systems based on rules holding NACs. Shortly, we will speak about adhesive HLR systems with NACs.

Some important theoretical results for the particular case of graph transformation with NACs have been presented already in [10]. The overall goal is to come up with practical techniques for conflict detection and analysis [11,9] in transformation systems. In practice though most of these results are needed for the instantiation of typed attributed graph transformation systems with application conditions. This more general kind of graph transformation technique is most significant for modeling and metamodeling in software engineering and visual languages. Therefore the availability of all results for typed attributed graph transformation with NACs is an important motivation for the generalization to adhesive HLR systems with NACs. In [3], it has been already proven that such a typed attributed graph transformation system is a valid instantiation of adhesive HLR systems. Moreover results within adhesive HLR systems can be applied to all other instantiations of adhesive HLR systems such as e.g. hypergraph, algebraic signature or specification transformations with NACs. In this paper, we concentrate on the generalization of parallelism and concurrency results to adhesive HLR systems with NACs. In [8], though, also results for critical pairs, embedding, extension and local confluence are generalized to adhesive HLR systems with NACs.

The structure of this paper is as follows. In Section 2, it is explained how to augment the adhesive HLR framework with negative application conditions. Section 3 then formulates all results needed to generalize the notion of parallelism to transformations with NACs. At first the local Church-Rosser property is described for transformations with NACs. Therefore a new notion of parallel and sequential independence is defined between transformations with NACs. This is necessary because it can not only happen that a transformation deletes a structure that is used by the second transformation as considered in the case without NACs. Moreover we should consider the case of the first transformation producing a structure which is forbidden by the second one. Using the generalized notion of sequential independence for transformations with NACs it is now possible to formulate a Parallelism Theorem for transformations with NACs. This theorem expresses how to summarize a sequence of two sequentially independent transformations into one parallel transformation step with the same effect. If sequential dependencies occur between direct transformations in a transformation sequence the Parallelism Theorem can

not be applied in order to summarize this transformation sequence into one transformation step. In Section 4, therefore it is explained how to construct a so-called concurrent rule with NACs establishing the same effect in one transformation step with NACs as the whole transformation sequence. In the Concurrency Theorem it is proven that a concurrent rule holding NACs equivalent to a sequence of rules holding NACs is applicable with the same result if and only if the rule sequence with NACs is applicable. The construction of the concurrent rule itself is analog to the case without NACs. It is necessary though to translate all NACs occurring in the rule sequence into equivalent NACs on the concurrent rule. Therefore we will use results for application conditions already described in [3] and some new results.

2 Adhesive HLR systems with NACs

In this section, we introduce adhesive HLR systems with NACs. Nacs are an important feature for the modeling of transformation systems, expressing that a certain structure is not present when performing the transformation [4] and thus enhancing the expressiveness of the transformation. In order to provide a rich theory for such transformations with NACs, they are integrated into the framework of adhesive HLR systems. For this integration, mainly some new morphism classes have to be defined which are necessary for successful adaption of the theory.

First we repeat the definition for an adhesive HLR category as introduced in [3].

Definition 2.1 [adhesive HLR category] A category \mathbf{C} with a morphism class \mathcal{M} is called an *adhesive HLR category*, if

- (i) \mathcal{M} is a class of monomorphisms closed under isomorphisms, composition and decomposition,
- (ii) \mathbf{C} has pushouts (PO) and pullbacks (PB) along \mathcal{M} -morphisms and \mathcal{M} -morphisms are closed under pushouts and pullbacks,
- (iii) pushouts in \mathbf{C} along \mathcal{M} -morphisms are Van Kampen squares.

Remark 2.2 In [3], adhesive HLR systems are based on either adhesive HLR or weak adhesive HLR categories, which only slightly differ in the van Kampen square property. In this paper, we only consider adhesive HLR categories and enhance them with NACs. Note though that all results formulated in this paper will be applicable as well in weak adhesive HLR categories with NACs such as e.g. for Petri net transformations with NACs.

For an *adhesive HLR category with NACs* we need in addition to an adhesive HLR category without NACs some additional properties on the special morphism classes in the category in order to be able to generalize all results. We distinguish three classes of morphisms, namely \mathcal{M} , \mathcal{M}' and \mathcal{Q} , and a class of pairs of morphisms \mathcal{E}' . \mathcal{M} is a subset of the class of all monomorphisms as given in [3] and the rule morphisms are always in \mathcal{M} . The non-existing morphism q in Def. 2.6 for negative application conditions is an element of the morphism class \mathcal{Q} . For pair factorization

in Def. 5.25 in [3] we need moreover the classes \mathcal{M}' and \mathcal{E}' . $\mathcal{M}, \mathcal{E}', \mathcal{M}'$ and \mathcal{Q} should have the properties described in the following definition. Note that to each condition a remark is made in which theorem, lemma or definition this condition is needed for the first time.

Definition 2.3 [adhesive HLR category with NACs] An *adhesive HLR category with NACs* is an adhesive HLR category \mathbf{C} with special morphism class \mathcal{M} and in addition three morphism classes \mathcal{M}' , \mathcal{E}' and \mathcal{Q} with the following properties:

- unique $\mathcal{E}' - \mathcal{M}'$ pair factorization (see Def. 5.25 in [3])
needed for Completeness Theorem (see [8]), Definition 4.6, Embedding Theorem (see [8]),
- epi - \mathcal{M} factorization *needed for Lemma 4.2,*
- $\mathcal{M} - \mathcal{M}'$ PO-PB decomposition property (see Def. 5.27 in [3])
needed for Induced Direct Transformation Lemma (see [8]), Definition 4.6, Embedding Theorem (see [8]),
- $\mathcal{M} - \mathcal{Q}$ PO-PB decomposition property (see Def. 5.25 in [3])
needed for Lemma 2.11,
- initial PO over \mathcal{M}' - morphisms (see Def. 6.1 in [3])
needed for Extension Theorem (see [8]),
- \mathcal{M}' is closed under PO's and PB's along \mathcal{M} - morphisms
needed for Completeness Theorem (see [8]), Definition 4.6, Extension Theorem (see [8]),
- \mathcal{Q} is closed under PO's and PB's along \mathcal{M} - morphisms
needed for Lemma 2.11, Lemma 4.2,
- induced PB-PO property for \mathcal{M} and \mathcal{Q} (see Def. 2.4)
needed for Lemma 4.2,
- If $f : A \rightarrow B \in \mathcal{Q}$ and $g : B \rightarrow C \in \mathcal{M}'$ then $g \circ f \in \mathcal{Q}$.
Composition property for morphisms in \mathcal{M}' and \mathcal{Q} ,
needed for Induced Direct Transformation Lemma (see [8]),
- If $g \circ f \in \mathcal{Q}$ and $g \in \mathcal{M}'$ then $f \in \mathcal{Q}$.
Decomposition property for morphisms in \mathcal{M}' and \mathcal{Q} ,
needed for Completeness Theorem (see [8]),
- \mathcal{Q} is closed under composition and decomposition
needed for Lemma 4.2, Lemma 2.11.

Note that these properties hold in particular for the case of graph transformation systems with NACs with $\mathcal{Q} = \mathcal{M}' = \mathcal{M}$, where \mathcal{M} is the set of all graph monomorphisms and \mathcal{E}' the set of jointly surjective pairs of graph morphisms.

Definition 2.4 [induced PB-PO property for \mathcal{M} and \mathcal{Q}] Given $a : A \rightarrow C \in \mathcal{Q}$

and $b : B \rightarrow C \in \mathcal{M}$ and the following PB and PO

$$\begin{array}{ccc} D & \xrightarrow{d_2} & B \\ d_1 \downarrow & (PB) & \downarrow b \\ A & \xrightarrow{a} & C \end{array} \qquad \begin{array}{ccc} D & \xrightarrow{d_2} & B \\ d_1 \downarrow & (PO) & \downarrow e_1 \\ A & \xrightarrow{e_2} & E \end{array}$$

then the induced morphism $x : E \rightarrow C$ with $x \circ e_1 = b$ and $x \circ e_2 = a$ is a monomorphism in \mathcal{Q} .

Remark 2.5 Theorem 5.1 in [7] proves this property in adhesive categories for a, b being mono with the result that x is also mono.

A negative application condition or NAC as introduced in [4] forbids a certain structure to be present before or after applying the rule.

Definition 2.6 [negative application condition, rule with NACs]

- A *negative application condition* or $NAC(n)$ on L is an arbitrary morphism $n : L \rightarrow N$. A morphism $g : L \rightarrow G$ satisfies $NAC(n)$ on L , written $g \models NAC(n)$, if and only if $\nexists q : N \rightarrow G$ such that $q \circ n = g$.

$$\begin{array}{ccc} L & \xrightarrow{n} & N \\ g \downarrow & & \downarrow X_q \\ G & \xleftarrow{} & \end{array}$$

A set of NACs on L is denoted by $NAC_L = \{NAC(n_i) | i \in I\}$. A morphism $g : L \rightarrow G$ satisfies NAC_L if and only if g satisfies all single NACs on L i.e. $g \models NAC(n_i) \forall i \in I$.

- A set of NACs NAC_L (resp. NAC_R) on L (resp. R) for a rule $p : L \xleftarrow{l} K \xrightarrow{r} R$ (with $l, r \in \mathcal{M}$) is called *left* (resp. *right*) NAC on p . $NAC_p = (NAC_L, NAC_R)$, consisting of a set of left and a set of right NACs on p is called a *set of NACs on p* . A *rule (p, NAC_p) with NACs* is a rule with a set of NACs on p .

