Composition and Transformation of High Level Petri Net-Processes

Karsten Gabriel

December 21, 2009

Diploma Thesis

Supervisors: Prof. Dr. Hartmut Ehrig
Dr. Claudia Ermel

Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin

Eidesstattliche Erklärung

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt
Berlin, 21. Dezember 2009

Karsten Gabriel

Acknowledgment

I would like to thank Prof. Dr. Hartmut Ehrig, Dr. Kathrin Hoffmann and Dr. Benjamin Braatz for their professional support and constructive criticism.
Furthermore special thanks to Antje Hübel for her support and understanding.

Abstract

In this thesis we study processes of algebraic high-level nets, where processes are modelled using petri nets and data handling is based on algebraic data types. The new contribution of the thesis is to define and analyze different kinds of composition and rule-based transformation for high-level processes in the sense of high-level replacement systems which generalize graph transformation systems.
The first main result is the existence and construction of direct transformations of high-level processes in the double pushout approach which allows a stepwise rule-based construction of high-level processes for a given algebraic high-level net.
The second main result shows how to compose processes of different algebraic high-level nets leading to a compositional high-level petri net process semantics.

Keywords: high-level petri nets, high-level processes, composition, decomposition, transformation, amalgamation

Zusammenfassung

In dieser Arbeit werden Prozesse von algebraischen High-Level Netzen untersucht. Die Prozesse sind dabei durch Petrinetze modelliert, während die Behandlung der Daten auf algebraischen Datentypen basiert. Der neue Beitrag dieser Arbeit besteht in der Definition und Analyse verschiedener Arten der Komposition sowie der regelbasierten Transformation von High-Level Prozessen im Sinne von High-Level Ersetzungssystemen, welche eine Generalisierung von Graphtransformationssystemen darstellen.
Das erste Hauptresultat dieser Arbeit ist die Existenz und Konstruktion direkter Transformationen von High-Level Prozessen im Doppel-Pushout-Ansatz. Dies erlaubt eine schrittweise regelbasierte Konstruktion von High-Level Prozessen zu einem gegebenen algebraischen High-Level Netz.
Das zweite Hauptresultat zeigt, wie Prozesse verschiedener algebraischer High-Level Netze komponiert werden können. Dies führt zu einer kompositionalen Semantik von High-Level Petrinetz-Prozessen.

Schlüsselwörter: High-Level Petrinetze, High-Level Prozesse, Komposition, Dekomposition, Transformation, Amalgamierung

Contents

1 Introduction **3**

2 Algebraic High-Level Processes **5**
2.1 Place/Transition Nets and Algebraic High-Level Nets 5
2.2 Algebraic High-Level Processes 10
2.3 Flattening and Skeleton Construction 13
2.4 Algebraic High-Level Processes with Instantiatons 20

3 Parallel and Sequential Composition and Independence of Algebraic High-Level Processes **24**
3.1 Parallel and Sequential Composition of Algebraic High-Level Processes with Instantiations 24
3.2 Equivalence and Independence of Algebraic High-Level Processes 31

4 Instantiations and Initializations **38**
4.1 Category of Instantiations 38
4.2 Preimage Constructions 40
4.3 Pushout Construction of Instantiations 46
4.4 Category of AHL-Nets with Instantiations 54

5 Composition of Algebraic High-Level Processes **61**
5.1 Composition of Algebraic High-Level Processes 61
5.2 Composition of Instantiations 68
5.3 Composition of AHL-Processes with Instantiations 71

6 Decomposition of Algebraic High-Level Processes **85**
6.1 Pushout Complements 85
6.2 Decomposition of AHL-Processes 89
6.3 Decomposition of AHL-Processes with Instantiations 92

7 Rule-based Transformation of Algebraic High-Level Processes **105**
7.1 Transformation of AHL-Processes 105
7.2 Transformation of AHL-Processes with Instantiations 113

8 Amalgamation of Algebraic High-Level Processes **120**
8.1 Amalgamation of Algebraic High-Level Processes 120
8.2 Amalgamation of Algebraic High-Level Processes with Instantiations 125

9 Conclusion: Summary of Results and Future Work **131**
9.1 Summary of Results 131
9.2 Ease of Use 133
9.3 Towards a Double Pushout Approach for AHL-Processes with Instantiations . 134
9.4 Process Grammars 136

A Example **138**

B Process Grammars for Low-Level Petri Nets **140**

C Detailed Proofs **146**
C.2 Algebraic High-Level Processes 146
C.3 Parallel and Sequential Composition and Independence of Algebraic High-Level Processes 156
C.4 Instantiations and Initializations 163
C.5 Composition of Algebraic High-Level Processes 192
C.6 Decomposition of Algebraic High-Level Processes 224
C.7 Rule-based Transformation of Algebraic High-Level Processes 250
C.8 Amalgamation of Algebraic High-Level Processes 259
C.9 Process Grammars for Low-Level Petri Nets 262

1 Introduction

Processes for low-level petri nets are essential to capture the non-sequential truly concurrent behaviour of low-level petri nets (see e.g. [GR83, Roz87, DMM89, Eng91, MMS97]). Since every high-level petri net N can be flattened to a low-level petri net $Flat(N)$ with the same firing behaviour as N, processes for high-level nets are often defined as processes of the low-level net which is obtained from flattening the high-level net. Since this leads to a loss of the high-level nature of the processes we use the process approach defined in [Ehr05, EHP+02] where high-level processes are based on a suitable notion of high-level occurrence nets which are defined independently of the flattening construction. Due to the so-called assignment conflicts the flattening of a high-level occurrence net does in general not lead to a low-level occurrence net. The essential idea is to generalize the concept of occurrence nets from the low-level to the high-level case. This means that the net structure of a high-level occurrence net has similar properties like a low-level occurrence net, i.e. unitarity, conflict freeness, acyclicity and finitarity.

High-level processes of this kind in contrast to low-level processes do not capture one concurrent computation but a set of different computations corresponding to different input parameters of the process. In fact, high-level processes can be considered to have a set of initial markings for the input places of the corresponding occurrence net, whereas there is only one implicit initial marking of the input places for low-level occurrence nets.

In [EHGP09] we introduced a composition construction for high-level processes. It allows a combination of sequential and parallel composition of high-level processes without loss of the process properties. In order to allow the modelling and analysis of more complex interactions of high-level processes the first main aim of this thesis is the definition and analysis of a rule-based transformation of high-level processes. For this purpose we extend the notion of the composition of high-level processes to non-sequential compositions and develop a suitable decomposition construction. The definition of both constructions as pushout in suitable categories of high-level processes allows to model the double pushout transformation of high-level processes as the combination of both constructions. This leads to a rule-based modification of high-level processes which is a powerful technique for the modelling and analysis of high-level processes.

While in the first aim we are considering different processes of one single high-level net, another main aim of this thesis is the composition and decomposition of processes of different high-level nets. This means an amalgamation of high-level processes analogously to the amalgamation of open low-level processes in [BCEH01]. Amalgamation means the composition and decomposition of high-level processes in correspondence to the composition and decomposition of the respective system nets of the processes leading to a compositional process semantics of algebraic high-level nets, i.e. the process semantics of one net can be expressed by the process semantics of its parts.

In [EHGP09] we extended the concept of high-level net processes with initial markings by a set of corresponding instantiations. An instantiation is a subnet of the flattening defining one concurrent computation of the process. The advantage is that we fix for a given initial marking a complete firing sequence where each transition fires exactly once. In order to allow also a compositional modelling of algebraic high-level processes with instantiations we extend the concepts mentioned above to modifications of algebraic high-level processes with instantiations.

The thesis is structured as follows. In Section 2 we review the notions of place/transition nets and algebraic high-level nets which are essential for our definition of high-level processes which are also presented in Section 2. Furthermore we give an overview on the well-known flattening and skeleton constructions of algebraic high-level nets. These constructions are the foundation for the concepts of instantiations introduced in [Ehr05] and algebraic high-level processes with instantiations introduced in [EHGP09].
Section 3 gives an overview on the results of [EHGP09]. It contains the parallel and sequential composition of algebraic high-level processes with instantiations of one high-level net. Moreover an equivalence on algebraic high-level processes with instantiations with the same input/output behaviour is presented which is then used for the concept of independence of sequential compositions. Independence of processes means that two processes can be sequentially composed in different orders such that the results are equivalent to each other.
In Section 4 we introduce the category of instantiations and the categories of algebraic high-level nets and processes with instantiations together with some basic properties and categorical constructions which are used in the subsequent sections.
In Section 5 we introduce a general version of the composition of algebraic high-level processes with instantiations which generalizes the parallel and sequential composition in 3. We show that the parallel and sequential composition can be seen as special cases of the general composition of algebraic high-level processes and that the composition constructions are pushout constructions in suitable categories.
In Section 6 we define a decomposition construction as inverse operation to the composition of algebraic high-level processes and algebraic high-level processes with instantiations. We show that the decomposition constructions are pushout complements (i.e. the inverse operation to the construction of pushouts) in suitable categories. Since in the case of algebraic high-level processes with instantiations these constructions may in general lead to ambiguous results we introduce further concepts in order to ensure unique decompositions of these processes.
The composition and decomposition as pushout and pushout complement constructions are used for the definition of a rule-based transformation for algebraic high-level processes with instantiations in Section 7. We introduce the concept of production rules for processes. The production rules can be used to specify parts of a process which are deleted and new parts which are added to the process. We give a characterization under which conditions a production rule can be applied to a process in the sense of the double pushout approach (see [EEPT06]) leading to a modified process.
In Section 8 we do also consider processes of different high-level nets. For given compositions and decompositions of high-level nets we define how to compose or decompose processes of these nets in a compositionally way leading to a process of the result net. We show that the defined constructions for high-level processes are inverse to each other leading to the fact that the process semantics of the composed net can be described by the process semantics of its single parts and vice versa.
Section 9 is the conclusion of the thesis. We give there an overview of the achieved results and of future work based on the results of this thesis.
All concepts in the thesis are illustrated by a running example of a simple intelligent alarm clock system introduced in Section 2. For a better readability of the thesis we often give only proof sketches in the main part and the full proofs can be found in Appendix C where the proofs are structured by the sections of the corresponding theorems, facts and corollaries (e.g. the proof for a theorem in Section 2 can be found in Appendix C.2).

2 Algebraic High-Level Processes

In this section we review the concept of algebraic high-level nets and we give a formal definition of high-level processes [Ehr05, EHP+02] based on high-level occurrence nets. Moreover we extend this definition by a suitable notation of instantiations for each initial marking according to [EHGP09] based on the flattening and skeleton constructions of high-level nets.

2.1 Place/Transition Nets and Algebraic High-Level Nets

As net formalism we use place/transition nets following the notation of "Petri nets are Monoids" in [MM90].

Definition 2.1 (Place/Transition Net)
A place/transition (P/T) net $N = (P, T, pre, post)$ consists of sets P and T of places and transitions respectively, and pre- and post domain functions $pre, post : T \to P^{\oplus}$ where $P^{\oplus}$ is the free commutative monoid over P.
A P/T-net morphism $f : N_1 \to N_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ satisfying

$$f_P^{\oplus} \circ pre_1 = pre_2 \circ f_T \text{ and } f_P^{\oplus} \circ post_1 = post_2 \circ f_T$$

where the extension $f_P^{\oplus} : P_1^{\oplus} \to P_2^{\oplus}$ of $f_P : P_1 \to P_2$ is defined by

$$f_P^{\oplus}(\sum_{i=1}^{n} k_i \cdot p_i) = \sum_{i=1}^{n} k_i \cdot f_P(p_i)$$

. A P/T-net morphism $f = (f_P, f_T)$ is called injective if f_P and f_T are injective and is called isomorphism if f_P and f_T are bijective.
The category defined by P/T-nets and P/T-net morphisms is denoted by **PTNet** where the composition of P/T-net morphisms is defined componentwise for places and transitions. $\triangle$

Because the notion of pushouts is essential for our main results we state the construction of pushouts in the category **PTNet** of place/transition nets. Intuitively a pushout means the gluing of two nets along an interface net. The construction is based on the pushouts for the sets of transitions and places in the category **SET**.

Definition 2.2 (Pushout of Sets)
In the category **SET** of sets and functions the pushout object D for given $f_1 : A \to B$ and $f_2 : A \to C$ is defined by the quotient set

$$D = B \uplus C/ \equiv$$

short $D = B \circ_A C$, where $B \uplus C$ is the disjoint union of B and C and $\equiv$ is the equivalence relation generated by $f_1(a) \equiv f_2(a)$ for all $a \in A$.
Moreover we define morphisms $g_1 : B \to D$ and $g_2 : C \to D$ with $g_1(b) = [b]_{\equiv}$ and $g_2(c) = [c]_{\equiv}$. Then diagram (1) is a pushout diagram in **SET**.

$$\begin{tikzcd} A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & B \arrow[d, "g_1"] \\ C \arrow[r, "g_2"'] & D \end{tikzcd}$$

△

In fact, D can be interpreted as the gluing of B and C along A: Starting with the disjoint union $B \uplus C$ we glue together the elements $f_1(a) \in B$ and $f_2(a) \in C$ for each $a \in A$.
The pushout object N_3 in the category **PTNet** is constructed componentwise for transitions and places in **SET** with corresponding pre- and post domain functions. For given P/T-net morphisms $f_1 : N_0 \to N_1$ and $f_2 : N_0 \to N_2$ the pushout of f_1 and f_2 is defined by the pushout diagram (PO) in **PTNet** and is denoted by $N_3 = N_1 \circ_{(N_0,f_1,f_2)} N_2$. For details we refer to [EEPT06].

Definition 2.3 (Pushout of Place/Transition Nets)
Given P/T-net morphisms $f_1 : N_0 \to N_1$ and $f_2 : N_0 \to N_2$ then the pushout diagram (1) and the pushout object N_3 in the category **PTNet**, written $N_3 = N_1 \circ_{(N_0,f_1,f_2)} N_2$, with $N_x = (P_x, T_x, pre_x, post_x)$ for $x = 0, 1, 2, 3$ is constructed as follows:

- $T_3 = T_1 \circ_{T_0} T_2$ with $f'_{1,T}$ and $f'_{2,T}$ as pushout (2) of $f_{1,T}$ and $f_{2,T}$ in **SET**.
- $P_3 = P_1 \circ_{P_0} P_2$ with $f'_{1,P}$ and $f'_{2,P}$ as pushout (3) of $f_{1,P}$ and $f_{2,P}$ in **SET**
- $pre_3(t) = \begin{cases} [pre_1(t_1)] & ;\text{ if } f'_{1,T}(t_1) = t \\ [pre_2(t_2)] & ;\text{ if } f'_{2,T}(t_2) = t \end{cases}$
- $post_3(t) = \begin{cases} [post_1(t_1)] & ;\text{ if } f'_{1,T}(t_1) = t \\ [post_2(t_2)] & ;\text{ if } f'_{2,T}(t_2) = t \end{cases}$

$$\begin{tikzcd} N_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & N_1 \arrow[d, "f'_1"] & T_0 \arrow[r, "f_{1,T}"] \arrow[d, "f_{2,T}"'] \arrow[dr, phantom, "(2)"] & T_1 \arrow[d, "f'_{1,T}"] & P_0 \arrow[r, "f_{1,P}"] \arrow[d, "f_{2,P}"'] \arrow[dr, phantom, "(3)"] & P_1 \arrow[d, "f'_{1,P}"] \\ N_2 \arrow[r, "f'_2"'] & N_3 & T_2 \arrow[r, "f'_{2,T}"'] & T_3 & P_2 \arrow[r, "f'_{2,P}"'] & P_3 \end{tikzcd}$$

△

Example 2.4 (Pushout of Place/Transition Nets)
A pushout of P/T-nets is shown in Figure 8 on page 28. The two nets L_{init_1} and L_{init_2} are glued together via inclusions at the place $(0, cp_2)$. △

In the following we review the definition of AHL-nets from [Ehr05, EHP+02].

Definition 2.5 (Algebraic High-Level Net)
An algebraic high-level (AHL) net $AN = (SP, P, T, pre, post, cond, type, A)$ consists of

- an algebraic specification $SP = (\Sigma, E; X)$ with signature $\Sigma = (S, OP)$, equations E, and additional variables X;
- a set of places P and a set of transitions T;
- pre- and post domain functions $pre, post : T \to (T_\Sigma(X) \otimes P)^\oplus$;
- firing conditions $cond : T \to \mathcal{P}_{fin}(Eqns(\Sigma; X))$;
- a type of places $type : P \to S$ and
- a (Σ, E)-algebra A

where the signature $\Sigma = (S, OP)$ consists of sorts S and operation symbols OP, $T_\Sigma(X)$ is the set of terms with variables over X,

$$(T_\Sigma(X) \otimes P) = \{(term, p) | term \in T_\Sigma(X)_{type(p)}, p \in P\}$$

and $Eqns(\Sigma; X)$ are all equations over the signature Σ with variables X.
An AHL-net morphism $f : AN_1 \to AN_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ satisfying

(1) $(id \otimes f_P)^\oplus \circ pre_1 = pre_2 \circ f_T$ and $(id \otimes f_P)^\oplus \circ post_1 = post_2 \circ f_T$,
(2) $cond_2 \circ f_T = cond_1$ and
(3) $type_2 \circ f_P = type_1$.

The category defined by AHL-nets and AHL-net morphisms is denoted by **AHLNet** where the composition of AHL-net morphisms is defined componentwise for places and transitions. △

Example 2.6 (AHL-Net)
Fig. 1 shows an AHL-net $Alarm = (SP - Alarm, P, T, pre, post, cond, type, A)$ which models a part of the control of an intelligent radio alarm clock which is connected to the heater, the light, and the radio in a room. The user of the alarm clock can define different time phases for specific actions like turning on the heat, the light and the radio and to increase the volume step-wise for a soft awakening.
An action can only be performed if the scheduled phase for that action is the current phase of the system which is represented as a token on the place *cp*. The current phase can be incremented by a clock by firing the transition *next phase*.
The specification *SP-Alarm* and the *SP-Alarm*-algebra A can be found in Appendix A. We use them as the specification and algebra in all AHL-nets which serve as examples in this thesis. △

In the following we omit the indices of functions f_P and f_T if no confusion arises.
The construction of pushouts in the category **AHLNet** of AHL-nets with fixed specification SP and algebra A can be analogously defined to the construction of pushouts in **PTNet** described above (for details see [EEPT06]).

Definition 2.7 (Pushout of AHL-Nets)
Given AHL-net morphisms $f_1 : N_0 \to N_1$ and $f_2 : N_0 \to N_2$ then the pushout diagram (1) and the pushout object N_3 in the category **AHLNet**, written $N_3 = N_1 \circ_{(N_0, f_1, f_2)} N_2$, with $N_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 0, 1, 2, 3$ is constructed as follows:

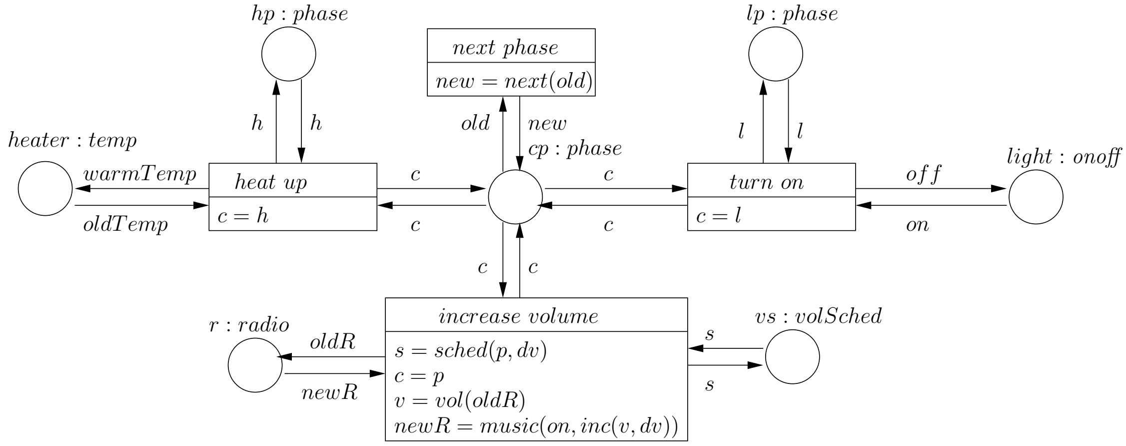


Figure 1: AHL-net *Alarm*

- $T_3 = T_1 \circ_{T_0} T_2$ with $f'_{1,T}$ and $f'_{2,T}$ as pushout (2) of $f_{1,T}$ and $f_{2,T}$ in **SET**.
- $P_3 = P_1 \circ_{P_0} P_2$ with $f'_{1,P}$ and $f'_{2,P}$ as pushout (3) of $f_{1,P}$ and $f_{2,P}$ in **SET**
- $pre_3(t) = \begin{cases} [pre_1(t_1)] & ;\text{ if } f'_{1,T}(t_1) = t \\ [pre_2(t_2)] & ;\text{ if } f'_{2,T}(t_2) = t \end{cases}$
- $post_3(t) = \begin{cases} [post_1(t_1)] & ;\text{ if } f'_{1,T}(t_1) = t \\ [post_2(t_2)] & ;\text{ if } f'_{2,T}(t_2) = t \end{cases}$
- $cond_3(t) = \begin{cases} [cond_1(t_1)] & ;\text{ if } f'_{1,T}(t_1) = t \\ [cond_2(t_2)] & ;\text{ if } f'_{2,T}(t_2) = t \end{cases}$
- $type_3(p) = \begin{cases} [type_1(p_1)] & ;\text{ if } f'_{1,P}(p_1) = p \\ [type_2(p_2)] & ;\text{ if } f'_{2,P}(p_2) = p \end{cases}$

$$\begin{tikzcd} N_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\mathbf{(1)}"] & N_1 \arrow[d, "f'_1"] & T_0 \arrow[r, "f_{1,T}"] \arrow[d, "f_{2,T}"'] \arrow[dr, phantom, "\mathbf{(2)}"] & T_1 \arrow[d, "f'_{1,T}"] & P_0 \arrow[r, "f_{1,P}"] \arrow[d, "f_{2,P}"'] \arrow[dr, phantom, "\mathbf{(3)}"] & P_1 \arrow[d, "f'_{1,P}"] \\ N_2 \arrow[r, "f'_2"'] & N_3 & T_2 \arrow[r, "f'_{2,T}"'] & T_3 & P_2 \arrow[r, "f'_{2,P}"'] & P_3 \end{tikzcd}$$

△

Example 2.8 (Pushout of AHL-Nets)
A pushout of AHL-nets is shown in Figure 7 on page 26. The two AHL-nets K_1 and K_2 are glued together via inclusions at the place cp_2. △

Another important construction is the pullback which is dual to the construction of pushouts.

Definition 2.9 (Pullback of Sets)
In the category **SET** of sets and functions the pushout object A for given $g_1 : B \to D$ and $g_2 : C \to D$ is defined by the the subset A of the cartesian product of B and C with

$$A = \{(b, c) \in B \times C \mid g_1(b) = g_2(c)\}$$

short $A = B \times_D C$.
Moreover we define morphisms $f_1 : A \to B$ and $f_2 : A \to C$ with $f_1(b, c) = b$ and $f_2(b, c) = c$. Then diagram (1) is a pullback diagram in **SET**.

$$\begin{tikzcd} A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & B \arrow[d, "g_1"] \\ C \arrow[r, "g_2"'] & D \end{tikzcd}$$

△

The set A can be interpreted as a common preimage of the functions g_1 and g_2.
Not only the pushout but also the pullback in the categories **PTNet** and **AHLNet** can be constructed componentwise in **SET** if at least one of the given morphisms is injective.

Definition 2.10 (Pullback of P/T-Nets Along Injective Morphisms)
Given P/T-net morphisms $g_1 : N_1 \to N_3$ and $g_2 : N_2 \to N_3$ where g_1 or g_2 is injective. Then the pullback diagram (1) and the pushout object N_0 in the category **PTNet**, written $N_0 = N_1 \times_{(N_3, g_1, g_2)} N_2$, with $N_x = (P_x, T_x, pre_x, post_x)$ for $x = 0, 1, 2, 3$ is constructed as follows:

- $T_0 = T_1 \times_{T_3} T_2$ with $f_{1,T}$ and $f_{2,T}$ as pullback (2) of $g_{1,T}$ and $g_{2,T}$ in **SET**.
- $P_0 = P_1 \times_{P_3} P_2$ with $f_{1,P}$ and $f_{2,P}$ as pullback (3) of $g_{1,P}$ and $g_{2,P}$ in **SET**
- $pre_3(t) = \begin{cases} [pre_1(t_1)] & ; \text{ if } f_{1,T}(t) = t_1 \\ [pre_2(t_2)] & ; \text{ if } f_{2,T}(t) = t_2 \end{cases}$
- $post_3(t) = \begin{cases} [post_1(t_1)] & ; \text{ if } f_{1,T}(t) = t_1 \\ [post_2(t_2)] & ; \text{ if } f_{2,T}(t) = t_2 \end{cases}$

$$\begin{tikzcd} N_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & N_1 \arrow[d, "g_1"] \\ N_2 \arrow[r, "g_2"'] & N_3 \end{tikzcd} \quad \begin{tikzcd} T_0 \arrow[r, "f_{1,T}"] \arrow[d, "f_{2,T}"'] \arrow[dr, phantom, "(\mathbf{2})"] & T_1 \arrow[d, "g_{1,T}"] \\ T_2 \arrow[r, "g_{2,T}"'] & T_3 \end{tikzcd} \quad \begin{tikzcd} P_0 \arrow[r, "f_{1,P}"] \arrow[d, "f_{2,P}"'] \arrow[dr, phantom, "(\mathbf{3})"] & P_1 \arrow[d, "g_{1,P}"] \\ P_2 \arrow[r, "g_{2,P}"'] & P_3 \end{tikzcd}$$

△

Example 2.11 (Pullback of Place/Transition Nets)
The pushout of P/T-nets shown in Figure 8 on page 28 is also a pullback. The place $(0, cp_2)$ is the only element in the nets L_{init_1} and L_{init_2} which is mapped via inclusions to the same element in the net L_{init}. △

We can extend the definition of pullbacks in **PTNet** to the one for pullbacks in **AHLNet** analogously to the way as the definition of pushouts in **PTNet** is extended to the definition of pushouts in **AHLNet**.

2.2 Algebraic High-Level Processes

Now we introduce high-level occurrence nets and processes according to [Ehr05, EHP+02]. The net structure of a high-level occurrence net has similar properties like a low-level occurrence net, but it captures a set of different concurrent computation due to different initial markings. In fact, high-level occurrence nets can be considered to have a set of initial markings for the input places, whereas there is only one implicit initial marking of the input places for low-level occurrence nets.

Definition 2.12 (AHL-Occurrence Net)
An AHL-occurrence net K is an AHL -net $K = (SP, P, T, pre, post, cond, type, A)$ such that for all $t \in T$ with $pre(t) = \sum_{i=1}^{n}(term_i, p_i)$ and notation $\bullet t = \{p_1, \dots, p_n\}$ and similarly $t\bullet$ we have

1. (*Unarity*): $\bullet t, t\bullet$ are sets rather than multisets for all $t \in T$, i.e. for $\bullet t$ the places $p_1 \dots p_n$ are pairwise distinct. Hence $| \bullet t| = n$ and the arc from p_i to t has a unary arc-inscription $term_i$.
2. (*No Forward Conflicts*): $\bullet t \cap \bullet t' = \emptyset$ for all $t, t' \in T, t \neq t'$
3. (*No Backward Conflicts*): $t \bullet \cap t' \bullet = \emptyset$ for all $t, t' \in T, t \neq t'$
4. (*Partial Order*): the causal relation $<\subseteq (P \times T) \cup (T \times P)$ defined by the transitive closure of $\{(p, t) \in P \times T \mid p \in \bullet t\} \cup \{(t, p) \in T \times P \mid p \in t\bullet\}$ is a finitary strict partial order, i.e. the partial order is irreflexive and for each element in the partial order the set of its predecessors is finite.

AHL-occurrence nets together with AHL-net morphisms between AHL-occurrence nets form the full subcategory **AHLONet** $\subseteq$ **AHLNet**. △

Example 2.13 (AHL-Occurrence net)
The AHL-net *Alarm* in Fig. 1 is not an AHL-occurrence net because it has for example a forward conflict at the place *cp*. Fig. 2 shows an AHL occurrence-net $K_{Heat/Light}$. △

Fact 2.14 (AHL-Morphisms Reflect AHL-Occurrence Nets)
Given two AHL-nets K_1, K_2 together with an AHL-morphism $f : K_1 \to K_2$.
If K_2 is an AHL-occurrence net then also K_1 is an AHL-occurrence net.

Proof sketch. Let us assume that K_1 is not an AHL-occurrence net. Then K_1 is not unary, it has a conflict or the causal relation is not a finitary strict partial order. Since all of these conditions depend on the pre and post conditions of transitions in the net K_1 and AHL-morphisms preserve pre and post conditions this implies that also K_2 is not an AHL-occurrence net which is a contradiction.
For a detailed proof see Detailed Proof C.1 in the appendix. □

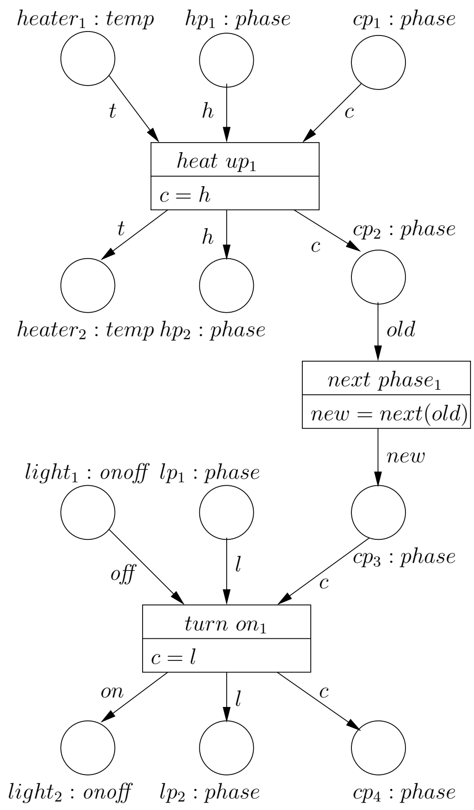


Figure 2: AHL-occurrence net $K_{Heat/Light}$

The notion of high-level net processes generalises the one of low-level net processes, where a P/T-process of a P/T-net N is a P/T-net morphism $p : K \to N$ and K is a low-level occurrence net, i.e. a net satisfying conditions 1.-4. in Def. 2.12.

Definition 2.15 (AHL-Process)
An AHL-process of an AHL-net AN is an AHL-net morphism $mp : K \to AN$ where K is an AHL-occurrence net. △

Example 2.16 (AHL-Process)
Let $mp : K_{Heat/Light} \to Alarm$ be an AHL-morphism which maps every place and every transition to the place respectively transition with the corresponding name (i.e. the same name but without the index). Then mp is an AHL-process of the AHL-net *Alarm*. The process mp represents the firing of the transitions *heat up*, *next phase* and *turn on* in that order. △

Morphisms between AHL-processes should be compatible with the process morphisms. The required compatibility corresponds to the definition of morphisms in a slice category.

Definition 2.17 (Slice Category)
Given a Category $\mathbf{C}$ and an Object $X \in Ob_{\mathbf{C}}$. The slice category $\mathbf{C} \setminus X$ is defined in the following way:

$$Ob_{\mathbf{C}\setminus X} = \{a : A \to X \mid A \in Ob_{\mathbf{C}}, a \in Mor_{\mathbf{C}}(A, X)\}$$

$$Mor_{\mathbf{C}\setminus X}(a : A \to X, b : B \to X) = \{f \in Mor_{\mathbf{C}}(A, B) \mid \text{ diagram (1) commutes }\}$$

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "a"'] & \arrow[d, phantom, "(1)"] & B \arrow[dl, "b"] \\ & X & \end{tikzcd}$$

$$g \circ_{\mathbf{C}\setminus X} f = g \circ_{\mathbf{C}} f$$

$$id_{f:A\to X} = id_A$$

△

Well-definedness. The well-definedness and associativity of the composition $\circ_{\mathbf{C}\setminus X}$ follows from the well-definedness respectively associativity of the composition $\circ_{\mathbf{C}}$.
The identitiy is well-defined because for $a : A \to X$ there is $a \circ id_A = a$. The neutrality of $id_{f:A\to X}$ follows from the neutrality of id_A. □

Definition 2.18 (Category of AHL-Processes)
Given an AHL-net AN. The category $\mathbf{AHLProc(AN)}$ of AHL-processes of AN is defined as the full subcategory of the slice category $\mathbf{AHLNet} \setminus AN$ where for objects $mp : K \to AN$ the net K is an AHL-occurrence net.
The objects of this category are AHL-process morphisms $mp : K \to AN$ and the morphisms of the category are AHL-net morphisms $f : K_1 \to K_2$ such that diagram (1) commutes.

$$\begin{tikzcd} K_1 \arrow[rr, "f"] \arrow[dr, "mp_1"'] & (1) & K_2 \arrow[dl, "mp_2"] \\ & AN & \end{tikzcd}$$

△

AHL-processes are not preserved by AHL-morphisms, i.e. for a given AHL-morphism $f : K_1 \to K_2$ where K_1 together with $mp_1 : K_1 \to AN$ is an AHL-process there does not necessarily exist a process morphism $mp_2 : K_1 \to K_2$. However, AHL-processes are reflected by AHL-morphisms.

Fact 2.19 (AHL-Morphisms Reflect Processes)
Given AHL-nets AN, K_1 and K_2 together with an AHL-morphism $f : K_1 \to K_2$.
If there is an AHL-morphism $mp_2 : K_2 \to AN$ such that K_2 together with mp_2 is a process of AN then there exists an AHL-morphism $mp_1 : K_1 \to AN$ such that K_1 together with mp_1 is a process of AN and f is an $\mathbf{AHLProc(AN)}$-morphism.

Proof. Let K_2 together with mp_2 be a process of AN. Since AHL-morphisms reflect AHL-occurrence nets and f is an AHL-morphism the net K_1 is an AHL-occurrence net.
Due to the well-defined composition of AHL-morphisms we obtain a process morphism

$$mp_1 := mp_2 \circ f : K_1 \to AN$$

leading to the fact that K_1 together with mp_1 is an AHL-process of AN and f is an $\mathbf{AHLProc(AN)}$-morphism. □

2.3 Flattening and Skeleton Construction

It is possible to construct for a given high-level net a low-level net with the same firing behaviour. The firing behaviour of an AHL-net is similar to the one of a Place/Transition-Net. The difference is that tokens on places in a high-level net can be data elements of an algebra and the transitions have sets of firing conditions. In order to enable a transition t there has be an assignment v of the variables in the conditions of t which satisfies the conditions of t and there have to be corresponding data elements on the places in the pre domain of the transition. The follower marking is then computed with the assignment v.

Definition 2.20 (Firing Behaviour of AHL-Nets)
A marking of an AHL-net AN is given by $M \in CP^{\oplus}$ where

$$CP = (A \otimes P) = \{(a, p) | a \in A_{type(p)}, p \in P\}$$

The set of variables $Var(t) \subseteq X$ of a transition $t \in T$ are the variables of the net inscriptions in $pre(t), post(t)$ and $cond(t)$.
Let $v : Var(t) \to A$ be a variable assignment with term evaluation $\overline{v} : T_\Sigma(Var(t)) \to A$, then (t, v) is a consistent transition assignment iff $cond_{AN}(t)$ is validated in A under v. The set CT of consistent transition assignments is defined by

$$CT = \{(t, v) | (t, v) \text{ consistent transition assignment}\}$$

A transition $t \in T$ is enabled in M under v iff

$$(t, v) \in CT \text{ and } pre_A(t, v) \leq M$$

where $pre_A : CT \to CP^{\oplus}$ defined by

$$pre_A(t, v) = \hat{v}(pre(t)) \in (A \otimes P)^{\oplus}$$

and

$$\hat{v} : (T_{\Sigma}(Var(t)) \otimes P)^{\oplus} \to (A \otimes P)^{\oplus}$$

is the obvious extension of $\overline{v}$ to terms and places (similar $post_A : CT \to CP^{\oplus}$). Then the follower marking is computed by

$$M' = M \ominus pre_A(t, v) \oplus post_A(t, v)$$

$\triangle$

The flattening of an AHL-net AN is a place/transition net $Flat(AN)$ where the data elements of the type of a place in the high-level net and consistent assignments are encrypted in the names of places and transitions in the flattening. This leads to a P/T-net with the same firing behaviour as AN, i.e. the marking graph of $Flat(AN)$ is isomorphic to the marking graph of AN. This means that for every firing sequence in AN there is a corresponding firing sequence in $Flat(AN)$ and vice versa.
The flattening of an AHL-morphism $f : N_1 \to N_2$ is a P/T-net morphism $Flat(f)$ which maps the places and transitions in a similar way as f does where the data elements and assignments encrypted in the names of the places respectively transitions are mapped identically.

Definition 2.21 (Flattening)
Given AHL-net AN as above then the flattening of AN is a P/T-net

$$Flat(AN) = N = (CP, CT, pre_A, post_A)$$

with

- $CP = A \otimes P = \{(a, p) | a \in A_{type(p)}, p \in P\}$,
- $CT = \{(t, v) | t \in T, v : Var(t) \to A \text{ s.t. } cond(t) \text{ valid in } A \text{ under } v \}$ and
- pre_A and $post_A$ as defined in Def. 2.20.

Given an AHL-net morphism $f : AN_1 \to AN_2$ by $f = (f_P, f_T)$ then

$$Flat(f) = (id_A \otimes f_P : CP_1 \to CP_2, f_C : CT_1 \to CT_2)$$

is given by

$$id_A \otimes f_P(a, p) = (a, f_p(p))$$

and

$$f_C(t, v) = (f_T(t), v)$$

$\triangle$

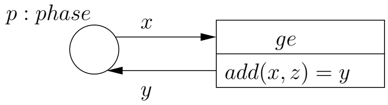


Figure 3: AHL-net GE

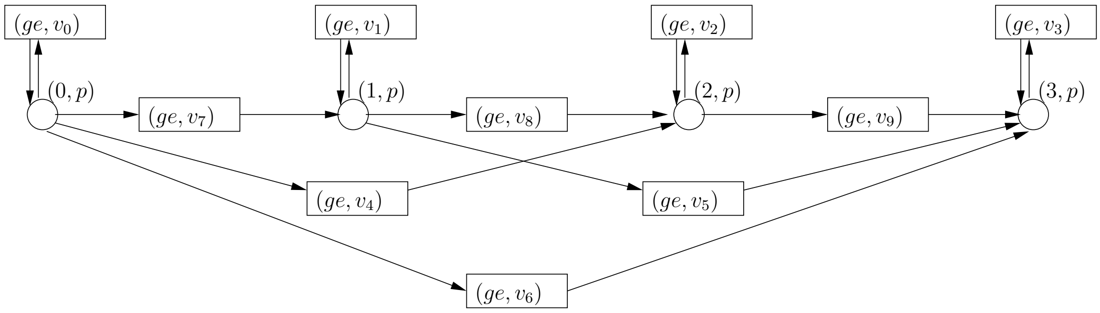


Figure 4: Part of the Flattening $Flat(GE)$ of the Net GE

Example 2.22 (Flattening)
Consider the AHL-net GE shown in Figure 3. The net contains a place p of the type *phase* and a transition ge which takes a phase $\overline{v}(x)$ from the place p and puts another phase on p which is greater than or equal to the phase $\overline{v}(x)$.
Figure 4 shows a part of the infinite flatteing $Flat(GE)$ of the net GE restricted to places for phases 0 to 3. In the complete flattening there are places (a, p) for all $a \in \mathbb{N}$. For each pair of places $(a_1, p), (a_2, p) \in Flat(GE)$ with $a_1 \leq a_2$ there is a transition $(t, v) \in Flat(GE)$ with $v(x) = a_1$ and $v(y) = a_2$ and there is $pre_{Flat(GE)}(t, v) = (a_1, p)$ and $post_{Flat(GE)}(t, v) = (a_2, p)$.

$\triangle$

The skeleton of an AHL-net AN is a place/transition net $Skel(AN)$ with the same structure as AN, i.e. the construction $Skel$ forgets all high-level elements like arc inscription terms, firing conditions and place types of the net AN. This means that for every firing sequence in AN there is also a corresponding firing sequence in $Skel(AN)$ but not vice versa.
The skeleton of an AHL-morphism $f : N_1 \to N_2$ is a P/T-net morphism $Skel(f)$ which maps places and transitions exactly in the same way as f does.

Definition 2.23 (Skeleton)
Given an AHL-net AN as above then the skeleton of AN is a P/T-net

$$Skel(AN) = (P, T, pre_S, post_S)$$

with

$$pre_S(t) = \sum_{i=1}^{n} p_i \text{ for } pre(t) = \sum_{i=1}^{n} (term_i, p_i)$$

and similar for $post_S : T \to P^{\oplus}$.

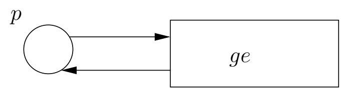


Figure 5: Skeleton $Skel(GE)$ of the Net GE

Given an AHL-net morphism $f : AN_1 \to AN_2$ by $f = (f_P, f_T)$ then

$$Skel(f) = f = (f_P : P_1 \to P_2, f_T : T_1 \to T_2)$$

$\triangle$

Example 2.24
Figure 5 shows the skeleton $Skel(GE)$ of the net GE depicted in Figure 3.
It is a P/T-net with the same structure and naming as the net AHL-net GE but the types, firing conditions and arc descriptions as well as the specification and algebra are omitted. $\triangle$

Remark 2.25. The flattening construction defined in Def. 2.21 and the skeleton construction defined in Def. 2.23 are well-defined and can be turned into a functor $Flat$: **AHLNet** $\to$ **PTNet** and a functor $Skel$: **AHLNet** $\to$ **PTNet** which preserve pushouts, i.e. given the pushout (1) in **AHLNet** then there are corresponding pushouts (2) and (3) in **PTNet**. Moreover we have for each AN a projection $proj(AN) : Flat(AN) \to Skel(AN)$ leading to a natural transformation $proj : Flat \to Skel$.

$$\begin{tikzcd}
AN_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & AN_1 \arrow[d, "f_1'"] \\
AN_2 \arrow[r, "f_2'"'] & AN_3
\end{tikzcd}$$

$$\begin{tikzcd}
Flat(AN_0) \arrow[r, "Flat(f_1)"] \arrow[d, "Flat(f_2)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(AN_1) \arrow[d, "Flat(f_1')"] \\
Flat(AN_2) \arrow[r, "Flat(f_2')"'] & Flat(AN_3)
\end{tikzcd}
\qquad
\begin{tikzcd}
Skel(AN_0) \arrow[r, "Skel(f_1)"] \arrow[d, "Skel(f_2)"'] \arrow[dr, phantom, "(\mathbf{3})"] & Skel(AN_1) \arrow[d, "Skel(f_1')"] \\
Skel(AN_2) \arrow[r, "Skel(f_2')"'] & Skel(AN_3)
\end{tikzcd}$$

Theorem 2.26 ($Flat$ is Functor)
The construction $Flat$: **AHLNet** $\to$ **PTNet** as defined in Def. 2.21 is a functor.

Proof sketch. Since $pre_A, post_A : CT \to CP^{\oplus}$ are well-defined functions the flattening of AHL-nets is a well-defined construction. The flattening of AHL-morphisms is well-defined because the flattening of a high-level net encrypts the firing behaviour of the net into the places and transitions of the flattening and AHL-morphisms preserve the firing behaviour of AHL-nets.
Furthermore there is $Flat(f) = id_{Flat(AN)}$ for $f = id_{AN}$ and the flattening of morphisms is compositional.
For a detailed proof see Detailed Proof C.2 in the appendix. $\square$

As mentioned above the functor *Flat* preserves pushouts.

Theorem 2.27 (*Flat* Preserves Pushouts)
Given Pushout (1) in **AHLNet** then (2) is Pushout in **PTNet**.

$$\begin{tikzcd} AN_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & AN_1 \arrow[d, "g_1"] \\ AN_2 \arrow[r, "g_2"'] & AN_3 \end{tikzcd} \qquad \begin{tikzcd} Flat(AN_0) \arrow[r, "Flat(f_1)"] \arrow[d, "Flat(f_2)"'] \arrow[dr, phantom, "(2)"] & Flat(AN_1) \arrow[d, "Flat(g_1)"] \\ Flat(AN_2) \arrow[r, "Flat(g_2)"'] & Flat(AN_3) \end{tikzcd}$$

Proof sketch. Pushouts in **AHLNet** and **PTNet** can be constructed componentwise for places and transitions in the category **SET**.
The places in the flattening are a subset of the cartesian product $A \times \_$ which preserves pushouts. Analogously the set of transitions in the flattening is also a product construction.
For a detailed proof see Detailed Proof C.3 in the appendix. □

Moreover the functor *Flat* preserves monomorphisms.

Theorem 2.28 (*Flat* Preserves Monomorphisms)
Given $f : AN_1 \to AN_2$ monomorphism in **AHLNet**, then $Flat(f) : Flat(AN_1) \to Flat(AN_2)$ is monomorphism in **PTNet**.

Proof. Since monomorphisms in **AHLNet** and **PTNet** are the injective morphisms, it suffices to show: If f is injective, $Flat(f)$ is injective.

Let us assume that f is injective. That means f_P and f_T are injective functions.
Let $Flat(f)_P(a_1, p_1) = Flat(f)_P(a_2, p_2)$. By Def. 2.21 we have:
$Flat(f)_P(a_1, p_1) = Flat(f)_P(a_2, p_2)$
$\Rightarrow (id_A \otimes f_P)(a_1, p_1) = (id_A \otimes f_P)(a_2, p_2)$
$\Rightarrow (a_1, f_P(p_1)) = (a_2, f_P(p_2))$
$\Rightarrow a_1 = a_2 \wedge f_P(p_1) = f_P(p_2)$
$\Rightarrow a_1 = a_2 \wedge p_1 = p_2$ because f_P is injective
$\Rightarrow (a_1, p_1) = (a_2, p_2)$
$\Rightarrow Flat(f)_P$ is injective.

Let $Flat(f)_T(t_1, v_1) = Flat(f)_T(t_2, v_2)$. By Def. 2.21 we have:
$Flat(f)_T(t_1, v_1) = Flat(f)_T(t_2, v_2)$
$\Rightarrow (f_T(t_1), v_1) = (f_T(t_2), v_2)$
$\Rightarrow f_T(t_1) = f_T(t_2) \wedge v_1 = v_2$
$\Rightarrow t_1 = t_2 \wedge v_1 = v_2$ because f_T is injective
$\Rightarrow (t_1, v_1) = (t_2, v_2)$
$\Rightarrow Flat(f)_T$ is injective.
$Flat(f)_P$ and $Flat(f)_T$ are injective implies that $Flat(f)$ is injective.

□

Theorem 2.29 (*Skel* is Functor)
The construction $Skel : \mathbf{AHLNet} \to \mathbf{PTNet}$ as defined in Def. 2.23 is a functor.

Proof. Given $f : AN_1 \to AN_2$ then $Skel(f) : Skel(AN_1) \to Skel(AN_2)$ is P/T-net morphism, i.e. (1) commutes componentwise.

$$\begin{tikzcd}
T_1 \arrow[r, shift left, "pre_{1,S}"] \arrow[r, shift right, "post_{1,S}"'] \arrow[d, "f_T"'] \arrow[dr, phantom, "(\mathbf{1})"] & P_1^{\oplus} \arrow[d, "f_P^{\oplus}"] \\
T_2 \arrow[r, shift left, "pre_{2,S}"] \arrow[r, shift right, "post_{2,S}"'] & P_2^{\oplus}
\end{tikzcd}$$

For $t_1 \in T_1$ and $pre_1(t_1) = \sum_{i=1}^{n}(term_i, p_i)$ we have
$f_P \circ pre_{1S}(t_1) = f_P^{\oplus}(\sum_{i=1}^{n} p_i) = \sum_{i=1}^{n} f_P(p_i)$
$pre_2 \circ f_T(t_1) = (id_A \otimes f_P)^{\oplus} \circ pre_1(t_1) = \sum_{i=1}^{n}(term_i, f_P(p_i))$ and hence
$pre_{2S} \circ f_T(t_1) = pre_{2S}(f_T(t_1)) = \sum_{i=1}^{n} f_P(p_i)$ which implies
$f_P \circ pre_{1S}(t_1) = pre_{2S} \circ f_T(t_1)$ and similar for $post_{1S}$ and $post_{2S}$. □

Not only the flattening functor but also the skeleton functor *Skel* preserves pushouts and monomorphisms.

Theorem 2.30 (*Skel* Preserves Pushouts)
Given Pushout (1) in **AHLNet** the (2) is Pushout in **PTNet**.

$$\begin{tikzcd}
AN_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & AN_1 \arrow[d, "g_1"] \\
AN_2 \arrow[r, "g_2"'] & AN_3
\end{tikzcd}
\qquad
\begin{tikzcd}
Skel(AN_0) \arrow[r, "Skel(f_1)"] \arrow[d, "Skel(f_2)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Skel(AN_1) \arrow[d, "Skel(g_1)"] \\
Skel(AN_2) \arrow[r, "Skel(g_2)"'] & Skel(AN_3)
\end{tikzcd}$$

Proof. Pushout (1) in **AHLNet** implies Pushouts (3) and (4) as in Theorem 2.27. But this implies that (2) is Pushout in **PTNet** , because Pushouts in **PTNet** are based on Pushouts (3) and (4) in **SET**. □

Theorem 2.31 (*Skel* Preserves Monomorphisms)
Given $f : AN_1 \to AN_2$ monomorphism in **AHLNet**, then $Skel(f) : Skel(AN_1) \to Skel(AN_2)$ is monomorphism in **PTNet**.

Proof. Similar to the proof that *Flat* preserves monomorphisms, we will show: If f is injective, $Skel(f)$ is injective.

Let us assume that $f : AN_1 \to AN_2$ is injective. That means f_P and f_T are injective functions.
Let $Skel(f)_P(p_1) = Skel(f)_P(p_2)$ for $p_1, p_2 \in P_{Skel(AN_1)}$. By Def. 2.23 we have:
$Skel(f)_P(p_1) = Skel(f)_P(p_2)$

$\Rightarrow f_P(p_1) = f_P(p_2)$
$\Rightarrow p_1 = p_2$ because f_P is injective. Hence $Skel(f)_P$ is injective.
Let $Skel(f)_T(t_1) = Skel(f)_T(t_2)$ for $t_1, t_2 \in T_{Skel(AN_1)}$. By Def. 2.23 we have:
$Skel(f)_T(t_1) = Skel(f)_T(t_2)$
$\Rightarrow f_T(t_1) = f_T(t_2)$
$\Rightarrow t_1 = t_2$ because f_T is injective. Hence $Skel(f)_T$ is injective. $\square$

There is a projection $proj : Flat \to Skel$ which forgets the data part of places and the assignments of transitions in the flattening. The projection $proj$ is a natural transformation.

Theorem 2.32 (Natural Transformation $proj : Flat \to Skel$)

$proj : Flat \to Skel$ defined for AHL-nets AN by
$proj(AN) : Flat(AN) \to Skel(AN)$ with
$proj(AN)_P(a, p) = p$ for $(a, p) \in CP = A \otimes P$ and
$proj(AN)_T(t, v) = t$ for $(t, v) \in CT$
is a natural transformation.

Proof. First we show that $proj(AN)$ is P/T-net morphism

$$\begin{tikzcd} Flat(AN) \arrow[d, "proj(AN)"'] \\ Skel(AN) \end{tikzcd} \qquad \begin{tikzcd} CT \arrow[r, shift left, "pre_A"] \arrow[r, shift right, "post_A"'] \arrow[d, "proj(AN)_T"'] \arrow[dr, phantom, "(\mathbf{1})"] & CP^{\oplus} \arrow[d, "proj(AN)_P^{\oplus}"] \\ T \arrow[r, shift left, "pre_S"] \arrow[r, shift right, "post_S"'] & P^{\oplus} \end{tikzcd}$$

Given $(t, v) \in CT$ with $pre(t) = \sum_{i=1}^{n}(term_i, p_i)$ we have
$proj(AN)_P^{\oplus} \circ pre_A(t, v) = proj(AN)_P^{\oplus}(\sum_{i=1}^{n}(\overline{v}(term_i), p_i)) = \sum_{i=1}^{n} p_i$
$pre_S \circ proj(AN)_T(t, v) = pre_S(t) = \sum_{i=1}^{n} p_i$
and similar for $post_A, post_S$.

In order to show that $proj$ is natural transformation let $f : AN_1 \to AN_2$ be an AHL-net morphism then we have to show commutativity of

$$\begin{tikzcd} Flat(AN_1) \arrow[r, "Flat(f)=(id_A \otimes f_P, f_C)"] \arrow[d, "proj(AN_1)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(AN_2) \arrow[d, "proj(AN_2)"] \\ Skel(AN_1) \arrow[r, "Skel(f)=(f_P, f_T)"'] & Skel(AN_2) \end{tikzcd}$$

This follows for $(a, p) \in A \otimes P_1$ from
$proj(AN_2)_P \circ (id_A \otimes f_P)(a, p) = f_P(p) = f_P \circ proj(AN_1)_P(a, p)$
and for $(t, v) \in CT_1$ from
$proj(AN_2)_T \circ f_C(t, v) = proj(AN_2)_T(f_T(t), v) = f_T(t) = f_T \circ proj(AN_1)_T(t, v)$. $\square$

Moreover the commutative diagram induced by the fact that $proj$ is a natural transformation is also a pullback in the category **PTNet**.

Theorem 2.33 (*proj* Induces Pullback)
Given AHL-nets K_1, K_2 and AHL-morphisms $f : K_1 \to K_2$. Then the diagram (1) is a Pullback in **PTNet**.

$$\begin{tikzcd}
X \arrow[dr, "g"] \arrow[drr, bend left, "g_2"] \arrow[ddr, bend right, "g_1"'] \arrow[dr, phantom, "\mathbf{(2)}"', shift right=4] & & \\
 & \mathit{Flat}(K_1) \arrow[r, "\mathit{Flat}(f)"] \arrow[d, "\mathit{proj}(K_1)"'] \arrow[u, phantom, "\mathbf{(3)}", shift left=8] & \mathit{Flat}(K_2) \arrow[d, "\mathit{proj}(K_2)"] \\
 & \mathit{Skel}(K_1) \arrow[r, "\mathit{Skel}(f)"'] \arrow[ur, phantom, "\mathbf{(1)}"] & \mathit{Skel}(K_2)
\end{tikzcd}$$

Proof sketch. Every element in $Flat(K_1)$ consists of a data part and a name part where the projection $proj(K_1)$ forgets the data part.
Given a P/T-net X together with morphisms $g_1 : X \to Skel(K_1)$ and $g_2 : X \to Flat(K_2)$ such that $Skel(f) \circ g_1 = proj(K_2) \circ g_2$ there exists exactly one morphism $g : X \to Flat(K_1)$ such that (2) and (3) commutes. The morphism g maps every element x in X to an element in $Flat(K_1)$ which has $g_1(x)$ as name part and the same data part as $g_2(x)$. Due to the fact that (1) commutes and $Flat(f)$ maps the data part of $Flat(K_1)$ identically every other morphism $g' : X \to Flat(K_1)$ with the required property maps all elements exactly in the same way as g does.
For a detailed proof see Detailed Proof C.4 in the appendix. □

2.4 Algebraic High-Level Processes with Instantiatons

Because in general there are different meaningful markings of an AHL-occurrence net K, we introduce a set of initial markings of the input places of K.

Definition 2.34 (AHL-Occurrence Net with Initial Markings)
An AHL-occurrence net with initial markings $(K, INIT)$ consists of an AHL-occurrence net K and a set $INIT$ of initial markings $init \in INIT$ of the input places $IN(K)$, where the input places of K are defined by

$$IN(K) = \{p \in P |\bullet p = \emptyset\}$$

and similarly the output places of K are defined by

$$OUT(K) = \{p \in P | p\bullet = \emptyset\}$$

△

Example 2.35 (AHL-Occurrence Net with Initial Markings)
Given the AHL-occurrence net $K_{Heat/Light}$ in Fig. 2 we define

$$INIT = \{init_1, init_2, init_3\}$$

where

$$init_1 = \{(0, heater_1), (0, hp_1), (0, cp_1), (0, light_1), (1, lp_1)\}$$

$$init_2 = \{(0, heater_1), (2, hp_1), (0, cp_1), (0, light_1), (1, lp_1)\}$$
$$init_3 = \{(0, heater_1), (0, hp_1), (0, cp_1), (1, light_1), (1, lp_1)\}$$

Then $(K_{Heat/Light}, INIT)$ is an AHL-occurrence net with initial markings.
In $init_1, init_2$ and $init_3$ the current phase in the beginning is phase 0 and the target temperature of the heater is 0°. In $init_1$ and $init_3$ the heating phase is phase 0 and $init_3$ it is phase 2. In all initializations the light phase is phase 1. The light is off in $init_1$ and $init_2$ and it is on in $init_3$. So the initial markings represent three different initial states of the process. $\triangle$

The following notion of instantiation defines one concurrent execution of a marked high-level occurrence net. In more detail an instantiation is a subnet of the flattening of the AHL-occurrence net corresponding to the initial marking. In [Ehr05, EHP+02] it is shown that for a marked AHL-occurrence net there exists a complete firing sequence if and only if there exists an instantiation which net structure is isomorphic to the AHL-occurrence net and has the initial marking of the AHL-occurrence net as input places. Note that in general we may have different instantiations for the same initial marking.
The flattening $Flat(AN)$ of an AHL-net AN results in a corresponding low-level net N, where the data type part (SP, A) and the firing behaviour of the AHL-net AN is encoded in the sets of places and transitions of N. Thus the flattening $Flat(AN)$ leads to an infinite P/T-net N if the algebra A is infinite. In contrast the skeleton $Skel(AN)$ of an AHL-net AN is a low-level net N' preserving the net structure of the AHL-net but dropping the net inscriptions. While there is a bijective correspondence between firing sequences of the AHL-net and firing sequences of its flattening, each firing of the AHL-net implies a firing of the skeleton, but not vice versa. For details we refer to [Ehr05, EHP+02].

Definition 2.36 (Instantiations of AHL-Occurrence Net)
Given an AHL-occurrence net with initial markings $(K, INIT)$ with $init \in INIT$. An instantiation L_{init} of $(K, init)$ is a low-level occurrence net $L_{init} \subseteq Flat(K)$ with input places $IN(L_{init}) = init$ such that the projection $proj : L_{init} \to Skel(K)$ defined by $proj_P(a, p) = p$ and $proj_T(t, v) = t$ is an isomorphism of low-level occurrence nets. $\triangle$

Example 2.37 (Instantiations of AHL-Occurrence Net)
Consider the AHL-occurrence net with initial markings $(K_{Heat/Light}, INIT)$ in Example 2.35. The P/T-net L_{init_1} depicted in Fig. 6 is an instantiation of $(K_{Heat/Light}, init_1)$.
There are no instantiations for the initial markings $init_2$ and $init_3$ because there exist no complete firing sequences with these initial markings. For $init_2$ the reason is that an assignment $v : Var(heat\ up_1) \to A$ with $v(c) = 0$ and $v(h) = 2$ the equation $c = h$ is not validated in A under v.
For $init_3$ the reason is that there exists no assignment $v : Var(turn\ on_1) \to A$ such that $\overline{v}(off) = 1$. $\triangle$

As mentioned above for a given initial marking of an AHL-occurrence net there exist in general more than one instantiation and thus different firing sequences resulting in different markings of the output places of the AHL-occurrence net. For this reason we introduce the new notion of AHL-occurrence nets and AHL-processes with instantiations, where we fix exactly one instantiation for a given initial marking, i.e. one concurrent execution of the marked AHL-occurrence net.

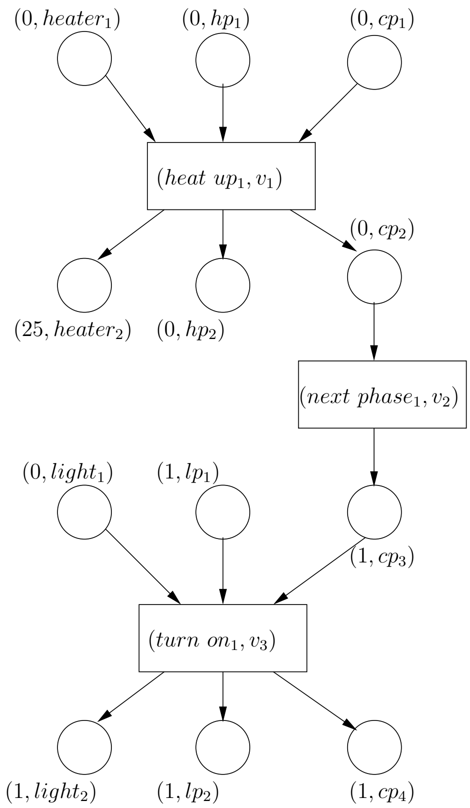


Figure 6: Instantiation L_{init_1} of $K_{Heat/Light}$

Definition 2.38 (AHL-Occurrence Net with Instantiations)
An AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ is an AHL-occurrence net with initial markings $(K, INIT)$ and a set INS of instantiations, such that for each $init \in INIT$ we have a distinguished instantiation $L_{init} \in INS$, i.e. $INS = \{L_{init} | init \in INIT\}$. An AHL-occurrence net with instantiations KI defines for each $init \in INIT$ with $IN(L_{init}) = init$ an output $out = OUT(L_{init})$ with $proj_P(out) = OUT(K)$. Let $EXIT$ be the set of all markings of the output places $OUT(K)$, then we obtain a function $inout : INIT \to EXIT$ by $inout(init) = OUT(L_{init})$. △

Example 2.39 (AHL-Occurrence Net with Instantiations)
Consider the AHL-occurrence net $K_{Heat/Light}$ and the set of initial markings

$$INIT = \{init_1\}$$

with

$$init_1 = \{(0, heater_1), (0, hp_1), (0, cp_1), (0, light_1), (1, lp_1)\}$$

Let $INS = \{L_{init_1}\}$ where L_{init_1} is the instantiation depicted in Fig. 6. Then $KI_{Heat/Light} = (K_{Heat/Light}, INIT, INS)$ is an AHL-occurrence net with instantiations. △

Definition 2.40 (AHL-Process with Instantiations)
An instantiated AHL-process of an AHL-net AN is an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ together with an AHL-net morphism $mp : K \to AN$. △

Example 2.41 (AHL-Process with Instantiations)
The AHL-occurrence net $KI_{Heat/Light}$ together with the morphism $mp : K_{Heat/Light} \to Alarm$ in Example 2.16 is an instantiated AHL-process. △

3 Parallel and Sequential Composition and Independence of Algebraic High-Level Processes

In this section we give a short overview of the results in [EHGP09] where a composition of High-Level processes has been defined. In section 5 we give a definition of a generalized composition of AHL-processes. Due to the fact that the composition defined in [EHGP09] is a special case of the generalized composition where the processes can only be composed in a sequential way we call it sequential composition whereas in [EHGP09] it is just called composition. Under certain conditions we call the sequential composition a parallel composition if the composition creates no connection between the composed components.
Moreover we define in the second part of this section an equivalence relation on AHL-processes with instantiations with the same input and output behaviour. Based on that equivalence a notion of independence of AHL-processes with instantiations is introduced where the independence of two processes means that they can be sequentially composed in any order leading to equivalent results.

3.1 Parallel and Sequential Composition of Algebraic High-Level Processes with Instantiations

Based on the construction of pushouts of low-level and high-level nets introduced in the previous section we define in this section the sequential composition of AHL-occurrence nets and AHL-processes with instantiations.
The sequential composition of two AHL-occurrence nets K_1 and K_2 is defined by merging some of the output places of K_1 with some of the input places of K_2, so that the result of the sequential composition definitely is an AHL-occurrence net. In general the composition of AHL-occurrence nets is not an AHL-occurrence net, because the result of gluing two high-level occurrence nets may contain forward and/or backward conflicts as well as the partial order might be violated. Thus we state suitable conditions, so that the sequential composition of AHL-occurrence nets leads to an AHL-occurrence net. Moreover we generalise this construction on the one hand to corresponding instantiations and on the other hand to AHL-net morphisms, so that the sequential composition of AHL-processes with instantiation leads to an AHL-process under suitable conditions.

Definition 3.1 (Sequential Composability of AHL-Occurrence Nets)
Given the AHL-occurrence nets $K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$ and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$ and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Then (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) iff $i_1(P_I) \subseteq OUT(K_1)$ and $i_2(P_I) \subseteq IN(K_2)$. △

Theorem 3.2 (Sequential Composition of AHL-Occurrence Nets)
Given the AHL- occurrence nets K_1, K_2 and I as above and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) is sequentially composable w.r.t. (I, i_1, i_2). Then the pushout diagram (PO) exists in the category **AHLNet** and the pushout object K, with $K = K_1 \circ_{(I,i_1,i_2)} K_2$, is an AHL-occurrence net and is called sequential composition of (K_1, K_2) w.r.t. (I, i_1, i_2).
Then diagram (PO) is called a composition diagram.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "\mathbf{(PO)}"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd}$$

Proof. Follows directly from Theorem 5.5 since by Fact 5.3 the sequential composability of AHL-occurrence nets is a special case of the composability of AHL-occurrence nets. □

Note that the order of K_1 and K_2 in the pair (K_1, K_2) and the result $K = K_1 \circ_{(I,i_1,i_2)} K_2$ is important because i_1 and i_2 relate output places of K_1 with input places K_2.

Example 3.3 (Sequential Composition of AHL-Occurrence Nets)
Fig. 7 shows an example of the sequential composition of AHL-occurrence nets where all of the morphisms i_1, i_2, i'_1 and i'_2 are inclusions.
The nets K_1 and K_2 are sequentially composable w.r.t. (I, i_1, i_2) because $i_1(cp_2) \in OUT(K_1)$ and $i_2(cp_2) \in IN(K_2)$. The composition result $K = K_1 \circ_{(I,i_1,i_2)} K_2$ is an AHL-occurrence net.
△

In the case of a sequential composition with an empty interface the pushout becomes the coproduct, i.e. no places are glued together. Hence the parallel composition of AHL-occurrence nets is a special case of the sequential composition.

Definition 3.4 (Parallel Composition of AHL-Occurrence Nets)
Given the AHL-occurrence nets $K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$ and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$ and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Let (K_1, K_2) be sequentially composable w.r.t. (I, i_1, i_2).
If $P_I = \emptyset$ then $K = K_1 \circ_{(I,i_1,i_2)} K_2$ is called a parallel composition. △

We have another special case if all output places of the first net are glued to all the input places in the second net.

Definition 3.5 (Strict Sequential Composition of AHL-Occurrence Nets)
Given the AHL-occurrence nets $K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$ and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$ and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Let (K_1, K_2) be sequentially composable w.r.t. (I, i_1, i_2).
If $i_1(P_I) = OUT(K_1)$ and $i_2(P_I) = IN(K_2)$ then $K = K_1 \circ_{(I,i_1,i_2)} K_2$ is called a strict sequential composition. △

Definition 3.6 (Sequential Composition of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2).
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$. Then (L_{init_1}, L_{init_2}) are sequentially composable

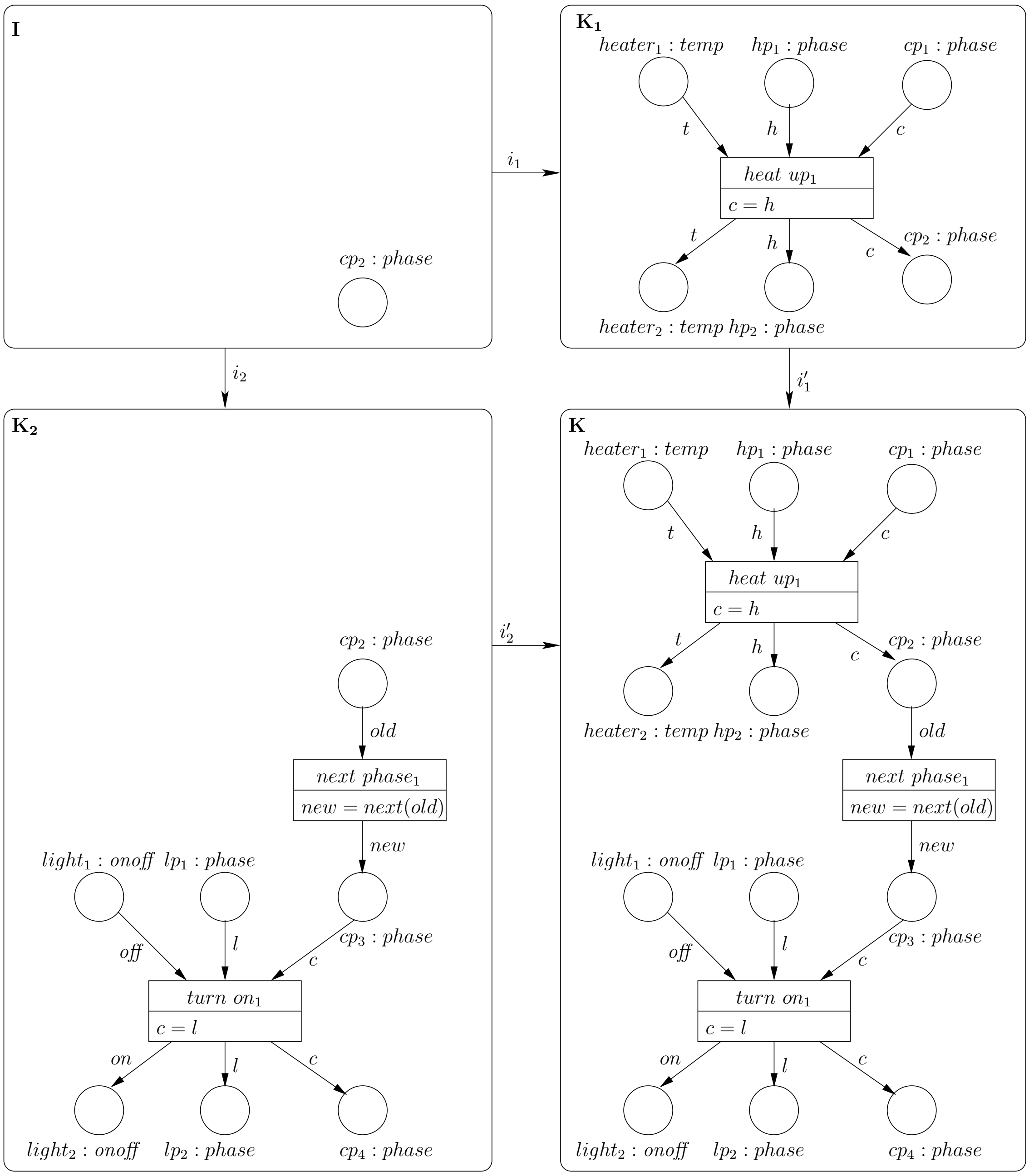


Figure 7: Sequential Composition of AHL-Occurrence Nets $K = K_1 \circ_{(I,i_1,i_2)} K_2$

w.r.t. (I, i_1, i_2) if for all $(a,p) \in A \otimes P_I : (a, i_1(p)) \in OUT(L_{init_1}) \Rightarrow (a, i_2(p)) \in IN(L_{init_2})$. From (I, i_1, i_2) we construct the induced instantiation interface (J, j_1, j_2) of (L_{init_1}, L_{init_2}) with $J = (P_J, T_J, pre_J, post_J)$ by

- $P_J = \{(a,p) | (a, i_1(p)) \in OUT(L_{init_1})\}$,
- $T_J = \emptyset$,
- $pre_J = post_J = \emptyset$ (the empty function) and
- $j_x : J \to L_{init_x}$ for $x = 1, 2$ defined by $j_{x,P} = id_A \otimes i_{x,P}$ and $j_{x,T} = \emptyset$.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i_1'"] \\ K_2 \arrow[r, "i_2'"'] & K \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(2)"] & L_{init_1} \arrow[d, "j_1'"] \\ L_{init_2} \arrow[r, "j_2'"'] & L_{init} \end{tikzcd} \qquad \begin{tikzcd} (J, I) \arrow[r, "(j_1, i_2)"] \arrow[d, "(j_2, i_2)"'] \arrow[dr, phantom, "(3)"] & (L_{init_1}, K_1) \arrow[d, "(j_1', i_1')"] \\ (L_{init_2}, K_2) \arrow[r, "(j_2', i_2')"'] & (L_{init}, K) \end{tikzcd}$$

The sequential composition of (L_{init_1}, L_{init_2}) w.r.t. the instantiation interface (J, j_1, j_2) induced by (I, i_1, i_2) is defined by the pushout diagram (2) in **PTNet** such that diagram (3) is a pushout in **INet**.
The sequential composition is denoted by $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ and diagram (2) is called a composition diagram of instantiations w.r.t. composition diagram (1). $\triangle$

Example 3.7 (Sequential Composition of Instantiations)
Consider the nets K_1 and K_2 in Fig. 7 and instantiations L_{init_1} of K_1 and L_{init_2} of K_2 in Fig. 8. Then L_{init_1} and L_{init_2} are composable w.r.t. (I, i_1, i_2) because there is

$$(0, i_1(cp_2)) \in OUT(L_{init_1}) \text{ and } (0, i_2(cp_2)) \in IN(L_{init_2})$$

The P/T-net J is the induced instantiation interface and the sequential composition of instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ is an instantiation of $K = K_1 \circ_{(I,i_1,i_2)} K_2$. $\triangle$

Theorem 3.8 (Sequential Composition of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2). Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations. Then the sequential composition of (KI_1, KI_2) w.r.t. (I, i_1, i_2) is defined by $KI = (K, INIT, INS)$ with

- $K = K_1 \circ_{(I,i_1,i_2)} K_2$,
- $INS = \{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} | L_{init_x} \in INS_x$ for $x = 1, 2, (L_{init_1}, L_{init_2})$ are sequentially composable w.r.t. $(I, i_1, i_2)\}$,
- and $INIT = \{IN(L_{init}) | L_{init} \in INS\}$

and $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ is an AHL-occurrence net with instantiations.

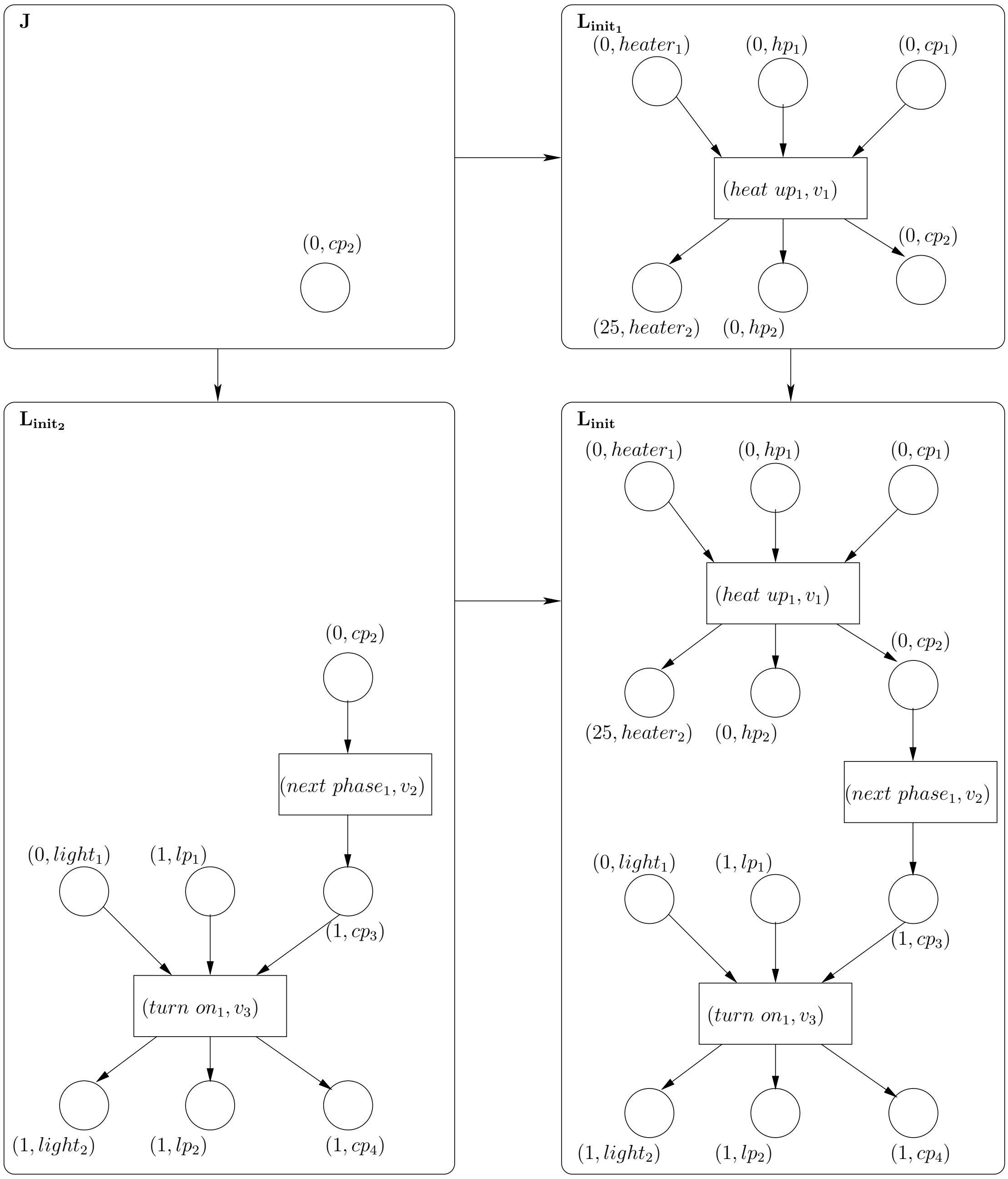


Figure 8: Sequential Composition of Instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$

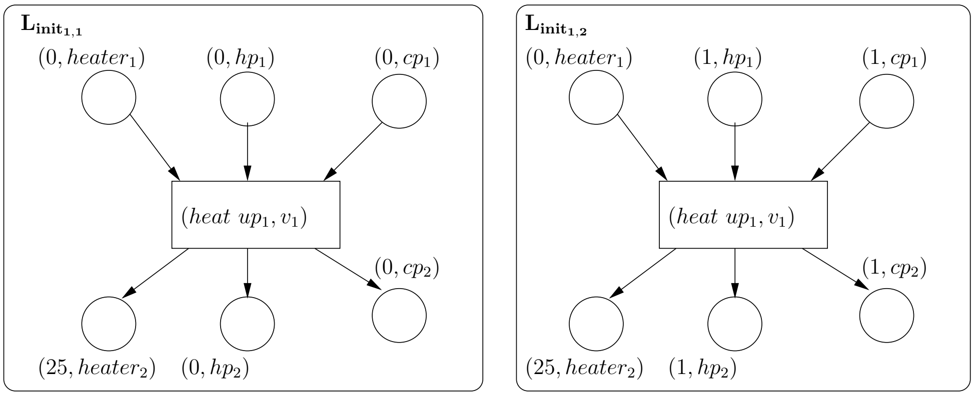


Figure 9: Instantiations of KI_1

Proof. From Fact 5.24 follows that (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2). By Fact 5.13 the sequentially composable instantiations of K_1 and K_2 are exactly the composable instantiations of K_1 and K_2 which by Fact 5.18 implies that the sequential composition of KI_1 and KI_2 w.r.t. (I, i_1, i_2) is exactly the composition of KI_1 and KI_2 w.r.t. (I, i_1, i_2) defined in Theorem 5.25. Hence by Theorem 5.25 the composition $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ is an AHL-occurrence net with instantiations. □

Example 3.9 (Sequential Composition of AHL-Occurrence Nets with Instantiations)
Consider AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ where K_1 and K_2 are the AHL-occurrence nets depicted in Fig. 7. Furthermore the set INS_1 contains the instanstiations $L_{init_{1,1}}$ and $L_{init_{1,2}}$ depicted in Fig. 9 and the set INS_2 contains the instantiations $L_{init_{2,1}}$ and $L_{init_{2,2}}$ depicted in Fig. 10. The sets $INIT_1$ and $INIT_2$ contain the corresponding initial markings, i.e. the input places of the nets in INS_1 and INS_2, respectively.
The sequential composition $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ is an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ where K is the composition $K = K_1 \circ_{(I,i_1,i_2)} K_2$ depicted in Fig. 7. The set INS contains only on instantiation L_{init} because there is only one pair $(L_{init_{1,1}}, L_{init_{2,1}})$ of composable instantiations.
The instantiation L_{init} is depicted in Fig. 8. The set $INIT$ contains one initial marking

$$init = \{(0, heater_1), (0, hp_1), (0, cp_1), (0, light_1), (1, lp_1)\}$$

with $init = IN(L_{init})$.

