

Electronic Notes in Theoretical Computer Science 203 (2008) 43–66

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

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

Leen Lambers [a](#_bookmark0)*,*[1](#_bookmark2) Hartmut Ehrig [a](#_bookmark0)*,*[2](#_bookmark2) Ulrike Prange [a](#_bookmark0)*,*[3](#_bookmark2)

Fernando Orejas[b](#_bookmark1)*,*[4](#_bookmark2)

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 condi- tions 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

# Introduction

Adhesive High-Level Replacement (HLR) categories as introduced in [[3](#_bookmark32)] 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](mailto:leen@cs.tu-berlin.de)

2 Email: [ehrig@cs.tu-berlin.de](mailto:ehrig@cs.tu-berlin.de)

3 Email: [uprange@cs.tu-berlin.de](mailto:uprange@cs.tu-berlin.de)

4 Email: [orejas@lsi.upc.edu](mailto:orejas@lsi.upc.edu)

1571-0661© 2008 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2008.10.042

abstract way how objects in adhesive HLR categories can be transformed. In [[3](#_bookmark32)], 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](#_bookmark32)] 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 condi- tion is the so-called negative application condition (NAC) as introduced in [[4](#_bookmark35)] and used e.g. in [[1](#_bookmark33),[5](#_bookmark36),[6](#_bookmark37),[12](#_bookmark43),[13](#_bookmark44)]. 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 transforma- tion with NACs have been presented already in [[10](#_bookmark41)]. The overall goal is to come up with practical techniques for conflict detection and analysis [[11](#_bookmark42),[9](#_bookmark40)] 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 model- ing 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](#_bookmark32)], 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 transforma- tions with NACs. In this paper, we concentrate on the generalization of parallelism and concurrency results to adhesive HLR systems with NACs. In [[8](#_bookmark39)], 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. Sec- tion 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 be- cause 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 independ- ence for transformations with NACs it is now possible to formulate a Parallelism Theorem for transformations with NACs. This theorem expresses how to summar- ize 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 trans- formation 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](#_bookmark32)] and some new results.

# Adhesive HLR systems with NACs

In this section, we introduce adhesive HLR systems with NACs. Nacs are an import- ant feature for the modeling of transformation systems, expressing that a certain structure is not present when performing the transformation [[4](#_bookmark35)] 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](#_bookmark32)].

**Definition 2.1** [adhesive HLR category] A category **C** with a morphism class *M*

is called an *adhesive HLR category*, if

* 1. *M* is a class of monomorphisms closed under isomorphisms, composition and decomposition,
  2. **C** has pushouts (PO) and pullbacks (PB) along *M*-morphisms and *M*- morphisms are closed under pushouts and pullbacks,
  3. pushouts in **C** along *M*-morphisms are Van Kampen squares.

**Remark 2.2** In [[3](#_bookmark32)], 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 *M, M'* and *Q*, and a class of pairs of morphisms *E'*. *M* is a subset of the class of all monomorphisms as given in [[3](#_bookmark32)] and the rule morphisms are always in *M*. The non-existing morphism *q* in Def. [2.6](#_bookmark4) for negative application conditions is an element of the morphism class *Q*. For pair factorization

in Def. 5.25 in [[3](#_bookmark32)] we need moreover the classes *M'* and *E'*. *M, E', M'* and *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 **C** with special morphism class *M* and in addition three morphism classes *M', E'* and *Q* with the following properties:

* unique *E' − M'* pair factorization (see Def. 5.25 in [[3](#_bookmark32)])

*needed for Completeness Theorem (see [*[*8*](#_bookmark39)*]), Deﬁnition* [*4.6*](#_bookmark25)*, Embedding Theorem (see [*[*8*](#_bookmark39)*])*,

* epi - *M* factorization *needed for Lemma* [*4.2*](#_bookmark23),
* *M− M'* PO-PB decomposition property (see Def. 5.27 in [[3](#_bookmark32)])

*needed for Induced Direct Transformation Lemma (see [*[*8*](#_bookmark39)*]), Deﬁnition* [*4.6*](#_bookmark25)*, Em- bedding Theorem (see [*[*8*](#_bookmark39)*])*,

* *M−Q* PO-PB decomposition property (see Def. 5.25 in [[3](#_bookmark32)])

*needed for Lemma* [*2.11*](#_bookmark6),

* initial PO over *M'* - morphisms (see Def. 6.1 in [[3](#_bookmark32)])

*needed for Extension Theorem (see [*[*8*](#_bookmark39)*])*,

* *M'* is closed under PO’s and PB’s along *M* - morphims

*needed for Completeness Theorem (see [*[*8*](#_bookmark39)*]), Deﬁnition* [*4.6*](#_bookmark25)*, Extension Theorem (see [*[*8*](#_bookmark39)*])*,

* *Q* is closed under PO’s and PB’s along *M* - morphisms

*needed for Lemma* [*2.11*](#_bookmark6)*, Lemma* [*4.2*](#_bookmark23),

* induced PB-PO property for *M* and *Q* (see Def. [2.4](#_bookmark3))

*needed for Lemma* [*4.2*](#_bookmark23),

* If *f* : *A → B ∈Q* and *g* : *B → C ∈ M'* then *g ◦ f ∈ Q*. Composition property for morphisms in *M'* and *Q*,

*needed for Induced Direct Transformation Lemma (see [*[*8*](#_bookmark39)*])*,

* If *g ◦ f ∈Q* and *g ∈ M'* then *f ∈ Q*.

Decomposition property for morphisms in *M'* and *Q*,

*needed for Completeness Theorem (see [*[*8*](#_bookmark39)*])*,

* *Q* is closed under composition and decomposition

*needed for Lemma* [*4.2*](#_bookmark23)*, Lemma* [*2.11*](#_bookmark6).

Note that these properties hold in particular for the case of graph transform- ation systems with NACs with *Q* = *M'* = *M*, where *M* is the set of all graph monomorphisms and *E'* the set of jointly surjective pairs of graph morphisms.

**Definition 2.4** [induced PB-PO property for *M* and *Q*] Given *a* : *A → C ∈ Q*

and *b* : *B → C ∈M* and the following PB and PO

*D*  *d*2 *B D*  *d*2 *B*

*d*1 (*PB*) *b d*1 (*PO*) *e*1

J *a* JJ *e*2 J

*A C A E*

then the induced morphism *x* : *E → C* with *x ◦ e*1 = *b* and *x ◦ e*2 = *a* is a monomorphism in *Q*.

**Remark 2.5** Theorem 5.1 in [[7](#_bookmark38)] 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](#_bookmark35)] 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 *N AC*(*n*) *on L* is an arbitrary morphism *n* : *L → N* . A morphism *g* : *L → G satisﬁes N AC*(*n*) on *L*, written *g |*= *N AC*(*n*), if and only if */ ∃ q* : *N → G ∈Q* such that *q ◦ n* = *g*.

*L*  *n*  *N g*

J *Xq*

¸*,*

*G*

A set of NACs on *L* is denoted by *NACL* = *{N AC*(*ni*)*|i ∈ I}*. A morphism *g* : *L → G satisﬁes N ACL* if and only if *g* satisfies all single NACs on *L* i.e. *g |*= *N AC*(*ni*) *∀i ∈ I*.

* + A set of NACs *N ACL* (resp. *N ACR*) on *L* (resp. *R*) for a rule *p* : *L ←l K →r R*

(with *l, r ∈ M*) is called *left* (resp. *right* ) NAC on *p*. *NACp* = (*N ACL,NACR*), 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, N ACp*) *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* = (**C***, M, M', E', Q,P* ) consists of an adhesive HLR category with NACs (**C***, M, M', E', Q*) and a set of rules with NACs *P* .
  + A *direct transformation G p,g H* via a rule *p* : *L ← K → R* with *N AC* =

*⇒*

*p*

(*N ACL,NACR*) and a match *g* : *L → G* consists of the double pushout [[2](#_bookmark34)] (DPO)

*L* ¸*, K*  *R*

*g*

J¸ *,*

*h*

J J

*G D H*

where *g* satisfies *N ACL*, written *g |*= *N ACL* and *h* : *R → H* satisfies *N ACR*, written *h |*= *N ACR*. Since pushouts along *M*-morphisms in an adhesive HLR

category always exist, the DPO can be constructed if the pushout complement of

*K → L → G* exists. If so, we say that the match *g* satisfies the gluing condition

of rule *p*. A *transformation*, denoted as *G ∗ G* , is a sequence *G ⇒ G ⇒*

*··· ⇒ Gn* of direct transformations.