Definition 2.7 [adhesive HLR system with NACs]

- An *adhesive HLR system with NACs* $AHS = (\mathbf{C}, \mathcal{M}, \mathcal{M}', \mathcal{E}', \mathcal{Q}, P)$ consists of an adhesive HLR category with NACs $(\mathbf{C}, \mathcal{M}, \mathcal{M}', \mathcal{E}', \mathcal{Q})$ and a set of rules with NACs P .
- A *direct transformation* $G \xRightarrow{p, g} H$ via a rule $p : L \leftarrow K \rightarrow R$ with $NAC_p = (NAC_L, NAC_R)$ and a match $g : L \rightarrow G$ consists of the double pushout [2] (DPO)

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ g \downarrow & & \downarrow & & \downarrow h \\ G & \longleftarrow & D & \longrightarrow & H \end{array}$$

where g satisfies NAC_L , written $g \models NAC_L$ and $h : R \rightarrow H$ satisfies NAC_R , written $h \models NAC_R$. Since pushouts along \mathcal{M} -morphisms in an adhesive HLR

category always exist, the DPO can be constructed if the pushout complement of $K \rightarrow L \rightarrow G$ exists. If so, we say that the match g satisfies the gluing condition of rule p . A *transformation*, denoted as $G_0 \xRightarrow{*} G_n$, is a sequence $G_0 \Rightarrow G_1 \Rightarrow \dots \Rightarrow G_n$ of direct transformations.

Remark 2.8 From now on we consider only adhesive HLR systems with rules having an empty set of right negative application conditions. This is without loss of generality, because each right NAC can be translated into an equivalent left NAC as explained in [3], where Def. 7.16 and Theorem 7.17 can be specialized to NACs as shown in the following construction and lemma.

Definition 2.9 [construction of left from right NACs] For each $NAC(n_i)$ on R with $n_i : R \rightarrow N_i$ of a rule $p = (L \leftarrow K \rightarrow R)$, the equivalent left application condition $L_p(NAC(n_i))$ is defined in the following way:

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ n'_i \downarrow & (2) & \downarrow & (1) & \downarrow n_i \\ N'_i & \longleftarrow & Z & \longrightarrow & N_i \end{array}$$

- If the pair $(K \rightarrow R, R \rightarrow N_i)$ has a pushout complement, we construct $(K \rightarrow Z, Z \rightarrow N_i)$ as the pushout complement (1). Then we construct pushout (2) with the morphism $n'_i : L \rightarrow N'_i$. Now we define $L_p(NAC(n_i)) = NAC(n'_i)$.
- If the pair $(K \rightarrow R, R \rightarrow N_i)$ does not have a pushout complement, we define $L_p(NAC(n_i)) = \text{true}$.

For each set of NACs on R , $NAC_R = \cup_{i \in I} NAC(n_i)$ we define the following set of left NACs:

$$L_p(NAC_R) = \cup_{i \in I'} L_p(NAC(n'_i))$$

with $i \in I'$ if and only if the pair $(K \rightarrow R, R \rightarrow N_i)$ has a pushout complement.

Remark 2.10 Note that Z is unique since pushout complements along \mathcal{M} -morphisms are unique up to isomorphism in adhesive HLR categories.

Lemma 2.11 (equivalence of left and right NACs) For every rule p with NAC_R a set of right NACs on p , $L_p(NAC_R)$ as defined in Def. 2.9 is a set of left NACs on p such that for all direct transformations $G \xRightarrow{p,g} H$ with comatch h ,

$$g \models L_p(NAC_R) \Leftrightarrow h \models NAC_R$$

Proof The proof corresponds to case 1 and 3 in the proof of Theorem 7.17 in [3]. \square

Definition 2.12 [inverse rule with NACs] For a rule $p : L \leftarrow K \rightarrow R$ with $NAC_p = (NAC_L, \emptyset)$, the inverse rule is defined by $p^{-1} = R \leftarrow K \rightarrow L$ with $NAC_{p^{-1}} = (L_{p^{-1}}(NAC_L), \emptyset)$.

Theorem 2.13 (Inverse Direct Transformation with NACs) For each direct transformation with NACs $G \Rightarrow H$ via a rule $p : L \leftarrow K \rightarrow R$ with NAC_p



Figure 1. graph model

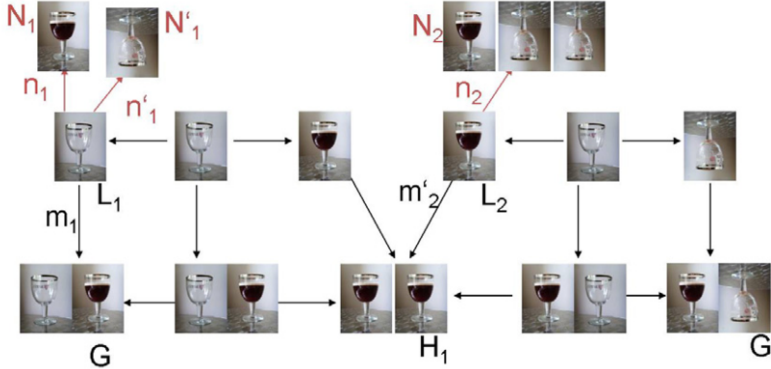


Figure 2. fill some glass and drink another one

a set of left NACs on p , there exists an inverse direct transformation with NACs $H \Rightarrow G$ via the inverse rule p^{-1} with $NAC_{p^{-1}}$.

Proof This follows directly from Def. 2.12 and Lemma 2.11. \square

Example 2.14 Consider as an example of an adhesive HLR system a graph transformation system consisting of two rules with NACs where $\mathcal{Q} = \mathcal{M} = \mathcal{M}'$ are injective typed graph morphisms. On the left hand side of Fig. 1 the type graph is shown and the following three pictures of a glass represent the different states that a glass can have in our system and how to represent them by a typed graph. An empty glass is represented by a single node of type G . If in addition an edge to a node of type F (resp. D) is present then the glass is full (resp. has been drunk or used already). The rules $p_1 = \text{fill}$ and $p_2 = \text{drink}$ of our example system are shown in the upper row of Fig. 2. They express respectively that a glass can be filled whenever it is empty and not used already and that a full glass can be drunk whenever two glasses have not been drunk already. In Fig. 2, you can see a two-step transformation in which the left glass is filled and the right one is being drunk. Note that the matching in our example is injective.

3 Parallelism in Adhesive HLR Systems with NACs

In order to generalize the notion of parallelism to adhesive HLR systems with NACs at first it is necessary to define when two direct transformations with NACs are parallel independent. For a pair of transformations with NACs it is not only possible that one transformation deletes a structure which is needed by the other one, but also that one transformation produces a structure which is forbidden by the other

one. For this new notion of parallel independence and thus also sequential independence it is possible to formulate the local Church-Rosser property with NACs and also a Parallelism Theorem with NACs as described in this section.

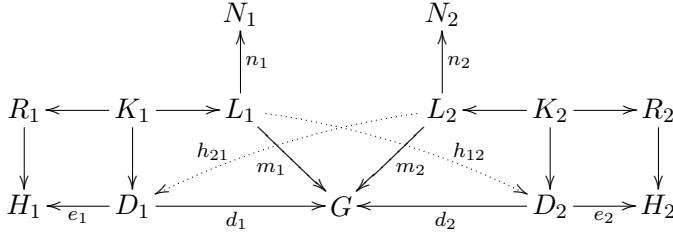
Definition 3.1 [parallel and sequential independence] Two direct transformations $G \xrightarrow{(p_1, m_1)} H_1$ with NAC_{p_1} and $G \xrightarrow{(p_2, m_2)} H_2$ with NAC_{p_2} are *parallel independent* if

$$\exists h_{12} : L_1 \rightarrow D_2 \text{ s.t. } (d_2 \circ h_{12} = m_1 \text{ and } e_2 \circ h_{12} \models NAC_{p_1})$$

and

$$\exists h_{21} : L_2 \rightarrow D_1 \text{ s.t. } (d_1 \circ h_{21} = m_2 \text{ and } e_1 \circ h_{21} \models NAC_{p_2})$$

as in the following diagram:



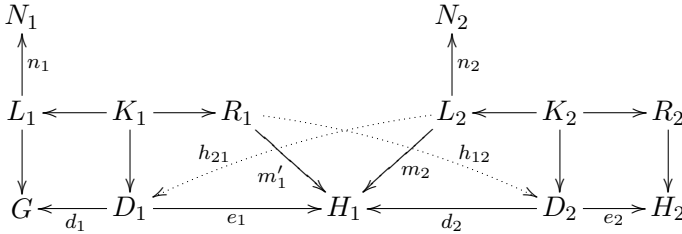
Two direct transformations $G \xrightarrow{(p_1, m_1)} H_1$ with NAC_{p_1} and $H_1 \xrightarrow{(p_2, m_2)} H_2$ with NAC_{p_2} are *sequentially independent* if

$$\exists h_{12} : R_1 \rightarrow D_2 \text{ s.t. } (d_2 \circ h_{12} = m'_1 \text{ and } e_2 \circ h_{12} \models NAC_{p_1^{-1}})$$

and

$$\exists h_{21} : L_2 \rightarrow D_1 \text{ s.t. } (e_1 \circ h_{21} = m_2 \text{ and } d_1 \circ h_{21} \models NAC_{p_2})$$

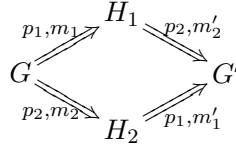
as in the following diagram:



Remark 3.2 Note that as for the case without NACs we have the following relationship between parallel and sequential independency: $G \xrightarrow{p_1} H_1 \xrightarrow{p_2} H_2$ are sequentially independent iff $G \xleftarrow{p_1^{-1}} H_1 \xrightarrow{p_2} H_2$ are parallel independent.