△

Definition 3.10 (Sequential Composability of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two instantiated AHL-processes of the AHL-net AN. Then (mp_1, mp_2) are sequentially composable w.r.t. (I, i_1, i_2) if

1. (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) and

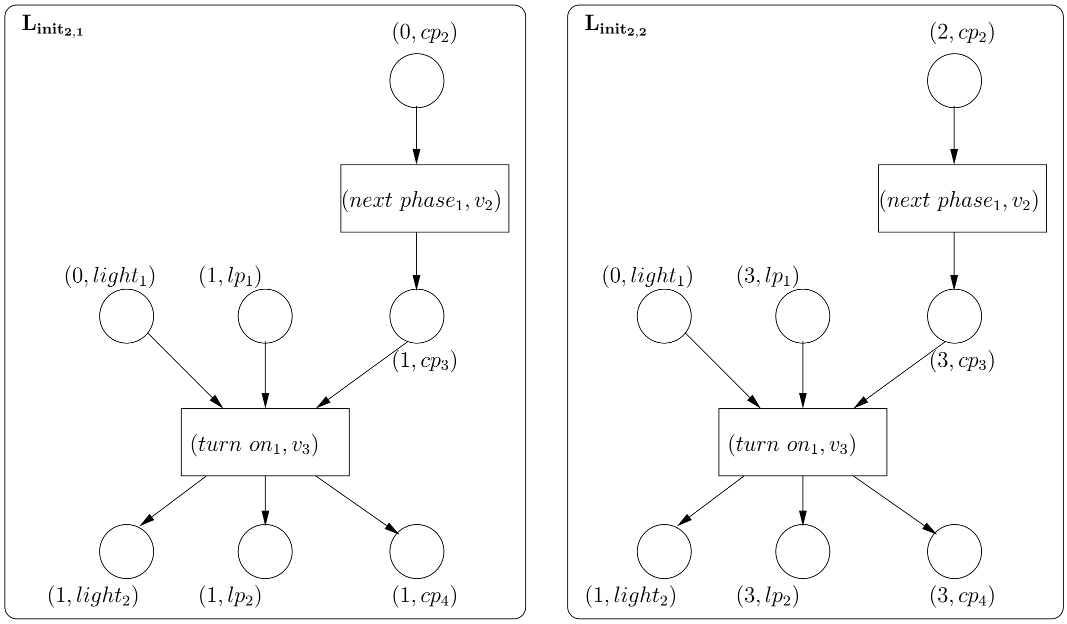


Figure 10: Instantiations of KI_2

2. $mp_1 \circ i_1 = mp_2 \circ i_2$.

$\triangle$

Theorem 3.11 (Sequential Composition of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two instantiated AHL-processes of the AHL-net AN such that (mp_1, mp_2) are sequentially composable w.r.t. (I, i_1, i_2). Then the instantiated AHL-occurrence net $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ together with the induced AHL-net morphism $mp : K \to AN$ is an instantiated AHL-process of the AHL-net AN, where K is the AHL-occurrence net of KI.

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "\mathbf{(PO)}"] & K_1 \arrow[d, "i_1'"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "i_2'"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\
& & AN
\end{tikzcd}$$

Proof. Fact 5.31 implies that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2). Hence by Theorem 5.32 KI together with the induced morphism $mp : K \to AN$ is an AHL-process with instantiations of the AHL-net AN. $\square$

Example 3.12 (Sequential Composition of AHL-Processes with Instantiations)
Consider the sequential composition of instantiations $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ in Example

3.9 and AHL-morphisms $mp_1 : K_1 \to Alarm$, $mp_2 : K_2 \to Alarm$ which map places and transitions to places respectively transitions with corresponding names (i.e. with the same name but without an index). Then there exists a unique morphism $mp : K \to Alarm$ which maps the elements in K in the same way to $Alarm$ as mp_1 and mp_2 do. The instantiated AHL-process KI together with mp is the sequential composition of KI_1 together with mp_1 and KI_2 together with mp_2. △

3.2 Equivalence and Independence of Algebraic High-Level Processes

Because for low-level occurrence nets the input/output behaviour is fixed by the net structure, two low-level occurrence nets should be considered to be equivalent if they are isormorphic. For high-level occurrence nets the input/output behaviour additionally depends on the marking of their input places and on corresponding variable assignments. Hence we introduce the equivalence of two AHL-processes with instantiations, where the net structures of equivalent AHL-processes may be different, but they have especially the same input/output behaviour. In more detail they have (up to renaming) the same sets of transitions and places and their instantiations are equivalent, i.e. there exist corresponding instantiations with the same input/output behaviour. So specific firing sequences of equivalent AHL-processes are comparable in the sense that they start and end up with the same data elements as marking of their input places and output places, respectively, but in general the corresponding transitions are fired in a different order.

Definition 3.13 (Equivalence of AHL-Processes with Instantiations)
Let $KI = (K, INIT, INS)$ and $KI' = (K', INIT', INS')$ together with AHL-net morphisms $mp : K \to AN$ and $mp' : K' \to AN$ two AHL-processes of an AHL-net AN. Then these two processes are called equivalent if

1. there are bijections $e_P : P_K \to P_{K'}$ and $e_T : T_K \to T_{K'}$ such the following diagram commutes componentwise

$$\begin{tikzcd} K \arrow[rr, shift left, "e_P"] \arrow[rr, shift right, "e_T"'] \arrow[dr, "mp"'] & {} \arrow[d, phantom, "="] & K' \arrow[dl, "mp'"] \\ & AN & \end{tikzcd}$$

2. and the instantiations are equivalent, i.e. for each $L_{init} \in INS$ there exists a $L_{init'} \in INS'$ and vice versa such that

$$\forall (a,p) \in A_{type(p)} \otimes P_K : \quad \begin{array}{l} (a,p) \in IN(L_{init}) \Leftrightarrow (a, e_P(p)) \in IN(L_{init'}) \text{ and} \\ (a,p) \in OUT(L_{init}) \Leftrightarrow (a, e_P(p)) \in OUT(L_{init'}) \end{array}$$

△

The equivalence of the instantiations means that there is a bijection between the input places $IN(K)$ and $IN(K')$ (resp. output places $OUT(K)$ and $OUT(K\prime)$), s.t. the input-output function $inout : INIT \to EXIT$ of KI and $inout' : INIT' \to EXIT'$ of KI' are equal up to bijection of input and output places. But it is not required that $e = (e_P, e_T) : K \to K'$ is an isomorphism, i.e. in general $e = (e_P, e_T)$ is not compatible with pre- and post domains.

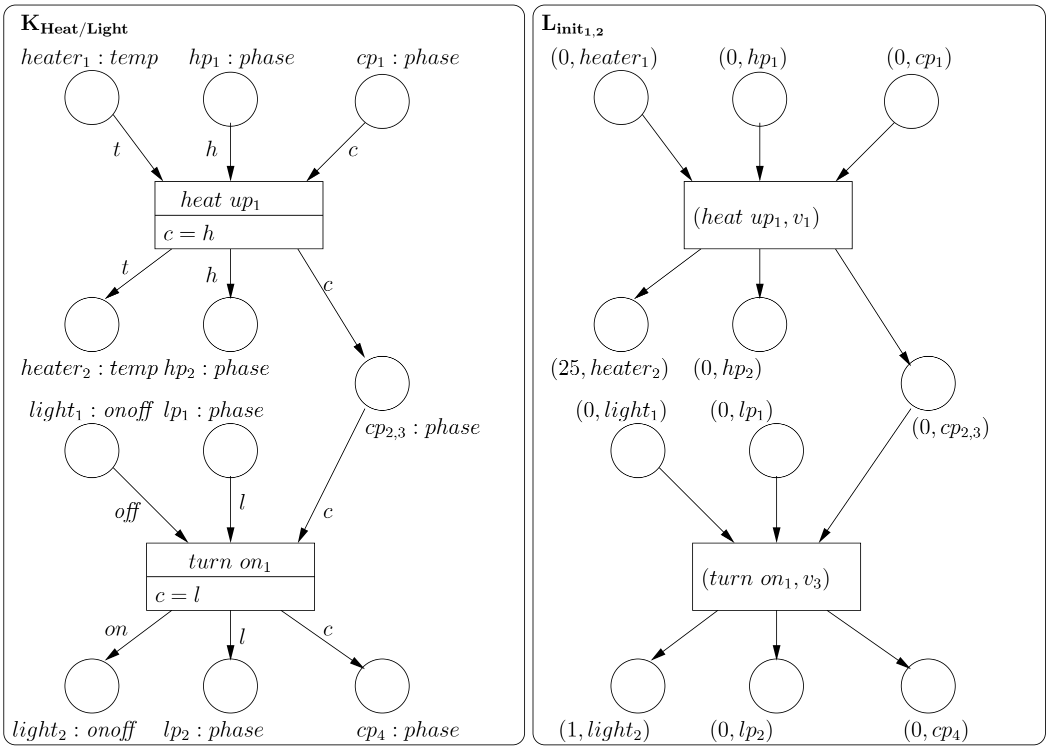


Figure 11: AHL-Occurrence Net with Instantiations $KI_{Heat/Light}$

Example 3.14 (Equivalence of AHL-Processes with Instantiations)
Consider the AHL-occurrence nets $KI_{Heat/Light}$ in Figure 11 and $KI_{Light/Heat}$ in Figure 12 together with suitable process morphisms $mp_1 : K_{Heat/Light} \to Alarm$ and $mp_2 : K_{Light/Heat} \to Alarm$.
Both of the processes model the heating up and turning on the light in phase 0 but in reverse order.
We can define bijections $e_P : P_{Heat/Light} \to P_{Light/Heat}$ and $e_T : T_{Heat/Light} \to T_{Light/Heat}$ where e_T is the identity. The function e_P maps all places except the cp_i places to identically and $e_P(cp_1) = cp_3$, $e_P(cp_{2,3}) = cp_{4,1}$ and $e_P(cp_4) = cp_2$.
$e = (e_P, e_T)$ is not an AHL-morphism because there is cp_1 in the pre domain of $heat\ up_1$ in $K_{Heat/Light}$ but $e_P(cp_1) = cp_3$ is not in the pre domain of $heat\ up_1 = e_T(heat\ up_1)$ in $K_{Light/Heat}$, i.e. the pre domains are not preserved.
The instantiations $L_{init_{1,2}}$ and $L_{init_{2,1}}$ are equivalent because there are the same data elements on corresponding places and the same assignments on corresponding transitions with respect to e_P and e_T in the respective instantiations.
So the AHL-processes mp_1 and mp_2 are equivalent.

$\triangle$

The main result in this context are suitable conditions s.t. AHL-net processes with instantiation can be composed in any order leading to equivalent high-level net processes. Here we use especially the assumption that the instantiations are consistent, i.e. there is a close

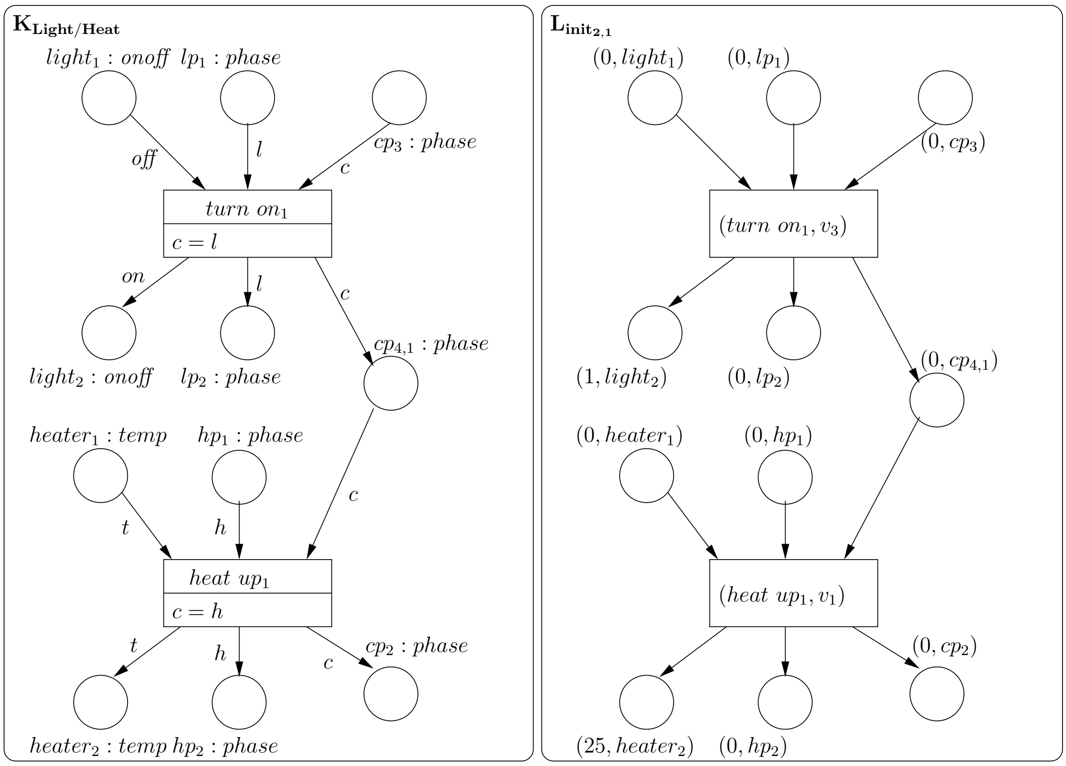


Figure 12: AHL-Occurrence Net with Instantiations $KI_{Light/Heat}$

relation between their input and output places.

Definition 3.15 (Consistency of Instantiations)
Given AHL-occurrence nets K_1, K_2 and I as in Def. 3.1 and injective AHL-net morphism $i_1 : I \to K_1$, $i_2 : I \to K_2$, $i_3 : I \to K_1$ and $i_4 : I \to K_2$ such that (K_1, K_2) is composable w.r.t. (I, i_1, i_2) and (K_2, K_1) is composable w.r.t. (I, i_4, i_3) with pushout (1) and (2), respectively. Moreover let $KI_x = (K_x, INIT_x, INS_x)$ be AHL-occurrence nets with instantiations for K_x $(x = 1, 2)$.
Then (INS_1, INS_2) is called consistent if for all composable $(L_{init_1}, L_{init_2}) \in INS_1 \times INS_2$ w.r.t. (J, j_1, j_2) induced by (I, i_1, i_2) with pushout (3) there are composable $(L_{init'_2}, L_{init'_1}) \in INS_2 \times INS_1$ w.r.t. (J, j_4, j_3) induced by (I, i_4, i_3) with pushout (4) and vice versa, s.t. in both cases the instantiations satisfy the following properties 1.-4. for gluing points GP defined below:

1. $IN(L_{init_x}) \setminus GP(L_{init_x}) = IN(L_{init'_x}) \setminus GP(L_{init'_x})$ and

2. $OUT(L_{init_x}) \setminus GP(L_{init_x}) = OUT(L_{init'_x}) \setminus GP(L_{init'_x})$ for $x = 1, 2$

Moreover we require for all $(a, p) \in A_{type(p)} \otimes P_I$:

3. $(a, i_3(p)) \in IN(L_{init_1}) \Leftrightarrow (a, i_2(p)) \in IN(L_{init'_2})$

4. $(a, i_1(p)) \in OUT(L_{init'_1}) \Leftrightarrow (a, i_4(p)) \in OUT(L_{init_2})$

The gluing points GP are defined by

- $GP(P_{K_1}) = i_1(P_I) \cup i_3(P_I)$, $GP(P_{K_2}) = i_2(P_I) \cup i_4(P_I)$,
- $GP(L_{init_x}) = \{(a, p) \in L_{init_x} | p \in GP(P_{K_x})\}$ and
- $GP(L_{init'_x}) = \{(a, p) \in L_{init'_x} | p \in GP(P_{K_x})\}$ for $x = 1, 2$.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd} \quad \begin{tikzcd} I \arrow[r, "i_4"] \arrow[d, "i_3"'] \arrow[dr, phantom, "(2)"] & K_2 \arrow[d, "i'_4"] \\ K_1 \arrow[r, "i'_3"'] & K \end{tikzcd} \quad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(3)"] & L_{init_1} \arrow[d, "j'_1"] \\ L_{init_2} \arrow[r, "j'_2"'] & L_{init} \end{tikzcd} \quad \begin{tikzcd} J \arrow[r, "j_4"] \arrow[d, "j_3"'] \arrow[dr, phantom, "(4)"] & L_{init'_2} \arrow[d, "j'_4"] \\ L_{init'_1} \arrow[r, "j'_3"'] & L_{init'} \end{tikzcd}$$

△

Example 3.16 (Consistency of Instantiations)
Consider the AHL-occurrence nets with instantiations KI_{Heat} depicted in Figure 13 and KI_{Light} depicted in Figure 14 together with suitable process morphisms $mp_{Heat} : K_{Heat} \to Alarm$ and $mp_{Light} : K_{Light} \to Alarm$.
Furthermore let I and I' be two interfaces where each of them contains one single place. The place in I is mapped to cp_2 in K_{Heat} and to cp_3 in K_{Light}. The place in I' is mapped to cp_4 in K_{Light} and to cp_1 in K_{Heat}. Then KI_{Heat} and KI_{Light} are composable with respect to these instantiations and also their instantiations are composable.
Since the required properties above are satisfied $(INS_{Heat}, INS_{Light})$ are consistent.

△

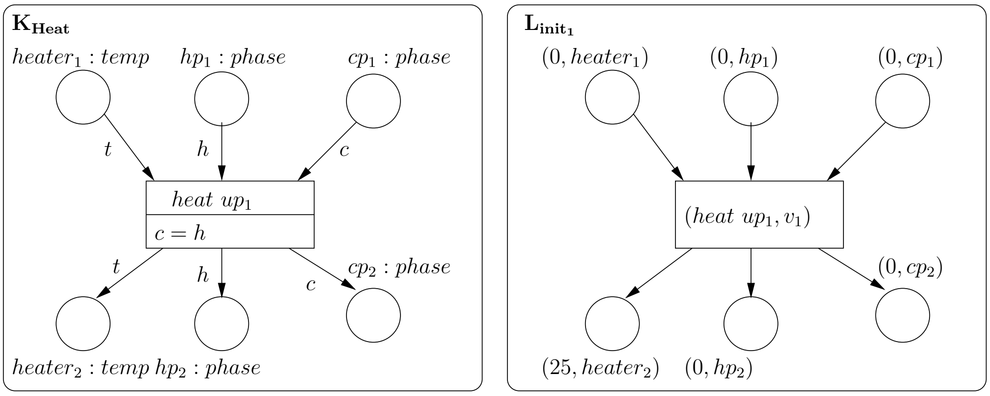


Figure 13: AHL-Occurrence Net with Instantiations KI_{Heat}

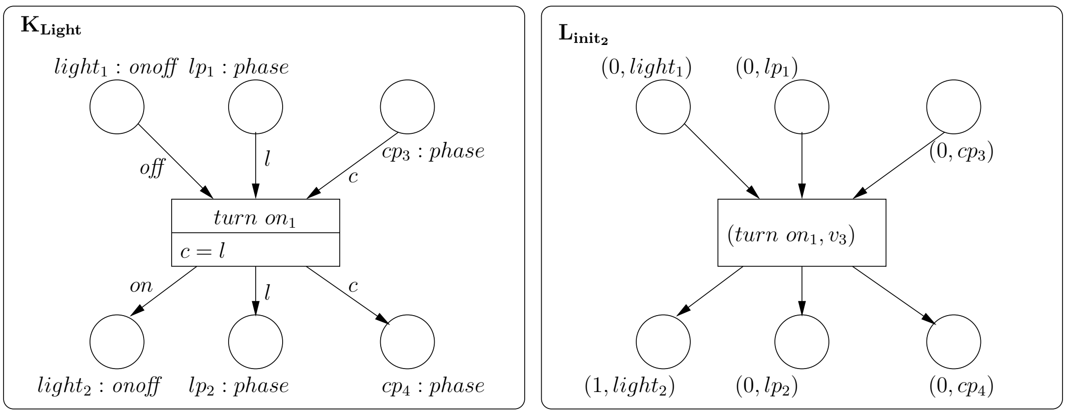


Figure 14: AHL-Occurrence Net with Instantiations KI_{Light}

Theorem 3.17 (Equivalence and Independence of AHL-Processes)
Given an AHL-net AN and AHL-occurrence nets $KI_x = (K_x, INIT_x, INS_x)$ with consistent instantiations as in Def. 3.15 with AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$. Then we have instantiated AHL-processes $KI = (K, INIT, INS)$ with $mp : K \to AN$ and $KI' = (K', INIT', INS')$ with $mp' : K' \to AN$ defined by opposite compositions $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ and $KI' = KI_2 \circ_{(I,i_4,i_3)} KI_1$ and both are equivalent processes of AN, provided that

1. K_1 and K_2 have no isolated places, i.e. $IN(K_x) \cap OUT(K_x) = \emptyset$ for $x = 1, 2$
2. mp_1 and mp_2 are compatible with i_1, i_2, i_3 and i_4, i.e.

$$mp_1 \circ i_1 = mp_2 \circ i_2 = mp_1 \circ i_3 = mp_2 \circ i_4 : I \to AN$$

Under these conditions KI_1 and KI_2 are called independent.

Proof sketch. The instantiated AHL-processes KI and KI' with $mp : K \to AN$ and $mp' : K' \to AN$ exist by Theorem 3.11. It remains to show that they are equivalent.

Construction of bijections:

The bijection $e_T : T_K \to T_{K'}$ follows from the fact that $I_T = \emptyset$ and hence $T_K \cong T_{K_1} \uplus T_{K_2}$ and $T_{K'} \cong T_{K_2} \uplus T_{K_1}$. In order to obtain the bijection $e_P : P_K \to P_{K'}$ we show that P_K and $P_{K'}$ can be represented by the following disjoint unions of gluing points GP and non gluing points NGP in pushout (1) and (2) in Def. 3.15.

$P_K = GP_1(P_K) \cup GP_2(P_K) \cup GP_3(P_K) \cup NGP(P_K)$ with
$GP_1(P_K) = i'_1 \circ i_3(P_I), GP_2(P_K) = i'_2 \circ i_4(P_I)$ and $GP_3(P_K) = i'_1 \circ i_1(P_I)$

$P_{K'} = GP_1(P_{K'}) \cup GP_2(P_{K'}) \cup GP_3(P_{K'}) \cup NGP(P_{K'})$ with
$GP_1(P_{K'}) = i'_4 \circ i_2(P_I), GP_2(P_{K'}) = i'_3 \circ i_1(P_I)$ and $GP_3(P_{K'}) = i'_3 \circ i_3(P_I)$

This allows to define $e_{P_x} : GP_x(P_K) \to GP_x(P_{K'})$ for $x = 1, 2, 3$ by $e_{P_1}(i'_1 \circ i_3(p)) = i'_4 \circ i_2(p)$ for all $p \in P_I$ and similar for e_{P_2} and e_{P_3}. Since i'_1, i_3, i'_4, and i_2 are all injective e_{P_1} is bijective and similar also e_{P_2} and e_{P_3} are bijective.

Finally also $e_{P_4} : NGP(P_K) \to NGP(P_{K'})$ can be defined as bijection. Using $IN(K_x) \cap OUT(K_x) = \emptyset$ for $x = 1, 2$ it can be shown that P_K (and similar $P_{K'}$) is a disjoint union of all four components leading to a bijection $e_P = e_{P_1} \cup e_{P_2} \cup e_{P_3} \cup e_{P_4} : P_K \to P_{K'}$. With these definitions it can be shown explicitly that the diagram in Definition 3.13 commutes componentwise.

Equivalence of instantiations:

Given $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ with pushout (3) in Definition 3.15 we have by consistency of (INS_1, INS_2) $L_{init'} = L_{init'_2} \circ_{(J,j_4,j_3)} L_{init'_1}$ with pushout (4) s.t. properties 1.-4. in Def. 3.15 are satisfied. This allows to show by case distinction using the definition of e_P above that we have for all $(a, p) \in A_{type(p)} \otimes P_K$: $(a, p) \in IN(L_{init}) \Leftrightarrow (a, e_P(p)) \in IN(L_{init'})$ and $(a, p) \in OUT(L_{init}) \Leftrightarrow (a, e_P(p)) \in OUT(L_{init'})$.

The opposite direction, where $L_{init'} = L_{init'_2} \circ_{(J,j_4,j_3)} L_{init'_1}$ is given with pushout (4), follows by symmetry.

For a detailed proof see Detailed Proof C.5 in the appendix. □

Equivalence of KI and KI' in Theorem 3.17 intuitively means that the AHL-processes KI_1 and KI_2 with consistent instantiations can be considered to be independent, because composition in each order leads to equivalent processes.

Example 3.18 (Equivalence and Independence of AHL-Processes)
The AHL-processes presented in Example 3.16 have consistent sets of instantiations. They can be composed in any order leading to the processes presented in Example 3.14 which are equivalent. $\triangle$

4 Instantiations and Initializations

Categorical Constructions like pushouts and pullbacks are only unique up to isomorphisms which in the case of sets and P/T-nets means up to renaming. Instantiations are special P/T-nets where parts of the data of a High-Level net are encrypted in the names of places and transitions. In order to preserve the data in the instantiations the category **PTNet** is not a suitable category for categorical constructions of instantiations. For this reason in this section we define a category **INet** which we can use for unique preimage constructions of instantiations and initializations which preserve their data parts. We can use these constructions for the composition and especially the decomposition of instantiations.

4.1 Category of Instantiations

In order to ensure unique constructions of instantiations we define a category **INet**. In this category the objects are instantiations not only of AHL-occurrence nets but of AHL nets. The reason for this is that we do not need the properties of occurrence nets for the unique construction so that we can handle these properties seperately to make it easier to work with the category. Otherwise one should always proof for every construction that the properties of AHL-occurrence nets are fulfilled even if it is not needed.
The morphisms of the category **INet** are pairs of one AHL-net morphism and one Low level net-morphism per **INet**-morphism where the data part of the Low level net-morphism is the identity.

Definition 4.1 (Category **INet**)
We define the category **INet** of instantiations as a subcategory of the product category **PTNet** $\times$ **AHLNet** in the following way:

$$\begin{aligned} Ob_{\mathbf{INet}} = \{(L,K) | \exists in : L \to Flat(K) \text{ inclusion and} \\ proj(K) \circ in : L \to Skel(K) \text{ isomorphism } \} \end{aligned}$$

$$\begin{aligned} Mor_{\mathbf{INet}}((L_1,K_1),(L_2,K_2)) = \{(f_L,f_H) | f_L : L_1 \to L_2 \ \mathbf{PTNet}\text{-morphism },\\ f_H : K_1 \to K_2 \ \mathbf{AHLNet}\text{-morphism, s.t. (1) commutes } \} \end{aligned}$$

$$\begin{tikzcd} L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(1)"] & L_2 \arrow[d, "in_2"] \\ Flat(K_1) \arrow[r, "Flat(f_H)"'] & Flat(K_2) \end{tikzcd}$$

$$(g_L,g_H) \circ (f_L,f_H) = (g_L \circ f_L, g_H \circ f_H)$$

$$id_{(L,K)} = (id_L, id_K)$$

Furthermore we define the the full subcategory **IONet** $\subseteq$ **INet** where for the objects (L,K) the net K is an AHL-occurrence net (and therefore L is a low-level-occurrence net). $\triangle$

Example 4.2 (Instantiations and Instantiation Morphisms)
Consider the P/T-nets L_1 and L_2, and AHL-nets GE_1 and GE_2 depicted in Figure 15. L_1

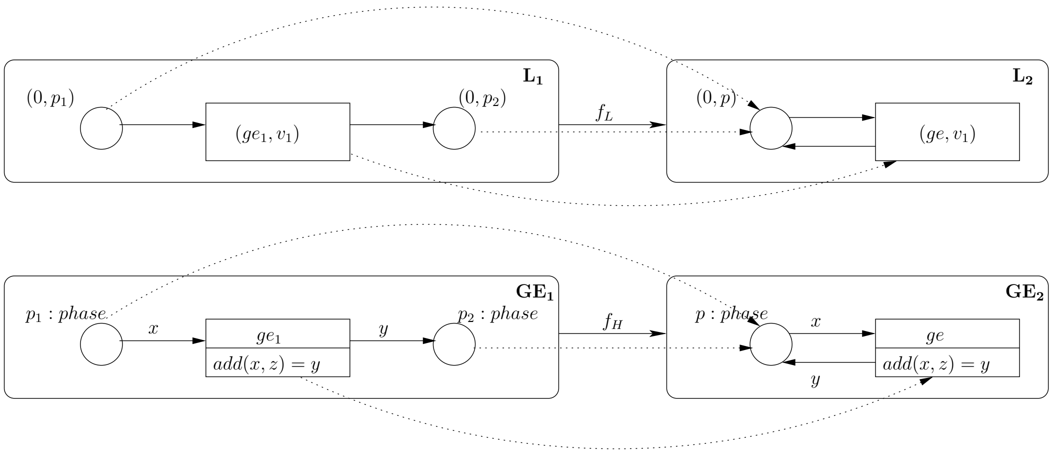


Figure 15: AHL-morphism f_H and corresponding instantiation morphism f_L

is an instantiation of GE_1 and L_2 is an instantiation of L_2. The P/T-morphism f_L maps the the data part identically and the name part in the same way as f_H does. (f_L, f_H) is an **INet**-morphism. △

Fact 4.3 (Category **INet**)
INet as defined in Definition 4.1 is a category.

Proof sketch. The composition of commutative diagrams leads to a commutatitive diagram. So due to the functor property of $Flat$ the composition is well-defined and associative and also the identity is well-defined. The neutrality of the identity follows from the neutrality in the product category in **PTNet** × **AHLNet**.
For a detailed proof see Detailed Proof C.10 in the appendix. □

Due to the definition of the morphisms of the category **INet** every morphism in the category is unique with respect to its high level net-morphism, i.e. if there are two **INet**-morphisms with the same high level part then the morphisms are equal.

Lemma 4.4 (Uniqueness of Instantiation morphisms)
Given **INet**-objects (L_1, K_1), (L_1', K_1) and (L_2, K_2), **AHLNet**-morphism $f_H : K_1 \to K_2$ and **PTNet**-morphisms $f_L : L_1 \to L_2$, $f_L' : L_1 \to L_2$.
If $f = (f_L, f_H)$ and $f' = (f_L', f_H)$ are **INet**-morphisms then $f = f'$, i.e. $f_L = f_L'$.

$$\begin{tikzcd}
L_1 \arrow[r, bend left, "f_L"] \arrow[r, bend right, "f_L'"'] \arrow[d, "in_1"'] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) \arrow[r, "Flat(f_H)"'] & Flat(K_2)
\end{tikzcd}$$

Proof. Due to the properties of **INet**-morphisms we have
$in_2 \circ f_L = Flat(f_H) \circ in_1 = in_2 \circ f_L'$

and since in_2 is a monomorphism this implies $f_L = f'_L$.

□

4.2 Preimage Constructions

We can use the uniqueness of **INet**-morphisms for a unique preimage-construction for every high level net morphism having an occurrence net with instantiations as codomain.
For each instantiation, every element in the domain of the morphism gets the data in that instantiation corresponding to the image of that element.

Lemma 4.5 (Unique Instantiation Preimage)
Given (L_2, K_2) in **INet** then for all $f_H : K_1 \to K_2$ in **AHLNet** there exists a unique instantiation L_1 and morphism $f_L : L_1 \to L_2$ in **PTNet** s.t. (L_1, K_1) is in **INet** and $f = (f_L, f_H) : (L_1, K_1) \to (L_2, K_2)$ is an **INet**-morphism.
Furthermore diagram (1) is a pullback in **PTNet**.

$$\begin{tikzcd}
L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(\mathbf{1})"] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2)
\end{tikzcd}$$

Proof sketch. Since the category **PTNet** has pullbacks we can construct the pullback (1) which implies that (1) commutes. Then due to the fact that *proj* induces pullbacks there is also (2) a pullback and hence (1)+(2) is a pullback which preserves monomorphisms. Thus $proj(K_1) \circ in_1$ is a monomorphim because $proj(K_2) \circ in_2$ is a monomorphism and hence injective. The surjectivity of $proj(K_2) \circ in_2$ can be show componentwise in **SET**.
The uniqueness of L_1 follows from the fact that the projection $proj(K_1) \circ in_1$ has to be an isomorphism and in_1 has to be an inclusion.
The uniqueness of f_L follows from the uniqueness of instantiation morphisms.
For a detailed proof see Detailed Proof C.11 in the appendix. □

Example 4.6 (Instantiation Preimage)
The instantiation L_1 together with morphism f_L in Example 4.2 is the unique instantiation preimage of the instantiation L_2 with respect to the morphism f_H depicted in Figure 15.
Due to the fact that the set A_{phase} in the algebra A is infinite we do not show the pullback (1) but the pullback (1)+(2) is depicted in Figure 16. Since (2) is a pullback by Theorem 2.33 it follows from the decomposition of pullbacks that (1) is the required pullback.

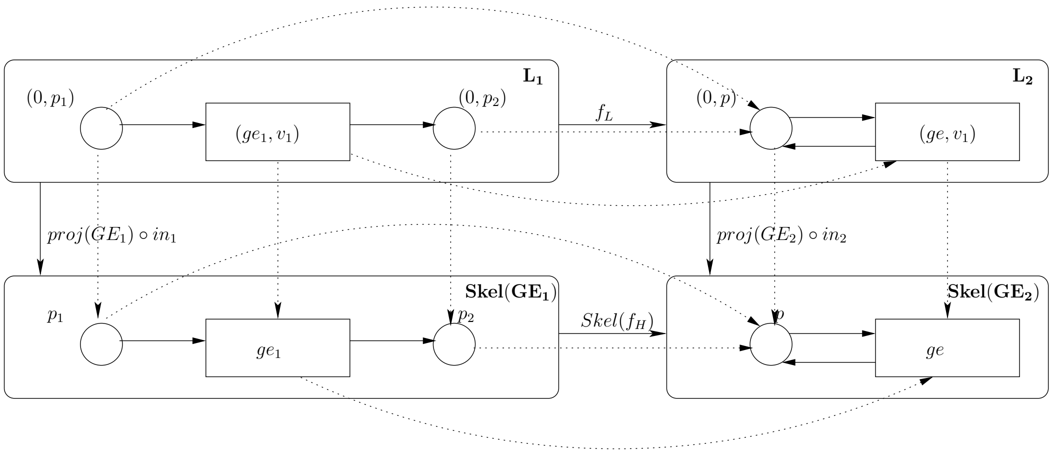


Figure 16: Pullback diagram in **PTNet**

$$\begin{tikzcd}
L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(\mathbf{1})"] & L_2 \arrow[d, "in_2"] \\
Flat(GE_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(GE_1)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(GE_2) \arrow[d, "proj(GE_2)"] \\
Skel(GE_1) \arrow[r, "Skel(f_H)"'] & Skel(GE_2)
\end{tikzcd}$$

△

Instantiations and corresponding initializations can be obtained from one high-level net K_2 along a high-level morphism $f_H : K_1 \to K_2$. The data elements in the instantiation or initialization of the net K_2 are transferred to the preimages of the corresponding places and transitions in the net K_1.
Using the uniqueness of **INet**-morphisms guarantees that the constructions lead to unique instantiations and initializations, respectively.

Lemma 4.7 (Instantiation Preimage Construction)
Given an AHL-occurrence net K_2, an instantiation L_2 of K_2 and an AHL-morphism $f_H : K_1 \to K_2$.
Then the unique preimage L_1 of L_2 w.r.t. f_H denoted $L_1 = PreIns(f_H)(L_2)$ and the morphism $f_L : L_1 \to L_2$ such that (f_L, f_H) in **INet** can be computed by

- $L_1 = (P_{L_1}, T_{L_1}, pre_{L_1}, post_{L_1})$
  with the set of places

$$P_{L_1} = \{(a, p) \mid p \in P_{K_1} \text{ and } (a, f_{H,P}(p)) \in P_{L_2}\}$$

  the set of transitions

$$T_{L_1} = \{(t, v) \mid t \in T_{K_1} \text{ and } (f_{H,T}(t), v) \in T_{L_2}\}$$

and pre conditions $pre_{L_1} : T_{L_1} \to P_{L_1}^{\oplus}$ with

$$pre_{L_1}(t, v) = \sum_{i=1}^{n} (a_i, p_i)$$

for

$$pre_{Skel(K_1)}(t) = \sum_{i=1}^{n} p_i$$

and

$$pre_{L_2}(f_{H,T}(t), v) = \sum_{i=1}^{n} (a_i, f_{H,P}(p_i))$$

and post conditions $post_{L_1} : T_{L_1} \to P_{L_1}^{\oplus}$ with

$$post_{L_1}(t, v) = \sum_{i=1}^{n} (a_i, p_i)$$

for

$$post_{Skel(K_1)}(t) = \sum_{i=1}^{n} p_i$$

and

$$post_{L_2}(f_{H,T}(t), v) = \sum_{i=1}^{n} (a_i, f_{H,P}(p_i))$$

- $f_L = (f_{L,P}, f_{L,T})$
  with the function for places

$$f_{L,P}(a, p) = (a, f_{H,P}(p))$$

and the function for transitions

$$f_{L,T}(t, v) = (f_{H,T}(t), v)$$

$$\begin{tikzcd}
L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2)
\end{tikzcd}$$

Proof sketch. Since $proj(K_2) \circ in_2$ is an isomorphism it is a bijection and hence injective. This means that the pullback (1) + (2) in **PTNet** can be constructed componentwise for places and transitions in **SET** leading to sets P_{L_1} and T_{L_1} and induced functions $pre_{L_1}, post_{L_1}$ such that L_1 is the P/T-net and f_L the **PTNet**-morphism as defined above.

$$\begin{tikzcd}
L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(\mathbf{1})"] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2)
\end{tikzcd}$$

Diagram (2) is a pullback because *proj* induces pullbacks which by pullback decomposition implies that also (1) is pullback.
Then by Lemma 4.5 L_1 is the unique instantiation and f_L the unique morphism such that (f_L, f_H) is an **INet**-morphism.
For a detailed proof see Detailed Proof C.12 in the appendix. □

Example 4.8 (Instantiation Preimage)
Consider the AHL-morphism $f_H : GE_1 \to GE_2$ and the instantiation L_2 of GE_2 depicted in Figure 4.2.
The instantiation L_1 and morphism f_L which is also depicted in Figure 4.2 can be constructed exactly in the way defined in Lemma 4.7. △

Analogously to the preimage construction for instantiations we define a preimage construction for initializations. Before we show that fact we show that every initialization can be obtained from a corresponding instantiation with a pullback construction.

Lemma 4.9 (Initialization is Pullback)
Given an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$.
For every $init \in INIT$ and $L_{init} \in INS$ and inclusions $inc_1 : init \to P_{L_{init}}$ and $inc_2 : IN(K) \to P_{Skel(K)}$ diagram (1) is a pullback in **SET** where

$$pr(a, p) = p$$

$$\begin{tikzcd}
init \arrow[r, "inc_1"] \arrow[d, "pr"'] \arrow[dr, phantom, "(\mathbf{1})"] & P_{L_{init}} \arrow[d, "(proj(K)\circ in)_P"] \\
IN(K) \arrow[r, "inc_2"'] & P_{Skel(K)}
\end{tikzcd}$$

Proof sketch. The construction of pullback (1) in **SET** leads to the initialization *init*.
For a detailed proof see Detailed Proof C.13 in the appendix. □

Example 4.10 (Initialization is Pullback)
Consider again the instantiation L_1 depicted in Figure 15. Figure 17 shows the pullback (1) described in Lemma 4.9. △

Since input places of a net K_1 are not necessarily mapped to input places of K_2 via a high-level morphism $f_H : K_1 \to K_2$ the initialization of K_1 is not computed from an initialization of K_2 but from an instantiation. The resulting initialization is the corresponding initialization to the instantiation preimage of f_H and K_2.

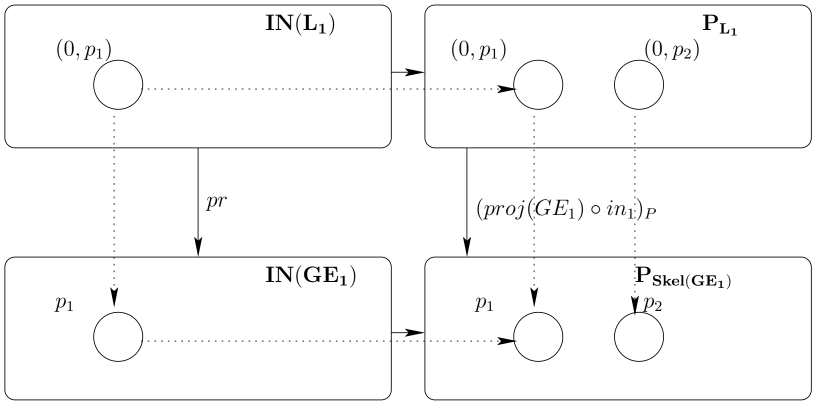


Figure 17: Pullback diagram in **SET**

Lemma 4.11 (Initialization Preimage Construction)
Given AHL-occurrence nets K_1, K_2, an AHL-morphism $f_H : K_1 \to K_2$ and an instantiation L_2 of K_2. Let L_1 and $f_L : L_1 \to L_2$ be the instantiation of K_1 and morphism induced by f_H. Then the initialization *init* corresponding to L_1 with $init = IN(L_1)$ can be computed by

$$init = \{(a, p) \mid p \in IN(K_1) \text{ and } (a, f_{H,P}(p)) \in P_{L_2}\}$$

and is denoted by $init = PreInit(f_H)(L_2)$.
With inclusions $inc_1 : init \to L_1$, $inc_2 : IN(K_1) \to P_{Skel(K_1)}$ diagram (1) is a pullback in **SET** where the projection $pr : init \to IN(K_1)$ with $pr(a, p) = p$ is an isomorphism.

$$\begin{tikzcd}
init_1 \arrow[r, "f_{L,P} \circ inc_1"] \arrow[d, "pr"'] \arrow[dr, phantom, "(1)"] & P_{L_2} \arrow[d, "(proj(K_2) \circ in_2)_P"] \\
IN(K_1) \arrow[r, "Skel(f_H)_P \circ inc_2"'] & P_{Skel(K_2)}
\end{tikzcd}$$

Proof sketch. We obtain pullback (3) from Lemma 4.7 and pullback (4) from Theorem 2.33 leading to pullback (3)+(4) by pullback composition. Furthermore there is (2) a pullback by Lemma 4.9 and hence (1) = (2)+(3)+(4) is a pullback in **SET**.

$$\begin{tikzcd}
init_1 \arrow[r, "inc_1"] \arrow[dd, "pr"'] \arrow[ddr, phantom, "(2)"] & P_{L_1} \arrow[r, "f_{L,P}"] \arrow[d, "in_{1,P}"'] \arrow[dr, phantom, "(3)"] & P_{L_2} \arrow[d, "in_{2,P}"] \\
& P_{Flat(K_1)} \arrow[r, "Flat(f_H)_P"] \arrow[d, "proj(K_1)_P"'] \arrow[dr, phantom, "(4)"] & P_{Flat(K_2)} \arrow[d, "proj(K_2)_P"] \\
IN(K_1) \arrow[r, "inc_2"'] & P_{Skel(K_1)} \arrow[r, "Skel(f_H)_P"'] & P_{Skel(K_2)}
\end{tikzcd}$$

For a detailed proof see Detailed Proof C.14 in the appendix. □

The preimage constructions for instantiations and initializations can be used to obtain sets of initializations and instantiations for an AHL-occurrence net K_1 via an **AHLNet**-morphism $f : K_1 \to K_2$ from an AHL-occurrence net with instantiations $KI_2 = (K_2, INIT_2, INS_2)$.

Definition 4.12 (Preimage of Initializations and Instantiations)
Given an AHL-occurrence net with instantiations $KI_2 = (K_2, INIT_2, INS_2)$ and an AHL-morphism $f : K_1 \to K_2$.
The preimage of initializations $INIT_1$ of INS_2 w.r.t. f is defined by

$$INIT_1 = \{init_1 \mid L_{init_2} \in INS_2 \text{ and } init_1 = PreInit(f)(L_{init_2})\}$$

and is denoted $INIT_1 = PreInit(f)(INS_2)$.
The preimage of instantiations INS_1 of INS_2 w.r.t. f is defined by

$$INS_1 = \{L_{init_1} \mid L_{init_2} \in INS_2 \text{ and } L_{init_1} = PreInit(f)(L_{init_2})\}$$

and is denoted $INS_1 = PreIns(f)(INS_2)$. △

The preimage of initializations and instantiations does not always lead to an AHL-occurrence net with instantiations because it is possible that two different instantiantions of KI_2 have different preimages of instantiations but the same preimage of initializations. Otherwise the preimages of initializations and instantiations lead to an AHL-occurrence net with instantiations.

Theorem 4.13 (Preimage of AHL-occurrence nets with instantiations)
Given an AHL-occurrence net K_1, an AHL-occurrence net with instantiations $KI_2 = (K_2, INIT_2, INS_2)$ and an AHL-morphism $f_H : K_1 \to K_2$.
Then $KI_1 = (K_1, INIT_1, INS_1)$ with

$$INIT_1 = PreInit(f_H)(INS_2)$$

and

$$INS_1 = PreIns(f_H)(INS_2)$$

is an AHL-occurrence net with instantiations iff

$$IN : PreIns(f_H)(INS_2) \to PreInit(f_H)(INS_2)$$

with

$$IN(PreIns(f_H)(L)) = PreInit(f_H)(L)$$

is injective.
We call KI_1 induced preimage of KI_2 and f_H.

$$\begin{tikzcd}
INS_2 \arrow[r, "PreIns(f_H)"] \arrow[dr, "PreInit(f_H)"'] & PreIns(f_H)(INS_2) \arrow[d, "IN"] \arrow[dl, phantom, "(=)"] \\
 & PreInit(f_H)(INS_2)
\end{tikzcd}$$

Proof sketch. Since for every $L_1 \in INS_1$ there is $L_2 \in INS_2$ with $PreIns(f_H)(L_2) = L_1$ and the construction $PreInit(f_H)(L_2)$ provides an initialization for the instantiation L_1 the function IN is surjective. Furthermore the sets INS_1 and $INIT_1$ are well-defined sets of instantiations and initializations of K_1, respectively. So KI_1 is an AHL-occurrence net iff there is a bijective correspondence between INS_1 and $INIT_1$, i.e. iff the surjection IN is also injective.
For a detailed proof see Detailed Proof C.15 in the appendix. □

Example 4.14 (Preimage of AHL-Occurrence Nets with Instantiations)
Consider the AHL-occurrence net with instantiations KI_2 depicted in Figure 18 and the AHL-occurrence net K_1 depicted in Figure 19 together with an inclusion $f : K_1 \to K_2$.
Then there are induced preimages of instantiations $L_{init_{1,1}}$ and $L_{init_{1,2}}$ together with corresponding preimages of initializations such that KI_1 as depicted in Figure 19 is an AHL-occurrence net with instantiations.

△

Remark 4.15. The construction $PreIns$ can be turned into a contravariant functor

$$PreIns : \mathbf{AHLNet} \to \mathbf{SET}$$

where $PreIns_{Ob}$ maps from an AHL-net to the set of all its instantiations and $PreIns_{Mor}$ maps from an AHL-morphism $f_H : AN_1 \to AN_2$ to a function $f_I : PreIns(AN_2) \to PreIns(AN_1)$ which maps every instantiation to its unique instantiation preimage.
The preimage construction $PreIns(f_H)(INS_2)$ then corresponds to taking the image of $PreIns(f_H)$ and the given set INS_2. The properties of contravariant functors would then imply that

$$PreIns(g \circ f) = PreIns(f) \circ PreIns(g)$$

and hence

$$PreIns(g \circ f)(INS_2) = PreIns(f)(PreIns(g)(INS_2))$$

However, this does also follow from the uniqueness of instantiation preimages.
A similar construction for the preimage of initializations would be more complicated since it is not computed with respect to another initialization but to an instantiation.

4.3 Pushout Construction of Instantiations

As we show in this section the category **INet** has pushouts. Since we are always considering instantiations with respect to specific high-level nets we are interested under which conditions there is a pushout of instantiations with respect to a specific pushout of high-level nets. This depends on the fact whether for a given span $K_1 \leftarrow I \rightarrow K_2$ of high-level nets there is a corresponding span $L_1 \leftarrow J \rightarrow L_2$ of instantiation L_1 of K_1 and L_2 of K_2.
In contrast to a preimage construction along one high-level morphism it is not always possible to get a common preimage of two instantiations (L_1, K_1) and (L_2, K_2) along different high-level morphisms $f_{1,H} : K_0 \to K_1$ and $f_{2,H} : K_0 \to K_2$.
The instantiations have to be compatible with the morphisms to construct a common preimage which can be used as an interface for the composition of instantiations.

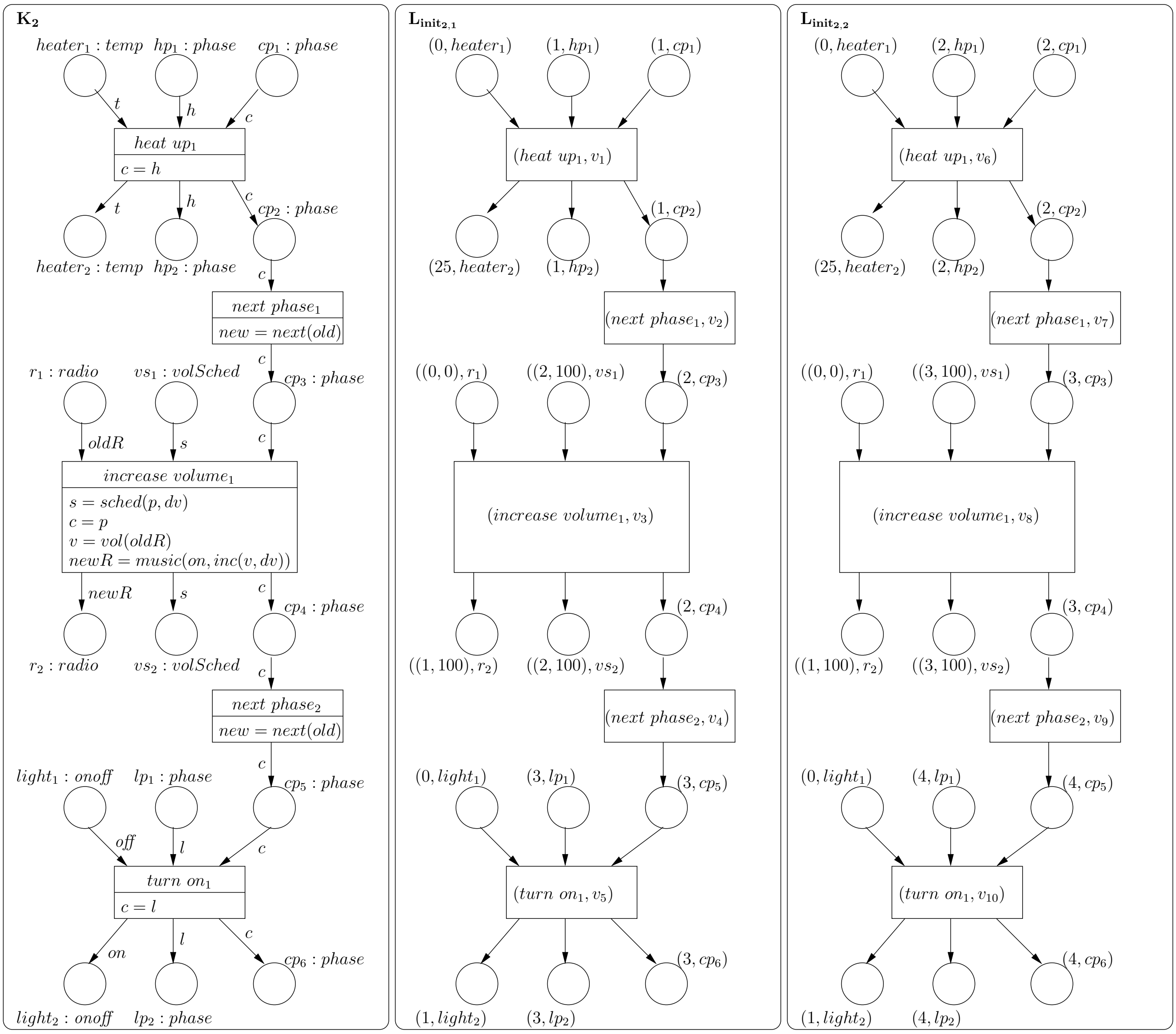


Figure 18: AHL-occurrence net with instantiations KI_2

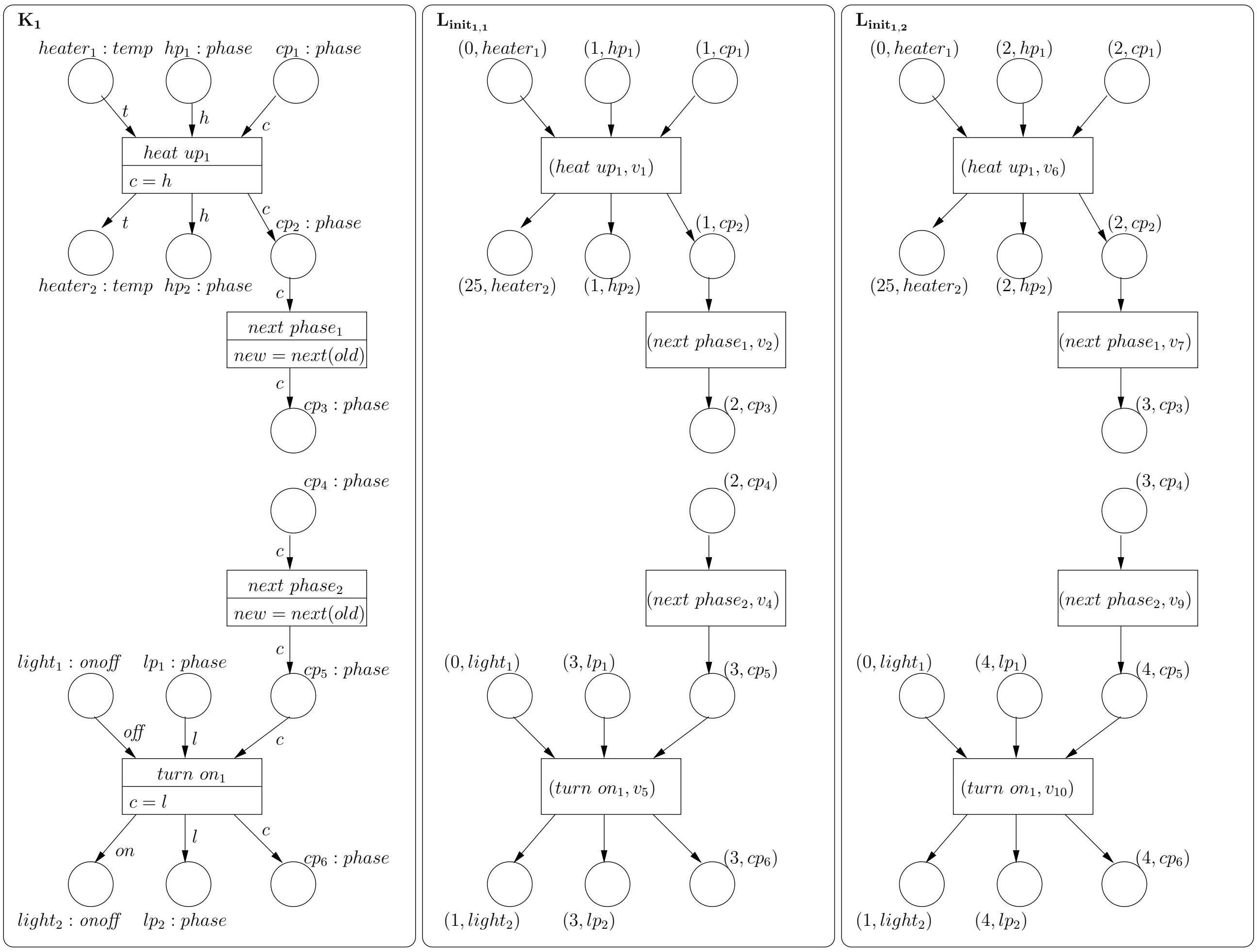


Figure 19: AHL-occurrence net with instantiations KI_1

Definition 4.16 (Compatibility of Instantiations)
Given (L_1, K_1), (L_2, K_2) in **INet** and **AHLNet**-morphisms $f_{1,H} : K_0 \to K_1$, $f_{2,H} : K_0 \to K_2$. Let $(I_1, e_{1,H}, m_{1,H})$, $(I_2, e_{2,H}, m_{2,H})$ be the epi-mono-factorization of $f_{1,H}$ resp. $f_{2,H}$ and J_1, J_2 the instantiations of I_1 resp. I_2 induced by $m_{1,H}$ resp. $m_{2,H}$.
Then (L_1, L_2) are compatible with $(f_{1,H}, f_{2,H})$ if there exists an **INet**-morphism

$$c = (c_L, c_H) : (J_1, I_1) \to (J_2, I_2)$$

with

$$c_H \circ e_{1,H} = e_{2,H}$$

$$\begin{tikzcd}
 & & & Skel(K_0) \arrow[dlll, "Skel(f_{1,H})"'] \arrow[dl, "Skel(e_{1,H})"] \arrow[dr, "Skel(e_{2,H})"] \arrow[drrr, "Skel(f_{2,H})"] & & & \\
Skel(K_1) & & Skel(I_1) \arrow[ll, "Skel(m_{1,H})"] \arrow[rr, "Skel(c_H)"] & & Skel(I_2) \arrow[rr, "Skel(m_{2,H})"'] & & Skel(K_2) \\
L_1 \arrow[u, "proj(K_1)\circ in_1"'] & & J_1 \arrow[ll, "m_{1,L}"] \arrow[rr, "c_L"'] \arrow[u, "proj(I_1)\circ inj_1"'] & & J_2 \arrow[rr, "m_{2,L}"'] \arrow[u, "proj(I_2)\circ inj_2"] & & L_2 \arrow[u, "proj(K_2)\circ in_2"]
\end{tikzcd}$$

△

Example 4.17 (Compatibility of Instantiations)
Consider the AHL-morphisms $f_{1,H} : I \to K_1$ and $f_{2,H} : I \to K_2$ together with epi-mono-factorizations $(I_1, e_{1,H}, m_{1,H})$ of $f_{1,H}$ and $(I_2, e_{2,H}, m_{2,H})$ of $f_{2,H}$ depicted in Figure 20.
We can define an AHL-morphism $c_H : I_1 \to I_2$ with $c_H \circ e_{1,H} = e_{2,H}$ as show in Figure 20.
Given the instantiations L_1 of K_1 and L_2 of K_2 in Figure 21 we can construct the preimages J_1 of L_1 and $m_{1,H}$ and J_2 of L_2 and $m_{2,H}$. There is a P/T-morphism $c_L : J_1 \to J_2$ as shown in Figure 21 such that (c_L, c_H) is an **INet**-morphism. Hence (L_1, L_2) are compatible with $(f_{1,H}, f_{2,H})$. △

The compatibility of instantiations with a given pair of high-level morphisms is a sufficient and necessary condition for the existence of a common preimage with respect to both of the high-level morphisms.

Theorem 4.18 (Instantiation Interface)
Given (L_1, K_1), (L_2, K_2) in **INet** and **AHLNet**-morphisms $f_{1,H} : K_0 \to K_1$, $f_{2,H} : K_0 \to K_2$. If and only if (L_1, L_2) are compatible with $(f_{1,H}, f_{2,H})$ then there exists a unique instantiation L_0 of K_0 and morphisms $f_{1,L} : L_0 \to L_1$, $f_{2,L} : L_0 \to L_2$ s.t. (L_0, K_0) and $(f_{1,L}, f_{1,H}), (f_{2,L}, f_{2,H})$ are in **INet**.

$$\begin{tikzcd}
L_1 \arrow[d, "in_1"'] \arrow[dr, phantom, "(1)"] & L_0 \arrow[l, "f_{L1}"'] \arrow[r, "f_{L2}"] \arrow[d, "in_0"] \arrow[dr, phantom, "(2)"] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) & Flat(K_0) \arrow[l, "Flat(f_{H1})"] \arrow[r, "Flat(f_{H2})"'] & Flat(K_2)
\end{tikzcd}$$

Proof sketch. The compatibility of instantiations implies the existence of a common preimage due to the well-defined composition and uniqueness of instantiation morphisms.

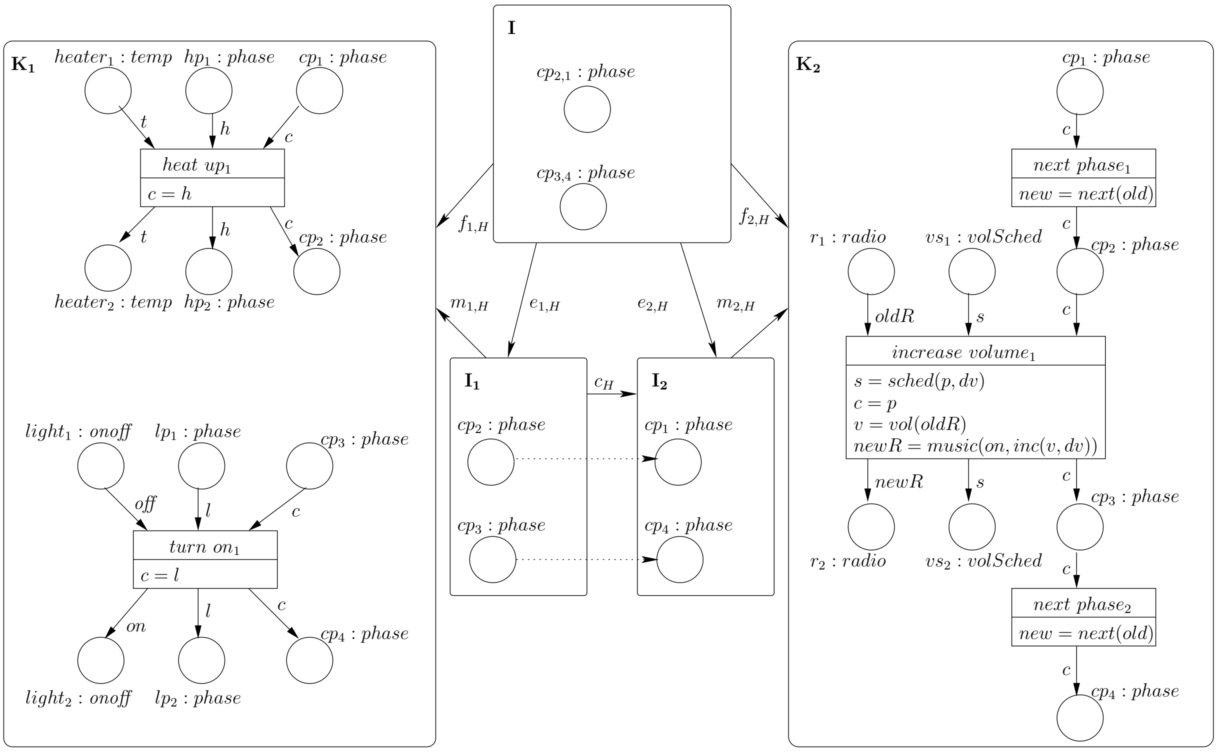


Figure 20: AHL-morphisms $f_{1,H}, f_{2,H}$ with epi-mono-factorization

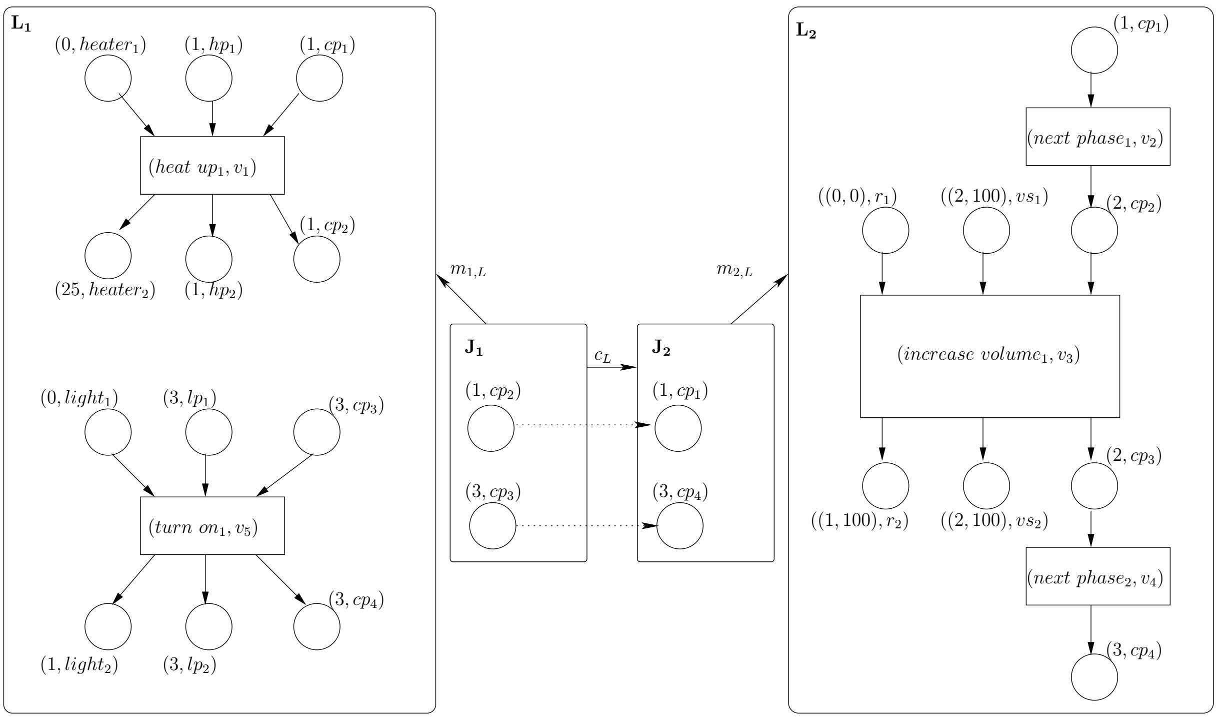


Figure 21: Compatibility of (L_1, L_2) with $(f_{1,H}, f_{2,H})$

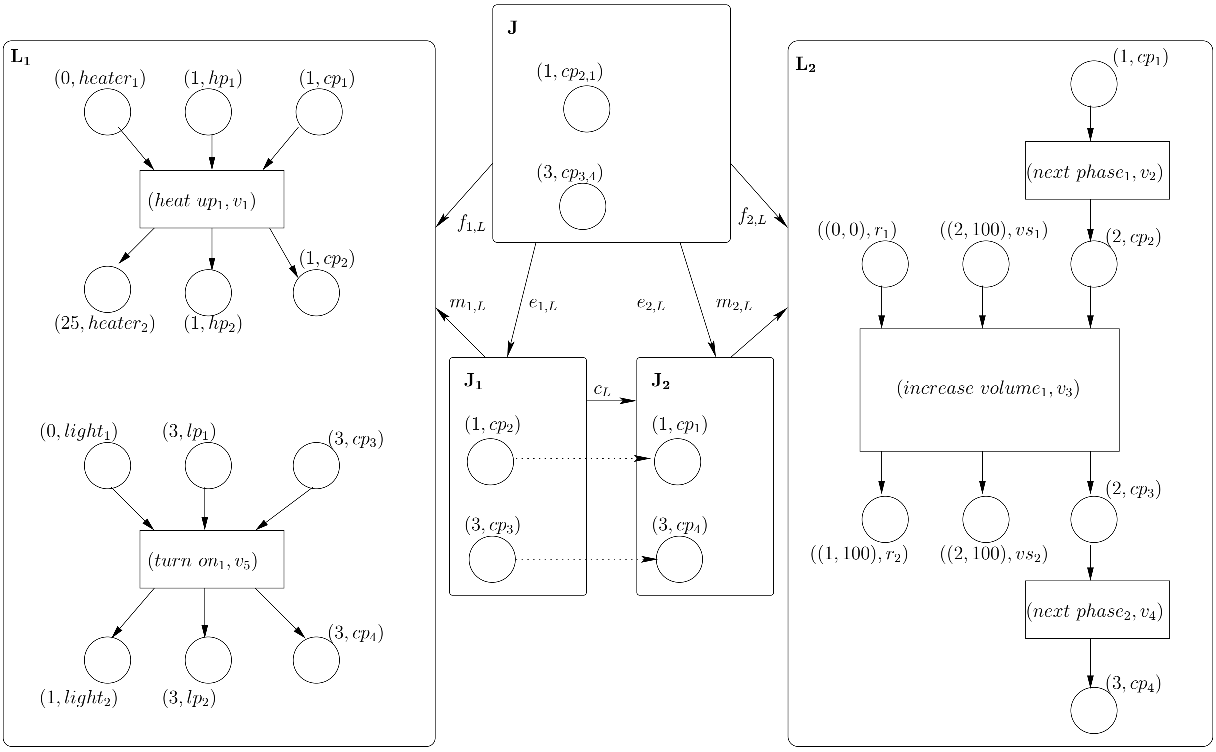


Figure 22: Instantiation interface

On the other hand the existence of a common preimage allows the construction of an **INet**-morphism between the induced instantiation preimages J_1 and J_2 of the epi-mono-factorizations of $f_{1,H}$ and $f_{2,H}$, respectively.
For a detailed proof see Detailed Proof C.16 in the appendix. □

Example 4.19 (Instantiation Interface)
Given the compatible instantiations in Example 4.17 we can construct the common interface J as shown in Figure 22 and (J, I) and all corresponding morphisms are in **INet**. △

Whenever there is a span $(L_1, K_1) \leftarrow (I, J) \rightarrow (L_2, K_2)$ in the category **INet** it is possible to create the pushout of that span.

Fact 4.20 (Category **INet** has Pushouts)
Given **INet**-objects (J, I), (L_1, K_1) and (L_2, K_2) together with **INet**-morphisms $f_1 = (f_{1,L}, f_{1,H}) : (J, I) \rightarrow (L_1, K_1)$ and $f_2 = (f_{2,L}, f_{2,H}) : (J, I) \rightarrow (L_2, K_2)$. Then there exists (L, K) together with morphisms $g_1 = (g_{1,L}, g_{1,H}) : (L_1, K_1) \rightarrow (L, K)$ and $g_2 = (g_{2,L}, g_{2,H}) : (L_2, K_2) \rightarrow (L, K)$ such that (1) is pushout in **INet**.
Furthermore the pushout can be constructed componentwisely, i.e. given pushout (1) in **INet** there are pushouts (2) in **PTNet** and (3) in **AHLNet**.

$$\begin{tikzcd}
(J,I) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & (L_1,K_1) \arrow[d, "g_1"] \\
(L_2,K_2) \arrow[r, "g_2"'] & (L,K)
\end{tikzcd}
\quad
\begin{tikzcd}
J \arrow[r, "f_{1,L}"] \arrow[d, "f_{2,L}"'] \arrow[dr, phantom, "(2)"] & L_1 \arrow[d, "g_{1,L}"] \\
L_2 \arrow[r, "g_{2,L}"'] & L
\end{tikzcd}
\quad
\begin{tikzcd}
I \arrow[r, "f_{1,H}"] \arrow[d, "f_{2,H}"'] \arrow[dr, phantom, "(3)"] & K_1 \arrow[d, "g_{1,H}"] \\
K_2 \arrow[r, "g_{2,H}"'] & K
\end{tikzcd}$$

Proof sketch. Since **PTNet** and **AHLNet** have pushouts we can construct pushouts (2) and (3). We obtain an isomorphism $proj(K) \circ in : L \to K$ from the fact that $proj(I) \circ in_I$, $proj(K_1) \circ in_1$ and $proj(K_2) \circ in_2$ are isomorphisms and the functors $Flat$ and $Skel$ preserve pushouts. Furthermore the decomposition of injective **PTNet**-morphisms implies that in is an injection which can be chosen to be an inclusion.
The pushout property of diagram (1) can be shown explicitely.
For a detailed proof see Detailed Proof C.17 in the appendix. □

Example 4.21 (Pushout in **INet**)
Figure 23 shows the pushout of (L_1, K_1) and (L_2, K_2) of Example 4.17 via the instantiation interface constructed in Example 4.19. △

One of the general properties of pushouts is that they are unique up to isomorphism. In the category **INet** the objects and morphisms consist of a high-level and a low-level part and the pushout can be constructed componentwisely for both parts as shown in Fact 4.20. For a given pushout in the high-level part which is unique up to isomorphism there is an absolutely unique corresponding pushout in the low-level part.

Fact 4.22 (Special Uniqueness of Pushouts in **INet**)
Given pushouts (1) and (2) in **INet** of the same span and via the same pushout (3) in **AHLNet** then $L = L'$.

$$\begin{tikzcd}
(J,I) \arrow[r, "(j_1{,}i_2)"] \arrow[d, "(j_2{,}i_2)"'] \arrow[dr, phantom, "(1)"] & (L_{init_1},K_1) \arrow[d, "(j_1'{,}i_1')"] \\
(L_{init_2},K_2) \arrow[r, "(j_2'{,}i_2')"'] & (L,K)
\end{tikzcd}
\quad
\begin{tikzcd}
(J,I) \arrow[r, "(j_1{,}i_2)"] \arrow[d, "(j_2{,}i_2)"'] \arrow[dr, phantom, "(2)"] & (L_{init_1},K_1) \arrow[d, "(j_3'{,}i_1')"] \\
(L_{init_2},K_2) \arrow[r, "(j_4'{,}i_2')"'] & (L',K)
\end{tikzcd}
\quad
\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(3)"] & K_1 \arrow[d, "i_1'"] \\
K_2 \arrow[r, "i_2'"'] & K
\end{tikzcd}$$

Proof sketch. Since pushouts in **INet** can be constructed componentwise there are pushouts (4) and (5) in **PTNet**.

$$\begin{tikzcd}
J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(4)"] & L_{init_1} \arrow[d, "j_1'"] \\
L_{init_2} \arrow[r, "j_2'"'] & L
\end{tikzcd}
\qquad
\begin{tikzcd}
J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(5)"] & L_{init_1} \arrow[d, "j_3'"] \\
L_{init_2} \arrow[r, "j_4'"'] & L'
\end{tikzcd}$$

This implies that j_1' and j_2' as well as j_3' and j_4' are jointly surjective. So the fact that instantiation morphisms map the data part of instantiations identically and the pushouts (1) and (2) are constructed via the same pushout (3) in **AHLNet** implying that the instantiations have also the same net part it follows that $L = L'$.
For a detailed proof see Detailed Proof C.18 in the appendix. □

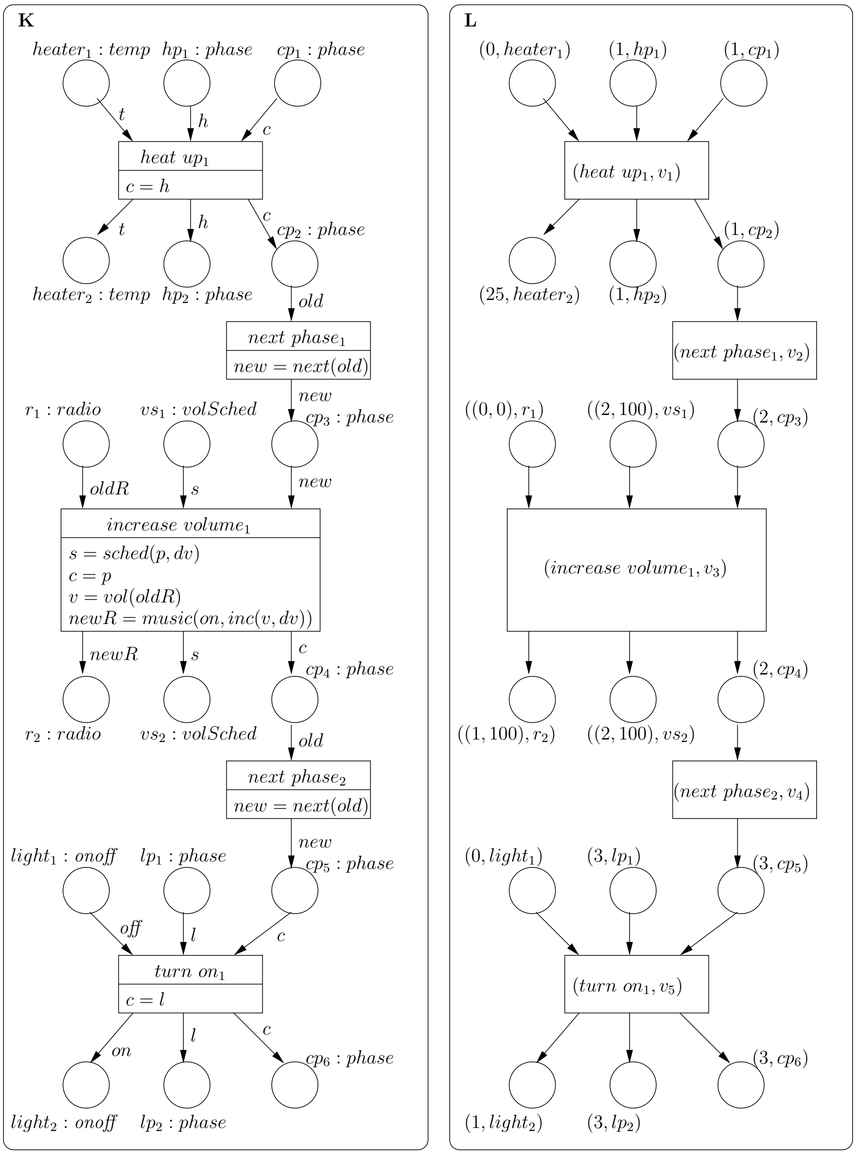


Figure 23: Pushout in **INet**

The special property of pushouts in the category **INet** does not only hold for the result but also for the arguments of the pushout construction.

Fact 4.23 (Pushout of Instantiations is Injective)
Given **INet**-objects (J, I), (J', I), (L_1, K_1), (L_3, K_1), (L_2, K_2) and (L_4, K_2) together with **INet**-morphisms $(j_1, i_1) : (J, I) \to (L_1, K_1), (j_2, i_2) : (J, I) \to (L_2, K_2)$ and $(j_3, i_1) : (J', I) \to (L_1, K_1), (j_4, i_2) : (J', I) \to (L_2, K_2)$. Then

$$(L_1, K_1) \circ_{(J,I)} (L_2, K_2) = (L_3, K_1) \circ_{(J',I)} (L_4, K_2) \Rightarrow L_1 = L_3 \text{ and } L_2 = L_4$$

$$\begin{tikzcd}
(J,I) \arrow[r, "{(j_1,i_1)}"] \arrow[d, "{(j_2,i_2)}"'] \arrow[dr, phantom, "(\mathbf{PO})"] & (L_1,K_1) \arrow[d, "{(j_1',i_1')}"] \\
(L_2,K_2) \arrow[r, "{(j_2',i_2')}"'] & (L,K)
\end{tikzcd}
\qquad
\begin{tikzcd}
(J',I) \arrow[r, "{(j_3,i_1)}"] \arrow[d, "{(j_4,i_2)}"'] \arrow[dr, phantom, "(\mathbf{PO})"] & (L_3,K_1) \arrow[d, "{(j_3',i_1')}"] \\
(L_4,K_2) \arrow[r, "{(j_4',i_2')}"'] & (L',K)
\end{tikzcd}$$

Proof sketch. Similar to the proof of Fact 4.22 we obtain that j_1' and j_2' as well as j_3' and j_4' are jointly surjective. So the assumption $L_1 \neq L_3$ or $L_2 \neq L_4$ would imply that $L \neq L'$ which by contraposition means that $L = L'$ implies that $L_1 = L_3$ and $L_2 = L_4$.
For a detailed proof see Detailed Proof C.19 in the appendix. □

4.4 Category of AHL-Nets with Instantiations

Based on the categories **AHLNet** of AHL-nets and **INet** of instantiations we can define a category for AHL-nets with instantiations. AHL-nets with instantiations are defined analogously to AHL-occurrence nets with instantiations in Definition 2.38 but without the restriction that the net has to be an occurrence net and without the requirement that there is a bijective correspondence between instantiations and initializations. We obtain the category of AHL-occurrence nets with instantiations as a full subcategory where the objects satisfy these requirements.

Definition 4.24 (Category **AHLNetI**)
We define the category **AHLNetI** of AHL-nets with instantiations as a subcategory of $\mathbf{AHLNet} \times \mathbf{SET}^{op}$ in the following way:

$$Ob_{\mathbf{AHLNetI}} = \{(K, INS) \mid K \text{ is an AHL-net and } INS \text{ is a set of instantiations of } K\ \}$$

$$\begin{aligned}
&Mor_{\mathbf{AHLNetI}}((K_1, INS_1), (K_2, INS_2)) = \\
&\quad \{(f_N, f_I) \mid f_N : K_1 \to K_2 \text{ is AHL-morphism and } f_I : INS_2 \to INS_1 \text{ is a function} \\
&\qquad\quad \text{s.t. for every } L \in INS_2 \text{ there exists an } \mathbf{INet}\text{-morphism} \\
&\qquad\quad (f_L, f_N) : (f_I(L), K_1) \to (L, K_2)\ \}
\end{aligned}$$

$$(g_N, g_I) \circ (f_N, f_I) = (g_N \circ f_N, f_I, g_I)$$

$$id_{(K,INS)} = (id_K, id_INS)$$

Furthermore we define the category of AHL-occurrence nets with instantiations as the full subcategory $\mathbf{AHLONetI} \subseteq \mathbf{AHLNetI}$ where for $(K, INS) \in Ob_{\mathbf{AHLONetI}}$ there is K an

AHL-occurrence net and for

$$INIT = \{IN(L) \mid L \in INS\}$$

there is a bijection $b : INIT \xrightarrow{\sim} INS$, i.e. $(K, INIT, INS)$ is an AHL-occurrence net with instantiations. △

The definition of the instantiation set part as the dual category of the category **SET** may be a bit counterinituitive because the functions go in the opposite direction of the direction of the corresponding morphism in **AHLNet**. The reason for that is that instantiations are reflected by AHL-morphisms whereas they are not preserved. So the fact that there is an **AHLNetI**-morphism $f = (f_N, f_I) : (K_1, INS_1) \to (K_2, INS_2)$ intuitively means that the reflection of all instantiations in INS_2 via f_N is a subset of INS_1.
The definitions of **AHLNetI** and **AHLONetI** provide well-defined categories which can be used for the categorical construction of composition, decomposition and transformation of AHL-occurrence nets with instantiations introduced in the following sections.

Fact 4.25 (Category **AHLNetI**)
AHLNetI as defined in Definition 4.24 is a category.