0 *⇒ n* 0 1

**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](#_bookmark32)], 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*(*ni*) on *R* with *ni* : *R → Ni* of a rule *p* = (*L ← K → R*), the equivalent left application condition *Lp*(*N AC*(*ni*)) is defined in the following way:

*L* ¸*, K*  *R*

*'* (2)

*n*

*i*

J*'*

(1) *ni*

JJ

*Ni* ¸*, Z*  *N i*

* If the pair (*K → R, R → Ni*) has a pushout complement, we construct (*K → Z, Z → Ni*) as the pushout complement (1). Then we construct pushout (2) with the morphism *n'* : *L → N'*. Now we define *Lp*(*N AC*(*ni*)) = *N AC*(*n'* ).

*i i* *i*

* If the pair (*K → R, R → Ni*) does not have a pushout complement, we define

*Lp*(*N AC*(*ni*)) = *true*.

For each set of NACs on *R*, *N ACR* = *∪i∈IN AC*(*ni*) we define the following set of left NACs:

*'*

*Lp*(*N ACR*) = *∪i∈I' Lp*(*N AC*(*ni*))

with *i ∈ I'* if and only if the pair (*K → R, R → Ni*) has a pushout complement.

**Remark 2.10** Note that *Z* is unique since pushout complements along *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* *N ACR a set of right NACs on p, Lp*(*N ACR*) *as deﬁned in Def.* [*2.9*](#_bookmark5) *is a set of left NACs on p such that for all direct transformations G p,g H with comatch h,*

*⇒*

*g |*= *Lp*(*N ACR*) *⇔ h |*= *N ACR*

**Proof** The proof corresponds to case 1 and 3 in the proof of Theorem 7.17 in [[3](#_bookmark32)].

**Definition 2.12** [inverse rule with NACs] For a rule *p* : *L ← K → R* with *N ACp* = (*N ACL, ∅*), the inverse rule is defined by *p−*1 = *R ← K → L* with *NACp−*1 = (*Lp−*1 (*N ACL*)*, ∅*).

**Theorem 2.13 (Inverse Direct Transformation with NACs)** *For each dir- ect transformation with NACs G ⇒ H via a rule p* : *L ← K → R with N ACp*



Figure 1. graph model

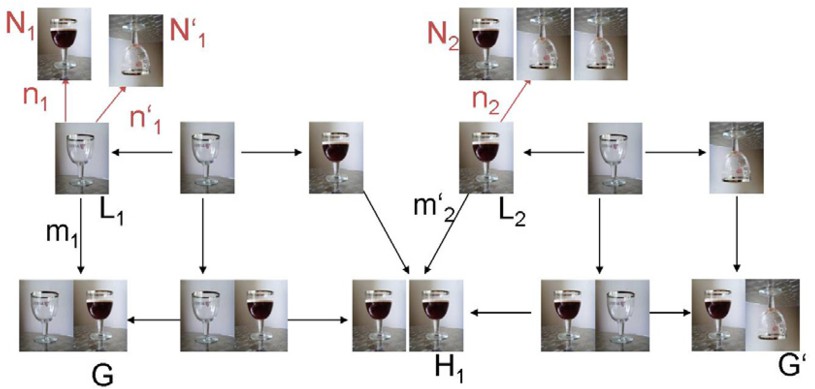


Figure 2. fill some glass and drink another one

*a set of left NACs on p, there exists an inverse direct transformation with NACs*

*H ⇒ G via the inverse rule p−*1 *with N ACp−*1 *.*

**Proof** This follows directly from Def. [2.12](#_bookmark7) and Lemma [2.11](#_bookmark6).

**Example 2.14** Consider as an example of an adhesive HLR system a graph trans- formation system consisting of two rules with NACs where *Q* = *M* = *M'* are injective typed graph morphisms. On the left hand side of Fig. [1](#_bookmark9) 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 = *fill* and *p*2 = *drink* of our example system are shown in the upper row of Fig. [2](#_bookmark10). 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](#_bookmark10), 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.

# 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 par- allel 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 inde- pendence 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 ⇒ H*1 with *N ACp*1 and *G ⇒ H*2 with *N ACp*2 are *parallel independent* if

(*p*=1*,m*1)

(*p*=2*,m*2)

and

*∃h*12 : *L*1 *→ D*2 *s.t.* (*d*2 *◦ h*12 = *m*1 *and e*2 *◦ h*12 *|*= *N ACp*1 )

*∃h*21 : *L*2 *→ D*1 *s.t.* (*d*1 *◦ h*21 = *m*2 *and e*1 *◦ h*21 *|*= *N ACp*2 )

as in the following diagram:

*N*,1 *,*

*n*1

*N*,2 *,*

*n*2

*R*1 ¸*, K*1 *L*1¸ *L*2 ¸*, K*2 *R* 2

*h* ¸¸¸¸ *h*

21 *m*1 ¸¸¸ *m*2 12

JJ *s*

z , *s*

zJ*˛* J

*H*1 ¸*e,*1

*D*1 *d*1

*G* ¸*,*

2

*d*

*D*2 *e*2 *H* 2

Two direct transformations *G* (*p*=1*,m*1)

*⇒*

*N ACp*2 are *sequentially independent* if

*H*1 with *N ACp*1

and *H*1

(*p*=2*,m*2)

*H*2 with

and

1

*∃h*12 : *R*1 *→ D*2 *s.t.* (*d*2 *◦ h*12 = *m'*

*and e*2 *◦ h*12 *|*= *N AC −*1 )

1

*⇒*

*p*

*∃h*21 : *L*2 *→ D*1 *s.t.* (*e*1 *◦ h*21 = *m*2 *and d*1 *◦ h*21 *|*= *N ACp*2 )

as in the following diagram:

*N*,1 *,*

*n*1

*N*,2 *,*

*n*2

*L*1 ¸*, K*1 *R* 1 ¸ *L*2 ¸*, K*2 *R* 2

*h* ¸¸¸¸ *h*

21 *'* ¸¸ *m*2 12

JJ *s m*1 ¸z , *s*

zJ*˛* J

¸*, D*1 *e*1 *H* 1 ¸*, d*

*G*

*d*1

2

*D*2 *e*2 *H* 2

**Remark 3.2** Note that as for the case without NACs we have the following rela-

tionship between parallel and sequential independency: *G p*1 *H p*2 *H* are sequen-

*⇒*

1 *⇒* 2

tially independent iff

*p−*1

*G ⇐*1

*p*

1 *⇒ H*2

*H* 2

are parallel independent.

**Theorem 3.3 (Local Church-Rosser Theorem with NACs)** *Given an ad- hesive HLR system with NACs AHS and two parallel independent direct trans-*

*⇒*

*formations with NACs H*1

*p*1*,m*1

*⇐*

*G p*2*,m*2

*H*2*, there are an object G' and direct*

*transformations H*1

*p*2*,m'*

*⇒* 2

*G' and H*2

*p*1*,m'*

*⇒* 1

*G' such that*

*p*1*,m*1

*G ⇒ H*1

*p*2*,m'*

*⇒* 2

*G' and*

*p*2*,m*2

*G ⇒ H*2

*p*1*,m'*

*⇒ G*

1 *'*

*are sequentially independent. Vice versa, given two sequentially*

*independent direct transformations with NACs G*

*p*1*,m*1

*⇒ H*1

*p*2*,m'*

*⇒* 2

*G' there are an*

*object and sequentially independent direct transformations*

*p*2*,m*2

*p*1*,m' '*

*H*2

*such that H*

*p*1*,m*1 *G p*2*,m*2 *H*

*are parallel independent:*

*G ⇒ H*2

*⇒* 1 *G*

1 *⇐ ⇒*

2

*p*1*,m*1. \_*H,*1 ¸¸¸*p*2*,m'*

...........

¸¸

*G* ¸¸¸¸¸¸¸¸¸¸¸

¸¸ 2

z*\_'*

.... \_*G,*

*p*2*,m*2¸ z*\_*...*p ,m'*

*H*2 1 1

## Proof

1. Given the parallel independent transformations *H*

*p*1*,m*1 *G p*2*,m*2 *H* :

1 *⇐ ⇒* 2

*L*1 ¸*, K*1 *R* 1 *L*2 ¸*, K*2 *R* 2

*m*1 *n*1 *m*2

J¸ *,* JJJ

*n*2

JJ

*f D*1 *g*1 *H* 1

*G*

*G*

1

*f*2

¸*, D*2 *g*2 *H* 2

Because of Def. [3.1](#_bookmark12) and the parallel independence with NACs of *H*1

*G p*2*,m*2

*p*1*,m*1

*⇐*

*⇒ H*2 we know that there exists *i*2 : *L*2 *→ D*1 (resp. *i*1 : *L*1 *→ D*2)

s.t. *f*1 *◦ i*2 = *m*2 (resp. *f*2 *◦ i*1 = *m*1) and moreover *g*1 *◦ i*2 *|*= *N ACp*2 (resp.