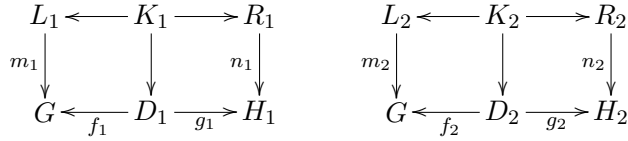
Theorem 3.3 (Local Church-Rosser Theorem with NACs) *Given an adhesive HLR system with NACs AHS and two parallel independent direct transformations with NACs $H_1 \xleftarrow{p_1, m_1} G \xrightarrow{p_2, m_2} H_2$, there are an object G' and direct transformations $H_1 \xrightarrow{p_2, m'_2} G'$ and $H_2 \xrightarrow{p_1, m'_1} G'$ such that $G \xrightarrow{p_1, m_1} H_1 \xrightarrow{p_2, m'_2} G'$ and*

$G \xRightarrow{p_2, m_2} H_2 \xRightarrow{p_1, m'_1} G'$ are sequentially independent. Vice versa, given two sequentially independent direct transformations with NACs $G \xRightarrow{p_1, m_1} H_1 \xRightarrow{p_2, m'_2} G'$ there are an object H_2 and sequentially independent direct transformations $G \xRightarrow{p_2, m_2} H_2 \xRightarrow{p_1, m'_1} G'$ such that $H_1 \xRightarrow{p_1, m_1} G \xRightarrow{p_2, m_2} H_2$ are parallel independent:

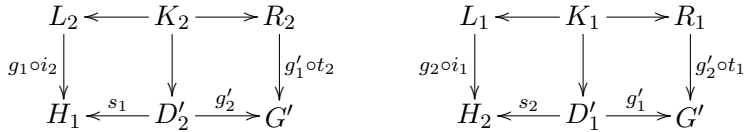


Proof

(i) Given the parallel independent transformations $H_1 \xRightarrow{p_1, m_1} G \xRightarrow{p_2, m_2} H_2$:



Because of Def. 3.1 and the parallel independence with NACs of $H_1 \xRightarrow{p_1, m_1} G \xRightarrow{p_2, m_2} H_2$ we know that there exists $i_2 : L_2 \rightarrow D_1$ (resp. $i_1 : L_1 \rightarrow D_2$) s.t. $f_1 \circ i_2 = m_2$ (resp. $f_2 \circ i_1 = m_1$) and moreover $g_1 \circ i_2 \models \text{NAC}_{p_2}$ (resp. $g_2 \circ i_1 \models \text{NAC}_{p_1}$). Because of the Local Church-Rosser Theorem for parallel independent transformations without NACs all necessary pushouts in $H_1 \xRightarrow{p_2} G'$ and $H_2 \xRightarrow{p_1} G'$ can be constructed s.t. $G \xRightarrow{p_1} H_1 \xRightarrow{p_2} G'$ and $G \xRightarrow{p_2} H_2 \xRightarrow{p_1} G'$ are sequentially independent according to Def. 5.9 in [3] for direct transformations without NACs. This means in particular that $t_1 : R_1 \rightarrow D'_2$ (resp. $t_2 : R_2 \rightarrow D'_1$) exist s.t. $s_1 \circ t_1 = n_1$ (resp. $s_2 \circ t_2 = n_2$) and the following pushout diagrams exist:



Since $g_1 \circ i_2 \models \text{NAC}_{p_2}$ and $g_2 \circ i_1 \models \text{NAC}_{p_1}$, $H_1 \xRightarrow{p_2} G'$ and $H_2 \xRightarrow{p_1} G'$ are valid direct transformations with NACs. For the sequential independence of $G \xRightarrow{p_1} H_1 \xRightarrow{p_2} G'$ we have to show that i_2, t_1 are the required morphisms. For i_2 we have $f_1 \circ i_2 = m_2$, and therefore $f_1 \circ i_2 \models \text{NAC}_{p_2}$ follows by assumption. Now we investigate $g'_2 \circ t_1$. Because of Theorem 2.13 and the fact that $g_2 \circ i_1 \models \text{NAC}_{p_1}$ it follows directly that also $g'_2 \circ t_1 \models \text{NAC}_{p_1^{-1}}$. Analogously the sequential independence of $G \xRightarrow{p_2} H_2 \xRightarrow{p_1} G'$ can be proven.

(ii) Given sequentially independent direct transformations with NACs $G \xRightarrow{p_1, m_1} H_1 \xRightarrow{p_2, m'_2} G'$ with comatches n'_1 and n'_2 , respectively, from Remark 3.2 we obtain

parallel independent direct transformations with NACs $G \xleftarrow{p_1^{-1}, n_1} H_1 \xRightarrow{p_2, m'_2} G'$. Now part (i) of the proof gives us sequentially independent direct transformations with NACs $H_1 \xRightarrow{p_1^{-1}, n_1} G \xRightarrow{p_2, m_2} H_2$ and $H_1 \xRightarrow{p_2, m'_2} G' \xRightarrow{p_1^{-1}, n'_1} H_2$. Applying again Remark 3.2 to the first transformation we obtain parallel independent direct transformations with NACs $H_1 \xleftarrow{p_1, m_1} G \xRightarrow{p_2, m_2} H_2$:

$$\begin{array}{ccc} & G & \\ \swarrow p_1^{-1}, n_1 & & \searrow p_2, m_2 \\ H_1 & & H_2 \\ \searrow p_2, m'_2 & & \swarrow p_1^{-1}, n'_1 \\ & G' & \end{array} \quad \square$$

Now we can generalize the notion of parallelism to adhesive HLR systems with NACs.

Remark 3.4 In order to build parallel rules we need as explained also in [3] the following property. Let for Definition 3.5, Theorem 3.6, Definition 3.7 and Theorem 3.8 $AHS = (\mathbf{C}, \mathcal{M}, \mathcal{M}', \mathcal{E}', \mathcal{Q}, P)$ be an adhesive HLR system with NACs, where $(\mathbf{C}, \mathcal{M})$ has binary coproducts compatible with \mathcal{M} (see Def. 5.14 in [3]).

Definition 3.5 [parallel rule and transformation with NAC] Given two rules $p_1 = (L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1)$ with NAC_{p_1} and $p_2 = (L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2)$ with NAC_{p_2} , the *parallel rule* $p_1 + p_2$ with $NAC_{p_1+p_2}$ is defined by the coproduct constructions over the corresponding objects and morphisms: $p_1 + p_2 = (L_1 + L_2 \xleftarrow{l_1+l_2} K_1 + K_2 \xrightarrow{r_1+r_2} R_1 + R_2)$ and $NAC_{p_1+p_2} = \{n_1 + id_{L_2} | n_1 \in NAC_{p_1}\} \cup \{id_{L_1} + n_2 | n_2 \in NAC_{p_2}\}$. A direct transformation $G \Rightarrow G'$ via $p_1 + p_2$ with $NAC_{p_1+p_2}$ and a match $m : L_1 + L_2 \rightarrow G$ satisfying $NAC_{p_1+p_2}$ is a *direct parallel transformation with NAC* or *parallel transformation with NAC* for short.

The following Theorem describes that two sequentially independent transformations with NACs can be synthesized to a parallel transformation with NACs. Please note that in order to apply this theorem an extra composition property should hold in the adhesive HLR category with NACs.

Theorem 3.6 (Parallelism Theorem with NACs : Synthesis) *Let the composition of a coproduct morphism with a morphism in \mathcal{Q} be again in \mathcal{Q} . Then given a sequentially independent direct transformation sequence with NACs $G \Rightarrow H_1 \Rightarrow G'$ via p_1, m_1 (resp. p_2, m'_2) with NAC_{p_1} (resp. NAC_{p_2}), there is a construction leading to a parallel transformation with NACs $G \Rightarrow G'$ via $[m_1, m_2]$ and the parallel rule $p_1 + p_2$ with $NAC_{p_1+p_2}$, called a synthesis construction.*

$$\begin{array}{ccc} & G & \\ \swarrow p_1, m_1 & & \searrow p_2, m_2 \\ H_1 & & H_2 \\ \searrow p_2, m'_2 & & \swarrow p_1, m'_1 \\ & G' & \end{array} \quad \begin{array}{c} \Downarrow [m_1, m_2] \\ \Downarrow p_1+p_2, m_1, m_2 \end{array}$$

Proof Given the sequentially independent direct transformations with NACs $G \xRightarrow{p_1, m_1} H_1 \xRightarrow{p_2, m'_2} G'$, using the Parallelism Theorem (Theorem 5.18 in [3]) without

NACs we can construct the following double pushout:

$$\begin{array}{ccccc} L_1 + L_2 & \longleftarrow & K_1 + K_2 & \longrightarrow & R_1 + R_2 \\ \downarrow [m_1, m_2] & & \downarrow & & \downarrow \\ G & \longleftarrow & D & \longrightarrow & G' \end{array}$$

Now we have to prove that $[m_1, m_2] \models NAC_{p_1+p_2}$ such that according to Def. 3.5 this double pushout becomes a direct parallel transformation with NACs. Suppose that $[m_1, m_2] \not\models NAC_{p_1+p_2} = \{n_1 + id_{L_2} | n_1 \in NAC_{p_1}\} \cup \{id_{L_1} + n_2 | n_2 \in NAC_{p_2}\}$. Then we have two cases.

- Suppose that $q : L_1 + N_2 \rightarrow G$ exists such that $q \circ id_{L_1} + n_2 = [m_1, m_2]$ for some $n_2 \in NAC_{p_2}$. Consider the following diagram in which $\eta_2 : L_2 \rightarrow L_1 + L_2$ and $\eta'_2 : N_2 \rightarrow L_1 + N_2$ are coproduct morphisms.

$$\begin{array}{ccccccc} L_2 & \xrightarrow{n_2} & N_2 & \xrightarrow{\eta'_2} & L_1 + N_2 & \xrightarrow{q} & G \\ & \searrow \eta_2 & & \nearrow id_{L_1} + n_2 & & \nearrow [m_1, m_2] & \\ & & L_1 + L_2 & & & & \end{array}$$

Then by $q \circ id_{L_1} + n_2 = [m_1, m_2]$ and the coproduct properties of $L_1 + N_2$ and $L_1 + L_2$ the following commutativity holds in this diagram: $m_2 = [m_1, m_2] \circ \eta_2 = q \circ id_{L_1} + n_2 \circ \eta_2 = q \circ \eta'_2 \circ n_2$. Because of the composition property on coproduct morphisms and \mathcal{Q} morphisms $q \circ \eta'_2 \in \mathcal{Q}$ and thus $m_2 \not\models NAC(n_2)$. This is a contradiction and thus $[m_1, m_2] \models NAC(id_{L_1} + n_2)$.