Proof sketch. The well-definedness of **AHLNetI** follows from the well-definedness of **AHLNet** and $\mathbf{SET}^{op}$ and the well-defined composition of **INet**-morphisms. Furthermore the associativity of the composition and the neutrality of the identity follow from the fact that **AHLNet** and $\mathbf{SET}^{op}$ are well-defined categories.
For a detailed proof see Detailed Proof C.20 in the appendix. □

Analogously to the uniqueness of instantiation morphisms there is a similar property for morphisms in **AHLNetI**.

Fact 4.26 (Uniqueness of Instantiation Set Morphisms)
Given **AHLNetI**-objects (K_1, INS_1) and (K_2, INS_2) and **AHLNetI**-morphisms $f = (f_N, f_I) : (K_1, INS_1) \to (K_2, INS_2)$ and $f' = (f_N, f'_I) : (K_1, INS_1) \to (K_2, INS_2)$ via the same **AHLNet**-morphism f_N.
Then there is $f = f'$, i.e. $f_I = f'_I$.

Proof. Let $L \in INS_2$. Due to Definition 4.24 there are **INet**-morphisms

$$(f_L, f_N) : (f_I(L), K_1) \to (L, K)$$

and

$$(f'_L, f_N) : (f'_I(L), K_1) \to (L, K)$$

which by Lemma 4.5 implies that $f_I(L) = f'_I(L)$.
Hence $f_I = f'_I$ and therefore $f = f'$. □

For every span in the category **AHLNetI** we can construct a corresponding pushout.

Fact 4.27 (**AHLNetI** has Pushouts)
Given a span $(K_1, INS_1) \xleftarrow{f} (K_0, INS_0) \xleftarrow{g} (K_2, INS_2)$ in **AHLNetI**. Then there exist **AHLNetI**-object (K_3, INS_3) together with **AHLNetI**-morphisms $f' : (K_1, INS_1) \to (K_3, INS_3)$ and $g' : (K_2, INS_2) \to (K_3, INS_3)$ such that (1) is pushout in **AHLNetI**.

$$\begin{tikzcd}
(K_0, INS_0) \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "f'"] \\
(K_2, INS_2) \arrow[r, "g'"'] & (K_3, INS_3)
\end{tikzcd}$$

Furthermore for the AHL-net K_3 of the pushout object there is (2) a pushout in **AHLNet** and INS_3 is the set

$$INS_3 = \{L_3 \mid L_1 \in INS_1, L_2 \in INS_2 \text{ with } f_I(L_1) = g_I(L_2) \text{ and (3) is pushout in } \mathbf{INet}\ \}$$

where (4) is a pullback in **SET**.
Then for all $L_3 \in INS_3$ with pushout (3) there is $f'_I(L_3) = L_1$ and $g'_I(L_3) = L_2$, i.e. $f'_I(L_3) = PreIns(f'_N)(L_3)$ and $g'_I(L_3) = PreIns(g'_N)(L_3)$.

$$\begin{tikzcd}
K_0 \arrow[r, "f_N"] \arrow[d, "g_N"'] \arrow[dr, phantom, "(2)"] & K_1 \arrow[d, "f'_N"] \\
K_2 \arrow[r, "g'_N"'] & K_3
\end{tikzcd}
\quad
\begin{tikzcd}
(L_0, K_0) \arrow[r, "(f_L, f_N)"] \arrow[d, "(g_L, g_N)"'] \arrow[dr, phantom, "(3)"] & (L_1, K_1) \arrow[d, "(f'_L, f'_N)"] \\
(L_2, K_2) \arrow[r, "(g'_L, g'_N)"'] & (L_3, K_3)
\end{tikzcd}
\quad
\begin{tikzcd}
INS_0 \arrow[dr, phantom, "(4)"] & INS_1 \arrow[l, "f_I"'] \\
INS_2 \arrow[u, "g_I"] & INS_3 \arrow[l, "g'_I"] \arrow[u, "f'_I"']
\end{tikzcd}$$

Proof sketch. We can construct the pushout (2) in **AHLNet** and the set INS_3 together with functions f'_I and g'_I as described above. For $L_1 \in INS_1$ and $L_2 \in INS_2$ the fact that $f_I(L_1) = f_I(L_2)$ implies that L_1 and L_2 have a common preimage which is a necessary condition for the construction of a pushout of instantiations L_1 and L_2. So together with Fact 4.23 this implies that INS_3 is isomorphic to the construction of pullbacks in **SET**.
Using the pushout property of (2) and pullback property of (4) we can show explicitely that (1) is a pushout in **AHLNetI**.
For a detailed proof see Detailed Proof C.21 in the appendix. □

Example 4.28 (Pushout in **AHLNetI**)
Consider the span $(I, INS_I) \xleftarrow{(f_{1,N}, f_{1,I})} (K_1, INS_1) \xrightarrow{(f_{2,N}, f_{2,I})} (K_2, INS_2)$ in **AHLNetI** where I, K_1 and K_2 are the AHL-nets in Figure 25 on page 65.
Furthermore the set INS_1 contains the instantiations depicted in Figure 27 on page 78 and INS_2 contains the instantiations depicted in Figure 28 on page 79.
The set INS_I contains instantiations depicted in Figure 24.
The morphisms $f_{1,N}$ and $f_{2,N}$ are inclusions and there is $f_I(L_{init_{1,1}}) = J_1$, $f_I(L_{init_{1,2}}) = J_2$, $f_I(L_{init_{1,3}}) = f_I(L_{init_{2,1}}) = J_3$, $f_I(L_{init_{2,2}}) = J_4$ and $f_I(L_{init_{2,3}}) = J_5$.
The pushout (K, INS) of the span $(I, INS_I) \xleftarrow{(f_{1,N}, f_{1,I})} (K_1, INS_1) \xrightarrow{(f_{2,N}, f_{2,I})} (K_2, INS_2)$ in **AHLNetI** can be constructed as pushout in **AHLNet** as shown in Figure 25 and $INS = \{L_{init}\}$ where L_{init} is the pushout in **PTNet** such that (K, L_{init}) is pushout in **INet** as shown in Figure 26 where $J = J_3$. △

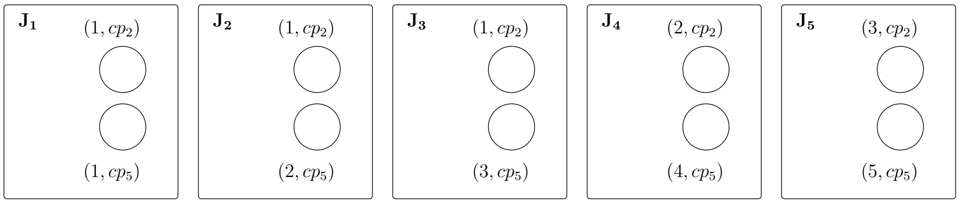


Figure 24: Instantiations of I

From every AHL-net with instantiations it is possible to obtain an AHL-net by forgetting the instantiations. Analogously from every **AHLNetI**-morphism it is possible to obtain a corresponding AHL-morphism. The combination of both forgetful constructions is a forgetful functor from **AHLNetI** to **AHLNet**.

Fact 4.29 (Forgetful Functor $Net : \mathbf{AHLNetI} \to \mathbf{AHLNet}$)
We define $Net : \mathbf{AHLNetI} \to \mathbf{AHLNet}$ with

- $Net = (Net_{Ob}, Net_{Mor})$,
- $Net_{Ob}(K, INS) = K$ for $(K, INS) \in Ob_{\mathbf{AHLNetI}}$ and
- $Net_{Mor}(f_N, f_I) = f_N$ for $(f_N, f_I) \in Mor_{\mathbf{AHLNetI}}$.

The construction Net is a functor.

Proof. *well-definedness*:
Net_{Ob} is well-defined because $(K, INS) \in Ob_{\mathbf{AHLNetI}}$ means $K \in Ob_{\mathbf{AHLNet}}$.
Let $(f_N, f_I) : (K_1, INS_1) \to (K_2, INS_2) \in Mor_{\mathbf{AHLNetI}}$. Then there is $f_N : K_1 \to K_2 \in Mor_{\mathbf{AHLNet}}$ and hence also Net_{Mor} is well-defined.

compositionality:
Let $(f_N, f_I) : (K_1, INS_1) \to (K_2, INS_2)$ and $(g_N, g_I) : (K_2, INS_2) \to (K_3, INS_3)$ in **AHLNetI**.

$$\begin{aligned} Net((g_N, g_I) \circ (f_N, f_I)) &= Net((g_N \circ f_N, f_I \circ g_I)) \\ &= g_N \circ f_N \\ &= Net(g_N, g_I) \circ f_N \\ &= Net(g_N, g_I) \circ Net(f_N, f_I) \end{aligned}$$

compatibility with identities:
Let (K, INS) in **AHLNetI**.

$$\begin{aligned} Net(id_{K,INS}) &= id_K \\ &= Net(id_K, id_{INS}) \\ &= Net(id_{(K,INS)}) \end{aligned}$$

□

On the other hand every AHL-net can be considered as a special AHL-net with instantiations where the set of instantiations is empty. Analogously every AHL-morphism is a special **AHLNetI**-morphism where the instantiation set part is the empty function. This leads to a functor from **AHLNet** to **AHLNetI**.

Fact 4.30 (Functor $Inst : \mathbf{AHLNet} \to \mathbf{AHLNetI}$)
We define $Inst : \mathbf{AHLNet} \to \mathbf{AHLNetI}$ with

- $Inst = (Inst_{Ob}, Inst_{Mor})$,
- $Inst_{Ob}(K) = (K, \emptyset)$ for $K \in Ob_{\mathbf{AHLNet}}$ and
- $Inst_{Mor}(f_N) = f$ for $f_N : K_1 \to K_2 \in Mor_{\mathbf{AHLNet}}$ where $f = (f_N, f_I)$ and $f_I : \emptyset \to \emptyset$ is the empty function.

The construction $Inst$ is a functor.

Proof. *well-definedness*:
$Inst_{Ob}$ is well-defined because $K \in Ob_{\mathbf{AHLNet}}$ means $(K, \emptyset) \in Ob_{\mathbf{AHLNetI}}$.
Let $f_N : K_1 \to K_2 \in Mor_{\mathbf{AHLNet}}$. Then there is $Inst(K_1) = (K_1, \emptyset)$ and $Inst(K_2) = (K_2, \emptyset)$ and hence $f = (f_N, f_I) : (K_1, \emptyset) \to (K_2, \emptyset) \in Mor_{\mathbf{AHLNet} \times \mathbf{SET}^{op}}$. Since there are no instantiations in $\emptyset$ there is also $f \in Mor_{\mathbf{AHLNetI}}$.

compositionality:
Let $f_N : K_1 \to K_2$ and $g_N : K_2 \to K_3$ in **AHLNet**.

$$\begin{aligned} Inst(g_N) \circ Inst(f_N) = (g_N, \emptyset) \circ (f_N, \emptyset) & \\ &= (g_N \circ f_N, \emptyset \circ \emptyset) \\ &= (g_N \circ f_N, \emptyset) \\ &= Inst(g_N \circ f_N) \end{aligned}$$

compatibility with identities:
Let K in **AHLNetI**.

$$\begin{aligned} Inst(id_K) &= (id_K, \emptyset) \\ &= (id_K, id_\emptyset) \\ &= id_{(K,\emptyset)} \\ &= id_{Inst(K)} \end{aligned}$$

$\square$

Enriching an AHL-net AN with an empty set of instantiations and forgetting the set of instantiations leads to the same AHL-net AN, i.e. the composition $Net \circ Inst$ is the identity of **AHLNet**. Therefore there is a natural transformation $id : Net \circ Inst \Rightarrow Id_{\mathbf{AHLNet}}$ which can be used to show that $Inst$ is cofree with respect to Net.

Lemma 4.31 (Natural Transformation $Net \circ Inst \Rightarrow Id_{\mathbf{AHLNet}}$)
The identities in **AHLNet** form a natural transformation $id : Net \circ Inst \Rightarrow Id_{\mathbf{AHLNet}}$.

Proof. Let AN be an AHL-net. Then there is

$$Net \circ Inst(AN) = Net(AN, \emptyset) = AN$$

Let $f : AN_1 \to AN_2$ be an AHL-morphism. Then there is

$$\begin{aligned} id_{AN_2} \circ Net \circ Inst(f) &= id_{AN_2} \circ Net(f, \emptyset) \\ &= id_{AN_2} \circ f \\ &= f \\ &= f \circ id_{AN_1} \\ &= Net(f, \emptyset) \circ id_{AN_1} \\ &= Net \circ Inst(f) \circ id_{AN_1} \end{aligned}$$

□

Fact 4.32 (*Inst* is Cofree Functor)
The functor $Inst$: **AHLNet** $\to$ **AHLNetI** is a cofree functor w.r.t. Net : **AHLNetI** $\to$ **AHLNet**.

Proof. We show that for every $A \in Ob_{\mathbf{AHLNet}}$ there is $(Inst(A), id_A)$ a cofree construction w.r.t. Net.
Let $A \in Ob_{\mathbf{AHLNet}}$ and let $(B, INS_B) \in Ob_{\mathbf{AHLNetI}}$ together with an AHL-morphism $f : Net(B, INS_B) \to A$.
There is $Net(B, INS_B) = B$, i.e. $f : B \to A$.
We define

$$f^* : Inst(A) \to (B, INS_B)$$

with $f^* = (f, f_I)$ where $f_I : \emptyset \to INS_B$ is the empty function.
Since $Inst(A) = (A, \emptyset)$ the morphism $f^* : Inst(A) \to (B, INS_B)$ is a well-defined **AHLNetI**-morphism. Furthermore there is

$$id_A \circ Net(f^*) = id_A \circ f = f$$

Let $\bar{f} = (\bar{f}_N, \bar{f}_I) : Inst(A) \to (B, INS_B)$ an **AHLNetI**-morphism with

$$id_A \circ Net(\bar{f}) = f$$

There is $Net(\bar{f}) = \bar{f}_N$ which means that

$$\bar{f}_N = Net(\bar{f}) = id_A \circ Net(\bar{f}) = f$$

Due to the initiality of $\emptyset$ in **SET** there is $\bar{f}_I : \emptyset \to INS_B$ the empty function, i.e. $\bar{f}_I = f_I$ and hence $\bar{f} = f^*$. □

We can use the cofreeness of $Inst$ to lift process morphisms $mp : K \to AN$ for AHL-occurrence nets with instantiations $KI = (K, INIT, INS)$ to process morphisms with instantiations.

Definition 4.33 (Category $\mathbf{AHLProcI(AN)}$)
Given an AHL-net AN. The category $\mathbf{AHLProcI(AN)}$ of instantiated AHL-processes of AN is defined as the full subcategory of the slice category $\mathbf{AHLNetI} \setminus Inst(AN)$ where for objects $(mp_N, mp_I) : (K, INS) \to Inst(AN)$ the domain (K, INS) is an **AHLONetI**-object. $\triangle$

5 Composition of Algebraic High-Level Processes

In section 3 a composition of high-level pretri net processes has been defined where the output places of one process are glued to the input places of another process. This allows a combination of parallel and sequential composition of two processes.
In this section we extend the composition such that it is not only possible to compose the processes sequentially or parallel but also in an interleaving way.

5.1 Composition of Algebraic High-Level Processes

In the case of sequential composition the fact that the causal relation of the result is a finitary strict partial order is basically ensured by the fact that only output places of the one net are glued to input places of the other net. But in the case of general composition there is not such a restriction for the gluing of places. So in order to check whether the result of the composition has any cycles or it is not finitary we define the induced causal relation over the places in the composition interface I. The relation is derived from the causal relation of the images of that places in the occurrence nets K_1 and K_2.

Definition 5.1 (Induced Causal Relation for Composition)
Given the AHL-occurrence nets K_1, K_2 and I with $T_I = \emptyset$ and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
We define the relation $\prec_{(i_1,i_2)}$ in the following way:

$$\prec_{(i_1,i_2)} = \{(x, y) \in P_I \times P_I | i_1(x) <_{K_1} i_1(y) \vee i_2(x) <_{K_2} i_2(y)\}$$

The transitive closure $<_{(i_1,i_2)}$ of $\prec_{(i_1,i_2)}$ is called induced causal relation of (I, i_1, i_2). △

The induced causal relation in Definition 5.1 is defined completely analogously to the one defined in [BCEH01] where it is used for the composition of open low-level processes.
The composability of AHL-occurrence nets requires additionally to the irreflexivity and finitarity of the induced causal relation that the composition will produce no conflicts. Therefore for the gluing of two places it is required that at most one of them is in the pre domain of a transition and at most one of them is in the post domain of a transition. This means that if one of the glued places is no input places because it is in the post domain of a transition the other one has to be an input place. Analogously if one of the glued places is no output place because it is in the pre domain of a transition the other one has to be an output place.

Definition 5.2 (Composability of AHL-Occurrence Nets)
Given the AHL-occurrence nets
$K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$
and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$
and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Then (K_1, K_2) are composable w.r.t. (I, i_1, i_2) if

- for all $x \in P_I$:
  $i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2)$ and
  $i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$
- the induced causal relation $<_{(i_1,i_2)}$ is a finitary strict partial order

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] & K_1 \\ K_2 & \end{tikzcd}$$

△

The sequential composability of AHL-occurrence nets is a special case of the composability of AHL-occurrence nets.

Fact 5.3 (Sequential Composability of AHL-Occurrence Nets)
Given the AHL-occurrence nets $K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$ and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$ and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
If (K_1, K_2) are sequential composable w.r.t. (I, i_1, i_2) then
(K_1, K_2) are composable w.r.t. (I, i_1, i_2).

Proof. Let (K_1, K_2) be sequential composable w.r.t. (I, i_1, i_2).

$i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2)$:
The implication is true because its conclusion is always true since $i_2(P_I) \subseteq IN(K_2)$.

$i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$:
The implication is true because its premise is always false since $i_1(P_I) \subseteq OUT(K_1)$.

induced causal relation:
Let $x, y \in P_I$. Then there is $i_1(x), i_1(y) \in OUT(K_1)$ which implies that neither $i_1(x)$ nor $i_1(y)$ is in the pre domain of a transition and hence $i_1(x) \not<_{K_1} i_1(y)$ and $i_1(y) \not<_{K_1} i_1(x)$. Furthermore there is $i_2(x), i_2(y) \in IN(K_2)$ which implies that neither $i_2(x)$ nor $i_2(y)$ is in the post domain of a transition and hence $i_2(x) \not<_{K_2} i_2(y)$ and $i_2(y) \not<_{K_2} i_2(x)$. This means that $x \not<_{(i_1,i_2)} y$ and $y \not<_{(i_1,i_2)} x$ and hence $<_{(i_1,i_2)}$ is empty which implies that it is finitary and irreflexive.

□

In contrast to the sequential composability in the case of the general composability the order of the nets K_1 and K_2 is not important.

Lemma 5.4 (Symmetry of Composability)
Given the AHL-occurrence nets
$K_x = (SP, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 1, 2$
and $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$ with $T_I = \emptyset$
and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Then (K_1, K_2) are composable w.r.t. (I, i_1, i_2) iff (K_2, K_1) are composable w.r.t. (I, i_2, i_1).

Proof sketch. The first condition of the composability is the contraposition of the composability in the respective other direction. Furthermore the induced causal relation $<_{(i_1,i_2)}$ equals the induced causal relation $<_{(i_2,i_1)}$ due to the commutativity of the logical disjunction.
For a detailed proof see Detailed Proof C.22 in the appendix. □

For two AHL-occurrence nets K_1 and K_2 the composability of (K_1, K_2) is a sufficient and necessery condition that the composition is an AHL-occurrence net.

Theorem 5.5 (Composition of AHL-Occurrence Nets)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
If and only if (K_1, K_2) are composable w.r.t. (I, i_1, i_2) then for the pushout diagram (PO) in the category **AHLNet** the pushout object K, with $K = K_1 \circ_{(I,i_1,i_2)} K_2$ is an AHL-occurrence net.
K is called composition of (K_1, K_2) w.r.t. (I, i_1, i_2) and the diagram (PO) is called a composition diagram.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "\mathbf{(PO)}"] & K_1 \arrow[d, "i_1'"] \\ K_2 \arrow[r, "i_2'"'] & K \end{tikzcd}$$

Proof sketch.

"$\Rightarrow$":
The net K is unary and it has no forward or backward conflicts because K_1 and K_2 are AHL-occurrence nets and AHL-morphisms preserve pre and post conditions.
The causal relation of the net K is derived from the causal relations of the nets K_1 and K_2 and additionally it is possible that places or transitions originated in different nets are causal related in K due to the gluing of places. If two different gluing places are causal related in the net K their preimages in the interface are related in the induced causal relation. So the fact that the causal relations $<_{K_1}$ and $<_{K_2}$ and the induced causal relation $<_{(i_1,i_2)}$ are finitary strict partial orders implies that also $<_K$ is a finitary strict partial order.

"$\Leftarrow$":
Since K is an AHL-occurrence net it has no forward or backward conflicts. So for $p_1 \in P_{K_1}$ and $p_2 \in P_{K_2}$ which are glued together in the net K, i.e. if there is $p_0 \in P_I$ with $p_1 = i_1(p_0)$ and $p_2 = i_2(p_0)$ then there is at most one of the places p_1 and p_2 not an input place or not an output place, respectively.
Since for places $x, y \in P_I$ with $x <_{(i_1,i_2)} y$ there is also $i_1'(i_1(x)) <_K i_1'(i_1(y))$ the fact that $<_K$ is a finitary strict partial order implies that also $<_{(i_1,i_2)}$ is a finitary strict partial order.

For a detailed proof see Detailed Proof C.23 in the appendix. □

Example 5.6 (Composition of AHL-Occurrence Nets)
Fig. 25 shows a composition diagram. The net AHL-occurrence net K_1 contains two concurrent transitions *heat up*$_1$ and *turn on*$_1$. The AHL-occurrence net K_2 contains an *increase volume*$_1$ transition between the two transitions *next phase*$_1$ and *next phase*$_2$. For inclusions $i_1 : I \to K_1$ and $i_2 : I \to K_2$ the nets (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Due to the composition $K = K_1 \circ_{(I,i_1,i_2)} K_2$ the net K_2 is glued into the net K_1 such that

the transitions *heat up*$_1$ and *turn on*$_1$ are not concurrent in the AHL-occurrence net K but there is a specific order of consecutive phases of heating, increasing the volume and turning on the light.

△

Corollary 5.7 (Pushout of AHL-Occurrence Nets)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Then pushout (1) in **AHLNet** is also pushout in **AHLONet** iff (K_1, K_2) are composable w.r.t. (I, i_1, i_2).

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd}$$

Proof. "⇒":
Let (1) be a pushout in the categories **AHLNet** and **AHLONet**. Then K is an AHL-occurrence net which by Theorem 5.5 implies that (K_1, K_2) are composable w.r.t. (I, i_1, i_2).

"⇐":
Let (K_1, K_2) be composable w.r.t. (I, i_1, i_2) and (1) pushout in **AHLNet**. Then by Theorem 5.5 the net K is an AHL-occurrence net.
Let X be an AHL-occurrence net with morphisms $x_1 : K_1 \to X$ and $x_2 : K_2 \to X$ with

$$x_1 \circ i_1 = x_2 \circ i_2$$

Due to the fact that AHL-occurrence nets are AHL-nets and (1) is a pushout in the category **AHLNet** there exists a unique morphism $x : K \to X$ with

$$x \circ i'_1 = x_1$$

and

$$x \circ i'_2 = x_2$$

Since K and X are AHL-occurrence nets the morphism $x : K \to X$ is a morphism in the category **AHLONet**. Hence (1) is pushout in the category **AHLONet**.

□

In order to compose not only AHL-occurrence nets but also AHL-processes it is not only necessary that the AHL-occurrence nets are composable but additionally it is necessary that the process morphisms are compatible with the composition of AHL-occurrence nets such that there is also a composition of the AHL-processes. This requirement is called composability of AHL-processes.

Definition 5.8 (Composability of AHL-Processes)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism

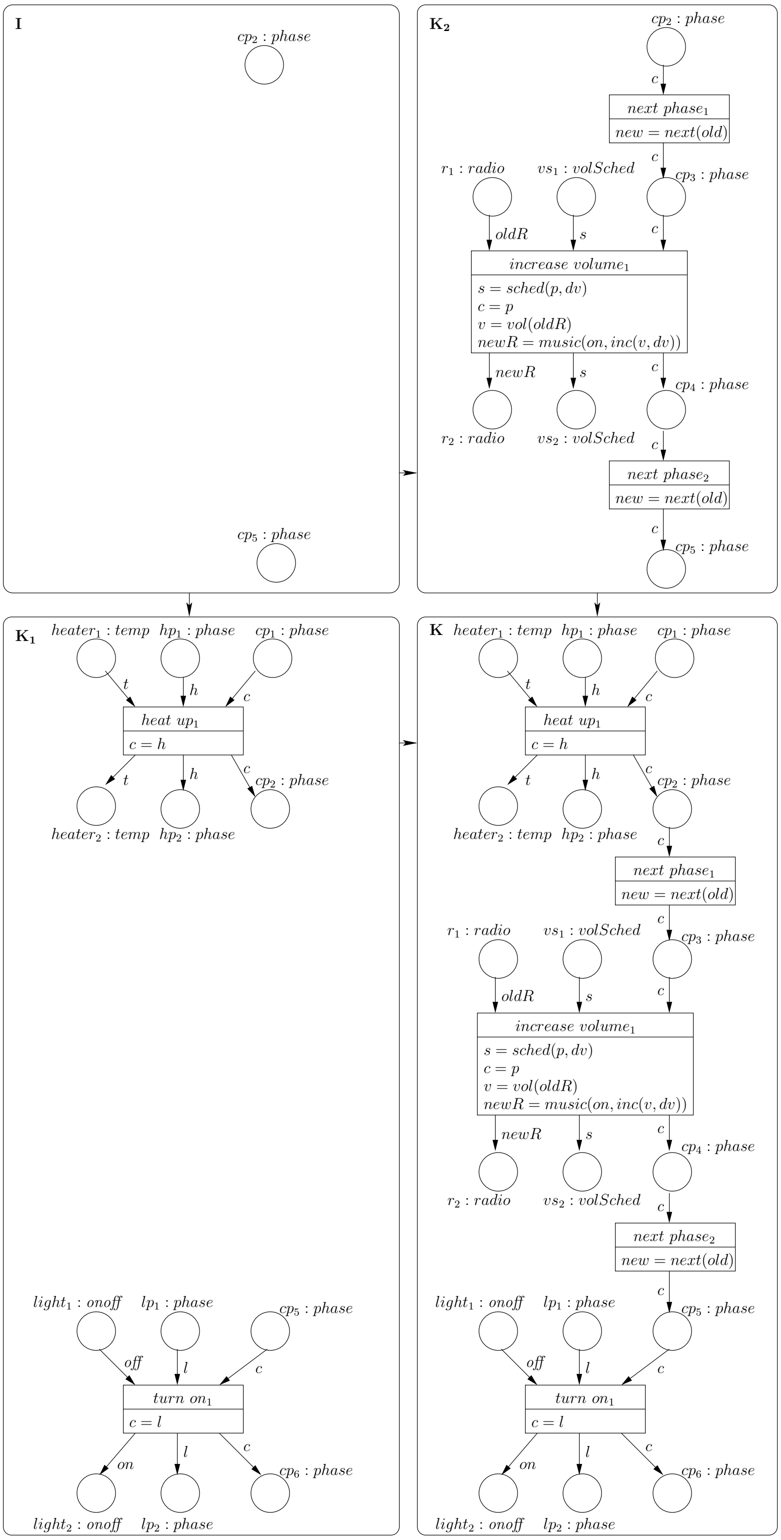


Figure 25: Composition of AHL-Occurrence Nets $K = K_1 \circ_{(I,i_1,i_2)} K_2$

$i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let K_x together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two AHL-processes of the AHL-net AN.
Then (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2) if

1. (K_1, K_2) are composable w.r.t. (I, i_1, i_2),
2. $mp_1 \circ i_1 = mp_2 \circ i_2$.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(=)"] & K_1 \arrow[d, "mp_1"] \\ K_2 \arrow[r, "mp_2"'] & AN \end{tikzcd}$$

△

The composition of AHL-processes can be computed by the composition of the respective AHL-occurrence nets which induces a unique process morphism which is compatible with the composed morphisms.

Theorem 5.9 (Composition of AHL-Processes)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let K_x together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two AHL-processes of the AHL-net AN such that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).
Then the composition $K = K_1 \circ_{(I,i_1,i_2)} K_2$ together with the induced AHL-net morphism $mp : K \to AN$ is an AHL-process of the AHL-net AN.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \arrow[ddr, bend left, "mp_1"] & \\ K_2 \arrow[r, "i'_2"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\ & & AN \end{tikzcd}$$

The composition is denoted $mp = mp_1 \circ_{(I,i_1,i_2)} mp_2$.

Proof. Due to the composability of (mp_1, mp_2) w.r.t. (I, i_1, i_2) there is (K_1, K_2) are composable w.r.t. (I, i_1, i_2) which implies that the composition K of the AHL-occurrence nets is an AHL-occurrence net and hence K together with $mp : K \to N$ is an AHL-process, where mp is the unique morphism induced by Pushout (1) with

$$mp = (mp_P, mp_T)$$
$$mp_P(p) = \begin{cases} mp_{1P}(p_1) & p = i'_{1P}(p_1) \\ mp_{2P}(p_2) & p = i'_{2P}(p_2) \end{cases}$$
$$mp_T(t) = \begin{cases} mp_{1T}(t_1) & t = i'_{1T}(t_1) \\ mp_{2T}(t_2) & t = i'_{2T}(t_2) \end{cases}$$

□

Example 5.10 (Composition of AHL-Processes)
Consider the composition $K = K_1 \circ_{(I,i_1,i_2)} K_2$ of AHL-occurrence nets in Figure 25. Moreover let $mp_1 : K_1 \to Alarm$ and $mp_2 : K_2 \to Alarm$ be two process morphisms where $Alarm$ is the AHL-net in Figure 1 and mp_1, mp_2 map elements in K_1 and K_2 to elements with the same name but without index in the net $Alarm$.
The processes (mp_1, mp_2) are composable with respect to (I, i_1, i_2) and there is a unique induced morphism $mp : K \to Alarm$ which maps the elements in K exactly in the same way as the morphisms mp_1 and mp_2 map these elements in K_1 and K_2, respectively. The process $mp : K \to Alarm$ is the composition of mp_1 and mp_2. △

The composition of AHL-processes of an AHL-net AN as pushout of the AHL-occurrence nets in the category **AHLNet** is also a pushout in the category **AHLProc**(**AN**) of processes of AN.

Corollary 5.11 (Pushout of AHL-Processes)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let K_x together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two AHL-processes of the AHL-net AN such that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).
Then there exist $mp_0 : I \to AN$ and $mp : K \to AN$ in **AHLProc**(**AN**) such pushout (1) in **AHLNet** is also pushout in **AHLProc**(**AN**) iff (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] \arrow[ddrr, bend right=60, "mp_0"'] & K_1 \arrow[d, "i_1'"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "i_2'"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\
 & & AN
\end{tikzcd}$$

Proof. "⇒":
Given pushout (1) in **AHLNet** and also in **AHLProc**(**AN**) then K is an AHL-occurrence net which by Theorem 5.5 implies that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).

"⇐":
Given pushout (1) in **AHLNet**. Since injective AHL-morphisms reflect AHL-processes and there is $i_1 : I \to K_1$ we obtain an AHL-process morphism $mp_0 : I \to AN$ with $i_1 \circ mp_1 = mp_0$ and hence i_1 is an **AHLProc**(**AN**)-morphism. Then there is

$$mp_0 = i_1 \circ mp_1 = i_2 \circ mp_2$$

because (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2) which implies that also i_2 is an **AHLProc**(**AN**)-morphism.
Due to the pushout property the fact that $mp_1 \circ i_1 = mp_2 \circ i_2$ implies a unique AHL-morphism $mp : K \to AN$ with $mp \circ i_1' = mp_1$ and $mp \circ i_2' = mp_2$. This means that i_1' and i_2' are **AHLProc**(**AN**) morphisms because K is an AHL-occurrence net by Theorem 5.5.

Let $mp_X : X \to AN$ in **AHLProc**(**AN**) together with **AHLProc**(**AN**)-morphisms $x_1 : K_1 \to X$ and $x_2 : K_2 \to X$ such that

$$x_1 \circ i_1 = x_2 \circ i_2$$

Due to the fact that **AHLProc**(**AN**) is a subcategory of **AHLNet**$\backslash AN$ the morphisms x_1 and x_2 are AHL-morphisms. So the pushout property of (1) in **AHLNet** implies a unique morphism $x : K \to X$ with $x \circ i'_1 = x_1$ and $x \circ i'_2 = x_2$.
Then we have

$$\begin{aligned} mp_X \circ x \circ i'_1 &= mp_X \circ x_1 \\ &= mp_1 \end{aligned}$$

and

$$\begin{aligned} mp_X \circ x \circ i'_2 &= mp_X \circ x_2 \\ &= mp_2 \end{aligned}$$

which due to the uniqueness of mp implies that $mp = mp_X \circ x$. Hence x is also a morphism in **AHLNet** $\setminus AN$.
Since **AHLProc**(**AN**) is a full subcategory of **AHLNet** $\setminus AN$ and K and X are AHL-occurrence nets the morphism x is also an **AHLProc**(**AN**)-morphism and hence (1) is pushout in **AHLProc**(**AN**).

□

5.2 Composition of Instantiations

The composability of instantiations remains basically the same as the sequential composability of instantiations with the difference that the corresponding high level nets do not have to be sequential composable and therefore the composability does not only consider input and output places but all places of the instantiations which are matched from the interface.

Definition 5.12 (Composability of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$, $L_{init_2} \in INS_2$.
Then (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) if for all $(a, p) \in A \otimes P_I$:

$$(a, i_1(p)) \in P_{L_{init_1}} \Rightarrow (a, i_2(p)) \in P_{L_{init_2}}$$

The set of all composable instantiations of KI_1 and KI_2 is defined as

$$\begin{aligned} Composable_{(I,i_1,i_2)}(INS_1, INS_2) \;=\; & \{(L_{init_1}, L_{init_2}) \in INS_1 \times INS_2 \mid \\ & (L_{init_1}, L_{init_2}) \text{ are composable w.r.t.} \\ & (J, j_1, j_2) \text{ induced by } (I, i_1, i_2)\} \end{aligned}$$

△

The sequential compositability of instantiations is a special case of the composability of instantiations.

Fact 5.13 (Sequential Composability of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2).
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and let $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$.
Then (L_{init_1}, L_{init_2}) are sequentially composable w.r.t. (I, i_1, i_2) iff (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2).

Proof sketch. Both the sequential and general composition require that the places matched by the interface at the high-level layer have the same data elements at the instantiation layer. So in the case that (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) the composability of instantiations of (K_1, K_2) is equal to the sequential composability of instantiations of (K_1, K_2).
For a detailed proof see Detailed Proof C.24 in the appendix. □

Not only the composability of AHL-occurrence nets but also the composability of instantiations is symmetric.

Lemma 5.14 (Symmetry of Composability of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$, $L_{init_2} \in INS_2$.
Then (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) implies that (L_{init_2}, L_{init_1}) are composable w.r.t. (I, i_2, i_1).

Proof. Let (K_1, K_2) be composable w.r.t. (I, i_1, i_2). From Lemma 5.4 follows that (K_2, K_1) are composable w.r.t. (I, i_2, i_1).
Let (L_{init_1}, L_{init_2}) be composable w.r.t. (I, i_1, i_2).
Let $(a, p) \in A \otimes P_I$ and $(a, i_2(p)) \in P_{L_{init_2}}$.
Due to the bijection $proj(K_1) \in in_1$ there is one data element $a' \in A_{type(i_1(p))}$ with

$$(proj(K_1) \circ in)^{-1}(i_1(p)) = (a', i_1(p)) \in P_{L_{init_1}}$$

The composability of (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2) implies

$$(a', i_2(p)) \in P_{L_{init_2}}$$

and we have

$$proj(K_2) \circ in_2(a', i_2(p)) = i_2(p) = proj(K_2) \circ in_2(a, i_2(p))$$

which implies $a' = a$ because $proj(K_2) \circ in_2$ is injective.
Hence there is $(a, i_1(p)) \in P_{L_{init_1}}$. □

In Section 4 we introduced the concept of compatibility of instantiations which is a sufficient and necesserary condition for the existence of an instantiation interface (see Theorem 4.18). Therefore it is very useful that the notions of composability and compatibility are equivalent to each other.

Lemma 5.15 (Equivalence of Composability and Compatibility)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$.
Then (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) iff (L_{init_1}, L_{init_2}) are compatible with (i_1, i_2).

Proof sketch. The images $I_1 = i_1(I)$ and $I_2 = i_2(I)$ lead to epi-mono-factorizations (I_1, e_1, m_1) of i_1 and (I_2, e_2, m_2) of i_2. For $J_1 = PreIns(m_1)(L_{init_1})$ and $J_2 = PreIns(m_2)(L_{init_2})$ the composability of (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2) requires that J_1 and J_2 have the same data elements on corresponding places. This requirement is also a necessary and sufficient condition for the existence of an **INet**-morphism $c = (c_L, c_H) : (J_1, I_1) \to (J_2, I_2)$ such that $c_H \circ e_1 = e_2$ which is equivalent to the fact that (L_{init_1}, L_{init_2}) are compatible with (i_1, i_2).
For a detailed proof see Detailed Proof C.25 in the appendix. □

For the composition of two AHL-occurrence nets with composable instantiation we obtain a unique corresponding composition of these instantiations using the special uniqueness of pushouts in the category **INet**.

Theorem 5.16 (Composition of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are composable w.r.t. (I, i_1, i_2). Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ such that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2).
Then there exists a unique P/T-net J together with **PTNet**-morphisms $j_1 : J \to L_{init_1}, j_2 : J \to L_{init_2}$ such that $(J, I), (j_1, i_1)$ and (j_2, i_2) are in **INet**.
Furthermore for pushout (1) in **AHLNet** there exists pushout (2) in **PTNet** with a unique pushout object L s.t. (3) is pushout in **INet**.
Then L is called the composition of L_{init_1} and L_{init_2} w.r.t. (I, i_1, i_2) and is denoted by $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$.
Under these conditions the diagram (2) is called a composition diagram of instantiations w.r.t. composition diagram (1).

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(2)"] & L_{init_1} \arrow[d, "j'_1"] \\ L_{init_2} \arrow[r, "j'_2"'] & L_{init} \end{tikzcd} \qquad \begin{tikzcd} (J,I) \arrow[r, "{(j_1,i_2)}"] \arrow[d, "{(j_2,i_2)}"'] \arrow[dr, phantom, "(3)"] & (L_{init_1}, K_1) \arrow[d, "{(j'_1,i'_1)}"] \\ (L_{init_2}, K_2) \arrow[r, "{(j'_2,i'_2)}"'] & (L_{init}, K) \end{tikzcd}$$

Proof. By Lemma 5.15 the composability of (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2) implies the compatibility of (L_{init_1}, L_{init_2}) with (i_1, i_2) which by Theorem 4.18 implies that there is a unique instantiation J of I together with morphisms $j_1 : J \to L_{init_1}, j_2 : J \to L_{init_2}$ s.t. (J, I), (j_1, i_1) and (j_2, i_2) are in **INet**. So the pushout of (L_1, K_1) and (L_2, K_2) w.r.t. (J, I) in the

category **INet** exists and it can be constructed componentwise in **AHLNet** and **PTNet** which means that there is pushout (2) in **PTNet** and pushout (3) in **INet**.
The uniqueness of L follows from pushout (3) and Fact 4.22. □

Example 5.17 (Composition of Instantiations)
Fig. 26 shows a composition diagram of instantiations w.r.t. the composition diagram depicted in Fig. 25. The instantiations (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) because they have the same data on the matched places. Therefore the induced instantiation interface J with morphisms j_1 and j_2 exists and the composition of instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ is an instantiation of $K = K_1 \circ_{(I,i_1,i_2)} K_2$.
△

Fact 5.18 (Sequential Composition of Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$ such that (K_1, K_2) are composable w.r.t. (I, i_1, i_2). Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ such that (L_{init_1}, L_{init_2}) are sequential composable w.r.t. (I, i_1, i_2).
Then the sequential composition $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ where (J, j_1, j_2) is induced by (I, i_1, i_2) is a composition of instantiations.

Proof. Fact 5.3 implies that (K_1, K_2) are composable w.r.t. (I, i_1, i_2) and Fact 5.13 implies that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2).
The definition of the instantiation interface (J, j_1, j_2) in Definition 3.6 is exactly the instantiation preimage $PreIns(i_1)(L_{init_1})$ and hence the sequential composition of instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ is exactly the composition of L_{init_1} and L_{init_2} w.r.t. (I, i_1, i_2). □

For a given composition $K_1 \circ_{(I,i_1,i_2)} K_2$ of AHL-occurrence nets K_1 and K_2 the composition of instantiations of K_1 and K_2 does not only behave functional but also injective.

Lemma 5.19 (Composition of Instantiations is Injective)
Given AHL-occurrence nets with instantiation $KI_1 = (K_1, INIT_1, INS_1)$ and
$KI_2 = (K_2, INIT_2, INS_2)$ s.t. (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Let $L_{init_1}, L_{init_3} \in INS_1$ and $L_{init_2}, L_{init_4} \in INS_2$ s.t. (L_{init_1}, L_{init_2}) and (L_{init_3}, L_{init_4}) are composable w.r.t. (I, i_1, i_2). Then

$$L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} = L_{init_3} \circ_{(J',j_3,j_4)} L_{init_4} \Rightarrow (L_{init_1}, L_{init_2}) = (L_{init_3}, L_{init_4})$$

Proof. Follows directly from Theorem 5.16 and Fact 4.23. □

5.3 Composition of AHL-Processes with Instantiations

Based on the composition of AHL-processes and the composition of instantiations we define the composition of AHL-occurrence nets with instantiations. Due to the non-sequential composition it is possible that different compositions of instantiations lead to different instantiations but the same initializations. For that reason we define the retained input places and

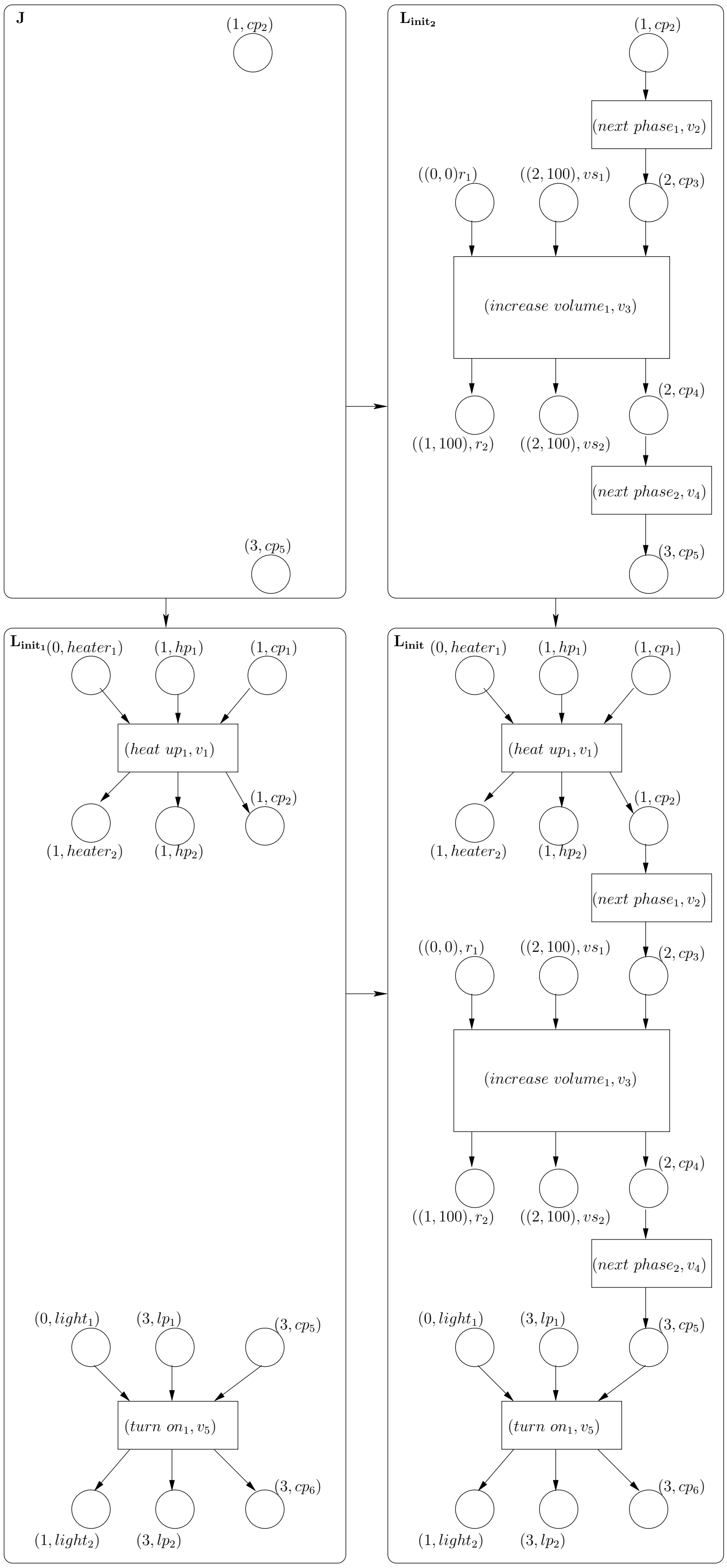


Figure 26: Composition of Instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$

retained initializations which allows us to check whether there is a bijection between composed instantiations and corresponding initializations before the composition is computed.
For AHL-occurrence nets K_1 and K_2 which are composable w.r.t. (I, i_1, i_2) the retained input places $Ret_{(I,i_1,i_2)}(K_1)$ and $Ret_{(I,i_1,i_2)}(K_2)$ are the input places of K_1 and K_2 which are also input places in the result of the composition $K_1 \circ_{(I,i_1,i_2)} K_2$.
For instantiations L_{init_1} of K_1 and L_{init_2} of K_2 which are composable w.r.t. (J, j_1, j_2) induced by (I, i_1, i_2) the retained initializations $Ret_{(I,i_1,i_2)}(L_{init_1})$ and $Ret_{(I,i_1,i_2)}(L_{init_2})$ are the input places of these instantiations which are also input places of the composition $L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$.
Finally for $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ the set of retained initialization pairs $RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$ contains for every initialization $init$ of $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ a set of retained input places R such that the elements in $init$ are images of elements in R.

Lemma 5.20 (Retained Input Places)
Given AHL-occurrence nets K_1, K_2 s.t. (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Let BI be the set of all places in the interface I which match to input places in both nets K_1 and K_2:

$$BI = \{p \in P_I \mid i_1(p) \in IN(K_1) \text{ and } i_2(p) \in IN(K_2)\}$$

For $x \in \{1, 2\}$ we define

$$RI_x = (IN(K_x) \setminus i_x(P_I)) \cup i_x(BI)$$

together with inclusions $ri_x : RI_x \to P_{K_x}$.
Then for composition diagram (1)

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd}$$

there is

$$IN(K) = [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2)$$

where $[i'_1 \circ ri_1, i'_2 \circ ri_2] : RI_1 + RI_2 \to P_K$ is the induced coproduct morphism such that (2) and (3) commute.

$$\begin{tikzcd} RI_1 \arrow[r, "\iota_1"] \arrow[dr, "i'_1 \circ ri_1"'] & RI_1 + RI_2 \arrow[d, "{[i'_1 \circ ri_1, i'_2 \circ ri_2]}"] & RI_2 \arrow[l, "\iota_2"'] \arrow[dl, "i'_2 \circ ri_2"] \\ & P_K & \end{tikzcd}$$

RI_x is called the set of retained input places of K_x w.r.t. (I, i_1, i_2), written

$$RI_x = Ret_{(I,i_1,i_2)}(K_x)$$

Proof sketch. For $x \in \{1, 2\}$ the set RI_x defines exactly the input places in K_x which are not glued to a place in the respective other net which is not an input place. This means that RI_x contains exactly the input places p of K_x which have an image $i'_x(p)$ that is also an input

place in the net K. Hence the image of the mediating morphism $[i'_1 \circ ri_1, i'_2 \circ ri_2]$ is the set of input places of the net K.
For a detailed proof see Detailed Proof C.26 in the appendix. □

Lemma 5.21 (Retained Initializations)
Given AHL-occurrence nets with instantiation $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ s.t. (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Let $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ be composable w.r.t. (I, i_1, i_2) and (J, j_1, j_2) the instantiation interface induced by (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2).
Let BJ be the set of all places in the instantiation interface J which match to input places in both nets L_{init_1} and L_{init_2}:

$$BJ = \{p \in P_J \mid j_1(p) \in IN(L_{init_1}) \text{ and } j_2(p) \in IN(L_{init_2})\}$$

Then we define for $x \in \{1, 2\}$

$$RJ_x = (init_x \setminus j_x(P_J)) \cup j_x(BJ)$$

together with inclusions $rj_x : RJ_x \to P_{L_{init_x}}$.
Then there is a bijection

$$r_x : RJ_x \to RI_x$$

with

$$r_x(a, p) = p$$

and for the composition diagram of instantiations (2) w.r.t. composition diagram (1)

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \\ K_2 \arrow[r, "i'_2"'] & K \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(2)"] & L_{init_1} \arrow[d, "j'_1"] \\ L_{init_2} \arrow[r, "j'_2"'] & L \end{tikzcd}$$

there is

$$IN(L) = [j'_1 \circ rj_1, j'_2 \circ rj_2](RJ_1 + RJ_2)$$

where $[j'_1 \circ rj_1, j'_2 \circ rj_2] : RJ_1 + RJ_2 \to P_L$ is the induced coproduct morphism such that (3) and (4) commute.

$$\begin{tikzcd} RJ_1 \arrow[r, "\kappa_1"] \arrow[dr, "j'_1 \circ rj_1"'] & RJ_1 + RJ_2 \arrow[d, "{[j'_1 \circ rj_1, j'_2 \circ rj_2]}"] & RJ_2 \arrow[l, "\kappa_2"'] \arrow[dl, "j'_2 \circ rj_2"] \\ & P_L & \end{tikzcd}$$

RJ_x is called the set of retained initializations of L_{init_x} w.r.t. (I, i_1, i_2), written

$$RJ_x = Ret_{(I, i_1, i_2)}(L_{init_x})$$

Proof sketch. For $x \in \{1,2\}$ the set RJ_x defines exactly the places (a,p) of L_{init_x} such that $p \in RI_x$. Therefore we get a bijection $r_x : RJ_x \to RI_x$ by restricting the bijective projection $(proj(K_x) \circ in_x)_P$ to the set RJ_x.
So we can use the properties of **INet**-morphisms (j'_1, i'_1) and (j'_2, i'_2) and the fact that

$$[i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2) = IN(K)$$

to obtain that

$$[j'_1 \circ rj_1, j'_2 \circ rj_2](RJ_1 + RJ_2) = IN(L)$$

For a detailed proof see Detailed Proof C.27 in the appendix. □

Definition 5.22 (Retained Initialization Pairs)
Given AHL-occurrence nets with instantiation $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ s.t. (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Then the set of retained initialization pairs of (INS_1, INS_2) w.r.t. (I, i_1, i_2) is defined by $RetInit_{(I,i_1,i_2)}(INS_1, INS_2) =$

$$\{RJ_1 \uplus RJ_2 \mid (L_{init_1}, L_{init_2}) \text{ are composable w.r.t. } (I, i_1, i_2) \text{ and } RJ_1 = Ret_{(I,i_1,i_2)}(L_{init_1}) \text{ and } RJ_2 = Ret_{(I,i_1,i_2)}(L_{init_2})\}$$

△

The retained initialization pairs together with the fact that the composition of instantiations with respect to a given composition of AHL-occurrence nets behaves functional and injective provide a necessery and sufficient condition whether the result of the composition of two AHL-occurrence nets with instantiations is an AHL-occurrence net with instantiations.

Definition 5.23 (Composability of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations. Then (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2) iff

1. (K_1, K_2) are composable w.r.t. (I, i_1, i_2) and
2. $|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$.

△

There is no sequential counterpart of the composability of AHL-occurrence nets with instantiations in Section 3. The reason for that is the fact that the sequential composability of two AHL-occurrence nets K_1 and K_2 with respect to an interface (I, i_1, i_2) is sufficient for the fact that for any sets of initializations and instantiations such that $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ are AHL-occurrence nets with instantiations there are (KI_1, KI_2) composable w.r.t. (I, i_1, i_2).

Fact 5.24 (Sequential Composability of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism

$i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations. If (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) then (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2).

Proof sketch. Due to the sequential composability of K_1 and K_2 all input places of K_1 are also input places of K and hence for composable $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ there is

$$Ret_{(I,i_1,i_2)}(L_{init_1}) = init_1$$

So different initialization pairs correspond to different instantiations of the net K_1 due to the bijective correspondence between initializations and instantiations in AHL-occurrence nets with instantiations.
Since the input places of L_{init_2} which are no retained initialization are completely determined by the output places of L_{init_1} different initialization pairs imply also different corresponding instantiations in the net K_2.
So together with the fact that for every pair of composable instantiations there is a pair of retained initialization pairs this means that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

Furthermore by Fact 5.3 there are (K_1, K_2) composable w.r.t. (I, i_1, i_2) which means that (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2).
For a detailed proof see Detailed Proof C.28 in the appendix. □

The composability of AHL-occurrence nets with instantiations is a necessary and sufficient condition that the composition leads to an AHL-occurrence net with instantiations. Similar to the sequential composition of AHL-occurrence nets with instantiations the general composition of AHL-occurrence nets with instantiations means the composition of the AHL-occurrence nets together with the set of all compositions of composable instantiations and the set of their respective initializations.

Theorem 5.25 (Composition of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations. If and only if (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2) then $KI = (K, INIT, INS)$ with

- $K = K_1 \circ_{(I,i_1,i_2)} K_2$, with composition diagram (1),
- $INS =$
  $\{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2),$
  (J, j_1, j_2) is induced instantiation interface w.r.t. $(I, i_1, i_2)\}$,
- $INIT = \{IN(L) \mid L \in INS\}$,

is an AHL-occurrence net with instantiations.
For every composition of instantiations $L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \in INS$ diagram (2) has to be a composition diagram of instantiations w.r.t. composition diagram (1).
Then KI is called composition of (KI_1, KI_2) w.r.t. (I, i_1, i_2), written $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$.

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "i_1'"] \\ K_2 \arrow[r, "i_2'"'] & K \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(\mathbf{2})"] & L_{init_1} \arrow[d, "j_1'"] \\ L_{init_2} \arrow[r, "j_2'"'] & L \end{tikzcd}$$

Proof sketch. There is a bijection

$$f : Composable_{(I,i_1,i_2)}(INS_1, INS_2) \to INS$$

because for every $L_{init} \in INS$ there are unique $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ such that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) and L_{init} is the composition of L_{init_1} and L_{init_2}. Furthermore the properties of retained initializations imply that there is a surjection

$$g : Composable_{(I,i_1,i_2)}(INS_1, INS_2) \to RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$$

and a bijection

$$h : RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \to INIT$$

$$\begin{tikzcd} RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[r, "\sim h"] & INIT \\ Composable_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[u, "g"] \arrow[r, "\sim f"'] & INS \end{tikzcd}$$

"$\Rightarrow$":

We can define a surjective function

$$i = f \circ g^{-1} \circ h^{-1} : INIT \to INS$$

By Theorem 5.5 there is K an AHL-occurrence nets. Furthermore

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

implies that g is a bijection and hence i is a bijection, i.e. KI is an AHL-occurrence net with instantiations.

"$\Leftarrow$":

By Theorem 5.5 there are (K_1, K_2) composable w.r.t. (I, i_1, i_2).
Furthermore there is a bijection

$$j : INIT \to INS$$

which we can use to define a bijection

$$k = h^{-1} \circ j \circ f$$

which implies that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

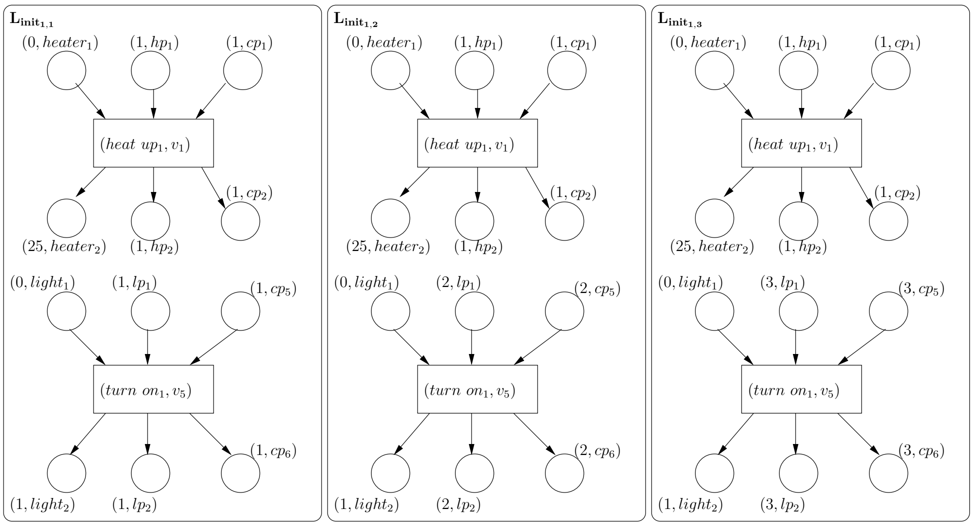


Figure 27: Instantiations of KI_1

For a detailed proof see Detailed Proof C.29 in the appendix. $\square$

Example 5.26 (Composition of AHL-Occurrence Nets with Instantiations)
Consider AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ where K_1 and K_2 are the AHL-occurrence nets in Fig. 25. Furthermore the set INS_1 contains the instantiations depicted in Fig. 27 and INS_2 contains the instantiations depicted in Fig. 28. The sets $INIT_1$ and $INIT_2$ contain the sets of input places of the instantiations in INS_1 and INS_2, respectively.
The composition $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ is an AHL-occurrence net $KI = (K, INIT, INS)$ which has one instantiation $L_{init} = L_{init_{1,3}} \circ_{(J,j_1,j_2)} L_{init_{2,1}}$ depicted in Fig. 26 and the corresponding initial marking $init = IN(L_{init})$.

$\triangle$

For AHL-occurrence nets with instantiations KI_1, K_2 which are composable with respect to a given span (I, i_1, i_2) in **AHLONet** the following definition provides a suitable object together with morphisms such that there is a corresponding span in the category **AHLONetI**.

Definition 5.27 (Interface of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_{1,N} : I \to K_1$ and $i_{2,N} : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations. Then the interface of (KI_1, KI_2) w.r.t. $(I, i_{1,N}, i_{2,N})$ is the AHL-occurrence net with instantiations $II = (I, INIT_I, INS_I)$ with

$$INIT_I = PreInit(i_{1,N})(INS_1) \cup PreInit(i_{2,N})(INS_2)$$

$$INS_I = PreIns(i_{1,N})(INS_1) \cup PreIns(i_{2,N})(INS_2)$$

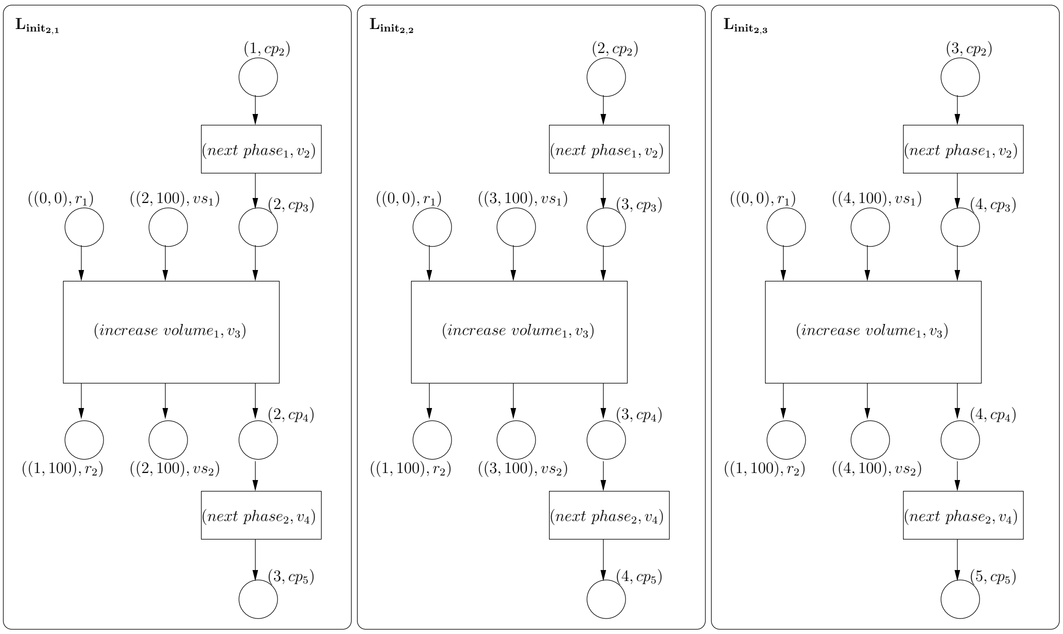


Figure 28: Instantiations of KI_2

together with **AHLNetI**-morphisms $i_1 = (i_{1,N}, i_{1,I})$, $i_2 = (i_{2,N}, i_{2,I})$ with

$$i_{1,I}(L) = PreIns(i_{1,N})(L)$$

and

$$i_{2,I}(L) = PreIns(i_{2,N})(L)$$

$\triangle$

Well-definedness. *II is AHL-occurrence net with instantiations*:

Due to Lemma 4.7 all instantiations $L \in INS_I$ are instantiations of I and due to Lemma 4.11 for every $init \in INIT_I$ there is $L \in INS_I$ with $init = IN(L)$. It remains to show that $IN : INS_I \to INIT_I$ is a bijection. Due to the fact that for every $init \in INIT_I$ there is $L \in INS_I$ with $init = IN(L)$ the function IN is surjective.
Let $L_1, L_2 \in INIT_I$ with $IN(L_1) = IN(L_2)$. Since $T_I = \emptyset$ there is $IN(L_1) = P_{L_1}$ and $IN(L_2) = P_{L_2}$ and hence $L_1 = L_2$, i.e. IN is injective. Thus IN is bijective.

i_1, i_2 *are* **AHLNetI**-*morphisms*:

$i_{1,N}$ and $i_{2,N}$ are AHL-morphisms by assumption. The well-definedness of the functions $i_{1,I}$ and $i_{2,I}$ follows from the definition of INS_I. Lemma 4.7 implies the existence of the required **INet**-morphisms.

$\square$

The partially set theoretical construction of the composition of AHL-occurrence nets with instantiations can also be performed as category theoretical pushout construction in the category **AHLNetI**.

Corollary 5.28 (Composition of AHL-Occurrence Nets with Instantiations is Pushout)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_{1,N} : I \to K_1$ and $i_{2,N} : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations.

1. Then for $KI = (K, INIT, INS)$ as defined in Theorem 5.25 with
   - $K = K_1 \circ_{(I,i_{1,N},i_{2,N})} K_2$,
   - $INS =$
     $\{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in \mathit{Composable}_{(I,i_{1,N},i_{2,N})}(INS_1, INS_2),$
     (J, j_1, j_2) is induced instantiation interface w.r.t. $(I, i_{1,N}, i_{2,N})\}$,
   - $INIT = \{IN(L) \mid L \in INS\}$,

   and for any set INS_I and any functions $i_{1,I}, i_{2,I}, i'_{1,I}$ and $i'_{2,I}$ such that (1) is a commuting diagram in **AHLNetI** the diagram (1) is a pushout in **AHLNetI**.
   Furthermore the required **AHLNetI**-object and **AHLNetI**-morphisms exist.

2. Given pushout diagram (1) in **AHLNetI** and **AHLONetI** then $KI = (K, INIT, INS)$ is the composition $KI = KI_1 \circ_{(I,i_{1,N},i_{2,N})} KI_2$ where $INIT = \{IN(L) \mid L \in INS\}$.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "{(i_{1,N},i_{1,I})}"] \arrow[d, "{(i_{2,N},i_{2,I})}"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "{(i'_{1,N},i'_{1,I})}"] \\
(K_2, INS_2) \arrow[r, "{(i'_{2,N},i'_{2,I})}"'] & (K, INS)
\end{tikzcd}$$

Proof sketch. Pushouts in the category **AHLNetI** can be constructed componentwise as pushout in **AHLNet** where the pushout object is exactly the composition $K = K_1 \circ_{(I,i_{1,N},i_{2,N})} K_2$ and as a pushout in $\mathbf{SET}^{op}$, i.e. as a pullback in **SET** where the pullback object is exactly the set INS as defined above.
The existence of a suitable interface follows from the well-definedness of Definition 5.27 and the pushout property of (1) can be shown explicitely.
For a detailed proof see Detailed Proof C.30 in the appendix. □

The composability of AHL-occurrence nets with instantiations KI_1 and KI_2 provides that the pushout in the category **AHLNetI** is also a pushout in the category **AHLONetI**.

Corollary 5.29 (Pushout of AHL-Occurrence Nets with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_{1,N} : I \to K_1$ and $i_{2,N} : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ for $x = 1, 2$ be two AHL-occurrence nets with instantiations and let (I, INS_I) be an AHL-net with instantiations together with **AHLNetI**-morphisms $i_1 = (i_{1,N}, i_{1,I})$ and $i_2 = (i_{2,N}, i_{2,I})$.
Then pushout (1) in **AHLNetI** is also pushout in **AHLONetI** iff (KI_1, KI_2) are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.

$$\begin{tikzcd}
(K_0, INS_0) \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "i_1'"] \\
(K_2, INS_2) \arrow[r, "i_2'"'] & (K_3, INS_3)
\end{tikzcd}$$

Proof. "$\Rightarrow$":

Given pushout (1) in **AHLNetI** and **AHLONetI**. Let $KI = KI_1 \circ_{(I, i_{1,N}, i_{2,N})} KI_2$. From Corollary 5.28 and the uniqueness of pushout objects follows that $KI = (K, INS)$. Due to the fact that (1) is pushout in **AHLONetI** KI is an AHL-occurrence net with instantiations and hence by Theorem 5.25 there are (KI_1, KI_2) composable w.r.t. $(I, i_{1,N}, i_{2,N})$.

"$\Leftarrow$":

Let (KI_1, KI_2) be composable w.r.t. $(I, i_{1,N}, i_{2,N})$. Then from Theorem 5.25 follows that $KI = KI_1 \circ_{(I, i_{1,N}, i_{2,N})} KI_2$ is an AHL-occurrence net. From Corollary 5.28 and the uniqueness of pushout objects follows that $KI = (K, INS)$, i.e. (K, INS) is an AHL-occurrence net with instantiations.

Since AHL-morphisms reflect AHL-occurrence nets due to $i_{1,N} : I \to K_1$ also I is an AHL-occurrence net. (I, INS_I) obviously is an AHL-occurrence net with instantiations because there is $T_I = \emptyset$ which means that there are only input places in I and hence there is a bijective correspondence between instantiations in INS_I and their input places.

So we have that (1) is a diagram in **AHLONetI**.

Let X an **AHLONetI**-object and $x_1 : (K_1, INS_1) \to X$, $x_2 : (K_2, INS_2) \to X$ **AHLONetI**-morpisms with

$$x_1 \circ i_1 = x_2 \circ i_2$$

Since **AHLONetI** is a subcategory of **AHLNetI** due to pushout (1) there is a unique morphism $x : (K, INS) \to X$ such that $x \circ i_1' = x_1$ and $x \circ i_2' = x_2$.

Since **AHLONetI** is a full subcategory of **AHLNetI** x is also an **AHLONetI**-morphism and hence (1) is pushout in **AHLONetI**.

□

In order to compose AHL-processes with instantiations we need the composability of the respective AHL-occurrence nets with instantiations as well as the composability of the AHL-processes. The combination of both requirements is the composability of AHL-processes with instantiations.

Definition 5.30 (Composability of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two instantiated AHL-processes of the AHL-net AN. Then (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2) if

1. (K_1, K_2) are composable w.r.t. (I, i_1, i_2),
2. $|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$ and

3. $mp_1 \circ i_1 = mp_2 \circ i_2$.

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(=)"] & K_1 \arrow[d, "mp_1"] \\
K_2 \arrow[r, "mp_2"'] & AN
\end{tikzcd}$$

△

Similar to the composability requirement for AHL-occurrence nets with instantiations the composability of AHL-processes with instantiations is a generalization of the sequential composability of AHL-processes. This means that if two AHL-processes are sequentially composable and the respective AHL-occurrence nets are AHL-occurrence nets with instantiations then these AHL-processes with instantiations are composable.

Fact 5.31 (Sequential Composability of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two instantiated AHL-processes of the AHL-net AN. If (mp_1, mp_2) are sequentially composable w.r.t. (I, i_1, i_2) then (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).

Proof. The sequential composability of (mp_1, mp_2) w.r.t. (I, i_1, i_2) implies that (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) which by Fact 5.24 implies that (KI_1, KI_2) are composable w.r.t. (I, i_1, i_2).
Furthermore there is $mp_1 \circ i_1 = mp_2 \circ i_2$ which means that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2). □

The composability of AHL-processes with instantiations provides that the composition leads to an AHL-process with instantiations.

Theorem 5.32 (Composition of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Let $KI_x = (K_x, INIT_x, INS_x)$ together with the AHL-net morphisms $mp_x : K_x \to AN$ for $x = 1, 2$ be two AHL-processes with instantions of the AHL-net AN such that (mp_1, mp_2) are composable w.r.t. (I, i_1, i_2).
Then the composition $KI = KI_1 \circ_{(I,i_1,i_2)} KI_2$ with $KI = (K, INIT, INS)$ together with the induced AHL-net morphism $mp : K \to AN$ is an AHL-process with instantiations of the AHL-net AN.

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i'_1"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "i'_2"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\
& & AN
\end{tikzcd}$$

Proof. Due to the composability of (mp_1, mp_2) w.r.t. (I, i_1, i_2) there is (K_1, K_2) are composable w.r.t. (I, i_1, i_2) and

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

which by Theorem 5.25 implies that the composition $(K, INIT, INS)$ of the AHL-occurrence nets with instantiations is an AHL-occurrence net with instantiations and hence $(K, INIT, INS)$ together with $mp : K \to N$ is an AHL-Process with instantiations, where mp is the unique morphism induced by Pushout (1) with

$$mp = (mp_P, mp_T)$$

$$mp_P(p) = \begin{cases} mp_{1P}(p_1) & p = i'_{1P}(p_1) \\ mp_{2P}(p_2) & p = i'_{2P}(p_2) \end{cases}$$

$$mp_T(t) = \begin{cases} mp_{1T}(t_1) & t = i'_{1T}(t_1) \\ mp_{2T}(t_2) & t = i'_{2T}(t_2) \end{cases}$$

□

Example 5.33 (Composition of AHL-Processes with Instantiations)
In Example 5.26 we show the composition of the AHL-occurrence nets with instantiations KI_1 and KI_2 leading to an AHL-occurrence net with instantiations KI. Moreover in Example 5.10 we show the composition of two AHL-processes $mp_1 : K_1 \to Alarm$ and $mp_2 : K_2 \to Alarm$ leading to an AHL-process $mp : K \to Alarm$. In the context of the AHL-occurrence nets with instantiations KI_1 and KI_2 the processes mp_1 and mp_2 are AHL-processes with instantiations.
So the morphism mp together with the AHL-occurrence net with instantiations KI form the AHL-process with instantiations which is the composition of the AHL-processes with instantiations mp_1 and mp_2. △

The composition of AHL-processes with instantiations of an AHL-net AN can also be performed as a pushout in the category **AHLProcI(AN)** of AHL-processes with instantiations of AN.

Corollary 5.34 (Pushout of AHL-Processes with Instantiations)
Given the AHL-occurrence nets K_1, K_2 and I as above and two injective AHL-net morphism $i_{1,N} : I \to K_1$ and $i_{2,N} : I \to K_2$.
Let $KI_x = (K_x, INIT_x, INS_x)$ together with AHL-morphisms $mp_{x,N} : K_x \to AN$ for $x = 1, 2$ be two AHL-processes with instantiations and let (I, INS_I) an **AHLNetI**-object together with **AHLNetI**-morphisms $i_1 = (i_{1,N}, i_{1,I})$ and $i_2 = (i_{2,N}, i_{2,I})$.
Then there exist **AHLNetI**-morphisms $mp_0 : (I, INS_I) \to (AN, \emptyset), mp_1 = (mp_{1,N}, mp_{1,I})$ and $mp_2 = (mp_{2,N}, mp_{2,I})$ such that pushout (1) in **AHLNetI** is also pushout in **AHLProcI(AN)** iff $(mp_{1,N}, mp_{2,N})$ are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(\mathbf{1})"] \arrow[ddrr, bend right=60, "mp_0"'] & (K_1, INS_1) \arrow[d, "i'_1"] \arrow[ddr, bend left, "mp_1"] & \\
(K_2, INS_2) \arrow[r, "i'_2"'] \arrow[drr, bend right, "mp_2"] & (K, INS) \arrow[dr, "mp"] & \\
& & (AN, \emptyset)
\end{tikzcd}$$

Proof sketch.

”⇒”:
The fact that the pushout (1) is a pushout in **AHLProcI**(**AN**) implies that (K, INS) is an AHL-occurrence net leading to the fact that (1) is also a pushout in **AHLONetI** which by Corollary 5.29 implies that (KI_1, KI_2) are composable w.r.t. $(I, i_{1,N}, i_{2,N})$. Together with the required commutativities of morphisms in the category **AHLProcI**(**AN**) this implies that $(mp_{1,N}, mp_{2,N})$ are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.

”⇐”:
Given pushout (1) in **AHLNetI** there is also a corresponding pushout in **AHLNet** which due to the composability of $(mp_{1,N}, mp_{2,N})$ w.r.t. $(I, i_{1,N}, i_{2,N})$ is also a pushout in **AHLProc**(**AN**). Using the cofree construction $Inst$ the pushout property in **AHLProc**(**AN**) implies that (1) is also a pushout in **AHLProcI**(**AN**).

For a detailed proof see Detailed Proof C.31 in the appendix. □

6 Decomposition of Algebraic High-Level Processes

For a rule-based transformation of processes it is necessary to have a decomposition of processes as inverse operation to the composition. While composing AHL-processes with the pushout construction we use the pushout complement as the inverse operation to the pushout for decomposition.

6.1 Pushout Complements

Considering the construction of a pushout with respect to an interface as a commutative operation with two arguments the pushout complement can be considered as the inverse operation to the pushout construction.

Definition 6.1 (Pushout Complement)
Given objects A, B, D and morphisms $f_1 : A \to B$, $g_1 : B \to D$ in a category $\mathbf{C}$. Then an object C together with morphisms $f_2 : A \to C$, $g_2 : C \to D$ is called pushout complement of $A \xrightarrow{f_1} B \xrightarrow{g_1} D$ in $\mathbf{C}$ iff (1) is a pushout in $\mathbf{C}$.

$$\begin{tikzcd} A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & B \arrow[d, "g_1"] \\ C \arrow[r, "g_2"'] & D \end{tikzcd}$$

(C, f_2, g_2) is called maximal pushout complement if for every pushout complement (C', f_2', g_2') of $A \xrightarrow{f_1} B \xrightarrow{g_1} D$ in $\mathbf{C}$ there is a unique morphism $c : C' \to C$ such that

$$c \circ f_2' = f_2 \text{ and } g_2 \circ c = g_2'$$

△

Example 6.2 (Pushout Complement)
Figure 26 on page 72 shows the pushout complement L_{init_2} of $J \to L_{init_1} \to L_{init}$ in **PTNet** where the given morphisms are inclusions. △

The categories **SET**, **PTNet** and **AHLNet** have unique pushout complements if the first of the given morphisms is injective. This follows from the fact that they are weak adhesive HLR categories (see [EEPT06]). It is an important property in order to obtain unambiguous results using the pushout complement but unfortunately this is not always the case. A solution is the definition of maximal pushout complements (see [KMO+05]).
While in a given category there may exist several different pushout complements there are unique maximal pushout complements in every category. We need the concept of maximal pushout complements for a unique decomposition of AHL-processes with instantiations.

Fact 6.3 (Uniqueness of Maximal Pushout Complements)
Given objects A, B, D and morphisms $f_1 : A \to B$, $g_1 : B \to D$ in a category $\mathbf{C}$. Let (C, f_2, g_2) be a maximal pushout complement of $A \xrightarrow{f_1} B \xrightarrow{g_1} D$ in $\mathbf{C}$.

1. For every maximal pushout complement (C', f_2', g_2') of $A \xrightarrow{f_1} B \xrightarrow{g_1} D$ in $\mathbf{C}$ there is an isomorphism $x : C \xrightarrow{\sim} C'$ such that

$$g_2' \circ x = g_2 \text{ and } x \circ f_2 = f_2'$$

2. For (C', f_2', g_2') with an isomorphism $x : C \xrightarrow{\sim} C'$ such that

$$g_2' \circ x = g_2 \text{ and } x \circ f_2 = f_2'$$

and g_2' is a monomorphism there is (C', f_2', g_2') a maximal pushout complement of $A \xrightarrow{f_1} B \xrightarrow{g_1} D$ in $\mathbf{C}$.

$$\begin{tikzcd}
 & A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dl, bend right, "f_2'"'] & B \arrow[d, "g_1"] \\
 & C \arrow[r, "g_2"'] & D \\
C' \arrow[ur, "x"] \arrow[urr, bend right, "g_2'"'] & &
\end{tikzcd}$$

Proof sketch.

Part 1:
Two maximal pushout complements have unique morphisms to the respective other maximal pushout complement and also unique morphisms to themselves implying that the composition of the unique morphisms are the identities and therefore they are isomorphic.

Part 2:
The isomorphism x is the unique required morphism which can be obtained from the monomorphism property of g_2'.

For a detailed proof see Detailed Proof C.32 in the appendix. □

A necessary and sufficient condition for the existence of a pushout complement usually is called gluing condition (see [EEPT06]) and consists of two parts: the identification condition and the dangling condition.
The identification condition requires the preservation of non-injectively matched elements. Since the morphisms in a composition diagram are all injective, the identification condition is always fulfilled.
But we have to check the dangling condition because the decomposition may delete places whereas transitions are not deleted.

Definition 6.4 (Dangling Condition)
Given AHL-occurrence nets I, K_1, K with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.

$$\begin{tikzcd} I \arrow[r, "f_1"] & K_1 \arrow[d, "g_1"] \\ & K \end{tikzcd}$$

The gluing points GP of K_1 are the parts of K_1 which are preserved by f_1:

$$GP = f_{1,P}(P_I)$$

The dangling points DP of K_1 are the places of K_1 which have an image in the pre or post domain of a transition in K that is not in $g_1(K_1)$:

$$\begin{aligned} DP \quad = \quad & \{p \in P_{K_1} \mid \exists t \in T_K \setminus g_{1,T}(T_{K_1}), term \in T_{OP}(X)_{type(p)} : \\ & (term, g_{1,P}(p)) \leq pre_K(t) \oplus post_K(t)\} \end{aligned}$$

The dangling condition requires that all dangling points are gluing points, i.e.

$$DP \subseteq GP$$

△

Example 6.5 (Dangling Condition)
Consider the span $I \to K_1 \to K$ in Figure 25 on page 65 where the given AHL-morphisms are inclusions. The places cp_2 and cp_5 in the net K_1 are dangling points because (old, cp_2) is a pre condition of the transition *next phase*$_1$ and (new, cp_5) is a post condition of the transition *next phase*$_2$ and both of the transitions are not in the image of g_1.
The dangling condition is satisfied because

$$DP = \{cp_2, cp_5\} \subseteq \{cp_2, cp_5\} = f_{1,P}(\{cp_2, cp_5\}) = f_{1,P}(P_I) = GP$$

Analogously for the span $I \to K_2 \to K$ in Figure 25 there are exactly the same dangling and gluing points and hence also for this span the dangling condition is satisfied. △

If the dangling condition is satisfied we can decompose the net K with a unique result by pushout complement. The other way round the existence of a pushout complement implies that the dangling condition is satisfied.

Fact 6.6 (Pushout Complement of AHL-Nets)
Given AHL-nets $I = (SP, P_I, T_I, pre_I, post_I, cond_I, type_I, A)$,
$K_1 = (SP, P_{K_1}, T_{K_1}, pre_{K_1}, post_{K_1}, cond_{K_1}, type_{K_1}, A)$ and
$K = (SP, P_K, T_K, pre_K, post_K, cond_K, type_K, A)$ with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.

$$\begin{tikzcd} I \arrow[r, "f_1"] & K_1 \arrow[d, "g_1"] \\ & K \end{tikzcd}$$

Then the pushout complement K_2 in **AHLNet** is defined as

$$K_2 = (SP, P_{K_2}, T_{K_2}, pre_{K_2}, post_{K_2}, cond_{K_2}, type_{K_2}, A)$$

with

- $P_{K_2} = (P_K \setminus g_{1,P}(P_{K_1})) \cup g_{1,P}(f_{1,P}(P_I))$
- $T_{K_2} = T_K \setminus g_{1,T}(T_{K_1})$
- $pre_{K_2} = pre_K|_{T_{K_2}}$
- $post_{K_2} = post_K|_{T_{K_2}}$
- $cond_{K_2} = cond_K|_{T_{K_2}}$
- $type_{K_2} = type_K|_{P_{K_2}}$

Furthermore we define AHL-morphisms $f_2 : I \to K_2$ as

$$f_2 = (f_{2,P}, f_{2,T})$$

with

- $f_{2,P}(p) = (g_1 \circ f_1)_P(p)$
- $f_{2,T} = \emptyset$ (the empty function)

and $g_2 : K_2 \to K$ as an inclusion.
The AHL-net K_2 and the morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ exists such that (1) is pushout in **AHLNet** iff the dangling condition is satisfied.
If K_2 exists it is unique up to isomorphism.

$$\begin{tikzcd}
I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\
K_2 \arrow[r, "g_2"'] & K
\end{tikzcd}$$

Proof sketch.