*g*2 *◦ i*1 *|*= *N ACp*1 ). Because of the Local Church-Rosser Theorem for parallel

independent transformations without NACs all necessary pushouts in *H*

*⇒*

*⇒*

*p*

1 *⇒ G*

2 *'*

and *H p*1

2 *⇒*

*G'* can be constructed s.t. *G p*1

*p*2

*H*1 *⇒*

*G'* and *G p*2

*p*1

*H*2 *⇒*

*G'* are

sequentially independent according to Def. 5.9 in [[3](#_bookmark32)] for direct transformations

without NACs. This means in particular that *t*1 : *R*1 *→ D'* (resp. *t*2 : *R*2 *→*

2

*D'* ) exist s.t. *s*1 *◦ t*1 = *n*1 (resp. *s*2 *◦ t*2 = *n*2) and the following pushout

1

diagrams exist:

*L*2 ¸*, K*2 *R* 2 *L*1 ¸*, K*1 *R* 1

*g*1*◦i*2

J *s*1

*g' ◦t*2

J *g'* J

1

2

1

*g*2*◦i*1 *g' ◦t*1

J *s* J *g'* J

2

*H*1 ¸*,*

*D'*  2 *G '*

*H*2 ¸*,*2 *D'*  1 *G '*

*p*2 *' p*1 *'*

*⇒*

*⇒*

Since *g*1 *◦ i*2 *|*= *N ACp*2 and *g*2 *◦ i*1 *|*= *N ACp*1 , *H*1 *G* and *H G* are

2

valid direct transformations with NACs. For the sequential independence of

*G p*1 *p*2 *'*

*⇒ H*1 *⇒ G*

we have to show that *i*2*, t*1 are the required morphisms. For *i*2 we

have *f*1*◦i*2 = *m*2, and therefore *f*1*◦i*2 *|*= *N ACp*2 follows by assumption. Now we

investigate *g' ◦ t*1. Because of Theorem [2.13](#_bookmark8) and the fact that *g*2 *◦ i*1 *|*= *N ACp*

2

it follows directly that also *g'*

2

1

* *t*1 *|*= *N AC −*1 . Analogously the sequential

*p*

1

independence of *G p*2 *H p*1 *G'* can be proven.

*⇒* 2 *⇒*

1. Given sequentially independent direct transformations with NACs *G p*1*,m*1

*⇒*

2

*p*2*,m'*

*⇒* 2 *G'* with comatches *n'*

*H*1 1

and *n'* , respectively, from Remark [3.2](#_bookmark13) we obtain

parallel independent direct transformations with NACs *G*

*p−*1*,n*1

1*⇐ H*1

*p*2*,m'*

*⇒ G*

2 *'*.

Now part (i) of the proof gives us sequentially independent direct transform-

ations with NACs *H*1

*p−*1*,n*1 1*⇒ G*

*p*2*,m*2

*⇒ H*2

and *H*1

*p*2*,m'*

*⇒* 2 *G*

*p−*1*,n'*

*'* 1*⇒* 1 *H*2

. Applying

again Remark [3.2](#_bookmark13) to the first transformation we obtain parallel independent

direct transformations with NACs *H*1

*p*1*,m*1

*⇐ G*

*p*2*,m*2

*⇒ H*2

*p−*1*,n*1.. \_*G,*¸¸¸*p*¸2*,m*2

1 .....

:

.

*H*

1 ¸¸¸¸¸¸

¸¸¸ z*\_*

\_*H,*2

..

..

*p*2*,m'* ¸ z*\_'* ...*p−*1*,n'*

2 *G* 1 1

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](#_bookmark32)] the following property. Let for Definition [3.5](#_bookmark15), Theorem [3.6](#_bookmark16), Definition [3.7](#_bookmark17) and Theorem

[3.8](#_bookmark18) *AHS* = (**C***, M, M', E', Q,P* ) be an adhesive HLR system with NACs, where (**C***, M*) has binary coproducts compatible with *M* (see Def. 5.14 in [[3](#_bookmark32)]).

**Definition 3.5** [parallel rule and transformation with NAC] Given two rules *p*1 =

(*L*1 *←l*1 *K*1 *→r*1 *R*1) with *N ACp* and *p*2 = (*L*2 *←l*2 *K*2 *→r*2 *R*2) with *N ACp* , the *parallel*

2

1

*rule p*1 + *p*2 with *N ACp*1+*p*2 is defined by the coproduct constructions over the corresponding objects and morphisms: *p*1+*p*2 = (*L*1+*L*2 *l*1*←*+*l*2 *K*1+*K*2 *r*1*→*+*r*2 *R*1+*R*2) and *N ACp*1+*p*2 = *{n*1 + *idL*2 *|n*1 *∈ N ACp*1 *}∪ {idL*1 + *n*2*|n*2 *∈ N ACp*2 *}*. A direct

transformation *G ⇒ G'* via *p*1 + *p*2 with *N ACp* +*p* and a match *m* : *L*1 + *L*2 *→*

1

2

*G* satisfying *N ACp*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 transforma- tions 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 com- position of a coproduct morphism with a morphism in Q be again in Q. Then given a sequentially independent direct transformation sequence with NACs G ⇒ H*1 *⇒ G'*

*via p*1*, m*1 *(resp. p*2*, m' ) with N ACp (resp. N ACp ), there is a construction lead-*

2

1

2

*ing to a parallel transformation with NACs G ⇒ G' via* [*m*1*, m*2] *and the parallel rule p*1 + *p*2 *with N ACp*1+*p*2 *, called a* synthesis construction.

*p*1*,m*.1... *G* ¸¸¸*p*¸2¸*,*¸*m*2

\_..*s* ¸ z*\_*

*H*1 ¸*p*¸1+*p*2*,*[*m*1*,m*2]. *H*2

¸¸¸

¸¸

c.*z*..

*p*2*,m'*

....

z*\_'* \_*ps*1*,m'*

2 *G* 1

**Proof** Given the sequentially independent direct transformations with NACs

*p*1*,m*1

*p*2*,m' '*

*G ⇒ H*1

*⇒* 2 *G* , using the Parallelism Theorem (Theorem 5.18 in [[3](#_bookmark32)]) without

NACs we can construct the following double pushout:

*L*1 + *L*2 ¸*, K*1 + *K*2 *R* 1 + *R*2

[*m*1*,m*2]

J¸ *,* J J *'*

*G D G*

Now we have to prove that [*m*1*, m*2] *|*= *N ACp*1+*p*2 such that according to Def. [3.5](#_bookmark15) this double pushout becomes a direct parallel transformation with NACs. Suppose that [*m*1*, m*2] */|*= *N ACp*1+*p*2 = *{n*1 + *idL*2 *|n*1 *∈ N ACp*1 *}∪ {idL*1 + *n*2*|n*2 *∈ N ACp*2 *}*. Then we have two cases.

* Suppose that *q* : *L*1 + *N*2 *→ G* exists such that *q ◦ idL*1 + *n*2 = [*m*1*, m*2] for some *n*2 *∈ N ACp*2 . Consider the following diagram in which *η*2 : *L*2 *→ L*1 + *L*2 and *η'* : *N*2 *→ L*1 + *N*2 are coproduct morphisms.

2

*L*2 *n*2 *N* 2

¸

*'*

2 *L*1+ *N*2

*η*

*¸*

*q* ,,\_ *G*

¸¸¸

*idL* +*n*2....

,,,,,,,

*η* ¸¸¸¸

,

1 ...,,,,,,

2 ¸z ,,,,

*L*1 + *L*2

[*m*1*,m*2]

Then by *q ◦ idL*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] *◦ η*2 =

*q ◦ idL*1 + *n*2 *◦ η*2 = *q ◦ η ◦ n*2. Because of the composition property on coproduct

*'*

2

morphisms and *Q* morphisms *q ◦ η' ∈ Q* and thus *m*2 */|*= *N AC*(*n*2). This is a

2

contradiction and thus [*m*1*, m*2] *|*= *N AC*(*idL*1 + *n*2).

* Suppose that *q* : *N*1 + *L*2 *→ G* exists such that *q ◦ n*1 + *idL*2 = [*m*1*, m*2]. Then we can prove analogously that *m*1 */|*= *N AC*(*n*1) and thus by contradiction that [*m*1*, m*2] *|*= *N AC*(*n*1 + *idL*2 ).

In order to be able to sequentialize a parallel transformation with NACs *G ⇒ G'* via a match *m* and a parallel rule *p*1 + *p*2 with *NACp*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' , m*2*, m'* occurring in the direct transformations when sequentializing the

1 2

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 *N ACp*1+*p*2 by *m*.