- Suppose that $q : N_1 + L_2 \rightarrow G$ exists such that $q \circ n_1 + id_{L_2} = [m_1, m_2]$. Then we can prove analogously that $m_1 \not\models NAC(n_1)$ and thus by contradiction that $[m_1, m_2] \models NAC(n_1 + id_{L_2})$.

□

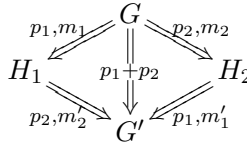
In order to be able to sequentialize a parallel transformation with NACs $G \Rightarrow G'$ via a match m and a parallel rule $p_1 + p_2$ with $NAC_{p_1+p_2}$ it is necessary to call for an extra condition on the parallel transformation called NAC-compatibility. This condition expresses that the NACs on rule p_1 and p_2 are satisfied by the matches m_1, m'_1, m_2, m'_2 occurring in the direct transformations when sequentializing the direct parallel transformation without NACs. It is necessary to ask for satisfiability of the single NACs by these matches, since in general this does not follow from the satisfaction of $NAC_{p_1+p_2}$ by m .

$$\begin{array}{ccccc} & & G & & \\ p_1, m_1 \swarrow & & \downarrow & & \searrow p_2, m_2 \\ H_1 & \xrightarrow{p_1+p_2, m} & & & H_2 \\ p_2, m'_2 \swarrow & & \downarrow & & \searrow p_1, m'_1 \\ & & G' & & \end{array}$$

Definition 3.7 [NAC-compatible Parallel Transformation] Given a parallel transformation with NACs $G \Rightarrow G'$ via match $m : L_1 + L_2 \rightarrow G$ and the parallel rule $p_1 + p_2$ with $NAC_{p_1+p_2}$. Let $m_1 : L_1 \rightarrow G$, $m_2 : L_2 \rightarrow G$ be the matches of the direct transformations $G \Rightarrow H_1$ and $G \Rightarrow H_2$ via p_1 resp. p_2 and m'_1 and m'_2 the matches of the direct transformations $H_2 \Rightarrow G'$ and $H_1 \Rightarrow G'$ via p_1 resp. p_2 as constructed in the Parallelism Theorem without NACs (Analysis part in Theorem 5.18 in [3]). The parallel transformation with NACs $G \Rightarrow G'$ is *NAC-compatible* if $m_1, m'_1 \models NAC_{p_1}$ and $m_2, m'_2 \models NAC_{p_2}$.

Theorem 3.8 (Parallelism Theorem with NACs: Analysis) • *Given a NAC-compatible direct parallel transformation with NACs $G \Rightarrow G'$ via $m : L_1 + L_2 \rightarrow G$ and the parallel rule $p_1 + p_2$ with $NAC_{p_1+p_2}$, then there is a construction leading to two sequentially independent transformation sequences with NACs $G \Rightarrow H_1 \Rightarrow G'$ via p_1, m_1 and p_2, m'_2 and $G \Rightarrow H_2 \Rightarrow G'$ via p_2, m_2 and p_1, m'_1 , called an analysis construction.*

- **Bijective Correspondence.** *The synthesis construction of Theorem 3.6 and the analysis construction are inverse to each other up to isomorphism.*



Proof

- Given a parallel transformation with NACs $G \xrightarrow{p_1+p_2, m} G'$ then because of the Parallelism Theorem without NACs (Theorem 5.18 in [3]) it follows that $G \Rightarrow H_1$ and $G \Rightarrow H_2$ are parallel independent without NACs and moreover the necessary double pushouts for $G \Rightarrow H_1 \Rightarrow G'$ via p_1, m_1 and p_2, m'_2 and $G \Rightarrow H_2 \Rightarrow G'$ via p_2, m_2 and p_1, m'_1 can be constructed s.t. they are sequentially independent without NACs. Moreover we know because of NAC-compatibility of $G \xrightarrow{p_1+p_2, m} G'$ that $m_1, m'_1 \models NAC_{p_1}$ and $m_2, m'_2 \models NAC_{p_2}$. Therefore $G \Rightarrow H_1$ and $G \Rightarrow H_2$ are parallel independent as transformations with NACs as defined in Def. 3.1. From Theorem 3.3 it follows that $G \Rightarrow H_1 \Rightarrow G'$ and $G \Rightarrow H_2 \Rightarrow G'$ are then sequentially independent with NACs.
- Because of the uniqueness of pushouts and pushout complements, the constructions are inverse to each other up to isomorphism.

□

Example 3.9 We continue with the example adhesive HLR system as described in Example 2.14. In Fig. 2, we had a two-step transformation in which the left glass is filled and the right one is being drunk. Since the direct transformations in this sequence are sequentially independent, in Fig. 3 the parallel transformation with NACs for this two-step transformation according to Theorem 3.6 and Def. 3.5 is shown. On the contrary, in Fig. 4, you can see a parallel transformation with rules $p_1 = \text{drink}$ and $p_2 = \text{drink}$ which is not NAC-compatible. In order to

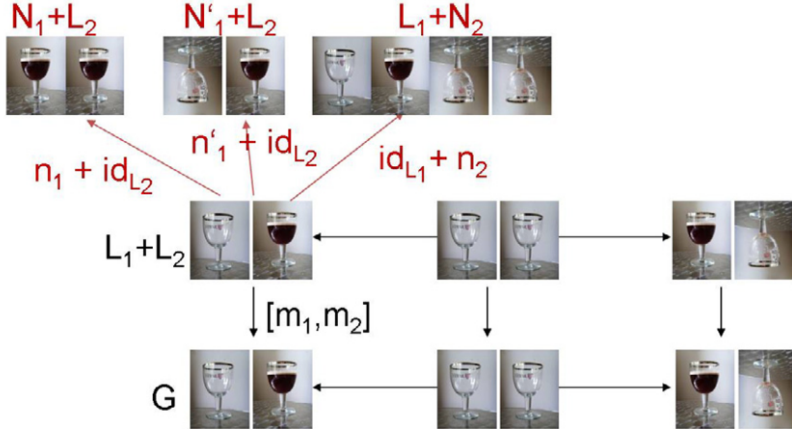


Figure 3. fill some glass and drink another one in parallel

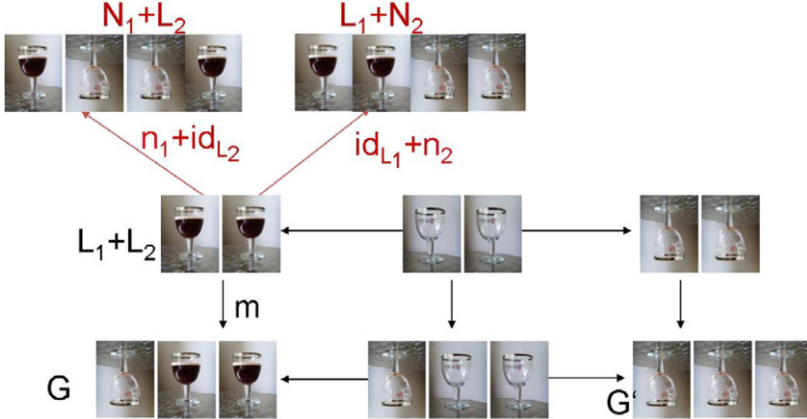


Figure 4. drink two glasses, if one is already drunk: parallel transformation which is not NAC-compatible

sequentialize a parallel transformation into two direct transformations with NACs it has to be NAC-compatible (see Theorem 3.8). In Fig. 5 the parallel transformation in Fig. 4 without NACs is sequentialized. Afterwards it is checked if the NACs $NAC(N_1)$ on $p_1 = drink$ (resp. $NAC(N_2)$ on $p_2 = drink$) are satisfied. It becomes clear that NAC-compatibility as defined in Def. 3.7 is violated because there exists $q'_2 : N_2 \rightarrow H_1$ with $q'_2 \circ n_2 = m'_2$ (and analogously $q'_1 : N_1 \rightarrow H_2$ with $q'_1 \circ n_1 = m'_1$). Intuitively speaking, the second transformation in Fig. 5 describes that another glass is drunk although two glasses have been drunk already and this is exactly what the NAC on the drink rule forbids.