"⇒":
A violation of the dangling condition results in a "dangling arc" in the net K_2, i.e. there is a transition with a non-existing place in its pre or post condition. Hence the net K_2 is not a well-defined AHL-net.

"⇐":
The satisfaction of the dangling condition ensures that there are no dangling arcs and since all morphisms are injective the net K_2 and the morphisms f_2 and g_2 are well-defined. The pushout property of (1) can be shown explicitely.

For a detailed proof see Detailed Proof C.33 in the appendix. □

Example 6.7 (Pushout Complement of AHL-Nets)
Consider the span $I \to K_1 \to K$ in Figure 25 on page 65 where the given AHL-morphisms are inclusions. As mentioned in Example 6.5 the dangling condition is satisfied. Therefore there exists the unique pushout complement K_2 of $I \to K_1 \to K$ in **AHLNet** which is also depicted in Figure 25.
Analogously K_1 is the unique pushout complement of $I \to K_2 \to K$ in **AHLNet**. △

6.2 Decomposition of AHL-Processes

The pushout complement of AHL-occurrence nets does not only lead to a to pushout diagram of AHL-nets but also to a composition diagram. This means that the dangling condition is a necessary and sufficient condition for the decomposition of AHL-occurrence nets.

Theorem 6.8 (Decomposition of AHL-Occurrence Nets)
Given AHL-occurrence nets I, K_1, K with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$. Then there exists an AHL-occurrence net K_2 and morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that (K_1, K_2) are composable w.r.t. (I, f_1, f_2) and $K = K_1 \circ_{(I,f_1,f_2)} K_2$ and K_2 is unique up to isomorphism iff the dangling condition is satisfied.

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

K_2 together with morphisms f_2 and g_2 is called decomposition of K w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$.

Proof sketch. ”⇒”:

The composition of AHL-occurrence nets is a pushout in the category **AHLNet** implying that K_2 is a pushout complement and hence by Fact 6.6 the dangling condition is satisfied.

”⇐”:

The satisfaction of the dangling condition by Fact 6.6 ensures the existence of a pushout complement K_2 together with required morphisms f_2 and g_2. K_2 is an AHL-occurrence nets because of the AHL-morphism g_2 and Fact 2.14.
Then the injectivity of $f_{1,P}$ and $f_{1,T}$ by construction of f_2 implies that also f_2 is injective.
Furthermore the fact that all parts of the net K_1 are deleted in the net K_2 ensures that places which are not input places in K_1 are input places in K_2 and the same holds for output places.
The fact that the induced causal relation $<_{(f_1,f_2)}$ is a finitary strict partial order can be derived from the fact that $<_K$ is a finitary strict partial order.
Hence K_1 and K_2 are composable w.r.t. (I, f_1, f_2).

For a detailed proof see Detailed Proof C.34 in the appendix. □

Example 6.9 (Decomposition of AHL-Occurrence Nets)
Consider the span $I \to K_1 \to K$ in Figure 25 on page 65 where the given AHL-morphisms are inclusions. The unique pushout complement K_2 of $I \to K_1 \to K$ in **AHLNet** which is

also depicted in Figure 25 is also a decomposition of the AHL-occurrence net K with respect to $I \to K_1 \to K$.
Analogously the unique pushout complement K_1 of $I \to K_2 \to K$ in **AHLNet** is a decomposition of K with respect to $I \to K_2 \to K$. △

The decomposition of AHL-occurrence nets which is defined as pushout complement in **AHLNet** is also a pushout complement in **AHLONet**.

Corollary 6.10 (Pushout Complement of AHL-Occurrence Nets)
Given AHL-nets I, K_1, K with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$ such that the dangling condition is satisfied. Then the pushout complement K_2 together with $f_2 : I \to K_2$, $g_2 : K_2 \to K$ in **AHLNet** is also pushout complement in **AHLONet** iff K is an AHL-occurrence net.

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

Proof. ”⇒”:
Given K_2 together with $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that K_2 is pushout complement in **AHLNet** and **AHLONet**. This means that (1) is pushout in **AHLONet** and hence K is an AHL-occurrence net.

”⇐”:
Given K_2 together with $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that K_2 is pushout complement in **AHLNet** and K is an AHL-occurrence net. Since AHL-morphisms reflect AHL-occurrence nets the AHL-morphisms g_1 and $g_1 \circ f_1$ imply that I and K_1 are AHL-occurrence nets.
Then from Theorem 6.8 follows the existence of a unique AHL-occurrence net K_2' such that (K_1, K_2') are composable w.r.t. (I, f_1, f_2') and $K = K_1 \circ_{(I,f_1,f_2)} K_2'$. Due to Theorem 5.5 diagram (2) is a pushout in **AHLNet**.

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2'"'] \arrow[dr, phantom, "(\mathbf{2})"] & K_1 \arrow[d, "g_1"] \\ K_2' \arrow[r, "g_2'"'] & K \end{tikzcd}$$

Due to Theorem 6.6 pushout complement in **AHLNet** is unique up to isomorphism which means that also (K_1, K_2) are composable w.r.t. (I, f_1, f_2). From Corollary 5.7 follows that pushout (1) in **AHLNet** is also pushout in **AHLONet** and hence K_2 is pushout complement in **AHLONet**.

□

The decomposition of AHL-processes can be computed by decomposition of the respective AHL-occurrence nets of the processes.

Theorem 6.11 (Decomposition of AHL-Processes)
Given AHL-occurrence nets I, K_1 and K, injective AHL-morphisms $f_1 : K_0 \to K_1$, $g_1 : K_1 \to K_3$ and a morphism $mp : K \to AN$.
If K_2 exists s.t. (1) is a composition diagram then there are morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ with $mp_1 = mp \circ g_1$ and $mp_2 = mp \circ g_2$.
mp_1 and mp_2 are called decompositions of mp.

$$\begin{tikzcd}
K_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "g_2"'] \arrow[drr, bend right, "mp_2"'] & K_3 \arrow[dr, "mp_3"] & \\
 & & AN
\end{tikzcd}$$

Proof. The existence of the morphisms follows from the well-defined composition of morphisms in the category **AHLNet**. □

Example 6.12 (Decomposition of AHL-Processes)
Consider the span $I \to K_1 \to K$ in Figure 25 on page 65 where the given AHL-morphisms are inclusions. We can define process morphisms $mp_I : I \to Alarm$, $mp_1 : K_1 \to Alarm$ and $mp : K \to Alarm$ where $Alarm$ is the AHL-net shown in Figure 1. The morphisms mp_I, mp_1 and mp map places and transitions to places and transitions with the same name but without index.
The decomposition K_2 of K with respect to $I \to K_1 \to K$ which is also shown in Figure 25 is an AHL-occurrence net and there is a process morphism $mp_2 : K_2 \to Alarm$ which maps the places and transitions in K_2 exactly in the same way as mp maps these places and transitions in K. △

The decomposition of AHL-processes of an AHL-net AN as pushout complement of the respective AHL-occurrence nets in the category **AHLNet** leads to a pushout complement in the category **AHLProc**(**AN**) of AHL-processes of AN.

Corollary 6.13 (Pushout Complement of AHL-Processes)
Given AHL-occurrence nets I, K_1 and K, injective AHL-morphisms $f_1 : K_0 \to K_1$, $g_1 : K_1 \to K_3$ and an AHL-morphism $mp : K \to AN$.
Let the the dangling condition be satisfied.
Then for the pushout complement K_2 together with $f_2 : I \to K_2$, $g_2 : K_2 \to K$ in **AHLNet** there exist AHL-morphisms $mp_0 : I \to AN, mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ such that mp_2 is the pushout complement of $mp_0 \xrightarrow{f_1} mp_1 \xrightarrow{g_1} mp$ in **AHLProc**(**AN**).

$$
\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(\mathbf{1})"] \arrow[ddrr, bend right=60, "mp_0"'] & K_1 \arrow[d, "i_1'"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "i_2'"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\
 & & AN
\end{tikzcd}
$$

Proof. Given pushout complement K_2 in **AHLNet**. From Corollary 6.10 follows that K_2 is also a pushout complement in **AHLONet** and hence (K_1, K_2) are composable w.r.t. (I, f_1, f_2) and (1) is a composition diagram. Then Theorem 6.11 implies AHL-morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ such that $mp_1 = mp \circ g_1$ and $mp_2 = mp \circ g_2$.
Then we have

$$
\begin{aligned}
mp_2 \circ f_2 &= mp \circ g_2 \circ f_2 \\
&= mp \circ g_1 \circ f_1 \\
&= mp_1 \circ f_1
\end{aligned}
$$

and hence (mp_1, mp_2) are composable w.r.t. (I, f_1, f_2) which by Corollary 5.11 implies that there exists $mp_0 : I \to AN$ such that pushout (1) in **AHLNet** is also pushout in **AHLProc(AN)**. □

6.3 Decomposition of AHL-Processes with Instantiations

In order to obtain composition and decomposition constructions which are inverse to each other, the decomposition of AHL-occurrence nets with instantiations does not lead to a unique result. One AHL-occurrence net with instantiations may be the composition of different AHL-occurrence nets with instantiations because there can be instantiations which are not composable and therefore these instantiations have no influence on the composition.
For this reason we introduce the notion of a cominimal decomposition. We call a decomposition cominimal if there are no instantiations which are not composable with any other instantiations.
Cominimality is the dual notion to minimality, i.e. cominimal means maximal. We do not say maximal but cominimal because it is more intuitive in the sense that cominimality of the decomposition means that the set of instantiations is minimal in the way that it is the smallest set of instantiations such that the AHL-occurrence net with these instantiations is a decomposition. As we show further below this concept of cominimality is consistent with the concept of maximal pushout complements.
For every AHL-occurrence net with instantiations which can be decomposed there is a decomposition which is cominimal. For the cominimal decomposition of an AHL-occurrence net with instantiations we can use the preimage constructions for instantiations and initializations in section 4.

Definition 6.14 ((Cominimal) Decomposition of AHL-Occurrence Nets with Instantiations)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and
$KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms

$f_1 : I \to K_1$ and $g_1 : K_1 \to K$.
An AHL-occurrence net with instantiations $KI_2 = (K_2, INIT_2, INS_2)$ together with morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ is called decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$.
The decomposition is called cominimal iff for every $L_{init_2} \in INS_2$ there is $L_{init_1} \in INS_1$ s.t. (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2). △

To decompose an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ it is necessary that the set INS of instantiations satisfies a specific condition. Since the composition $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ of AHL-occurrence nets contains every composition of composable instantiations of KI_1 and KI_2, the set INS must be complete in the way that it has to contain also the composition of all decomposed parts of its instantiations which are composable.

Definition 6.15 (Decomposability of AHL-Occurrence Net with Instantiations)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_{1,N} : K_1 \to K$. Furthermore let $g_{1,I} : INS \to INS_1$ be a function such that $g_1 = (g_{1,N}, g_{1,I})$ is an **AHLNetI**-morphism.
KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_{1,N}}{\to} K$ and $g_{1,I}$ iff

- the decomposition K_2 of K w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ exists such that (1) is a composition diagram,

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

- the function
$$IN : PreIns(g_2)(INS) \to PreInit(g_2)(INS)$$
with
$$IN(PreIns(g_2)(L_{init})) = PreInit(g_2)(L_{init})$$
is injective

- and for all $L_{init_1} \in INS_1, L_{init} \in INS$:
If there is
$$PreIns(f_1)(L_{init_1}) = PreIns(f_1)(g_{1,I}(L_{init}))$$
then
$$(L_{init_1}, PreIns(g_2)(L_{init})) \in Composable_{(I,f_1,f_2)}$$
and
$$L_{init_1} \circ_{(J,j_1,j_2)} PreIns(g_2)(L_{init}) \in INS$$
where (J, j_1, j_2) is the instantiation interface induced by (I, f_1, f_2).

△

Example 6.16 (Decomposability of AHL-Occurrence Nets with Instantiations)
Consider the span $I \to K_1 \to K$ of AHL-occurrence nets in Figure 25 on page 65. Let $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$ be AHL-occurrence nets with instantiations where INS_1 contains the instantiations depicted in Figure 27 on page 78 and INS contains the instantiation L_{init} depicted in Figure 26 on page 72.
The instantiation preimage of L_{init} with respect to g_1 is the instantiation $L_{init_{1,3}}$ in Figure 27. This allows us to define a function $g_{1,I} : INS \to INS_1$ with $g_{1,I}(L_{init}) = L_{init_{1,3}}$ and $(g_1, g_{1,I})$ is an **AHLNetI**-morphism.
Furthermore the instantiation preimage of L_{init} with respect to g_2 is the instantiation $L_{init_{2,1}}$ in Figure 28 on page 79.
For the instantiation J in Figure 26 there is

$$PreIns(f_1)(L_{init_{1,3}}) = J = PreIns(f_1)(g_{1,I}(L_{init}))$$

and there is

$$(L_{init_{1,3}}, L_{init_{2,1}}) \in Composable_{(I,f_1,f_2)}$$

and

$$L_{init_{1,3}} \circ_{(J,j_1,j_2)} L_{init_{2,1}} = L_{init} \in INS$$

Hence KI is decomposable with respect to $I \xrightarrow{f_1} K_1 \xrightarrow{g_{1,N}} K$ and $g_{1,I}$. △

Theorem 6.17 (Cominimal Decomposition of AHL-Occurrence Nets with Instantiations)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$, injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$ and a function $g_{1,I} : INS \to INS_1$ such that $(g_1, g_{1,I})$ is an **AHLNetI**-morphism.
If and only if KI is decomposable w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$ leading to composition diagram (1)

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

then the preimage

$$KI_2 = (K_2, INIT_2, INS_2)$$

induced by KI and g_2 (Theorem 4.13) with

$$INIT_2 = PreIns(g_2)(INS)$$

and

$$INS_2 = PreIns(g_2)(INIT)$$

is a cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.

Proof sketch.

"$\Rightarrow$":

From the decomposability follows the existence of a decomposition K_2 of K w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$. Then due to the injectivity of the function IN by Theorem 4.13 KI_2 as defined above is an AHL-occurrence net with instantiations.
The last property of the decomposability and the construction of KI_2 ensure that KI_2 is a cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.

"$\Leftarrow$":

Since for the decomposition $KI_2 = (K_2, INIT_2, INS_2)$ of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ the net K_2 is a decomposition of K w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ the first condition of the decomposability is satisfied.
The due to the fact that KI_2 is an AHL-occurrence net from Fact 4.13 follows that the second condition is satisfied. The last condition is ensured by the fact that KI is the composition of KI_1 and KI_2 w.r.t. (I, f_1, f_2).

For a detailed proof see Detailed Proof C.35 in the appendix. □

Example 6.18 (Cominimal Decomposition of AHL-Occurrence Nets with Instantiations)
Consider the again the span $I \to K_1 \to K$ of AHL-occurrence nets in Figure 25 on page 65 and AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$ as in Example 6.16.
In Example 5.26 we show that KI is the composition of KI_1 and $KI_2 = (K_2, INIT_2, INS_2)$ where K_2 is depicted in Figure 25 and INS_2 contains the instantiations shown in Figure 28. So KI_2 is a decomposition of KI w.r.t. $I \to K_1 \to K$ but the decomposition is not cominimal because there is no $init_1 \in INS_1$ such that $(L_{init_1}, L_{init_{2,2}})$ are composable w.r.t. (I, i_1, i_2).
In fact for an AHL-occurrence net with instantiations $KI'_2 = (K_2, INIT'_2, INS'_2)$ with $INS'_2 = INS''_2 \cup X$ where $INS''_2 = \{L_{init_{2,1}}\}$ and X is an arbitrary set of instantiations which are not composable with any instantiation in INS_1 the net KI'_2 is a decomposition of KI w.r.t. $I \to K_1 \to K$. Choosing the set $X = \emptyset$ we obtain a cominimal decomposition. △

A decomposition of AHL-occurrence nets with instantiations is a pushout complement in **AHLNetI** and **AHLONetI**.

Corollary 6.19 (Decomposition of AHL-Occurrence Nets with Instantiations is Pushout Complement)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$, injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$ and a function $g_{1,I} : INS \to INS_1$ such that $(g_1, g_{1,I})$ is an **AHLNetI**-morphism and KI is decomposable w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$.

1. Let $KI_2 = (K_2, INIT_2, INS_2)$ be a decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$. and INS_I an arbitrary set and $i_{1,I}, i_{2,I}, i'_{1,I}$ and $i'_{2,I}$ arbitrary functions such that (1) is a commuting diagram in **AHLNetI**.
   Then (K_2, INS_2) is a pushout complement of $(I, INS_I) \xrightarrow{(f_1, f_{1,I})} (K_1, INS_1) \xrightarrow{(g_1, g_{1,I})} (K, INS)$ in **AHLNetI** and in **AHLONetI**.

2. Given pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and in **AHLONetI**. Then $KI_2 = (K_2, INIT_2, INS_2)$ is a decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$ where

$$INIT_2 = \{IN(L) \mid L \in INS_2\}$$

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "{(f_1,f_{1,I})}"] \arrow[d, "{(f_2,f_{2,I})}"'] \arrow[dr, phantom, "\mathbf{(1)}"] & (K_1, INS_1) \arrow[d, "{(g_1,g_{1,I})}"] \\
(K_2, INS_2) \arrow[r, "{(g_2,g_{2,I})}"'] & (K, INS)
\end{tikzcd}$$

Proof. Part 1:

Since KI_2 is the decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$ there is $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$.
Then from Corollary 5.28 follows that (1) is pushout in **AHLNetI**.
The fact that $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ implies that (KI_1, KI_2) are composable w.r.t. (I, f_1, f_2) by Theorem 5.25 and hence there is (1) also a pushout in **AHLONetI** by Corollary 5.29. This means that (K_2, INS_2) is a pushout complement of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and **AHLONetI**.

Part 2:

Given pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and in **AHLONetI**, i.e. (1) is pushout in **AHLNetI** and in **AHLONetI**.
This means that there is a set $INIT_2 = \{IN(L) \mid L \in INS_2\}$ such that $(K_2, INIT_2, INS_2)$ is an AHL-occurrence nets with instantiations.
By Corollary 5.28 there is $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ which means that KI_2 is a decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$.

□

The decomposability of AHL-occurrence nets with instantiations is a necessary condition for the fact that the pushout complement in **AHLNetI** is also a pushout complement in **AHLONetI**.

Corollary 6.20 (Necessary Condition for Pushout Complements of AHL-Occurrence Nets with Instantiations)
Given AHL-nets with instantiations $KI_1 = (K_1, INS_1)$ and $KI = (K, INS)$, (I, INS_I) with $T_I = \emptyset$, injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$ and functions $f_{1,I} : INS_1 \to INS_I$, $g_{1,I} : INS \to INS_1$ such that $(f_1, f_{1,I})$ and $(g_1, g_{1,I})$ are **AHLNetI**-morphisms. If the pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** is also pushout complement in **AHLONetI** then KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

Proof. Given the pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and also in **AHLONetI**.

Then Corollary 6.19 implies that $(K_2, INIT_2, INS_2)$ is a decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$ where

$$INIT_2 = \{IN(L) \mid L \in INS_2\}$$

Due to Theorem 6.17 there is KI decomposable w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$. □

For a categorical characterization of cominimal decompositions we need the following lemma which is a characterization of induced preimages of AHL-occurrence nets with instantiations (see Theorem 4.13) by **AHLNetI**-morphisms.

Lemma 6.21 (Preimage-Surjection-Lemma)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$,
$KI_2 = (K_2, INIT_2, INS_2)$ and an AHL-morphism $f : K_1 \to K_2$.
If and only if KI_1 is the induced preimage of KI_2 and f, i.e.

$$INIT_1 = PreInit(f)(INS_2)$$

and

$$INS_1 = PreIns(f)(INS_2)$$

then there is a surjection $s : INS_2 \to INS_1$ such (f, s) is an **AHLNetI**-morphism.

$$K_1 \xrightarrow{f} K_2 \qquad\qquad (K_1, INS_1) \xrightarrow{(f,s)} (K_2, INS_2)$$

Proof sketch. KI_1 is the induced preimage of KI_2 and f if and only if for every instantiation $L_2 \in INS_2$ there is an instantiation $L_1 \in INS_1$ such that $L_1 = PreIns(f)(L_2)$ leading to an instantiation morphism $f_L : L_1 \to L_2$ and therefore to a surjection $s : INS_2 \to INS_1$ such (f, s) is an **AHLNetI**-morphism.
For a detailed proof see Detailed Proof C.36 in the appendix. □

Since the construction of cominimal decompositions in Theorem 6.17 is defined via an induced instantiation preimage (see Theorem 4.13) it does not surprise that we can use the characterization of instantiation preimages to give a characterization of cominimal decompositions.

Theorem 6.22 (Characterization of Cominimal Decompositions (I))
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and
$KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.
Let $KI_2 = (K_2, INIT_2, INS_2)$ together with morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that KI_2 is a decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.
Then the decomposition is cominimal iff there exists a surjection $s : INS \to INS_2$ such that (g_2, s) is an **AHLNetI**-morphism.

Proof sketch. The definition of the composition of AHL-occurrence nets with instantiations allows the definition of a well-defined function $s : INS \to INS_2$ with the required property. The fact that the composition is cominimal implies that s is a surjection.
On the other hand Lemma 6.21 implies that KI_2 is the induced preimage of KI and g_2 which by Theorem 6.17 implies that KI_2 is a cominimal decomposition.
For a detailed proof see Detailed Proof C.37 in the appendix. □

Due to Fact 4.27 the pushout in **AHLNetI** can be constructed componentwise as pushout in **AHLNet** and as pullback in **SET**. So in order to decompose **AHLNetI** objects we do not only need the pushout complement construction but also the dual construction which are pullback complements.

Definition 6.23 ((Minimal) Pullback Complement)
Given objects A, B, D and morphisms $f_1 : A \to B$, $g_1 : B \to D$ in a category $\mathbf{C}$. Then an object C together with morphisms $f_2 : A \to C$, $g_2 : C \to D$ is called pullback complement of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$ iff (1) is a pullback in $\mathbf{C}$.

$$\begin{tikzcd} A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & B \arrow[d, "g_1"] \\ C \arrow[r, "g_2"'] & D \end{tikzcd}$$

(C, f_2, g_2) is called minimal pullback complement if for every pullback complement (C', f_2', g_2') of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$ there is a unique morphism $c : C \to C'$ such that

$$c \circ f_2 = f_2' \text{ and } g_2' \circ c = g_2$$

△

Similar to maximal pushout complements also minimal pullback complements are unique up to isomorphism in every category.

Fact 6.24 (Uniqueness of Minimal Pullback Complements)
Given objects A, B, D and morphisms $f_1 : A \to B$, $g_1 : B \to D$ in a category $\mathbf{C}$. Let (C, f_2, g_2) be a minimal pullback complement of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$.

1. For every minimal pullback complement (C', f_2', g_2') of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$ there is an isomorphism $x : C \overset{\sim}{\to} C'$ such that

$$g_2' \circ x = g_2 \text{ and } x \circ f_2 = f_2'$$

2. If g_2 is a monomorphism then for (C', f_2', g_2') with an isomorphism $x : C \overset{\sim}{\to} C'$ such that

$$g_2' \circ x = g_2 \text{ and } x \circ f_2 = f_2'$$

there is (C', f_2', g_2') a minimal pullback complement of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$.

$$\begin{tikzcd} & A \arrow[r, "f_1"] \arrow[d, "f_2"] \arrow[ddl, bend right, "f_2'"'] & B \arrow[d, "g_1"] \\ & C \arrow[r, "g_2"] \arrow[dl, "x"'] & D \\ C' \arrow[urr, bend right, "g_2'"'] & & \end{tikzcd}$$

Proof sketch. Part 1 follows from dualization of part 1 of Fact 6.3. The proof for part 2 can be done completely analogously to the proof of part 2 of Fact 6.3.
For a detailed proof see Detailed Proof C.38 in the appendix. □

The definition of minimal pullbacks allows to give another characterization of cominimal decompositions. The reason for the ambiguity of pushout complements in the category **AHLNetI** is the ambiguity of pullback complements in **SET**. Intuively cominimality of a decomposition means that we take the smallest set of instantiations such that the result is a decomposition. In other words this means that the instantiation part of the result is a minimal pullback complement in **SET**.

Theorem 6.25 (Characterization of Cominimal Decompositions (II))
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.
Let $KI_2 = (K_2, INIT_2, INS_2)$ together with morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that KI_2 is a decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$.
By Corollary 6.19 for a set INS_I and functions $f_{1,I} : INS_1 \to INS_I$, $g_{1,I} : INS \to INS_1$ such that diagram (1) is in **AHLNetI** there is (K_2, INS_2) together with $(f_2, f_{2,I}), (g_2, g_{2,I})$ a pushout complement of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and in **AHLONetI**.
Then the decomposition is cominimal iff INS_2 together with $f_{2,I}, g_{2,I}$ is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "(f_1,f_{1,I})"] \arrow[d, "(f_2,f_{2,I})"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "(g_1,g_{1,I})"] \\
(K_2, INS_2) \arrow[r, "(g_2,g_{2,I})"'] & (K, INS)
\end{tikzcd}
\qquad
\begin{tikzcd}
INS_I \arrow[dr, phantom, "(2)"] & INS_1 \arrow[l, "f_{1,I}"'] \\
INS_2 \arrow[u, "f_{2,I}"] & INS \arrow[l, "g_{2,I}"] \arrow[u, "g_{1,I}"']
\end{tikzcd}$$

Proof sketch. "$\Rightarrow$":

From Fact 4.27 we obtain pullback (2) in **SET**. Then Theorem 6.22 together with Fact 4.26 imply that $g_{2,I}$ is surjective which can be used to define the required morphism $x : INS_2 \to X$ for every pullback complement X. The requirements of x together with the fact that $g_{2,I}$ is an epimorphism imply that x is unique.

"$\Leftarrow$":

Part "$\Rightarrow$" implies that for the cominimal decomposition $KI_2' = (K_2', INIT_2', INS_2')$ defined in Theorem 6.17 the set INS_2' is also a minimal pullback and there is a surjective function $g_{2,I}' : INS \to INS_2'$. Then we can use the induced isomorphism $x : INS_2' \to INS_2$ and the decomposition of surjective functions to show that also $g_{2,I}$ is surjective and hence by Theorem 6.22 KI_2 is a cominimal decomposition.

For a detailed proof see Detailed Proof C.39 in the appendix. □

If and only if the instantiation part in a pushout complement in **AHLNetI** is a minimal pullback complement in **SET** then the pushout complement is maximal.

Lemma 6.26 (Characterization of Maximal Pushout Complements in **AHLNetI**)
Given AHL-nets with instantiations $KI_1 = (K_1, INS_1)$ and $KI = (K, INS)$, (I, INS_I) with $T_I = \emptyset$, injective AHL-morphisms $f_{1,N} : I \to K_1$ and $g_{1,N} : K_1 \to K$ and functions $f_{1,I} : INS_1 \to INS$, $g_{1,I} : INS \to INS_1$ such that $f_1 = (f_{1,N}, f_{1,I})$ and $g_1 = (g_{1,N}, g_{1,I})$ are **AHLNetI**-morphisms. Let $((K_2, INS_2), f_2, g_2)$ be a pushout complement of $(I, INS_I) \xrightarrow{f_1} (K_1, INS_1) \xrightarrow{g_1} (K, INS)$ in **AHLNetI**.
The pushout complement is maximal iff INS_2 is a minimal pullback complement of $INS \xrightarrow{g_{1,I}} INS_1 \xrightarrow{f_{1,I}} INS_I$ in **SET**.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "g_1"] \\
(K_2, INS_2) \arrow[r, "g_2"'] & (K, INS)
\end{tikzcd}
\qquad
\begin{tikzcd}
INS_I \arrow[dr, phantom, "(2)"] & INS_1 \arrow[l, "f_{1,I}"'] \\
INS_2 \arrow[u, "f_{2,I}"] & INS \arrow[l, "g_{2,I}"] \arrow[u, "g_{1,I}"']
\end{tikzcd}$$

Proof sketch. By Fact 4.27 the pushout (1) in the category **AHLNetI** implies pushouts in the underlying categories which in the case of instantiations results in a pullback in **SET** by duality. The maximality of (1) implies also maximality of the underlying pushouts, i.e. in the minimality of pullback (2) by duality.
The other way round minimality of pullback complement (2) in **SET** means that (2) is a maximal pushout complement in $\mathbf{SET}^{op}$ and since **AHLNet** has unique pushout complements all pushout complements in **AHLNet** are maximal pushout complements implying that (1) is a maximal pushout complement.
For a detailed proof see Detailed Proof C.40 in the appendix. □

From the characterization of cominimal decompositions by minimal pullback complements together with the characterization of maximal pushout complements we obtain a third characterization of cominimal decompositions.

Theorem 6.27 (Characterization of Cominimal Decompositions (III))
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.
Let $KI_2 = (K_2, INIT_2, INS_2)$ together with morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ be a decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.
The decomposition is cominimal iff for **AHLNetI**-object (I, INS_I) and **AHLNetI**-morphisms such that (1) is diagram in **AHLNetI** there is (K_2, INS_2) a maximal pushout complement of $(I, INS_I) \xrightarrow{(f_1, f_{1,I})} (K_1, INS_1) \xrightarrow{(g_1, g_{1,I})} (K, INS)$ in **AHLNetI**.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "(f_1, f_{1,I})"] \arrow[d, "(f_2, f_{2,I})"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "(g_1, g_{1,I})"] \\
(K_2, INS_2) \arrow[r, "(g_2, g_{2,I})"'] & (K, INS)
\end{tikzcd}$$

Proof. "$\Rightarrow$":

Due to Corollary 6.19 (1) is a pushout complement in **AHLNetI**.

Theorem 6.25 implies that (2) is a maximal pullback complement in **SET** which by Lemma 6.26 implies that (1) is maximal pushout complement in **AHLNetI**.

$$\begin{tikzcd} INS_I & INS_1 \arrow[l, "f_{1,I}"'] \\ INS_2 \arrow[u, "f_{2,I}"] \arrow[ur, phantom, "(2)"] & INS \arrow[l, "g_{2,I}"] \arrow[u, "g_{1,I}"'] \end{tikzcd}$$

"$\Leftarrow$":

Lemma 6.26 implies that (2) is a maximal pullback complement in **SET** which by Theorem 6.25 implies that KI_2 is a cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.

□

The characterization of cominimal decompositions by maximal pushout complements implies the uniqueness of cominimal decompositions.

Fact 6.28 (Uniqueness of Cominimal Decompositions of AHL-Occurrence Nets with Instantiations)
Given AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI = (K, INIT, INS)$, an AHL-occurrence net I with $T_I = \emptyset$ and injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$.
Let $KI_2 = (K_2, INIT_2, INS_2)$ together with morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ and $KI_2' = (K_2', INIT_2', INS_2')$ together with morphisms $f_2' : I \to K_2'$, $g_2' : K_2' \to K$ such that KI_2 and KI_2' are cominimal decompositions of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.
Then there exists an **AHLNetI**-isomorphism $x : (K_2, INS_2) \xrightarrow{\sim} (K_2', INS_2')$ such that for **AHLNetI**-object (I, INS_I) and **AHLNetI**-morphisms such that (1) and (2) are diagrams in **AHLNetI** there is

$$x \circ (f_2, f_{2,I}) = (f_2', f_{2,I}') \text{ and } (g_2', g_{2,I}') \circ x = (g_2, g_{2,I})$$

$$\begin{tikzcd} (I, INS_I) \arrow[r, "{(f_1,f_{1,I})}"] \arrow[d, "{(f_2,f_{2,I})}"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "{(g_1,g_{1,I})}"] \\ (K_2, INS_2) \arrow[r, "{(g_2,g_{2,I})}"'] & (K, INS) \end{tikzcd} \qquad \begin{tikzcd} (I, INS_I) \arrow[r, "{(f_1,f_{1,I})}"] \arrow[d, "{(f_2',f_{2,I}')}"'] \arrow[dr, phantom, "(2)"] & (K_1, INS_1) \arrow[d, "{(g_1,g_{1,I})}"] \\ (K_2', INS_2') \arrow[r, "{(g_2',g_{2,I}')}"'] & (K, INS) \end{tikzcd}$$

Proof. Due to Theorem 6.27 there are (K_2, INS_2) together with $(f_2, f_{2,I}), (g_2, g_{2,I})$ and (K_2', INS_2') together with $(f_2', f_{2,I}'), (g_2', g_{2,I}')$ maximal pushout complements of $(I, INS_I) \xrightarrow{(f_1,f_{1,I})} (K_1, INS_1) \xrightarrow{(g_1,g_{1,I})} (K, INS)$ in **AHLNetI**.
From the uniqueness of maximal pushout complements (6.3) follows the existence of an isomorphism $x : (K_2, INS_2) \to (K_2', INS_2')$ such that

$$x \circ (f_2, f_{2,I}) = (f_2', f_{2,I}') \text{ and } (g_2', g_{2,I}') \circ x = (g_2, g_{2,I})$$

□

A maximal pushout complement in **AHLNetI** is also a pushout complement in **AHLONetI**.

Corollary 6.29 (Maximal Pushout Complement of AHL-Occurrence Nets with Instantiations)
Given AHL-nets with instantiations $KI_1 = (K_1, INS_1)$ and $KI = (K, INS)$, (I, INS_I) with $T_I = \emptyset$, injective AHL-morphisms $f_1 : I \to K_1$ and $g_1 : K_1 \to K$ and functions $f_{1,I} : INS_1 \to INS_I$, $g_{1,I} : INS \to INS_1$ such that $(f_1, f_{1,I})$ and $(g_1, g_{1,I})$ are **AHLNetI**-morphisms. The maximal pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** is also a pushout complement in **AHLONetI** iff KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "g_1"] \\
(K_2, INS_2) \arrow[r, "g_2"'] & (K, INS)
\end{tikzcd}$$

Proof. "$\Rightarrow$":

By Lemma 6.26 there is INS_2 a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "g_1"] \\
(K_2, INS_2) \arrow[r, "g_2"'] & (K, INS)
\end{tikzcd}
\qquad
\begin{tikzcd}
INS_I \arrow[dr, phantom, "(2)"] & INS_1 \arrow[l, "f_{1,I}"'] \\
INS_2 \arrow[u, "f_{2,I}"] & INS \arrow[l, "g_{2,I}"] \arrow[u, "g_{1,I}"']
\end{tikzcd}$$

This implies by Theorem 6.25 that $KI_2 = (K_2, INIT_2, INS_2)$ is a cominimal decomposition of KI w.r.t. $I \overset{f_{1,N}}{\to} K_1 \overset{g_{1,N}}{\to} K$ which by Theorem 6.17 implies that KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

"$\Leftarrow$":

By Corollary 6.20 the pushout complement (K_2, INS_2) in **AHLNetI** is also pushout complement in **AHLONetI** because KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$. □

As a combination of the decomposition of AHL-processes and the decomposition of AHL-occurrence nets with instantiations we obtain the decomposition of AHL-processes with instantiations.

Theorem 6.30 (Decomposition of AHL-Processes with Instantiations)
Given an AHL-occurrence net I and AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$, $KI = (K, INIT, INS)$, injective AHL-morphisms $f_1 : K_0 \to K_1$, $g_1 : K_1 \to K_3$ and a morphism $mp : K \to AN$.
If $KI_2 = (K_2, INIT_2, INS_2)$ exists s.t. (1) is a composition diagram then there are morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ with $mp_1 = mp \circ g_1$ and $mp_2 = mp \circ g_2$.

$$\begin{tikzcd}
K_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "g_2"'] \arrow[drr, bend right, "mp_2"'] & K_3 \arrow[dr, "mp_3"] & \\
 & & AN
\end{tikzcd}$$

Proof. The existence of the morphisms follows from the well-defined composition of morphisms in the category **AHLNet**. □

Example 6.31 (Decomposition of AHL-Processes with Instantiations)
Consider the decomposition KI_2 of KI w.r.t. $I \to K_1 \to K$ in Example 6.18. In this context the decomposition $mp_2 : K_2 \to Alarm$ of the process $mp : K \to Alarm$ in Example 6.12 is an AHL-process with instantiations. △

Also the decomposition of AHL-processes with instantiations can be constructed as a pushout complement.

Corollary 6.32 (Pushout Complement of AHL-Processes with Instantiations)
Given an AHL-occurrence net I and AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$, $KI = (K, INIT, INS)$, injective AHL-morphisms $f_1 : K_0 \to K_1$, $g_1 : K_1 \to K_3$ and a morphism $mp_N : K \to AN$.
Let (I, INS_I) be an **AHLNetI**-object and $(f_1, f_{1,I}), (g_1, g_{1,I})$ **AHLNetI**-morphisms.
Then for the maximal pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** there exist **AHLNetI**-morphisms mp_0, mp_1, mp_2 and unique $mp = (mp_N, mp_I)$ such that mp_2 is a pushout complement of $mp_0 \overset{(f_1,f_{1,I})}{\to} mp_1 \overset{(g_1,g_{1,I})}{\to} mp$ in **AHLProcI**(**AN**) iff KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "(f_1, f_{1,I})"] \arrow[d, "(f_2, f_{2,I})"'] \arrow[dr, phantom, "(\mathbf{1})"] \arrow[ddrr, bend right=60, "mp_0"'] & (K_1, INS_1) \arrow[d, "(g_1, g_{1,I})"] \arrow[ddr, bend left, "mp_1"] & \\
(K_2, INS_2) \arrow[r, "(g_2, g_{2,I})"'] \arrow[drr, "mp_2"'] & (K, INS) \arrow[dr, "mp"] & \\
 & & (AN, \emptyset)
\end{tikzcd}$$

Proof sketch.

"⇒":
The pushout complement (1) in **AHLProcI**(**AN**) is also a pushout complement in **AHLONetI** which by Corollary 6.29 implies that KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

"$\Leftarrow$":

Since KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$ Corollary 6.29 implies that (1) is also a pushout complement in **AHLONetI**.

Furthermore the decomposability implies that the dangling condition is satisfied which allows to use Corollary 6.13 and the cofree functor $Inst$ to obtain the required unique morphisms such that (1) is also a pushout complement in **AHLProcI**(**AN**).

For a detailed proof see Detailed Proof C.41 in the appendix. □

7 Rule-based Transformation of Algebraic High-Level Processes

In this section we combine composition and decomposition to the rule-based transformation of processes in the double pushout approach.

7.1 Transformation of AHL-Processes

A production in the double pushout approach is a span $L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ of AHL-net morphisms.

Definition 7.1 (Production)
A span $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ of $\mathbf{C}$-morphisms is called a production in $\mathbf{C}$.

$$L \xleftarrow{l} I \xrightarrow{r} R$$

$\triangle$

If the category $\mathbf{C}$ has pushouts and pushout complements under certain conditions then a production in $\mathbf{C}$ can be used for the double pushout transformation of $\mathbf{C}$-objects.

Definition 7.2 (Double Pushout Transformation)
Given a production $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ and a morphism $m : L \to K$ in a category $\mathbf{C}$.
Then a double pushout transformation $K \overset{(p,m)}{\Rightarrow} K'$ in $\mathbf{C}$ is given by pushouts (1) and (2) in $\mathbf{C}$.
The morphism $m : L \to K$ is called match and the morphism $n : R \to K'$ is called comatch of the transformation.

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(2)"] & R \arrow[d, "n"] \\
K & C \arrow[l, "d"] \arrow[r, "e"'] & K'
\end{tikzcd}$$

$\triangle$

From the results of the Sections 5 and 6 we know that for the transformation of AHL-processes we require that the pushouts of a double pushout transformation are not only pushout diagrams but also composition diagrams and therefore we allow only productions with special properties for AHL-occurrence nets.

Definition 7.3 (Production for AHL-Occurrence Nets)
A span $L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ of injective AHL-net morphisms is called a production for AHL-occurrence nets if L, I and R are AHL-occurrence nets and $T_I = \emptyset$.

$$L \xleftarrow{l} I \xrightarrow{r} R$$

$\triangle$

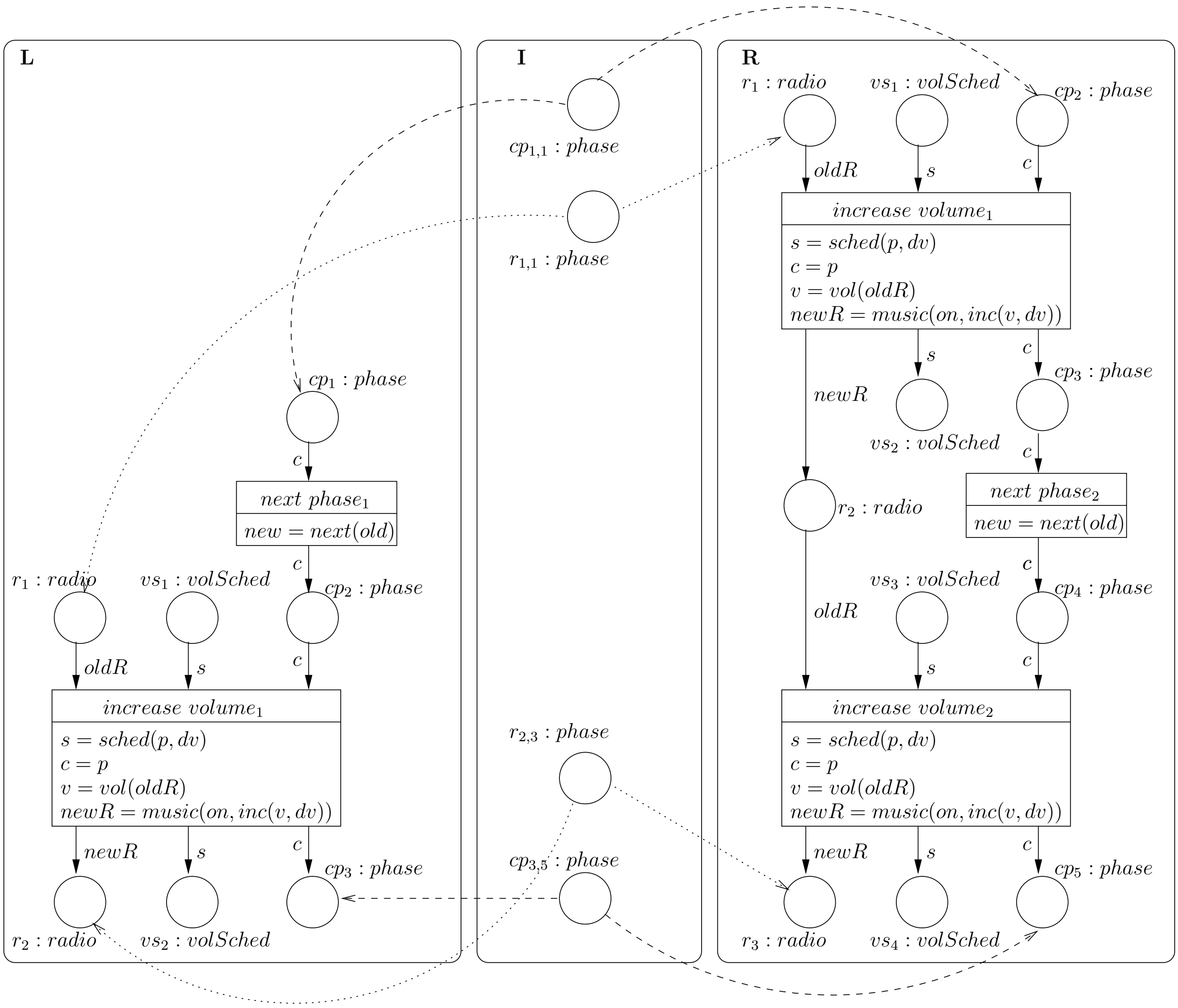


Figure 29: Production for AHL-Occurrence Nets $p : L \xleftarrow{l} I \xrightarrow{r} R$

Example 7.4 (Production for AHL-Occurrence Nets)
Fig. 29 shows an example of a production for AHL-occurrence nets which adds an increase of the volume in the phase before an existing increase of the volume.

△

Every production which newly creates a transition in the pre or post domain of a place at the right hand side of a production can be used to create a cycle in the double pushout approach. So the prevention of cycles is not a question of the type of production but of the applicability. To analyze if a transformation would create a cycle we define a gluing relation for transformations. This is a relation over the places of the interface derived from the causal relation of the part of the net on which the production should be applied without the parts which are deleted and the right hand side of the production, i.e. the parts that are newly created.

Definition 7.5 (Gluing Relation for Transformations)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ and a match $m : L \to K$.
We define the relations

$$\prec_{(K,m)} \subseteq (P_K \times (T_K \setminus m_T(T_L))) \uplus ((T_K \setminus m_T(T_L)) \times P_K)$$

$$\prec_{(K,m)} = \{(x, y) \mid x \in \bullet y\}$$

and $<_{(K,m)}$ as the transitive closure of $\prec_{(K,m)}$.
Furthermore we define

$$\prec_{(p,m)} \subseteq P_I \times P_I$$

$$\prec_{(p,m)} = \{(x, y) \mid m \circ l(x) <_{(K,m)} m \circ l(y) \vee r(x) <_R r(y)\}$$

The transitive closure $<_{(p,m)}$ of $\prec_{(p,m)}$ is called gluing relation of production p under match m. △

The gluing relation for transformations is quite similar to the induced causal relation for the composition of AHL-occurrence nets (see Def. 5.1).

Lemma 7.6 (Gluing Relation Lemma)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ and a match $m : L \to K$.
Let the AHL-occurrence net C together with morphisms $c : I \to C$ and $d : C \to K$ be the decomposition of K w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$.
Then for $x, y \in P_I$ there is

1. $m \circ l(x) <_{(K,m)} m \circ l(y) \Leftrightarrow c(x) <_C c(y)$ and

2. $x <_{(p,m)} y \Leftrightarrow x <_{(c,r)} y$

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] & R \\
K & C \arrow[l, "d"]
\end{tikzcd}$$

Proof sketch. The causal relation in an AHL-net depends on the pre and post conditions of its transitions. So the fact that $<_{(K,m)}$ is defined analogously to $<_K$ but without the relations induced by transitions which are not in C implies that we have the same elements in relation $<_{(K,m)}$ as in $<_C$.
Then part 2 follows directly from part 1 and the definitions of $<_{(p,m)}$ and $<_{(c,r)}$.
For a detailed proof see Detailed Proof C.42 in the appendix. □

Furthermore the application of a production for AHL-occurrence nets should not create conflicts. To ensure the conflict-freeness of the transformation result the conflict-freeness condition defines sets of in and out points which have to be input places and output places in the right hand side of the production, respectively.

Definition 7.7 (Conflict-Freeness Condition)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$, an AHL-occurrence net K and an injective morphism $m : L \to K$. The set of in points InP is defined as

$$InP = \{x \in P_I \mid l(x) \in IN(L) \text{ and } m \circ l(x) \notin IN(K)\}$$

and set set of out points $OutP$ is defined as

$$OutP = \{x \in P_I \mid l(x) \in OUT(L) \text{ and } m \circ l(x) \notin OUT(K)\}$$

The conflict-freeness condition is satisfied iff

$$r(InP) \subseteq IN(R) \text{ and } r(OutP) \subseteq OUT(R)$$

△

The extended gluing condition for productions for AHL-occurrence nets requires that all conditions are satisfied to ensure that the transformation result exists and it is an AHL-occurrence net with instantiations.

Definition 7.8 (Extended Gluing Condition for Productions for AHL-Occurrence Nets)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ and an AHL-occurrence net K. Then p satisfies the extended gluing condition under an injective (match) morphism $m : L \to K$ iff

- the dangling condition is satisfied,
- the gluing relation of p under m is a finitary strict partial order and
- the conflict-freeness condition is satisfied.

△

Theorem 7.9 (Construction of Direct Transformations of AHL-Occurrence Nets)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ and an AHL-occurrence net K together with an injective morphism $m : L \to K$.
If and only if p satisfies the extended gluing condition under m then there exists the decomposition C of K w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$ with composition diagram (1) and the composition $K' = C \circ_{(I,c,r)} R$ with composition diagram (2).
We call $K \overset{p,m}{\Rightarrow} K'$ a direct process preserving transformation.

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"'] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"'] \\
K & C \arrow[l, "d"'] \arrow[r, "e"] & K'
\end{tikzcd}$$

Proof sketch.

"$\Rightarrow$":
The satisfaction of the dangling condition ensures that there exists the decomposition (1) by Theorem 6.8. Then we can use the gluing relation lemma to obtain from the fact that the gluing relation of p under m is a finitary strict partial order that also $<_{(c,r)}$ is a finitary strict partial order. Finally the satisfaction of the conflict-freeness condition guarantees that there are the required input and output places in R such that (C, R) are composable w.r.t. (I, c, r) leading to composition (2).

"$\Leftarrow$":
The existence of the context net C implies the satisfaction of the dangling condition by by Theorem 6.8. Furthermore composition (2) implies by Theorem 5.5 that (C, R) are composable w.r.t. (I, c, r) which by the gluing relation lemma implies that the gluing relation is a finitary strict partial order.
The satisfaction of the conflict-freeness condition follows from the composabilities implied by compositions (1) and (2).

For a detailed proof see Detailed Proof C.43 in the appendix. □

Example 7.10 (Direct Transformation of AHL-Occurrence Nets)
The production p in Fig. 29 is applicable via a match m to the AHL-occurrence net K in Fig. 30. Therefore the context net C and the transformation result K' exist and C and K' are AHL-occurrence nets.

△

Corollary 7.11 (Double Pushout Transformation of AHL-Occurrence Nets)
Given a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ and an AHL-occurrence net K together with an injective morphism $m : L \to K$.
Let $K \overset{(p,m)}{\Rightarrow} K'$ be a double pushout transformation in **AHLNet**.
If and only if p satisfies the extended gluing condition under m then $K \overset{(p,m)}{\Rightarrow} K'$ is also a double pushout transformation in **AHLONet**.

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "\mathbf{(1)}"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "\mathbf{(2)}"] & R \arrow[d, "n"] \\ K & C \arrow[l, "d"] \arrow[r, "e"'] & K' \end{tikzcd}$$

Proof. "$\Rightarrow$":
The fact that p satisfies the extended gluing condition under m implies by Theorem 7.9 that there exists composition diagrams (3) and (4).

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "\mathbf{(3)}"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c'"] \arrow[dr, phantom, "\mathbf{(4)}"] & R \arrow[d, "n'"] \\ K & C' \arrow[l, "d'"] \arrow[r, "e'"'] & K'' \end{tikzcd}$$

Since composition diagrams are pushout diagrams in **AHLNet** and pushouts and pushout complements in **AHLNet** are unique also diagrams (1) and (2) are composition diagrams.

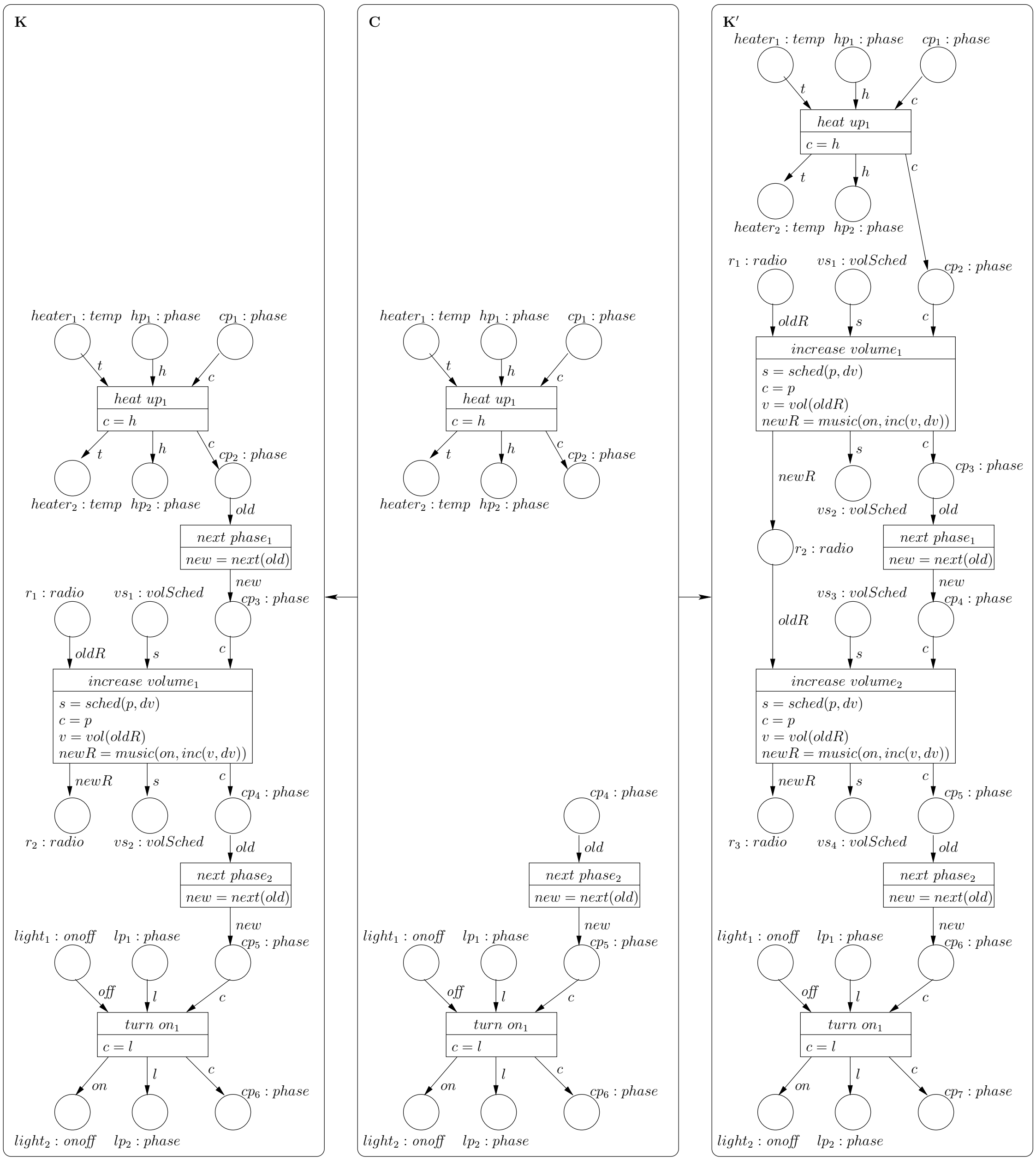


Figure 30: Direct Transformation $K \overset{(p,m)}{\Rightarrow} K'$

Then by Theorem 5.5 there are (L, C) composable w.r.t. (I, l, c) and (R, C) composable w.r.t. (I, r, c) which by Corollary 5.7 implies that (1) and (2) are pushouts in **AHLONet**.

"$\Leftarrow$":

The fact that pushouts (1) and (2) in **AHLNet** are also pushouts in **AHLONet** implies by Corollary 5.7 that (L, C) are composable w.r.t. (I, l, c) and (R, C) are composable w.r.t. (I, r, c).

This means by Theorem 5.5 that (1) and (2) are composition diagrams which by Theorem 7.9 implies that p satisfies the extended gluing condition under m.

□

For the rule-based transformation of processes we do also need information of how the newly added parts of the transformed AHL-occurrence net are matched via process morphism. Therefore we define productions for AHL-processes with an extra process morphism for the right hand side of the production.

Definition 7.12 (Production for AHL-Processes)
Given an AHL-net AN. A production for AHL-processes $pp = (p, rp)$ of AN is a production for AHL-occurrence nets $p : L \xleftarrow{l} I \xrightarrow{r} R$ together with a morphism $rp : R \to AN$. △

In the case of AHL-processes the extended gluing condition is extended by the property that the process morpism of the right hand side of the production has to be compatible with the match and the process morphism which is matched.

Definition 7.13 (Extended Gluing Condition for AHL-Processes)
Given an AHL-net AN and a production for AHL-processes $pp = (p, rp)$ of AN with $p : L \xleftarrow{l} I \xrightarrow{r} R$ and $rp : R \to AN$ and an AHL-process $mp : K \to AN$ together with an injective morphism $m : L \to N$. The production pp satisfies the extended gluing condition for AHL-processes under m iff

- p satisfies the extended gluing condition and
- $mp \circ m \circ l = rp \circ r$.

$$\begin{tikzcd} L \arrow[d, "m"'] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, phantom, "(=)"] & R \arrow[d, "rp"] \\ K \arrow[rr, "mp"'] & & AN \end{tikzcd}$$

△

Theorem 7.14 (Construction of Direct Transformation of AHL-Processes)
Given an AHL-net AN, a production for AHL-processes $pp = (p, rp)$ of AN and an AHL-process $mp : K \to AN$ together with an injective morphism $m : L \to N$.
If and only if p satisfies the extended gluing condition for AHL-processes under m then

there exist composition diagrams (1) and (2) and unique process morphisms $cp : C \to AN$, $mp' : K' \to AN$ such that

$$cp = mp \circ d = mp' \circ e \text{ and } rp = mp' \circ n$$

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \arrow[dd, bend left=60, "rp"] \\
K \arrow[drr, "mp"'] & C \arrow[l, "d"] \arrow[r, "e"'] \arrow[dr, "cp"] & N' \arrow[d, "mp'"] \\
 & & AN
\end{tikzcd}$$

Proof. "$\Rightarrow$":

We obtain composition diagrams (1) and (2) from Theorem 7.9.
Then we obtain a process morphism $cp : C \to AN$ by composition $cp = mp \circ d$ in the category **AHLNet**. Obviously for any $cp' : C \to AN$ with $cp' = mp \circ d$ there is $cp' = cp$.
Due to the commutativity of (1) there is

$$\begin{aligned} cp \circ c &= mp \circ d \circ c \\ &= mp \circ m \circ l \\ &= rp \circ r \end{aligned}$$

which by the pushout property of (2) implies that there is a unique morphism mp' : $K' \to AN$ such that

$$mp' \circ e = cp \text{ and } mp' \circ n = rp$$

"$\Leftarrow$":

By 7.9 the existence of composition diagrams (1) and (2) implies that p satisfies the extended gluing condition under match m. Furthermore from the given commutativities follows that

$$\begin{aligned} mp \circ m \circ l &= mp \circ d \circ c \\ &= cp \circ c \\ &= mp' \circ e \circ c \\ &= mp' \circ n \circ r \\ &= mp' \circ rp \end{aligned}$$

Hence pp satisfies the extended gluing condition for AHL-processes under match m.

$\square$

Theorem 7.15 (Double Pushout Transformation of AHL-Processes)
Given an AHL-net AN, a production for AHL-processes $pp = (p, rp)$ of AN and an AHL-process $mp : K \to AN$ together with an injective morphism $m : L \to N$.
If and only if pp satisfies the extended gluing condition for AHL-processes under m then the double pushout transformation $mp \overset{(p,m)}{\Rightarrow} mp'$ in **AHLNet** exists and is a double pushout transformation in **AHLProc(AN)**.

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[ddrr, bend right=60, "lp"'] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dl, phantom, "\mathbf{(1)}"] \arrow[dr, phantom, "\mathbf{(2)}"] & R \arrow[d, "n"] \arrow[dd, bend left=60, "rp"] \\
K & C \arrow[l, "d"'] \arrow[r, "e"] \arrow[dr, "cp"] & N' \arrow[d, "mp'"] \\
K \arrow[urr, phantom, ""] & & AN
\end{tikzcd}$$

Proof sketch.

"$\Rightarrow$":

Since p satisfies the extended gluing condition under m from Theorem 7.9 follows that the composition diagrams (1) and (2) exists and from Corollary 7.11 follows that $K \overset{(p,m)}{\Rightarrow} K'$ is a double pushout transformation in **AHLONet**. Due to the pushout properties and the well-defined composition of morphisms in **AHLONet** we can define the required process morphisms and it can be shown that they form a double pushout in **AHLProc**(**AN**).

"$\Leftarrow$":

The double pushout in **AHLProc**(**AN**) leads to pushouts in **AHLONet** Corollary 7.11 implies that p satisfies the extended gluing condition under match m. The required equation follows from the commuting diagrams (1) and (2) and the properties of **AHLProc**(**AN**)-morphisms.

For a detailed proof see Detailed Proof C.44 in the appendix. □

7.2 Transformation of AHL-Processes with Instantiations

For the transformation of AHL-occurrence nets with instantiations additionally to the information about changes in the structure of the transformed AHL-occurrence net we need information about the data of the new parts of the transformed instantiation.

Definition 7.16 (Production for AHL-Occurrence Nets with Instantiations)
Given a production for AHL-occurrence nets $p : L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R$ and sets of initializations $INIT_R$ and instantiations INS_R s.t. $(R, INIT_R, INS_R)$ is an AHL-occurrence net with instantiations. Then $pi = (p, INIT_R, INS_R)$ is called a production for AHL-occurrence nets with instantiations. △

Definition 7.17 (Direct Process Preserving Transformation of AHL-Occurrence Nets with Instantiations)
Given a production for AHL-occurrence nets with instantiations $pi = (p, INIT_R, INS_R)$ and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ together with an injective match $m : L \to K$. Let $RI = (R, INIT_R, INS_R)$.
A direct process preserving transformation of AHL-occurrence nets with instantiations denoted $KI \overset{pi,m}{\Rightarrow} KI'$ is defined by the cominimal decomposition $CI = (C, INIT_C, INS_C)$ of KI w.r.t. $I \overset{l}{\rightarrow} L \overset{m}{\rightarrow} K$ and the composition $KI' = CI \circ_{(I,c,r)} RI$ with composition diagrams (1) and (2).

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \\ K & C \arrow[l, "a"] \arrow[r, "b"'] & K' \end{tikzcd}$$

△

Definition 7.18 (Extended Gluing Condition for Productions for AHL-Occurrence Nets with Instantiations)
Given a production for AHL-occurrence nets with instantiations $pi = (p, INIT_R, INS_R)$ with $p : L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R$ and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$.
For an injective (match) morphism $m : L \rightarrow K$ the production pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations under match m iff

- p satisfies the extended gluing condition under match m, i.e.
  - the dangling condition is satisfied,
  - the gluing relation of p under m is a finitary strict partial order and
  - the conflict-freeness condition is satisfied,
- the induced preimage $LI = (L, INIT_L, INS_L)$ of KI and m exists,
- KI is decomposable w.r.t. $I \overset{l}{\rightarrow} L \overset{m}{\rightarrow} K$ leading to cominimal decomposition $CI = (C, INIT_C, INS_C)$ (see Theorem 6.17) and
- $|RetInit_{(I,c,r)}(INS_C, INS_R)| = |Composable_{(I,c,r)}(INS_C, INS_R)|$.

△

Theorem 7.19 (Construction of Direct Transformations of AHL-Occurrence Nets with Instantiations)
Given a production for AHL-occurrence nets with instantiations $pi = (p, INIT_R, INS_R)$ and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ together with an injective match $m : L \rightarrow K$.
Then there exists a direct process preserving transformation of AHL-occurrence nets with instantiations $KI \overset{pi,m}{\Rightarrow} KI'$ iff pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations under match m.

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \\ K & C \arrow[l, "a"] \arrow[r, "b"'] & K' \end{tikzcd}$$

Proof sketch. The first condition of the extended gluing condtion for productions of AHL-occurrence nets with instantiations is the necessary and sufficient condition that there are compositions (1) and (2) of AHL-occurrence nets.
The rest of the conditions are exactly the necessary and sufficient conditions such that decomposition (1) and composition (2) lead to AHL-occurrence nets with instantiations.
For a detailed proof see Detailed Proof C.45 in the appendix. □

Example 7.20 (Direct Transformation of AHL-Occurrence Nets with Instantiations)
Consider again the production p in Figure 29 and the net K in Figure 30. Let $KI = (K, INIT, INS)$ and $INS = \{L_{init}\}$ where L_{init} is depicted in Figure 31 and $INIT$ contains $IN(L_{init})$. Furthermore let $INS_R = \{L_{init_R}\}$ (see Figure 31) and $INIT_R$ containing $IN(L_{init_R})$. Then $(p, INIT_R, INS_R)$ is a production for AHL-occurrence nets with instantiations which is applicable to the net KI. The result is an AHL-occurrence net with instantiations $KI' = (K', INIT', INS')$ with the net K' in Figure 30 and the instantiation $L_{init'}$ in Figure 31.
While in the instantiation L_{init} the volume is increased in one single step to a hundred percent, in $L_{init'}$ the volume is first increased by half of the amount and then again in the next step.

△

If a production for AHL-occurrence nets is satisfying the extended gluing condition for AHL-occurrence nets with instantiations then production can be completed from the instantiations of the matched AHL-occurrence net with instantiations leading to a production in the category **AHLONetI**.

Fact 7.21 (Completion of Production for AHL-Occurrence Nets with Instantiations)
Given a production for AHL-occurrence nets with instantiations $pi = (p, INIT_R, INS_R)$ with $p : L \overset{l_N}{\leftarrow} I \overset{r_N}{\rightarrow} R$ and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ together with an injective match $m_N : L \rightarrow K$.
If pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations then we can define the completion (pii, m) of pi and m_N as the **AHLONetI**-production

$$pii : (L, INS_L) \overset{l}{\leftarrow} (I, INS_I) \overset{r}{\rightarrow} (R, INS_R)$$

together with the **AHLONetI**-morphism $m = (m_N, m_I)$ where (L, INS_L) is the preimage of KI and $m_N : L \rightarrow K$,

$$INS_I = PreIns(l)(INS_L) \cup PreIns(r)(INS_R)$$

and $l = (l_N, l_I)$, $r = (r_N, r_I)$ where the functions $m_I : INS \rightarrow INS_L, l_I : INS_L \rightarrow INS_I$ and $r_I : INS_R \rightarrow INS_I$ map to the respective instantiation preimages of the elements in their domain.

Proof. The fact that pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations implies that the preimage $(L, INIT_L, INS_L)$ of KI and m_N exists.
Since $T_I = \emptyset$ there is $P_I = IN(I)$ and therefore obviously the preimage (I, INS_I) as defined above is an **AHLONetI**-object and the morphism definitions above lead to well-defined **AHLONetI**-morphisms. □

The other way round for a given **AHLNetI**-production together with a match morphism we can construct a corresponding production for AHL-occurrence nets with instantiations together with a corresponding match morphism.

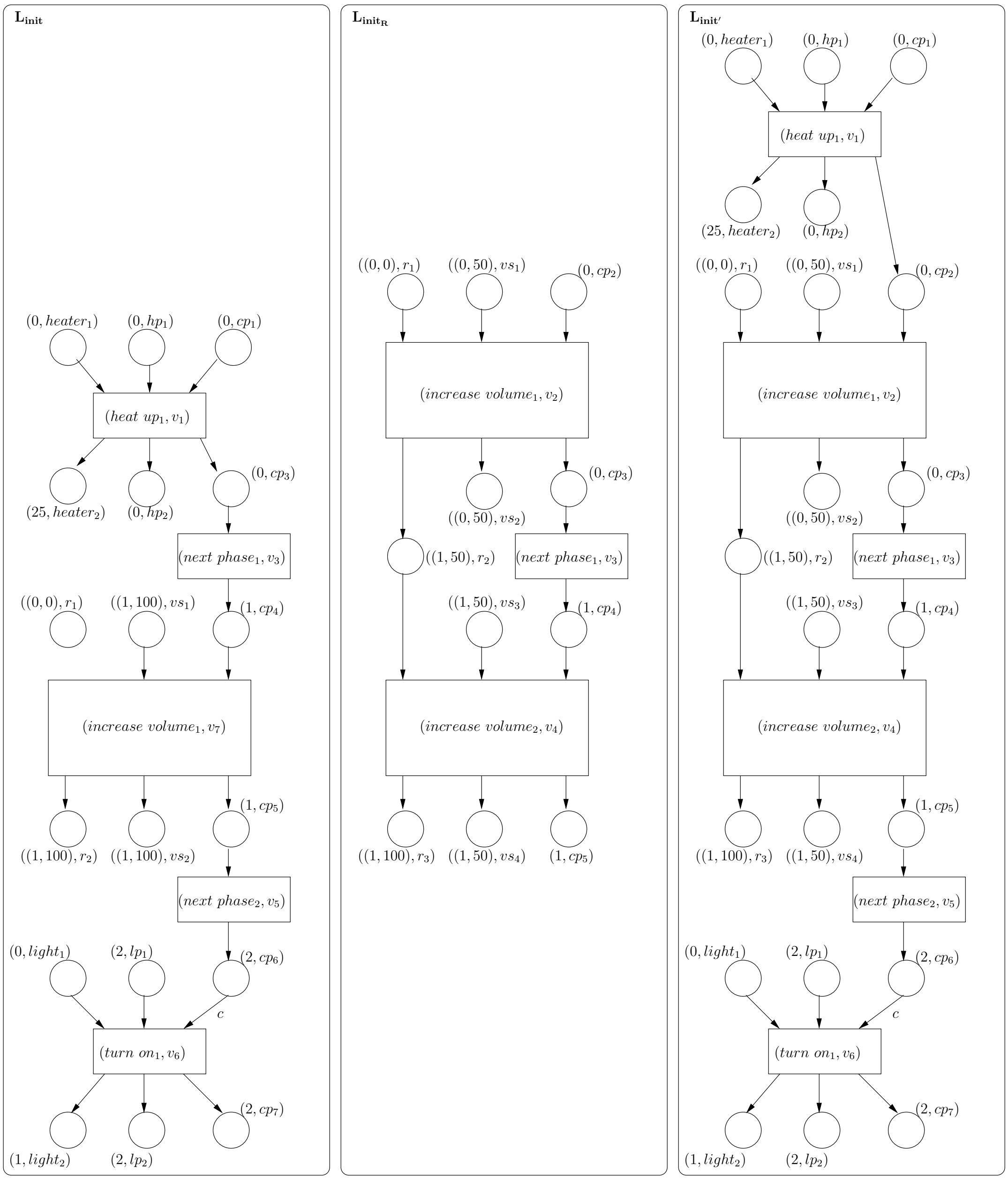


Figure 31: Instantiations of K, R and K', respectively

Fact 7.22 (Co-Completion of Production for AHL-Occurrence Nets with Instantiations)
Given an **AHLONetI**-production

$$pii : (L, INS_L) \stackrel{l}{\leftarrow} (I, INS_I) \stackrel{r}{\rightarrow} (R, INS_R)$$

and an **AHLNetI**-object (K, INS) together with a match $m = (m_N, m_I) : (L, INS_L) \to (K, INS)$ where m_N is injective. The co-completion of pii and m can be defined as the production for AHL-occurrence nets with instantiations $pi = (p, INIT_R, INS_R)$ and the match m_N where $p : L \stackrel{l_N}{\leftarrow} I \stackrel{r_N}{\rightarrow} R$ and $INIT_R = \{IN(L) \mid L \in INS_R\}$.

Proof. Due to the definition of **AHLONetI**-morphisms their net components are **AHLNet**-morphisms and therefore p is a production of AHL-occurrence nets and m_N an injective **AHLNet**-morphism. Furthermore due to the definition of **AHLONetI**-object there is $RI = (R, INIT_R, INS_R)$ an AHL-occurrence net with instantiations and hence pi is a production for AHL-occurrence nets with instantiations. there □

Corollary 7.23 (Double Pushout Transformation of AHL-Occurrence Nets with Instantiations)
Given a production $pii : (L, INS_L) \stackrel{l}{\leftarrow} (I, INS_I) \stackrel{r}{\rightarrow} (R, INS_R)$ in **AHLONetI** and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ together with a match $m : (L, INS_L) \to (K, INS)$ where m_N is injective and m_I is surjective.
Let $pi = (p, INIT_R, INS_R)$ and m_N be the co-completion of pii and m.
If and only if pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations under match $m_N : L \to K$ then the double pushout transformation $(K, INS) \stackrel{(pii,m)}{\Rightarrow} (K', INS')$ in **AHLNetI** via pushout diagrams (1) and (2) where (C, INS_C) is a maximal pushout complement exists and $(K, INS) \stackrel{(pii,m)}{\Rightarrow} (K', INS')$ is also a double pushout transformation in **AHLONetI**.

$$\begin{tikzcd}
(L, INS_L) \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & (I, INS_I) \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"'] \arrow[dr, phantom, "(2)"] & (R, INS_R) \arrow[d, "n"'] \\
(K, INS) & (C, INS_C) \arrow[l, "a"'] \arrow[r, "b"] & (K', INS')
\end{tikzcd}$$

Proof sketch. In Theorem 7.19 we have shown that the extended gluing condition for AHL-occurrence nets with instantiations is a necessary and sufficient condition for the transformation $(K, INS) \stackrel{(pi,m)}{\Rightarrow} (K', INS')$ via cominimal decomposition and composition. By Corollary 6.27 a cominimal decomposition is a maximal pushout complement in **AHLNetI** which by Corollary 6.19 is also a pushout complement in **AHLONetI**.
The composition of AHL-occurrence nets by Theorem 5.5 is a pushout in **AHLNetI** and by Corollary 5.28 also a pushout in **AHLONetI**.
For a detailed proof see Detailed Proof C.46 in the appendix. □

Definition 7.24 (Production for AHL-Processes with Instantiations)
Given an AHL-net AN. A production for AHL-processes with instantiations $ppi = (pi, rp)$ of AN is a production for AHL-occurrence nets $pi = (p, INIT_R, INS_R)$ with $p : L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R$ together with an AHL-morphism $rp : R \to AN$. △

Definition 7.25 (Extended Gluing Condition for AHL-Processes with Instantiations)
Given an AHL-net AN and a production for AHL-processes $ppi = (pi, rp)$ with $pi = (p, INIT_R, INS_R)$ and $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ together with an injective AHL-morphism $m : L \to K$ where $KI = (K, INIT, INS)$ is an AHL-occurrence net with instantiations.
The production ppi satisfies the extended gluing condition for AHL-processes with instantiations under match m iff

- pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations and
- (p, rp) satisfies the extended gluing condition for AHL-processes.

△

Theorem 7.26 (Construction of Direct Transformation of AHL-Processes with Instantiations)
Given an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ and a production for AHL-processes with instantiations $ppi = ((p, INIT_R, INS_R), rp)$ of an AHL-net AN with $p : L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R$.
Let AHL-process $mp : K \to AN$ together with an injective morphism $m : L \to K$.
If and only if ppi satisfies the extended gluing condition for AHL-processes with instantiations under match m then there exists a direct transformation of AHL-occurrence nets with instantiations $KI \overset{p,m}{\Rightarrow} KI'$ leading to composition diagrams (1) and (2) and there exist unique process morphisms $cp : C \to AN$, $mp' : K' \to AN$ such that

$$cp = mp \circ d = mp' \circ e \text{ and } rp = mp' \circ n$$

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \arrow[dd, bend left, "rp"] \\
K \arrow[drr, "mp"'] & C \arrow[l, "d"] \arrow[r, "e"'] \arrow[dr, "cp"] & N' \arrow[d, "mp'"] \\
 & & AN
\end{tikzcd}$$

Proof. ”⇒”:
The existence of the direct transformation of AHL-occurrence nets with instantiations $KI \overset{p,m}{\Rightarrow} KI'$ follows from Theorem 7.19.
Since this means that there is also a direct transformation of AHL-occurrence nets $K \overset{p,m}{\Rightarrow} K'$ the existence of the unique morphisms cp and mp' follows from Theorem 7.14.

”⇐”:
The existence of the direct transformation of AHL-occurrence nets with instantiations $KI \overset{p,m}{\Rightarrow} KI'$ by Theorem 7.19 implies that pi satisfies the extended gluing condition for AHL-occurrence nets with instantiations. The existence of the unique morphisms cp and mp' by Theorem 7.14 implies that (p, rp) satisfies the extended gluing condition for AHL-processes and hence ppi satisfies the extended gluing condition for AHL-processes with instantiations.

□

Example 7.27 (Direct Transformation of AHL-Processes with Instantiations)
Consider the transformation of an AHL-occurrence net with instantiations given in Example 7.20. We can define process morphisms $mp : K \to Alarm$ and $rp : R \to Alarm$ which map places and transitions to elements in $Alarm$ (see Figure 1) with the same name but without index.
The transformation $KI \overset{(pi,m)}{\Rightarrow} KI'$ leads to an AHL-occurrence net with instantiations together with a morphism $mp' : K' \to Alarm$ which maps the elements in K' in the same way as mp and rp map the elements of K and R, respectively. The morphism mp' is an AHL-process with instantiations. △

Theorem 7.28 (Double Pushout Transformation of AHL-Processes with Instantiations)
Given an AHL-process with instantiations $mp : (K, INS) \to Inst(AN)$ with AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ and an **AHLONetI**-production $pii : (L, INS_L) \overset{l}{\leftarrow} (I, INS_I) \overset{r}{\to} (R, INS_R)$ together with an **AHLONetI**-morphism $m : (L, INS_L) \to (K, INS)$ where m_N is injective and m_I is surjective.
Let $pi = (p, INIT_R, INS_R)$ and m_N be the co-completion of pii and m, and let $rp = (rp_N, rp_I) : (R, INS_R) \to Inst(AN)$ be an **AHLNetI**-morphism.
If and only if (pi, rp) satisfies the extended gluing condition for AHL-processes with instantiations under match m_N then a double pushout transformation $(K, INS) \overset{(pi,m)}{\Rightarrow} (K', INS')$ in **AHLNetI** via pushouts (1) and (2) where (C, INS_C) is a maximal pushout complement exists and is also a double pushout transformation in **AHLProcI**(**AN**).

$$\begin{tikzcd}
(L,INS_L) \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] \arrow[ddrr, bend right, "lp"'] & (I,INS_I) \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(2)"] & (R,INS_R) \arrow[d, "n"] \arrow[dd, bend left, "rp"] \\
(K,INS) \arrow[drr, "mp"'] & (C,INS_C) \arrow[l, "d"] \arrow[r, "e"] \arrow[dr, "cp"] & (K',INS') \arrow[d, "mp'"] \\
& & Inst(AN)
\end{tikzcd}$$

Proof sketch. By Corollary 7.23 the double pushout transformation in **AHLNetI** is a double pushout transformation in **AHLONetI** iff the extended gluing condition is satisfied. The pushout property ensures unique process morphisms such that the pushout in **AHLONetI** is exactly the pushout in **AHLProcI**(**AN**).
For a detailed proof see Detailed Proof C.47 in the appendix. □

8 Amalgamation of Algebraic High-Level Processes

In order to establish a clear relationship between the well-known pushout composition of AHL-nets and the composition of corresponding processes as presented in Section 5 we introduce the amalgamation of AHL-processes analogously to the amalgamation of open net processes in [BCEH01] for open low-level petri nets.
In Section 5 we define a composition of AHL-processes of a given AHL-net AN as a pushout in the category $\mathbf{AHLProc(AN)}$. In this section we consider processes of different AHL-nets AN_1 and AN_2. Given a composition of the nets AN_1 and AN_2 via a span of AHL-morphisms $f_1 : AN_0 \to AN_1$ and $f_2 : AN_0 \to AN_2$ we would like to have a construction which provides a corresponding composition of the processes of AN_1 and AN_2 in a compositionally way.

8.1 Amalgamation of Algebraic High-Level Processes

To define a compositional relationship between processes of AHL-nets we introduce the concept of the projection of AHL-processes analogously to the projection of open net processes in [BCEH01]. Given a process mp_2 of an AHL-net AN_2 and an AHL-morphism $f : AN_1 \to AN_2$ the morphism mp_2 can be projected along f leading to a corresponding process mp_1 of AN_1. In the case of an injective morphism f the projection intuitively means that the process mp_2 is restricted to the subnet AN_1.

Definition 8.1 (Projection of AHL-Processes)
Given an AHL-process $mp_2 : K_2 \to AN_2$ and an AHL-morphism $f : AN_1 \to AN_2$. A projection (mp_1, ϕ) of mp_2 along f is an AHL-process $mp_1 : K_1 \to AN_1$ together with an AHL-morphism $\phi : K_1 \to K_2$ defined by the pullback (1) in $\mathbf{AHLNet}$.

$$\begin{tikzcd} K_1 \arrow[r, "\phi"] \arrow[d, "mp_1"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_2 \arrow[d, "mp_2"] \\ AN_1 \arrow[r, "f"'] & AN_2 \end{tikzcd}$$

△

Well-definedness. The category $\mathbf{AHLNet}$ has pullbacks. $mp_1 : K_1 \to AN_1$ is an AHL-process because K_1 is an AHL-occurrence net which follows by Fact 2.14 from the AHL-morphism $\phi : K_1 \to K_2$ and the fact that K_2 is an AHL-occurrence net. □

Based on the projection of AHL-processes we can define a suitable condition under which we can continue the composition of AHL-nets via a span of AHL-morphisms $f_1 : AN_0 \to AN_1$ and $f_2 : AN_0 \to AN_2$ to a composition of their processes. Two processes mp_1 and mp_2 of AN_1 and AN_2, respectively, "agree" on the net AN_0 if we can construct a common projection of the processes leading to a process of AN_0 which can be used as a composition interface for mp_1 and mp_2.

Definition 8.2 (Agreement of AHL-Processes)
Given two AHL-processes $mp_1 : K_1 \to AN_1$ and $mp_2 : K_2 \to AN_2$ and two AHL-morphisms $f_1 : AN_0 \to AN_1$, $f_2 : AN_0 \to AN_2$. The processes mp_1 and mp_2 agree on AN_0 if there

exist projections (mp_0, ϕ_i) of mp_i along f_i for $i \in \{1, 2\}$ such that for $mp_0 : K_0 \to AN_0$ the AHL-occurrence nets (K_1, K_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2).
(mp_0, ϕ_1) and (mp_0, ϕ_2) are called agreement projections for mp_1 and mp_2.

$$\begin{tikzcd}
 & & K_0 \arrow[d, "mp_0"] \arrow[dll, "\phi_1"'] \arrow[drr, "\phi_2"] & & \\
 & (\mathbf{PB}) & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & (\mathbf{PB}) & \\
K_1 \arrow[r, "mp_1"'] & AN_1 & & AN_2 & K_2 \arrow[l, "mp_2"]
\end{tikzcd}$$

△

If two processes $mp_1 : K_1 \to AN_1$ and $mp_2 : K_2 \to AN_2$ agree then they can be amalgamated. This means that they are composed to a process $mp_3 : K_3 \to AN_3$ of the composition of AN_1 and AN_2 such that mp_1 and mp_2 are projections of mp_3.