*p*1*,m*.1... *G* ¸¸¸*p*¸2¸*,*¸*m*2

\_..*s*

*H*1 ¸¸¸ *p*1+*p*2*,m*

¸ z*\_*

.. 2

*H*

¸¸

*'* ¸

c.*z*.. *'*

*p*2*,m*

...

z*\_'* \_*ps*1*,m*

2 *G* 1

**Definition 3.7** [NAC-compatible Parallel Transformation] Given a parallel trans- formation with NACs *G ⇒ G'* via match *m* : *L*1 + *L*2 *→ G* and the parallel rule *p*1 + *p*2 with *N ACp*1+*p*2 . Let *m*1 : *L*1 *→ G*, *m*2 : *L*2 *→ G* be the matches of the

direct transformations *G ⇒ H*1 and *G ⇒ H*2 via *p*1 resp. *p*2 and *m'* and *m'* the

1

2

matches of the direct transformations *H*2 *⇒ G'* and *H*1 *⇒ G'* via *p*1 resp. *p*2 as

constructed in the Parallelism Theorem without NACs (Analysis part in Theorem 5.18 in [[3](#_bookmark32)]). The parallel transformation with NACs *G ⇒ G'* is *NAC-compatible* if

2

*m*1*, m'*

1

*|*= *N ACp*1

and *m*2*, m'*

*|*= *N ACp*2 .

**Theorem 3.8 (Parallelism Theorem with NACs: Analysis)** *• Given a NAC-compatible direct parallel transformation with NACs G ⇒ G' via m* : *L*1 + *L*2 *→ G and the parallel rule p*1 + *p*2 *with N ACp*1+*p*2 *, then there is a construction leading to two sequentially independent transformation sequences*

*with NACs G ⇒ H*1 *⇒ G' via p*1*, m*1 *and p*2*, m' and G ⇒ H*2 *⇒ G' via p*2*, m*2

2

*and p*1*, m' , called an* analysis construction*.*

1

* Bijective Correspondence. *The synthesis construction of Theorem* [*3.6*](#_bookmark16) *and the analysis construction are inverse to each other up to isomorphism.*

*p*1*,m*.1... *G* ¸¸¸*p*¸2¸*,*¸*m*2

\_..*s*

*H*1 ¸¸¸¸ *p*1+*p*2

¸ z*H\_*

... 2

¸*'*¸¸

c.*z* *'*

*p*2*,m*2 z*\_'* \_*ps*1*,m*1

*G*

## Proof

* Given a parallel transformation with NACs *G p*1=+*p*2*,m*

*⇒*

*G'* then because of the

Parallelism Theorem without NACs (Theorem 5.18 in [[3](#_bookmark32)]) it follows that *G ⇒ H*1

and *G ⇒ H*2 are parallel independent without NACs and moreover the necessary

double pushouts for *G ⇒ H*1 *⇒ G'* via *p*1*, m*1 and *p*2*, m'* and *G ⇒ H*2 *⇒ G'*

2

via *p*2*, m*2 and *p*1*, m'* can be constructed s.t. they are sequentially independent

1

without NACs. Moreover we know because of NAC-compatibility of *G p*1=+*p*2*,m G'*

*⇒*

that *m*1*, m' |*= *N ACp* and *m*2*, m' |*= *N ACp* . Therefore *G ⇒ H*1 and *G ⇒ H*2

1

2

1 2

are parallel independent as transformations with NACs as defined in Def. [3.1](#_bookmark12). From Theorem [3.3](#_bookmark14) it follows that *G ⇒ H*1 *⇒ G'* and *G ⇒ H*2 *⇒ G'* are then sequentially independent with NACs.

* Because of the uniqueness of pushouts and pushout complements, the construc- tions 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](#_bookmark11). In Fig. [2](#_bookmark10), 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](#_bookmark19) the parallel transformation with NACs for this two-step transformation according to Theorem [3.6](#_bookmark16) and Def.

[3.5](#_bookmark15) is shown. On the contrary, in Fig. [4](#_bookmark20), you can see a parallel transformation with rules *p*1 = *drink* and *p*2 = *drink* which is not NAC-compatible. In order to

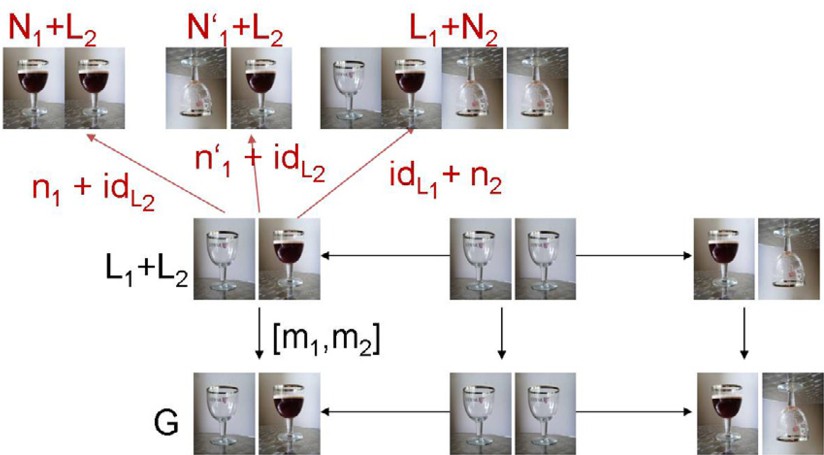


Figure 3. fill some glass and drink another one in parallel

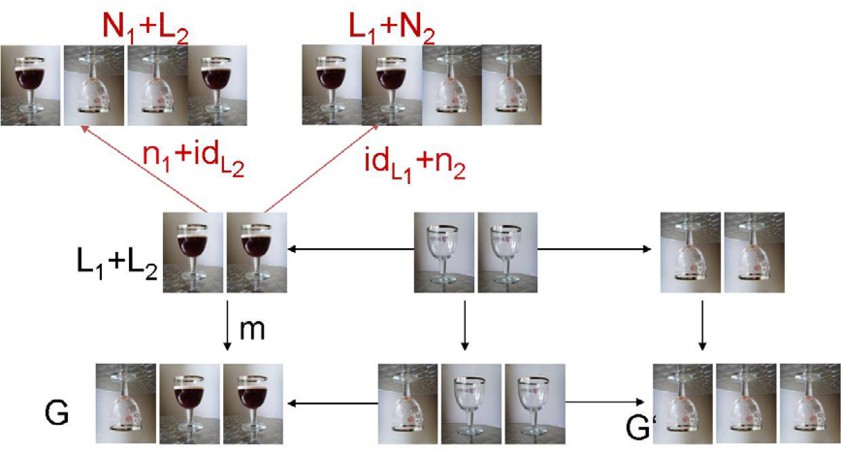


Figure 4. drink two glasses, if one is already drunk: parallel transformation which is not NAC-compatible

sequentialize a parallel transformation into two direct transformations with NACs it has to be NAC-compatible (see Theorem [3.8](#_bookmark18)). In Fig. [5](#_bookmark21) the parallel transformation in Fig. [4](#_bookmark20) without NACs is sequentialized. Afterwards it is checked if the NACs *N AC*(*N*1) on *p*1 = *drink* (resp. *N AC*(*N*2) on *p*2 = *drink* are satisfied. It becomes clear that NAC-compatibility as defined in Def. [3.7](#_bookmark17) is violated because there exists

*q'* : *N*2 *→ H*1 with *q' ◦ n*2 = *m'* (and analogously *q'* : *N*1 *→ H*2 with *q' ◦ n*1 = *m'* ).

2 2 2 1 1 1

Intuitively speaking, the second transformation in Fig. [5](#_bookmark21) 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.

# Concurrency in adhesive HLR Systems with NACs

Let *t* be a transformation via the rules *p*0*, ··· , pn−*1 with NACs and matches *g*0*, ··· , gn−*1. In general there will be dependencies between several direct trans- formations in this transformation sequence. Therefore it is not possible to apply the Parallelism Theorem in order to summarize the transformation sequence into

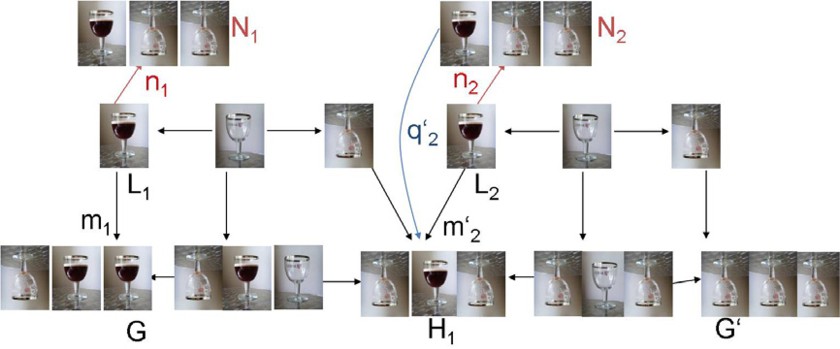


Figure 5. drink a glass, if one is already drunk and afterwards drink another one

one equivalent transformation step. It is possible though to formulate a Concur- rency 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](#_bookmark32)]. Moreover we have to translate all the NACs occuring in the rule sequence *p*0*, ··· , pn−*1 backward into an equivalent set of NACs on the concurrent rule *pc* of this rule sequence. This means, a set *N ACpc* should be found such that this set of NACs is equivalent to *NACp*0 *, ··· ,NACpn−*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:

*N*,0 *, N*,1 *,*

*L*0 ¸*, K*0 *R* 0 ¸ *L*1 ¸*, K*1 *R* 1

JJ

¸¸¸¸

*h*0 ¸¸¸

*G* 1 ¸*,*

*g*1

JJ

*G*0 ¸*, D*0

z,*s*

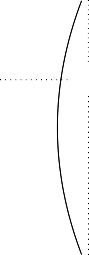
*D*1 *G*2

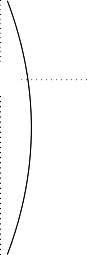
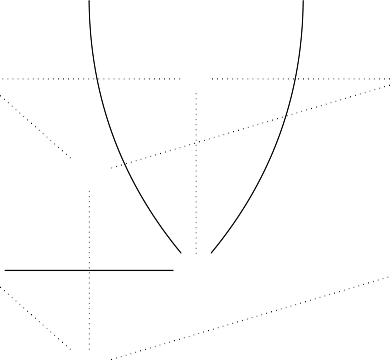
The goal is to translate all NACs on *p*0 and *p*1 into an equivalent set of NACs on the concurrent rule *pc* : *Lc ← Kc → Rc* inducing as explained in Theorem 5.23 in [[3](#_bookmark32)] a concurrent transformation *G*0 *⇒ G*2 via *pc* and match *gc* as shown in the following

diagram:

*N*,0 *,*

*N*,1 *,*

*L*0 ¸*, K*0 *R* 0 *L*1 ¸*, K*1 *R* 1



\_*C*1

J

*R c*

J

JJ

J

*Lc* ¸*,*

*gc*

tJ*z*

*e*0

J

*C*0 ,*¸*

*h*0

*Kc*

tJ*z*

*e*1

z¸,*,s*

*E*

*g*1

*h*

zJ*z*

*G*0 ¸*, D*0 ,*¸*

J

*Dc*

*G* 1 ¸*,* \_*D* 1 *G* 2

Consequently the two necessary steps are:

* Translate each set of NACs on *L*0 into an equivalent set of NACs on *Lc*.
* Translate each set of NACs on *L*1 into an equivalent set of NACs on *Lc*.

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

**Definition 4.1** [construction of NACs on *Lc* from NACs on *L*0] Consider the fol- lowing diagram:

*Nj ei* *N '*



,*,* ,*i ,*

*nj* (1) *n'*

*i*

*L*0 *m*0 *Lc*

For each *N AC*(*nj*) on *L*0 with *nj* : *L*0 *→ Nj* and *m*0 : *L*0 *→ Lc*, let

*Dm* (*N AC*(*nj*)) = *{N AC*(*n'* )*|i ∈ I, n'* : *Lc → N'}*

*o i i* *i*

where *I* and *n'* are constructed as follows: *i ∈ I* if and only if (*ei, n'* ) with *ei* : *Nj →*

*i* *i*

*N'* jointly epimorphic, *ei ◦ nj* = *n' ◦ m*0 and *ei ∈ Q*.

*i* *i*

For each set of NACs *N ACL*0 = *{N AC*(*nj*)*|j ∈ J}* on *L*0 the downward translation

of *N ACL*0 is then defined as:

*Dmo* (*N ACL*0 ) = *∪j∈J Dmo* (*N AC*(*nj*))

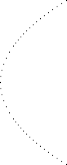
## Lemma 4.2 (equivalence of set of NACs on *L*0 and set of NACs on *Lc*)

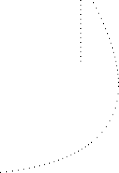
*Given gc* : *Lc → G*0*, m*0 : *L*0 *→ Lc with N ACL*0 *and g*0 = *gc ◦ m*0 *as in the*

*following diagram:*

*then the following holds :*

*Nj*  *ei*  *N '*

*q*



,*,* ,*i ,*

*nj*

(1) *n'*

*i*

*L*0

*m*0

*Lc*

*g*0

=

z

*G*0 ¸*,*

J,

*s*

*g*

*c*

*q'*

*g*0 *|*= *N ACL*0 *e gc |*= *Dm*0 (*N ACL*0 )

*.*

## Proof

* (*⇒*) Let *gc /|*= *Dm*0 (*N AC*(*L*0)) = *∪j∈J Dm*0 (*N AC*(*nj*)) with *N ACL*0 =

*{N AC*(*nj*)*|j ∈ J}*. Then for some *j ∈ J* there is a NAC *n'* : *Lc → N' ∈*

*i*

*i*

*Dm* (*N AC*(*nj*)) and *ei* : *Nj → N'* for which holds that *gc /|*= *N AC*(*n'* ), (*ei, n'* )

0 *i i* *i*

jointly epi, *ei ∈Q* and *ei ◦ nj* = *n' ◦ m*0. Consequently there exists a morphism

*i*

*q'* : *N' → G*0 *∈ Q* such that *q' ◦ n'* = *gc*. Since *g*0 = *gc ◦ m*0 = *q' ◦ n' ◦ m*0 =

*i i* *i*

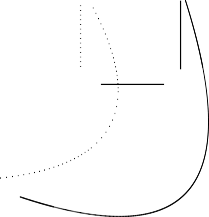
*q' ◦ ei ◦ nj* there exists a morphism *q* : *Nj → G*0 defined by *q* = *q' ◦ ei* s.t. *q ◦nj* = *q' ◦ei ◦nj* = *g*0. Because of the composition property for morphisms in *Q* we have *q ∈Q* since *q' ∈Q* and *ei* in *Q*. Hence *g*0 */|*= *N AC*(*nj*) *⇒ g*0 */|*= *N ACL* .

0

* (*⇐*) Let *g*0 */|*= *N ACL*0 with *N ACL*0 = *{N AC*(*nj*)*|j ∈ J}*. Then for some *j ∈ J* a morphism *q* : *Nj → G*0 *∈ Q* exists such that *q ◦ nj* = *g*0. Let (*e∗, m∗*) be an epi-*ł*-factorization of *gc*. Construct *X* with *p*1 : *X → E* and *m*1 : *X → Nj* as pullback of *m∗* and *q*.

*Nj* ¸*m,*1 *X*

*p*2*∈Q*

*Nj*  *ei*  *N '*

z*Y ˛*

*q* (*PB*) *p*1

*E*

,*,*

*nj* (1) *n'*

*i*

,*i ,* ,*,*

*m*2

J

*m∗*

*G*0 ¸¸*¸,*

J¸ *e,∗ L*

*q L*0

*m*0 *Lc*  *E*

*gc* =

*e∗*

*g*

*c*

0 *g*

*c*

tJ*z* , *s q'*

¸*,*

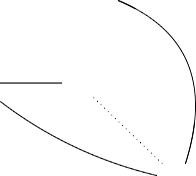
*G*0 ¸*¸*

*x*

Then we have *m*1 *∈ł* and *p*1 *∈ Q*, since *m∗ ∈ ł*, *q ∈ Q*, PBs preserve *ł* and PBs along *ł* preserve *Q*. Now construct *Y* with *m*2 : *E → Y* and *p*2 : *Nj → Y* as pushout of *m*1 and *p*1. Then we have *m*2 *∈ ł*, *p*2 *∈ Q*, since *m*1 *∈ ł*, *p*1 *∈ Q*, POs preserve *ł* and POs along *ł* preserve *Q*. Because of the induced PB-PO property the induced morphism *x* : *Y → G*0 with *x ◦ m*2 = *m∗* and *x ◦ p*2 = *q* is

a monomorphism in *Q*.

*X*  *p*1 *E*



*m*1 (*PO*) *m*2

*m∗*

J *p*2 J

*Nj Y*

*x*

*q* v *z)*

z

*G*0

It holds moreover that *p*2*, m*2 *◦ e∗* jointly epimorphic because *e∗* epimorphic and *p*2*, m*2 jointly epimorphic. Summarizing we have the following equations: *x◦m*2 *◦ e∗ ◦ m*0 = *m∗ ◦ e∗ ◦ m*0 = *gc ◦ m*0 = *g*0 = *q ◦ nj* = *x ◦ p*2 *◦ nj* and since *x* mono we have *m*2 *◦ e∗ ◦ m*0 = *p*2 *◦ nj*. Since *m*2 *◦ e∗* and *p*2 are jointly epimorphic, *p*2 *◦ nj* = (*m*2 *◦ e∗*) *◦ m*0 and *p*2 *∈ Q* we can conclude that *m*2 *◦ e∗* : *Lc → Y*

equals one of the morphisms *n'* : *Lc → N' ∈ Dm* (*N AC*(*nj*)). Moreover since

*i i* 0

*x◦m*2 *◦e∗* = *m∗ ◦e∗* = *gc* and *x ∈Q* it holds that *gc /|*= *N AC*(*m*2 *◦e∗*) = *N AC*(*n'* )

*i*

and consequently *gc /|*= *NAC*(*n'* ) *⇒ gc /|*= *Dm* (*N AC*(*nj*)) *⇒ gc /|*= *Dm* (*N ACL* ).