4 Concurrency in adhesive HLR Systems with NACs

Let t be a transformation via the rules p_0, \dots, p_{n-1} with NACs and matches g_0, \dots, g_{n-1} . In general there will be dependencies between several direct transformations in this transformation sequence. Therefore it is not possible to apply the Parallelism Theorem in order to summarize the transformation sequence into

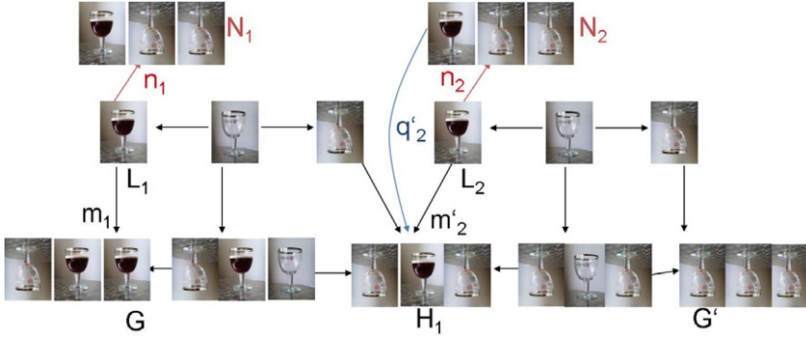
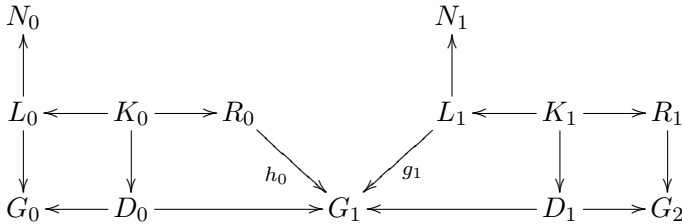


Figure 5. drink a glass, if one is already drunk and afterwards drink another one

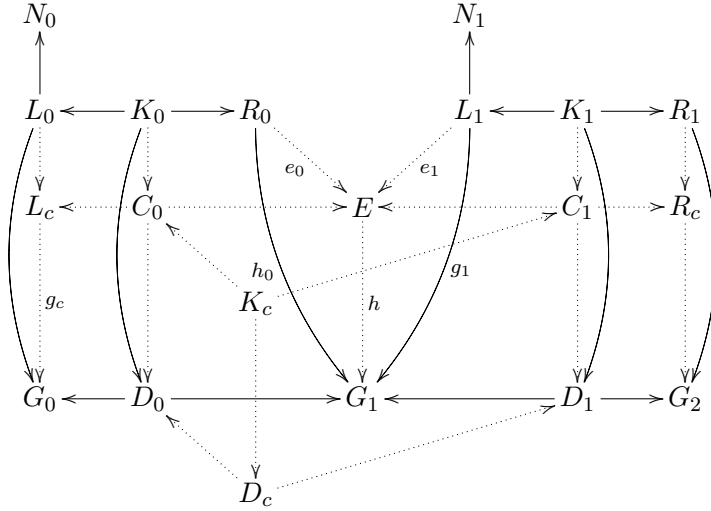
one equivalent transformation step. It is possible though to formulate a Concurrency Theorem which expresses how to translate such a sequence into one equivalent transformation step anyway. Therefore we build on the notion of a concurrent rule of a transformation sequence without NACs as introduced in [3]. Moreover we have to translate all the NACs occurring in the rule sequence p_0, \dots, p_{n-1} backward into an equivalent set of NACs on the concurrent rule p_c of this rule sequence. This means, a set NAC_{p_c} should be found such that this set of NACs is equivalent to $NAC_{p_0}, \dots, NAC_{p_{n-1}}$ for the transformation t . This section describes gradually how to obtain this concurrent NAC and generalizes then the Concurrency Theorem to transformations with NACs.

Let us consider at first a two-step transformation with NACs:



The goal is to translate all NACs on p_0 and p_1 into an equivalent set of NACs on the concurrent rule $p_c : L_c \leftarrow K_c \rightarrow R_c$ inducing as explained in Theorem 5.23 in [3] a concurrent transformation $G_0 \Rightarrow G_2$ via p_c and match g_c as shown in the following

diagram:



Consequently the two necessary steps are:

- Translate each set of NACs on L_0 into an equivalent set of NACs on L_c .
- Translate each set of NACs on L_1 into an equivalent set of NACs on L_c .

We can prove the first step as described in the following construction and Lemma.

Definition 4.1 [construction of NACs on L_c from NACs on L_0] Consider the following diagram:

$$\begin{array}{ccc}
 N_j & \xrightarrow{e_i} & N'_i \\
 n_j \uparrow & (1) & n'_i \uparrow \\
 L_0 & \xrightarrow{m_0} & L_c
 \end{array}$$

For each $NAC(n_j)$ on L_0 with $n_j : L_0 \rightarrow N_j$ and $m_0 : L_0 \rightarrow L_c$, let

$$D_{m_0}(NAC(n_j)) = \{NAC(n'_i) | i \in I, n'_i : L_c \rightarrow N'_i\}$$

where I and n'_i are constructed as follows: $i \in I$ if and only if (e_i, n'_i) with $e_i : N_j \rightarrow N'_i$ jointly epimorphic, $e_i \circ n_j = n'_i \circ m_0$ and $e_i \in \mathcal{Q}$.

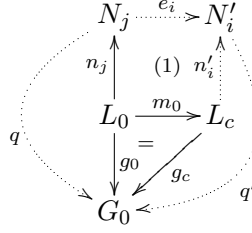
For each set of NACs $NAC_{L_0} = \{NAC(n_j) | j \in J\}$ on L_0 the downward translation of NAC_{L_0} is then defined as:

$$D_{m_0}(NAC_{L_0}) = \cup_{j \in J} D_{m_0}(NAC(n_j))$$

Lemma 4.2 (equivalence of set of NACs on L_0 and set of NACs on L_c)

Given $g_c : L_c \rightarrow G_0$, $m_0 : L_0 \rightarrow L_c$ with NAC_{L_0} and $g_0 = g_c \circ m_0$ as in the

following diagram:

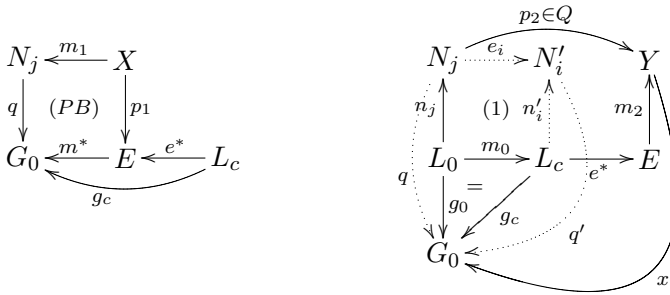


then the following holds :

$$g_0 \models NAC_{L_0} \Leftrightarrow g_c \models D_{m_0}(NAC_{L_0})$$

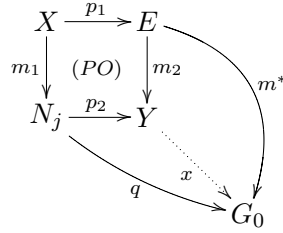
Proof

- (\Rightarrow) Let $g_c \not\models D_{m_0}(NAC(L_0)) = \cup_{j \in J} D_{m_0}(NAC(n_j))$ with $NAC_{L_0} = \{NAC(n_j) | j \in J\}$. Then for some $j \in J$ there is a NAC $n'_i : L_c \rightarrow N'_i \in D_{m_0}(NAC(n_j))$ and $e_i : N_j \rightarrow N'_i$ for which holds that $g_c \not\models NAC(n'_i)$, (e_i, n'_i) jointly epi, $e_i \in \mathcal{Q}$ and $e_i \circ n_j = n'_i \circ m_0$. Consequently there exists a morphism $q' : N'_i \rightarrow G_0 \in \mathcal{Q}$ such that $q' \circ n'_i = g_c$. Since $g_0 = g_c \circ m_0 = q' \circ n'_i \circ m_0 = q' \circ e_i \circ n_j$ there exists a morphism $q : N_j \rightarrow G_0$ defined by $q = q' \circ e_i$ s.t. $q \circ n_j = q' \circ e_i \circ n_j = g_0$. Because of the composition property for morphisms in \mathcal{Q} we have $q \in \mathcal{Q}$ since $q' \in \mathcal{Q}$ and e_i in \mathcal{Q} . Hence $g_0 \models NAC(n_j) \Rightarrow g_0 \models NAC_{L_0}$.
- (\Leftarrow) Let $g_0 \not\models NAC_{L_0}$ with $NAC_{L_0} = \{NAC(n_j) | j \in J\}$. Then for some $j \in J$ a morphism $q : N_j \rightarrow G_0 \in \mathcal{Q}$ exists such that $q \circ n_j = g_0$. Let (e^*, m^*) be an epi- \mathcal{M} -factorization of g_c . Construct X with $p_1 : X \rightarrow E$ and $m_1 : X \rightarrow N_j$ as pullback of m^* and q .



Then we have $m_1 \in \mathcal{M}$ and $p_1 \in \mathcal{Q}$, since $m^* \in \mathcal{M}$, $q \in \mathcal{Q}$, PBs preserve \mathcal{M} and PBs along \mathcal{M} preserve \mathcal{Q} . Now construct Y with $m_2 : E \rightarrow Y$ and $p_2 : N_j \rightarrow Y$ as pushout of m_1 and p_1 . Then we have $m_2 \in \mathcal{M}$, $p_2 \in \mathcal{Q}$, since $m_1 \in \mathcal{M}$, $p_1 \in \mathcal{Q}$, POs preserve \mathcal{M} and POs along \mathcal{M} preserve \mathcal{Q} . Because of the induced PB-PO property the induced morphism $x : Y \rightarrow G_0$ with $x \circ m_2 = m^*$ and $x \circ p_2 = q$ is

a monomorphism in \mathcal{Q} .