Definition 8.3 (Amalgamation of AHL-Processes)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{N_0} = \emptyset$. Let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2, 3\}$ and let (mp_0, ϕ_1) and (mp_0, ϕ_2) be agreement projections for mp_1 and mp_2.
Then mp_3 is called amalgamation of mp_1 and mp_2, written $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$, if there exist projections (mp_1, ψ_1) and (mp_2, ψ_2) of mp_3 along g_1 and g_2, respectively, such that the outer square is a pushout in **AHLNet**.

$$\begin{tikzcd}
 & & K_0 \arrow[d, "mp_0"] \arrow[dll, "\phi_1"'] \arrow[drr, "\phi_2"] & & \\
 & (\mathbf{PB}) & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & (\mathbf{PB}) & \\
K_1 \arrow[r, "mp_1"'] \arrow[ddrr, "\psi_1"'] & AN_1 \arrow[dr, "g_1"'] & (\mathbf{PO}) & AN_2 \arrow[dl, "g_2"] & K_2 \arrow[l, "mp_2"] \arrow[ddll, "\psi_2"] \\
 & (\mathbf{PB}) & AN_3 & (\mathbf{PB}) & \\
 & & K_3 \arrow[u, "mp_3"] & &
\end{tikzcd}$$

△

Using the composition of AHL-occurrence nets in Section 5 we can state a construction for the amalgamation of processes.

Theorem 8.4 (Amalgamation Construction for AHL-Processes)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2. Let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2\}$ and let (mp_0, ϕ_1) and (mp_0, ϕ_2) be

agreement projections for mp_1 and mp_2.
Then the amalgamation $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$ is a process $mp_3 : K_3 \to AN_3$ where the AHL-occurrence net K_3 can be obtained as composition of AHL-occurrence nets $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ and the morphism $mp_3 : K_3 \to AN_3$ is the unique morphism induced by the pushout property of the corresponding pushout in **AHLNet**. Hence $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$ is unique up to isomorphism.

$$
\begin{tikzcd}
 & & K_0 \arrow[d, "mp_0"] \arrow[dll, "\phi_1"'] \arrow[drr, "\phi_2"] & & \\
 & \arrow[r, phantom, "(\mathbf{PB})"] & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] \arrow[r, phantom, "(\mathbf{PB})"] & \, & \\
K_1 \arrow[r, "mp_1"'] \arrow[ddrr, "\psi_1"'] & AN_1 \arrow[dr, "g_1"'] & (\mathbf{PO}) & AN_2 \arrow[dl, "g_2"] & K_2 \arrow[l, "mp_2"] \arrow[ddll, "\psi_2"] \\
 & (\mathbf{1}) & AN_3 & (\mathbf{2}) & \\
 & & K_3 \arrow[u, dashed, "mp_3"] & &
\end{tikzcd}
$$

Proof. The fact that (mp_0, ϕ_1) and (mp_0, ϕ_2) are agreement projections for mp_1 and mp_2 implies that (K_1, K_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2) which by Theorem 5.5 implies that the composition $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ is an AHL-occurrence net. Since the composition is defined as pushout in **AHLNet** and there is

$$g_1 \circ mp_1 \circ phi_1 = g_2 \circ mp_2 \circ phi_2$$

the pushout property implies a unique AHL-morphism $mp_3 : K_3 \to AN_3$ such that (1) and (2) commute.
From the fact that f_1, f_2, g_1 and g_2 are injective it follows that the above diagram is a weak Van Kampen cube with pushouts as top and bottom faces and pullbacks as back faces. The Van Kampen property implies that (1) and (2) are pullbacks and hence mp_3 is the amalgamation $mp_1 \circ_{\phi_1,\phi_2} mp_2$. □

Vice versa given the composition AN_3 of AHL-nets AN_1 and AN_2 we can state a construction which provides AHL-processes mp_1 of AN_1 and mp_2 of AN_2 which agree and mp_3 is the amalgamation of mp_1 and mp_2.

Theorem 8.5 (Amalgamation Decomposition of AHL-Processes)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{AN_0} = \emptyset$. Let $mp_3 : K_3 \to AN_3$ be an AHL-process and let (mp_1, ψ_1), (mp_2, ψ_2) be projections of mp_3 along g_1 and g_2, respectively. Then mp_3 can be recovered as a suitable amalgamation of mp_1 and mp_2.

$$
\begin{tikzcd}
 & & K_0 \arrow[dll, dashed, "\phi_1"'] \arrow[d, dashed, "mp_0"] \arrow[drr, dashed, "\phi_2"] & & \\
 & \textbf{(3)} & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \textbf{(4)} & \\
K_1 \arrow[r, dashed, "mp_1"'] \arrow[drr, dashed, "\psi_1"'] & AN_1 \arrow[dr, "g_1"'] & \textbf{(PO)} & AN_2 \arrow[dl, "g_2"] & K_2 \arrow[l, dashed, "mp_2"] \arrow[dll, dashed, "\psi_2"] \\
 & \textbf{(1)} & AN_3 & \textbf{(2)} & \\
 & & K_3 \arrow[u, "mp_3"] & &
\end{tikzcd}
$$

Proof. The projections (mp_1, ψ_1) and (mp_2, ψ_2) imply pullbacks (1) and (2) in **AHLNet**. We obtain the projection (mp_0, ϕ_1) of mp_1 along f_1 as pullback (3) in **AHLNet**. Since the pullback in **AHLNet** along the injective morphism f_1 can be constructed componentwise is **SET** the pullback property of the transition component together with the fact that $T_{AN_0} = \emptyset$ implies that $T_{K_0} = \emptyset$.
Furthermore there is

$$g_2 \circ f_2 \circ mp_0 = g_1 \circ f_1 \circ mp_0 = g_1 \circ mp_1 \circ \phi_1 = mp_3 \circ \psi_1 \circ \phi_1$$

which by the pullback property of (2) implies that there is a unique AHL-morphism $\phi_2 : K_0 \to K_2$ such that (2) and the outer square above commute.
Due to the composition of pullbacks (1)+(3) is a pullback. Since (PO) and the outer square commute there is (1)+(3) = (2)+(4) which implies that (2)+(4) is a pullback and hence by pullback decomposition (4) is a pullback.
So we have that the above cube is a weak Van Kampen cube where all side faces are pullbacks and the bottom is a pushout. Hence the Van Kampen property implies that the top face (i.e. the outer square) is a pushout in **AHLNet**.
Due to Fact 2.14 the nets K_0, K_1 and K_2 are AHL-occurrence nets. This implies that the outer square is a composition of AHL-occurrence nets which by Theorem 5.5 implies that (K_1, K_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2). Hence (mp_0, ϕ_1) and (mp_0, ϕ_2) are agreement projections for mp_1 and mp_2 which means that mp_3 is an amalgamation of mp_1 and mp_2. □

The results of amalgamation composition and decomposition constructions are unique up to isomorphism. In order to capture the bijective correspondence of these constructions we define isomorphism classes of AHL-processes and spans of AHL-processes analogously to isomorphism classes of open net processes and spans of these processes in [BCEH01].

An isomorphism between processes $mp : K \to AN$ and $mp' : K' \to AN$ of an AHL-net AN is an isomorphism $iso : K \to K'$ in the category **AHLNet** which is also a morphism in **AHLProc(AN)**, i.e. diagram (1) below commutes. We denote the isomorphism class of a process mp as the set $[mp] = \{mp' \mid mp' \cong mp\}$ of all processes which are isomorphic to mp.

$$\begin{tikzcd} K \arrow[rr, "iso"] \arrow[dr, "mp"'] & (1) & K' \arrow[dl, "mp'"] \\ & AN & \end{tikzcd}$$

An isomorphism of spans of processes $(mp_1 \xleftarrow{\phi_1} mp_0 \xrightarrow{\phi_2} mp_2) \cong (mp'_1 \xleftarrow{\phi'_1} mp'_0 \xrightarrow{\phi'_2} mp'_2)$ means that there are process isomorphisms $iso_i : mp_i \to mp'_i$ such that the following diagram commutes.

$$\begin{tikzcd} & K_1 \arrow[dl, "iso_1"'] \arrow[ddl, "mp_1"] & & K_0 \arrow[ll, "\phi_1"] \arrow[rr, "\phi_2"] \arrow[dl, "iso_0"'] \arrow[ddl, "mp_0"] & & K_2 \arrow[dl, "iso_2"'] \arrow[ddl, "mp_2"] \\ K'_1 \arrow[d, "mp'_1"'] & & K'_0 \arrow[ll, "\phi'_1"] \arrow[rr, "\phi'_2"] \arrow[d, "mp'_0"'] & & K'_2 \arrow[d, "mp'_2"'] & \\ AN_1 & & AN_0 \arrow[ll, "f_1"] \arrow[rr, "f_2"] & & AN_2 & \end{tikzcd}$$

Definition 8.6 (Sets of Isomorphism Classes)
The set of all isomorphism classes of processes of a given AHL-net AN is defined as

$$Proc(AN) = \{[mp] \mid mp : K \to AN \text{ is a process } \}$$

The set of all isomorphism classes of spans of agreeing processes with respect to a given span of AHL-morphisms is defined as

$$Proc(AN_1 \xleftarrow{f_1} AN_0 \xrightarrow{f_2} AN_2) =$$

$$\{\left[mp_1 \xleftarrow{\phi_1} mp_0 \xrightarrow{\phi_2} mp_2\right] \mid \phi_1, \phi_2 \text{ are agreement projections of } mp_1, mp_2 \text{ along } f_1, f_2\}$$

△

Analogously to the Amalgamation Theorem of open net processes in [BCEH01] the Amalgamation Theorem states the bijective correspondence between amalgamation composition and decomposition constructions of AHL-processes.

Theorem 8.7 (Amalgamation Theorem for AHL-Processes)
Given the pushout (PO) in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{AN_0} = \emptyset$. Then there are composition and decomposition functions

$$Comp : Proc(AN_1 \xleftarrow{f_1} AN_0 \xrightarrow{f_2} AN_2) \to Proc(AN)$$

and

$$Decomp : Proc(AN) \to Proc(AN_1 \xleftarrow{f_1} AN_0 \xrightarrow{f_2} AN_2)$$

establishing a bijective correspondence between $Proc(AN)$ and $Proc(AN_1 \xleftarrow{f_1} AN_0 \xrightarrow{f_2} AN_2)$.

$$\begin{tikzcd}
 & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\
AN_1 \arrow[dr, "g_1"'] & (\mathbf{PO}) & AN_2 \arrow[dl, "g_2"] \\
 & AN_3 &
\end{tikzcd}$$

Proof sketch. The functions $Comp$ and $Decomp$ can be defined as the isomorphism class of the result of the amalgamation construction in Theorem 8.4 and the amalgamation decomposition in Theorem 8.5, respectively, for a representative of the given isomorphism class.
Due to the uniqueness of pushouts and pullbacks up to isomorphism the definitions of $Comp$ and $Decomp$ lead to well-defined functions which are inverse to each other.
For a detailed proof see Detailed Proof C.48 in the appendix. □

Example 8.8 (Amalgamation of AHL-Processes)
Figure 32 shows an amalgamation of AHL-processes of AHL-nets GE and $NEXT$. The nets GE and $NEXT$ are glued together at the single places in the net $GE/NEXT$. The process $mp_3 : K_3 \to GE/NEXT$ is an amalgamation of the processes $mp_1 : K_1 \to GE$ and $mp_2 : K_2 \to NEXT$. Vice versa the processes mp_1 and mp_2 are decompositions of mp_3.

△

The example illustrates the important meaning of our compositionality result for the amalgamation of AHL-processes. The processes of the composed net $GE/NEXT$ can be described by the processes of its part GE and $NEXT$. Vice versa the processes of the smaller nets GE and $NEXT$ can be described by processes of the composed net $GE/NEXT$. So by the modification of AHL-nets by composition or decomposition we compositionally obtain also the process semantics of the modificated nets.

8.2 Amalgamation of Algebraic High-Level Processes with Instantiations

In the case of AHL-processes with instantiations the agreement of two processes additionally requires that the corresponding AHL-occurrence nets with instantiations are composable.

Definition 8.9 (Agreement of AHL-Processes with Instantiations)
Given two AHL-occurrence nets with instantiations $KI_1 = (K_1, INIT_1, INS_1)$ and $KI_2 = (K_2, INIT_2, INS_2)$ together with AHL-processes with instantiations $mp_1 : K_1 \to AN_1$ and $mp_2 : K_2 \to AN_2$ and two AHL-morphisms $f_1 : AN_0 \to AN_1$, $f_2 : AN_0 \to AN_2$.
The processes with instantiations mp_1 and mp_2 agree on AN_0 if there exist projections (mp_0, ϕ_i) of mp_i along f_i for $i \in \{1,2\}$ such that for $mp_0 : K_0 \to AN_0$ the AHL-occurrence nets with instantiations (KI_1, KI_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2).
(mp_0, ϕ_1) and (mp_0, ϕ_2) are called agreement projections for AHL-processes with instantiations mp_1 and mp_2.

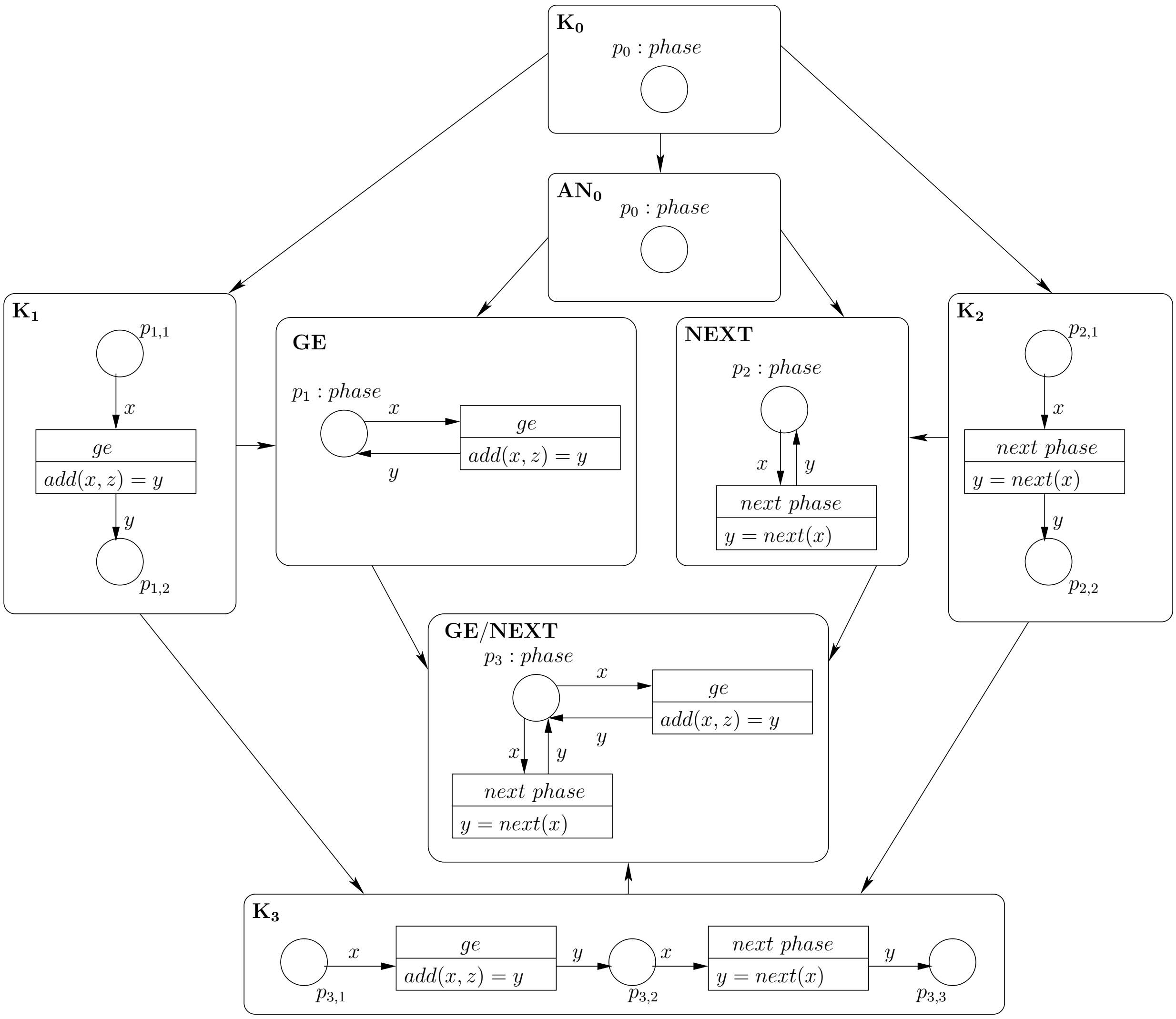


Figure 32: Amalgamation of AHL-Processes

$$\begin{tikzcd}
 & & K_0 \arrow[ddll, "\phi_1"'] \arrow[d, "mp_0"] \arrow[ddrr, "\phi_2"] & & \\
 & (\mathbf{PB}) & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & (\mathbf{PB}) & \\
K_1 \arrow[r, "mp_1"'] & AN_1 & & AN_2 & K_2 \arrow[l, "mp_2"]
\end{tikzcd}$$

△

Definition 8.10 (Amalgamation of AHL-Processes with Instantiations)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{N_0} = \emptyset$. Let $KI_i = (K_i, INIT_i, INS_i)$ for $i \in \{1, 2, 3\}$ be AHL-occurrence nets with instantiations and let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2, 3\}$ and let (mp_0, ϕ_1) and (mp_0, ϕ_2) be agreement projections for AHL-processes with instantiations mp_1 and mp_2. Then mp_3 is called amalgamation of AHL-processes with instantiations mp_1 and mp_2, written $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$, if there exist projections (mp_1, ψ_1) and (mp_2, ψ_2) of mp_3 along g_1 and g_2, respectively, such that the outer square is a pushout in **AHLNet** and there is $KI_3 = KI_1 \circ_{(K_0,\phi_1,\phi_2)} KI_2$.

$$\begin{tikzcd}
 & & K_0 \arrow[ddll, "\phi_1"'] \arrow[d, "mp_0"] \arrow[ddrr, "\phi_2"] & & \\
 & (\mathbf{PB}) & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & (\mathbf{PB}) & \\
K_1 \arrow[r, "mp_1"'] \arrow[ddrr, "\psi_1"'] & AN_1 \arrow[dr, "g_1"'] & (\mathbf{PO}) & AN_2 \arrow[dl, "g_2"] & K_2 \arrow[l, "mp_2"] \arrow[ddll, "\psi_2"] \\
 & (\mathbf{PB}) & AN_3 & (\mathbf{PB}) & \\
 & & K_3 \arrow[u, "mp_3"] & &
\end{tikzcd}$$

△

If two AHL-processes with instantiations agree then the almagamation construction for these processes leads to an AHL-process with instantiations.

Theorem 8.11 (Amalgamation Construction for AHL-Processes with Instantiations)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{N_0} = \emptyset$. Let $KI_i = (K_i, INIT_i, INS_i)$ for $i \in \{1, 2\}$ be AHL-occurrence nets with instantiations and let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2\}$ and let (mp_0, ϕ_1) and (mp_0, ϕ_2) be agreement projections for AHL-processes with instantiations mp_1 and mp_2.
Then the amalgamation mp_3 of AHL-processes with instantiations mp_1 and mp_2 can be obtained as composition of AHL-occurrence nets with instantiations $KI_3 = KI_1 \circ_{(K_0,\phi_1,\phi_2)} KI_2$ and the morphism $mp_3 : K_3 \to AN_3$ is induced by the corresponding pushout $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ in **AHLNet**.

$$
\begin{tikzcd}
 & & K_0 \arrow[lld, "\phi_1"'] \arrow[d, "mp_0"] \arrow[rrd, "\phi_2"] & & \\
 & \textbf{(PB)} & AN_0 \arrow[ld, "f_1"'] \arrow[rd, "f_2"] & \textbf{(PB)} & \\
K_1 \arrow[r, "mp_1"'] \arrow[rrdd, "\psi_1"'] & AN_1 \arrow[rd, "g_1"] & \textbf{(PO)} & AN_2 \arrow[ld, "g_2"'] & K_2 \arrow[l, "mp_2"] \arrow[lldd, "\psi_2"] \\
 & \textbf{(1)} & AN_3 & \textbf{(2)} & \\
 & & K_3 \arrow[u, dashed, "mp_3"] & &
\end{tikzcd}
$$

Proof. From agreement projections (mp_0, ϕ_1) and (mp_0, ϕ_2) for AHL-processes with instantiations mp_1 and mp_2 follows that (KI_1, KI_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2) which be Theorem 5.25 implies that there exists the AHL-occurrence net with instantiations $KI_3 = KI_1 \circ_{(K_0,\phi_1,\phi_2)} KI_2$. Since the composition of AHL-occurrence nets with instantiations is defined over the composition $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ in **AHLNet** we have by Theorem 8.4 that mp_3 is the amalgamation of AHL-processes mp_1 and mp_2 which implies that there are projections (mp_1, ψ_1) and (mp_2, ψ_2) of mp_3 along g_1 and g_2, respectively. $\square$

For a decomposition construction of the amalgamation of AHL-processes with instantiations we need another decomposability condition. The decomposability of an AHL-process with instantiations KI as introduced in Section 6 requires that one of the two parts of KI already is an AHL-occurrence net with instantiations. Given a composition of AHL-occurrence nets $K = K_1 \circ_{(K_0,f_1,f_2)} K_2$ and an AHL-occurrence net with instantiations $KI = (K, INIT, INS)$ the notion of mutual decomposability means that we can construct the induced preimage KI_1 for the part K_1 of the net K and then decompose KI with respect to KI_1 leading to the AHL-occurrence net with instantiations KI_2 for the other part K_2 of K. Since the condition is defined symmetrically it means that each of the two parts of the net KI is decomposable with respect to the other one.

Definition 8.12 (Mutual Decomposability of AHL-Occurrence Nets with Instantiations)
Given an the composition of AHL-occurrence nets $K = K_1 \circ_{(K_0,f_1,f_2)} K_2$ (see diagram below) and the AHL-occurrence net with instantiations $KI = (K, INIT, INS)$.
KI is mutually decomposable w.r.t. composition (1) iff

- for $i \in \{1, 2\}$ the functions

$$IN_i : PreIns(g_i)(INS) \to PreInit(g_i)(INS)$$

  with

$$IN_i(PreIns(g_i)(L_{init})) = PreInit(g_i)(L_{init})$$

  are injective

- and for all $L_{init}, L_{init'} \in INS$:
  If there is

$$(PreIns(g_1)(L_{init}), PreIns(g_2)(L_{init'})) \in Composable_{(K_0,f_1,f_2)}$$

then there is

$$PreIns(g_1)(L_{init}) \circ_{(J,j_1,j_2)} PreIns(g_2)(L_{init'}) \in INS$$

where (J, j_1, j_2) is the instantiation interface induced by (K_0, f_1, f_2).

$$\begin{tikzcd} K_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

△

Lemma 8.13 (Mutual Decomposability of AHL-Occurrence Nets with Instantiations)
Given an the composition of AHL-occurrence nets $K = K_1 \circ_{(K_0,f_1,f_2)} K_2$ (see diagram below) and the AHL-occurrence net with instantiations $KI = (K, INIT, INS)$.
If KI is mutually decomposable w.r.t. composition (1) then the induced preimage $KI_1 = (K_1, INIT_1, INS_1)$ of KI and g_1 exists together with a function $g_{1,I} : INS \to INS_1$ such that $(g_1, g_{1,I})$ is an **AHLNetI**-morphism and KI is decomposable w.r.t. $K_0 \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

$$\begin{tikzcd} K_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

Proof sketch. By Theorem 4.13 the induced preimage KI_1 of KI and g_1 exists and induces an **AHLNetI**-morphism $(g_1, g_{1,I}) : (K_1, INS_1) \to (K, INS)$ where by Lemma 6.21 the function $g_{1,I}$ is surjective. The surjectivity of $g_{1,I}$ can be used to derive the requirements of the decomposability of KI from the requirements of the mutual decomposability.
For a detailed proof see Detailed Proof C.49 in the appendix. □

Given an AHL-process with instantiations $mp_3 : K_3 \to AN_3$ where AN_3 is the composition of two AHL-nets AN_1 and AN_2. The AHL-process mp_3 can be decomposed into AHL-processes mp_1 and mp_2 of AN_1 and AN_2, respectively. If additionally mp_3 is mutual decomposable with respect to the amalgamation decomposition of the AHL-processes then the decomposition does also lead to a decomposition of AHL-processes with instantiations.

Theorem 8.14 (Amalgamation Decomposition of AHL-Processes with Instantiations)
Given the pushout (PO) below in **AHLNet** over injective AHL-morphisms f_1 and f_2 and $T_{AN_0} = \emptyset$.
Let $KI_3 = (K_3, INIT_3, INS_3)$ be an AHL-occurrence net with instantiations and $mp_3 : K_3 \to AN_3$ be an AHL-process with instantiations.
Furthermore let (mp_1, ψ_1), (mp_2, ψ_2) be projections of mp_3 along g_1 and g_2, respectively. By Theorem 8.5 mp_3 can be recovered as a suitable amalgamation of mp_1 and mp_2 leading to composition of AHL-occurrence nets $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ in the diagram below.
Then mp_3 can also be recovered as a suitable amalgamation of AHL-process with instantiations if KI_3 is mutual decomposable w.r.t. the composition $K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$.

$$\begin{tikzcd}
 & & K_0 \arrow[ddll, "\phi_1"'] \arrow[d, "mp_0"] \arrow[ddrr, "\phi_2"] & & \\
 & \textbf{(PB)} & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \textbf{(PB)} & \\
K_1 \arrow[r, "mp_1"'] \arrow[ddrr, "\psi_1"'] & AN_1 \arrow[dr, "g_1"] & \textbf{(PO)} & AN_2 \arrow[dl, "g_2"'] & K_2 \arrow[l, "mp_2"] \arrow[ddll, "\psi_2"] \\
 & \textbf{(PB)} & AN_3 & \textbf{(PB)} & \\
 & & K_3 \arrow[u, "mp_3"] & &
\end{tikzcd}$$

Proof. Due to the mutual decomposability of KI by Lemma 8.13 the induced preimage $KI_1 = (K_1, INIT_1, INS_1)$ of KI and g_1 exists together with a function $g_{1,I} : INS \to INS_1$ such that $(g_1, g_{1,I})$ is an **AHLNetI**-morphism and KI is decomposable w.r.t. $K_0 \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$.
Then by Theorem 6.17 we can construct the cominimal decomposition $KI_2 = (K_2, INIT_2, INS_2)$ of KI w.r.t. $K_0 \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$, i.e. $KI_3 = KI_1 \circ_{(K_0,\phi_1,\phi_2)} KI_2$. From Theorem 5.25 follows that (KI_1, KI_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2) which means that the agreement projections (mp_0, ϕ_1) and (mp_0, ϕ_2) for mp_1 and mp_2 induced by Theorem 8.5 are agreement projections for AHL-processes with instantiations.
Hence mp_3 is a amalgamation of AHL-processes with instantiations mp_1 and mp_2. □

Remark 8.15. Note that due to the fact that the AHL-occurrence net with instantiations KI_1 is constructed as an induced preimage by Lemma 6.21 there is a surjection $s : INS \to INS_1$ such that (g_1, s) is an **AHLNetI**-morphism. By Theorem 6.22 this implies that also KI_1 is a cominimal decomposition of KI w.r.t. $K_0 \xrightarrow{f_2} K_2 \xrightarrow{g_2} K$. Therefore the construction is symmetric.

In contrast to the amalgamation of AHL-processes without instantiations the amalgamation composition and decomposition of AHL-processes with instantiations do not have a bijective correspondence. The reason is that there may be some instantiations of the amalgamated processes which are not composable with another instantiation of the respective other process. Therefore these instantiations have no influence on the result of the composition. In this case the amalgamation "forgets" some of the instantiations which are not restored by the decomposition. These cases may be avoided by restricting the composition by the requirement that for every instantiation of a given process there is a composable instantiation of the respective other process.

9 Conclusion: Summary of Results and Future Work

In this section we give an overview of the results achieved in this thesis. Moreover we discuss what can be done to support the work with the structures and constructions introduced in the thesis. Finally we outline some future work based on the results like the research of a double pushout approach for high-level processes and how it may be used for the development of process grammars allowing a better handling of the process semantics of high-level petri nets.

9.1 Summary of Results

The first main aim of this thesis has been the definition of a rule-based transformation of AHL-processes according to the rule-based transformation of graphs in the double pushout approach (see [EEPT06]). This has been achieved (see Section 7) by a generalization of the composition of AHL-processes introduced in [EHGP09] (see Sections 3 and 5) and the definition of a suitable decomposition construction (see Section 6). Moreover we transfered the concept of the amalgamation of open net processes in [BCEH01] for open low-level nets to the amalgamation of AHL-processes (see Section 8). We showed that the amalgamation composition and decomposition constructions are inverse to each other leading to a compositional modification of AHL-nets and their processes. This is a very important result since it allows to describe the process semantics of an AHL-net by the process semantics of parts of it and vice versa.
Furthermore we showed that the results of the sequential and parallel composability are special cases of the corresponding results of the composability. The corresponding proofs can be found in Section 5 below the respective definitions and theorems for the generalized version of the composability and composition, respectively. Finally for every of the above constructions we also researched corresponding constructions for AHL-processes with instantiations.

An overview of the main achievements of this thesis can be seen in Figure 33. The overview is structured by the different topics and different types of models. It also roughly shows the dependencies of the results, i.e. a result A depends on a result B if B is used for the proof of A. The dependencies between different rows and different columns, respectively, are marked with arrows where an arrow from A to B means that results in A depend on results in B.
It is clear that the results of the decomposition constructions depend on the corresponding composition constructions since the decomposition is defined as the inverse operation to the composition. The transformation depends on both, the composition and decomposition, because it is a combination of both concepts. Also the results of the amalgamation depend on the results of composition as well as decomposition since there are amalgamated composition and decomposition constructions. The empty cells in the amalgamation row do not have the meaning that there is a lack of theory. Amalgamation means the compositional modification of processes with respect to the modification of their underlying system nets. In the case of AHL-occurrence nets (with instantiations) without a process morphism there is no underlying system net. So the amalgamation of AHL-occurrence nets without process morphisms just does not make any sense.
The results for AHL-processes depend on results for AHL-occurrence nets since the composition or decomposition means a composition or decomposition of the respective AHL-occurrence net. Analogously the results for AHL-occurrence nets with instantiations depend

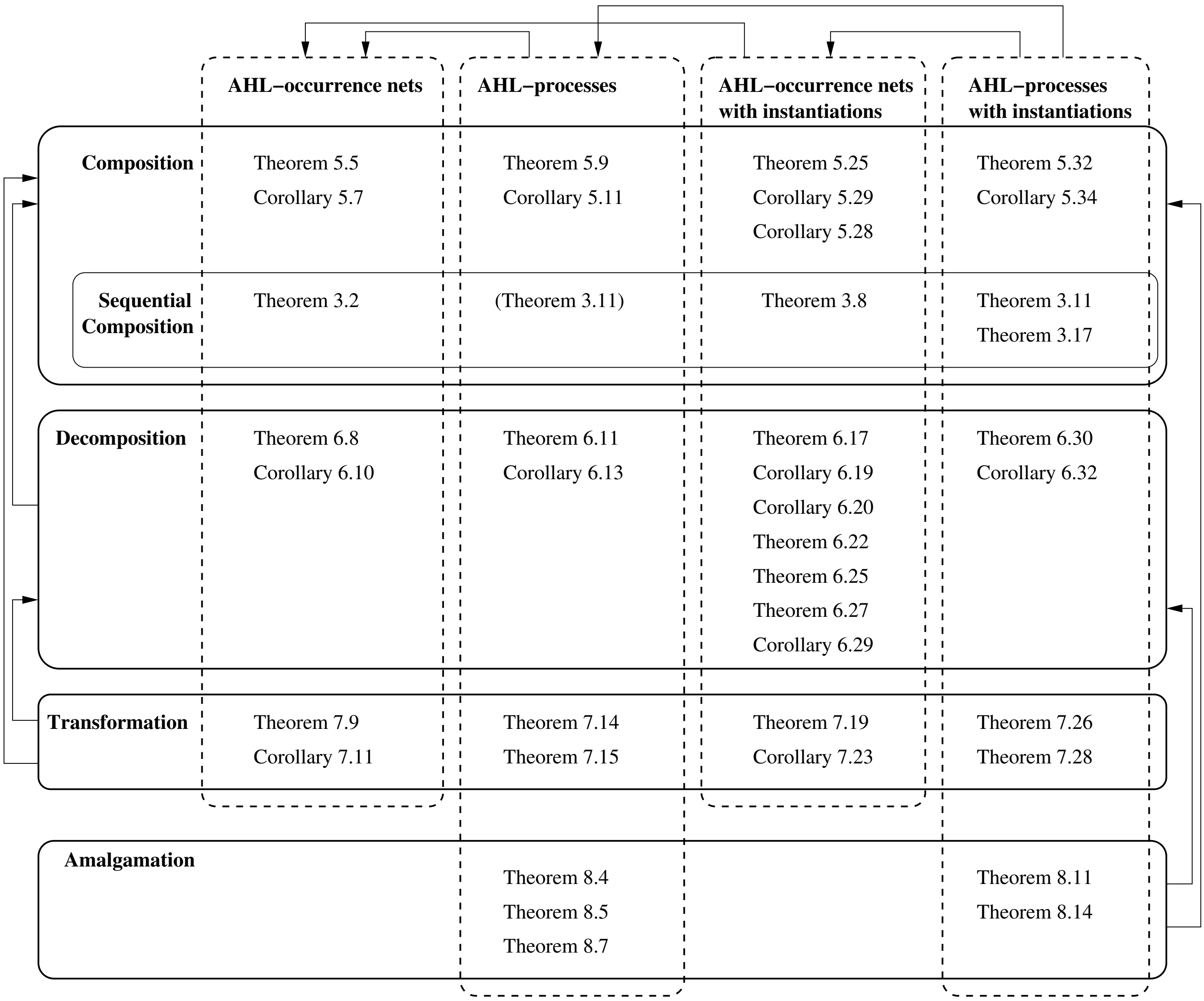


Figure 33: Overview of the main results of this thesis

on the results for AHL-occurrence nets. Since an AHL-process with instantiations is a special AHL-process and the corresponding AHL-occurrence net is an AHL-occurrence net with instantiations, the results for AHL-processes with instantiations depend on the results for AHL-processes as well as on the results of AHL-occurrence nets with instantiations (and of course transitively on the results for AHL-occurrence nets).
Reading the arrows in the inverse direction we obtain a generalization relation, i.e. instead of saying A depends on B we can say B is a special case of A or A is a generalization of B. This emphasizes the fact that the transformation of AHL-processes with instantiations is the most powerful and therefore the most important result of this thesis. All results above and left of the Theorems 7.26 and 7.28 in Figure 33 can be considered as special cases of these results. An AHL-process without instantiations can be considered as an AHL-process with instantiations where the set of instantiations is empty. Also AHL-occurrence nets without a process morphism can be considered as AHL-processes in a given context. This can be done by constructing the pushout over the interface (there is a suitable interface object in the context of each of the concepts mentioned above) and the corresponding morphisms from the interface to the respective AHL-occurrence nets. The pushout results in an AHL-net AN and AHL-morphisms from the different AHL-occurrence nets to AN which can be considered as suitable process morphisms.
A composition can be considered as the application of a non-deleting rule, i.e. a production where the left hand side is the interface. Analogously a decomposition can be considered as the application of a production where the right hand side is the interface.
In a similar way the amalgamation of AHL-processes with instantiations can be seen as a generalization of the concepts above except the transformation. A composition of AHL-processes with instantiations is an amalgamation where all system nets are the same net and the corresponding AHL-morphisms are identities. This does also work for the decomposition of AHL-processes with instantiations. Even the composition and decomposition of AHL-occurrence nets with instantiations can be considered as an amalgamation where the system nets are exactly the composed or decomposed AHL-occurrence nets and the process morphisms are identities.
So also the amalgamation of AHL-processes is a very powerful result. At the moment it remains a future task to unite the two main results of this thesis, i.e. to research an almalgamated transformation of AHL-processes with instantiations providing a compositional transformation of AHL-nets together with their processes.

9.2 Ease of Use

One aspect of the ease of use is the implementation of a tool support. This may happen as an extension of an existing tool which allows the transformation of high-level nets like the *RON Editor* (see [RON09]). The editor supports the modelling of Reconfigurable Object Nets (RONs) which are a simplified version of Algebraic Higher-Order Nets (i.e. a special case of Algebraic High-Level Nets with nets and transformation rules as tokens). At the moment the token nets (also called object nets) are low-level nets but the developers are extending the editor to the support of high-level object nets. An extension which also supports AHL-processes with instantiations and transformations of these processes as tokens of RONs would benefit from the analysis features for nets and rules which are currently implemented in the editor.
For a better performance of the implementation it may be useful to consider special cases

of the transformation of AHL-processes with instantiations with a smaller number of conditions or conditions which can be verified with less effort. An example are non-deleting rules which can be considered as compositions or maybe sequential compositions. In the case of the sequential composition of AHL-processes with instantiations it is sufficient to check whether the AHL-processes are composable without consideration of the instantiations (see Fact 5.24). So the ability to recognize a composition as a sequential composition may result in a higher performance. Similar results may be achieved by the special case of transformations where any "gap" in the flow of the process which is produced by decomposition is then closed by the composition, i.e. transformations which do not produce new input places. So in this case it should be possible to check the applicability of a production of AHL-processes with instantiations without consideration of the instantiations. Probably there are even more interesting special cases which are due to be investigated.
In order to make it more comfortable to work with the categories and constructions presented in this thesis it would be helpful to introduce further constructions which allow to ommit parts which are unnecessary in a given situation. An example is the cofree construction $Inst$ which is already defined in this thesis (see Fact 4.32). The idea of the construction is to consider an AHL-net as an AHL-net with instantiation where the set of instantiations is empty. It allows to regard an AHL-process with instantiations as an **AHLNet**-morphism as well as an **AHLNetI**-morphism.
Other examples are **AHLNetI**-morphisms where the set of instantiations is not empty. Alternatively to the category **AHLNetI** it is possible to define a category of AHL-nets with instantiations where a morphism f only consist of the AHL-morphism f_N but not of the set morphism f_I of the instantiation part. In the alternative category the existence of f_I is a requirement which leads to a category which is isomorphic to **AHLNetI**. This allows to freely switch between represantations with or without the instantiation morphism dependent on the question whether it is needed in a given situation or not similar to the case of the free commutative monoid $\oplus$ where we often write $P^{\oplus}$ instead of $(P^{\oplus}, 0, \oplus)$ because the monoid structure is not needed explicitely.

9.3 Towards a Double Pushout Approach for AHL-Processes with Instantiations

In Section 7 we introduce the direct transformation of AHL-processes with instantiations according to direct double pushout transformations in the algrabraic graph transformation approach (see [EEPT06]). A future task is to consider also sequences of transformations for AHL-processes with instantiations. The analysis of AHL-process transformation systems may lead to interesting and useful results for the analysis of the interaction of processes.
An example is a transfer of the Embedding and Extension Theorem in [EEPT06] from the case of graphs to the case of AHL-processes. Given two AHL-processes $mp_1 : K_1 \rightarrow AN$ and $mp_2 : K_2 \rightarrow AN$ and a $\mathbf{AHLProc}(\mathbf{AN})$-morphism $f : mp_1 \rightarrow mp_2$. The question is under which conditions for a given transformation $mp_1 \overset{t}{\Rightarrow} mp_1'$ there is a corresponding transformation $mp_2 \overset{t^*}{\Rightarrow} mp_2'$ such that there is a $\mathbf{AHLProc}(\mathbf{AN})$-morphism $f' : mp_1' \rightarrow mp_2'$ (see Figure 34) and vice versa.
Other examples are the sequential and parallel independence and the Local Church Rosser Theorem. The transfer of these concepts to the transformation of AHL-processes means the question under which conditions for two productions of AHL-processes p_1 and p_2 the

$$
\begin{tikzcd}
K_1 \arrow[rr, Rightarrow, "t"] \arrow[dd, "f"'] \arrow[dr, "mp_1"] & & K_1' \arrow[dl, "mp_1'"'] \arrow[dd, "f'"] \\
& AN & \\
K_2 \arrow[rr, Rightarrow, "t^*"'] \arrow[ur, "mp_2"'] & & K_2' \arrow[ul, "mp_2'"]
\end{tikzcd}
$$

Figure 34: Extension diagram of AHL-processes

$$
\begin{tikzcd}
& K_0 \arrow[dl, Rightarrow, "{(p_1,m_1)}"'] \arrow[d, "mp_0"] \arrow[dr, Rightarrow, "{(p_2,m_2)}"] & \\
K_1 \arrow[r, "mp_1"] \arrow[dr, Rightarrow, "{(p_2,m_2')}"'] & AN & K_2 \arrow[l, "mp_2"'] \arrow[dl, Rightarrow, "{(p_1,m_1')}"] \\
& K_3 \arrow[u, "mp_3"] &
\end{tikzcd}
$$

Figure 35: Local Church Rosser diagram of processes

commuting diagram in Figure 35 exists.

In Section 3 we presented a notion of independence in the context of the seqential composition of AHL-processes. It would be interesting to transfer this concept to the indepedence of transformations of AHL-processes, i.e. under which conditions two transformations can be applied in different orders such that the results are equivalent to each other (see Figure 36).

The concept of weak adhesive HLR categories (see [EEPT06]) is an abstract categorical framework which is very helpful to show several of the desired results by showing that the considered category is a weak adhesive HLR category. This applies to the categories **PTNet** as well as **AHLNet** but unfortunately the categories **AHLONet**, **AHLONetI**, **AHLProc**(**AN**) and **AHLProcI**(**AN**) are no weak adhesive HLR categories. This can be seen from the fact that a weak adhesive HLR category **C** consists of a set $\mathcal{M}$ of monomorphisms such that the category **C** has pushouts along $\mathcal{M}$-morphisms (i.e. for a given arbitrary span where at least one of the morphisms is in $\mathcal{M}$ the pushout of this span can be constructed in **C**). There exists no suitable set $\mathcal{M}$ for any of the categories above because the question whether the pushout exists in general depends on both of the given span morphisms (see Definition 5.2 and Theorem 5.5). Nevertheless it should be possible to research some interesting topics in the double pushout approach using the results of this thesis.

Additionally it would be interesting to investigate an amalgameted transformation of AHL-processes with instantiations according to the amalgamation composition and decomposition of AHL-processes with instantiations in Section 8.

$$
\begin{tikzcd}
 & K_0 \arrow[dl, Rightarrow, "{(p_1,m_1)}"'] \arrow[d, "mp_0"] \arrow[dr, Rightarrow, "{(p_2,m_2)}"] & \\
K_1 \arrow[r, "mp_1"] \arrow[d, Rightarrow, "{(p_2,m_2')}"'] & AN & K_2 \arrow[l, "mp_2"'] \arrow[d, Rightarrow, "{(p_1,m_1')}"] \\
K_3 \arrow[ur, "mp_3"'] \arrow[rr, equal] & & K_3' \arrow[ul, "mp_3'"]
\end{tikzcd}
$$

Figure 36: Independence of transformations

9.4 Process Grammars

Based on the rule-based transformation of AHL-occurrence nets with instantiations it should be possible to define process grammars for the generation of AHL-processes with instantiations by analogy to graph grammars for the generation of graphs (cf. [EEPT06]).
A graph grammar $GG = (S, R)$ is a start graph S together with a set of graph productions R. The set of graphs derived from the start graph using the production rules in R is called the language $L(GG)$ of the grammar GG.
So analogously a process grammar $PG(AN) = (sp, PR)$ of an AHL-net AN should contain a start process $sp : S \to AN$ and a set PR of productions for AHL-occurrence nets. The set of processes derived from the start process using production rules in PR in a process preserving way is then called the process language $PL(PG(AN))$ of the process grammar $PG(AN)$.
Consider a special process grammar $PSG(AN)$ for a given finite AHL-net AN where the start process is the empty process. The set of productions of $PSG(AN)$ contains rules which allow the creation of copies of single places p of the net AN together with a process mapping to that place p in the net AN. Furthermore for transitions t in the net AN there are rules to create an AHL-occurrence net K_t containing a single transition t' together with an AHL-morphism which maps t' to t. On the left hand side of the productions are the pre and post conditions of t'.
The process grammar $PSG(AN)$ can be used to generate the set of all finite processes of the net AN, i.e. the process language of $PSG(AN)$ is the process semantics of AN which in general is infinite while the process grammar is finite.
In Section B in the appendix we outline this concept for low-level-processes. For this purpose we transfer the concept of the process preserving transformation of AHL-processes to a process preserving transformation of low-level processes.
Low-level nets can be considered as special AHL-nets where the specification contains one single sort and no operations and the algebra is the final algebra $(\{\bullet\}, \emptyset)$. Analogously low-level-occurrence nets can be considered as special AHL-occurrence nets because the skeleton of an AHL-occurrence net is a low-level-occurrence net.
So our results can be transferred to low-level processes. We do that in Section B in the appendix in an intuitive way and since it is just an outline we do not explicitely proof the correctness of that transfer.

The outline demonstrates that it is possible to generate the finite process semantics of a given finite low-level net. One of the future tasks is to transfer the concepts in Appendix B to AHL-processes which in the case of a fixed specification and algebra should be very similar.
For a practical transfer to AHL-processes with instantiations some more ideas may be needed since for a given atomic AHL-occurrence net it is possible that there is an infinite set of corresponding instantiations and therefore a process semantics generating grammar for AHL-occurrence nets with instantiations will in general have an infinite set of productions.
A solution for this problem may be achieved by generalizing the results in this thesis to AHL-nets where the algebra (and maybe also the specification) is not fixed. Choosing the term algebra with variables or in the case of specifications with equations a representative algebra we obtain “abstract instantiations” where the data elements are terms. This would allow the definition of schemes for instantiations of the right hand side of productions representing a possibly infinite set of instantiations. Under certain conditions the abstract instantiations then could be “instantiated” to concrete instantiations by a given match and the instantiations of the process on which the production is applied. It should then be possible to denote a finite set of productions with abstract instantiations for the generation of all finite AHL-processes with instantiations for a given finite AHL-net.

A Example

In this section we define the specification *SP-Alarm* and the *SP-Alarm*-algebra A that are used in the AHL-nets which serve as examples in the different sections of this thesis.

Definition A.1 (Specification *SP-Alarm*)
The specification *SP-Alarm* is defined in the following way:

$\underline{\underline{SP\text{-}Alarm}} =$

$\underline{sorts}$:
$phase$
$temp$
$onoff$
$volume$
$radio$
$volSched$

$\underline{opns}$:
$initial : \to phase$
$next : phase \to phase$
$add : phase\ phase \to phase$
$warmTemp :\to temp$
$on : \to onoff$
$off\ : \to onoff$
$switch : onoff\ \to onoff$
$mute : \to volume$
$inc : volume\ volume \to volume$
$music : onoff\ \ volume \to radio$
$vol : radio \to volume$
$sched : phase\ volume \to volume$

$\underline{vars}$: $\emptyset$

$\underline{eqns}$: $\emptyset$

$\triangle$

Definition A.2 (*SP-Alarm*-Algebra A)
We define an *SP-Alarm*-algebra A in the following way:

$\underline{\underline{A}} =$
$A_{phase} = \mathbb{N}$
$A_{temp} = \{0, \ldots, 40\}$
$A_{onoff} = \{0, 1\}$
$A_{volume} = \{0, \ldots, 100\}$
$A_{radio} = A_{onoff} \times A_{volume}$
$A_{volSched} = A_{phase} \times A_{volume}$

$initial_A \in A_{phase}$

$$initial_A = 0$$

$next_A : A_{phase} \to A_{phase}$

$$p \mapsto p + 1$$

$add_A : A_{phase} \times A_{phase} \to A_{phase}$

$$(p_1, p_2) \mapsto p_1 + p_2$$

$warmTemp_A \in A_{temp}$

$$warmTemp_A = 25$$

$on_A \in A_{onoff}$

$$on_A = 1$$

$off_A \in A_{onoff}$

$$off_A = 0$$

$switch_A : A_{onoff} \to A_{onoff}$

$$x \mapsto (x + 1)\ mod\ 2$$

$mute_A \in A_{volume}$

$$mute_A = 0$$

$inc_A : A_{volume} \times A_{volume} \to A_{volume}$

$$(v_1, v_2) \mapsto max\{v_1 + v_2, 40\}$$

$music_A : A_{onoff} \times A_{volume} \to A_{radio}$

$$(o, v) \mapsto (o, v)$$

$vol_A : A_{radio} \to A_{volume}$

$$(o, v) \mapsto v$$

$sched_A : A_{phase} \times A_{volume} \to A_{volSched}$

$$(p, v) \mapsto (p, v)$$

$\triangle$

B Process Grammars for Low-Level Petri Nets

In this section we sketch how the transformation of processes can be used to generate the finite process semantics of a finite low-level net.

First we define the notion of atomic processes which is either the empty process or an occurrence net containing one single place or one single transition with its pre and post domain.

Definition B.1 (Empty Net)
We define $N_\emptyset = (P_\emptyset, T_\emptyset, pre_\emptyset, post_\emptyset)$ as the empty net with

$$\begin{array}{rcll} P_\emptyset & = & \emptyset & \\ T_\emptyset & = & \emptyset & \\ pre_\emptyset & = & \emptyset & \text{(the empty function)} \\ post_\emptyset & = & \emptyset & \text{(the empty function)} \end{array}$$

△

Definition B.2 (Atomic Place Process Net)
Given a P/T-net $N = (P, T, pre, post)$ and a place $p \in P$.
Then the P/T-net $N_p = (P_p, T_p, pre_p, post_p)$ is defined by

$$\begin{array}{rcll} P_p & = & \{p\} & \\ T_p & = & \emptyset & \\ pre_p & = & \emptyset & \text{(the empty function)} \\ post_p & = & \emptyset & \text{(the empty function)} \end{array}$$

△

Definition B.3 (Atomic Transition Process Net)
Given a P/T-net $N = (P, T, pre, post)$ and a transition $t \in T$.
Let

$$pre(t) = \sum_{p \in P} \lambda_p p$$

and

$$post(t) = \sum_{p \in P} \mu_p p$$

Then the P/T-net $N_t = (P_t, T_t, pre_t, post_t)$ is defined by

$$\begin{array}{rcl} P_t & = & \{p_i \mid p \in P, 1 \leq i \leq \lambda_p\} \cup \\ & & \{p'_i \mid p \in P, 1 \leq i \leq \mu_p\} \\ T_t & = & \{t\} \\ pre_t(t) & = & \sum_{p \in P} \sum_{i=1}^{\lambda_p} p_i \\ post_t(t) & = & \sum_{p \in P} \sum_{i=1}^{\mu_p} p'_i \end{array}$$

$\triangle$

Definition B.4 (Atomic Process Nets)
Given a P/T-net $N = (P, T, pre, post)$.
The set of atomic place process nets of N is defined as

$$AtomPN_P(N) = \{N_p \mid p \in P\}$$

The set of atomic transition process nets of N is defined as

$$AtomPN_T(N) = \{N_t \mid t \in T\}$$

The set of atomic process nets of N is defined as

$$AtomPN(N) = \{N_\emptyset\} \cup AtomPN_P(N) \cup AtomPN_T(N)$$

$\triangle$

Definition B.5 (Atomic Place Process)
Given a P/T-net $N = (P, T, pre, post)$ and a place $p \in P$.
Then the atomic process morphism $ap : N_p \to N$ is defined by

$$ap = (ap_P, ap_T)$$

$$ap_P(p) = p$$

$$ap_T = \emptyset$$

$\triangle$

Well-definedness. The well-definedness of the P/T-net morphism ap follows from the fact that $p \in P$ and $T_p = \emptyset$. $\square$

Definition B.6 (Atomic Transition Process)
Given a P/T-net $N = (P, T, pre, post)$ and a transition $t \in T$.
Then the atomic process morphism $ap : N_t \to N$ is defined by

$$ap = (ap_P, ap_T)$$

$$ap_P(p_i) = p$$

$$ap_P(p'_i) = p$$

$$ap_T(t) = t$$

$\triangle$

Well-definedness. See Detailed Proof C.50 in the appendix. $\square$

Definition B.7 (Atomic Processes)
Given a P/T-net $N = (P, T, pre, post)$.
The set of atomic place processes of N is defined as

$$AtomProcs_P(N) = \{ap : N_p \to N \mid N_p \in AtomPN_P(N)\}$$

The set of atomic transition processes of N is defined as

$$AtomProcs_T(N) = \{ap : N_t \to N \mid N_t \in AtomPN_T(N)\}$$

The set of atomic processes of N is defined as

$$AtomProcs(N) = \{\emptyset : N_\emptyset \to N\} \cup AtomProcs_P(N) \cup AtomProcs_T(N)$$

$\emptyset : N_\emptyset \to N$ is the unique morphism given by initiality of $N_\emptyset$ in the category **PTNet**.
The set of non-empty atomic Processes is called $AtomProcs^+(N)$, i.e.

$$\begin{aligned} AtomProcs^+(N) &= AtomProcs(N) \setminus \{\emptyset : N_\emptyset \to N\} \\ &= AtomProcs_P(N) \cup AtomProcs_T(N) \end{aligned}$$

Remark B.8 *(Corresponding atomic processes)* . Due to the definition of atomic processes there is a one-to-one correspondence between the atomic process nets and the atomic processes of N, i.e. there is a bijection

$$proc : AtomProcNets(N) \xrightarrow{\sim} AtomProcs(N)$$

with

$$proc(N_\emptyset) = \emptyset : N_\emptyset \to N$$

and

$$proc(K) = ap : K \to N$$

for $K \neq N_\emptyset$.

$\triangle$

For the rule-based generation of processes we need interfaces for the productions of occurrence nets. For every type of atomic process we define a corresponding interface.

Definition B.9 (Atomic Interfaces)
Given a P/T-net $N = (P, T, pre, post)$.
The interface for the empty process net is defined as

$$AtomInterface(N, N_\emptyset) = N_\emptyset$$

Let $p \in P$. The interface for the atomic place process net N_p of N is defined as

$$AtomInterface(N, N_p) = N_\emptyset$$

Let $t \in T$. The interface for the atomic transition process net N_t of N is defined as

$$AtomInterface(N, N_t) = I$$

where $I = (P_I, T_I, pre_I, post_I)$ with

$$P_I = P_t$$
$$T_I = \emptyset$$
$$pre_I = post_I = \emptyset$$

Remark B.10 . Note that atomic interfaces are not necesserily atomic process nets since for $t \in T$ with more than one pre or post arc there is more than one place in N_t and hence also in $AtomInterface(N, N_t)$.
However, for every atomic process net K of N there is $I = AtomInterface(N, K)$ a subnet of K which implies an inclusion $f : I \hookrightarrow K$.
Moreover the net I has no transitions, i.e. $T_I = \emptyset$.

△

Definition B.11 ((Finite) Process Semantics)
The (finite) process semantics of a P/T-net $N = (P, T, pre, post)$ is given by the set of all (finite) processes to the net:

$$Procs(N) = \{mp : K \to N \mid K \text{ is an occurrence net and } mp \text{ a P/T-net morphism } \}$$

$$FinProcs(N) = \{mp : K \to N \in Procs(N) \mid K \textit{ is finite}\}$$

A net is called finite if the sets of places and transitions are finite. △

Fact B.12 (Atomic Processes are Finite Processes)
Given a P/T-net $N = (P, T, pre, post)$. The atomic processes of N are finite processes, i.e.

$$AtomProcs(N) \subseteq FinProcs(N)$$

Proof sketch. The empty process and the atomic places processes are obviously process because the corresponding nets contain no transitions.
An atomic transition process contains only one single transition which due to its definition satisfies the requirements for occurrence nets.
For a detailed proof see Detailed Proof C.51 in the appendix. □

Using the atomic interfaces every finite process of a finite net N can be generated by composition of atomic processes.

Theorem B.13 (Generation of Finite Process Semantics by Composition)
Given a finite P/T-net $N = (P, T, pre, post)$. The finite process semantics of N can be generated by iterated composition of atomic processes.
Let the set of generated processes be defined as

$$\begin{aligned} GenProcs(N) \quad = \quad & \{\emptyset : N_\emptyset \to N\} \cup \\ & \{mp_1 \circ_{(I,f_1,f_2)} mp_2 \mid mp_1 : K_1 \to N \in AtomProcs^+(N), \\ & \quad mp_2 : K_2 \to N \in GenProcs(N), \\ & \quad I = AtomInterface(N, K_1), \\ & \quad f_1 : I \hookrightarrow K_1 \text{ inclusion } \} \end{aligned}$$

This generation of processes leads to the set of all processes of N:

$$GenProcs(N) = FinProcs(N)$$

Proof sketch.

"$\subseteq$":
: Follows from the fact that the composition of processes again leads to processes. The composition of the finite processes is finte because it is constructed by a gluing of two finite processes.

"$\supseteq$":
: Since composition and decomposition are inverse to each other it is sufficient to show that every process can be step-wise decomposed into atomic processes. If the process is not empty and hence the decomposition not yet terminated in every step we can distinguish the case that the net has a transition or not. If it has a transition then there is an atomic transition process which can be used for decomposition. If it has no transition the there is an atomic place process which can be used for decomposition. This leads to smaller processes in every step and hence to the empty process.

For a detailed proof see Detailed Proof C.52 in the appendix. □

Definition B.14 (Process Grammar)
Given a P/T-net $N = (P, T, pre, post)$.
A process grammar of N is defined as

$$PG = (sp, PR)$$

where

$$sp \in Procs(N)$$

is called start process and

$$\begin{aligned} PR \quad = \quad & \{(p, rp) \mid p : L \xleftarrow{l} I \xrightarrow{r} R \text{ is a production for occurrence nets,} \\ & rp : R \to N \text{ is a process morphism}\} \end{aligned}$$

is called set of process rules of the process grammar. △

Definition B.15 (Process Language)
Given a P/T-net $N = (P, T, pre, post)$ and a process grammar $PG = (sp, PR)$ of N. Then the process language generated by PG is defined as

$$PL(PG) = \{mp \mid sp \overset{*}{\Rightarrow}_{PR} mp \text{ is a process preserving transformation of processes}\}$$

△

Fact B.16 (Process Language is Set of Processes)
Given a P/T-net $N = (P, T, pre, post)$ and a process grammar $PG = (sp, PR)$ of N.
The process language generated by PG is a set of processes of N, i.e.

$$PL(PG) \subseteq Procs(N)$$

Proof. Follows directly from the fact that sp is a process of N and the process preserving transformation of processes leads to processes of N. □

Definition B.17 (Process Semantics Generating Grammar)
Given a P/T-net $N = (P, T, pre, post)$.
The Process Semantics Generating Grammar $PSG(N)$ of N is defined as

$$PSG(N) = (sp, PR)$$

where

$$sp = \emptyset : N_\emptyset \to N \in AtomProcs(N)$$

and

$$\begin{aligned} PR \quad = \quad & \{(I \xleftarrow{id_I} I \xrightarrow{r} K, ap) \mid ap : K \to N \in AtomProcs^+(N), \\ & I = AtomInterface(N, K),\ r \text{ is inclusion}\} \end{aligned}$$

△

Fact B.18 (Process Semantics Generating Grammar is Process Grammar)
Given a P/T-net $N = (P, T, pre, post)$.
The Process Semantics Generating Grammar $PSG(N)$ of N is a process grammar of N.

Proof. The start process sp of $PSG(N)$ is an atomic process of N which by Fact B.12 implies that sp is a process of N. Moreover for similar reasons $ap : K \to N$ is a process of N.
It remains to show that for $ap : K \to N \in AtomProcs(N)$, $I = AtomInterface(N, K)$ and inclusion $r : I \to K$ there is $I \xleftarrow{id_I} I \xrightarrow{r} K$ a process preserving production. This follows from the fact that $T_I = \emptyset$ and id_I and r are injective morphisms. □

Theorem B.19 (Generation of Process Semantics by Grammar)
Given a finite P/T-net $N = (P, T, pre, post)$.
The process language generated by the Process Semantics Generating Grammar $PSG(N)$ is the process semantics of N, i.e.

$$PL(PSG(N)) = FinProcs(N)$$

Proof sketch.

"$\subseteq$":
Follows from the fact that the process preserving transformation of processes again leads to processes. The result of the transformation remains finite because the right hand sides of the productions are finite.

"$\supseteq$":
Since the rules are non-deleting the context net of a transformation is identic to the net which is transformed and hence also an occurrence net. So the require decomposition exists. From Theorem B.13 follows that every process can be generated by the step-wise composition of atomic processes implying the existence of the required composition for a transformation.

For a detailed proof see Detailed Proof C.53 in the appendix. □

C Detailed Proofs

This section contains the detailed proofs of the facts, theorems and corollaries in the thesis.

C.2 Algebraic High-Level Processes

Detailed Proof C.1 (AHL-Morphisms Reflect AHL-Occurrence Nets)
See Fact 2.14.

Proof. In order to show that K_1 is an AHL-occurrence net we have to show that it is unary, there are no forward or backward conflicts and the causal relation $<_{K_1}$ is a finitary strict partial order.

K_1 *is unary*:
Let us assume that K_1 is not unary, i.e. there are $p \in P_{K_1}$, $t \in T_{K_1}$ with

$$(term_1, p) \oplus (term_2, p) \leq pre_{K_1}(t)$$

or

$$(term_1, p) \oplus (term_2, p) \leq post_{K_1}(t)$$

Let $(term_1, p) \oplus (term_2, p) \leq pre_{K_1}(t)$.
For AHL-morphism f there is

$$(id_{T_{OP}(X)} \otimes f_P)^{\oplus} \circ pre_{K_1}(t) = pre_{K_2}(f_T(t))$$

and hence

$$\begin{array}{rl} & (term_1, f_P(p)) \oplus (term_2, f_P(p)) \\ = & (id_{T_{OP}(X)} \otimes f_P)^{\oplus}((term_1, p) \oplus (term_2, p)) \\ \leq & pre_{K_2}(f_T(t)) \end{array}$$

This implies that K_2 is not unary, contradicting the fact that K_2 is an AHL-occurrence net.
The case that $(term_1, p) \oplus (term_2, p) \leq post_{K_1}(t)$ works analogously. Hence K_1 is unary.

K_1 *has no forward conflict*:
Let us assume that K_1 has a forward conflict, i.e. there is $p \in P_{K_1}, t_1 \neq t_2 \in T_{K_1}$ with

$$p \in \bullet t_1 \cap \bullet t_2$$

This means that there are $term_1, term_2 \in T_{OP}(X)_{type(p)}$ such that

$$(term_1, p) \leq pre_{K_1}(t_1)$$

and

$$(term_2, p) \leq pre_{K_1}(t_2)$$

Since AHL-morphisms preserve pre and post conditions we obtain

$$\begin{array}{rcl} (term_1, f_P(p)) & = & (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(term_1, p) \\ & \leq & pre_{K_2}(f_T(t_1)) \end{array}$$

and

$$\begin{aligned}(term_2, f_P(p)) &= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(term_2, p) \\ &\leq pre_{K_2}(f_T(t_2))\end{aligned}$$

This means that K_2 has a forward conflict, contradicting the fact that K_2 is an AHL-occurrence net.
Hence K_1 has no forward conflict.

K_1 has no backward conflict:
The proof for this case works analogously to the one for forward conflicts because AHL-morphisms preserve post as well as pre conditions and K_2 has no backward conflicts.

$<_{K_1}$ is a finitary strict partial order:
We have to show that $<_{K_1}$ is finitary and irreflexive.

$<_{K_1}$ is finitary:
Let us assume that $<_{K_1}$ is not finitary. Then there is an element $x \in P_{K_1} \uplus T_{K_1}$ with an infinite number of predecessors. Let

$$S = \{y \in P_{K_1} \uplus T_{K_1} \mid y <_{K_1} x\}$$

be the infinite set of predecessors of x.
Since AHL-morphisms preserve pre and post conditions, $a <_{K_1} b$ implies $f(a) <_{K_2} f(b)$.
This means that there is an infinite set

$$S' = \{f(y) \mid y \in S\}$$

where for every $f(y) \in S'$ there is

$$f(y) <_{K_2} f(x)$$

This means that $f(x)$ has an infinite number of predecessors implying that $<_{K_2}$ is not finitary. This contradicts the fact that K_2 is an AHL-occurrence net and hence $<_{K_1}$ is finitary.

$<_{K_1}$ is irreflexive:
Let us assume that $<_{K_1}$ is not irreflexive, i.e. there exists a cycle $x <_{K_1} x$. This implies $f(x) <_{K_2} f(x)$ contradicting the fact that $<_{K_2}$ is irreflexive and hence $<_{K_1}$ is irreflexive.

□

Detailed Proof C.2 (*Flat* is Functor)
See Theorem 2.26.

Proof.

1. $Flat(AN)$ is well-defined:
We have to show that $pre_A, post_A \in CP^{\oplus}$. This follows for pre_A from the fact that $term_i$ of $type(p_i)$ implies $v^{\sharp}(term_i) \in A_{type(p_i)}$ and similar for $post_A$.

2. $Flat(f)$ is well-defined:

   (a) Let $(a,p) \in CP_1$.
   By definition we have $Flat(f)_P(a,p) = (id_A \otimes f_P)(a,p) = (id_A(a), f_P(p)) = (a, f_P(p)) \in CP_2$ because $f_P(p) \in P_2$ and $a \in A_{type_1(p)} = A_{type_2(f_P(p))}$.

   (b) $(t,v) \in CT_1 \Rightarrow f_C(t,v) \in CT_2$
   $(t,v) \in CT_1$ means $v : Var(t) \to A$ s.t. $cond_1(t)$ valid in A under v,
   $f_C(t,v) = (f_T(t),v) \in CT_2$ means $v : Var(f_T(t)) \to A$ s.t. $cond_2(f_T(t)) = cond_1(t)$ valid in A under v. This follows from $(t,v) \in CT_1$ because

$$\begin{aligned} Var(t) &= Var(cond_1(t)) \cup Var(pre_1(t)) \cup Var(post_1(t)) \\ &= Var(cond_2(f_T(t))) \cup Var(pre_2(f_T(t))) \cup Var(post_2(f_T(t))) \\ &= Var(f_T(t)) \end{aligned}$$

3. $Flat(f)$ is P/T-net morphism:
   For symmetry reasons it suffices to show commutativity for pre_A of

$$\begin{tikzcd} CT_1 \arrow[r, "pre_{1,A}"] \arrow[d, "f_C"'] \arrow[dr, phantom, "(\mathbf{1})"] & CP_1^{\oplus} \arrow[d, "(id_A \otimes f_P)^{\oplus}"] \\ CT_2 \arrow[r, "pre_{2,A}"'] & CP_2^{\oplus} \end{tikzcd}$$

   Given $(t_1, v_1) \in CT_1$ with $pre_1(t_1) = \sum_{i=1}^{n}(term_i, p_i)$ we have
   $pre_2(f_T(t_1)) = (id_{T_\Sigma(X)} \otimes f_P)^{\oplus}(pre_1(t_1)) = (id_{T_\Sigma(X)} \otimes f_P)^{\oplus}(\sum_{i=1}^{n}(term_i, p_i))$
   $= \sum_{i=1}^{n}(term_i, f_P(p_i))$
   because f is an AHL-morphisms and hence
   $(id_A \otimes f_P)^{\oplus}(pre_{1A}(t_1, v_1)) = (id_A \otimes f_P)^{\oplus}(\sum_{i=1}^{n}(v_1^{\sharp}(term_i), p_i))$
   $= \sum_{i=1}^{n}(v_1^{\sharp}(term_i), f_P(p_i)) = pre_{2A}(f_T(t_1), v_1) = pre_{2A}(f_C(t_1, v_1))$.

4. Obviously we have $Flat(f) = id_{Flat(AN)}$ for $f = id_{AN}$. Furthermore we have for the composition

$$\begin{aligned} Flat(g \circ f) &= ((id_A \otimes (g \circ f)_P)^{\oplus}, (g \circ f)_C) \\ &= ((id_A \otimes (g \circ f)_P)^{\oplus}, g_C \circ f_C) \\ &= ((id_A \circ id_A \otimes (g \circ f)_P)^{\oplus}, g_C \circ f_C) \\ &= ((id_A \otimes g_P)^{\oplus} \circ (id_A \otimes f_P)^{\oplus}, g_C \circ f_C) \\ &= Flat(g) \circ Flat(f) \end{aligned}$$

□

Detailed Proof C.3 (*Flat* Preserves Pushouts)
See Theorem 2.27.

Proof.

$$
\begin{tikzcd}
AN_0 \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & AN_1 \arrow[d, "g_1"] \\
AN_2 \arrow[r, "g_2"'] & AN_3
\end{tikzcd}
\qquad
\begin{tikzcd}
Flat(AN_0) \arrow[r, "Flat(f_1)"] \arrow[d, "Flat(f_2)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(AN_1) \arrow[d, "Flat(g_1)"] \\
Flat(AN_2) \arrow[r, "Flat(g_2)"'] & Flat(AN_3)
\end{tikzcd}
$$

Pushout (1) in **AHLNet** is constructed componentwise by the Pushouts (3) and (4) in **SET**.

$$
\begin{tikzcd}
T_0 \arrow[r, "f_{1,T}"] \arrow[d, "f_{2,T}"'] \arrow[dr, phantom, "(\mathbf{3})"] & T_1 \arrow[d, "g_{1,T}"] \\
T_2 \arrow[r, "g_{2,T}"'] & T_3
\end{tikzcd}
\qquad
\begin{tikzcd}
P_0 \arrow[r, "f_{1,P}"] \arrow[d, "f_{2,P}"'] \arrow[dr, phantom, "(\mathbf{4})"] & P_1 \arrow[d, "g_{1,P}"] \\
P_2 \arrow[r, "g_{2,P}"'] & P_3
\end{tikzcd}
$$

Since Pushouts in **PTNet** are also constructed componentwise in **SET** it suffices to show that (5) and (6) are Pushouts in **SET**.

$$
\begin{tikzcd}
CT_0 \arrow[r, "f_{1,C}"] \arrow[d, "f_{2,C}"'] \arrow[dr, phantom, "(\mathbf{5})"] & CT_1 \arrow[d, "g_{1,C}"] \\
CT_2 \arrow[r, "g_{2,C}"'] & CT_3
\end{tikzcd}
\qquad
\begin{tikzcd}
A \otimes P_0 \arrow[r, "id_A \otimes f_{1,P}"] \arrow[d, "id_A \otimes f_{2,P}"'] \arrow[dr, phantom, "(\mathbf{6})"] & A \otimes P_1 \arrow[d, "id_A \otimes g_{1,P}"'] \arrow[ddr, bend left, "h_1"] & \\
A \otimes P_2 \arrow[r, "id_A \otimes g_{2,P}"] \arrow[drr, bend right, "h_2"'] & A \otimes P_3 \arrow[dr, dotted, "h"] \arrow[d, phantom, "(\mathbf{8})"] \arrow[r, phantom, "(\mathbf{7})"] & \\
 & & X
\end{tikzcd}
$$

We show the universal properties for (6), because we cannot directly use that the cartesian product $A \times \_$ preserves Pushouts, since $A \otimes P \subsetneq A \times P$. Given h_1, h_2 with $h_1 \circ (id_A \otimes f_{1P}) = h_2 \circ (id_A \otimes f_{2P})$ we define $h : A \otimes P_3 \to X$ by

$$
h(a, p_3) = \begin{cases} h_1(a, p_1) & \text{for } p_3 = g_{1P}(p_1) \text{ with } p_1 \in P_1 \\ h_2(a, p_2) & \text{for } p_3 = g_{2P}(p_2) \text{ with } p_2 \in P_2 \end{cases}
$$

It suffices to show that h is well-defined, i.e. $p_3 = g_{1P}(p_1) = g_{2P}(p_2)$ implies $h_1(a, p_1) = h_2(a, p_2)$, because with this definition (7) and (8) commute by construction and h is unique with this property.
Given $p_3 = g_{1P}(p_1) = g_{2P}(p_2)$ we have by definition of Pushout (4) in **SET** a sequence $p_{01}, \ldots, p_{0n} \in P_0$ with

$$
\begin{tikzcd}[column sep=small]
 & p_{01} \arrow[dl, mapsto, "f_{1,P}"'] \arrow[dr, mapsto, "f_{2,P}"] & & p_{02} \arrow[dl, mapsto, "f_{2,P}"'] \arrow[dr, mapsto, "f_{1,P}"] & & p_{03} \arrow[dl, mapsto, "f_{1,P}"'] & \cdots & p_{0n} \arrow[dr, mapsto, "f_{2,P}"] & \\
p_1 & & p_{21} & & p_{11} & & \cdots & & p_2
\end{tikzcd}
$$

This implies

$$
\begin{aligned}
h_1(a, p_1) &= h_1(a, f_{1P}(p_{01})) \\
&= h_1 \circ (id_A \otimes f_{1P})(a, p_{01}) \\
&= h_2 \circ (id_A \otimes f_{2P})(a, p_{01}) \\
&= h_2(a, f_{2P}(p_{01})) \\
&= h_2(a, p_{21}) \\
&= h_2(a, f_{2P}(p_{02})) \\
&= h_2 \circ (id_A \otimes f_{2P})(a, p_{02}) \\
&= h_1 \circ (id_A \otimes f_{1P})(a, p_{02}) \\
&= \dots \\
&= h_1 \circ (id_A \otimes f_{1P})(a, p_{03}) \\
&= \dots \\
&= h_2 \circ (id_A \otimes f_{2P})(a, p_{0n}) \\
&= h_2(a, f_{2P}(p_{0n})) \\
&= h_2(a, p_2)
\end{aligned}
$$

All steps are well-defined, because $(a, p_{01}), \dots, (a, p_{0n}) \in A \otimes P_0$:
In fact $(a, p_3) \in A \otimes P_3$ implies $a \in A_{type_3(p_3)} = A_{type_1(p_1)} = A_{type_0(p_{0i})}$ for $i = 1, \dots, n$ using type compability of g_{1P} for $p_3 = g_{1P}(p_1)$ and of f_{1P}, f_{2P}, g_{2P} for the other cases (see proof of Theorem 2.26).

For similar reasons we can show explicitely the universal properties of (5) using Pushout (3). Given $k_1 : CT_1 \to X, k_2 : CT_2 \to X$ with $k_1 \circ f_{1C} = k_2 \circ f_{2C}$ there is a unique $k : CT_3 \to X$ defined by

$$
k(t_3, v) = \begin{cases} k_1(t_1, v) & \text{for } t_3 = g_{1T}(t_1) \text{ with } t_1 \in T_1 \\ k_2(t_2, v) & \text{for } t_3 = g_{2T}(t_2) \text{ with } t_2 \in T_2 \end{cases}
$$

Similar to above we can show that $t_3 = g_{1T}(t_1) = g_{2T}(t_2)$ implies $k_1(t_1, v) = k_2(t_2, v)$ where all steps are well-defined using Theorem 2.26. □

Detailed Proof C.4 (*proj* Induces Pullback)
See Theorem 2.33.

Proof.

$$
\begin{tikzcd}
X \arrow[dr, "g"] \arrow[drr, bend left, "g_2"] \arrow[ddr, bend right, "g_1"'] \arrow[drr, phantom, "\mathbf{(3)}"] & & \\
\mathbf{(2)} & Flat(K_1) \arrow[r, "Flat(f)"] \arrow[d, "proj(K_1)"'] \arrow[dr, phantom, "\mathbf{(1)}"] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
& Skel(K_1) \arrow[r, "Skel(f)"'] & Skel(K_2)
\end{tikzcd}
$$

Since *proj* is a natural transformation, diagram (1) commutes.
Let X be a P/T-net and $g_1 : X \to Skel(K_1)$, $g_2 : X \to Flat(K_2)$ P/T-net morphisms with $Skel(f) \circ g_1 = proj(K_2) \circ g_2$.
We have to show that $Flat(K_1)$ satisfies the universal property, that there is a unique morphism $g : X \to Flat(K_1)$ s.t. (2) and (3) commute.

existence of $g : X \to Flat(K_1)$:

We define $g : X \to Flat(K_1)$ with

$$g_P(p) = (\pi_1(g_{2,P}(p)), g_{1,P}(p))$$

and

$$g_T(t) = (g_{1,T}(t), \pi_2(g_{2,T}(t)))$$

where π_1, π_2 are projections with $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$.

well-definedness of g:

well-definedness of g_P:

$g_{2,P}(x) \in P_{Flat(K_2)}$ implies $p \in P_{K_2}$ and $a \in A_{type_{K_2}}(p)$ with $(a, p) = g_{2,P}(x)$.
Then we have

$$\begin{aligned} p &= proj(K_2)_P(a, p) \\ &= proj(K_2)_P(g_{2,P}(x)) \\ &= Skel(f)_P(g_{1,P}(x)) \end{aligned}$$

Since AHL-morphisms preserve types and $g_{1,P}(x) \in P_{Skel(K_1)}$ implies $g_{1,P}(x) \in P_{K_1}$ we have

$$\begin{aligned} type_{K_1}(g_{1,P}(x)) &= type_{K_2} \circ f_P(g_{1,P}(x)) \\ &= type_{K_2}(Skel(f)_P(g_{1,P}(x))) \\ &= type_{K_2}(p) \end{aligned}$$

and hence $a \in A_{type_{K_1}(g_{1,P}(x))}$ which implies $(a, g_{1,P}(x)) \in P_{Flat(K_1)}$.
So g_P is well-defined.

well-definedness of g_T:

$g_{2,T}(x) \in T_{Skel(K_2)}$ implies $(t, v) \in CT_{K_2}$ with $(t, v) = g_{2,T}(x)$. Then we have

$$\begin{aligned} t &= proj(K_2)_T(t, v) \\ &= proj(K_2)_T(g_{2,T}(x)) \\ &= Skel(f)_T(g_{1,T}(x)) \end{aligned}$$

Since AHL-morphisms preserve conditions as well as pre and post conditions and $g_{1,T}(x) \in T_{Skel(K_1)}$ implies $g_{1,T}(x) \in T_{K_1}$, we have

$$\begin{aligned} cond_{K_1}(g_{1,T}(x)) &= cond_{K_2} \circ f_T(g_{1,T}(x)) \\ &= cond_{K_2}(Skel(f)_T(g_{1,T}(x))) \\ &= cond_{K_2}(t) \end{aligned}$$

and analogously for *pre* and *post*, i.e. we have $Var_{K_1}(g_{1,T}(x)) = Var_{K_2}(t)$ and hence $(g_{1,T}(x), v) \in CT_{K_1}$ which implies $(g_{1,T}(x), v) \in T_{Flat(K_1)}$.
So also g_T is well-defined.

g is P/T-net morphism:
For the well-definedness of g it remains to show that g preserves pre and post conditions.
Let

$$pre_X(t) = \sum_{i=1}^{n} p_i$$

Then we have

$$\begin{aligned} g_P^{\oplus}(pre_X(t)) &= g_P^{\oplus}(\sum_{i=1}^{n} p_i) \\ &= \sum_{i=1}^{n} g_P(p_i) \\ &= \sum_{i=1}^{n} (\pi_1(g_{2,P}(p_i), g_{1,P}(p_i))) \end{aligned}$$

On the other hand we have

$$pre_{Flat(K_1)}(g_T(t)) = pre_{Flat(K_1)}(g_{1,T}(t), \pi_2(g_{2,T}(x)))$$

and since

$$\begin{aligned} pre_{Flat(K_1)}(\hat{t}, v) &= pre_A(\hat{t}, v) \\ &= \sum_{i=1}^{m} (\overline{v}(\hat{term_i}), \hat{p}_i) \end{aligned}$$

for

$$pre_{K_1}(\hat{t}) = \sum_{i=1}^{m} (\hat{term_i}, \hat{p}_i)$$

we have for $\hat{t} = g_{1,T}(t)$ that

$$pre_{Flat(K_1)}(g_{1,T}(t), \pi_2(g_{2,T}(t))) = \sum_{i=1}^{n} (\overline{\pi_2(g_{2,T}(t))}(term_i), g_{1,P}(p_i))$$

where

$$pre_{K_1}(g_{1,T}) = \sum_{i=1}^{n} (term_i, g_{1,P}(p_i))$$

because

$$pre_{Skel(K_1)}(g_{1,T}(t)) = g_{1,P}^{\oplus}(pre_X(t)) = \sum_{i=1}^{n} g_{1,P}(p_i)$$

and for

$$pre_{K_1}(\hat{t}) = \sum_{i=1}^{m} (\hat{term_i}, \hat{p}_i)$$

the pre condition of the skeleton is defined by

$$pre_{Skel(K_1)}(\hat{t}) = \sum_{i=1}^{m} \hat{p}_i$$

So we have to show for $1 \leq i \leq n$ that

$$\overline{\pi_2(g_{2,T}(t))}(term_i) = \pi_1(g_{2,P}(p_i))$$

Since AHL-morphisms preserve arc descriptions and pre conditions we obtain

$$\begin{aligned}
pre_{K_2}(\pi_1(g_{2,T}(t))) &= pre_{K_2}(proj(K_2)(g_{2,T}(t))) \\
&= pre_{K_2}(Skel(f)_T(g_{1,T}(t))) \\
&= pre_{K_2}(f_T(g_{1,T}(t))) \\
&= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(pre_{K_1}(g_{1,T}(t))) \\
&= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(\sum_{i=1}^{n}(term_i, g_{1,P}(p_i))) \\
&= \sum_{i=1}^{n}(term_i, f_P(g_{1,P}(p_i))) \\
&= \sum_{i=1}^{n}(term_i, Skel(f)_P(g_{1,P}(p_i))) \\
&= \sum_{i=1}^{n}(term_i, proj(K_2)(g_{2,P}(p_i))) \\
&= \sum_{i=1}^{n}(term_i, \pi_2(g_{2,P}(p_i)))
\end{aligned}$$

and hence we have

$$\begin{aligned}
\sum_{i=1}^{n}(\overline{\pi_2(g_{2,T}(t))}(term_i), \pi_2(g_{2,P}(p_i))) &= pre_A(\pi_1(g_{2,T}(t)), \pi_2(g_{2,T}(t))) \\
&= pre_A(g_{2,T}(t)) \\
&= pre_{Flat(K_2)}(g_{2,T}(t)) \\
&= g_{2,P}^{\oplus} \circ pre_X(t) \\
&= g_{2,P}^{\oplus}(\sum_{i=1}^{n} p_i) \\
&= \sum_{i=1}^{n} g_{2,P}(p_i) \\
&= \sum_{i=1}^{n}(\pi_1(g_{2,P}(p_i)), \pi_2(g_{2,P}(p_i)))
\end{aligned}$$

which implies $\pi_1(g_{2,P}(p_i)) = \overline{\pi_2(g_{2,T}(t))}(term_i)$ for $1 \leq i \leq n$.
So g preserves pre conditions. The proof for the post conditions works analogously.

commutativity of (2):

$$\begin{aligned}(proj(K_1) \circ g)_P(p) &= proj(K_1)_P \circ g_P(p) \\ &= proj(K_1)_P(\pi_1(g_{2,P}(p)), g_{1,P}(p)) \\ &= g_{1,P}(p)\end{aligned}$$

and

$$\begin{aligned}(proj(K_1) \circ g)_T(t) &= proj(K_1)_T \circ g_T(t) \\ &= proj(K_1)_T(g_{1,T}(t), \pi_2(g_{2,T}(t))) \\ &= g_{1,T}(t)\end{aligned}$$

Commutativity of (3):

$$\begin{aligned}(Flat(f) \circ g)_P(p) &= Flat(f)_P \circ g_P(p) \\ &= Flat(f)_P(\pi_1(g_{2,P}(p)), g_{1,P}(p)) \\ &= (\pi_1(g_{2,P}(p)), f_P(g_{1,P}(p))) \\ &= (\pi_1(g_{2,P}(p)), Skel(f)_P(g_{1,P}(p))) \\ &= (\pi_1(g_{2,P}(p)), (Skel(f) \circ g_1)_P(p)) \\ &= (\pi_1(g_{2,P}(p)), (proj(K_2) \circ g_2)_P(p)) \\ &= (\pi_1(g_{2,P}(p)), \pi_2(g_{2,P}(p))) \\ &= g_{2,P}(p)\end{aligned}$$

and

$$\begin{aligned}(Flat(f) \circ g)_T(t) &= Flat(f)_T \circ g_T(t) \\ &= Flat(f)_T(g_{1,T}(t), \pi_2(g_{2,T}(t))) \\ &= (f_T(g_{1,T}(t)), \pi_2(g_{2,T}(t))) \\ &= (Skel(f)_T(g_{1,T}(t)), \pi_2(g_{2,T}(t))) \\ &= ((Skel(f) \circ g_1)_T(t), \pi_2(g_{2,T}(t))) \\ &= ((proj(K_2) \circ g_2)_T(t), \pi_2(g_{2,T}(t))) \\ &= (\pi_2(g_{2,T}(t)), \pi_2(g_{2,T}(t))) \\ &= g_{2,T}(t)\end{aligned}$$

uniqueness of g:

To show the uniqueness of g we assume $g' : X \to Flat(K_1)$ with $proj(K_1) \circ g' = g_1$ and $Flat(f) \circ g' = g_2$.
From $(Flat(f) \circ g')_P = g_{2,P}$ we obtain for every $p \in P_X$ that

$$\begin{aligned}\pi_1(g'_P(p)) &= \pi_1((id_A \otimes f_P)(g'_P(p))) \\ &= \pi_1(Flat(f)_P(g'_P(p))) \\ &= \pi_1((Flat(f) \circ g')_P(p)) \\ &= \pi_1(g_{2,P}(p)) \\ &= \pi_1(g_P(p))\end{aligned}$$

and from the fact that

$$(proj(K_1) \circ g')_P = g_{1,P}$$

we obtain

$$\begin{aligned}\pi_2(g'_P(p)) &= proj(K_1)_P \circ g'_P(p)\\ &= (proj(K_1) \circ g')_P(p)\\ &= g_{1,P}(p)\\ &= \pi_2(g_P(p))\end{aligned}$$

which implies that

$$\begin{aligned}g'_P(p) &= (\pi_1(g'_P(p)), \pi_2(g'_P(p)))\\ &= (\pi_1(g_P(p)), \pi_2(g_P(p)))\\ &= g_P(p)\end{aligned}$$

and hence $g'_P = g_P$.
Furthermore for every $t \in T_X$ we can use

$$(proj(K_1) \circ g')_T = g_{1,T}$$

to get

$$\begin{aligned}\pi_1(g'_T(t)) &= proj(K_1)_T \circ g'_T(t)\\ &= (proj(K_1) \circ g')_T(t)\\ &= g_{1,T}(t)\\ &= \pi_1(g_T(t))\end{aligned}$$

and we can use

$$(Flat(f) \circ g')_T = g_{2,T}$$

to get

$$\begin{aligned}\pi_2(g'_T(t)) &= \pi_2(f_T(\pi_1(g'_T(t))), \pi_2(g'_T(t)))\\ &= \pi_2(Flat(f)_T(g'_T(t)))\\ &= \pi_2((Flat(f) \circ g')_T(t))\\ &= \pi_2(g_{2,T}(t))\\ &= \pi_2(g_T(t))\end{aligned}$$

and hence we have

$$\begin{aligned}g'_T(t) &= (\pi_1(g'_T(t)), \pi_2(g'_T(t)))\\ &= (\pi_1(g_T(t)), \pi_2(g_T(t)))\\ &= g_T(t)\end{aligned}$$

With $g'_P = g_P$ and $g'_T = g_T$ we have $g' = g$.