*i* 0 0 0

**Remark 4.3** It is possible to cancel the fact that *Q* is a class of special morphisms, by defining that *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 *ł* or each match is in *ł*. (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 explicitely in [[8](#_bookmark39)].

In Def. [2.9](#_bookmark5) and Lemma [2.11](#_bookmark6) 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 *Lc* from NACs on *L*1] Given an E- dependency relation (*e*0*, e*1) *∈ E'* for the rules *p*0 and *p*1 and *pc* = *p*0 *∗E p*1 : *Lc ← Kc → Rc* the E-concurrent rule of *p*0 and *p*1 as depicted in the following diagram:

*N*

*j*

,*,*



*z*1

*N'* ,*s nj*

*L*0 ¸*, K*0 *R* 0

JJ

,*i ,*

(1)

*'*

*n*

*j*

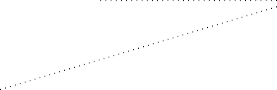
*e*0 *e*1

z,*s*

*L*1 ¸*, K*1 *R* 1

JJ

*Lc* ¸*, C*0 ,*¸*



*E* ¸*,*

\_*C*1

*R c*

*Kc*

For each *N AC*(*nj*) on *L*1 with *nj* : *L*1 *→ Nj*:

*DLpc* (*N AC*(*nj*)) = *Lp*(*De*1 (*N AC*(*nj*)))

with *p* : *Lc ← C*0 *→ E* and *De*1 *, Lp* according to Def. [4.1](#_bookmark22) and Def. [2.9](#_bookmark5).

For each set of NACs *N ACL*1 = *{N AC*(*nj*)*|j ∈ J}* on *L*1 the down- and leftward

translation of *N ACL*1 is defined as:

*DLpc* (*N ACL*1 ) = *∪j∈J DLpc* (*N AC*(*nj*))

**Lemma 4.5 (equivalence of NACs on rule** *p*1 **and NACs on** *pc***)** *Given a two-step E-related transformation via p*0 : *L*0 *← K*0 *→ R*0 *and p*1 : *L*1 *← K*1 *→ R*1

*N*

*j*

,*,*

*L*0 ¸*, K*0 *R* 0 ¸¸

¸¸¸

*nj*

*L*1 ¸*, K*1 *R* 1



JJ

*h*0 ¸¸¸

*g*1

JJ

*G*0 ¸*, D*0

z,*s*

*D*1 *G*2

*G* 1 ¸*,*

*with gc being the match from the LHS of the E-concurrent rule pc* = *p*0 *∗E p*1 *into G*0 *(as described in the synthesis construction of Theorem 5.23 in [*[*3*](#_bookmark32)*]) then the following holds:*

*g*1 *|*= *N ACL*1 *e gc |*= *DLpc* (*N ACL*1 )*.*

**Proof** Consider the following diagram:

*N*

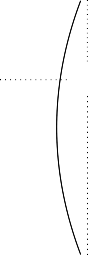
*j*

,*,*



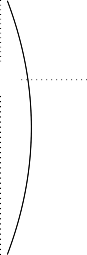
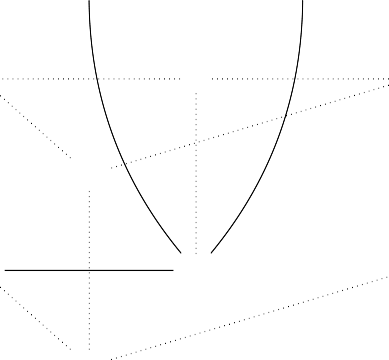
*z*1

*N'* ,*s nj*

*L*0 ¸*, K*0 *R* 0

,*i ,*

(1)



\_*C*1

J

*R c*

J

JJ

*'*

*n*

*j*

*L*1 ¸*, K*1 *R* 1

J

*Lc* ¸*,*

*gc*

tJ*z*

*e*0

J

*C*0 ,*¸*

*h*0

*Kc*

tJ*z*

*e*1

z,¸*s,*

*E*

*g*1

*h*

zJ*z*

*G*0 ¸*, D*0 ,*¸*

J

*Dc*

*G* 1 ¸*,* \_*D* 1 *G* 2

The E-concurrent rule *p*0 *∗E p*1 is the rule *pc* : *Lc ← Kc → Rc*, as described in

Def. 5.21 in [[3](#_bookmark32)]. The derived span of the E-concurrent transformation *G*0

*pc,gc*

*⇒ G*2

is *G*0 *← Dc → G*2. Because of Lemma [4.2](#_bookmark23) *g*1 *|*= *N ACL*1 *e h |*= *De*1 (*N ACL*1 ). Moreover because of Lemma [2.9](#_bookmark5) *gc |*= *Lp*(*De*1 (*N ACL*1 )) *e h |*= *De*1 (*N ACL*1 ) with *p* : *Lc ← C*0 *→ E*. Note that *Lp*(*De*1 (*N ACL*1 )) = *∪j∈J Lp*(*De*1 (*N AC*(*nj*))) =

*∪j∈J DLpc* (*N AC*(*nj*)). Consequently, it holds that *g*1 *|*= *N ACL*1 *e gc |*=

*DLpc* (*N ACL*1 )).

**Definition 4.6** [concurrent rule with NAC, concurrent (co-)match induced by

*G*0 =*n*+*⇒*1 *Gn*+1]

*n* = 0 For a direct transformation *G*0 *⇒ G*1 via match *g*0 : *L*0 *→ G*0, comatch *g*1 : *R*1 *→ G*1 and rule *p*0 : *L*0 *← K*0 *→ R*0 with *N ACp*0 the *concurrent rule pc with NAC* induced by *G*0 *⇒ G*1 is defined by *pc* = *p*0 with *N ACpc* = *N ACp*0 , the *concurrent comatch hc* is defined by *hc* = *g*1 and the *concurrent match gc* by *gc* = *g*0 : *L*0 *→ G*0.

*n ≥* 1 Consider *p'*

*c*

: *L'*

*← K'*

*→ R'*

(resp. *g'*

: *L'*

*→ G*0, *h'*

: *R'*

*→ Gn*), the

*n*

concurrent rule with NACs (resp. concurrent match, comatch) induced by *G*0 =*⇒*

*c*

*c*

*c*

*c*

*c*

*c*

*c*

*Gn*. Let ((*e' , en*)*, h*) be the *E' − ł'* pair factorization of the comatch *h'* and

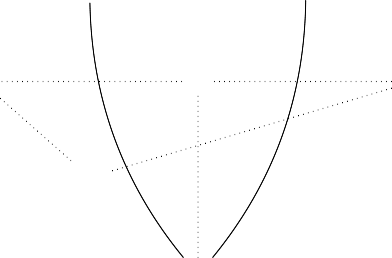
*c c*

match *gn* of *Gn ⇒ Gn*+1. According to Fact 5.29 in [[3](#_bookmark32)] PO-PB decomposition,

PO composition and decomposition lead to the diagram below in which (1) is a pullback and all other squares are pushouts:

*L'* ¸*, K'*  *R ' Ln* ¸*, Kn*  *R n*

*c c c*



J

*R c*

*gn*

*c*

J

*m*

*c*

*e' en*

J

*l*

*Lc* ¸*,*

*'*

*g*

*c*

*gc*

J

*Cc* ,*¸*

*kc*

(1)

*Kc*

*c*  z,¸*s,*

*h*

*E*

J *r*

\_*Cn*

*kn*

+1

*h*

tJ*z* J

*' gn*

*c*

*h*

zJ*z* J

*G*0 ¸*, Dn*  *G n* ¸*, D*  *G n*+1

For a transformation sequence *G*0 =*n*+*⇒*1 *Gn*+1 the *concurrent rule pc with NACs*

(resp. concurrent match, comatch) induced by *G*0 =*n*+*⇒*1 *Gn*+1 is defined by *pc* =

*l◦k*

*← K*

*c*

*Lc c*

*→ Rc* (*gc* : *Lc → G*0, *hc* : *Rc → Gn*+1). Thereby *N ACpc* is defined by

*N ACpc* = *DLpc* (*N ACLn* ) *∪ Dmc* (*N ACL'* ).