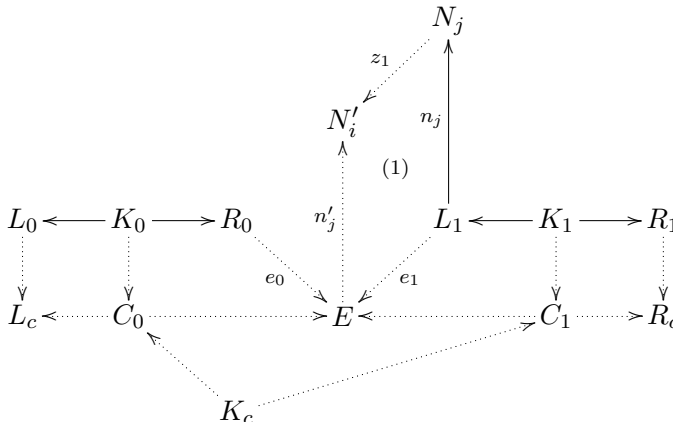


It holds moreover that $p_2, m_2 \circ e^*$ jointly epimorphic because e^* epimorphic and p_2, m_2 jointly epimorphic. Summarizing we have the following equations: $x \circ m_2 \circ e^* \circ m_0 = m^* \circ e^* \circ m_0 = g_c \circ m_0 = g_0 = q \circ n_j = x \circ p_2 \circ n_j$ and since x mono we have $m_2 \circ e^* \circ m_0 = p_2 \circ n_j$. Since $m_2 \circ e^*$ and p_2 are jointly epimorphic, $p_2 \circ n_j = (m_2 \circ e^*) \circ m_0$ and $p_2 \in \mathcal{Q}$ we can conclude that $m_2 \circ e^* : L_c \rightarrow Y$ equals one of the morphisms $n'_i : L_c \rightarrow N'_i \in D_{m_0}(NAC(n_j))$. Moreover since $x \circ m_2 \circ e^* = m^* \circ e^* = g_c$ and $x \in \mathcal{Q}$ it holds that $g_c \notin NAC(m_2 \circ e^*) = NAC(n'_i)$ and consequently $g_c \notin NAC(n'_i) \Rightarrow g_c \notin D_{m_0}(NAC(n_j)) \Rightarrow g_c \notin D_{m_0}(NAC_{L_0})$. \square

Remark 4.3 It is possible to cancel the fact that \mathcal{Q} is a class of special morphisms, by defining that \mathcal{Q} is the class of all morphisms in the category, and thus generalize the definition of NAC-satisfiability. We should assume in this case though either that the NAC-morphism is in \mathcal{M} or each match is in \mathcal{M} . (1) would be constructed then as a pushout instead of as a set of jointly epimorphic pairs of morphisms. This result is formulated and proven explicitly in [8].

In Def. 2.9 and Lemma 2.11 it is explained how to construct an equivalent set of left NACs from a set of right NACs on a rule. Now we are ready to define a set of equivalent NACs on the left hand side of the concurrent rule of a two-step transformation from the set of NACs on the LHS of the second rule of this transformation.

Definition 4.4 [construction of NACs on L_c from NACs on L_1] Given an E-dependency relation $(e_0, e_1) \in \mathcal{E}'$ for the rules p_0 and p_1 and $p_c = p_0 *_E p_1 : L_c \leftarrow K_c \rightarrow R_c$ the E-concurrent rule of p_0 and p_1 as depicted in the following diagram:



For each $NAC(n_j)$ on L_1 with $n_j : L_1 \rightarrow N_j$:

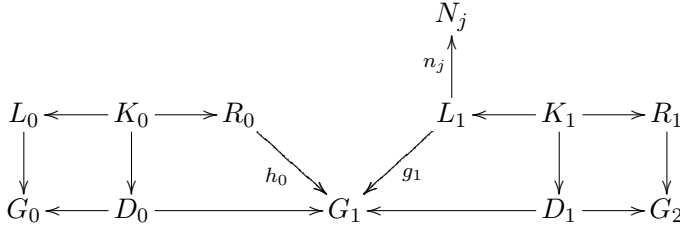
$$DL_{p_c}(NAC(n_j)) = L_p(D_{e_1}(NAC(n_j)))$$

with $p : L_c \leftarrow C_0 \rightarrow E$ and D_{e_1}, L_p according to Def. 4.1 and Def. 2.9.

For each set of NACs $NAC_{L_1} = \{NAC(n_j) | j \in J\}$ on L_1 the down- and leftward translation of NAC_{L_1} is defined as:

$$DL_{p_c}(NAC_{L_1}) = \cup_{j \in J} DL_{p_c}(NAC(n_j))$$

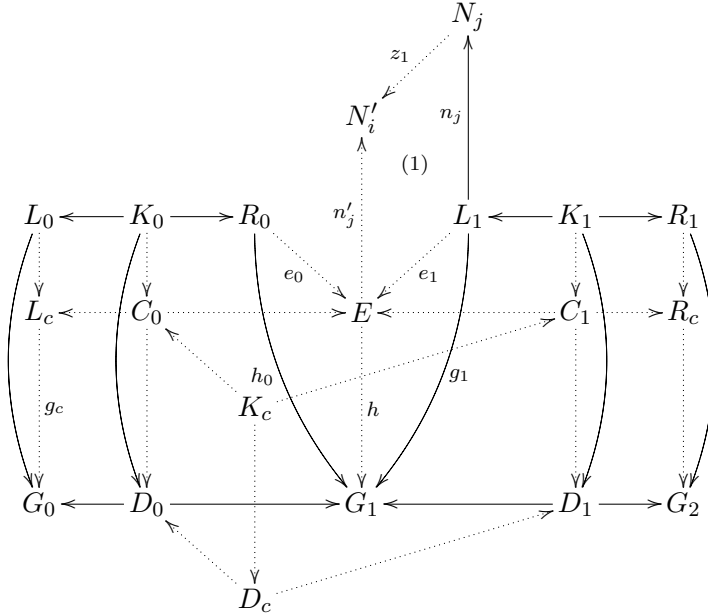
Lemma 4.5 (equivalence of NACs on rule p_1 and NACs on p_c) *Given a two-step E -related transformation via $p_0 : L_0 \leftarrow K_0 \rightarrow R_0$ and $p_1 : L_1 \leftarrow K_1 \rightarrow R_1$*



with g_c being the match from the LHS of the E -concurrent rule $p_c = p_0 *_E p_1$ into G_0 (as described in the synthesis construction of Theorem 5.23 in [3]) then the following holds:

$$g_1 \models NAC_{L_1} \Leftrightarrow g_c \models DL_{p_c}(NAC_{L_1}).$$

Proof Consider the following diagram:



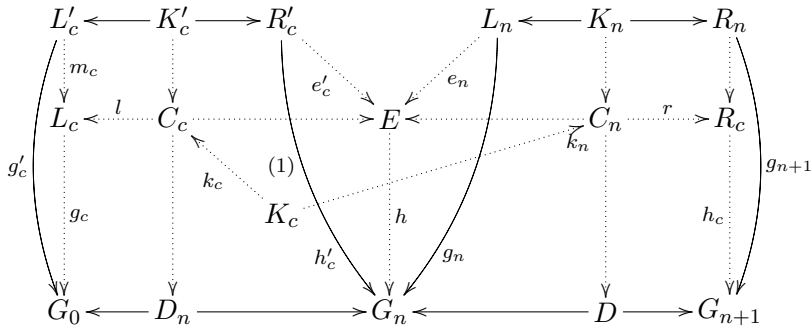
The E -concurrent rule $p_0 *_E p_1$ is the rule $p_c : L_c \leftarrow K_c \rightarrow R_c$, as described in Def. 5.21 in [3]. The derived span of the E -concurrent transformation $G_0 \xrightarrow{p_c, g_c} G_2$

is $G_0 \leftarrow D_c \rightarrow G_2$. Because of Lemma 4.2 $g_1 \models NAC_{L_1} \Leftrightarrow h \models D_{e_1}(NAC_{L_1})$. Moreover because of Lemma 2.9 $g_c \models L_p(D_{e_1}(NAC_{L_1})) \Leftrightarrow h \models D_{e_1}(NAC_{L_1})$ with $p : L_c \leftarrow C_0 \rightarrow E$. Note that $L_p(D_{e_1}(NAC_{L_1})) = \cup_{j \in J} L_p(D_{e_1}(NAC(n_j))) = \cup_{j \in J} DL_{p_c}(NAC(n_j))$. Consequently, it holds that $g_1 \models NAC_{L_1} \Leftrightarrow g_c \models DL_{p_c}(NAC_{L_1})$. \square

Definition 4.6 [concurrent rule with NAC, concurrent (co-)match induced by $G_0 \xRightarrow{n+1} G_{n+1}$]

$n = 0$ For a direct transformation $G_0 \Rightarrow G_1$ via match $g_0 : L_0 \rightarrow G_0$, comatch $g_1 : R_1 \rightarrow G_1$ and rule $p_0 : L_0 \leftarrow K_0 \rightarrow R_0$ with NAC_{p_0} the *concurrent rule* p_c with NAC induced by $G_0 \Rightarrow G_1$ is defined by $p_c = p_0$ with $NAC_{p_c} = NAC_{p_0}$, the *concurrent comatch* h_c is defined by $h_c = g_1$ and the *concurrent match* g_c by $g_c = g_0 : L_0 \rightarrow G_0$.

$n \geq 1$ Consider $p'_c : L'_c \leftarrow K'_c \rightarrow R'_c$ (resp. $g'_c : L'_c \rightarrow G_0$, $h'_c : R'_c \rightarrow G_n$), the concurrent rule with NACs (resp. concurrent match, comatch) induced by $G_0 \xRightarrow{n} G_n$. Let $((e'_c, e_n), h)$ be the $\mathcal{E}' - \mathcal{M}'$ pair factorization of the comatch h'_c and match g_n of $G_n \Rightarrow G_{n+1}$. According to Fact 5.29 in [3] PO-PB decomposition, PO composition and decomposition lead to the diagram below in which (1) is a pullback and all other squares are pushouts:



For a transformation sequence $G_0 \xRightarrow{n+1} G_{n+1}$ the *concurrent rule* p_c with NAC s (resp. concurrent match, comatch) induced by $G_0 \xRightarrow{n+1} G_{n+1}$ is defined by $p_c = L_c \xleftarrow{lok_c} K_c \xrightarrow{rok_n} R_c$ ($g_c : L_c \rightarrow G_0$, $h_c : R_c \rightarrow G_{n+1}$). Thereby NAC_{p_c} is defined by $NAC_{p_c} = DL_{p_c}(NAC_{L_n}) \cup D_{m_c}(NAC_{L'_c})$.