$\square$

C.3 Parallel and Sequential Composition and Independence of Algebraic High-Level Processes

Detailed Proof C.5 (Equivalence and Independence of AHL-Processes)
See Theorem 3.17.

Proof. We have to show that

1. there are bijections $e_P : P_K \to P_{K'}$ and $e_T : T_K \to T_{K'}$, s.t. the diagram in Def. 3.10 commutes componentwise and

2. the instantiations INS and INS' are equivalent using that the instantiations INS_1 and INS_2 are consistent.

Proof of Part 1. Let us define gluing points $GP(P_K)$ and $GP(P_{K'}$ by

- $GP(P_K) = (i'_1 \circ i_3(P_I) \cup i'_2 \circ i_4(P_I) \cup i'_1 \circ i_1(P_I))$ where $i'_1 \circ i_1(P_I) = i'_2 \circ i_2(P_I)$
- $GP(P_{K'}) = (i'_4 \circ i_2(P_I) \cup i'_3 \circ i_1(P_I) \cup i'_3 \circ i_3(P_I))$ where $i'_3 \circ i_3(P_I) = i'_4 \circ i_4(P_I)$.

First we show that the three components of $GP(P_K)$ are disjoint which follows symmetrically for $GP(P_{K'})$.

$$i'_1 \circ i_3(P_I) \cap i'_1 \circ i_1(P_I) = i'_1(i_3(P_I) \cap i_1(P_I)) \subseteq i'_1(IN(K_1) \cap OUT(K_1)) = \emptyset$$

by assumption $IN(K_1) \cap OUT(K_1) = \emptyset$.

$$i'_2 \circ i_4(P_I) \cap i'_2 \circ i_2(P_I) = i'_2(i_4(P_I) \cap i_2(P_I)) \subseteq i'_2(OUT(K_2) \cap IN(K_2)) = \emptyset$$

by assumption $IN(K_2) \cap OUT(K_2) = \emptyset$.
For the third intersection we need

Lemma C.6
$IN(K) \cap OUT(K) = \emptyset$ and similar $IN(K') \cap OUT(K') = \emptyset$.

Using Lemma C.6 we can show $i'_1 \circ i_3(P_I) \cap i'_2 \circ i_4(P_I) = \emptyset$:

$$i'_1 \circ i_3(P_I) \cap i'_2 \circ i_4(P_I) \subseteq i'_1(IN(K_1)) \cap i'_2(OUT(K_2)) \subseteq IN(K) \cap OUT(K) = \emptyset$$

Proof of Lemma C.6 Assume that there exists $p \in IN(K) \cap OUT(K)$, we have

$$IN(K) \subseteq i'_1(IN(K_1)) \cup i'_2(IN(K_2))$$

and

$$OUT(K) \subseteq i'_1(OUT(K_1)) \cup i'_2(OUT(K_2)).$$

Case 1 $p \in i'_1(IN(K_1)) \cap i'_1(OUT(K_1)) = i'_1(IN(K_1)) \cap OUT(K_1)) = \emptyset$ (contradiction)

Case 2 $p \in i'_2(IN(K_2)) \cap i'_2(OUT(K_2)) = \emptyset$ (contradiction)

Case 3

$$\begin{aligned} & p \in i'_1(IN(K_1)) \cap i'_2(OUT(K_2)) \\ \Rightarrow \; & \exists p_1 \in IN(K_1), p_2 \in OUT(K_2) \text{ with } p = i'_1(p_1) = i'_2(p_2) \\ \Rightarrow \; & \exists p \in P_I : i_1(p) = p_1, i_2(p) = p_2 \text{ by pushout (1)} \\ \Rightarrow \; & i_1(p) = p_1 \in OUT(K_1) \end{aligned}$$

contradicts $p_1 \in IN(K_1)$ and $IN(K_1) \cap OUT(K_1) = \emptyset$.

Case 4 symmetric to Case 3.

This implies that $GP(P_K)$ is the disjoint union

$$GP(P_K) = GP_1(P_K) \cup GP_2(P_K) \cup GP_3(P_K)$$

with $GP_x(P_K)$ for $x = 1, 2, 3$ defined below.
Now we are able to define $e_P : P_K \to P_{K'}$ via the non gluing points NGP. Let $NGP(P_K) = P_K \setminus GP(P_K)$ then we have the following disjoint unions:

$$\begin{aligned} P_K &= GP_1(P_K) \cup GP_2(P_K) \cup GP_3(P_K) \cup NGP(P_K) \\ P_{K'} &= GP_1(P_{K'}) \cup GP_2(P_{K'}) \cup GP_3(P_{K'}) \cup NGP(P_{K'}) \end{aligned}$$

with

$$\begin{aligned} GP_1(P_K) = i'_1 \circ i_3(P_I), GP_1(P_{K'}) = i'_4 \circ i_2(P_I) \\ GP_2(P_K) = i'_2 \circ i_4(P_I), GP_2(P_{K'}) = i'_3 \circ i_1(P_I) \\ GP_3(P_K) = i'_1 \circ i_1(P_I), GP_3(P_{K'}) = i'_3 \circ i_3(P_I) \end{aligned}$$

and we define e_P by $e_{P_x} : GP_x(P_K) \to GP_x(P_{K'})$ for $x = 1, 2, 3$ for all $p \in P_I$ by

5. $e_{P_1}(i'_1 \circ i_3(p)) = i'_4 \circ i_2(p)$

6. $e_{P_2}(i'_2 \circ i_4(p)) = i'_3 \circ i_1(p)$

7. $e_{P_3}(i'_1 \circ i_1(p)) = i'_3 \circ i_3(p)$

which are bijections because all morphisms are injective. Moreover we have $GP(P_{K_1}) = i_1(P_I) \cup i_3(P_I)$ and $GP(P_{K_2}) = i_2(P_I) \cup i_4(P_I)$.
Now let $NGP(P_{K_x}) = P_{K_x} \setminus GP(P_{K_x})$ for $x = 1, 2$ then we have by Lemma C.7 below:

$$\begin{aligned} NGP(P_K) &= i'_1(NGP(P_{K_1})) \cup i'_2(NGP(P_{K_2})) = NGP_1(P_K) \cup NGP_2(P_K) \\ NGP(P_{K'}) &= i'_3(NGP(P_{K_1})) \cup i'_4(NGP(P_{K_2})) = NGP_1(P_{K'}) \cup NGP_2(P_{K'}) \end{aligned}$$

and we define for $p_1 \in NGP(P_{K_1})$ and $p_2 \in NGP(P_{K_2})$

8. $e_{P_4} : NGP_1(P_K) \to NGP_1(P_{K'})$ by $e_{P_4}(i'_1(p_1)) = i'_3(p_1)$

9. $e_{P_5} : NGP_2(P_K) \to NGP_2(P_{K'})$ by $e_{P_5}(i'_2(p_2)) = i'_4(p_2)$

which are bijections because all morphisms are injective.
Alltogether we have a bijection

$$e_P = e_{P_1} + e_{P_2} + e_{P_3} + e_{P_4} + e_{P_5} : P_K \to P_{K'}$$

Moreover there is a bijection $e_T : T_K \to T_{K'}$ because we have the following disjoint unions

$$T_K \cong T_{K_1} \uplus T_{K_2} \text{ and } T_{K'} \cong T_{K_1} \uplus T_{K_2}$$

Lemma C.7
$NGP(P_K) = P_K \setminus GP(P_K) = i_1'(NGP(P_{K_1})) \cup i_2'(NGP(P_{K_2}))$ and similar for $NGP(P_{K'})$.

Proof of Lemma C.7

$$\begin{array}{rl} & i_1'(NGP(P_{K_1})) \cup i_2'(NGP(P_{K_2})) \\ = & i_1'(P_{K_1} \setminus GP(P_{K_1})) \cup i_2'(P_{K_2} \setminus GP(P_{K_2})) \\ = & i_1'(P_{K_1} \setminus (i_1(P_I) \cup i_3(P_I))) \cup i_2'(P_{K_2} \setminus (i_2(P_I) \cup i_4(P_I))) \\ = & i_1'(P_{K_1}) \setminus (i_1' \circ i_1(P_I) \cup i_1' \circ i_3(P_I)) \cup i_2'(P_{K_2}) \setminus (i_2' \circ i_2(P_I) \cup i_2' \circ i_4(P_I)) \\ = & i_1'(P_{K_1}) \cup i_2'(P_{K_2}) \setminus (GP_1(P_K) \cup GP_2(P_K) \cup GP_3(P_K)) \\ = & P_K \setminus GP(P_K) \end{array}$$

It remains to show for Proof of 1. the following:

Lemma C.8 (Compatibility of e_p and e_T with mp and mp')
The following diagram commutes componentwise.

$$\begin{tikzcd} K \arrow[d, shift right, "e_P"'] \arrow[d, shift left, "e_T"] \arrow[dr, "mp"] & \\ K' \arrow[r, "mp'"'] & AN \end{tikzcd}$$

Proof of Lemma C.8 mp and mp' are defined by the induced morphisms of pushout (1) and (2), where the outer diagrams commute by compatibility of mp_1 and mp_2 with i_1, i_2, i_3 and i_4.

$$\begin{tikzcd} & K_1 \arrow[dr, "i_1'"] \arrow[drr, "mp_1"] & & \\ I \arrow[ur, "i_1"] \arrow[dr, "i_2"'] \arrow[rr, phantom, "(1)"] & & K \arrow[r, "mp"] & AN \\ & K_2 \arrow[ur, "i_2'"'] \arrow[urr, "mp_2"'] & & \end{tikzcd}$$

$$\begin{tikzcd} & K_2 \arrow[dr, "i_4'"] \arrow[drr, "mp_2"] & & \\ I \arrow[ur, "i_4"] \arrow[dr, "i_3"'] \arrow[rr, phantom, "(2)"] & & K' \arrow[r, "mp'"] & AN \\ & K_1 \arrow[ur, "i_3'"'] \arrow[urr, "mp_1"'] & & \end{tikzcd}$$

1. The bijection $e_T : T_K \to T_{K'}$ is induced by id_1 and id_2 in the following diagrams (3) and (4), s.t. (5) commutes if the outer diagrams commute.

$$\begin{tikzcd}
T_{K_1} \arrow[r, "i'_1"] \arrow[d, "id_1"'] \arrow[dr, phantom, "(\mathbf{3})"] & T_K \arrow[dr, "mp"] \arrow[d, "e_T"'] & \\
& & T_{AN} \\
T_{K_1} \arrow[r, "i'_3"'] & T_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
T_{K_2} \arrow[r, "i'_2"] \arrow[d, "id_2"'] \arrow[dr, phantom, "(\mathbf{4})"] & T_K \arrow[dr, "mp"] \arrow[d, "e_T"'] & \\
& & T_{AN} \\
T_{K_2} \arrow[r, "i'_4"'] & T_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

The outer diagrams commute because we have $mp \circ i'_1 = mp_1 = mp' \circ i'_3$ and $mp \circ i'_2 = mp_2 = mp' \circ i'_4$.

2. The bijection $e_P : P_K \rightarrow P_{K'}$ is given by $e_{P_1} + e_{P_2} + e_{P_3} + e_{P_4} + e_{P_5}$. Note that the bijections e_{P_x} are defined by commutativity of diagrams (x) for $x = 1, \ldots, 5$ and the required digram (6) commutes if all the outer diagram commute.

(a)

$$\begin{aligned} mp \circ i'_1 \circ i_3 = &\ mp_1 \circ i_3 \\ = &\ mp_2 \circ i_2 \qquad \text{(by compatibility)} \\ = &\ mp' \circ i'_4 \circ i_2 \end{aligned}$$

$$\begin{tikzcd}
& GP_1(P_K) \arrow[r, hook] \arrow[dd, "e_{P_1}"] & P_K \arrow[dr, "mp"] \arrow[dd, "e_P"'] & \\
P_I \arrow[ur, "i'_1 \circ i_3"] \arrow[dr, "i'_4 \circ i_2"'] & (\mathbf{1}) & & P_{AN} \\
& GP_1(P_{K'}) \arrow[r, hook] & P_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

(b)

$$\begin{aligned} mp \circ i'_2 \circ i_4 = &\ mp_2 \circ i_4 \\ = &\ mp_1 \circ i_1 \qquad \text{(by compatibility)} \\ = &\ mp' \circ i'_3 \circ i_1 \end{aligned}$$

$$\begin{tikzcd}
& GP_2(P_K) \arrow[r, hook] \arrow[dd, "e_{P_2}"] & P_K \arrow[dr, "mp"] \arrow[dd, "e_P"'] & \\
P_I \arrow[ur, "i'_2 \circ i_4"] \arrow[dr, "i'_3 \circ i_1"'] & (\mathbf{2}) & & P_{AN} \\
& GP_2(P_{K'}) \arrow[r, hook] & P_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

(c)

$$\begin{aligned} mp \circ i'_1 \circ i_1 = &\ mp_1 \circ i_1 \\ = &\ mp_2 \circ i_4 \qquad \text{(by compatibility)} \\ = &\ mp' \circ i'_4 \circ i_4 \\ = &\ mp' \circ i'_3 \circ i_3 \end{aligned}$$

$$\begin{tikzcd}
 & GP_3(P_K) \arrow[r, hook] \arrow[dd, "e_{P_3}"] & P_K \arrow[dd, "e_P"] \arrow[dr, "mp"] & \\
P_I \arrow[ur, "i'_1 \circ i_1"] \arrow[dr, "i'_3 \circ i_3"'] \arrow[r, phantom, "(\mathbf{3})"] & & \arrow[r, phantom, "(\mathbf{6})"] & P_{AN} \\
 & GP_3(P_{K'}) \arrow[r, hook] & P_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

(d)

$$\begin{aligned} mp \circ i'_1 &= mp_1 \\ &= mp' \circ i'_3 \end{aligned}$$

$$\begin{tikzcd}
 & NGP_1(P_K) \arrow[r, hook] \arrow[dd, "e_{P_4}"] & P_K \arrow[dd, "e_P"] \arrow[dr, "mp"] & \\
NGP(P_{K_1}) \arrow[ur, "i'_1"] \arrow[dr, "i'_3"'] \arrow[r, phantom, "(\mathbf{4})"] & & \arrow[r, phantom, "(\mathbf{6})"] & P_{AN} \\
 & NGP_1(P_{K'}) \arrow[r, hook] & P_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

(e)

$$\begin{aligned} mp \circ i'_2 &= mp_2 \\ &= mp' \circ i'_4 \end{aligned}$$

$$\begin{tikzcd}
 & NGP_2(P_K) \arrow[r, hook] \arrow[dd, "e_{P_5}"] & P_K \arrow[dd, "e_P"] \arrow[dr, "mp"] & \\
NGP(P_{K_2}) \arrow[ur, "i'_2"] \arrow[dr, "i'_4"'] \arrow[r, phantom, "(\mathbf{5})"] & & \arrow[r, phantom, "(\mathbf{6})"] & P_{AN} \\
 & NGP_2(P_{K'}) \arrow[r, hook] & P_{K'} \arrow[ur, "mp'"'] &
\end{tikzcd}$$

Remark C.9 . For the existence of mp and mp' we need already the compatibilities $mp_1 \circ i_1 = mp_2 \circ i_2$ and $mp_1 \circ i_3 = mp_2 \circ i_4$. In 2(a) - 2(c) we need in addition that all these morphisms are equal.

Proof of Part 2. Given $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ with pushout (3) we have by consistency of INS_1 and INS_2 $L_{init'} = L_{init'_2} \circ_{(J,j_4,j_3)} L_{init'_1}$ with pushout (4) and vice versa, s.t. properties 1-4 in Def. 3.15 are satisfied.
First we have to show according to Def. 3.13.2 that

$$\forall (a,p) \in A_{type(p)} \otimes P_K : (a,p) \in IN(L_{init}) \Leftrightarrow (a, e_P(p)) \in IN(L_{init'})$$

”⇒” Let $(a,p) \in IN(L_{init})$. By construction we have

$$IN(L_{init}) = j'_1(IN(L_{init_1})) \cup j'_2(IN(L_{init_2}) \setminus j_2(J))$$

and

$$IN(L_{init'}) = .j'_4(IN(L_{init'_2})) \cup j'_3(IN(L_{init'_1}) \setminus j_3(J)).$$

Case 1 Let $(a,p) \in j_1'(IN(L_{init_1}))$. Then there exists $(a,p_1) \in IN(L_{init_1})$ with

$$j_1'(a,p_1) = (a, i_1'(p_1)) = (a,p)$$

So we have $i_1'(p_1) = p$.

Case 1.1 Let $(a,p_1) \notin GP(L_{init_1})$. Then we have

$$\begin{aligned}
& (a,p_1) \in IN(L_{init_1}) \setminus GP(L_{init_1}) \subseteq IN(L_{init_1'}) \\
& \qquad\qquad \text{(by consistency)} \\
\Rightarrow\ & j_3'(a,p_1) = (a, i_3'(p_1)) \in IN(L_{init'}) \\
& \qquad\qquad \text{(because } p_1 \notin i_3(P_I) \text{ and Def. } IN(L_{init'})) \\
\Rightarrow\ & (a, e_P(p)) = (a, e_p \circ i_1'(p_1)) = (a, i_3'(p_1)) \in IN(L_{init'}) \\
& \qquad\qquad \text{(by } e_P \text{ commutes on } NGP \text{ (see 8.))} \\
\Rightarrow\ & (a, e_P(p)) \in IN(L_{init'})
\end{aligned}$$

Case 1.2 Let $(a,p_1) \in GP(L_{init_1})$.

Case 1.2.1 Let $p_1 = i_3(p_I)$ for $p_I \in P_I$.

$$\begin{aligned}
& (a,p_1) \in IN(L_{init_1}) \\
\Rightarrow\ & (a, i_3(p_I)) \in IN(L_{init_1}) \\
\Rightarrow\ & (a, i_2(p_I)) \in IN(L_{init_2'}) && \text{(by consistency (see 3.))} \\
\Rightarrow\ & j_4'(a, i_2(p_I)) \in IN(L_{init'}) && \text{(by Def.} IN(L_{init'})) \\
\Rightarrow\ & (a, i_4' \circ i_2(p_I)) \in IN(L_{init'})
\end{aligned}$$

Then we have $(a, e_P(p)) = (a, e_p(i_1' \circ i_3(p_I)) = (a, i_4' \circ i_2(p_I)) \in IN(L_{init'})$ using $p = i_1' \circ i_3(p_I) \Rightarrow e_p(p) = i_4' \circ i_2(p_I)$ (see 5.).

Case 1.2.2 Let $p_1 = i_1(p_I)$ for $p_I \in P_I$. Then $p_1 \in OUT(K_1) \cap IN(K_1)$ which is a contradiction to the assumption that

$$OUT(K_1) \cap IN(K_1) = \emptyset$$

Case 2 Let $(a,p) \in j_2'(IN(L_{init_2}) \backslash j_2(J))$. Then there exists $(a,p_2) \in IN(L_{init_2}) \backslash j_2(J)$ with

$$j_2'(a,p_2) = (a, i_2'(p_2)) = (a,p)$$

So we have $i_2'(p_2) = p$.

Case 2.1 Let $(a,p_2) \notin GP(L_{init_2})$. Then we have

$$\begin{aligned}
& (a,p_2) \in IN(L_{init_2}) \setminus GP(L_{init_2}) \subseteq IN(L_{init_2'}) \\
& \qquad\qquad \text{(by consistency (see 1.))} \\
\Rightarrow\ & j_4'(a,p_2) = (a, i_4'(p_2)) \in IN(L_{init'}) \\
& \qquad\qquad \text{(by Def. } IN(L_{init'})) \\
\Rightarrow\ & (a, e_P(p)) = (a, e_p \circ i_2'(p_2)) = (a, i_4'(p_2)) \in IN(L_{init'}) \\
& \qquad\qquad \text{(by } e_P \text{ commutes on } NGP \text{ (see 9.))} \\
\Rightarrow\ & (a, e_P(p)) \in IN(L_{init'})
\end{aligned}$$

Case 2.2 Let $(a,p_2) \in GP(L_{init_2})$.

Case 2.2.1 Let $p_2 \in i_2(I)$. Then $(a,p_2) \in j_2(J)$ which is a contradiction to the assumption $(a,p) \in j_2'(j_2(J))$ which implies $(a,p_2) \notin j_2(J)$.

Case 2.2.2 Let $p_2 \in i_4(I)$.Then $p_2 \in OUT(K_2) \cap IN(K_2)$ which is a contradiction to the assumption that $OUT(K_2) \cap IN(K_2) = \emptyset$.

"$\Leftarrow$" Let $(a,p) \in IN(L_{init'})$ for $p \in P_{K'}$. We have to show $(a, e_p^{-1}(p)) \in IN(L_{init})$. By construction we have

$$IN(L_{init'}) = j_4'(IN(L_{init_2'})) \cup j_3'(IN(L_{init_1'}) \setminus j_3(J)).$$

Case 1 Let $(a,p) \in j_4'(IN(L_{init_2'}))$. Then there exists $(a,p_2) \in IN(L_{init_2'})$ with

$$j_4'(a,p_2) = (a, i_4'(p_2)) = (a,p)$$

So we have $i_4'(p_2) = p$.

Case 1.1 Let $(a,p_2) \notin GP(L_{init_2'})$. Then we have

$$\begin{array}{ll} & (a,p_2) \in IN(L_{init_2'}) \setminus GP(L_{init_2'}) \subseteq IN(L_{init_2}) \\ & \qquad\qquad\qquad\qquad\qquad\qquad\text{(by consistency)} \\ \Rightarrow & j_2'(a,p_2) = (a, i_2'(p_2)) \in IN(L_{init}) \\ & \qquad\qquad\qquad\text{(because } p_2 \notin i_3(P_I) \text{ and Def. } IN(L_{init})) \\ \Rightarrow & (a, e_P^{-1}(p)) = (a, e_P^{-1} \circ i_4'(p_2)) = (a, i_2'(p_2)) \in IN(L_{init}) \\ & \qquad\qquad\qquad\text{(by } e_P \text{ commutes on } NGP \text{ (see 9.))} \\ \Rightarrow & (a, e_P(p)) \in IN(L_{init}) \end{array}$$

Case 1.2 Let $(a,p_2) \in GP(L_{init_2'})$.

Case 1.2.1 Let $p_2 = i_2(p_I)$ for $p_I \in P_I$.

$$\begin{array}{lll} & (a,p_2) \in IN(L_{init_2'}) & \\ \Rightarrow & (a, i_2(p_I)) \in IN(L_{init_2'}) & \\ \Rightarrow & (a, i_3(p_I)) \in IN(L_{init_1}) & \text{(by consistency)} \\ \Rightarrow & j_1'(a, i_3(p_I)) \in IN(L_{init}) & \text{(by Def.} IN(L_{init})) \\ \Rightarrow & (a, i_1' \circ i_3(p_I)) \in IN(L_{init}) & \end{array}$$

Then we have

$$(a, e_P^{-1}(p)) = (a, e_p^{-1}(i_4' \circ i_2(p_I)) = (a, i_1' \circ i_3(p_I)) \in IN(L_{init})$$

using $p = i_1' \circ i_3(p_I) \Rightarrow e_p(p) = i_4' \circ i_2(p_I)$ (see 5.).

Case 1.2.2 Let $p_2 = i_4(p_I)$ for $p_I \in P_I$. Then $p_2 \in OUT(K_2) \cap IN(K_2)$ which is a contradiction to the assumption that

$$OUT(K_2) \cap IN(K_2) = \emptyset$$

Case 2 Let $(a,p) \in j_3'(IN(L_{init_1'}) \backslash j_3(J))$. Then there exists $(a,p_1) \in IN(L_{init_1'}) \backslash j_3(J)$ with

$$j_3'(a,p_1) = (a, i_3'(p_1)) = (a,p)$$

So we have $i_3'(p_1) = p$.

Case 2.1 Let $(a, p_1) \notin GP(L_{init'_1})$. Then we have

$$\begin{array}{rl} & (a, p_1) \in IN(L_{init'_1}) \setminus GP(L_{init'_1}) \subseteq IN(L_{init_1}) \\ & \hfill \text{(by consistency)} \\ \Rightarrow & j'_1(a, p_1) = (a, i'_1(p_1)) \in IN(L_{init}) \\ & \hfill \text{(by Def. } IN(L_{init})) \\ \Rightarrow & (a, e_P^{-1}(p)) = (a, e_P^{-1} \circ i'_3(p_1)) = (a, i'_1(p_1)) \in IN(L_{init}) \\ & \hfill \text{(by } e_P \text{ commutes on } NGP \text{ (see 8.))} \\ \Rightarrow & (a, e_P^{-1}(p)) \in IN(L_{init}) \end{array}$$

Case 2.2 Let $(a, p_1) \in GP(L_{init'_1})$.

Case 2.2.1 Let $p_1 \in i_3(I)$. Then $(a, p_1) \in j_3(J)$ which is a contradiction to the assumption $(a, p) \notin j'_3(j_3(J))$ which implies $(a, p_1) \notin j_3(J)$.

Case 2.2.2 Let $p_1 \in i_1(I)$.Then $p_1 \in OUT(K_1) \cap IN(K_1)$ which is a contradiction to the assumption that $OUT(K_1) \cap IN(K_1) = \emptyset$.

Moreover we have to show according to Def. 3.13.2 that

$$\forall (a, p) \in A_{type(p)} \otimes P_K : (a, p) \in OUT(L_{init}) \Leftrightarrow (a, e_P(p)) \in OUT(L_{init'})$$

This can be shown analogously to **Case 1** and **Case 2** above.

□

C.4 Instantiations and Initializations

Detailed Proof C.10 (Category **INet**)
See Fact 4.3.

Proof.

the composition is well-defined:

Given two morphisms $f : (L_1, K_1) \to (L_2, K_2), g : (L_2, K_2) \to (L_3, K_3)$ in **INet**, then the diagrams (1) and (2) commute.

$$\begin{tikzcd} L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(1)"] & L_2 \arrow[r, "g_L"] \arrow[d, "in_2"] \arrow[dr, phantom, "(2)"] & L_3 \arrow[d, "in_3"] \\ Flat(K_1) \arrow[r, "Flat(f_H)"'] & Flat(K_2) \arrow[r, "Flat(g_H)"'] & Flat(K_3) \end{tikzcd}$$

We also have
$Flat(g_H) \circ Flat(f_H) = Flat(g_H \circ f_H)$
because $Flat$ is functor and hence

$$\begin{array}{rcl} Flat((g \circ f)_H) \circ in_1 & = & Flat(g_H \circ f_H) \circ in_1 \\ & = & Flat(g_H) \circ Flat(f_H) \circ in_1 \\ & = & Flat(g_H) \circ in_2 \circ f_L \\ & = & in_3 \circ g_L \circ f_L \\ & = & in_3 \circ (g \circ f)_L \end{array}$$

which implies that $g \circ f : (L_1, K_1) \to (L_3, K_3)$ is an **INet**-morphism.

the composition is associative:

$$\begin{aligned} h \circ (g \circ f) &= h \circ (g_L \circ f_L, g_H \circ f_H) \\ &= (h_L \circ (g_L \circ f_L), h_H \circ (g_H \circ f_H)) \\ &= ((h_L \circ g_L) \circ f_L, (h_H \circ g_H) \circ f_H) \\ &= (h_L \circ g_L, h_H, \circ g_H) \circ f \\ &= (h \circ g) \circ f \end{aligned}$$

the identity is well-defined:

$$\begin{aligned} in \circ id_L &= in \\ &= id_{Flat(K)} \circ in \\ &= Flat(id_K) \circ in \end{aligned}$$

the identity is neutral:

$$id_{(L,K)} \circ f = (id_L, id_K) \circ (f_L, f_H) = (id_L \circ f_L, id_K \circ f_H) = (f_L, f_H) = f$$

and

$$f \circ id_{(L,K)} = (f_L, f_H) \circ (id_L, id_K) = (f_L \circ id_L, f_H \circ id_K) = (f_L, f_H) = f$$

□

Detailed Proof C.11 (Unique Instantiation Preimage)
See Lemma 4.5.

Proof.

$$\begin{tikzcd} L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] \arrow[dr, phantom, "(\mathbf{1})"] & L_2 \arrow[d, "in_2"] \\ Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(K_2) \arrow[d, "proj(K_2)"] \\ Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2) \end{tikzcd}$$

existence of L_1:
We construct L_1 as Pullback (1) of $Flat(K_1) \to Flat(K_2) \leftarrow L_2$ and obtain morphisms in_1 and f_L s.t. (1) commutes.

in_1 *is inclusion*:
Since Pullbacks preserve monomorphisms and in_2 is a monomorphism also in_1 is monomorphism and can be chosen to be an inclusion.

$proj(K_1) \circ in_1$ is an isomorphism:

$proj(K_1) \circ in_1$ is injective:
Due to the fact that (1) is Pullback by construction and (2) is a Pullback because $proj$ induces Pullbacks we have that (1)+(2) is a Pullback which preserves monomorphisms. $proj(K_2) \circ in_2$ is a monomorphism because it is an isomorphism and hence $proj(K_1) \circ in_1$ is a monomorphism which means that it is injective.

$(proj(K_1) \circ in_1)_P$ is surjective:
We have to show that for each $x \in P_{Skel(K_1)}$ there exists $yinP_{L_1}$ s.t. $(proj(K_1) \circ in_1)_P(y) = x$.
Let $x \in P_{Skel(K_1)}$.
Then there is $f_{H,P}(x) \in P_{Skel(K_2)}$ and since $proj(K_2) \circ in_2$ is an isomorphism we have a unique $a \in A_{type(f_H(x))}$ s.t. $(a, f_{H,P}(x)) \in P_{L_2}$.
Furthermore $x \in P_{Skel(K_1)}$ implies $x \in P_{K_1}$ and hence by the definition of $Flat$ there exists $(a, x) \in P_{Flat(K_1)}$.
Due to the fact that

$$\begin{aligned} in_{2,P}(a, f_{H,P}(x)) &= (a, f_{H,P}(x)) \\ &= Flat(f_H)_P(a, x) \end{aligned}$$

the Pullback-property implies that there exists $y \in P_{L_1}$ with $in_{1,P}(y) = (a, x)$ and hence

$$(proj(K_1) \circ in_1)_P(y) = proj(K_1)_P(a, x) = x$$

$$\begin{tikzcd} y \arrow[r, mapsto, "f_{L,P}"] \arrow[d, mapsto, "in_{1,P}"'] & (a, f_{H,P}(x)) \arrow[d, mapsto, "in_{2,P}"] \\ (a, x) \arrow[r, mapsto, "Flat(f_H)_P"'] \arrow[d, mapsto, "proj(K_1)_P"'] & (a, f_{H,P}(x)) \arrow[d, mapsto, "proj(K_2)_P"] \\ x \arrow[r, mapsto, "Skel(f_H)_P"'] & f_{H,P}(x) \end{tikzcd} \qquad \begin{tikzcd} y \arrow[r, mapsto, "f_{L,T}"] \arrow[d, mapsto, "in_{1,T}"'] & (f_{H,T}(x), v) \arrow[d, mapsto, "in_{2,T}"] \\ (x, v) \arrow[r, mapsto, "Flat(f_H)_T"'] \arrow[d, mapsto, "proj(K_1)_T"'] & (f_{H,T}(x), v) \arrow[d, mapsto, "proj(K_2)_T"] \\ x \arrow[r, mapsto, "Skel(f_H)_T"'] & f_{H,T}(x) \end{tikzcd}$$

$(proj(K_1) \circ in_1)_T$ is surjective:
We have to show that for each $x \in T_{Skel(K_1}$ there exists $y \in T_{L_1}$ s.t. $(proj(K_1) \circ in_1)_T(y) = x$.
Let $x \in T_{Skel(K_1)}$.
Then there is $f_{H,T}(x) \in T_{Skel(K_2)}$ and since $proj(K_2) \circ in_2$ is an isomorphism there exists a unique consistent assignment
$v : Var(f_{H,T}(x)) \to A$ s.t. $(f_{H,T}(x), v) \in T_{L_2}$.
This means that v satisfies the conditions $cond_{K_2}(f_{H,T}(x))$ and since AHL-morphisms preserve conditions as well as pre and post conditions we have

$$cond_{K_2}(f_{H,T}(x)) = cond_{K_1}(x)$$

and

$$pre_{K_2}(f_{H,T}(x)) = pre_{K_1}(x)$$

and

$$post_{K_2}(f_{H,T}(x)) = post_{K_1}(x)$$

i.e.

$$Var(f_{H,T}(x)) = Var(x)$$

which implies that $v : Var(x) \to A$ is also a consistent assignment of x and hence $(x, v) \in T_{Flat(K_1)}$.
Due to the fact that

$$\begin{aligned} in_{2,T}(f_{H,T}(x), v) &= (f_{H,T}(x), v) \\ &= Flat(f_H)_T(x, v) \end{aligned}$$

the Pullback-property implies that there exists $y \in T_{L_1}$ with $in_{1,T}(y) = (x, v)$ and hence

$$(proj(K_1) \circ in_1)_T(y) = proj(K_1)_T(x, v) = x$$

So $proj(K_1) \circ in_1$ is bijective which means that it is an isomorphism and therefore (L_1, K_1) is in **INet**.

(f_L, f_H) *is* **INet**-*morphism*:
The morphism $(f_L, f_H) : (L_1, K_1) \to (L_2, K_2)$ is an **INet**-morphism because (1) commutes.

uniqueness of L_1:

$$\begin{tikzcd}
L_1' \arrow[rr, bend left, "f_L'"] \arrow[dr, "in_1'"'] & L_1 \arrow[r, "f_L"] \arrow[d, "in_1"] & L_2 \arrow[d, "in_2"] \\
& Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
& Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2)
\end{tikzcd}$$

Let us assume an instantiation L_1' together with an inclusion in_1' and a morphism $f_L' : L_1' \to L_2$ s.t. (L_1', K_1) and (f_L', f_H) are in **INet** and we have $L_1 \neq L_1'$.

- **Case 1:** For $p \in P_{Skel(K_1)}$ there are $a \neq a' \in A_{sort(p)}$ with $(a, p) \in P_{L_1}$ and $(a', p) \in P_{L_1'}$.
  Then we have

$$\begin{aligned} in_{2,P} \circ f_{L,P}(a, p) &= (in_2 \circ f_L)_P(a, p) \\ &= (Flat(f_H) \circ in_1)_P(a, p) \\ &= Flat(f_H)_P(a, p) \\ &= (id_A \otimes f_{H,P})(a, p) \\ &= (a, f_H(p)) \end{aligned}$$

and

$$\begin{array}{rcl} in_{2,P} \circ f'_{L,P}(a', p) & = & (in_2 \circ f'_L)_P(a', p) \\ & = & (Flat(f_H) \circ in'_1)_P(a', p) \\ & = & Flat(f_H)_P(a', p) \\ & = & (id_A \otimes f_{H,P})(a', p) \\ & = & (a', f_{H,P}(p)) \end{array}$$

and since $in_{2,P}$ is an inclusion this means $(a, f_{H,P}(p)), (a', f_{H,P}(p)) \in P_{L_2}$.
Together with the fact that $(proj(K_2) \circ in_2)_P$ is a bijection and

$$\begin{array}{rcl} (proj(K_2) \circ in_2)_P(a, f_{H,P}(p)) & = & f_{H,P}(p) \\ & = & (proj(K_2) \circ in_2)_P(a', f_{H,P}(p)) \end{array}$$

This implies that $a = a'$ which contradicts the assumption that $a \neq a'$.

- **Case 2:** For $t \in T_{Skel(K_1)}$ there are $v, v' : Var(t) \to A$ with $v \neq v'$ and $(t, v) \in T_{L_1}$ and $(t, v') \in T_{L'_1}$.
  Then we have

$$\begin{array}{rcl} in_{2,T} \circ f_{L,T}(t, v) & = & (in_2 \circ f_L)_T(, t, v) \\ & = & (Flat(f_H) \circ in_1)_T(t, v) \\ & = & Flat(f_H)_T(t, v) \\ & = & (f_{H,T}(t), v) \end{array}$$

  and

$$\begin{array}{rcl} in_{2,T} \circ f'_{L,T}(t, v') & = & (in_2 \circ f'_L)_T(t, v') \\ & = & (Flat(f_H) \circ in'_1)_T(t, v') \\ & = & (f_{H,T}(t), v') \end{array}$$

  and since $in_{2,T}$ is an inclusion this means that
  $(f_{H,T}(t), v), (f_{H,T}(t), v') \in T_{L_2}$.
  We can use the fact that $(proj(K_2) \circ in_2)_T$ is a bijection and

$$\begin{array}{rcl} (proj(K_2) \circ in_2)_T(f_{H,T}(t), v) & = & f_{H,T}(t) \\ & = & (proj(K_2) \circ in_2)_T(f_{H,T}(t), v') \end{array}$$

  and obtain that $v = v'$ which contradicts the assumption that $v \neq v'$.

So we have that L_1 and L'_1 have the same data elements on corresponding places and the same assignments on corresponding transitions and since both of them are isomorphic to $Skel(K_1)$ they are also isomorphic to each other which implies that $L_1 = L'_1$ and $in_1 = in'_1$ because both in_1 and in'_1 are inclusions.
It remains to show that $f_L = f'_L$ which follows directly from the uniqueness of instantiation morphisms.

□

Detailed Proof C.12 (Instantiation Preimage Construction)
See Lemma 4.7.

Proof.

$$\begin{tikzcd}
L_1 \arrow[r, "f_L"] \arrow[d, "in_1"'] & L_2 \arrow[d, "in_2"] \\
Flat(K_1) \arrow[r, "Flat(f_H)"'] \arrow[d, "proj(K_1)"'] & Flat(K_2) \arrow[d, "proj(K_2)"] \\
Skel(K_1) \arrow[r, "Skel(f_H)"'] & Skel(K_2)
\end{tikzcd}$$

well-definedness of L_1:

well-definedness of P_{L_1}:
Let $p \in P_{K_1}$ then there is $f_{H,P}(p) \in P_{K_2}$ and due to the isomorphism $proj \circ in_2$ there is $a \in A_{sort(f_{H,P}(p))}$ s.t.

$$(proj(K_2) \circ in_2)_P^{-1}(f_{H,P}(p)) = (a, f_{H,P}(p)) \in P_{L_2}$$

well-definedness of T_{L_1}:
Analogously let $t \in T_{K_1}$ then there is $f_{H,T}(t) \in T_{K_2}$ and an assignment $v : Var(f_{H,T}(t)) \to A$ s.t.

$$(proj(K_2) \circ in_2)_T^{-1}(f_{H,T}(t)) = (f_{H,T}(t), v) \in T_{L_2}$$

well-definedness of pre_{L_1} *and* $post_{L_1}$:
Let $(t, v) \in T_{L_1}$ then by the definition of T_{L_1} there is $t \in T_{K_1} = T_{Skel(K_1)}$ and

$$\begin{aligned}
(f_{H,T}(t), v) &= (proj(K_2) \circ in_2)_T^{-1}(f_{H,T}(t)) \\
&= ((proj(K_2) \circ in_2)^{-1} \circ Skel(f_H))_T(t) \in T_{L_2}
\end{aligned}$$

Let

$$pre_{Skel(K_1)}(t) = \sum_{i=1}^{n} p_i$$

Since P/T-morphisms preserve pre conditions we have

$$\begin{aligned}
pre_{L_2}(f_{H,T}(t), v) &= pre_{L_2} \circ ((proj(K_2) \circ in_2)^{-1} \circ Skel(f_H))_T(t) \\
&= ((proj(K_2) \circ in_2)^{-1} \circ Skel(f_H))_P^{\oplus}(pre_{Skel(K_1)}(t)) \\
&= ((proj(K_2) \circ in_2)^{-1} \circ Skel(f_H))_P^{\oplus}(\sum_{i=1}^{n} p_i) \\
&= ((proj(K_2) \circ in_2)^{-1})_P^{\oplus}(\sum_{i=1}^{n} f_{H,P}(p_i)) \\
&= \sum_{i=1}^{n} (a_i, f_{H,P}(p_i))
\end{aligned}$$

with $a_i \in A_{sort(f_{H,P}(p_i))}$.
So pre_{L_1} and analogously $post_{L_1}$ are well-defined which means that L_1 is well-defined.

well-definedness of f_L:

well-definedness of $f_{L,P}$ and $f_{L,T}$:
$f_{L,P}$ and $f_{L,T}$ are well-defined functions by the definition of L_1 because $(a,p) \in P_{L_1}$ means $f_{L,P}(a,p) = (a, f_{H,P}(p)) \in P_{L_2}$ and $(t,v) \in T_{L_1}$ means $f_{L,T}(t,v) = (f_{H,T}(t), v) \in T_{L_2}$.

f_L is P/T-net morphism:
Furthermore for $(t,v) \in T_{L_1}$ with

$$pre_{L_1}(t,v) = \sum_{i=1}^{n}(a_i, p_i)$$

there is $(f_{H,T}(t), v) \in T_{L_1}$ with

$$pre_{L_2}(f_{H,T}(t), v) = \sum_{i=1}^{n}(a_i, f_{H,P}(p_i))$$

and hence

$$\begin{aligned} pre_{L_2}(f_{L,T}(t,v)) &= pre_{L_2}(f_{H,T}(t), v) \\ &= \sum_{i=1}^{n}(a_i, f_{H,P}(p_i)) \\ &= f_{L,P}^{\oplus}(\sum_{i=1}^{n}(a_i, p_i)) \\ &= f_{L,P}^{\oplus}(pre_{L_1}(t,v)) \end{aligned}$$

and analogously we have

$$post_{L_2} \circ f_{L,T} = f_{L,P}^{\oplus} \circ post_{L_1}$$

i.e. f_L is a P/T-net morphism.

L_1 is instantiation of K_1:

existence of an inclusion $in_1 : L_1 \to Flat(K_1)$:
$(a,p) \in P_{L_1}$ implies $a \in A_{sort(f_H(p))} = A_{sort(p)}$,
i.e. $(a,p) \in P_{Flat(K_1)}$.
$(t,v) \in T_{L_1}$ implies $v : Var(f_{H,T}(t)) \to A$
and $Var(f_{H,T}(t)) = Var(t)$ because

$$cond_{K_2} \circ f_{H,T} = cond_{K_1}$$

and

$$pre_{K_2} \circ f_{H,T} = (id_{T_{OP}(X)} \otimes f_{H,P})^{\oplus} \circ pre_{K_1}$$

and

$$post_{K_2} \circ f_{H,T} = (id_{T_{OP}(X)} \otimes f_{H,P})^{\oplus} \circ post_{K_1}$$

which means that $(t, v) \in T_{Flat(K_1)}$.
So we have $P_{L_1} \subseteq P_{Flat(K_1)}$ and $T_{L_1} \subseteq T_{Flat(K_1)}$ leading to inclusions $in_{1,P} : P_{L_1} \to P_{Flat(K_1)}$ and $in_{1,T} : T_{L_1} \to T_{Flat(K_1)}$. We have to show that $in_1 = (in_{1,P}, in_{1,T})$ preserves pre and post conditions.

in_1 is P/T-net morphism:
Let $(t, v) \in T_{L_1}$ with

$$pre_{L_1}(t, v) = \sum_{i=1}^{n} (a_i, p_i)$$

Then there is

$$\begin{aligned} pre_{Flat(K_2)}(f_{H,T}(t), v) &= pre_{Flat(K_2)} \circ (in_2 \circ f_L)_T(t, v) \\ &= (in_2 \circ f_L)_P^{\oplus} \circ pre_{L_1}(t, v) \\ &= (in_2 \circ f_L)_P^{\oplus}(\sum_{i=1}^{n} (a_i, p_i)) \\ &= \sum_{i=1}^{n} (a_i, f_{H,P}(p_i)) \end{aligned}$$

Due to fact that $pre_{Flat(K_2)} = pre_A$ this means that

$$pre_{K_2}(f_{H,T}(t)) = \sum_{i=1}^{n} (term_i, f_{H,P}(p_i))$$

with $\overline{v}(term_i) = a_i$ for $1 \leq i \leq n$.
Since AHL-morphisms preserve pre and post conditions we have

$$\begin{aligned} pre_{K_2}(f_{H,T}(t)) &= (id_{T_{OP}(X)} \otimes f_{H,P})^{\oplus} \circ pre_{K_1}(t) \\ &= \sum_{i=1}^{n} (term_i, f_{H,P}(p_i)) \end{aligned}$$

which means that

$$pre_{K_1}(t) = \sum_{i=1}^{n} (term_i, p_i)$$

and hence

$$\begin{aligned} pre_{Flat(K_1)}(t, v) &= pre_A(t, v) \\ &= \sum_{i=1}^{n} (\overline{v}(term_i), p_i) \\ &= \sum_{i=1}^{n} (a_i, p_i) \end{aligned}$$

So we have

$$\begin{array}{rcl} in^{\oplus}_{1,P}(pre_{L_1}(t,v)) & = & in^{\oplus}_{1,P}(\sum_{i=1}^{n}(a_i,p_i)) \\ & = & \sum_{i=1}^{n}(a_i,p_i) \\ & = & pre_{Flat(K_1)}(t,v) \\ & = & pre_{Flat(K_1)}(in_{1,T}(t,v)) \end{array}$$

which means that in_1 preserves pre conditions.
The proof for post conditions works analogously.

$proj(K_1) \circ in_1$ is an isomorphism:

$proj(K_1) \circ in_1$ is surjective:
$(proj(K_1) \circ in_1)_P$ is surjective because for every $p \in P_{Skel(K_1)}$ there is

$$(proj(K_2) \circ in_2)^{-1}_P \circ Skel(f_H)_P(p) = (a, f_{H,P}(p)) \in P_{L_2}$$

and hence there is $(a,p) \in P_{L_1}$ with

$$(proj(K_1) \circ in_1)_P(a,p) = p$$

For every $t \in T_{Skel(K_1)}$ there is

$$(proj(K_2) \circ in_2)^{-1}_T \circ Skel(f_H)_T(t) = (f_{H,T}(t), v) \in T_{L_2}$$

and hence hence there $(t,v) \in T_{L_1}$ with

$$(proj(K_1) \circ in_1)_T(t,v) = t$$

which means that $(proj(K_1) \circ in_1)_T$ is surjective.

$proj(K_1) \circ in_1$ is injective:
Let us assume $(a_1,p),(a_2,p) \in P_{L_1}$ with

$$(proj(K_1) \circ in_1)_P(a_1,p) = p = (proj(K_1) \circ in_1)_P(a_2,p)$$

Then we have

$$f_{L,P}(a_1,p) = (a_1, f_{H,P}(p)) \in P_{L_2}$$

and

$$f_{L,P}(a_2,p) = (a_2, f_{H,P}(p)) \in P_{L_2}$$

with

$$\begin{array}{rcl} (proj(K_2) \circ in_2)_P(a_1, f_{H,P}(p)) & = & f_{H,P}(p) \\ & = & (proj(K_2) \circ in_2)_P(a_2, f_{H,P}(p)) \end{array}$$

which means that $a_1 = a_2$ because $(proj(K_2) \circ in_2)_P$ is injective and hence $(proj(K_1) \circ in_1)_P$ is injective.
Analogously we get that $(proj(K_1) \circ in_1)_T$ is injective because $(proj(K_2) \circ in_2)_T$ is injective.

The fact that $proj(K_1) \circ in_1$ is injective together with the fact that $proj(K_1) \circ in_1$ is surjective means that $proj(K_1) \circ in_1$ is bijective and hence it is an isomorphism.

uniqueness of L_1 and f_L:

Due to the fact that in_1 and in_2 are injections and $f_{L,P}$ and $f_{L,T}$ are defined in the same ways as $Flat(f_H)_P$ and $Flat(f_H)_T$ respectively it follows that

$$in_2 \circ f_L = Flat(f_H) \circ in_1$$

and hence (f_L, f_H) is an **INet**-morphism.
From Lemma 4.5 follows that there is a unique instantiation L_1 together with a unique morphism $f_L : L_1 \to L_2$ such that (f_L, f_H) is an **INet**-morphism.

□

Detailed Proof C.13 (Initialization is Pullback)
See 4.9.

Proof.

$$\begin{tikzcd}
X \arrow[dr, "f"] \arrow[drr, bend left, "f_1"] \arrow[ddr, bend right, "f_2"'] \arrow[drr, phantom, "\mathbf{(3)}", near start] \arrow[ddr, phantom, "\mathbf{(2)}"] & & \\
& init \arrow[r, "inc_1"] \arrow[d, "pr"'] \arrow[dr, phantom, "\mathbf{(1)}"] & P_{L_{init}} \arrow[d, "(proj(K)\circ in)_P"] \\
& IN(K) \arrow[r, "inc_2"'] & P_{Skel(K)}
\end{tikzcd}$$

We have to show the commutativity of (1) and the universal property of pullbacks.

commutativity of (1):

The diagram (1) commutes because

$$\begin{aligned}
inc_2 \circ pr(a,p) &= inc_2(p) \\
&= p \\
&= (proj(K) \circ in)_P(a,p) \\
&= (proj(K) \circ in)_P \circ inc_1(a,p)
\end{aligned}$$

universal property:

Let X be a set and $f_1 : X \to P_{L_{init}}$, $f_2 : X \to IN(K)$ set morphisms with

$$proj(K) \circ in \circ f_1 = inc_2 \circ f_2$$

existence of morphism $f : X \to init$:

We define the morphism $f : X \to init$ by $f(x) = f_1(x)$.

For the well-definedness of f we have to check if for every $x \in X$ there is $f_1(x) \in init$. Let $f_1(x) = (a, p)$. Then we have

$$\begin{aligned} p &= (proj(K) \circ in)_P(a, p) \\ &= (proj(K) \circ in)_P \circ f_1(x) \\ &= inc_2 \circ f_2(x) \\ &= f_2(x) \end{aligned}$$

which means that p is an input place of K and hence (a, p) is an input place of L_{init}, i.e. $f_1(x) = (a, p) \in init$.

commutativity of (2):

$$inc_1 \circ f(x) = f(x) = f_1(x)$$

commutativity of (3):
The equation

$$\begin{aligned} inc_2 \circ pr \circ f &= (proj(K) \circ in)_P \circ inc_1 \circ f \\ &= (proj(K) \circ in)_P \circ f_1 \\ &= inc_2 \circ f_2 \end{aligned}$$

together with the fact that inc_2 is a monomorphism implies that (3) commutes.

uniqueness of f:
Let us assume $f' : X \to init$ with $inc_1 \circ f' = f_1$ and $pr \circ f' = f_2$.
Then we have $inc_1 \circ f' = f_1 = inc_1 \circ f$ and since inc_1 is a monomorphism there is $f = f'$, i.e. f is unique.

□

Detailed Proof C.14 (Initialization Preimage Construction)
See Lemma 4.11.

Proof.

$$\begin{tikzcd} init_1 \arrow[r, "f_{L,P} \circ inc_1"] \arrow[d, "pr"'] \arrow[dr, phantom, "(1)"] & P_{L_2} \arrow[d, "(proj(K_2) \circ in_2)_P"] \\ IN(K_1) \arrow[r, "Skel(f_H)_P \circ inc_2"'] & P_{Skel(K_2)} \end{tikzcd}$$

well-definedness of init:
The construction of *init* is well-defined because

$$p \in IN(K_1) \subseteq P_{Skel(K_1)}$$

means

$$f_{H,P}(p) \in P_{Skel(K_2)}$$

which due to the isomorphism $proj(K_2) \circ in_2$ implies $a \in A_{sort(f_{H,P}(p))}$ with

$$(a, f_{H,P}(p)) \in P_{L_2}$$

well-definedness of pr:

$(a,p) \in init$ means by the definition of $init$ that there is $p \in IN(K_1)$.

pr *is surjective*:

The projection is surjective because as mentioned above $p \in IN(K_1)$ implies $(a, f_{H,P}(p)) \in P_{L_2}$ and hence $(a,p) \in init$ with $pr(a,p) = p$.

existence of inc_1:

The inclusion inc_1 exists because

$$p \in IN(K_1) \subseteq P_{Skel(K_1)}$$

by the construction of L_1 means that there is $a \in A_{sort(p)}(p)$ with

$$(a,p) \in P_{L_1}$$

$init = IN(L_1)$:

Since $proj(K_1) \circ in_1$ is an isomorphism of P/T-nets we have that for $(a,p) \in P_{L_1}$ with

$$(proj(K_1) \circ in_1)_P(a,p) \in IN(Skel(K_1)) = IN(K_1)$$

there is $(a,p) \in IN(L_1)$.
So due to the fact that for every $(a,p) \in init$ we have

$$(proj(K_1) \circ in_1)_P \circ inc_1(a,p) = (proj(K_1) \circ in_1)_P(a,p) = p$$

and also

$$p = pr(a,p) \in IN(K_1)$$

we know that $(a,p) \in IN(L_1)$, i.e. $init \subseteq IN(L_1)$.
The surjectivity of pr implies

$$|init| \geq |IN(K_1)| = |IN(L_1)|$$

which together with the fact that $init \subseteq IN(L_1)$ means that $init = IN(L_1)$.

projection pr *is isomorphism*:

$$\begin{tikzcd}
init_1 \arrow[r, "inc_1"] \arrow[dd, "pr"'] \arrow[ddr, phantom, "(2)"] & P_{L_1} \arrow[r, "f_{L,P}"] \arrow[d, "in_{1,P}"'] \arrow[dr, phantom, "(3)"] & P_{L_2} \arrow[d, "in_{2,P}"] \\
 & P_{Flat(K_1)} \arrow[r, "Flat(f_H)_P"] \arrow[d, "proj(K_1)_P"'] \arrow[dr, phantom, "(4)"] & P_{Flat(K_2)} \arrow[d, "proj(K_2)_P"] \\
IN(K_1) \arrow[r, "inc_2"'] & P_{Skel(K_1)} \arrow[r, "Skel(f_H)_P"'] & P_{Skel(K_2)}
\end{tikzcd}$$

By Lemma 4.9 diagram (2) is a Pullback which preserves monomorphism and since $(proj(K_1) \circ in_1)_P$ is an isomorphism of sets it is also a monomorphism implying that pr is a monomorphism. So pr is surjective and injective which means that it is bijective, i.e. it is an isomorphism.

diagram (1) is pullback:

It remains to show that (1) is a Pullback which follows from Pullback composition of (2) and (3)+(4) where diagram (3) is Pullback by Lemma 4.5 and (4) is Pullback by Theorem 2.33 leading to Pullback (3)+(4) by Pullback composition.

□

Detailed Proof C.15 (Preimage of AHL-Occurrence Nets with Instantiations)
See Theorem 4.13

Proof.

well-definedness of IN:

By Lemma 4.11 we have for

$$L_1 = PreIns(f_H)(L_2)$$

and

$$init_1 = PreInit(f_H)(L_2)$$

that

$$init_1 = IN(L_1)$$

and hence

$$IN(PreIns(f_H)(L)) = PreInit(f_H)(L)$$

IN injective $\Rightarrow KI_1$ is AHL-occurrence net with instantiations:

First let us assume that the function IN is injective.
By Lemma 4.7 all $L_1 \in INS_1$ are instantiations of K_1 and by Lemma 4.11 all $init_1 \in INIT_1$ are initializations of K_1.
Due to the definition of $INIT_1$ for every $init_1 \in INIT_1$ there is $L_2 \in INS_2$ with $PreInit(f_H)(L_2) = init_1$ and there is $L_1 = PreIns(f_H)(L_2)$ with $IN(L_1) = init_1$ which means that IN is surjective.
Since IN by assumption is injective, it is bijective and therefore there is a bijection $L = IN^{-1} : INIT_1 \to INS_1$ with

$$IN(L_{init}) = IN(IN^{-1}(init)) = init$$

So we have that KI_1 is an AHL-occurrence net with instantiations.

KI_1 is AHL-occurrence net with instantiations $\Rightarrow IN$ injective:

For the other direction let us assume that KI_1 is an AHL-occurrence net with instantiations.
Then there is a bijection $L : INIT_1 \to INS_1$ with $IN(L_{init}) = init$ implying that IN is a coretraction in sets, i.e. it is injective.

□

Detailed Proof C.16 (Instantiation Interface)
See Theorem 4.18.

Proof. Given (L_1, K_1), (L_2, K_2) in **INet** and **AHLNet**-morphisms $f_{1,H} : K_0 \to K_1$, $f_{2,H} : K_0 \to K_2$.
Let $(I_1, e_{1,H}, m_{1,H})$, $(I_2, e_{2,H}, m_{2,H})$ be the epi-mono-factorization of $f_{1,H}$ resp. $f_{2,H}$ and J_1, J_2 the instantiations of I_1 resp. I_2 together with morphisms $m_{1,L} : J_1 \to L_1, m_{2,L} : J_2 \to L_2$ induced by $m_{1,H}$ resp. $m_{2,H}$.

$$\begin{tikzcd}
 & & Skel(K_0) \arrow[dll, "Skel(f_{1,H})"'] \arrow[dl, "Skel(e_{1,H})"] \arrow[dr, "Skel(e_{2,H})"'] \arrow[drr, "Skel(f_{2,H})"] & & \\
Skel(K_1) & Skel(I_1) \arrow[l, "Skel(m_{1,H})"] & & Skel(I_2) \arrow[r, "Skel(m_{2,H})"'] & Skel(K_2) \\
L_1 \arrow[u, "proj(K_1)\circ in_1"'] & J_1 \arrow[l, "m_{1,L}"] \arrow[u, "proj(I_1)\circ inj_1"'] & & J_2 \arrow[r, "m_{2,L}"'] \arrow[u, "proj(I_2)\circ inj_2"] & L_2 \arrow[u, "proj(K_2)\circ in_2"]
\end{tikzcd}$$

"$\Rightarrow$":

First let (L_1, L_2) be compatible with $(f_{1,H}, f_{2,H})$.
Then there exists an **INet**-morphism $c = (c_L, c_H) : (J_1, I_1) \to (J_2, I_2)$ with $c_H \circ e_{1,H} = e_{2,H}$.
By Lemma 4.5 we obtain a unique instantiation $L_{0,1}$ of K_0 together with a unique morphism $e_{1,L} : L_{0,1} \to J_1$ s.t. $(e_{1,L}, e_{1,H})$ is in **INet** and a unique instantiation $L_{0,2}$ of K_0 together with a unique morphism $e_{2,L} : L_{0,2} \to J_2$ s.t. $(e_{2,L}, e_{2,H})$ is in **INet**.

$$\begin{tikzcd}
Skel(I_1) & & Skel(K_0) \arrow[ll, "Skel(e_{1,H})"'] \arrow[rr, "Skel(e_{2,H})"] & & Skel(I_2) \\
J_1 \arrow[u, "proj(I_1)\circ inj_1"] & L_{0,1} \arrow[l, "e_{1,L}"] \arrow[ur, "proj(K_0)\circ in_{0,1}"] & & L_{0,2} \arrow[r, "e_{2,L}"'] \arrow[ul, "proj(K_0)\circ in_{0,2}"'] & J_2 \arrow[u, "proj(I_2)\circ inj_2"']
\end{tikzcd}$$

Since $f_{1,H} = m_{1,H} \circ e_{1,H}$ we have that

$$\begin{aligned}
(m_{1,L}, m_{1,H}) \circ (e_{1,L}, e_{1,H}) &= (m_{1,L} \circ e_{1,L}, m_{1,H} \circ e_{1,H}) \\
&= (m_{1,L} \circ e_{1,L}, f_{1,H})
\end{aligned}$$

i.e. $L_{0,1}$ together with $f_{1,L} := m_{1,L} \circ e_{1,L}$ is the unique interface and morphism induced by $f_{1,H}$.
Analogously with $f_{2,H} = m_{2,H} \circ e_{2,H}$ we obtain that $L_{0,2}$ together with $f_{2,L} := m_{2,L} \circ e_{2,L}$ is the unique interface and morphism induced by $f_{2,H}$.

$$\begin{tikzcd}
Skel(K_1) & Skel(I_1) \arrow[l, "Skel(m_{1,H})"'] & & Skel(K_0) \arrow[ll, "Skel(e_{1,H})"'] \arrow[rr, "Skel(e_{2,H})"] \arrow[lll, bend right, "Skel(f_{1,H})"'] \arrow[rrr, bend left, "Skel(f_{2,H})"] & & Skel(I_2) \arrow[r, "Skel(m_{2,H})"] & Skel(K_2) \\
L_1 \arrow[u, "proj(K_1)\circ in_1"] & J_1 \arrow[l, "m_{1,L}"] \arrow[u, "proj(I_1)\circ inj_1"'] & L_{0,1} \arrow[l, "e_{1,L}"] \arrow[ur, "proj(K_0)\circ in_{0,1}"] \arrow[ll, bend left, "f_{1,L}"] & & L_{0,2} \arrow[r, "e_{2,L}"'] \arrow[ul, "proj(K_0)\circ in_{0,2}"'] \arrow[rr, bend right, "f_{2,L}"'] & J_2 \arrow[r, "m_{2,L}"'] \arrow[u, "proj(I_2)\circ inj_2"] & L_2 \arrow[u, "proj(K_2)\circ in_2"']
\end{tikzcd}$$

We have to show that $L_{0,1} = L_{0,2}$.

$$\begin{tikzcd}
Skel(I_1) \arrow[rrrr, bend left, "Skel(c_H)"] & & Skel(K_0) \arrow[ll, "Skel(e_{1,H})"'] \arrow[rr, "Skel(e_{2,H})"] & & Skel(I_2) \\
J_1 \arrow[u, "proj(I_1)\circ inj_1"] \arrow[rrrr, bend right, "c_L"'] & L_{0,1} \arrow[l, "e_{1,L}"] \arrow[ur, "proj(K_0)\circ in_{0,1}"] & & L_{0,2} \arrow[ul, "proj(K_0)\circ in_{0,2}"'] \arrow[r, "e_{2,L}"'] & J_2 \arrow[u, "proj(I_2)\circ inj_2"']
\end{tikzcd}$$

By the composition of **INet**-morphisms we get

$$(c_L, c_H) \circ (e_{1,L}, e_{1,H}) = (c_L \circ e_{1,L}, c_H \circ e_{1,H})$$

So we have instantiations $L_{0,1}$ and $L_{0,2}$ of K_0 together with morphisms $c_L \circ e_{1,L}$ and $e_{2,L}$ both induced by $c_H \circ e_{1,H} = e_{2,H}$ which by Lemma 4.5 implies that $L_{0,1} = L_{0,2}$.

"$\Leftarrow$":

Let us assume a unique instantiation L_0 of K_0 and morphisms $f_{1,L} : L_0 \to L_1$, $f_{2,L} : L_0 \to L_2$ s.t. (L_0, K_0) and $(f_{1,L}, f_{1,H}), (f_{2,L}, f_{2,H})$ are in **INet**.
Let $L_{0,1}$ be the unique instantiation of K_0 together with the unique morphism $e_{1,L} : L_{0,1} \to J_1$ s.t. $(e_{1,L}, e_{1,H})$ is in **INet** and $L_{0,2}$ the unique instantiation of K_0 together with the unique morphism $e_{2,L} : L_{0,2} \to J_2$ s.t. $(e_{2,L}, e_{2,H})$ is in **INet**.
From

$$f_{1,H} = m_{1,H} \circ e_{1,H}$$

and

$$f_{2,H} = m_{2,H} \circ e_{2,H}$$

by Lemma 4.5 we get that

$$L_0 = L_{0,1} = L_{0,2}$$

Furthermore from the fact that $e_{1,H}$ is an epimorphism in **AHLNet**, we get that $Skel(e_{1,H})$ is an epimorphism because $Skel$ preserves epimorphisms and hence $e_{1,L}$ is an epimorphism because $proj(K_0) \circ in_0$ and $proj(I_1) \circ inj_1$ are isomorphisms and the following diagram commutes.

$$\begin{tikzcd}
Skel(K_0) \arrow[r, "Skel(e_{1,H})"] & Skel(I_1) \\
L_0 \arrow[u, "proj(K_0)\circ in_0"] \arrow[r, "e_{1,L}"'] & J_1 \arrow[u, "proj(I_1)\circ inj_1"']
\end{tikzcd}$$

We define a morphism

$$c := (c_L, c_H) : (J_1, I_1) \to (J_2, I_2)$$

with

$$c_H : I_1 \to I_2$$

$$c_{H,P}(e_{1,H,P}(p)) = e_{2,H,P}(p)$$

$$c_{H,T}(e_{1,H,T}(t)) = e_{2,H,T}(t)$$

and

$$c_L : J_1 \to J_2$$

$$c_{L,P}(e_{1,L,P}(a,p)) = e_{2,L,T}(a,p)$$

$$c_{L,T}(e_{1,L,T}(t,v)) = e_{2,L,T}(t,v)$$

We have to show that c is well-defined.
Since $e_{1,H}$ and $e_{1,L}$ are epimorphisms in **AHLNet** resp. **PTNet** the functions $e_{1,H,P}$, $e_{1,H,T}$, $e_{1,L,P}$ and $e_{1,L,T}$ are surjective.
Therefore for every element x in I_1 there is an element y in K_0 s.t. $x = e_{1,H}(y)$ and for every element x in J_1 there is an element y in L_0 s.t. $x = e_{1,L}(y)$.
So we have that $c_{H,P}, c_{H,T}, c_{L,P}$ and $c_{L,T}$ are well-defined functions.
It is obviously that c_H is an **AHLNet**-morphism because $e_{1,H}$ and $e_{2,H}$ are **AHLNet**-morphisms and c_L is a **PTNet**-morphism because $e_{1,L}$ and $e_{2,L}$ are **PTNet**-morphisms.
The fact that $c_H \circ e_{1,H} = e_{2,H}$ and $c_L \circ e_{1,L} = e_{2,L}$ follows directly from the definition of c_H and c_L.
It remains to show that c is an **INet**-morphism, i.e. diagram (2) commutes.

$$\begin{tikzcd}
Flat(K_0) \arrow[r, "Flat(e_{1,H})"'] \arrow[rr, bend left, "Flat(e_{2,H})"] & Flat(I_1) \arrow[r, "Flat(c_H)"'] & Flat(I_2) \\
L_0 \arrow[u, "in_0"] \arrow[r, "e_{1,L}"] \arrow[rr, bend right, "e_{2,L}"'] \arrow[ur, phantom, "(\mathbf{1})"] & J_1 \arrow[u, "inj_1"] \arrow[r, "c_L"] \arrow[ur, phantom, "(\mathbf{2})"] & J_2 \arrow[u, "inj_2"']
\end{tikzcd}$$

The diagram (1) commutes because $(e_{1,L}, e_{1,H})$ is an **INet**-morphism.
So we have

$$\begin{aligned}
(inj_2 \circ c_L \circ e_{1,L})_P(a,p) &= inj_{2,P} \circ (c_L \circ e_{1,L})_P(a,p) \\
&= inj_{2,P} \circ e_{2,L,P}(a,p) \\
&= Flat(e_{2,H})_P \circ in_{0,P}(a,p) \\
&= (id_A \otimes e_{2,H,P}) \circ in_{0,P}(a,p) \\
&= (a, e_{2,H,P}(p)) \\
&= (a, c_{H,P}(e_{1,H,P}(p))) \\
&= (id_A \otimes c_{H,P})(a, e_{1,H,P}(p)) \\
&= (id_A \otimes c_{H,P}) \circ (id_A \otimes e_{1,H,P})(a,p) \\
&= (id_A \otimes c_{H,P}) \circ (id_A \otimes e_{1,H,P}) \circ in_{0,P}(a,p) \\
&= Flat(c_H)_P \circ Flat(e_{1,H})_P \circ in_{0,P}(a,p) \\
&= Flat(c_H)_P \circ inj_{1,P} \circ e_{1,L,P}(a,p) \\
&= (Flat(c_H) \circ inj_1 \circ e_{1,L})_P(a,p)
\end{aligned}$$

and

$$\begin{aligned}
(inj_2 \circ c_L \circ e_{1,L})_T(t,v) &= inj_{2,T} \circ (c_L \circ e_{1,L})_T(t,v)\\
&= inj_{2,T} \circ e_{2,L,T}(t,v)\\
&= Flat(e_{2,H})_T \circ in_{0,T}(t,v)\\
&= (e_{2,H,T}(t),v)\\
&= (c_{H,T}(e_{1,H,T}(t)),v)\\
&= Flat(c_H)_T(e_{1,H,T}(t),v)\\
&= Flat(c_H)_T \circ Flat(e_{1,H})_T(t,v)\\
&= Flat(c_H \circ e_{1,H})_T(t,v)\\
&= Flat(c_H \circ e_{1,H})_T \circ in_{0,T}(t,v)\\
&= Flat(c_H)_T \circ Flat(e_{1,H})_T \circ in_{0,T}(t,v)\\
&= Flat(c_H)_T \circ inj_{1,T} \circ e_{1,L,T}(t,v)\\
&= (Flat(c_H) \circ inj_1 \circ e_{1,L})_T(t,v)
\end{aligned}$$

and hence we have

$$inj_2 \circ c_L \circ e_{1,L} = Flat(c_H) \circ inj_1 \circ e_{1,L}$$

which implies

$$inj_2 \circ c_L = Flat(c_H) \circ inj_1$$

because $e_{1,L}$ is an epimorphism.
So (c_L, c_H) is an **INet**-morphism.

□

Detailed Proof C.17 (Category **INet** has Pushouts)
See Fact 4.20.

Proof.

$$\begin{tikzcd}
(J,I) \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & (L_1,K_1) \arrow[d, "g_1"] \\
(L_2,K_2) \arrow[r, "g_2"'] & (L,K)
\end{tikzcd}
\qquad
\begin{tikzcd}
J \arrow[r, "f_{1,L}"] \arrow[d, "f_{2,L}"'] \arrow[dr, phantom, "(\mathbf{2})"] & L_1 \arrow[d, "g_{1,L}"] \\
L_2 \arrow[r, "g_{2,L}"'] & L
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "f_{1,H}"] \arrow[d, "f_{2,H}"'] \arrow[dr, phantom, "(\mathbf{3})"] & K_1 \arrow[d, "g_{1,H}"] \\
K_2 \arrow[r, "g_{2,H}"'] & K
\end{tikzcd}$$

We construct pushout (2) in **AHLNet** leading to pushout (4) in **PTNet** because *Flat* preserves pushouts (Theorem 2.27).

$$\begin{tikzcd}
I \arrow[r, "f_{1,H}"] \arrow[d, "f_{2,H}"'] \arrow[dr, phantom, "(\mathbf{2})"] & K_1 \arrow[d, "g_{1,H}"] \\
K_2 \arrow[r, "g_{2,H}"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
Flat(I) \arrow[r, "Flat(f_{1,H})"] \arrow[d, "Flat(f_{2,H})"'] \arrow[dr, phantom, "(\mathbf{4})"] & Flat(K_1) \arrow[d, "Flat(g_{1,H})"] \\
Flat(K_2) \arrow[r, "Flat(g_{2,H})"'] & Flat(K)
\end{tikzcd}$$

Furthermore we construct pushout (3) and due to the fact that f_1 and f_2 are **INet**-morphisms we have

$$\begin{aligned} Flat(g_{1,H}) \circ in_1 \circ f_{1,L} &= Flat(g_{1,H}) \circ Flat(f_{1,H}) \circ in_I \\ &= Flat(g_{2,H}) \circ Flat(f_{2,H}) \circ in_I \\ &= Flat(g_{2,H}) \circ in_2 \circ f_{2,L} \end{aligned}$$

which due to the pushout property of (3) implies that there is a unique **PTNet**-morphism $in : L \to K$ with

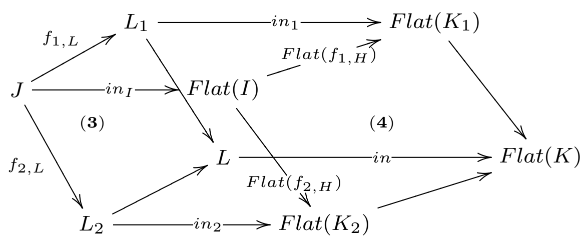
$$in \circ g_{1,L} = Flat(g_{1,H}) \circ in_1$$

and

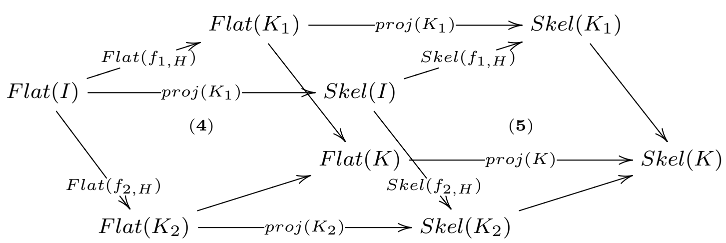
$$in \circ g_{2,L} = Flat(g_{2,H}) \circ in_2$$

$proj(K) \circ in$ is isomorphism:

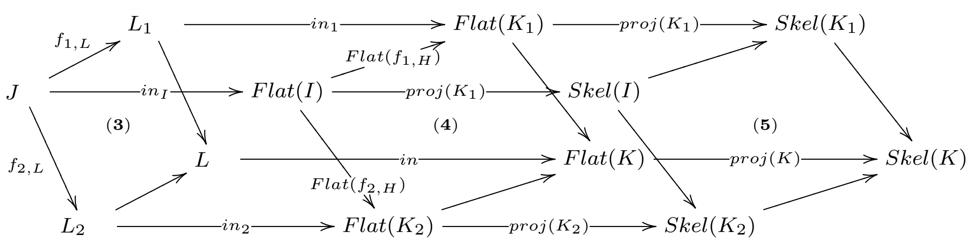
The following cube commutes:



Also the following cube commutes because $proj$ is a natural transformation:



Since $proj(K_1) \circ in_1$, $proj(K_2) \circ in_2$ and $proj(I) \circ in_I$ are isomorphisms and the following cube commutes, the fact that $Skel(K)$ is Pushout implies that $proj(K) \circ in$ is an isomorphism.



injectivity of in:
Due to the fact that $proj(K) \circ in$ is isomorphic, it is injective and hence in is injective and can be chosen to be an inclusion.

So we have that (L, K) and $g_1 = (g_{1,L}, g_{1,H}), g_2 = (g_{2,L}, g_{2,H})$ are in **INet**.
It remains to show that diagram (1) is a pushout in **INet**. The commutativity of (1) follows from the commutativity of (2) and (3).
Let (L', K') an **INet**-object together with **INet**-morphisms $h_1 = (h_{1,L}, h_{1,H}) : (L_1, K_1) \to (L', K')$ and $h_2 = (h_{2,L}, h_{2,H}) : (L_2, K_2) \to (L', K')$ such that

$$h_1 \circ f_1 = h_2 \circ f_2$$

Due to the fact that the composition in **INet** is defined componentwise there is

$$\begin{aligned} (h_{1,L} \circ f_{1,L}, h_{1,H} \circ f_{1,H}) &= h_1 \circ f_1 \\ &= h_2 \circ f_2 \\ &= (h_{2,L} \circ f_{2,L}, h_{2,H} \circ f_{2,H}) \end{aligned}$$

and hence

$$h_{1,L} \circ f_{1,L} = h_{2,L} \circ f_{2,L}$$

which due to pushout (3) implies that there is a unique $h_L : L \to L'$ with

$$h_L \circ g_{1,L} = h_{1,L}$$

and

$$h_L \circ g_{2,L} = h_{2,L}$$

and due to pushout (2) there is a unique $h_H : K \to K'$ with

$$h_H \circ g_{1,H} = h_{1,H}$$

and

$$h_H \circ g_{2,H} = h_{2,H}$$

For the fact that $h = (h_L, h_H)$ is an **INet**-morphism we have to show that

$$h_L \circ in = in' \circ Flat(h_H)$$

We do the proof componentwise for places and transitions. Since the pushout in **PTNet** can be constructed componentwise in **SET** there is P_L the pushout of P_{L_1} and P_{L_2} and T_L the pushout of T_{L_1} and T_{L_2}. So we have that $g_{1,P}$ and $g_{2,P}$ and also $g_{1,T}$ and $g_{2,T}$ are jointly surjective.
Let $(a, p) \in P_L$ then there is $(a_1, p_1) \in P_{L_1}$ with $(a, p) = g_{1,L}(a_1, p_1)$ or there is $(a_2, p_2) \in P_{L_2}$ with $(a, p) = g_{2,L}(a_2, p_2)$.