*r◦kn*

*c*

**Theorem 4.7 (Concurrency Theorem with NACs)** (i) Synthesis. *Given a*

*transformation sequence t* : *G* =*∗ G via a sequence of rules p ,p ,...,p ,*

0 *⇒ n*+1

0 1 *n*

*then there is a* synthesis construction *leading to a direct transformation G*0 *⇒*

*Gn*+1 *via the concurrent rule pc* : *Lc ← Kc → Rc with N ACpc , match gc* :

=

*G*

*∗*

*Lc → G*0

*and comatch hc*

: *Rc*

*→ Gn*+1

*induced by t* : *G*0

*⇒ n*+1*.*

1. Analysis. *Given a direct transformation G'*

0

*'*

*n*+1

*⇒ G*

=

*via the concurrent rule pc* :

*Lc ← Kc*

*→ Rc*

*with N ACpc*

*induced by t* : *G*0

*∗*

*⇒ n*+1

*G*

*via a sequence of rules*

*p*0*, p*1*,..., pn then there is an* analysis construction *leading to a transformation*

*sequence t'* : *G'* =*∗ G' with NACs via p ,p ,...,p .*

0 *⇒ n*+1

0 1 *n*

1. 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.

1. *Synthesis.*

Basis. n=0. For a direct transformation *t* : *G*0

*p*0*,g*0

*⇒ G*1

via match *g*0

: *L*0

*→ G*0

and rule *p*0 : *L*0 *← K*0 *→ R*0 with *N ACp*0 the *concurrent rule pc with NAC* induced by *G*0 *⇒ G*1 is defined by *pc* = *p*0 with *N ACpc* = *N ACp*0 and the *concurrent match gc* is defined by *gc* = *g*0 : *L*0 *→ G*0. Therefore the

synthesis construction is equal to *G*

*pc,gc G* .

0 *⇒* 1

Induction Step. Consider *t* : *G*0 =*n⇒ Gn ⇒ Gn*+1 via the rules *p*0*, p*1 *..., pn*.

*c*

*c*

*c*

*c*

*c*

*c*

*c*

Let *p'*

*c*

: *L'*

*← K'*

*→ R'*

(resp. *g'*

: *L'*

*→ G*0, *h'*

: *R'*

*→ Gn*), be the

concurrent rule with NACs (resp. concurrent match, comatch) induced by

*n p' ,g'*

*G*0 =*⇒ Gn*. Suppose that *G*0 =*c⇒c Gn* is a direct transformation with NAC

leading to *Gn*. Let ((*e' , en*)*, h*) be the *E' − ł'* pair factorization of the co- match *h'* and match *gn* of *Gn ⇒ Gn*+1. PO-PB decomposition, PO compos- ition and decomposition as described in Fact 5.29 in [[3](#_bookmark32)] lead to the diagram

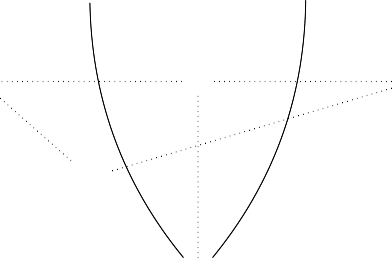
*c*

*c*

below in which (1) is a pullback and all other squares are pushouts:

*L'* ¸*, K'*  *R ' Ln* ¸*, Kn*  *R n*

*c c c*



J

*R c*

*gn*

*c*

J

*m*

*c*

*e' en*

J

*l*

*Lc* ¸*,*

*'*

*g*

*c*

*gc*

J

*Cc* ,*¸*

*kc*

(1)

*Kc*

*c*  z,¸*s,*

*h*

*E*

J *r*

\_*Cn*

*kn*

+1

*h*

tJ*z* J

*' gn*

*c*

*h*

zJ*z* J

*G*0 ¸*, Dn*  *G n* ¸*, D*  *G n*+1

The *concurrent rule pc with NACs* (resp. concurrent match, comatch) in-

*→*

duced by *G*0

=*n*+*⇒*1

*Gn*+1

is *pc*

= *Lc*

*l◦kc*

*c*

*←*

*K*

*r◦kn*

*Rc* (*gc*

: *Lc*

*→ G*0, *hc* :

*Rc → Gn*+1). Thereby *N ACpc* is *N ACpc* = *DLpc* (*N ACLn* ) *∪ Dmc* (*N ACL'* ).

*c*

*⇒*

We should prove that *G*0

*p*=*c,gc*

*Gn*+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](#_bookmark32)] can be done. Thus,

in a second step we shall show that *gc* satisfies *N ACp* if *g'* satisfies *N ACp'*

*c c c*

and *gn* satisfies *N ACpn* . This follows because of Lemma [4.2](#_bookmark23), Lemma [4.5](#_bookmark24)

*p' ,g'*

and the fact that *G*0 =*c⇒c Gn* is a direct transformation via the rule *p'* with

*c*

concurrent NAC *N ACp'* .

*c*

1. *Analysis.*

Basis. n=0. For a direct transformation *G'*

0

*⇒ G'*

1

via the concurrent rule

0

*pc* = *p*0 with *N ACpc*

= *N ACp*0

the analysis construction is equal to *G' ⇒*

*G'* .

1

Induction Step. Given a direct transformation *G'*

0

*'*

*n*+1

*⇒ G*

via the concurrent

rule *pc*

: *Lc*

*← Kc*

*→ Rc*

with *N ACpc*

induced by *t* : *G*0

*∗*

*⇒ n*+1

=

*G*

via a

sequence of rules *p*0*, p*1*,..., pn*. The concurrent rule *pc* induced by *t* can be

interpreted as *p' ∗E pn* in which the *En*-dependency relation between the

*c*

*n*

rules is induced by the *E' − ł'* pair factorization of the comatch *h'* induced

by *G*0

*∗*

*⇒ Gn*

=

and the match *gn*

*c*

of *Gn → Gn*+1 as described in Def. [4.6](#_bookmark25). So

*c*

*n*

we have a direct transformation *G'*

0

*'*

*n*+1

*⇒ G*

via *pc* = *p' ∗E*

*pn* and because

of the Analysis part of Theorem 5.23 in [[3](#_bookmark32)] there is an analysis construction

*n*

leading to a transformation sequence without NACs *G'*

0

*⇒ G'*

*'*

*n*+1

*⇒ G*

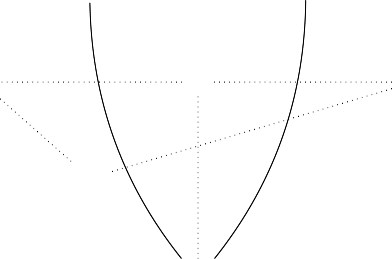
via

*p'* and *pn* and matches *g''* resp. *g'* .

*c c n*

*L'* ¸*, K'*  *R '*

*Ln* ¸*, Kn*  *R n*



J

*R c*

*g'*

*n*

*'* J

*c c c*

*m*

*c*

*e' en*

J

*l*

*Lc* ¸*,*

*''*

*g*

*c*

*g'*

tJ*' z*

J

*Cc* ,*¸*

*kc*

J*'*

(1)

*Kc*

*c*  z,¸*s,*

zJ*'z*

*E*

J *r*

\_*Cn*

*kn*

+1

J

*G*0 ¸*, Dn*  *G n* ¸*, D'*  *G n*+1

We know by assumption that the match *g'* of *G'*

0

*'*

*n*+1

*⇒ G*

satisfies *N ACpc* .

Since Lemma [4.2](#_bookmark23) and Lemma [4.5](#_bookmark24) hold in both directions, i.e. translate NACs

in an equivalent way, we can conclude that *NACp'* and *N ACpn* are satisfied

*c*

by *g''* resp. *g'* . Therefore *G' ⇒ G' ⇒ G'*

is a valid transformation

*c n* 0 *n n*+1

sequence with NACs. Because of the induction hypothesis there exists an

1

*n*

0

*n*

analysis construction *G'*

0

*⇒ G'*

*⇒ ... G'*

via *p*0*, p*1*,..., pn−*1 for *G'*

*⇒ G'*

via *p'* . Thus we obtain a transformation sequence with NACs *G' ⇒ G' ⇒*

*c*

*... G'*

via *p*0*, p*1*,..., pn* for the direct transformation *G'*

0

*⇒ G'*

1

via the

*n*+1

concurrent rule *pc* : *Lc ← Kc → Rc* with *N ACpc* .

0 *n*+1

1. *Bijective Correspondence.* The bijective correspondence follows from the fact that the *E' − ł'* pair factorization is unique, and pushout and pullback con- structions are unique up to isomorphism.