Theorem 4.7 (Concurrency Theorem with NACs) (i) *Synthesis.* Given a transformation sequence $t : G_0 \xRightarrow{*} G_{n+1}$ via a sequence of rules p_0, p_1, \dots, p_n , then there is a synthesis construction leading to a direct transformation $G_0 \Rightarrow G_{n+1}$ via the concurrent rule $p_c : L_c \leftarrow K_c \rightarrow R_c$ with NAC_{p_c} , match $g_c : L_c \rightarrow G_0$ and comatch $h_c : R_c \rightarrow G_{n+1}$ induced by $t : G_0 \xRightarrow{*} G_{n+1}$.

(ii) *Analysis.* Given a direct transformation $G'_0 \Rightarrow G'_{n+1}$ via the concurrent rule $p_c : L_c \leftarrow K_c \rightarrow R_c$ with NAC_{p_c} induced by $t : G_0 \xRightarrow{*} G_{n+1}$ via a sequence of rules p_0, p_1, \dots, p_n then there is an analysis construction leading to a transformation sequence $t' : G'_0 \xRightarrow{*} G'_{n+1}$ with NAC s via p_0, p_1, \dots, p_n .

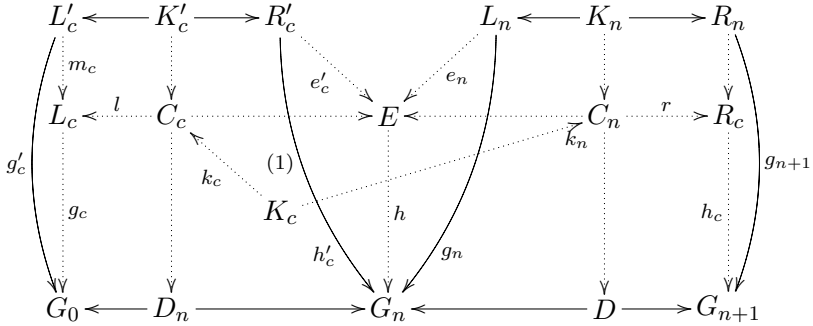
- (iii) Bijective Correspondence. *The synthesis and analysis constructions are inverse to each other up to isomorphism.*

Proof We prove this theorem by induction over the number of transformation steps $n + 1$.

(i) *Synthesis.*

Basis. $n=0$. For a direct transformation $t : G_0 \xrightarrow{p_0, g_0} G_1$ via match $g_0 : L_0 \rightarrow G_0$ and rule $p_0 : L_0 \leftarrow K_0 \rightarrow R_0$ with NAC_{p_0} the *concurrent rule* p_c with NAC induced by $G_0 \Rightarrow G_1$ is defined by $p_c = p_0$ with $NAC_{p_c} = NAC_{p_0}$ and the *concurrent match* g_c is defined by $g_c = g_0 : L_0 \rightarrow G_0$. Therefore the synthesis construction is equal to $G_0 \xrightarrow{p_c, g_c} G_1$.

Induction Step. Consider $t : G_0 \xRightarrow{n} G_n \Rightarrow G_{n+1}$ via the rules p_0, p_1, \dots, p_n . Let $p'_c : L'_c \leftarrow K'_c \rightarrow R'_c$ (resp. $g'_c : L'_c \rightarrow G_0$, $h'_c : R'_c \rightarrow G_n$), be the concurrent rule with NACs (resp. concurrent match, comatch) induced by $G_0 \xRightarrow{n} G_n$. Suppose that $G_0 \xrightarrow{p'_c, g'_c} G_n$ is a direct transformation with NAC leading to G_n . Let $((e'_c, e_n), h)$ be the $\mathcal{E}' - \mathcal{M}'$ pair factorization of the comatch h'_c and match g_n of $G_n \Rightarrow G_{n+1}$. PO-PB decomposition, PO composition and decomposition as described in Fact 5.29 in [3] lead to the diagram below in which (1) is a pullback and all other squares are pushouts:



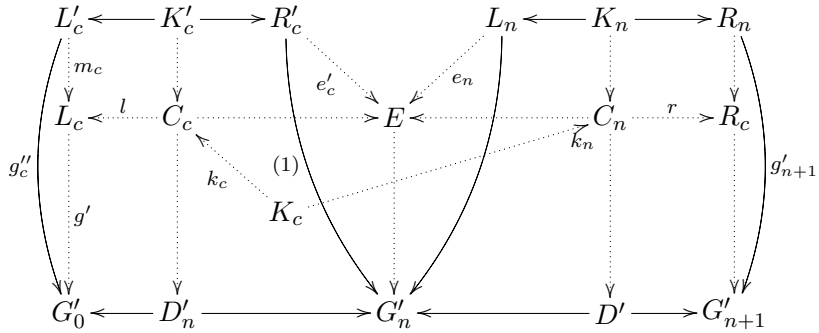
The *concurrent rule* p_c with NAC s (resp. concurrent match, comatch) induced by $G_0 \xRightarrow{n+1} G_{n+1}$ is $p_c = L_c \xleftarrow{lok_c} K_c \xrightarrow{rok_n} R_c$ ($g_c : L_c \rightarrow G_0$, $h_c : R_c \rightarrow G_{n+1}$). Thereby NAC_{p_c} is $NAC_{p_c} = DL_{p_c}(NAC_{L_n}) \cup D_{m_c}(NAC_{L'_c})$. We should prove that $G_0 \xrightarrow{p_c, g_c} G_{n+1}$ is a valid direct transformation with NACs. At first an analog synthesis construction to the one for two direct transformations without NACs in Theorem 5.23 in [3] can be done. Thus, in a second step we shall show that g_c satisfies NAC_{p_c} if g'_c satisfies $NAC_{p'_c}$ and g_n satisfies NAC_{p_n} . This follows because of Lemma 4.2, Lemma 4.5 and the fact that $G_0 \xrightarrow{p'_c, g'_c} G_n$ is a direct transformation via the rule p'_c with concurrent NAC $NAC_{p'_c}$.

(ii) *Analysis.*

Basis. $n=0$. For a direct transformation $G'_0 \Rightarrow G'_1$ via the concurrent rule $p_c = p_0$ with $NAC_{p_c} = NAC_{p_0}$ the analysis construction is equal to $G'_0 \Rightarrow$

G'_1 .

Induction Step. Given a direct transformation $G'_0 \Rightarrow G'_{n+1}$ via the concurrent rule $p_c : L_c \leftarrow K_c \rightarrow R_c$ with NAC_{p_c} induced by $t : G_0 \xRightarrow{*} G_{n+1}$ via a sequence of rules p_0, p_1, \dots, p_n . The concurrent rule p_c induced by t can be interpreted as $p'_c *_{E_n} p_n$ in which the E_n -dependency relation between the rules is induced by the $\mathcal{E}' - \mathcal{M}'$ pair factorization of the comatch h'_c induced by $G_0 \xRightarrow{*} G_n$ and the match g_n of $G_n \rightarrow G_{n+1}$ as described in Def. 4.6. So we have a direct transformation $G'_0 \Rightarrow G'_{n+1}$ via $p_c = p'_c *_{E_n} p_n$ and because of the Analysis part of Theorem 5.23 in [3] there is an analysis construction leading to a transformation sequence without NACs $G'_0 \Rightarrow G'_n \Rightarrow G'_{n+1}$ via p'_c and p_n and matches g'_c resp. g'_n .



We know by assumption that the match g' of $G'_0 \Rightarrow G'_{n+1}$ satisfies NAC_{p_c} . Since Lemma 4.2 and Lemma 4.5 hold in both directions, i.e. translate NACs in an equivalent way, we can conclude that $NAC_{p'_c}$ and NAC_{p_n} are satisfied by g'_c resp. g'_n . Therefore $G'_0 \Rightarrow G'_n \Rightarrow G'_{n+1}$ is a valid transformation sequence with NACs. Because of the induction hypothesis there exists an analysis construction $G'_0 \Rightarrow G'_1 \Rightarrow \dots \Rightarrow G'_n$ via p_0, p_1, \dots, p_{n-1} for $G'_0 \Rightarrow G'_n$ via p'_c . Thus we obtain a transformation sequence with NACs $G'_0 \Rightarrow G'_1 \Rightarrow \dots \Rightarrow G'_{n+1}$ via p_0, p_1, \dots, p_n for the direct transformation $G'_0 \Rightarrow G'_{n+1}$ via the concurrent rule $p_c : L_c \leftarrow K_c \rightarrow R_c$ with NAC_{p_c} .

- (iii) *Bijective Correspondence.* The bijective correspondence follows from the fact that the $\mathcal{E}' - \mathcal{M}'$ pair factorization is unique, and pushout and pullback constructions are unique up to isomorphism.