Case 1: $(a, p) = g_{1,L}(a_1, p_1)$

Then from the facts that h_1 and g_1 are **INet**-morphisms and $Flat$ is a functor we obtain

$$\begin{aligned} in' \circ h_L(a,p) &= in' \circ h_L(g_{1,L}(a_1,p_1)) \\ &= in' \circ h_{1,L}(a_1,p_1) \\ &= Flat(h_{1,H}) \circ in_1(a_1,p_1) \\ &= Flat(h_H \circ g_{1,H}) \circ in_1(a_1,p_1) \\ &= Flat(h_H) \circ Flat(g_{1,H}) \circ in_1(a_1,p_1) \\ &= Flat(h_H) \circ in \circ g_{1,L}(a_1,p_1) \\ &= Flat(h_H) \circ in(g_{1,L}(a_1,p_1)) \end{aligned}$$

Case 2: $(a, p) = g_{2,L}(a_2, p_2)$

Then from the facts that h_2 and g_2 are **INet**-morphisms and $Flat$ is a functor we obtain

$$\begin{aligned} in' \circ h_L(a,p) &= in' \circ h_L(g_{2,L}(a_2,p_2)) \\ &= in' \circ h_{2,L}(a_2,p_2) \\ &= Flat(h_{2,H}) \circ in_2(a_2,p_2) \\ &= Flat(h_H \circ g_{2,H}) \circ in_2(a_2,p_2) \\ &= Flat(h_H) \circ Flat(g_{2,H}) \circ in_1(a_2,p_2) \\ &= Flat(h_H) \circ in \circ g_{2,L}(a_2,p_2) \\ &= Flat(h_H) \circ in(g_{2,L}(a_2,p_2)) \end{aligned}$$

The proof for the transitions works completely analogously. Hence $h = (h_L, h_H)$ is an **INet**-morphism with

$$h \circ g_1 = (h_L, h_H) \circ (g_{1,L}, g_{1,H}) = (h_L \circ g_{1,L}, h_H \circ g_{1,H}) = (h_{1,L}, h_{1,H}) = h_1$$

and

$$h \circ g_2 = (h_L, h_H) \circ (g_{2,L}, g_{2,H}) = (h_L \circ g_{2,L}, h_H \circ g_{2,H}) = (h_{2,L}, h_{2,H}) = h_2$$

To show the uniqueness of h let us assume that there is an **INet**-morphism $h' = (h'_L, h'_H) : (L, K) \to (L', K')$ with

$$h' \circ g_1 = h_1$$

and

$$h' \circ g_2 = h_2$$

Then due to the componentwise composition of **INet**-morphisms there is

$$(h'_L \circ g_{1,L}, h'_H \circ g_{1,H}) = h' \circ g_1 = h_1 = (h_{1,L}, h_{1,H})$$

and

$$(h'_L \circ g_{2,L}, h'_H \circ g_{2,H}) = h' \circ g_2 = h_2 = (h_{2,L}, h_{2,H})$$

This means that

$$h'_L \circ g_{1,L} = h_{1,L}$$

and

$$h'_L \circ g_{2,L} = h_{2,L}$$

which due to the uniqueness of h_L implies that $h_L = h'_L$.
Furthermore it means that

$$h'_H \circ g_{1,H} = h_{1,H}$$

and

$$h'_H \circ g_{2,H} = h_{2,H}$$

which due to the uniqueness of h_H implies that $h_H = h'_H$.
Hence there is

$$h = (h_L, h_H) = (h'_L, h'_H) = h'$$

□

Detailed Proof C.18 (Special Uniqueness of Pushouts in **INet**)
See Fact 4.22.

Proof. For $(a,p) \in P_{L_{init_1}}$ there is

$$\begin{aligned} j'_1(a,p) &= in \circ j'_1(a,p) \\ &= Flat(i'_1) \circ in_1(a,p) \\ &= Flat(i'_1)(a,p) \\ &= (a, i'_1(p)) \in P_L \end{aligned}$$

and

$$\begin{aligned} j'_3(a,p) &= in' \circ j'_3(a,p) \\ &= Flat(i'_1) \circ in_1(a,p) \\ &= Flat(i'_1)(a,p) \\ &= (a, i'_1(p)) \in P_{L'} \end{aligned}$$

and analogously we get for $(a,p) \in P_{L_{init_2}}$ that there is

$$(a, i'_2(p)) \in P_L \text{ and } (a, i'_2(p)) \in P_{L'}$$

Since the pushout in **INet** can be constructed componentwise there are pushouts (4) and (5) in **PTNet**.

$$\begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(4)"] & L_{init_1} \arrow[d, "j'_1"] \\ L_{init_2} \arrow[r, "j'_2"'] & L \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(5)"] & L_{init_1} \arrow[d, "j'_3"] \\ L_{init_2} \arrow[r, "j'_4"'] & L' \end{tikzcd}$$

Due to the pushout property j_1 and j_2 and also j_3 and j_4 are jointly surjective which means that we have for $(a,p) \in P_L$ that also $(a,p) \in P_{L'}$ and hence

$$\begin{aligned} (proj(K) \circ in')_P^{-1} \circ (proj(K) \circ in)_P(a,p) &= (proj(K) \circ in')_P^{-1}(p) \\ &= (a,p) \end{aligned}$$

i.e. the bijection $(proj(K) \circ in')_P^{-1} \circ (proj(K) \circ in)_P$ is the identity.
Analogously we obtain that the bijection $(proj(K) \circ in')_T^{-1} \circ (proj(K) \circ in)_T$ is the identity which implies that

$$(proj(K) \circ in')^{-1} \circ (proj(K) \circ in) = id_L$$

i.e. $L = L'$. □

Detailed Proof C.19 (Pushout of Instantiations is Injective)
See Fact 4.23.

Proof. Given pushouts (1) in **AHLNet** and (2) and (3) in **PTNet** such that (4) and (5) are pushouts in **INet**, i.e.

$$L = L_1 \circ_{(J,I)} L_2$$

and

$$L' = L_3 \circ_{(J',I)} L_4$$

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "i_1'"] \\ K_2 \arrow[r, "i_2'"'] & K \end{tikzcd} \qquad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(2)"] & L_1 \arrow[d, "j_1'"] \\ L_2 \arrow[r, "j_2'"'] & L \end{tikzcd} \qquad \begin{tikzcd} J' \arrow[r, "j_3"] \arrow[d, "j_4"'] \arrow[dr, phantom, "(3)"] & L_3 \arrow[d, "j_3'"] \\ L_4 \arrow[r, "j_4'"'] & L' \end{tikzcd}$$

$$\begin{tikzcd} (J,I) \arrow[r, "(j_1,i_1)"] \arrow[d, "(j_2,i_2)"'] \arrow[dr, phantom, "(4)"] & (L_1,K_1) \arrow[d, "(j_1',i_1')"] \\ (L_2,K_2) \arrow[r, "(j_2',i_2')"'] & (L,K) \end{tikzcd} \qquad \begin{tikzcd} (J',I) \arrow[r, "(j_3,i_1)"] \arrow[d, "(j_4,i_2)"'] \arrow[dr, phantom, "(5)"] & (L_3,K_1) \arrow[d, "(j_3',i_1')"] \\ (L_4,K_2) \arrow[r, "(j_4',i_2')"'] & (L',K) \end{tikzcd}$$

Let

$$(L_1, L_2) \neq (L_3, L_4)$$

i.e. $L_1 \neq L_3$ and $L_2 \neq L_4$. We have to show that $L \neq L'$.
The fact that $(L_1, L_2) \neq (L_3, L_4)$ means

$$L_1 \neq L_3 \text{ or } L_2 \neq L_4$$

Due to symmetry of pushouts without any loss of generality we can assume that

$$L_1 \neq L_3$$

Since L_1 and L_3 are instantiations of K_1 for every $p \in P_{Skel(K_1)}$ there is

$$(a, p) = (proj(K_1) \circ in_1)_P^{-1}(p) \in P_{L_1}$$

and

$$(a', p) = (proj(K_1) \circ in_3)_P^{-1}(p) \in P_{L_3}$$

Furthermore for every $t \in T_{Skel(K_1)}$ there is

$$(t, v) = (proj(K_1) \circ in_1)_T^{-1}(t) \in T_{L_1}$$

and
$$(t, v') = (proj(K_1) \circ in_3)_T^{-1}(T) \in T_{L_3}$$
The fact that $proj(K_1) \circ in_1$ and $proj(K_1) \circ in_3$ are isomorphisms means that
$$L_1 \cong L_3$$
i.e. L_1 and $L3$ are equal up to renaming.
So $L_1 \neq L_3$ implies that there is $p \in P_{Skel(K_1)}$ and $a \neq a' \in A_{type(p)}$ s.t.
$$(a, p) \in P_{L_1} \text{ and } (a', p) \in P_{L_3}$$
or there is $t \in T_{Skel(K_1)}$ and $v \neq v' : X \to A$ s.t.
$$(t, v) \in T_{L_1} \text{ and } (t, v') \in T_{L_3}$$

<u>*Case 1:*</u> There is $p \in P_{Skel(K_1)}$ and $a \neq a' \in A_{type(p)}$ s.t. $(a, p) \in P_{L_1}$ and $(a', p) \in P_{L_3}$.
Then there is
$$i'_1(p) \in P_K$$
$$j'_1(a, p) \in P_L$$
$$j'_3(a', p) \in P_{L'}$$
and since the morphisms (j'_1, i'_1) and (j'_3, i'_1) are **INet**-morphisms we have
$$\begin{aligned} j'_1(a, p) &= in \circ j'_1(a, p) \\ &= Flat(i'_1) \circ in_1(a, p) \\ &= Flat(i'_1)(a, p) \\ &= (a, i'_1(p)) \in P_L \end{aligned}$$
and
$$\begin{aligned} j'_3(a', p) &= in \circ j'_3(a', p) \\ &= Flat(i'_1) \circ in_3(a', p) \\ &= Flat(i'_1)(a', p) \\ &= (a', i'_1(p)) \in P_{L'} \end{aligned}$$
This implies that $L \neq L'$ because L and L' are instantiations of K and there is $i'_1(p) \in P_K$ and $a \neq a' \in A_{type(i'_1(p))}$ with
$$(a, i'_1(p)) \in P_L \text{ and } (a', i'_1(p)) \; inP_{L'}$$

<u>*Case 2:*</u> There is $t \in T_{Skel(K_1)}$ and $v \neq v' : X \to A$ s.t. $(t, v) \in T_{L_1}, (t, v') \in T_{L_3}$.
This implies
$$\begin{aligned} j'_1(t, v) &= in \circ j'_1(t, v) \\ &= Flat(i'_1) \circ in_1(t, v) \\ &= Flat(i'_1)(t, v) \\ &= (a, i'_1(p)) \in T_L \end{aligned}$$

and

$$\begin{aligned} j_3'(t,v') &= in \circ j_3'(t,v') \\ &= Flat(i_1') \circ in_3(t,v') \\ &= Flat(i_1')(t,v') \\ &= (a', i_1'(p)) \in T_{L'} \end{aligned}$$

which analogously to case 1 implies that $L \neq L'$.

So we have that

$$(L_1, L_2) \neq (L_3, L_4) \Rightarrow L_1 \circ_{(J,I)} L_2 \neq L_3 \circ_{(J',I)} L_4$$

which by contraposition means that

$$L_1 \circ_{(J,I)} L_2 = L_3 \circ_{(J',I)} L_4 \Rightarrow (L_1, L_2) = (L_3, L_4)$$

□

Detailed Proof C.20 (Category **AHLNetI**)
See Fact 4.25.

Proof.

the composition is well-defined:
Let $f = (f_N, f_I) : (K_1, INS_1) \to (K_2, INS_2)$ and $g = (g_N, g_I) : (K_2, INS_2) \to (K_3, INS_3)$ be **AHLNetI**-morphisms. Then for every $L \in INS_2$ there is an **INet**-morphim $(f_L, f_N) : (f_I(L), K_1) \to (L, K_2)$ and for every $L \in INS_3$ there is an **INet**-morphism $(g_L, g_N) : (g_I(L), K_2) \to (L, K_3)$. Note that normally the index L at the name of a morphism just indicates that it is a low-level morphism but this time we mean the name of the instantiation L. The composition of (f_N, f_I) and (g_N, g_I) is defined as

$$g \circ f = (g_N, g_I) \circ (f_N, f_I) = (g_N \circ f_N, f_I, g_I)$$

where due to the well-defined composition of AHL-morphisms $(g \circ f)_N = g_N \circ f_N : K_1 \to K_3$ is an AHL-morphism and due to the well-defined composition of functions $(g \circ f)_I = f_I \circ g_I : INS_3 \to INS_1$ is a function.
Furthermore for every $L \in INS_3$ there is

$$(g_L, g_N) : (g_I(L), K_2) \to (L, K_3)$$

and

$$(f_{g_I(L)}, f_N) : (f_I \circ g_I(L), K_1) \to (g_I(L), K_2)$$

So we can define

$$((g \circ f)_L, (g \circ f)_N) = (f_{g_I(L)} \circ g_L, (g \circ f)_N) : (L, K_3) \to (f_I \circ g_I(L), K_1)$$

which due to the well-defined composition in the category **INet** is an **INet**-morphism.

the composition is associative:

$$\begin{aligned}(h_N, h_I) \circ ((g_N, g_I) \circ f_N, f_I) &= (h_N, h_I) \circ (g_N \circ f_N, f_I \circ g_I)\\ &= (h_N \circ (g_N \circ f_N), (f_I \circ g_I) \circ h_I)\\ &= ((h_N \circ g_N) \circ f_N, f_I \circ (g_I \circ h_I))\\ &= (h_N \circ g_N, g_I \circ h_I) \circ (f_N, f_I)\\ &= ((h_N, h_I) \circ (g_N, g_I)) \circ (f_N, f_I)\end{aligned}$$

the identity is well-defined:

Let (K, INS) in **AHLNetI**. For every $L \in INS$ there exists the **INet**-morphism $id_{(L,K)} : (L, K) \to (L, K)$.

the identity is neutral:

$$id_{(K,INS)} \circ (f_N, f_I) = (id_K, id_{INS}) \circ (f_N, f_I) = (id_K \circ f_N, f_I \circ id_{INS}) = (f_N, f_I)$$

and

$$(f_N, f_I) \circ id_{(K,INS)} = (f_N, f_I \circ (id_K, id_{INS}) = (f_N \circ id_K, id_{INS} \circ f_I) = (f_N, f_I)$$

□

Detailed Proof C.21 (**AHLNetI** has Pushouts)
See Fact 4.27.

Proof.

$$\begin{tikzcd} K_0 \arrow[r, "f_N"] \arrow[d, "g_N"'] \arrow[dr, phantom, "(2)"] & K_1 \arrow[d, "f'_N"] \\ K_2 \arrow[r, "g'_N"'] & K_3 \end{tikzcd} \qquad \begin{tikzcd} (L_0, K_0) \arrow[r, "(f_L, f_N)"] \arrow[d, "(g_L, g_N)"'] \arrow[dr, phantom, "(3)"] & (L_1, K_1) \arrow[d, "(f'_L, f'_N)"] \\ (L_2, K_2) \arrow[r, "(g'_L, g'_N)"'] & (L_3, K_3) \end{tikzcd} \qquad \begin{tikzcd} INS_0 \arrow[dr, phantom, "(4)"] & INS_1 \arrow[l, "f_I"'] \\ INS_2 \arrow[u, "g_I"] & INS_3 \arrow[l, "g'_I"] \arrow[u, "f'_I"'] \end{tikzcd}$$

construction of (K, INS):

First we construct the pushout (2) in **AHLNet**.

Due to the definition of **AHLNetI**-morphisms for $L_1 \in INS_1$ and $L_2 \in INS_2$ with

$$f_I(L_1) = L_0 = g_I(L_2)$$

there are **INet**-morphisms $(f_L, f_N) : (L_0, K_0) \to (L_1, K_1)$ and $(g_L, g_N) : (L_0, K_0) \to (L_2, K_2)$. Thus there exists a pushout $(L_1, K_1) \circ_{(L_0,K_0)} (L_2, K_2)$ and we can define

$INS_3 = \{L_3 \mid L_1 \in INS_1, L_2 \in INS_2$ with $f_I(L_1) = g_I(L_2)$ and (3) is pushout in **INet** $\}$

$$\begin{tikzcd} (L_0, K_0) \arrow[r, "(f_L, f_N)"] \arrow[d, "(g_L, g_N)"'] \arrow[dr, phantom, "(3)"] & (L_1, K_1) \arrow[d, "(f'_L, f'_N)"] \\ (L_2, K_2) \arrow[r, "(g'_L, g'_N)"'] & (L_3, K_3) \end{tikzcd}$$

We define $f' : (K_1, INS_1) \to (K_3, INS_3)$ with $f' = (f'_N, f'_I)$ where

$$f'_I(L_3) = PreIns(f'_N)(L_3)$$

and we define $g' : (K_2, INS_2) \to (K_3, INS_3)$ with $g' = (g'_N, g'_I)$ where

$$f'_I(L_3) = PreIns(g'_N)(L_3)$$

well-definedness:
Due to the definition of INS_3 for $L_3 \in INS_3$ there are $L_1 \in INS_1$ and $L_2 \in INS_2$ such that (3) is pushout in **INet** which together with the uniqueness of instantiation preimages implies that L_1 and L_2 are unique instantiations such that

$$L_1 = PreIns(f'_N)(L_3)$$

and

$$L_2 = PreIns(g'_N)(L_3)$$

and there are the required **INet**-morphism. Hence f'_I and g'_I are well-defined functions and (f'_N, f'_I) and (g'_N, g'_I) are well-defined **AHLNetI**-morphisms.

(1) is pushout:
Furthermore due to pushout (2) there is

$$f'_N \circ f_N = g'_N \circ g_N$$

and due to the construction of INS_3 there is

$$f_I \circ f'_I = g_I \circ g'_I$$

and hence

$$(f'_N, f'_I) \circ (f_N, f_I) = (g'_N, g'_I) \circ (g_N, g_I)$$

$$\begin{tikzcd}
(K_0, INS_0) \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(1)"] & (K_1, INS_1) \arrow[d, "f'"] \arrow[ddr, bend left, "x"] & \\
(K_2, INS_2) \arrow[r, "g'"'] \arrow[drr, bend right, "y"'] & (K_3, INS_3) & \\
& & (X, INS_X)
\end{tikzcd}$$

To show that (1) is pushout let (X, INS_X) an AHL-net with instantiations together with $x : (K_1, INS_1) \to (X, INS_X)$ and $y : (K_2, INS_2) \to (X, INS_X)$ such that

$$x \circ f = y \circ g$$

Then there is

$$x_N \circ f_N = y_N \circ g_N$$

which due to pushout (2) implies that there is a unique morphism $z_N : K_3 \to X$ such that

$$z_N \circ f'_N = x_N$$

and

$$z_N \circ g'_N = y_N$$

Furthermore we define $z_I : INS_X \to INS_3$ with

$$z_I(L) = L_3$$

where (L_3, K_3) is the pushout object of pushout (3) below where $L_1 = x_I(L)$, $L_2 = y_I(L)$ and $L_0 = f_I \circ x_I(L)$.
$L_3 \in INS_3$ because

$$f_I(L_1) = f_I \circ x_I(L) = g_I \circ y_I(L) = g_I(L_2)$$

and due to Fact 4.22 the pushout object (L_3, K_3) is unique which means that z_I is a well-defined function.
Furthermore there is

$$f'_I \circ z_I(L) = f'_I(L_3) = L_1 = x_I(L)$$

and

$$g'_I \circ z_I(L) = g'_I(L_3) = L_2 = y_I(L)$$

$$\begin{tikzcd}
(L_0, K_0) \arrow[r, "{(f_L,f_N)}"] \arrow[d, "{(g_L,g_N)}"'] \arrow[dr, phantom, "\mathbf{(3)}"] & (L_1, K_1) \arrow[d, "{(f'_L,f'_N)}"] \\
(L_2, K_2) \arrow[r, "{(g'_L,g'_N)}"'] & (L_3, K_3)
\end{tikzcd}$$

Since x and y are **AHLNetI**-morphisms there are **INet**-morphisms $(x_L, x_N) : (L_1, K_1) \to (L, X)$ and $(y_L, y_N) : (L_2, K_2) \to (L, X)$. From Lemma 4.4 and the fact that

$$x_N \circ f_N = y_N \circ g_N$$

follows that

$$x_L \circ f_L = y_L \circ g_L$$

and hence

$$(x_L, x_N) \circ (f_L, f_N) = (x_L \circ f_L, x_N \circ f_N) = (y_L \circ g_L, y_N \circ g_N) = (y_L, y_N) \circ (g_L, g_N)$$

So the pushout property of (3) implies that there is a unique **INet**-morphism $(z_L, z'_N) : (L_3, K_3) \to (L, X)$ such that

$$(z_L, z'_N) \circ (f'_L, f'_N) = (x_L, x_N)$$

and

$$(z_L, z'_N) \circ (g'_L, g'_N) = (y_L, y_N)$$

Since this implies that $z'_N \circ f'_N = x_N$ and $z'_N \circ g'_N = y_N$ from the uniqueness of z_N follows that $z'_N = z_N$.
Hence we have that for every $L \in INS_X$ there is an **INet**-morphism $(z_L, z_N) : (z_I(L), K_3) \to (L, X)$ which means that $(z_N, z_I) : (L_3, INS_3) \to (X, INS_X)$ is a well-defined **AHLNetI**-morphism. The equalities

$$(z_N, z_I) \circ (f'_N, f'_I) = (x_N, x_I)$$

and

$$(z_N, z_I) \circ (g'_N, g'_I) = (y_N, y_I)$$

follow from the equalities for the components.
It remains to show that (z_N, z_I) is unique.
Let $(z'_N, z'_I) : (K_3, INS_3) \to (X, INS_X)$ with

$$(z'_N, z'_I) \circ (f'_N, f'_I) = (x_N, x_I)$$

and

$$(z'_N, z'_I) \circ (g'_N, g'_I) = (y_N, y_I)$$

Since this implies that $z'_N \circ f'_N = x_N$ and $z'_N \circ g'_N = y_N$ from the uniqueness of z_N follows that $z'_N = z_N$.
Let $L \in INS_X$ and $z'_I(L) = L'$.
Using the properties of **AHLNetI**-morphism we obtain the **INet**-morphisms in diagram (5).

$$\begin{tikzcd}
(L_0, K_0) \arrow[r, "(f_L, f_N)"] \arrow[d, "(g_L, g_N)"'] \arrow[dr, phantom, "(\mathbf{5})"] & (L_1, K_1) \arrow[d, "(f''_L, f'_N)"] \\
(L_2, K_2) \arrow[r, "(g''_L, g'_N)"'] & (L', K_3)
\end{tikzcd}$$

We show that (5) is a pushout in **INet**.
From the commutativity of diagram (2) and Lemma 4.4 follows that (5) commutes. Furthermore from $z'_I(L) = L'$ follows that there is an **INet**-morphism $(z'_L, z_N) : (L', K_3) \to (L, X)$.
Let

$$L_1 = x_I(L)$$

implying **INet**-morphisms

$$(x_L, x_N) : (L_1, K_1) \to (L, X)$$

and let

$$L_2 = y_I(L) = g'_I \circ z'_I(L) = g'_I(L')$$

implying **INet**-morphisms

$$(y_L, y_N) : (L_2, K_2) \to (L, X)$$

From $z_N \circ f'_N = x_N$ follows $z'_L \circ f''_L = x_L$ and from $z_N \circ g'_N = y_N$ follows $z'_L \circ g''_L = y_L$ due to Lemma 4.4.
That means that

$$(z'_L, z_N) \circ (f''_L, f'_N) = (z'_L \circ f''_L, z_N \circ f'_N) = (x_L, x_N)$$

and

$$(z'_L, z_N) \circ (g''_L, g'_N) = (z'_L \circ g''_L, z_N \circ g'_N) = (y_L, y_N)$$

Let $(z''_L, z''_N) : (L', K_3) \to (L, X)$ with

$$(z''_L, z''_N) \circ (f''_L, f'_N) = (x_L, x_N)$$

and

$$(z''_L, z''_N) \circ (g''_L, g'_N) = (y_L, y_N)$$

This implies $z''_N \circ f'_N = x_N$ and $z''_N \circ g'_N = y_N$ and hence $z''_N = z_N$. Then from Lemma 4.4 follows that $z''_L = z'_L$ and hence $(z''_L, z''_N) = (z'_L, z_N)$.
So we have that the commutative diagram (5) is a pushout. We know that for pushout (3) in **INet** there is $z_I(L) = L_3$.

$$\begin{tikzcd}
(L_0, K_0) \arrow[r, "{(f_L, f_N)}"] \arrow[d, "{(g_L, g_N)}"'] \arrow[dr, phantom, "\mathbf{(3)}"] & (L_1, K_1) \arrow[d, "{(f'_L, f'_N)}"] \\
(L_2, K_2) \arrow[r, "{(g'_L, g'_N)}"'] & (L_3, K_3)
\end{tikzcd}$$

From Fact 4.22 follows that

$$z'_I(L) = L' = L_3 = z_I(L)$$

Hence there is

$$(z'_N, z'_I) = (z_N, z_I)$$

(4) is pullback:

$$\begin{tikzcd}
INS_0 \arrow[dr, phantom, "\mathbf{(4)}"] & INS_1 \arrow[l, "f_I"'] \\
INS_2 \arrow[u, "g_I"] & INS_3 \arrow[l, "g'_I"] \arrow[u, "f'_I"']
\end{tikzcd}$$

We construct the pullback (6) in **SET**.

$$\begin{tikzcd}
INS_0 \arrow[dr, phantom, "\mathbf{(6)}"] & INS_1 \arrow[l, "f_I"'] \\
INS_2 \arrow[u, "g_I"] & INS_4 \arrow[l, "g''_I"] \arrow[u, "f''_I"']
\end{tikzcd}$$

Then by the construction of pullbacks in **SET** there is

$$INS_4 = \{(L_1, L_2) \in INS_1 \times INS_2 \mid f_I(L_1) = g_I(L_2)\}$$

We define a function $i : INS_4 \to INS_3$ with

$$i(L_1, L_2) = L_1 \circ_{(J,I)} L_2$$

where $J = f_I(L_1) = g_I(L_2)$. The function is well-defined because INS_3 contains exactly the pushouts of the elements in INS_4 and due to Fact 4.22 pushouts in **INet** are unique.
The function i is surjective because for every $L_3 \in INS_3$ there are $L_1 \in INS_1, L_2 \in INS_2$ with $f_I(L_1) = J = g_I(L_2)$ and $L_3 = L_1 \circ_{(J,I)} L_2$.
The function i is injective due to Fact 4.23.
Hence i is bijective which means that it is an isomorphism in **SET** and due to the uniqueness of pullbacks it follows that (4) is pullback in **SET**.

□

C.5 Composition of Algebraic High-Level Processes

Detailed Proof C.22 (Symmetry of Composability)
See Lemma 5.4.

Proof. We show that the corresponding conditions of the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) and (K_2, K_1) w.r.t. (I, i_2, i_1) are logically equivalent.

Condition 1:
Using the contraposition to both parts of the condition we obtain for $x \in P_I$:

$$\begin{aligned} & i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2) \\ \Leftrightarrow \quad & i_2(x) \notin IN(K_2) \Rightarrow i_1(x) \in IN(K_1) \end{aligned}$$

and

$$\begin{aligned} & i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2) \\ \Leftrightarrow \quad & i_2(x) \notin OUT(K_2) \Rightarrow i_1(x) \in OUT(K_1) \end{aligned}$$

leading to the fact that also the conjunction

$$i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2) \text{ and } i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$$

is equivalent to

$$i_2(x) \notin IN(K_2) \Rightarrow i_1(x) \in IN(K_1) \text{ and } i_2(x) \notin OUT(K_2) \Rightarrow i_1(x) \in OUT(K_1)$$

Condition 2:
The equality of the relations $\prec_{(i_1,i_2)}$ and $\prec_{(i_2,i_1)}$ follows from the commutativity of logical disjunction:

$$\begin{aligned} \prec_{(i_1,i_2)} &= \{(x,y) \in P_I \times P_I | i_1(x) <_{K_1} i_1(y) \vee i_2(x) <_{K_2} i_2(y)\} \\ &= \{(x,y) \in P_I \times P_I | i_2(x) <_{K_2} i_2(y) \vee i_1(x) <_{K_1} i_1(y)\} \\ &= \prec_{(i_2,i_1)} \end{aligned}$$

Since the induced causal relation $<_{(i_1,i_2)}$ is the transitive closure of $\prec_{(i_1,i_2)}$ and $<_{(i_2,i_1)}$ is the transitive closure of $\prec_{(i_2,i_1)}$ it follows that

$$<_{(i_1,i_2)} = <_{(i_2,i_1)}$$

So obviously $<_{(i_1,i_2)}$ is a finitary strict partial order if and only if $<_{(i_2,i_1)}$ is a finitary strict partial order.

Hence we have that both of the single conditions of the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) and (K_2, K_1) w.r.t. (I, i_2, i_1) are equivalent to each other implying that also the conjunctions of both are equivalent to each other. □

Detailed Proof C.23 (Composition of AHL-Occurrence Nets)
See Theorem 5.5.

Proof.

$$\begin{array}{ccc} I & \xrightarrow{i_1} & K_1 \\ {\scriptstyle i_2}\downarrow & (\mathbf{PO}) & \downarrow{\scriptstyle i'_1} \\ K_2 & \xrightarrow[i'_2]{} & K \end{array}$$

"⇒":

Given the AHL-occurrence nets K_1, K_2 and I and the pushout diagram (PO) in the category **AHLNet** as above. We have to show that the net K is an AHL-occurrence net, i.e. K is unary, it has no forward or backward conflicts, and the causal relation $<_K$ is a finitary strict partial order.

K is unary:

Let us assume that K is not unary, i.e. there are $p \in P_K$, $t \in T_K$ with

$$(term_1, p) \oplus (term_2, p) \leq pre_K(t)$$

or

$$(term_1, p) \oplus (term_2, p) \leq post_K(t)$$

Let $(term_1, p) \oplus (term_2, p) \leq pre_K(t)$.
Due to the pushout property there is $a \in \{1, 2\}$ and $t' \in T_{K_a}$ with

$$i'_{a,T}(t') = t$$

and

$$pre_K(t) = (id_{T_{OP}(X)} \otimes i'_{a,P})^{\oplus} \circ pre_{K_a}(t')$$

So the fact that $(term_1, p) \oplus (term_2, p) \leq pre_K(t)$ implies

$$(term'_1, p_1) \oplus (term'_2, p_2) \leq pre_{K_a}(t')$$

with
$$(id_{T_{OP}(X)} \otimes i'_{a,P})(term'_1, p_1) = (term_1, p)$$
and
$$(id_{T_{OP}(X)} \otimes i'_{a,P})(term'_2, p_2) = (term_2, p)$$
Since i_1 and i_2 are injective also i'_1 and i'_2 are injective, because (PO) is pushout and hence
$$i'_{a,P}(p_1) = p = i'_{a,P}(p_2)$$
leads to
$$p_1 = p_2$$
which means that K_a is not unary. This is a contradiction because K_a is an AHL-occurrence net.
The case $(term_1, p) \oplus (term_2, p) \leq post_K(t)$ works analogously.
Hence K is unary.

K has no forward conflicts:
Let us now assume that K has a forward conflict,
i.e. there is $p \in P_K, t_1 \neq t_2 \in T_K$ with
$$p \in \bullet t_1 \cap \bullet t_2$$

Case 1: for $a \in \{1,2\}$ there is $t_1, t_2 \in i_{a,T}(T_{K_a})$
Then we have $t'_1, t'_2 \in T_{K_a}$ with
$$i'_{a,T}(t'_1) = t_1$$
and
$$i'_{a,T}(t'_2) = t_2$$
and there is $p' \in P_{K_a}$ with
$$i'_{a,P}(p') = p$$
and
$$p' \in \bullet t'_1 \cap \bullet t'_2$$
because AHL-morphisms preserve pre conditions.
This means that K_a has a forward conflict which contradicts the fact that K_a is an AHL-occurrence net.

Case 2: $t_1 \in i_{1,T}(T_{K_1})$ and $t_2 \in i_{2,T}(T_{K_2})$
Then we have $t'_1 \in T_{K_1}, t'_2 \in T_{K_2}$ with
$$i'_{1,T}(t'_1) = t_1$$
and
$$i'_{2,T}(t'_2) = t_2$$
and since AHL-morphisms preserve pre conditions,
there are $p_1 \in P_{K_1}, p_2 \in P_{K_2}$ with
$$i'_{1,P}(p_1) = p,\ \ p_1 \in \bullet t'_1$$

and

$$i'_{2,P}(p_2) = p,\ p_2 \in \bullet t'_2$$

Due to the pushout property this implies $p_0 \in P_I$ with

$$i_{1,P}(p_0) = p_1$$

and

$$i_{2,P}(p_0) = p_2$$

The fact that $p_1 \in \bullet t'_1$ means that

$$i_{1,P}(p_0) \notin OUT(K_1)$$

which by the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) implies that

$$i_{2,P}(p_0) \in OUT(K_2)$$

contradicting the fact that

$$i_{2,P}(p_0) = p_2 \in \bullet t'_2$$

Hence K has no forward conflict.

K has no backward conflicts:
Let us now assume that K has a backward conflict,
i.e. there is $p \in P_K, t_1 \neq t_2 \in T_K$ with

$$p \in t_1 \bullet \cap t_2 \bullet$$

<u>*Case 1:*</u> *for $a \in \{1, 2\}$ there is $t_1, t_2 \in i_{a,T}(T_{K_a})$*
Then we have $t'_1, t'_2 \in T_{K_a}$ with

$$i'_{a,T}(t'_1) = t_1$$

and

$$i'_{a,T}(t'_2) = t_2$$

and there is $p' \in P_{K_a}$ with

$$i'_{a,P}(p') = p$$

and

$$p' \in t'_1 \bullet \cap t'_2 \bullet$$

because AHL-morphisms preserve post conditions.
This means that K_a has a backward conflict which contradicts the fact that K_a is an AHL-occurrence net.

<u>*Case 2:*</u> $t_1 \in i_{1,T}(T_{K_1})$ *and* $t_2 \in i_{2,T}(T_{K_2})$

Then we have $t'_1 \in T_{K_1}, t'_2 \in T_{K_2}$ with

$$i'_{1,T}(t'_1) = t_1$$

and

$$i'_{2,T}(t'_2) = t_2$$

and since AHL-morphisms preserve post conditions,
there are $p_1 \in P_{K_1}, p_2 \in P_{K_2}$ with

$$i'_{1,P}(p_1) = p,\ \ p_1 \in t'_1\bullet$$

and

$$i'_{2,P}(p_2) = p,\ \ p_2 \in t'_2\bullet$$

Due to the pushout property this implies $p_0 \in P_I$ with

$$i_{1,P}(p_0) = p_1$$

and

$$i_{2,P}(p_0) = p_2$$

The fact that $p_1 \in t'_1\bullet$ means that

$$i_{1,P}(p_0) \notin IN(K_1)$$

which by the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) implies that

$$i_{2,P}(p_0) \in IN(K_2)$$

contradicting the fact that

$$i_{2,P}(p_0) = p_2 \in t'_2\bullet$$

Hence K has no backward conflict.

$<_K$ *is finitary strict partial order*:

We have to show that $<_K$ is finitary and irreflexive.
Due to the fact that AHL-morphisms preserve pre and post conditions we obtain the causal relation of $<_K$ as the transitive closure of

$$\bigcup_{a\in\{1,2\}} \{(i'_a(x), i'_a(y)) \mid x, y \in P_{K_a} \uplus T_{K_a}, x <_{K_a} y\}$$

This means that elements which are causal related in K_1 or K_2 are also causal related in K. Additionally it is possible that elements in the net K are related due to the gluing of one or more elements.
Furthermore if for two interface elements $x_0, y_0 \in P_I$ the images of these elements are causal related in K, there is $x_0 <_{(i_1,i_2)} y_0$. We prove that fact because we need

it in the following.
Let $x_0, y_0 \in P_I$ with

$$i_1'(i_1(x_0)) <_K i_1'(i_1(y_0))$$

Then there is $a \in \{1, 2\}$ such that either there is

$$i_a(x_0) <_{K_a} i_a(y_0)$$

or there is $z_0 \in P_I$ with

$$i_a(x_0) <_{K_a} i_a(z_0)$$

and

$$i_1'(i_1(x_0)) <_K i_1'(i_1(z_0)) <_K i_1'(i_1(y_0))$$

This recursively leads to the fact that $x_0 <_{(i_1,i_2)} y_0$ because the induced causal relation is transitive.

$<_K$ *is irreflexive*:
Let us assume that $<_K$ is not irreflexive,
i.e. there exists $x \in P_K \uplus T_K$ s.t.

$$x <_K x$$

This means that there is a cycle in K and hence because of the structure of AHL-nets there exists $x' \in P_K \uplus T_K$ with

$$x <_K x' \text{ and } x' <_K x$$

Let us assume that there is no $z \in P_I$ with

$$x <_K i_1'(i_1(z)) <_K x'$$

i.e. the causal relation of x and x' is not the result of a gluing but is directly obtained from a causal relation in K_1 or K_2.
Then there is $a \in \{1, 2\}$ and $y, y' \in P_{K_a} \uplus T_{K_a}$ s.t.

$$i_a'(y) = x \text{ and } i_a'(y') = x'$$

and we have

$$y <_{K_a} y' \text{ and } y' <_{K_a} y$$

Due to the transitivity of $<_{K_a}$ there is $y <_{K_a} y$ which contradicts the fact that $<_{K_a}$ is irreflexive because K_a is an AHL-occurrence net.
So there is $z \in P_I$ with

$$x <_K i_1'(i_1(z)) <_K x'$$

Due to the transitivity of $<_K$ there is

$$i_1'(i_1(z)) <_K i_1'(i_1(z))$$

because

$$i_1'(i_1(z)) <_K x' <_K x <_K i_1'(i_1(z))$$

As shown above this implies

$$z <_{(i_1,i_2)} z$$

contradicting the fact that by the composability of K_1 and K_2 w.r.t. (I, i_1, i_2) the induced causal relation $<_{(i_1,i_2)}$ is irreflexive.
Hence $<_K$ is irreflexive.

$<_K$ *is finitary*:
We define $P(x)$ as the set of all predecessors of x, i.e.

$$P(x) = \{x' \mid x' <_K x\}$$

and $PI(x)$ as the set of all interface elements which have an image that is predecessor of x, i.e.

$$PI(x) = \{z \mid i'_1(i_1(z)) <_K x\}$$

For the finitarity of $<_K$ we have to show that for every $x \in P_K \uplus T_K$ there is a finite number $n \in \mathbb{N}$ such that $|P(x)| = n$.

Let us first assume that there is $x \in P_K \uplus T_K$ which has an infinite set of predecessors which are the images of interface elements, i.e. $PI(x)$ is an infinite set.
Then we also have that $P(x)$ is an infinite set because $i'_1(i_1(PI(x))) \subseteq P(x)$ and i_i, i'_1 are injective. Due to the finitarity of $<_{K_1}$ and $<_{K_2}$ there are finite many elements which have a causal relation to x directly obtained from the net K_1 or K_2.
So there is $y \in PI(x)$ where for $\tilde{x} = i'_1(i_1(y))$ there is $PI(\tilde{x})$ an infinite set.
For every $z \in PI(\tilde{x})$ there is $z <_{(i_1,i_2)} y$ implying that the induced causal relation $<_{(i_1,i_2)}$ is not finitary. This contradicts the fact that K_1 and K_2 are composable w.r.t. (I, i_1, i_2).
Thus for every $x \in P_K \uplus T_K$ there is $m \in \mathbb{N}$ such that $|PI(x)| \leq m$ which allows us to do a mathematical induction over m to show that for all $m \in \mathbb{N}$ the following property holds:
For every $x \in P_K \uplus T_K$ with $PI(x) \leq m$ there exists $n \in \mathbb{N}$ such that $|P(x)| = n$.

<u>induction basis:</u> $m = 0$

Let $x \in P_K \uplus T_K$ with $|PI(x)| \leq m = 0$.
Due to the pushout property i'_1 and i'_2 are jointly surjective and hence there is $a \in \{1, 2\}$ and $x' \in P_{K_a} \uplus T_{K_a}$ with

$$i'_a(x') = x$$

The fact that there is no element $y \in P_I$ with

$$i'_1(i_1(y)) <_K x$$

implies that the causal relation of all predecessors of x is directly derived from the net K_a, i.e. for every $z \in P_K \uplus T_K$ with

$$z <_K x$$

there is $z' \in P_{K_a} \uplus T_{K_a}$ with

$$i'_a(z') = z \text{ and } z' <_{K_a} x'$$

because AHL-morphisms preserve pre and post conditions.
Due to the finitarity of $<_{K_a}$ there is a finite number of predecessors $z' <_{K_a} x'$ and hence there is a finite number $n \in \mathbb{N}$ of $z \in P_K \uplus T_K$ with

$$z = i'_a(z') <_K i'_a(x') = x$$

i.e. $|P(x)| = n$.

induction hypothesis:
For $m \in \mathbb{N}$ for every $x \in P_K \uplus T_K$ with $|PI(x)| \leq m$ there exists $n \in \mathbb{N}$ such that $|P(x)| = n$.

induction step:
Let $x \in P_K \uplus T_K$ with $|PI(x)| \leq m + 1$.
Let us assume that x has an infinite number of predecessors, i.e. there is no $n \in \mathbb{N}$ such that $|P(x)| = n$.
As mentioned above in the induction basis there is a finite number of elements y with $y <_K x$ for which there is no $z \in PI(x)$ such that

$$y <_K i'_1(i_1(z))$$

Hence there is $z \in PI(x)$ such that $\tilde{x} = i'_1(i_1(z))$ has an infinite number of predecessors.
Due to the irreflexivity of $<_K$ there is $|PI(\tilde{x})| \leq m$ which by the induction hypothesis implies that there exists $\tilde{n} \in \mathbb{N}$ such that $|P(\tilde{x})| = \tilde{n}$ contradicting the fact that $\tilde{x}$ has an infinite number of predecessors.

So we have for $x \in P_K \uplus T_K$ with $m \in \mathbb{N}$ such that $|PI(x)| \leq m$ that there is a finite number $n \in \mathbb{N}$ of predecessors of x in K.
Let $m \in \mathbb{N}$ such that

$$m = \max_{x \in P_K \uplus T_K} |PI(x)|$$

Then we have that for every $x \in P_K \uplus T_K$ there is $|PI(x)| \leq m$ which implies that there is $n_x \in \mathbb{N}$ such that $|P(x)| = n_x$.
Hence $<_K$ is also finitary.

"$\Leftarrow$":
Let there be the pushout diagram (PO) such that K is an AHL-occurrence net.

$i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2)$:
Let $x \in P_I$ and

$$i_1(x) \notin IN(K_1)$$

Let us assume that

$$i_2(x) \notin IN(K_2)$$

Then $i_1(x)$ and $i_2(x)$ both are in the post domain of transitions, i.e. there are $t_1 \in T_{K_1}$ and $t_2 \in T_{K_2}$ such that

$$i_1(x) \in t_1\bullet$$

and

$$i_2(x) \in t_2\bullet$$

Since AHL-morphisms preserve post conditions there is

$$i'_1(i_1(x)) \in i'_1(t_1)\bullet$$

and

$$i'_2(i_2(x)) \in i'_1(t_2)\bullet$$

The fact that (PO) is pushout implies that it commutes and hence

$$i'_1(i_1(x)) = i'_2(i_2(x))$$

which implies

$$i'_1(i_1(x)) \in i'_1(t_1) \bullet \cap i'_2(t_2)\bullet$$

Due to the fact that K is an AHL-occurrence net it has no backward conflict implying that

$$i'_1(t_1) = i'_2(t_2)$$

So due to the pushout property there is $t_0 \in T_I$ with

$$i_1(t_0) = t_1 \text{ and } i_2(t_0) = t_2$$

But this contradicts the fact that $T_I = \emptyset$.
Hence there is $i_2(x) \in IN(K_2)$.

$i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$:
Let $x \in P_I$ and

$$i_1(x) \notin OUT(K_1)$$

Let us assume that

$$i_2(x) \notin OUT(K_2)$$

Then $i_1(x)$ and $i_2(x)$ both are in the pre domain of transitions, i.e. there are $t_1 \in T_{K_1}$ and $t_2 \in T_{K_2}$ such that

$$i_1(x) \in \bullet t_1$$

and

$$i_2(x) \in \bullet t_2$$

Since AHL-morphisms preserve pre conditions there is

$$i'_1(i_1(x)) \in \bullet i'_1(t_1)$$

and

$$i'_2(i_2(x)) \in \bullet i'_1(t_2)$$

The fact that (PO) is pushout implies that it commutes and hence

$$i'_1(i_1(x)) = i'_2(i_2(x))$$

which implies

$$i'_1(i_1(x)) \in \bullet i'_1(t_1) \cap \bullet i'_2(t_2)$$

Due to the fact that K is an AHL-occurrence net it has no forward conflict implying that

$$i'_1(t_1) = i'_2(t_2)$$

So due to the pushout property there is $t_0 \in T_I$ with

$$i_1(t_0) = t_1 \text{ and } i_2(t_0) = t_2$$

But this contradicts the fact that $T_I = \emptyset$.
Hence there is $i_2(x) \in OUT(K_2)$.

induced causal relation is a finitary strict partial order:
Let $x, y \in P_I$ with $x \prec_{(i_1,i_2)} y$.
Then there is

$$i_1(x) <_{K_1} i_1(y) \text{ or } i_2(x) <_{K_2} i_2(y)$$

and by the fact that $i'_1 \circ i_1 = i'_2 \circ i_2$ we have

$$i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)$$

because AHL-morphisms preserve pre and post conditions.
Since $<_K$ is transitive there is also for the transitive closure $<_{(i_1,i_2)}$ of $\prec_{(i_1,i_2)}$ that

$$x <_{(i_1,i_2)} y \Rightarrow i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)$$

Let us assume that $<_{(i_1,i_2)}$ is not finitary, i.e. there is $y \in P_I$ with an infinite set of predecessors

$$S = \{x \in P_I \mid x <_{(i_1,i_2)} y\}$$

leading to an infinite set

$$S' = \{x \in P_I \mid i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)\}$$

which contradicts the fact that $<_K$ is finitary.
Let us assume that $<_K$ is not irreflexive, i.e. there is $x \in P_I$ with $x <_{(i_1,i_2)} x$.
Then there is

$$i'_1 \circ i_1(x) <_K i'_1 \circ i_1(x)$$

contradicting the fact that $<_K$ is irreflexive.
So we have that the induced causal relation $<_{(i_1,i_2)}$ is finitary and irreflexive and hence it is a finitary strict partial order.

□

Detailed Proof C.24 (Sequential Composability of Instantiations)
See Fact 5.13.

Proof.

"$\Rightarrow$":

From Fact 5.3 follows that (K_1, K_2) are composable w.r.t. (I, i_1, i_2).
Then from the sequential composability of (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2) follows that for $(a, p) \in A \otimes P_I$ there is

$$\begin{aligned} & (a, i_1(p)) \in P_{L_{init_1}} \\ \Rightarrow\ & (a, i_1(p)) \in OUT(L_{init_1}) \\ \Rightarrow\ & (a, i_2(p)) \in IN(L_{init_2}) \\ \Rightarrow\ & (a, i_2(p)) \in P_{L_{init_2}} \end{aligned}$$

and hence (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2).

"$\Leftarrow$":

There is $OUT(L_{init_1}) \subseteq P_{L_{init_1}}$ and due to the sequential composability of (K_1, K_2) w.r.t. (I, i_1, i_2) for $p \in P_I$ there is $i_2(p) \in IN(K_2)$.
Let $a \in A_{type(i_2(p))}$ and $(a, i_2(p)) \in P_{L_{init_2}}$. Then due to the bijectivity of the projection $proj(K_2) \circ in_2$ there is

$$(a, i_2(p)) = (proj(K_2) \circ in)^{-1}(i_2(p))$$

which implies that $(a, i_2(p)) \in IN(L_{init_2})$. So we have for $(a, p) \in A \otimes P_I$:

$$\begin{aligned} & (a, i_1(p)) \in OUT(L_{init_1}) \\ \Rightarrow\ & (a, i_1(p)) \in P_{L_{init_1}} \\ \Rightarrow\ & (a, i_2(p)) \in P_{L_{init_2}} \\ \Rightarrow\ & (a, i_2(p)) \in IN(L_{init_2}) \end{aligned}$$

and since (K_1, K_2) are sequentially composable w.r.t. (I, i_1, i_2) this implies that (L_{init_1}, L_{init_2}) are sequentially composable w.r.t. (I, i_1, i_2).

□

Detailed Proof C.25 (Equivalence of Composability and Compatibility)
See Lemma 5.15

Proof. For $x \in \{1, 2\}$ we have that $(I_x, e_{x,H}, m_{x,H})$ is an epi-mono-factorization of i_x where

$$I_x = i_x(I)$$

and $e_{x,H}$ is an **AHLNet**-morphism with

$$e_{x,H,P}(p) = i_{x,P}(p)$$

and

$$e_{x,H,T} = \emptyset$$

and $m_{x,H}$ is an inclusion.
Let J_x and $m_{x,L}$ be the unique instantiation and morphism induced by $m_{x,H}$ s.t. (J_x, I_x) and $(m_{x,L}, m_{x,H})$ are in **INet**.

$$\begin{tikzcd}
 & & Skel(I) \arrow[dll, "Skel(i_1)"'] \arrow[dl, "Skel(e_{1,H})"] \arrow[dr, "Skel(e_{2,H})"'] \arrow[drr, "Skel(i_2)"] & & \\
Skel(K_1) & Skel(I_1) \arrow[l, "Skel(m_{1,H})"] & & Skel(I_2) \arrow[r, "Skel(m_{2,H})"'] & Skel(K_2) \\
L_1 \arrow[u, "proj(K_1)\circ in_1"'] & J_1 \arrow[l, "m_{1,L}"] \arrow[u, "proj(I_1)\circ inj_1"'] & & J_2 \arrow[r, "m_{2,L}"'] \arrow[u, "proj(I_2)\circ inj_2"] & L_2 \arrow[u, "proj(K_2)\circ in_2"]
\end{tikzcd}$$

composability $\Rightarrow$ *compatibility*:

Let (L_{init_1}, L_{init_2}) be composable w.r.t. (I, i_1, i_2).
Then we define

$$c := (c_H, c_L) : (J_1, I_1) \to (J_2, I_2)$$

with

$$c_H : I_1 \to I_2$$

$$c_{H,P}(e_{1,H,P}(p)) = e_{2,H,P}(p)$$

$$c_{H,T} = \emptyset$$

and

$$c_L : J_1 \to J_2$$

$$c_{L,P}(a, e_{1,H,P}(p)) = (a, e_{2,H,P}(p))$$

$$c_{L,T} = \emptyset$$

well-definedness of functions $c_{H,P}$, $c_{H,T}$, $c_{L,P}$ and $c_{L,T}$:

Due to $T_I = \emptyset$ we also have

$$T_{I_1} = i_{1,T}(T_I) = \emptyset$$

and

$$T_{J_1} = \emptyset$$

because $T_{J_1} \cong T_{I_1}$ which implies that $c_{H,T}$ and $c_{L,T}$ are well-defined functions.
Since $e_{1,H}$ is an epimorphism $e_{1,H,P}$ is surjective which implies that for every $p' \in P_{Skel(I_1)}$ there is $p \in P_{Skel(K_0)}$ with

$$p' = e_{1,H,P}(p)$$

and hence $c_{H,P}$ is a well-defined function.
It remains to show that $c_{L,P}$ is a well-defined function.
Since

$$P_{J_1} \subseteq A \otimes P_{I_1}$$

because J_1 is an instantiation of I_1, for every $x \in P_{J_1}$ there is a $p \in P_I$ and $a \in A_{type(e_{1,H,P}(p))}$ s.t.

$$x = (a, e_{1,H,P}(p))$$

Furthermore

$$e_{1,H,P}(p) = i_{1,P}(p)$$

means that

$$\begin{aligned} m_{1,L,P}(a, e_{1,H,P}(p)) &= m_{1,L,P}(a, i_{1,P}(p)) \\ &= (id_A \otimes m_{1,H,P})(a, i_{1,P}(p)) \\ &= (a, i_{1,P}(p)) \in P_{L_{init_1}} \end{aligned}$$

implying that

$$(a, i_{2,P}(p)) \in P_{L_{init_2}}$$

because of the composability of (L_{init_1}, L_{init_2}) w.r.t. (I, i_1, i_2).
The fact that

$$i_{2,P}(p) \in i_{2,P}(P_I) = P_{I_2}$$

with

$$\begin{aligned} m_{2,H,P}(i_{2,P}(p)) &= i_{2,P}(p) \\ &= proj(K_2) \circ in_2(a, i_{2,P}(p)) \end{aligned}$$

implies $p_0 \in P_{J_2}$ with

$$(proj(I_2) \circ inj_2)_P(p_0) = i_{2,P}(p)$$

and

$$m_{2,L}(p_0) = (a, i_{2,P}(p))$$

because by Theorem 2.33 and Lemma 4.5 diagram (1) is a pullback.

$$\begin{tikzcd} Skel(I_2) \arrow[r, "Skel(m_{2,H})"] & Skel(K_2) \\ J_2 \arrow[u, "proj(I_2) \circ inj_2"] \arrow[r, "m_{2,L}"'] \arrow[ur, phantom, "(1)"] & L_{init_2} \arrow[u, "proj(K_2) \circ in_2"'] \end{tikzcd}$$

The fact that

$$m_{2,L}(p_0) = (id_A \otimes m_{2,H,P})(a, i_{2,P}(p))$$

implies

$$\begin{aligned} p_0 &= (a, i_{2,P}(p)) \\ &= (a, e_{2,H,P}(p)) \in P_{J_2} \end{aligned}$$

Hence we have that also $c_{L,P}$ is a well-defined function.

well-definedness of morphisms c_H and c_L:
Since

$$T_{I_1} = T_{J_1} = \emptyset$$

the morphisms c_H and c_L obviously preserve pre and post conditions, i.e. c_H is a well-defined **AHLNet**-morphism and c_L is a well-defined **PTNet**-morphism.

c is **INet**-*morphism*:

There is

$$\begin{aligned}
(inj_2 \circ c_L)_P(a, e_{1,H,P}(p)) &= inj_{2,P} \circ c_{L,P}(a, e_{1,H,P}(p)) \\
&= inj_{2,P}(a, c_{H,P}(e_{1,H,P}(p))) \\
&= (a, c_{H,P}(e_{1,H,P}(p))) \\
&= Flat(c_H)_P(a, e_{1,H,P}(p)) \\
&= Flat(c_H)_P \circ inj_{1,P}(a, e_{1,H,P}(p)) \\
&= (Flat(c_H) \circ inj_1)_P(a, e_{1,H,P}(p))
\end{aligned}$$

and

$$\begin{aligned}
(inj_2 \circ c_L)_T &= \emptyset \\
&= (Flat(c_H) \circ inj_1)_T
\end{aligned}$$

which means that

$$inj_2 \circ c_L = Flat(c_H) \circ inj_1$$

Hence c is an **INet**-morphism.

The fact that

$$c_H \circ e_{1,H} = e_{2,H}$$

follows directly from the definition of c_H.
So (L_{init_1}, L_{init_2}) are compatible with (i_1, i_2).

compatibility $\Rightarrow$ *composability*:

Let (L_{init_1}, L_{init_2}) be compatible with (i_1, i_2).
Then there exists an **INet**-morphism $c = (c_H, c_L) : (J_1, I_1) \to (J_2, I_2)$ with $c_H \circ e_{1,H} = e_{2,H}$.

$$\begin{tikzcd}
 & Skel(I) \arrow[dl, "Skel(i_1)"'] \arrow[d, "Skel(e_{1,H})"] \arrow[dr, "Skel(e_{2,H})"] \arrow[drr, "Skel(i_2)"] & & \\
Skel(K_1) & Skel(I_1) \arrow[l, "Skel(m_{1,H})"] \arrow[r, "Skel(c_H)"] & Skel(I_2) \arrow[r, "Skel(m_{2,H})"'] & Skel(K_2) \\
L_1 \arrow[u, "proj(K_1)\circ in_1"'] & J_1 \arrow[l, "m_{1,L}"'] \arrow[u, "proj(I_1)\circ inj_1"'] \arrow[r, "c_L"'] & J_2 \arrow[u, "proj(I_2)\circ inj_2"] \arrow[r, "m_{2,L}"'] & L_2 \arrow[u, "proj(K_2)\circ in_2"]
\end{tikzcd}$$

We have to show that for all $(a, p) \in A \otimes P_I$:

$$(a, i_1(p)) \in P_{L_{init_1}} \Rightarrow (a, i_2(p)) \in P_{L_{init_2}}$$

Let $(a, p) \in A \otimes P_I$, i.e. $p \in P_I$ and $a \in A_{type_I(p)}$.
Then we have

$$i_1(p) \in i_1(P_I) = P_{I_1}$$

with
$$Skel(m_{1,H,P}(i_1(p))) = i_1(p) \in P_{K_1} = P_{Skel(K_1)}$$
For $(a, i_1(p)) \in P_{L_{init_1}}$ there is
$$proj(K_1) \circ in_1(a, i_1(p)) = i_1(p) \in P_{Skel(K_1)}$$
which implies that there is $p_0 \in P_{J_1}$ with
$$inj_P(p_0) = i_1(p)$$
and
$$m_{1,L,P}(p_0) = (a, i_1(p))$$
because by Theorem 2.33 and Lemma 4.5 diagram (2) is a pullback.

$$\begin{tikzcd}
Skel(I_1) \arrow[r, "Skel(m_{1,H})"] & Skel(K_1) \\
J_1 \arrow[u, "proj(I_1)\circ inj_1"] \arrow[r, "m_{1,L}"'] \arrow[ur, phantom, "(\mathbf{2})"] & L_{init_1} \arrow[u, "proj(K_1)\circ in_1"']
\end{tikzcd}$$

The fact that
$$\begin{aligned} m_{1,L,P}(p_0) &= (id_A \otimes m_{1,H,P})(p_0) \\ &= (a, i_1(p)) \end{aligned}$$
implies
$$p_0 = (a, i_1(p))$$
and therefore there is
$$\begin{aligned} c_{L,P}(a, i_{1,P}(p)) &= (a, c_{H,P}(i_{1,P}(p))) \\ &= (a, c_{H,P}(e_{1,H,P}(p))) \\ &= (a, e_{2,H,P}(p)) \\ &= (a, i_{2,P}(p)) \in P_{J_2} \end{aligned}$$
Since $m_{1,H}$ is an inclusion we obtain
$$\begin{aligned} m_{1,L,P}(a, i_2(p)) &= (a, m_{1,H,P}(i_2(p))) \\ &= (a, i_2(p)) \in P_{L_{init_2}} \end{aligned}$$
Hence (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2).

□

Detailed Proof C.26 (Retained Input Places)
See Lemma 5.20.

Proof. We show the equation in two steps.

$[i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2) \subseteq IN(K)$:

Let $x \in \{1,2\}$ and $p \in RI_x$, i.e. $\iota_x(p) \in RI_1 + RI_2$.
Then there is $p \in IN(K_x)$ and there is either $p \notin i_x(P_I)$ or $p \in i_x(BI)$.

Case 1: $p \notin i_x(P_I)$

This means that p is not glued to any other place. So the fact that there is no $t \in T_{K_x}$ with $p \in t\bullet$ implies that there is no $t' \in T_K$ with $i'_x(p) \in t'\bullet$ because AHL-morphisms preserve post conditions.
So we have that $i'_x(p) \in IN(K)$ and hence

$$\begin{aligned}[i'_1 \circ ri_1, i'_2 \circ ri_2](\iota_x(p)) &= i'_x \circ ri_x(p)\\ &= i'_x(p) \in IN(K)\end{aligned}$$

Case 2: $p \in i_x(BI)$

This means that there is $p_0 \in P_I$ with

$$i_x(p_0) = p$$

Furthermore there is $y \in \{1,2\}$ with $y \neq x$ and in the net K_y is a place $i_y(p_0)$ such that

$$i_y(p_0) \in IN(K_y)$$

and since (1) commutes there is

$$i'_x(i_x(p_0)) = i'_y(i_y(p_0))$$

So the fact that there is neither $t_x \in T_{K_x}$ with $p \in t_x\bullet$ nor $t_y \in T_{K_y}$ with $i_y(p_0) \in t_y\bullet$ implies that there is no $t \in T_K$ with

$$i'_x(p) = i'_y(i_y(p_0)) \in t\bullet$$

because AHL-morphisms preserve post conditions.
So also in this case we have

$$\begin{aligned}[i'_1 \circ ri_1, i'_2 \circ ri_2](\iota_x(p)) &= i'_x \circ ri_x(p)\\ &= i'_x(p) \in IN(K)\end{aligned}$$

$IN(K) \subseteq [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2)$:

Let $p \in IN(K)$.
Then there is no $t \in T_K$ with $p \in t\bullet$.

Case 1: *There is $p_1 \in P_{K_1}$ with $i'_1(p_1) = p$.*

Case 1.1: *There is $p_2 \in P_{K_2}$ with $i'_2(p_2) = p$.*

Let us assume that there is $t_1 \in T_{K_1}$ with $p_1 \in t_1\bullet$. The fact that AHL-morphims preserve post conditions implies

$$p = i'_1(p_1) \in i'_1(t_1)\bullet$$

contradicting the fact that there is no $t \in T_K$ with $p \in t\bullet$. Analogously the assumption of $t_2 \in T_{K_2}$ with $p_2 \in t_2\bullet$ leads to the same contradiction.
So we have that

$$p_1 \in IN(K_1) \text{ and } p_2 \in IN(K_2)$$

Due to the pushout property the fact that

$$i'_1(p_1) = p = i'_2(p_2)$$

implies $p_0 \in P_I$ with

$$i_1(p_0) = p_1 \text{ and } i_2(p_0) = p_2$$

and

$$p_0 \in BI$$

This implies

$$\begin{aligned} & p_1 \in RI_1 \text{ and } p_2 \in RI_2 \\ \Rightarrow \quad & \iota_1(p_1), \iota_2(p_2) \in RI_1 + RI_2 \end{aligned}$$

with

$$\begin{aligned} [i'_1 \circ ri_1, i'_2 \circ ri_2](\iota(p_1)) &= i'_1 \circ ri_1(p_1) \\ &= i'_1(p_1) \\ &= p \end{aligned}$$

i.e.

$$p \in [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2)$$

<u>*Case 1.2:*</u> *There is no $p_2 \in P_{K_2}$ with $i'_2(p_2) = p$.* This means that there is no $p_0 \in P_I$ with $i_1(p_0) = p_1$, i.e.

$$\begin{aligned} & p_1 \in IN(K_1) \setminus i_1(P_I) \\ \Rightarrow \quad & p_1 \in (IN(K_1) \setminus i_1(P_I)) \cup i_1(BI) \\ \Leftrightarrow \quad & p_1 \in RI_1 \end{aligned}$$

and hence there is $\iota_1(p_1) \in RI_1 + RI_2$ with

$$\begin{aligned} [i'_1 \circ ri_1, i'_2 \circ ri_2](\iota_1(p_1)) &= i'_1 \circ ri_1(p_1) \\ &= i'_1(p_1) \\ &= p \end{aligned}$$

i.e.

$$p \in [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2)$$

<u>*Case 2:*</u> *There is $p_2 \in P_{K_2}$ with $i'_2(p_2) = p$.*

<u>*Case 2.1:*</u> *There is $p_1 \in P_{K_1}$ with $i'_1(p_1) = p$.*
This case is equivalent to case 1.1.

<u>*Case 2.2:*</u> *There is no $p_1 \in P_{K_1}$ with $i'_1(p_1) = p$.*
This means that there is no $p_0 \in P_I$ with $i_2(p_0) = p_2$, i.e.

$$\begin{aligned} & p_2 \in IN(K_2) \setminus i_2(P_I) \\ \Rightarrow \quad & p_2 \in (IN(K_2) \setminus i_2(P_I)) \cup i_2(BI) \\ \Leftrightarrow \quad & p_2 \in RI_2 \end{aligned}$$

and hence there is $\iota_2(p_2) \in RI_1 + RI_2$ with

$$\begin{aligned}
[i'_1 \circ ri_1, i'_2 \circ ri_2](\iota_2(p_2)) &= i'_2 \circ ri_2(p_2) \\
&= i'_2(p_2) \\
&= p
\end{aligned}$$

i.e.

$$p \in [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2)$$

□

Detailed Proof C.27 (Retained Initializations)
See Lemma 5.21.

Proof. Let $x \in \{1,2\}$.

existence of bijection $r_x : RJ_x \to RI_x$

Due to the bijectivity of $(proj(K_x) \circ in_x)_P$ there is

$$\begin{aligned}
init_x &= IN(L_{init_x}) \\
&= (proj(K_x) \circ in_x)_P^{-1}(IN(Skel(K_x))) \\
&= (proj(K_x) \circ in_x)_P^{-1}(IN(K_x))
\end{aligned}$$

Furthermore due to the bijectivity of $proj(I) \circ in_I$ there is

$$\begin{aligned}
P_J &= (proj(I) \circ in_I)_P^{-1}(Skel(P_I)) \\
&= (proj(I) \circ in_I)_P^{-1}(P_I)
\end{aligned}$$

Since (j_1, i_1) and (j_2, i_2) are **INet**-morphism, diagram (5) and (6) commute

$$\begin{tikzcd}
P_J \arrow[r, "j_1"] \arrow[d, "proj(I)\circ in_I"'] \arrow[dr, phantom, "(\mathbf{5})"] & P_{L_{init_1}} \arrow[d, "proj(K_1)\circ in_1"] \\
P_{Skel(I)} \arrow[r, "Skel(i_1)"'] & P_{Skel(K_1)}
\end{tikzcd}
\qquad
\begin{tikzcd}
P_J \arrow[r, "j_2"] \arrow[d, "proj(I)\circ in_I"'] \arrow[dr, phantom, "(\mathbf{6})"] & P_{L_{init_2}} \arrow[d, "proj(K_2)\circ in_2"] \\
P_{Skel(I)} \arrow[r, "Skel(i_2)"'] & P_{Skel(K_2)}
\end{tikzcd}$$

and hence there is

$$\begin{aligned}
j_x(P_J) &= j_x((proj(I) \circ in_I)_P^{-1}(Skel(P_I))) \\
&= j_x \circ (proj(I) \circ in_I)_P^{-1}(Skel(P_I)) \\
&= (proj(K_x) \circ in_x)_P^{-1} \circ Skel(i_x)(Skel(P_I)) \\
&= (proj(K_x) \circ in_x)_P^{-1}(Skel(i_x)(Skel(P_I))) \\
&= (proj(K_x) \circ in_x)_P^{-1}(i_x(P_I))
\end{aligned}$$

and

$$\begin{array}{rcl} BJ & = & \{(a,p) \in P_J \mid j_1(a,p) \in IN(L_{init_1}) \text{ and } j_2(a,p) \in IN(L_{init_2})\} \\ & = & \{(proj(I) \circ in_I)_P^{-1}(p) \mid p \in P_I, j_{1,P}((proj(I) \circ in_I)_P^{-1}(p)) \in IN(L_{init_1}) \\ & & \text{ and } j_{2,P}((proj(I) \circ in_I)_P^{-1}(p)) \in IN(L_{init_2})\} \\ & = & \{(proj(I) \circ in_I)_P^{-1}(p) \mid p \in P_I, (proj(K_1) \circ in_1)_P^{-1}(i_{1,P}(p)) \in IN(L_{init_1}) \\ & & \text{ and } (proj(K_2) \circ in_2)_P^{-1}(i_{2,P}(p)) \in IN(L_{init_2})\} \\ & = & \{(proj(I) \circ in_I)_P^{-1}(p) \mid p \in P_I, i_{1,P}(p) \in IN(K_1) \\ & & \text{ and } i_{2,P}((p)) \in IN(K_2)\} \\ & = & (proj(I) \circ in_I)_P^{-1}(BI) \end{array}$$

So we have that

$$\begin{array}{rcl} RJ_x & = & (init_x \setminus j_x(P_J)) \cup j_x(BI) \\ & = & \{(a,p) \mid (a,p) \in (init_x \setminus j_x(P_J)) \cup j_x(BI)\} \\ & = & \{(proj(K_x) \circ in_x)_P^{-1}(p) \mid p \in (IN(K_x) \setminus i_x(P_I)) \cup i_x(BI)\} \\ & = & \{(proj(K_x) \circ in_x)_P^{-1}(ri_x(p)) \mid p \in RI_x\} \\ & = & (proj(K_x) \circ in_x)_P^{-1} \circ ri_x(RI_x) \end{array}$$

leading to epimorphism $e_x : RI_x \to RJ_x$ and an inclusion $m_x : RJ_x \to P_{L_{init_x}}$ with $e_x(p) = (proj(K_x) \circ in_x)_P^{-1} \circ ri_x(p)$ and diagram (7) commutes.

$$\begin{tikzcd} RI_x \arrow[rr, "(proj(K_x)\circ in_x)_P^{-1}\circ ri_x"] \arrow[dr, "e_x"'] & \mathbf{(7)} & P_{L_{init_x}} \\ & RJ_x \arrow[ur, "m_x"'] & \end{tikzcd}$$

Since $(proj(K_x) \circ in_x)_P$ is bijective and ri_x is an inclusion, $(proj(K_x) \circ in_x)_P^{-1} \circ ri_x$ is injective which by the decomposition of injective functions means that also e is injective. This implies that e is bijective because it is an epimorphism which means that it is surjective.
So there is a bijection $r_x : RJ_x \to RI_x$ with

$$\begin{array}{rcl} r_x(a,p) & = & e_x^{-1}(a,p) \\ & = & ((proj(K_x) \circ in_x)_P^{-1} \circ ri_x)^{-1}(a,p) \\ & = & ri_x^{-1} \circ ((proj(K_x) \circ in_x)_P^{-1})^{-1}(a,p) \\ & = & ri_x^{-1} \circ proj(K_x) \circ in_x(a,p) \\ & = & ri_x^{-1}(p) \\ & = & p \end{array}$$

$IN(L) = [j_1' \circ rj_1, j_2' \circ rj_2](RJ_1 + RJ_2)$:
For $r_1 : RJ_1 \to RI_1$ and $r_2 : RJ_2 \to RI_2$ there is

$$r_1 + r_2 : RJ_1 + RJ_2 \to RI_1 + RI_2$$

such that the back faces of the following cube are commuting.

$$\begin{tikzcd}
RJ_1 \arrow[rr, "\kappa_1"] \arrow[dr, "rj_1"] \arrow[dd, "r_1"'] & & RJ_1 + RJ_2 \arrow[dr, "{[j_1' \circ rj_1, j_2' \circ rj_2]}"] \arrow[dd, "r_1 + r_2"] & & RJ_2 \arrow[ll, "\kappa_2"'] \arrow[dr, "rj_2"] \arrow[dd, "r_2"] & \\
& P_{L_{init_1}} \arrow[rr, "j_1'"'] \arrow[dd, "proj(K_1) \circ in_1"] & & P_L \arrow[dd, "proj(K) \circ in"] & & P_{L_{init_2}} \arrow[ll, "j_2'"] \arrow[dd, "proj(K_2) \circ in_2"] \\
RI_1 \arrow[rr, "\iota_1"] \arrow[dr, "ri_1"'] & & RI_1 + RI_2 \arrow[dr, "{[i_1' \circ ri_1, i_2' \circ ri_2]}"'] & & RI_2 \arrow[ll, "\iota_2"'] \arrow[dr, "ri_2"] & \\
& P_{Skel(K_1)} \arrow[rr, "Skel(i_1')"'] & & P_{Skel(K)} & & P_{Skel(K_2)} \arrow[ll, "Skel(i_2')"]
\end{tikzcd}$$

The top and bottom faces of the cube commute and since (j_1', i_1') and (j_2', i_2') are **INet**-morphisms also the front faces commute.
Furthermore we have for $x \in \{1, 2\}$:

$$\begin{aligned}
proj(K_x) \circ in_x \circ rj_x(a,p) &= proj(K_x) \circ in_x(a,p) \\
&= p \\
&= ri_x(p) \\
&= ri_x \circ r_x(a,p)
\end{aligned}$$

i.e. also the side faces of the cube commute.
So for $(a,p) \in RJ_x$ we have

$$\begin{aligned}
& proj(K) \circ in \circ [j_1' \circ rj_1, j_2' \circ rj_2] \circ \kappa_x(a,p) \\
= \;& proj(K) \circ in \circ j_x' \circ rj_x(a,p) \\
= \;& Skel(i_x') \circ proj(K_x) \circ in_x \circ rj_x(a,p) \\
= \;& Skel(i_x') \circ ri_x \circ r_x(a,p) \\
= \;& [i_1' \circ ri_1, i_2' \circ ri_2] \circ \iota_x \circ r_x(a,p) \\
= \;& [i_1' \circ ri_1, i_2' \circ ri_2] \circ (r_1 + r_2) \circ \kappa_x(a,p)
\end{aligned}$$

which means that

$$\begin{aligned}
& [j_1' \circ rj_1, j_2' \circ rj_2](\kappa_x(a,p)) \\
= \;& (proj(K) \circ in)^{-1} \circ [i_1' \circ ri_1, i_2' \circ ri_2] \circ (r_1 + r_2)(\kappa_x(a,p))
\end{aligned}$$

This implies

$$[j_1' \circ rj_1, j_2' \circ rj_2] = (proj(K) \circ in)^{-1} \circ [i_1' \circ ri_1, i_2' \circ ri_2] \circ (r_1 + r_2)$$

because due to the universal property of coproducts κ_1 and κ_2 are jointly surjective. Since the coproduct of epimorphisms is an epimorphism the morphism $r_1 + r_2$ is an epimorphism because r_1 are r_2 are bijective which means that they are surjective and hence epimorphisms. So $r_1 + r_2$ is surjective which means

$$(r_1 + r_2)(RJ_1 + RJ_2) = RI_1 + RI_2$$

Due to the fact that

$$[i_1' \circ ri_1, i_2' \circ ri_2](RI_1 + RI_2) = IN(K)$$

we have

$$[i'_1 \circ ri_1, i'_2 \circ ri_2] \circ (r_1 + r_2)(RJ_1 + RJ_2) = IN(K)$$

Furthermore for the bijection $proj(K) \circ in$ there is

$$(proj(K) \circ in)^{-1}(IN(K)) = IN(L)$$

and hence

$$\begin{aligned} & [j'_1 \circ rj_1, j'_2 \circ rj_2](RJ_1 + RJ_2) \\ = \; & (proj(K) \circ in)^{-1} \circ [i'_1 \circ ri_1, i'_2 \circ ri_2] \circ (r_1 + r_2)(RJ_1 + RJ_2) \\ = \; & (proj(K) \circ in)^{-1} \circ [i'_1 \circ ri_1, i'_2 \circ ri_2](RI_1 + RI_2) \\ = \; & (proj(K) \circ in)^{-1}(IN(K)) \\ = \; & IN(L) \end{aligned}$$

□

Detailed Proof C.28 (Sequential Composability of AHL-Occurrence Nets with Instantiations)
See Fact 5.24

Proof. Let (K_1, K_2) be sequentially composable w.r.t. (I, i_1, i_2). By Fact 5.3 there are (K_1, K_2) composable w.r.t. (I, i_1, i_2).
Since $i_1(P_I) \subseteq OUT(K_1)$ and $i_2(P_I) \subseteq IN(K_2)$ there is for $x \in P_I$:

$$i_1(x) \in IN(K_1) \Rightarrow x \in BI$$

This implies that all input places of K_1 are retained input places:

$$RI_1 = (IN(K_1) \setminus i_1(P_I)) \cup i_1(BI) = IN(K_1)$$

and hence for $init_1 \in INIT_1$ there is

$$Ret_{(I,i_1,i_2)}(L_{init_1}) = RJ_1 = init_1$$

Due to the definition of $RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$ we can define a function

$$f : Composable_{(I,i_1,i_2)}(INS_1, INS_2) \to RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$$

because for every $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ such that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, i_1, i_2) there is exactly one set of retained initialization pairs $RetInit_{(I,i_1,i_2)}(L_{init_1}, L_{init_2})$.
The function f is surjective because for every $R \in RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$ there are composable (L_{init_1}, L_{init_2}) such that

$$R = RetInit_{(I,i_1,i_2)}(L_{init_1}, L_{init_2})$$

To show the injectivity of f let us assume

$$(L_{init_1}, L_{init_2}) \neq (L_{init_3}, L_{init_4}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

with

$$f(L_{init_1}, L_{init_2}) = RJ_1 + RJ_2 = f(L_{init_3}, L_{init_4})$$

which means that

$$RetInit_{(I,i_1,i_2)}(L_{init_1}, L_{init_2}) = RJ_1 + RJ_2 = RetInit_{(I,i_1,i_2)}(L_{init_3}, L_{init_4})$$

Then there is

$$init_1 = RJ_1 = init_3$$

and since KI_1 is an AHL-occurrence net with instantations there is a bijective correspondence between $INIT_1$ and INS_1 which means that

$$L_{init_1} = L_{init_3}$$

So $(L_{init_1}, L_{init_2}) \neq (L_{init_3}, L_{init_4})$ implies $L_{init_2} \neq L_{init_4}$, i.e. $init_2 \neq init_4$.
Furthermore it implies that the induced instantiation interfaces J, J' for both pairs of instantiations are equal. For $(a,p) \in A \otimes P_I$ there is

$$(a, i_1(p)) \in OUT(L_{init_1}) \Rightarrow (a, i_2(p)) \in IN(L_{init_2})$$

and

$$(a, i_1(p)) \in OUT(L_{init_1}) = OUT(L_{init_3}) \Rightarrow (a, i_2(p)) \in IN(L_{init_4})$$

which implies that for instantiation interface morphisms $j_2 : J \to L_{init_2}$ and $j_4 : J \to L_{init_4}$ there is

$$j_2(a,p) = (a, i_2(p)) = j_4(a,p)$$

and hence

$$j_2(P_J) = j_4(P_J) \text{ and } j_2(BJ) = j_4(BJ)$$

So due to the fact that

$$j_2(BJ) \subseteq j_2(P_J) \text{ and } j_4(BJ) \subseteq j_4(P_J)$$

and

$$j_2(P_J) \subseteq init_2 \text{ and } j_4(P_J) \subseteq init_4$$

there is

$$\begin{aligned} & (init_2 \setminus j_2(P_J)) \cup j_2(BJ) = RJ_2 = (init_4 \setminus j_4(P_J)) \cup j_4(BJ) \\ \Leftrightarrow\quad & init_2 \setminus (j_2(P_J) \setminus j_2(BJ)) = RJ_2 = init_4 \setminus (j_4(P_J) \setminus j_4(BJ)) \\ \Leftrightarrow\quad & init_2 = init_4 \end{aligned}$$

This is a contradiction to $init_2 \neq init_4$ and hence f is injective.
So we have that f is a bijection which means that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

$\square$

Detailed Proof C.29 (Composition of AHL-Occurrence Nets with Instantiations)
See Theorem 5.25.

Proof. Before we show the implications in both directions seperately we show the following facts because they are very helpful for both directions:

1. There is a bijection $f : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to INS$,
2. there is a surjection $g : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to \mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2)$ and
3. there is a bijection $h : \mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2) \to INIT$.

$$\begin{tikzcd}
\mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[r, "\sim h"] & INIT \\
\mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[u, "g"] \arrow[r, "\sim f"'] & INS
\end{tikzcd}$$

1. bijection $f : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to INS$:
Since for every $(L_{init_1}, L_{init_2}) \in INS_1 \times INS_2$ which are composable w.r.t. (I, i_1, i_2) there is a unique $L = L_{init_1} \circ_{(I,i_1,i_2)} L_{init_2}$, we obtain a well-defined function

$$f : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to INS$$

with

$$f(L_{init_1}, L_{init_2}) = L_{init_1} \circ_{(I,i_1,i_2)} L_{init_2}$$

f is surjective because for every $L \in INS$ there are

$$(L_{init_1}, L_{init_2}) \in \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2)$$

such that

$$L = L_{init_1} \circ_{(I,i_1,i_2)} L_{init_2}$$

Due to Lemma 5.19 f is also injective and hence it is bijective.