**Example 4.8** We continue with Example [2.14](#_bookmark11). In Fig. [6](#_bookmark27) you can see a trans- formation 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](#_bookmark25) and Theorem [4.7](#_bookmark26) which is shown in Fig. [10](#_bookmark31). The following three steps explain this synthesis construction of the two-step transformation. In Fig. [7](#_bookmark28) the construction of the concurrent rule according to Def. [4.6](#_bookmark25) induced by this two-step transformation without NACs is shown. In Fig. [8](#_bookmark29) the construction is shown of the equivalent NACs on *Lc* from *N AC*(*n*1) and *N AC*(*n*2) on *L*0. In Fig. [9](#_bookmark30) the construction is shown of

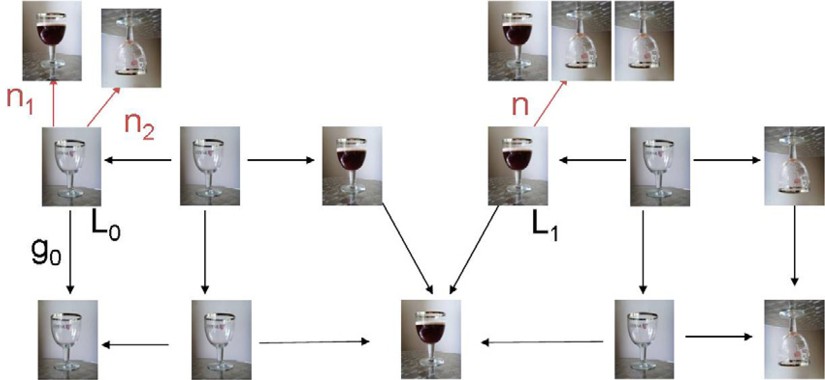


Figure 6. fill some glass and drink the same one

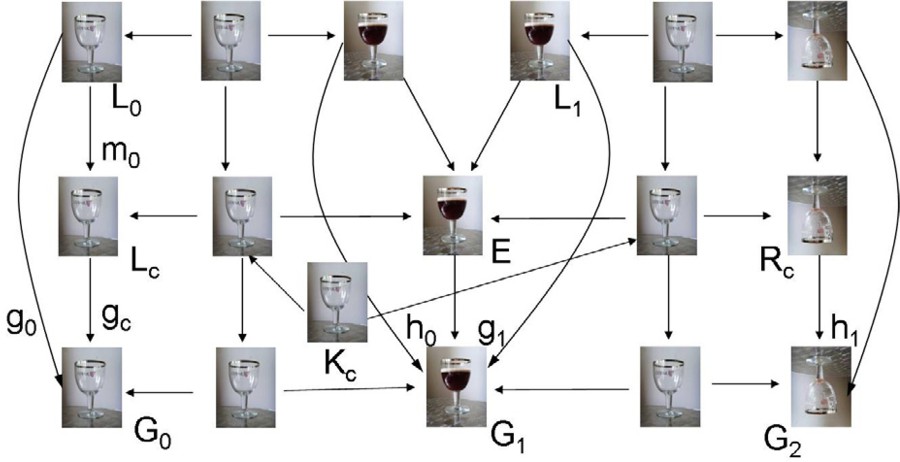


Figure 7. concurrent rule induced by two-step transformation

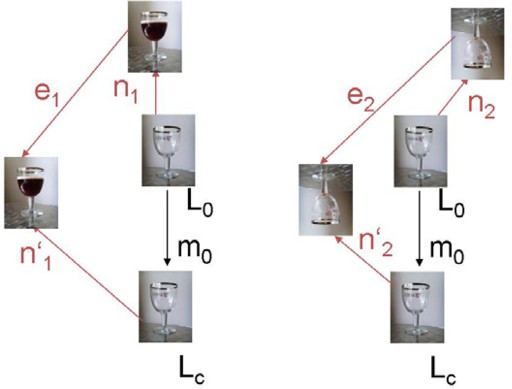


Figure 8. equivalent *NAC* on *Lc* to *NAC*(*n*1) and *NAC*(*n*2) on *L*0

the equivalent NAC on *Lc* from *N AC*(*n*) on *L*1 according to Lemma [4.5](#_bookmark24).

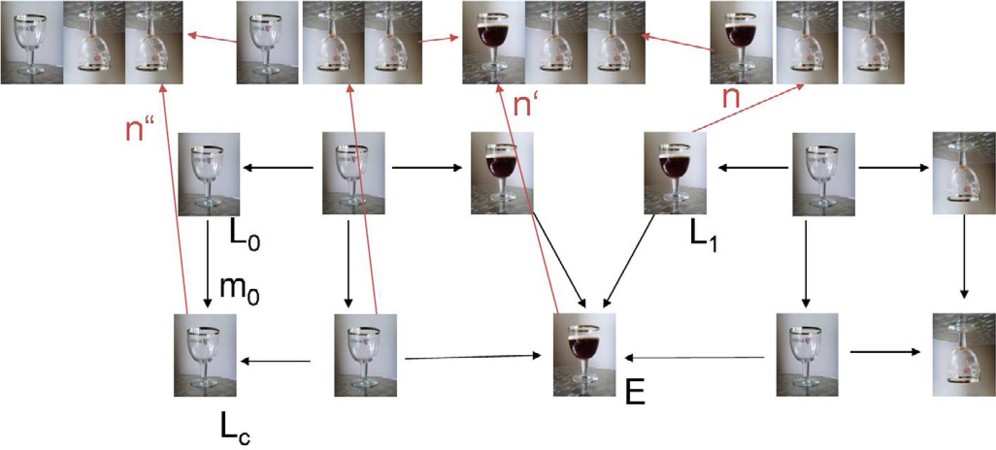


Figure 9. equivalent *NAC* on *Lc* to *NAC*(*n*) on *L*1

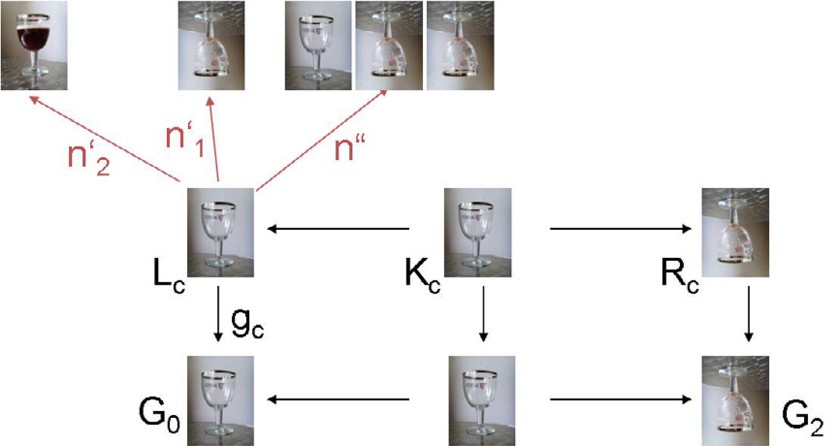


Figure 10. fill some glass and drink the same one concurrently

# Conclusion

In this paper results for parallelism and concurrency are described in order to ex- tend these notions on Algebraic Graph Transformation in [[3](#_bookmark32)] to Transformations with Negative Application Conditions. Summarizing we have proven the Local- Church-Rosser Theorem, Parallelism Theorem and Concurrency Theorem for trans- formations with NACs. These results are formulated in the context of the Adhes- ive High-Level Replacement Framework introduced in [[3](#_bookmark32)] with an extra necessary morphism class *Q*. This makes these results on parallelism and concurrency ap- plicable to all transformation systems with NACs fitting into the adhesive HLR framework. Moreover these results serve as a basis for the extension of other no- tions like Embedding and Local Confluence for adhesive HLR systems with NACs described technically already in [[8](#_bookmark39)]. Future work will be necessary on the applicab- ility and refinement of all new results and on the development of efficient analysis algorithms for transformations with NACs.

# References

1. Bottoni, P., A. Schu¨rr 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. L¨owe, *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>

1. Habel, A., R. Heckel and G. Taentzer, *Graph grammars with negative application conditions* **26** (1996),

pp. 287–313.

1. 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.
2. Koch, M., L. Mancini and F. Parisi-Presicce, *Graph-based Specification of Acces Control Policies*, in:

*JCSS 71*, 2005, pp. 1–33.

1. Lack, S. and P. Sobocinski, *Adhesive and quasiadhesive categories*, Theoretical Informatics and Applications **39(2)** (2005), pp. 522–546.
2. Lambers, L., *Adhesive high-level replacement systems with negative application conditions*, Technical report, Technische Universita¨t Berlin (2007).
3. 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/LEO05.pdf>

1. 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/LEO06a.pdf>

1. 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/LEO06.pdf>

1. 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](http://www.cs.tu-berlin.de/~gabi/gMTR04.pdf)

1. 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](http://tfs.cs.tu-berlin.de/publikationen/Papers05/TEG%2B05.pdf)