□

Example 4.8 We continue with Example 2.14. In Fig. 6 you can see a transformation in which a glass is filled and the same glass is being drunk. Since these direct transformations are sequentially dependent because they use the same glass, it is only possible to build the concurrent transformation according to Def. 4.6 and Theorem 4.7 which is shown in Fig. 10. The following three steps explain this synthesis construction of the two-step transformation. In Fig. 7 the construction of the concurrent rule according to Def. 4.6 induced by this two-step transformation without NACs is shown. In Fig. 8 the construction is shown of the equivalent NACs on L_c from $NAC(n_1)$ and $NAC(n_2)$ on L_0 . In Fig. 9 the construction is shown of

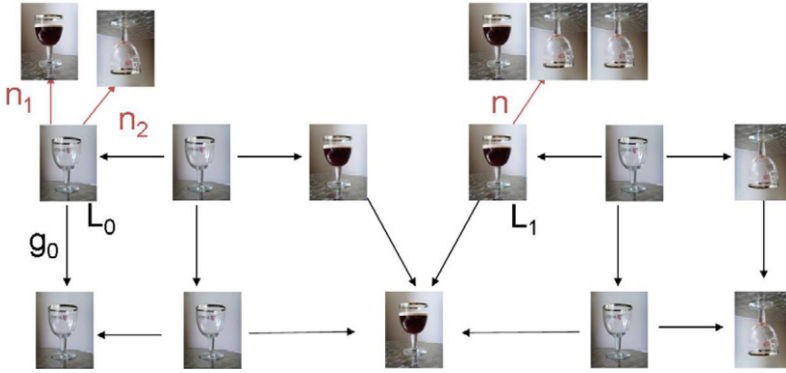


Figure 6. fill some glass and drink the same one

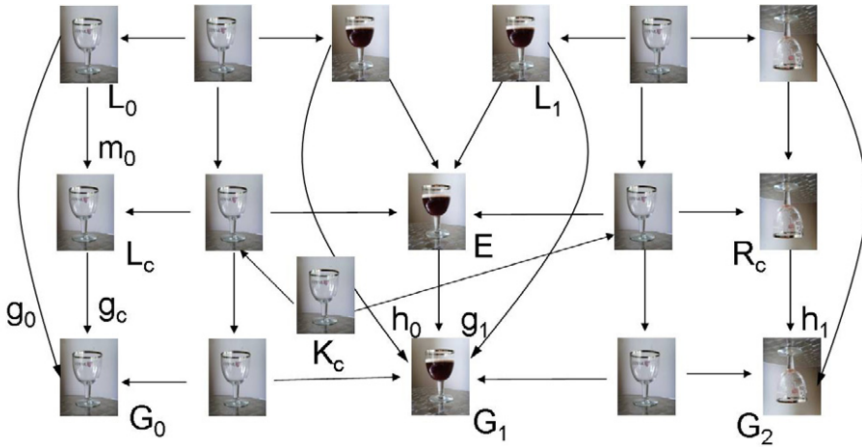
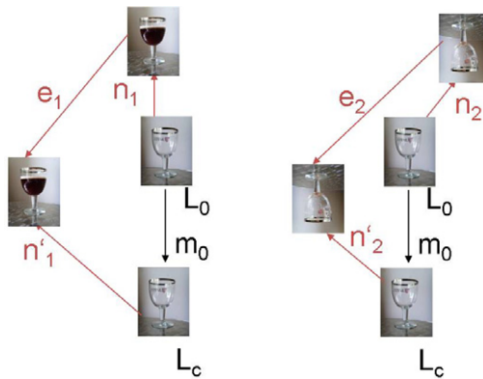


Figure 7. concurrent rule induced by two-step transformation

Figure 8. equivalent NAC on L_c to $NAC(n_1)$ and $NAC(n_2)$ on L_0

the equivalent NAC on L_c from $NAC(n)$ on L_1 according to Lemma 4.5.

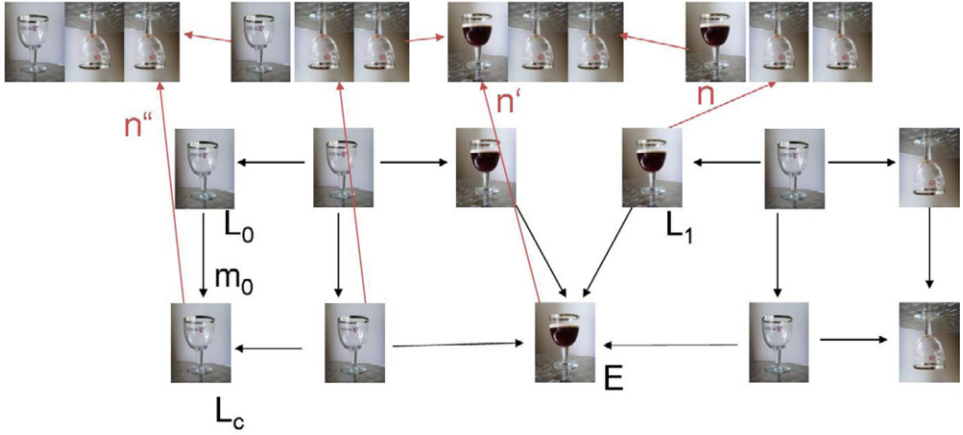
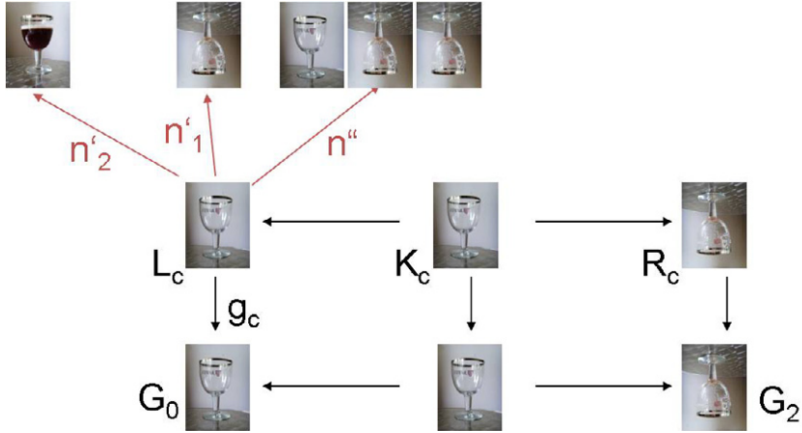
Figure 9. equivalent NAC on L_c to $NAC(n)$ on L_1 

Figure 10. fill some glass and drink the same one concurrently

5 Conclusion

In this paper results for parallelism and concurrency are described in order to extend these notions on Algebraic Graph Transformation in [3] to Transformations with Negative Application Conditions. Summarizing we have proven the Local-Church-Rosser Theorem, Parallelism Theorem and Concurrency Theorem for transformations with NACs. These results are formulated in the context of the Adhesive High-Level Replacement Framework introduced in [3] with an extra necessary morphism class \mathcal{Q} . This makes these results on parallelism and concurrency applicable to all transformation systems with NACs fitting into the adhesive HLR framework. Moreover these results serve as a basis for the extension of other notions like Embedding and Local Confluence for adhesive HLR systems with NACs described technically already in [8]. Future work will be necessary on the applicability and refinement of all new results and on the development of efficient analysis algorithms for transformations with NACs.

References

- [1] Bottoni, P., A. Schürr and G. Taentzer, *Efficient Parsing of Visual Languages based on Critical Pair Analysis and Contextual Layered Graph Transformation*, in: *Proc. IEEE Symposium on Visual Languages*, 2000, long version available as technical report SI-2000-06, University of Rom.
- [2] Corradini, A., U. Montanari, F. Rossi, H. Ehrig, R. Heckel and M. Löwe, *Algebraic Approaches to Graph Transformation I: Basic Concepts and Double Pushout Approach*, in: G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific, 1997 pp. 163–245.
- [3] Ehrig, H., K. Ehrig, U. Prange and G. Taentzer, “Fundamentals of Algebraic Graph Transformation,” EATCS Monographs in Theoretical Computer Science, Springer, 2006.
URL <http://www.springer.com/3-540-31187-4>
- [4] Habel, A., R. Heckel and G. Taentzer, *Graph grammars with negative application conditions* **26** (1996), pp. 287–313.
- [5] Hausmann, J., R. Heckel and G. Taentzer, *Detection of Conflicting Functional Requirements in a Use Case-Driven Approach*, in: *Proc. of Int. Conference on Software Engineering 2002*, Orlando, USA, 2002, to appear.
- [6] Koch, M., L. Mancini and F. Parisi-Presicce, *Graph-based Specification of Access Control Policies*, in: *JCSS 71*, 2005, pp. 1–33.
- [7] Lack, S. and P. Sobocinski, *Adhesive and quasiadhesive categories*, *Theoretical Informatics and Applications* **39(2)** (2005), pp. 522–546.
- [8] Lambers, L., *Adhesive high-level replacement systems with negative application conditions*, Technical report, Technische Universität Berlin (2007).
- [9] Lambers, L., H. Ehrig and F. Orejas, *Efficient detection of conflicts in graph-based model transformation*, in: *Proc. International Workshop on Graph and Model Transformation (GraMoT’05)*, *Electronic Notes in Theoretical Computer Science* (2005).
URL <http://tfs.cs.tu-berlin.de/publikationen/Papers05/LE005.pdf>
- [10] Lambers, L., H. Ehrig and F. Orejas, *Conflict Detection for Graph Transformation with Negative Application Conditions*, in: *Proc. Third International Conference on Graph Transformation (ICGT’06)*, LNCS **4178** (2006), pp. 61–76.
URL <http://tfs.cs.tu-berlin.de/publikationen/Papers06/LE006a.pdf>
- [11] Lambers, L., H. Ehrig and F. Orejas, *Efficient Conflict Detection in Graph Transformation Systems by Essential Critical Pairs*, in: *Proc. International Workshop on Graph Transformation and Visual Modeling Techniques (GTVMT’06)*, ENTCS (2006).
URL <http://tfs.cs.tu-berlin.de/publikationen/Papers06/LE006.pdf>
- [12] Mens, T., G. Taentzer and O. Runge, *Detecting Structural Refactoring Conflicts using Critical Pair Analysis*, in: R. Heckel and T. Mens, editors, *Proc. Workshop on Software Evolution through Transformations: Model-based vs. Implementation-level Solutions (SETra’04)*, Satellite Event of ICGT’04 (2004).
URL <http://www.cs.tu-berlin.de/~gabi/gMTR04.pdf>
- [13] Taentzer, G., K. Ehrig, E. Guerra, J. de Lara, L. Lengyel, T. Levendovsky, U. Prange, D. Varro and S. Varro-Gyapay, *Model Transformation by Graph Transformation: A Comparative Study*, in: *Proc. Workshop Model Transformation in Practice*, Montego Bay, Jamaica, 2005.
URL <http://tfs.cs.tu-berlin.de/publikationen/Papers05/TEG+05.pdf>