2. surjection $g : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to \mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2)$:
Furthermore we define a function

$$g : \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2) \to \mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2)$$

with

$$g(L_{init_1}, L_{init_2}) = \mathit{Ret}_{(I,i_1,i_2)}(L_{init_1}) \uplus \mathit{Ret}_{(I,i_1,i_2)}(L_{init_2})$$

The well-definedness of g follows directly from the definition of $\mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2)$.
The function g is surjective because for every $R \in \mathit{RetInit}_{(I,i_1,i_2)}(INS_1, INS_2)$ there are $(L_{init_1}, L_{init_2}) \in \mathit{Composable}_{(I,i_1,i_2)}(INS_1, INS_2)$ such that

$$\begin{aligned} R &= \mathit{Ret}_{(I,i_1,i_2)}(L_{init_1}) \uplus \mathit{Ret}_{(I,i_1,i_2)}(L_{init_2}) \\ &= g(L_{init_1}, L_{init_2}) \end{aligned}$$

3. *bijection* $h : RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \to INIT$:

$$h : RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \to INIT$$

with

$$h(g(L_{init_1}, L_{init_2})) = IN(f(L_{init_1}, L_{init_2}))$$

well-definedness:

Since g is surjective, for every $R \in RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$ there are

$$(L_{init_1}, L_{init_2}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

such that

$$R = g(L_{init_1}, L_{init_2})$$

So for the well-definedness of h it remains to show that for

$$(L_{init_1}, L_{init_2}), (L_{init_3}, L_{init_4}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

with

$$g(L_{init_1}, L_{init_2}) = g(L_{init_3}, L_{init_4})$$

there is

$$IN(f(L_{init_1}, L_{init_2})) = IN(f(L_{init_3}, L_{init_4}))$$

Let

$$L_{init} = L_{init_1} \circ_{(I,j_1,j_2)} L_{init_2} = f(L_{init_1}, L_{init_2})$$

and

$$L_{init'} = L_{init_3} \circ_{(I,j_3,j_4)} L_{init_4} = f(L_{init_3}, L_{init_4})$$

with composition diagrams of instantiations (2) and (3) w.r.t. composition diagram (1).

$$\begin{tikzcd} I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "i_1'"] \\ K_2 \arrow[r, "i_2'"'] & K \end{tikzcd} \quad \begin{tikzcd} J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(\mathbf{2})"] & L_{init_1} \arrow[d, "j_1'"] \\ L_{init_2} \arrow[r, "j_2'"'] & L_{init} \end{tikzcd} \quad \begin{tikzcd} J' \arrow[r, "j_1"] \arrow[d, "j_3"'] \arrow[dr, phantom, "(\mathbf{3})"] & L_{init_3} \arrow[d, "j_3'"] \\ L_{init_4} \arrow[r, "j_4'"'] & L_{init'} \end{tikzcd}$$

Furthermore let

$$g(L_{init_1}, L_{init_2}) = RJ_1 \uplus RJ_2$$

and

$$g(L_{init_3}, L_{init_4}) = RJ_3 \uplus RJ_4$$

together with inclusions $rj_x : RJ_x \to P_{L_{init_x}}$ for $x \in \{1, 2, 3, 4\}$ (see Lemma 5.21). For

$$g(L_{init_1}, L_{init_2}) = R = g(L_{init_3}, L_{init_4})$$

from Lemma 5.21 follows that

$$IN(L_{init}) = [j_1' \circ rj_1, j_2' \circ rj_2](R)$$

and

$$IN(L_{init'}) = [j'_3 \circ rj_3, j'_4 \circ rj_4](R)$$

Due to the fact that for $x \in \{1,2,3,4\}$ the nets L_{init_x} are instantiations there are inclusions $in_1 : L_{init_1} \to K_1$, $in_2 : L_{init_2} \to K_2$, $in_3 : L_{init_3} \to K_1$ and $in_4 : L_{init_4} \to K_2$. Furthermore due to the composition of instantiations there are inclusions $in : L_{init} \to K$ and $in' : L_{init'} \to K$ and (j'_1, i'_1),(j'_3, i'_1) are **INet**-morphisms.
Therefore we have

$$\begin{array}{rcl} j'_1 \circ rj_1(a,p) & = & j'_1(a,p) \\ & = & in \circ j'_1(a,p) \\ & = & Flat(i'_1) \circ in_1(a,p) \\ & = & Flat(i'_1)(a,p) \\ & = & Flat(i'_1) \circ in_3(a,p) \\ & = & in' \circ j'_3(a,p) \\ & = & j'_3(a,p) \\ & = & j'_3 \circ rj_3(a,p) \end{array}$$

and analogously $j'_2 \circ rj_2(a,p) = j'_4 \circ rj_4(a,p)$.
This implies for coproduct injections $\kappa_x : RJ_x \to R$ ($x \in \{1,2,3,4\}$) that

$$\begin{array}{rcl} [j'_1 \circ rj_1, j'_2 \circ rj_2] \circ \kappa_1(a,p) & = & j'_1 \circ rj_1(a,p) \\ & = & j'_3 \circ rj_3(a,p) \\ & = & [j'_3 \circ rj_3, j'_4 \circ rj_4] \circ \kappa_3(a,p) \end{array}$$

and

$$\begin{array}{rcl} [j'_1 \circ rj_1, j'_2 \circ rj_2] \circ \kappa_2(a,p) & = & j'_2 \circ rj_2(a,p) \\ & = & j'_4 \circ rj_4(a,p) \\ & = & [j'_3 \circ rj_3, j'_4 \circ rj_4] \circ \kappa_4(a,p) \end{array}$$

and hence

$$[j'_1 \circ rj_1, j'_2 \circ rj_2] = [j'_3 \circ rj_3, j'_4 \circ rj_4]$$

because due to the universal properties of coproducts κ_1, κ_2 and κ_3, κ_4 are jointly surjective. So we have that

$$IN(L_{init}) = [j'_1 \circ rj_1, j'_2 \circ rj_2](R) = [j'_3 \circ rj_3, j'_4 \circ rj_4](R) = IN(L_{init'})$$

i.e. h is a well-defined function.

surjectivity:
The function h is surjective because for every $init \in INIT$ there is $L_{init} \in INS$ with $init = IN(L_{init})$ and since f is surjective there are

$$(L_{init_1}, L_{init_2}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

such that

$$f(L_{init_1}, L_{init_2}) = L_{init}$$

and

$$h(g(L_{init_1}, L_{init_2})) = IN(L_{init}) = init$$

injectivity:

Let

$$(RJ_1 \uplus RJ_2), (RJ_3 \uplus RJ_4) \in RetInit_{(I,i_1,i_2)}(INS_1, INS_2)$$

with

$$h(RJ_1 \uplus RJ_2) = init = h(RJ_3 \uplus RJ_4)$$

and let

$$g(L_{init_1}, L_{init_2}) = RJ_1 \uplus RJ_2$$
$$g(L_{init_3}, L_{init_4}) = RJ_3 \uplus RJ_4$$

and

$$L = f(L_{init_1}, L_{init_2})$$
$$L' = f(L_{init_3}, L_{init_4})$$

with composition diagrams of instantiations (2) and (3) w.r.t. composition diagram (1).

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "i'_1"] \\
K_2 \arrow[r, "i'_2"'] & K
\end{tikzcd}
\quad
\begin{tikzcd}
J \arrow[r, "j_1"] \arrow[d, "j_2"'] \arrow[dr, phantom, "(\mathbf{2})"] & L_{init_1} \arrow[d, "j'_1"] \\
L_{init_2} \arrow[r, "j'_2"'] & L
\end{tikzcd}
\quad
\begin{tikzcd}
J' \arrow[r, "j_3"] \arrow[d, "j_4"'] \arrow[dr, phantom, "(\mathbf{3})"] & L_{init_3} \arrow[d, "j'_3"] \\
L_{init_4} \arrow[r, "j'_4"'] & L'
\end{tikzcd}$$

For

$$RI_1 = Ret_{(I,i_1,i_2)}(K_1)$$

and

$$RI_2 = Ret_{(I,i_1,i_2)}(K_2)$$

there are bijective functions

$$r_x : RJ_x \to RI_1 \text{ for } x \in \{1, 3\}$$

and

$$r_x : RJ_x \to RI_2 \text{ for } x \in \{2, 4\}$$

with $r_x(a, p) = p$ for $x \in \{1, 2, 3, 4\}$ (see Lemma 5.20) leading to bijections

$$s_1 = r_3^{-1} \circ r_1 : RJ_1 \to RJ_3$$

with $r_3^{-1} \circ r_1(a_1, p) = (a_3, p)$ and

$$s_2 = r_4^{-1} \circ r_2 : RJ_2 \to RJ_4$$

with $r_4^{-1} \circ r_2(a_2, p) = (a_4, p)$.

Due to Lemma 5.21 there is

$$[j'_1 \circ rj_1, j'_2 \circ rj_2](RJ_1 \uplus RJ_2) = init = [j'_3 \circ rj_3, j'_4 \circ rj_4](RJ_3 \uplus RJ_4)$$

leading for $x \in \{1,2,3,4\}$ to well-defined functions

$$t_x : RJ_x \to init$$

with

$$t_x(a,p) = j'_x \circ rj_x(a,p)$$

For inclusions $inc : init \to P_L$ given by the fact that $init \subseteq P_L$ and $inc' : init \to P_{L'}$ given by the fact that $init \subseteq P_{L'}$ we obtain the following diagrams where diagram (7) commutes.

$$\begin{tikzcd}
RJ_1 \arrow[dr, "r_1"] \arrow[dd, "s_1"'] \arrow[rrrrd, bend left, "t_1"] & & \mathbf{(5)} & L \arrow[dl, "proj(K)\circ in"'] & \\
& RI_1 \arrow[l, phantom, "\mathbf{(4)}"] \arrow[r, "i'_1\circ ri_1"] & P_K & \mathbf{(7)} & init \arrow[ul, "inc"'] \arrow[dl, "inc'"] \\
RJ_3 \arrow[ur, "r_3"'] \arrow[rrrru, bend right, "t_3"'] & & \mathbf{(6)} & L' \arrow[ul, "proj(K)\circ in'"] &
\end{tikzcd}$$

Diagram (4) commutes by the definition of s_1.
Furthermore there is

$$\begin{aligned}
proj(K) \circ in \circ inc \circ t_1(a,p) &= proj(K) \circ in \circ inc \circ j'_1 \circ rj_1(a,p) \\
&= proj(K) \circ in \circ inc \circ j'_1(a,p) \\
&= proj(K) \circ in \circ inc(a, i'_1(p)) \\
&= proj(K) \circ in(a, i'_1(p)) \\
&= i'_1(p) \\
&= i'_1 \circ ri_1(p) \\
&= i'_1 \circ ri_1 \circ r_1(a,p)
\end{aligned}$$

i.e. diagram (5) commutes and

$$\begin{aligned}
proj(K) \circ in' \circ inc' \circ t_3(a,p) &= proj(K) \circ in' \circ inc' \circ j'_3 \circ rj_3(a,p) \\
&= proj(K) \circ in' \circ inc' \circ j'_3(a,p) \\
&= proj(K) \circ in' \circ inc'(a, i'_1(p)) \\
&= proj(K) \circ in'(a, i'_1(p)) \\
&= i'_1(p) \\
&= i'_1 \circ ri_1(p) \\
&= i'_1 \circ ri_1 \circ r_3(a,p)
\end{aligned}$$

i.e. diagram (6) commutes implying that

$$\begin{aligned} proj(K) \circ in \circ inc \circ t_1 &= i'_1 \circ ri_1 \circ r_1 \\ &= i'_1 \circ ri_1 \circ r_3 \circ s_1 \\ &= proj(K) \circ in' \circ inc' \circ t_3 \circ s_1 \\ &= proj(K) \circ in \circ inc \circ t_3 \circ s_1 \end{aligned}$$

i.e. diagram (8) and analogously (9) commutes because $proj(K) \circ in$ is an isomorphism in **SET** which means that it is a monomorphism and hence $proj(K) \circ in \circ inc$ is a monomorphism because inc is an inclusion which means that it is a monomorphism.

$$\begin{tikzcd} RJ_1 \arrow[r, "t_1"] \arrow[d, "s_1"'] \arrow[dr, phantom, "(\mathbf{8})"] & init \\ RJ_3 \arrow[ur, bend right, "t_3"'] & \end{tikzcd} \qquad \begin{tikzcd} RJ_2 \arrow[r, "t_2"] \arrow[d, "s_2"'] \arrow[dr, phantom, "(\mathbf{9})"] & init \\ RJ_4 \arrow[ur, bend right, "t_4"'] & \end{tikzcd}$$

Let $x \in \{1, 2\}$ and $y \in \{3, 4\}$ with $y = x + 2$ and let

$$(a_x, p) \in RJ_x$$

and

$$s_x(a_x, p) = (a_y, p) \in RJ_y$$

which means that

$$((a_x, p), x) \in RJ_1 \uplus RJ_2$$

and

$$((a_y, p), x) \in RJ_3 \uplus RJ_4$$

Then we have

$$\begin{aligned} (a_x, i'_x(p)) &= j'_x(a_x, p) \\ &= j'_x \circ rj_x(a_x, p) \\ &= t_x(a_x, p) \\ &= t_y \circ s_x(a_x, p) \\ &= t_y(a_y, p) \\ &= j'_y \circ rj_y(a_y, p) \\ &= j'_y(a_y, p) \\ &= (a_y, i'_x(p)) \end{aligned}$$

and hence we have

$$a_x = a_y$$

implying that for

$$((a_x, p), x) \in RJ_1 \uplus RJ_2$$

there is

$$(s_1 + s_2)((a_x, p), x) = ((a_y, p), x) = ((a_x, p), x) \in RJ_3 \uplus RJ_4$$

which means that $s_1 + s_2 : RJ_1 \uplus RJ_2 \to RJ_3 \uplus RJ_4$ is an inclusion.
Due to the fact that s_1 and s_2 are bijective they are epimorphisms which implies that $s_1 + s_2$ is an epimorphism, i.e. it is surjective.
This means that $s_1 + s_2$ is the identity and hence

$$RJ_1 \uplus RJ_2 = RJ_3 \uplus RJ_4$$

i.e. h is injective which together with the fact that h is surjective means that h is bijective.

Now we have everything that we need to proof the required implications.

"$\Rightarrow$":
We have to show

1. K is an AHL-occurrence net,
2. all $init \in INIT$ are initializations of K,
3. all $L_{init} \in INS$ are instantiations of K and
4. there is a bijection $b : INIT \to INS$.

Part 1: Follows directly from Theorem 5.5.

Part 2: Follows directly from the definition of $INIT$.

Part 3: Follows directly from Theorem 5.16.

Part 4: The fact that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

implies a bijection

$$c : RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \to Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

which we can use to define a function

$$b = f \circ c \circ h^{-1} : INIT \to INS$$

b is bijective because c, f and h are bijective functions.

$$\begin{tikzcd}
RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[r, "h"] \arrow[d, "c"] \arrow[dr, phantom, "(=)"] & INIT \arrow[d, "b"] \\
Composable_{(I,i_1,i_2)}(INS_1, INS_2) \arrow[r, "f"'] & INS
\end{tikzcd}$$

"$\Leftarrow$":
Let $KI = (K, INIT, INS)$ be an AHL-occurrence net with instantiations.
We have to show that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

Since KI is an AHL-occurrence net with instantiations there is a bijection

$$b : INIT \to INS$$

which we can use to define a function

$$c : RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \to Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

with

$$c = f^{-1} \circ b \circ h$$

c is bijective because b, f and h are bijective functions and hence

$$RetInit_{(I,i_1,i_2)}(INS_1, INS_2) \cong Composable_{(I,i_1,i_2)}(INS_1, INS_2)$$

which means that

$$|RetInit_{(I,i_1,i_2)}(INS_1, INS_2)| = |Composable_{(I,i_1,i_2)}(INS_1, INS_2)|$$

□

Detailed Proof C.30 (Composition of AHL-Occurrence Nets with Instantiations is Pushout) See Corollary 5.28.

Proof.

Part 1:

For $K = K_1 \circ_{(I,i_{1,N},i_{2,N})} K_2$ there is (2) pushout in **AHLNet**.

$$\begin{tikzcd} I \arrow[r, "i_{1,N}"] \arrow[d, "i_{2,N}"'] \arrow[dr, phantom, "(\mathbf{2})"] & K_1 \arrow[d, "i'_{1,N}"] \\ K_2 \arrow[r, "i'_{2,N}"'] & K \end{tikzcd}$$

Due to Theorem 4.18 for $L_{init_1} \in INS_1, L_{init_2} \in INS_2$ there is a common preimage

$$PreIns(i_{1,N})(L_{init_1}) = PreIns(i_{2,N}(L_{init_2}))$$

of L_{init_1} and L_{init_2} iff (L_{init_1}, L_{init_2}) are compatible with $(i_{1,N}, i_{2,N})$ which by Lemma 5.15 is equivalent to the fact that (L_{init_1}, L_{init_2}) are composable w.r.t. $(I, i_{1,N}, i_{2,N})$. Therefore there is

$$\begin{aligned} INS &= \{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in Composable_{(I,i_{1,N},i_{2,N})}(INS_1, INS_2), \\ &\quad\ (J, j_1, j_2) \text{ is induced instantiation interface w.r.t. } (I, i_{1,N}, i_{2,N})\} \\ &= \{L \mid (L_{init_1}, L_{init_2}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2) \\ &\quad\ \text{ and (3) is pushout in } \mathbf{INet}\} \\ &= \{L \mid L_{init_1} \in INS_1, L_{init_2} \in INS_2 \text{ with} \\ &\quad\ PreIns(i_{1,N})(L_{init_1}) = PreIns(I_{2,N}(L_{init_2})) \\ &\quad\ \text{ and (3) is pushout in } \mathbf{INet}\} \\ &= \{L \mid L_{init_1} \in INS_1, L_{init_2} \in INS_2 \text{ with } i_{1,I}(L_{init_1}) = i_{2,I}(L_{init_2}) \\ &\quad\ \text{ and (3) is pushout in } \mathbf{INet}\} \end{aligned}$$

$$\begin{tikzcd}
(L_0, I) \arrow[r, "{(i_{1,L}, i_{1,N})}"] \arrow[d, "{(i_{2,L}, i_{2,N})}"'] \arrow[dr, phantom, "(\mathbf{3})"] & (L_{init_1}, K_1) \arrow[d, "{(i'_{1,L}, i'_{1,N})}"] \\
(L_{init_2}, K_2) \arrow[r, "{(i'_{1,L}, i'_{2,N})}"'] & (L, K)
\end{tikzcd}$$

which by Fact 4.27 implies that (1) is pushout in **AHLNetI** where

$$i'_{1,I}(L_3) = PreIns(i'_{1,N})(L_3) \text{ and } i'_{2,I}(L_3) = PreIns(i'_{2,N})(L_3)$$

The existence of the required interface follows from the well-defined interface in Definition 5.27.

Part 2:
Given pushout (1) in **AHLNetI** there is pushout (2) in **AHLNet** which means that $K = K_1 \circ_{(I, i_{1,N}, i_{2,N})} K_2$.

$$\begin{tikzcd}
I \arrow[r, "i_{1,N}"] \arrow[d, "i_{2,N}"'] \arrow[dr, phantom, "(\mathbf{2})"] & K_1 \arrow[d, "i'_{1,N}"] \\
K_2 \arrow[r, "i'_{2,N}"'] & K
\end{tikzcd}$$

Due to the uniqueness of instantiation preimages for $L_1 \in INS_1$ there is $i_{1,I}(L_1) = PreIns(i_{1,N})(L_1)$ and for $L_2 \in INS_2$ there is $i_{2,I}(L_2) = PreIns(i_{2,N})(L_2)$. Therefore $i_{1,I}(L_1) = i_{2,I}(L_2)$ means that $PreIns(i_{1,N})(L_1) = PreIns(i_{1,N})(L_2)$. Then from Theorem 4.18 follows that (L_1, L_2) are compatible with $(i_{1,N}, i_{2,N})$ which by Lemma 5.15 is equivalent to the fact that (L_1, L_2) are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.
Therefore due to Fact 4.27 and the definition of the composition of instantiations there is

$$\begin{aligned}
INS &= \{L \mid L_{init_1} \in INS_1, L_{init_2} \in INS_2 \text{ with } i_{1,I}(L_{init_1}) = i_{2,I}(L_{init_2}) \\
&\qquad \text{and (3) is pushout in } \mathbf{INet}\} \\
&= \{L \mid L_{init_1} \in INS_1, L_{init_2} \in INS_2 \text{ with} \\
&\qquad PreIns(i_{1,N})(L_{init_1}) = PreIns(I_{2,N}(L_{init_2})) \\
&\qquad \text{and (3) is pushout in } \mathbf{INet}\} \\
&= \{L \mid (L_{init_1}, L_{init_2}) \in Composable_{(I, i_1, i_2)}(INS_1, INS_2) \\
&\qquad \text{and (3) is pushout in } \mathbf{INet}\} \\
&= \{L_{init_1} \circ_{(J, j_1, j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in Composable_{(I, i_{1,N}, i_{2,N})}(INS_1, INS_2), \\
&\qquad (J, j_1, j_2) \text{ is induced instantiation interface w.r.t. } (I, i_{1,N}, i_{2,N})\}
\end{aligned}$$

$$\begin{tikzcd}
(L_0, I) \arrow[r, "{(i_{1,L}, i_{1,N})}"] \arrow[d, "{(i_{2,L}, i_{2,N})}"'] \arrow[dr, phantom, "(\mathbf{3})"] & (L_{init_1}, K_1) \arrow[d, "{(i'_{1,L}, i'_{1,N})}"] \\
(L_{init_2}, K_2) \arrow[r, "{(i'_{1,L}, i'_{2,N})}"'] & (L, K)
\end{tikzcd}$$

From the fact that (1) is also pushout in **AHLONetI** follows that K_1, K_2 and K are AHL-occurrence nets with instantiations and hence we have for

$$INIT = \{IN(L) \mid L \in INS\}$$

that $KI = (K, INIT, INS)$ is the composition $KI = KI_1 \circ_{(I,i_{1,N},i_{2,N})} KI_2$.

$\square$

Detailed Proof C.31 (Pushout of AHL-Processes with Instantiations)
See Corollary 5.34.

Proof. "$\Rightarrow$":
Given pushout (1) in **AHLNetI** and also in **AHLProcI**(**AN**). Then (K, INS) is an AHL-occurrence net with instantiations which means that the commuting diagram (1) is also a commuting diagram in **AHLONetI**.
Let (X, INS_X) be an AHL-occurrence net with instantiations together with **AHLONetI**-morphisms $x_1 : (K_1, INS_1) \to (X, INS_X)$ and $x_2 : (K_2, INS_2) \to (X, INS_X)$ such that

$$x_1 \circ i_1 = x_2 \circ i_2$$

Since **AHLONetI** is a subcategory of **AHLNetI** due to pushout property of (1) in **AHLNetI** there is a unique morphism $x : (K, INS) \to (X, INS_X)$ with $x \circ i'_1 = x_1$ and $x \circ i'_2 = x_2$. **AHLONetI** is a full subcategory of **AHLNetI** which means that morphism x between AHL-occurrence nets with instantiations (K, INS) and (X, INS_X) is an **AHLONetI**-morphism. Hence (1) is also pushout in **AHLONetI** which by Corollary 5.29 implies that (KI_1, KI_2) are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.
From the required commutativities of morphisms in the category **AHLProcI**(**AN**) we obtain

$$mp_{1,N} \circ i_{1,N} = mp_{0,N} = mp_{2,N} \circ i_{2,N}$$

and hence $(mp_{1,N}, mp_{2,N})$ are composable w.r.t. $(I, i_{1,N}, i_{2,N})$.

"$\Leftarrow$":
Given pushout (1) in **AHLNetI** and $(mp_{1,N}, mp_{2,N})$ are composable w.r.t. $(I, i_{1,N}, i_{2,N})$. Then there is also (KI_1, KI_2) composable w.r.t. $(I, i_{1,N}, i_{2,N})$ which by Corollary 5.29 implies that (1) is also pushout in **AHLONetI**.
From Fact 4.27 and the fact that (1) is pushout in **AHLNetI** follows that (2) is pushout in **AHLNet**. Then the composability of (mp_1, mp_2) w.r.t. $(I, i_{1,N}, i_{2,N})$ implies that there exist $mp_{0,N} : I \to AN$ and $mp_N : K \to AN$ such that (2) is pushout in **AHLProc**(**AN**) due to Corollary 5.11.

$$\begin{tikzcd}
I \arrow[r, "i_1"] \arrow[d, "i_{2,N}"'] \arrow[dr, phantom, "(2)"] \arrow[ddrr, bend right=60, "mp_{0,N}"'] & K_1 \arrow[d, "i'_{1,N}"] \arrow[ddr, bend left, "mp_{1,N}"] & \\
K_2 \arrow[r, "i'_{2,N}"'] \arrow[drr, bend right, "mp_{2,N}"] & K \arrow[dr, "mp_N"] & \\
 & & AN
\end{tikzcd}$$

Furthermore let $mp_{0,I} : \emptyset \to INS_I$, $mp_{1,I} : \emptyset \to INS_1$, $mp_{2,I} : \emptyset \to INS_2$ and $mp : \emptyset \to INS$ be empty functions.
Then obviously $mp_0 = (mp_{0,N}, mp_{0,I})$, $mp_1 = (mp_{1,N}, mp_{1,I})$, $mp_2 = (mp_{2,N}, mp_{2,I})$ and $mp = (mp_N, mp_I)$ are **AHLNetI**-morphisms and i_1, i_2 are **AHLProcI**(**AN**) morphisms because $i_{1,N}$ and $i_{2,N}$ are **AHLProc**(**AN**)-morphisms and the corresponding functions commute due to initiality of $\emptyset$ in **SET**.
The fact that

$$mp_1 \circ i_1 = mp \circ i'_1 \circ i_1 = mp \circ i'_2 \circ i_2 = mp_2 {\circ}_2$$

implies that mp is the unique morphism with $mp \circ i'_1 = mp_1$ and $mp \circ i'_2 = mp_2$ because (1) is pushout in **AHLNetI**.
Let $mp_X : (X, INS_X) \to (AN, \emptyset)$ be an instantiated AHL-process together with **AHLProcI**(**AN**)-morphisms $x_1 : (K_1, INS_1) \to (X, INS_X)$ and $x_2 : (K_2, INS_2) \to (X, INS_X)$ such that

$$x_1 \circ i_1 = x_2 \circ i_2$$

Since this means that (X, INS_X) and x_1, x_2 are in **AHLNetI** the pushout property of (1) in **AHLNetI** implies a unique **AHLNetI** morphism $x : (K, INS) \to (X, INS_X)$ with $x \circ i'_1 = x_1$ and $x \circ i'_2 = x_2$.
Then we have

$$mp_X \circ x \circ i'_1 = mp_X \circ x_1 = mp_1$$

and

$$mp_X \circ x \circ i'_2 = mp_X \circ x_2 = mp_2$$

which due to the uniqueness of mp implies that $mp_X \circ x = mp$, i.e. x is an **AHLProcI**(**AN**)-morphism. Hence (1) is a pushout in **AHLProcI**(**AN**).

□

C.6 Decomposition of Algebraic High-Level Processes

Detailed Proof C.32 (Uniqueness of Maximal Pushout Complements)
See Fact 6.3.

Proof. *Part 1*:
Since (C', f'_2, g'_2) is a maximal pushout complement of $A \xrightarrow{g_1} B \xrightarrow{f_1} D$ in **C** there is a unique morphism $x : C \to C'$ such that

$$x \circ g_2 = g'_2 \text{ and } f'_2 \circ x = f_2$$

$$\begin{tikzcd}
 & A \arrow[r, "f_1"] \arrow[d, "f_2"] \arrow[ddl, bend right, "f'_2"'] & B \arrow[d, "g_1"] \\
 & C \arrow[r, "g_2"'] \arrow[dl, "x"'] & D \\
C' \arrow[urr, bend right, "g'_2"'] & &
\end{tikzcd}$$

Furthermore $id_{C'} : C' \to C'$ is the unique morphism such that

$$id_{C'} \circ g'_2 = g'_2 \text{ and } f'_2 \circ id_{C'} = f'_2$$

Since (C, f_2, g_2) is a maximal pushout complement of $A \overset{g_1}{\to} B \overset{f_1}{\to} D$ in $\mathbf{C}$ there is a unique morphism $x' : C' \to C$ such that

$$x' \circ g'_2 = g_2 \text{ and } f_2 \circ x' = f'_2$$

$$\begin{tikzcd}
& A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[ddl, bend right, "f'_2"'] & B \arrow[d, "g_1"] \\
& C \arrow[r, "g_2"'] & D \\
C' \arrow[ur, "x'"] \arrow[urr, bend right, "g'_2"'] & &
\end{tikzcd}$$

Furthermore $id_C : C \to C$ is the unique morphism such that

$$id_C \circ g_2 = g_2 \text{ and } f_2 \circ id_C = f_2$$

Then we have

$$f_2 \circ x' \circ x = f'_2 \circ x = f_2 = f_2 \circ id_C$$

and

$$x' \circ x \circ g_2 = x' \circ g'_2 = g_2 = id_C \circ g_2$$

which due to the uniqueness of id_C implies that $x' \circ x = id_C$.
Analogously we obtain

$$f'_2 \circ x \circ x' = f_2 \circ x' = f'_2 = f'_2 \circ id_{C'}$$

and

$$x \circ x' \circ g'_2 = x \circ g_2 = g'_2 = id_{C'} \circ g'_2$$

which due to the uniqueness of $id_{C'}$ implies that $x \circ x' = id_{C'}$.
Hence x and x' are isomorphisms.

Part 2:

$$\begin{tikzcd}
& A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[ddl, bend right, "f'_2"'] & B \arrow[d, "g_1"] \\
& C \arrow[r, "g_2"'] & D \\
C' \arrow[ur, "x"] \arrow[urr, bend right, "g'_2"'] & &
\end{tikzcd}$$

To show the uniqueness of x let $x' : C \to C'$ be a morphism such that

$$g_2' \circ x' = g_2 \text{ and } x' \circ f_2 = f_2'$$

Then there is

$$g_2' \circ x' = g_2 = g_2' \circ x$$

which due to the fact that g_2' is a monomorphism implies that $x' = x$.
Hence (C', f_2', g_2') is a maximal pushout complement of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$.

□

Detailed Proof C.33 (Pushout Complement of AHL-Nets)
See Fact 6.6

Proof. "$\Rightarrow$":
Let us assume that the dangling condition is not satisfied. This means that there is dangling point which is no gluing point, i.e. there is a transition $t \in T_K \setminus g_{1,T}(T_{K_1})$ and a place $p \in P_{K_1} \setminus f_{1,P}(P_I)$ together with a term $term \in T_{OP}(X)_{type(p)}$ such that

$$(term, g_{1,P}(p)) \leq pre_K(t) \oplus post_K(t)$$

The fact that

$$t \notin g_{1,T}(T_{K_1})$$

implies

$$t \in T_{K_2}$$

and the fact that

$$p \notin f_{1,P}(P_I)$$

implies

$$g_{1,P}(p) \notin g_{1,P}(f_{1,P}(P_I))$$

and hence

$$g_{1,P}(p) \notin P_{K_2}$$

So we have

$$\begin{aligned} & (term, g_{1,P}(p)) \leq pre_K(t) \oplus post_K(t) \\ \Rightarrow\ & term, g_{1,P}(p)) \leq pre_K|_{T_{K_2}}(t) \oplus post_K|_{T_{K_2}}(t) \\ \Leftrightarrow\ & term, g_{1,P}(p)) \leq pre_{K_2}(t) \oplus post_{K_2}(t) \end{aligned}$$

which means that K_2 has a dangling arc because $g_{1,P}(p) \notin P_{K_2}$ and hence K_2 is not a well-defined AHL-net.

"$\Leftarrow$":
Let the dangling condition be satisfied.

well-definedness of AHL-net K_2:
For the well-definedness of K_2 we have to show that the functions $pre_{K_2}, post_{K_2}, cond_{K_2}$ and $type_{K_2}$ are well-defined.

well-definedness of $type_{K_2}$:
Let $SP = (S, OP, E; X)$. Then for every $p \in P_{K_2}$ there has to be $type_{K_2}(p) \in S$. That is true because for $p \in P_{K_2}$ there is

$$type_{K_2}(p) = type_K|_{P_{K_2}}(p) = type_K(p) \in S$$

well-definedness of pre_{K_2}:
For every $t \in T_{K_2}$ and $(term, p) \leq pre_{K_2}(t)$ there has to be $p \in P_{K_2}$ and $term \in T_{OP}(X)_{type(p)}$.
A place $p \in P_K$ is also in P_{K_2} if for the case that there is $p' \in P_{K_1}$ with $g_{1,P}(p') = p$ there is $p' \in f_{1,P}(P_I)$.
Let us assume that there is a place $p' \in P_{K_1}$ with $g_{1,P}(p') = p$ and let $(term, p) \leq pre_{K_2}(t)$. Then there is

$$\begin{aligned} & (term, p) \leq pre_{K_2}(t) \\ \Leftrightarrow\ & (term, p) \leq pre_K|_{T_{K_2}}(t) \\ \Rightarrow\ & (term, p) \leq pre_K(t) \end{aligned}$$

From the fact that $t \in T_{K_2}$ follows that $t \in T_K \setminus g_{1,T}(T_{K_1})$ and hence p' is a dangling point.
Due to the dangling condition p' is a gluing point, i.e. $p' \in f_{1,P}(P_I)$ and therefore $p \in P_{K_2}$.
The fact that $term \in T_{OP}(X)_{type(p)}$ follows from the fact that $(term, p) \leq pre_K(t)$ and the well-definedness of pre_K.

well-definedness of $post_{K_2}$:
The proof for $post_{K_2}$ works analogously to the one for pre_{K_2}.

well-definedness of $cond_{K_2}$:
The well-definedness of $cond_{K_2}$ follows from the fact that K_2 has the same specification part SP as the net K and $cond_K$ is well-defined.

well-definedness of f_2:
The well-definedness of the function $f_{2,P}$ and follows from the well-definedness of $f_{1,P}$ and $g_{1,T}$. The well-definedness of $f_{2,T}$ and the fact that f_2 is an AHL-morphism follows from $T_I = \emptyset$.

well-definedness of g_2:
The inclusions $g_{2,P}$ and $g_{2,T}$ are well-defined functions because $P_{K_2} \subseteq P_K$ and $T_{K_2} \subseteq T_K$.
It remains to show that g_2 is an AHL-morphism which is trivial because the pre, post, cond and type functions are restrictions of the corresponding functions in K to the set of transitions in K_2.
So we have for example

$$\begin{aligned} pre_K \circ g_{2,T}(t) &= pre_K(t) \\ &= pre_K|_{T_{K_2}}(t) \\ &= pre_{K_2}(t) \\ &= g_{2,P}^{\oplus} \circ pre_{K_2}(t) \end{aligned}$$

The proof for the other functions works analogously.

diagram (1) *is pushout in* **AHLNet**:

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

diagram (1) *commutes*:
For $p \in P_I$ there is

$$g_{1,P} \circ f_{1,P}(p) = f_{2,P}(p) = g_{2,P} \circ f_{2,P}(p)$$

and there is

$$g_{1,T} \circ f_{1,T} = \emptyset = g_{2,T} \circ f_{2,T}$$

definition of morphism $x : K \to X$:
Let X be an AHL-net and $x_1 : K_1 \to X$, $x_2 : K_2 \to X$ two AHL-morphisms with $x_1 \circ f_1 = x_2 \circ f_2$.
We define a morphism $x : K \to X$ in the following way:

$$x = (x_P, x_T)$$

with

$$x_P(p) = \begin{cases} x_{1,P}(p') & \text{, if there exists } p' \in P_{K_1} : g_{1,P}(p') = p; \\ x_{2,P}(p') & \text{, if there exists } p' \in P_{K_2} : g_{2,P}(p') = p. \end{cases}$$

and

$$x_T(t) = \begin{cases} x_{1,T}(t') & \text{, if there exists } t' \in T_{K_1} : g_{1,T}(t') = t; \\ x_{2,T}(t') & \text{, if there exists } t' \in T_{K_2} : g_{2,T}(t') = t. \end{cases}$$

well-definedness of x_P *and* x_T:
We have to show that for every $p \in P_K$, and every $t \in T_K$ respectively, there is one of the cases true.
Let $p \in P_K$ and let us assume that none of the cases is true, i.e. there is no $p_1 \in P_{K_1}$ with $g_{1,P}(p_1) = p$ and there is no $p_2 \in P_{K_2}$ with $g_{2,P}(p_2) = p$.
Then there is $p \in P_K \setminus g_{1,P}(P_{K_1})$ implying that $p \in P_{K_2}$ with $g_{2,P}(p) = p$. This is a contradiction to the assumption that none of the cases is true.
For $t \in T_K$ we obtain analogously that $t \notin g_{1,T}(T_{K_1})$ implies that $t \in g_{2,T}(T_{K_2})$.
Since the cases in the definitions of x_P resp. x_T are not necessarily disjoint there is to show that the cases in the definitions of x_P and x_T do not contradict each other.
Let $p \in P_K$ such that there is $p_1 \in P_{K_1}$ with $g_{1,P}(p_1) = p$ and there is $p_2 \in P_{K_2}$ with $g_{2,P}(p_2) = p$. Since g_2 is an inclusion we have $p_2 = p$.
So the fact that $p \in P_{K_2}$ by the definition of P_{K_2} implies that there is $p_0 \in P_I$ with $f_{1,P}(p_0) = p_1$. Furthermore there is $f_{2,P}(p_0) = p_2$ because (1) commutes.

Finally from $x_1 \circ f_1 = x_2 \circ f_2$ this implies

$$\begin{array}{rcl} x_{1,P}(p_1) & = & x_{1,P}(f_{1,P}(p_0)) \\ & = & x_{2,P}(f_{2,P}(p_0)) \\ & = & x_{2,P}(p_2) \end{array}$$

which means that x_P is a well-defined function.
Let $t \in T_K$ such that there is $t_1 \in T_{K_1}$ with $g_{1,T}(t_1) = t$ and there is $t_2 \in T_{K_2}$ with $g_{2,T}(t_2) = t$. Since g_2 is an inclusion we have $t_2 = t$, i.e. $t \in T_{K_2} = T_K \setminus g_{1,T}(T_{K_1})$ contradicting the fact that $t = g_{1,T}(t_1)$. Hence the cases in the definition of x_T are disjoint and x_T is a well-defined function.

well-definedness of x:
We have to show that x is an AHL-morphism.
Let $t \in T_K$.

Case 1: There exists $t' \in T_{K_1}$ with $g_{1,T}(t') = t$.
From the fact that g_1 is an AHL-morphism which preserves pre conditions we obtain

$$\begin{array}{rcl} pre_K(t) & = & pre_K(g_{1,T}(t)) \\ & = & (id_{T_{OP}(X)} \otimes g_{1,P})^{\oplus}(pre_{K_1}(t')) \end{array}$$

which means that for all places p in the pre domain of t there is $p' \in P_{K_1}$ with $g_{1,P}(p') = p$.
Therefore we have

$$\begin{array}{rl} & (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ pre_K(t) \\ = & (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ pre_K \circ g_{1,T}(t') \\ = & (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ (id_{T_{OP}(X)} \otimes g_{1,P})^{\oplus} \circ pre_{K_1}(t') \\ = & (id_{T_{OP}(X)} \otimes (x_P \circ g_{1,P}))^{\oplus} \circ pre_{K_1}(t') \\ = & (id_{T_{OP}(X)} \otimes (x_{1,P} \circ g_{1,P}))^{\oplus} \circ pre_{K_1}(t') \\ = & pre_X(x_{1,T} \circ g_{1,T}(t')) \\ = & pre_X(x_T(t')) \end{array}$$

The proof for the post conditions works analogously.
For the conditions we obtain

$$\begin{array}{rcl} cond_X \circ x_T(t) & = & cond_X \circ x_{1,T}(t') \\ & = & cond_{K_1}(t') \\ & = & cond_K \circ g_{1,T}(t') \\ & = & cond_K(t) \end{array}$$

Case 2: There exists $t' \in T_{K_1}$ with $g_{1,T}(t') = t$.
This case is completely analogous to case 1 due to the symmetrical definition of x and the fact that x_2 and g_2 are AHL-morphisms as well as x_1 and g_1.

Let $p \in P_K$ and let $p' \in P_{K_1}$ with $p = g_{1,P}(p')$. Then we have

$$\begin{aligned} type_X \circ x_P(p) &= type_X \circ x_{1,P}(p') \\ &= type_{K_1}(p') \\ &= type_K \circ g_{1,P}(p') \\ &= type_K(p) \end{aligned}$$

The case that there is $p' \in P_{K_2}$ with $p = g_{2,P}(p')$ works analogously.
Hence x is a well-defined AHL-morphism.

universal property:
We have to show the commutativity of (2) and (3) which follows directly from the definition of x.

$$\begin{tikzcd}
& K_1 \arrow[dr, "g_1"] \arrow[drr, bend left, "x_1"] \arrow[drr, phantom, "\mathbf{(2)}"] & & \\
I \arrow[ur, "f_1"] \arrow[dr, "f_2"'] & \mathbf{(1)} & K \arrow[r, "x"] & X \\
& K_2 \arrow[ur, "g_2"'] \arrow[urr, bend right, "x_2"'] \arrow[urr, phantom, "\mathbf{(3)}"] & &
\end{tikzcd}$$

uniqueness of x:
Let $x' : K \to X$ with $x' \circ g_1 = x_1$ and $x' \circ g_2 = x_2$.
As mentioned above in the proof of the well-definedness of x_P and x_T for every element in K there is a preimage in K_1 or K_2.
Let $p \in P_K$.

<u>*Case 1:*</u> There is p' in P_{K_1} with $p = g_{1,P}(p')$.
Then there is

$$\begin{aligned} x'_P(p) &= x'(g_{1,P}(p')) \\ &= x_{1,P}(p') \\ &= x_P(g_{1,P}(p')) \\ &= x_P(p) \end{aligned}$$

<u>*Case 2:*</u> There is p' in P_{K_2} with $p = g_{2,P}(p')$.
Then there is

$$\begin{aligned} x'(p)_P &= x'(g_{2,P}(p')) \\ &= x_{2,P}(p') \\ &= x_P(g_{2,P}(p')) \\ &= x_P(p) \end{aligned}$$

So we have that $x'_P = x_P$. The proof for $x'_T = x_T$ works completely analogously due to the analogous definition of x_T.

K_2 is unique up to isomorphism:
The proof of the uniqueness of pushout complements in **AHLNet** can be found in [EEPT06].

□

Detailed Proof C.34 (Decomposition of AHL-Occurrence Nets)
See Theorem 6.8

Proof. "$\Rightarrow$":

Let K_2 be an AHL-occurrence net such that $K = K_1 \circ_{(I,f_1,f_2)} K_2$. Due to the definition of the composition of AHL-occurrence nets diagram (1) is a pushout in **AHLNet** and hence by Fact 6.6 the dangling condition is satisfied.

"$\Leftarrow$":

Let the dangling condition be satisfied and let the AHL-net K_2 together with morphisms f_2 and g_2 as defined in Fact 6.6. Since g_2 is an AHL-morphism and K is an AHL-occurrence net from Fact 2.14 follows that K_2 is an AHL-occurrence net.
We have to show that K_1 and K_2 are composable w.r.t. (I, f_1, f_2).
Therefore we have to show that

1. f_2 is injective,
2. for every $x \in P_I$:
   $f_1(x) \notin IN(K_1) \Rightarrow f_2(x) \in IN(K_2)$ and
   $f_1(x) \notin OUT(K_1) \Rightarrow f_2(x) \in OUT(K_2)$ and
3. the gluing relation $<_{(f_1,f_2)}$ is a finitary strict partial order.

Part 1:

The empty function $f_{1,T} = \emptyset$ is injective. Furthermore the fact that f_1 and g_1 are injective implies that $f_{1,P}$ and $g_{1,P}$ are injective and hence $f_{2,P}$ with

$$f_{2,P}(p) = g_{1,P} \circ f_{1,P}(p)$$

is injective. Thus f_2 is an injective morphism since its components are injective functions.

Part 2:

$f_1(x) \notin IN(K_1) \Rightarrow f_2(x) \in IN(K_2)$:

Let $x \in P_I$ with $f_{1,P}(x) \notin IN(K_1)$.
Let us assume that

$$f_{2,P}(x) \notin IN(K_2)$$

i.e. there is $t \in T_{K_2}$ with $f_{2,P}(x) \in t\bullet$.
Then there is $t = g_{2,T}(t) \in T_K$ and by the definition of T_{K_2} there is

$$t \notin g_{1,T}(T_{K_1})$$

The fact that $f_{1,P}(x) \notin IN(K_1)$ implies $t' \in T_{K_1}$ with

$$f_{1,P}(x) \in t'\bullet$$

Then there is

$$g_{1,P}(f_{1,P}(x)) \in g_{1,T}(t')\bullet$$

since AHL-morphisms preserve post conditions.
Due to the pushout property there is

$$g_{1,P}(f_{1,P}(x)) = g_{2,P}(f_{2,P}(x))$$

and since the inclusion g_2 is an AHL-morphism there is

$$g_{1,P}(f_{1,P}(x)) = g_{2,P}(f_{2,P})(x) \in t\bullet$$

From the fact that $t \notin g_{1,T}(T_{K_1})$ follows that $t \neq g_{1,T}(t')$ and hence

$$g_{1,P}(f_{1,P}(x)) \in g_{1,T}(t') \bullet \ \cap t\bullet$$

implies that K has a backward conflict which contradicts the fact that K is an AHL-occurrence net.
Hence our assumption is wrong and there is $f_{2,P}(x) \in IN(K_2)$.

$f_1(x) \notin OUT(K_1) \Rightarrow f_2(x) \in OUT(K_2)$:
Let $x \in P_I$ with $f_{1,P}(x) \notin OUT(K_1)$.
Let us assume that

$$f_{2,P}(x) \notin OUT(K_2)$$

i.e. there is $t \in T_{K_2}$ with $f_{2,P}(x) \in \bullet t$.
Then there is $t = g_{2,T}(t) \in T_K$ and by the definition of T_{K_2} there is

$$t \notin g_{1,T}(T_{K_1})$$

The fact that $f_{1,P}(x) \notin OUT(K_1)$ implies $t' \in T_{K_1}$ with

$$f_{1,P}(x) \in \bullet t'$$

Then there is

$$g_{1,P}(f_{1,P}(x)) \in \bullet g_{1,T}(t')$$

since AHL-morphisms preserve pre conditions.
Due to the pushout property there is

$$g_{1,P}(f_{1,P}(x)) = g_{2,P}(f_{2,P}(x))$$

and since the inclusion g_2 is an AHL-morphism there is

$$g_{1,P}(f_{1,P}(x)) = g_{2,P}(f_{2,P}((x))) \in \bullet t$$

From the fact that $t \notin g_{1,T}(T_{K_1})$ follows that $t \neq g_{1,T}(t')$ and hence

$$g_{1,P}(f_{1,P}(x)) \in \bullet g_{1,T}(t') \cap \bullet t$$

implies that K has a forward conflict which contradicts the fact that K is an AHL-occurrence net.
Hence our assumption is wrong and there is $f_{2,P}(x) \in OUT(K_2)$.

Part 3:

Let $x, y \in P_I$ with $x \prec_{(I,f_1,f_2)} y$.
Then there is $f_1(x) <_{K_1} f_1(y)$ or $f_2(x) <_{K_2} f_2(y)$ and by the fact that $g_1 \circ f_1 = g_2 \circ f_2$ we have

$$g_1 \circ f_1(x) <_K g_1 \circ f_1(y)$$

because AHL-morphisms preserve pre and post conditions.
Since $<_K$ is transitive there is also for the transitive closure $<_{(f_1,f_2)}$ of $\prec_{(I,f_1,f_2)}$ that

$$x <_{(f_1,f_2)} y \Rightarrow g_1 \circ f_1(x) <_K g_1 \circ f_1(y)$$

Let us assume that $<_{(f_1,f_2)}$ is not finitary, i.e. there is $y \in P_I$ with an infinite set of predecessors

$$S = \{x \in P_I \mid x <_{(f_1,f_2)} y\}$$

leading to an infinite set

$$S' = \{x \in P_I \mid g_1 \circ f_1(x) <_K g_1 \circ f_1(y)\}$$

which contradicts the fact that $<_K$ is finitary.
Let us assume that $<_{(f_1,f_2)}$ is not irreflexive, i.e. there is $x \in P_I$ with $x <_{(f_1,f_2)} x$.
Then there is

$$g_1 \circ f_1(x) <_K g_1 \circ f_1(x)$$

contradicting the fact that $<_K$ is irreflexive.
So we have that the gluing relation $<_{(f_1,f_2)}$ is finitary and irreflexive and hence it is a finitary strict partial order.

Hence we have that (K_1, K_2) are composable w.r.t. (I, f_1, f_2) and since K_2 is constructed as pushout complement there is

$$K = K_1 \circ_{(I,f_1,f_2)} K_2$$

due to the uniqueness of pushouts.

□

Detailed Proof C.35 (Cominimal Decomposition of AHL-Occurrence Nets with Instantiations)
See Theorem 6.17

Proof.

"$\Rightarrow$":

Let KI be decomposable w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$.

Then the decomposition K_2 of K w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ exists leading to composition diagram (1)

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(1)"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

The AHL-occurrence net with instantiations KI_2 as defined above exists because

$$IN : PreIns(g_2)(INS) \to PreInit(g_2)(INS)$$

is injective. This follows from Theorem 4.13.
It remains to show that the preimage construction leads to a cominimal decomposition of KI.
Therefore it is to show that

- $INS =$
  $\{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in Composable_{(I,f_1,f_2)}(INS_1, INS_2),$
  (J, j_1, j_2) is induced instantiation interface w.r.t. $(I, f_1, f_2)\}$,
- $INIT = \{IN(L) \mid L \in INS\}$, and
- for every $L_{init_2} \in INS_2$ there is $L_{init_1} \in INS_1$ s.t. (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2)

Part 1:
We show the equation in two steps by showing the subset relation in both directions.
"$\subseteq$":
We have to show that for every $L_{init} \in INS$ there are $L_{init_1} \in INS_1$ and $L_{init_2} \in INS_2$ such that L_{init_1} and L_{init_2} are composable w.r.t. (I, f_1, f_2) and L_{init} is the composition of L_{init_1} and L_{init_2}.
Let $L_{init} \in INS$. We choose

$$L_{init_1} = g_{1,I}(L_{init})$$

and

$$L_{init_2} = PreIns(g_2)(L_{init})$$

L_{init_1} and L_{init_2} are composable w.r.t. (I, f_1, f_2) if for all $(a, p) \in A \otimes P_I$ there is

$$(a, f_1(p)) \in P_{L_{init_1}} \Rightarrow (a, f_2(p)) \in P_{L_{init_2}}$$

So let $(a, p) \in A \otimes P_I$ and $(a, f_1(p)) \in P_{L_{init_1}}$.
For $L_{init_1} = g_{1,I}(L_{init})$ there is a **PTNet**-moprhism $g_{1,L} : L_{init_1} \to L_{init}$ such that $(g_{1,L}, g_1)$ is an **INet**-morphism implying that there is

$$g_{1,L}(a, f_1(p)) = (a, g_1(f_1(p))) \in P_{L_{init}}$$

Due to the pushout property there is $g_1 \circ f_1 = g_2 \circ f_2$ and hence

$$(a, g_2(f_2(p))) \in P_{L_{init}}$$

By Lemma 4.7 there is

$$(a, f_2(p)) \in P_{L_{init_2}}$$

i.e. (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2).
Therefore by Theorem 4.18 there is a unique induced instantiation interface (J, j_1, j_2) w.r.t. (I, f_1, f_2).

We show that $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$.
We obtain pushout (1) by the fact that $K = K_1 \circ_{(I,f_1,f_2)} K_2$ leading to pushout (2) because $Flat$ preserves pushouts.

$$\begin{tikzcd}
I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\
K_2 \arrow[r, "g_2"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
Flat(I) \arrow[r, "Flat(f_1)"] \arrow[d, "Flat(f_2)"'] \arrow[dr, phantom, "(\mathbf{2})"] & Flat(K_1) \arrow[d, "Flat(g_1)"] \\
Flat(K_2) \arrow[r, "Flat(g_2)"'] & Flat(K)
\end{tikzcd}$$

Furthermore there are **INet**-morphisms (j_1, f_1) and (j_2, f_2) by theorem 4.18 implying that the back faces of the following cube are pullbacks due to the uniqueness of induced instantiation preimages (Lemma 4.5).

$$\begin{tikzcd}
& J \arrow[rr, "j_1"] \arrow[dl, "j_2"'] \arrow[dd, hook, "in_I"] \arrow[dr, phantom, "(\mathbf{3})"] & & L_{init_1} \arrow[dl, "g_{1,L}"'] \arrow[dd, hook, "in_1"] \\
L_{init_2} \arrow[rr, "g_{2,L}"'] \arrow[dd, hook, "in_2"] & & L_{init} \arrow[dd, hook, "in"] & \\
& Flat(I) \arrow[rr, "Flat(f_1)"] \arrow[dl, "Flat(f_2)"'] \arrow[dr, phantom, "(\mathbf{2})"] & & Flat(K_1) \arrow[dl, "Flat(g_1)"] \\
Flat(K_2) \arrow[rr, "Flat(g_2)"'] & & Flat(K) &
\end{tikzcd}$$

$Flat(f_1)$ is a monomorphism because f_1 is injective and hence a monomorphism and $Flat$ preserves monomorphisms. So together with the fact that all the vertical morphisms are inclusions and hence monomorphisms we have that (2) is a weak Van Kampen square.
By Lemma 4.5 also the front faces of the cube are pullbacks which implies that the top face (3) is a pushout.
Thus by Theorem 5.16 there is $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$.
"$\supseteq$":
Let

$$(L_{init_1}, L_{init_2}) \in Composable_{(I,f_1,f_2)}$$

and let

$$L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$$

where (J, j_1, j_2) is the instantiation interface induced by (I, f_1, f_2).
We have to show that $L_{init} \in INS$.
By Lemma 5.15 the composability of (L_{init_1}, L_{init_2}) w.r.t. (I, f_1, f_2) is equivalent to the fact that (L_{init_1}, L_{init_2}) are compatible with (f_1, f_2).
From Theorem 4.18 and the compatibility of L_{init_1} and L_{init_2} follows that

$$PreIns(f_1)(L_{init_1}) = L_{init_0} = PreIns(f_2)(L_{init_2})$$

Furthermore there is an **INet**-morphism $(f_{2,L}, f_2) : (L_{init_0}, I) \to (L_{init_2}, K_2)$.
Due to the construction of INS_2 there is $L_{init'} \in INS$ such that

$$L_{init_2} = PreIns(g_2)(L_{init'})$$

implying an **INet**-morphism $(g_{2,L}, g_2) : (L_{init_2}, K_2) \to (L_{init'}, K)$. Let

$$L_{init'_1} = g_{1,I}(L_{init})$$

implying an **INet**-morphism $(g'_{1,L}, g_1) : (L_{init'_1}, K_1) \to (L_{init'}, K)$ and let

$$L_{init'_0} = PreIns(f_1)(L_{init'_1})$$

implying an **INet**-morphism $(f'_{1,L}, f_1) : (L_{init'_0}, I) \to (L_{init'_1}, K_1)$.
Since (1) is pushout the diagram (1) commutes. So we have that

$$\begin{aligned}(g'_{1,L}, g_1) \circ (f'_{1,L}, f_1) &= (g'_{1,L} \circ f'_{1,L}, g_1 \circ f_1) \\ &= (g'_{1,L} \circ f'_{1,L}, g_2 \circ f_2)\end{aligned}$$

which together with

$$(g_{2,L}, g_2) \circ (f_{2,L}, f_2) = (g_{2,L} \circ f_{2,L}, g_2 \circ f_2)$$

by Lemma 4.5 implies that $L_{init_0} = L_{init'_0}$, i.e.

$$\begin{aligned}PreIns(f_1)(L_{init_1}) &= PreIns(f_1)(L_{init'_1}) \\ \Leftrightarrow PreIns(f_1)(L_{init_1}) &= PreIns(f_1)(g_{1,I}(L_{init'}))\end{aligned}$$

Then from the decomposability of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$ follows that

$$L_{init_1} \circ_{(J,j_1,j_2)} PreIns(g_2)(L_{init'}) \in INS$$

$$\Leftrightarrow L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \in INS$$

$$\Leftrightarrow L_{init} \in INS$$

Part 2:

$$INIT = \{IN(L) \mid L \in INS\}$$

follows from the fact that KI_2 is an AHL-occurrence net with instantiations.

Part 3:

Let $L_{init_2} \in INS_2$.
Due to the construction of INS_2 there is $L_{init} \in INS$ such that

$$L_{init_2} = PreIns(g_2)(L_{init})$$

and there is an **INet**-morphism $(g_{2,L}, g_2) : (L_{init_2}, K_2) \to (L_{init}, K)$.
Let

$$L_{init_1} = g_{1,I}(L_{init})$$

implying an **INet**-morphism $(g_{1,L}, g_1) : (L_{init_1}, K_1) \to (L_{init}, K)$.
Then for $L_{init_0} = PreIns(f_1)(L_{init_1})$ and $L_{init'_0} = PreIns(f_2)(L_{init_2})$ there are **INet**-morphisms $(f_{1,L}, f_1) : (L_{init_0}, I) \to (L_{init_1}, K_1)$ and $(f_{2,L}, f_2) : (L_{init'_0}, I) \to$

(L_{init_2}, K_2).
Due to the commutatitivity of (1) there is

$$\begin{aligned}(g_{1,L}, g_1) \circ (f_{1,L}, f_1) &= (g_{1,L} \circ f_{1,L}, g_1 \circ f_1) \\ &= (g_{1,L} \circ f_{1,L}, g_2 \circ f_2)\end{aligned}$$

which together with

$$(g_{2,L}, g_2) \circ (f_{2,L}, f_2) = (g_{2,L} \circ f_{2,L}, g_2 \circ f_2)$$

by Lemma 4.5 implies that $L_{init_0} = L_{init'_0}$, i.e.

$$PreIns(f_1)(L_{init_1}) = PreIns(f_2)(L_{init_2})$$

Then by Theorem 4.18 (L_{init_1}, L_{init_2}) are compatible with (f_1, f_2) which by Lemma 5.15 is equivalent to the fact that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2).

"$\Leftarrow$":
Let $KI_2 = (K_2, INIT_2, INS_2)$ be the induced preimage of KI and g_2 and AHL-morphisms $f_2 : I \to K_2$, $g_2 : K_2 \to K$ such that KI_2 together with f_2 and g_2 is a cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.
This implies that

$$KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$$

and hence

$$K = K_1 \circ_{(I,f_1,f_2)} K_2$$

i.e. K_2 is a decomposition of K w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ leading to composition diagram (1).

$$\begin{tikzcd} I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \\ K_2 \arrow[r, "g_2"'] & K \end{tikzcd}$$

Furthermore Corollary 5.28 implies that there are **AHLNetI**-morphisms such that (2) is a pushout in **AHLNetI** where $II = (I, INIT_I, INS_I)$ is the interface defined in Definition 5.27.

$$\begin{tikzcd} (I, INS_I) \arrow[r, "(f_1, f_{1,I})"] \arrow[d, "(f_2, f_{2,I})"'] \arrow[dr, phantom, "(\mathbf{2})"] & (K_1, INS_1) \arrow[d, "(g_1, g_{1,I})"] \\ (K_2, INS_2) \arrow[r, "(g_2, g_{2,I})"'] & (K, INS) \end{tikzcd}$$

The first condition of the decomposability of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$ is satisfied.
From Theorem 4.13 follows that the function

$$IN : PreIns(g_2)(INS) \to PreInit(g_2)(INS)$$

is injective and hence the second condition of the decomposability of KI w.r.t. $I \stackrel{f_1}{\to} K_1 \stackrel{g_1}{\to} K$ and $g_{1,I}$ is satisfied.
For the last condition let $L_{init_1} \in INS_1$ and $L_{init} \in INS$ with

$$PreIns(f_1)(L_{init_1}) = PreIns(f_1)(g_{1,I}(L_{init}))$$

Let

$$L_{init_2} = PreIns(g_2)(L_{init})$$

We have to show that

$$L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \in INS$$

where (J, j_1, j_2) is the induced instantiation interface w.r.t. (I, f_1, f_2). So the first step is to show that (L_{init_1}, L_{init_2}) are compatible with (f_1, f_2) because this is necessary for the existence of the common interface (J, j_1, j_2).
Let

$$L_{init_0} = PreIns(f_1)(L_{init_1}) = PreIns(f_1)(g_{1,I}(L_{init}))$$

Due to definition 5.27 there is

$$f_{1,I}(L_{init_1}) = L_{init_0} = f_{1,I}(g_{1,I}(L_{init}))$$

and

$$f_{2,I}(L_{init_2}) = PreIns(f_2)(L_{init_2})$$

and due to Fact 4.27 there is

$$g_{2,I}(L_{init}) = L_{init_2}$$

From the fact that (2) is pushout follows that it commutes which means that also its components commute and hence

$$L_{init_0} = f_{1,I}(g_{1,I}(L_{init})) = f_{2,I}(g_{2,I}(L_{init})) = PreIns(f_2)(L_{init_2})$$

which by Theorem 4.18 implies that (L_{init_1}, L_{init_2}) are compatible with (f_1, f_2). From Lemma 5.15 follows that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2) and hence because $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ there is

$$L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \in INS$$

where (J, j_1, j_2) is the induced instantiation interface w.r.t. (I, f_1, f_2).

□

Detailed Proof C.36 (Preimage-Surjection-Lemma)
See Lemma 6.21

Proof. "⇒":
Let $KI_1 = (K_1, INIT_1, INS_1)$ be the induced preimage of KI_2 and f, i.e.

$$INIT_1 = PreInit(f)(INS_2)$$

and

$$INS_1 = PreIns(f)(INS_2)$$

We define a function $s : INS_2 \to INS_1$ with

$$s(L_{init_2}) = PreIns(f)(L_{init_2})$$

The function is well-defined because for every L_{init_2} by Lemma 4.7 there is a unique preimage L_{init_1} of L_{init_2} w.r.t. f.
The function s is surjective because for every $L_{init_1} \in INS_1$ there is $L_{init_2} \in INS$ such that

$$L_{init_1} = PreIns(f)(L_{init_2})$$

leading by Lemma 4.7 to a morphism $f_L : L_{init_1} \to L_{init_2}$ such that (f_L, f) is an **INet**-morphism. Hence s is the surjection with the required property that (f, s) is an **AHLNetI**-morphism.

”$\Leftarrow$”:

Let $s : INS_2 \to INS_1$ be a surjection such that (f, s) is an **AHLNetI**-morphism, i.e. for every $L_{init_2} \in INS_2$ there is an **INet**-morphism $(f_L, f) : (s(L_{init_2}), K_1) \to (L_{init_2}, K_2)$.
Let $L_{init_2} \in INS_2$ and

$$L_{init_1} = PreIns(f)(L_{init_2})$$

together with the induced morphism $f'_L : L_{init_1} \to L_{init_2}$ such that $(f'_L, f) : (L_{init_1}, K_1) \to (L_{init_2}, K)$ is an **INet**-morphism.
Due to Lemma 4.5 for $L_{init_2} \in INS$ and $f : K_1 \to K_2$ there exists a unique instantiation L_{init_1} together and morphism f'_L such that (f'_L, f) is an **INet**-morphism which implies that

$$L_{init_1} = s(L_{init_2})$$

and

$$f_L = f'_L$$

and hence for every $L_{init_2} \in INS$ there is

$$s(L_{init_2}) = PreIns(f)(L_{init_2})$$

Thus due to the surjectivity of s there is

$$\begin{aligned} INS_1 &= \{s(L_{init_2}) \mid L_{init_2} \in INS_2\} \\ &= \{PreIns(f)(L_{init_2}) \mid L_{init_2} \in INS_2\} \\ &= PreIns(f)(INS) \end{aligned}$$

Since KI_1 is an AHL-occurrence net with instantiations there is

$$\begin{aligned} INIT_1 &= \{IN(L) \mid L \in INS_1\} \\ &= \{IN(PreIns(f)(L_{init_2})) \mid L_{init_2} \in INS_2\} \\ &= \{PreInit(f)(L_{init_2}) \mid L_{init_2} \in INS_2\} \\ &= PreInit(f)(INS) \end{aligned}$$

and hence KI_1 is the induced preimage of KI_2 and f.

$\square$

Detailed Proof C.37 (Characterization of Cominimal Decompositions (I))
See Theorem 6.22

Proof. "$\Rightarrow$":

Let KI_2 be a cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$.
Since $KI = KI_1 \circ_{(I,f_1,f_2)} KI_2$ due to Theorem 5.25 there is

$$\begin{aligned} INS \;=\; & \{L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \mid (L_{init_1}, L_{init_2}) \in Composable_{(I,i_1,i_2)}(INS_1, INS_2), \\ & (J, j_1, j_2) \text{ is induced instantiation interface w.r.t. } (I, i_1, i_2)\} \end{aligned}$$

which means that for every $L_{init} \in INS$ there are $L_{init_1} \in INS_1, L_{init_2} \in INS_2$ such that

$$L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$$

and due to the definition of the composition of instantiations there is an **INet**-morphism

$$(g_{2,L}, g_2) : (L_{init_2}, K_2) \to (L_{init}, K)$$

i.e. $L_{init_2} = PreIns(g_2)(L_{init})$.
This implies that $s : INS \to INS_2$ with

$$s(L_{init}) = PreIns(g_2)(L_{init})$$

is a well-defined function such that (g_2, s) is an **AHLNetI**-morphism.
s is surjective because for every for every $L_{init_2} \in INS_2$ there is $L_{init_1} \in INS$ such that (L_{init_1}, L_{init_2}) are composable w.r.t. (I, f_1, f_2) because KI_2 is a cominimal decomposition which implies that there is $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2} \in INS$ with $s(L_{init}) = L_{init_2}$.

"$\Leftarrow$":

Given a surjection $s : INS \to INS_2$ such that (g_2, s) is an **AHLNetI**-morphism. Then from Lemma 6.21 follows that KI_2 is the induced preimage of KI and g_2, i.e.

$$INIT_2 = PreInit(g_2)(INS)$$

and

$$INS_2 = PreIns(g_2)(INS)$$

Since K_2 is the decomposition of K w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ we have that KI_2 is the cominimal decomposition of KI w.r.t. $I \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ defined in Theorem 6.17.

□

Detailed Proof C.38 (Uniqueness of Minimal Pullback Complements)
See Fact 6.24

Proof. *Part 1*:

Since (C, f_2, g_2) is a minimal pullback complement of $A \xrightarrow{g_1} B \xrightarrow{f_1} D$ in **C** there is a unique morphism $x : C \to C'$ such that

$$x \circ g_2 = g_2' \text{ and } f_2' \circ x = f_2$$

$$
\begin{tikzcd}
 & A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[ddl, bend right, "f_2'"'] & B \arrow[d, "g_1"] \\
 & C \arrow[r, "g_2"'] \arrow[dl, "x"'] & D \\
C' \arrow[urr, bend right, "g_2'"'] & &
\end{tikzcd}
$$

Furthermore $id_C : C \to C$ is the unique morphism such that

$$id_C \circ g_2 = g_2 \text{ and } f_2 \circ id_C = f_2$$

Since (C', f_2', g_2') is a minimal pullback complement of $A \stackrel{g_1}{\to} B \stackrel{f_1}{\to} D$ in $\mathbf{C}$ there is a unique morphism $x' : C' \to C$ such that

$$x' \circ g_2' = g_2 \text{ and } f_2 \circ x' = f_2'$$

$$
\begin{tikzcd}
 & A \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[ddl, bend right, "f_2'"'] & B \arrow[d, "g_1"] \\
 & C \arrow[r, "g_2"'] & D \\
C' \arrow[ur, "x'"] \arrow[urr, bend right, "g_2'"'] & &
\end{tikzcd}
$$

Furthermore $id_{C'} : C' \to C'$ is the unique morphism such that

$$id_{C'} \circ g_2' = g_2' \text{ and } f_2' \circ id_{C'} = f_2'$$

Then we have

$$f_2 \circ x' \circ x = f_2' \circ x = f_2 = f_2 \circ id_C$$

and

$$x' \circ x \circ g_2 = x' \circ g_2' = g_2 = id_C \circ g_2$$

which due to the uniqueness of id_C implies that $x' \circ x = id_C$.
Analogously we obtain

$$f_2' \circ x \circ x' = f_2 \circ x' = f_2' = f_2' \circ id_{C'}$$

and

$$x \circ x' \circ g_2' = x \circ g_2 = g_2' = id_{C'} \circ g_2'$$

which due to the uniqueness of $id_{C'}$ implies that $x \circ x' = id_{C'}$.
Hence x and x' are isomorphisms.

Part 2:
Since $x : C \to C'$ is an isomorphism there is $x' : C' \to C$ such that

$$x' \circ x = id_C \text{ and } x \circ x' = id_{C'}$$

Then there is

$$g_2 \circ x' = g_2' \circ x \circ x' = g_2' \circ id_{C'} = g_2'$$

and

$$x' \circ f_2' = x' \circ x \circ f_2 = id_C \circ f_2 = f_2$$

To show the uniqueness of x' let $x'' : C' \to C$ be a morphism such that

$$g_2 \circ x'' = g_2' \text{ and } x'' \circ f_2' = f_2$$

Then there is

$$g_2 \circ x'' = g_2' = g_2 \circ x'$$

which due to the fact that g_2 is a monomorphism implies that $x'' = x'$.
Hence (C', f_2', g_2') is a minimal pullback complement of $A \overset{f_1}{\to} B \overset{g_1}{\to} D$ in $\mathbf{C}$.

□

Detailed Proof C.39 (Characterization of Cominimal Decompositions (II))
See Theorem 6.25

Proof. ”⇒”:
Let the decomposition be cominimal. Then from Theorem 6.22 follows that there is a surjective function $s : INS \to INS_1$ such that (g_2, s) is an **AHLNetI**-morphism.
From Fact 4.26 follows that $s = g_{2,I}$ which means that $g_{2,I}$ is surjective.
Due to Fact 4.27 diagram (2) is a pullback, i.e. $(INS_2, f_{2,I}, g_{2,I})$ is a pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.
Let $(X, f_{2,I}', g_{2,I}')$ be a pushout complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{tikzcd} INS_I & INS_1 \arrow[l, "f_{1,I}"'] \\ X \arrow[u, "f_{2,I}'"] \arrow[ur, phantom, "(3)"] & INS \arrow[l, "g_{2,I}'"] \arrow[u, "g_{1,I}"'] \end{tikzcd}$$

Due to the surjectivity of $g_{2,I}$ for every $L_2 \in INS_2$ there is $L \in INS$ such that $g_{2,I}(L) = L_2$. So we can define a function $x : INS_2 \to X$ with

$$x(g_{2,I}(L)) = g_{2,I}'(L)$$

Then there is

$$x \circ g_{2,I} = g_{2,I}'$$

and

$$f_{2,I}'(x(g_{2,I}(L))) = f_{2,I}'(g_{2,I}'(L)) = f_{1,I}(g_{1,I}(L)) = f_{2,I}(g_{2,I}(L))$$

which means that

$$f_{2,I}' \circ x = f_{2,I}$$

Let $x' : INS_2 \to X$ with $x' \circ g_{2,I} = g_{2,I}'$ and $f_{2,I}' \circ x' = f_{2,I}$. Then there is

$$x'(g_{2,I}(L)) = g_{2,I}'(L) = x(g_{2,I}(L))$$

and hence $x = x'$ because $g_{2,I}$ is surjective. So we have that INS_2 together with $f_{2,I}, g_{2,I}$ is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

"$\Leftarrow$":

Let INS_2 together with $f_{2,I}, g_{2,I}$ be a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{array}{ccc} INS_I & \xleftarrow{f_{1,I}} & INS_1 \\ {\scriptstyle f_{2,I}}\uparrow & (\mathbf{2}) & \uparrow{\scriptstyle g_{1,I}} \\ INS_2 & \xleftarrow[g_{2,I}]{} & INS \end{array}$$

Furthermore let $(K_2', INIT_2', INS_2')$ be the cominimal decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ as defined in Theorem 6.17.
From the part "$\Rightarrow$" we know that also INS_2' together with $f_{2,I}', g_{2,I}'$ is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{array}{ccc} INS_I & \xleftarrow{f_{1,I}} & INS_1 \\ {\scriptstyle f_{2,I}'}\uparrow & (\mathbf{4}) & \uparrow{\scriptstyle g_{1,I}} \\ INS_2' & \xleftarrow[g_{2,I}']{} & INS \end{array}$$

The uniqueness of pullback complements in Fact 6.24 implies an isomorphism $x : INS_2' \to INS_2$ such that

$$x \circ g_{2,I}' = g_{2,I} \text{ and } f_{2,I} \circ x = f_{2,I}'$$

Lemma 6.21 implies a surjection $s : INS \to INS_2'$ such that (g_2', s) is an **AHLNetI**-morpism because KI_2 is the induced preimage of KI and g_2 as defined in Theorem 6.17.
Due to Fact 4.26 there is $s = g_{2,I}'$, i.e. $g_{2,I}'$ is surjective.
Then due to the composition of surjective functions and $g_{2,I} = x \circ g_{2,I}'$ it follows that $g_{2,I}$ is surjective.
Together with the fact that $(g_2, g_{2,I})$ is an **AHLNetI**-morphism from Lemma 6.21 follows that KI_2 is the induced preimage of KI and g_2 which by Theorem 6.17 implies that KI_2 is a cominimal decomposition of KI w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$

□

Detailed Proof C.40 (Characterization of Maximal Pushout Complements in **AHLNetI**)
See Lemma 6.26.

Proof. "$\Rightarrow$":

By Fact 4.27 there is pushout (3) in **AHLNet** and pullback (2) in **SET**.

$$\begin{array}{ccc} I & \xrightarrow{f_{1,N}} & K_1 \\ {\scriptstyle f_{2,N}}\downarrow & (\mathbf{3}) & \downarrow{\scriptstyle g_{1,N}} \\ K_2 & \xrightarrow[g_{2,N}]{} & K \end{array} \qquad \begin{array}{ccc} INS_I & \xleftarrow{f_{1,I}} & INS_1 \\ {\scriptstyle f_{2,I}}\uparrow & (\mathbf{2}) & \uparrow{\scriptstyle g_{1,I}} \\ INS_2 & \xleftarrow[g_{2,I}]{} & INS \end{array}$$

Let $KI_2' = (K_2', INIT_2', INS_2')$ together with $f_{2,N} : I \to K_2'$ and $g_{2,N} : K_2' \to K$ be the cominimal decomposition of KI w.r.t. $I \overset{f_{1,N}}{\to} K_1 \overset{g_{1,N}}{\to} K$ as defined in Theorem 6.17. Then KI_2' is the induced preimage of KI and $g_{2,N}'$ which by Lemma 6.21 implies that there is a surjective function $g_{2,I}' : INS \to INS_2'$ such that $g_2' = (g_{2,N}', g_{2,I}')$ is an **AHLNetI**-morphism. Due to the surjectivity of $g_{2,I}'$ for every $L_2 \in INS_2'$ there is $L \in INS$ with $g_{2,I}(L) = L_2$. We define $f_{2,I}' : INS_2 \to INS_I$ with

$$f_{2,I}'(g_{2,I}'(L)) = f_{1,I}(g_{1,I}(L))$$

Then by definition there is $f_{2,I}' \circ g_{2,I}' = f_{1,I} \circ g_{2,I}$.
By Theorem 6.17 there is $KI = KI_1 \circ_{(I,f_{1,N},f_{2,N}')} KI_2'$ which means that for $L \in INS$ there are $L_1 \in INS_1, L_2 \in INS_2'$ such that

$$L = L_1 \circ_{(J,j_1,j_2)} L_2$$

where (J, j_1, j_2) is the induced instantiation interface w.r.t. $(I, f_{1,N}, f_{2,N}')$. This means that there is pushout (4) in **INet** due to the definition of the composition of instantiations.

$$\begin{tikzcd}
(J,I) \arrow[r, "{(f_{1,L},f_{1,N})}"] \arrow[d, "{(f_{2,L},f_{2,N}')}"'] \arrow[dr, phantom, "\mathbf{(4)}"] & (L_1,K_1) \arrow[d, "{(g_{1,L},g_{1,N})}"] \\
(L_2,K_2) \arrow[r, "{(g_{2,L},g_{2,N}')}"'] & (L,K)
\end{tikzcd}$$

Due to the uniqueness of instantiation preimages there is

$$J = PreIns(g_{1,N} \circ f_{1,N})(L) = f_{1,I} \circ g_{1,I}(L) = f_{2,I}'(L)$$

which means that for $L_2 \in INS_2'$ there is an **INet**-morphism $(f_{2,L}, f_{2,N}') : (f_{2,I}'(L_2), I) \to (L_2, K_2)$.
Hence $f_2' = (f_{2,N}, f_{2,I})$ is an **AHLNetI**-morphism.
Since K_2' is constructed via pushout complement in **AHLNet** there is $g_{2,N}' \circ f_{2,N}' = g_{1,N} \circ f_{1,N}$ and hence

$$g_2' \circ f_2' = g_1 \circ f_1$$

$$\begin{tikzcd}
(I,INS_I) \arrow[r, "f_1"] \arrow[d, "f_2'"'] \arrow[dr, phantom, "\mathbf{(5)}"] & (K_1,INS_1) \arrow[d, "g_1"] \\
(K_2',INS_2') \arrow[r, "g_2'"'] & (K,INS)
\end{tikzcd}$$

So we have that diagram (5) is a commuting diagram in **AHLNetI** which by Corollary 6.19 implies that (K_2', INS_2') is a pushout complement of $(I, INS_I) \overset{f_1}{\to} (K_1, INS_1) \overset{g_1}{\to} (K, INS)$ in **AHLNetI**. Due to Fact 4.27 and pushout (5) in **AHLNetI** there is also pushout (6) in **AHLNet**.

$$
\begin{tikzcd}
I \arrow[r, "f_{1,N}"] \arrow[d, "f_{2,N}"'] \arrow[dr, phantom, "(\mathbf{3})"] & K_1 \arrow[d, "g_{1,N}"] \\
K_2 \arrow[r, "g_{2,N}"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "f_{1,N}"] \arrow[d, "f'_{2,N}"'] \arrow[dr, phantom, "(\mathbf{6})"] & K_1 \arrow[d, "g_{1,N}"] \\
K'_2 \arrow[r, "g'_{2,N}"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "f_{2,N}"] \arrow[d, "f'_{2,N}"'] & K_2 \arrow[d, "g_{2,N}"] \arrow[dl, "x_N"] \\
K'_2 \arrow[r, "g'_{2,N}"'] & K
\end{tikzcd}
$$

From the uniqueness of pushout complements in the category **AHLNet** follows that there is an isomorphism $x_N : K_2 \to K'_2$ with

$$x_N \circ f_{2,N} = f'_{2,N} \text{ and } g'_{2,N} \circ x_N = g_{2,N}$$

Let $x_I : INS'_2 \to INS_2$ with

$$x_I(g'_{2,I}(L)) = g_{2,I}(L)$$

x_I is a well-defined function because $g'_{2,I}$ and $g_{2,I}$ are well-defined functions and $g'_{2,I}$ is surjective.
Furthermore there is

$$f_{2,I} \circ x_I(g'_{2,I}(L)) = f_{2,I}(g_{2,I}(L)) = f_{1,I}(g_{1,I}(L)) = f'_{2,I}(g'_{2,I}(L))$$

which means that there is

$$x_I \circ g'_{2,I} = g_{2,I} \text{ and } f_{2,I} \circ x_I = g'_{2,I}$$

From the uniqueness of instantiation preimages we obtain

$$
\begin{aligned}
PreIns(x_N)(g'_{2,I}(L)) = &= PreIns(x_N)(PreIns(g'_{2,N})(L)) \\
&= PreIns(g'_{2,N} \circ x_N)(L) \\
&= PreIns(g_{2,N})(L) \\
&= g_{2,I}(L)
\end{aligned}
$$

which means that there is an **INet**-morphism $(x_L, x_N) : (x_I(g'_{2,I}(L)), K_2) \to (g'_{2,I}(L), K'_2)$, i.e. $x = (x_N, x_I)$ is an **AHLNetI**-morphism with

$$x \circ f_2 = f'_2 \text{ and } g'_2 \circ x = g_2$$

Since (K_2, INS_2) is a maximal pushout complement in **AHLNetI** there is an **AHLNetI**-morphism $x' = (x'_N, x'_I) : (K'_2, INS'_2) \to (K_2, INS_2)$ such that

$$x' \circ f'_2 = f_2 \text{ and } g_2 \circ x' = g'_2$$

$$
\begin{tikzcd}
I \arrow[r, "f_{1,N}"] \arrow[d, "f_{2,N}"'] \arrow[dr, phantom, "(\mathbf{3})"] & K_1 \arrow[d, "g_{1,N}"] \\
K_2 \arrow[r, "g_{2,N}"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "f_{1,N}"] \arrow[d, "f'_{2,N}"'] \arrow[dr, phantom, "(\mathbf{6})"] & K_1 \arrow[d, "g_{1,N}"] \\
K'_2 \arrow[r, "g'_{2,N}"'] & K
\end{tikzcd}
\qquad
\begin{tikzcd}
I \arrow[r, "f_{2,N}"] \arrow[d, "f'_{2,N}"'] & K_2 \arrow[d, "g_{2,N}"] \arrow[dl, bend left, "x_N"] \\
K'_2 \arrow[r, "g'_{2,N}"'] \arrow[ur, bend left, "x'_N"] & K
\end{tikzcd}
$$

This means there is also

$$x'_N \circ f'_{2,N} = f_{2,N} \text{ and } g_{2,N} \circ x'_N = g'_{2,N}$$

Since pushouts in **AHLNet** along monomorphisms are also pullbacks and injective AHL-morphisms are monomorphisms pushouts (3) and (6) are also pullbacks. Due to the fact that pullbacks preserve monomorphisms $g_{2,N}$ and $g'_{2,N}$ are monomorphisms and hence from

$$g'_{2,N} \circ x_N \circ x'_N = g_{2,N} \circ x'_N = g'_{2,N} = g'_{2,N} \circ id_{K'_2}$$

and

$$g_{2,N} \circ x'_N \circ x_N = g'_{2,N} \circ x_N = g_{2,N} = g_{2,N} \circ id_{K_2}$$

follows that

$$x_N \circ x'_N = id_{K'_2} \text{ and } x'_N \circ x_N = id_{K_2}$$

i.e. x_N and x'_N are inverse to each other.
From the well-defined composition in **AHLNetI** we obtain **AHLNetI**-morphisms

$$x'_N \circ x_N : (K_2, INS_2) \to (K_2, INS_2) \text{ and } x_N \circ x'_N : (K'_2, INS'_2) \to (K'_2, INS'_2)$$

and since **AHLNetI** is a category there are identities

$$id_{(K_2,INS_2)} : (K_2, INS_2) \to (K_2, INS_2) \text{ and } id_{(K'_2,INS'_2)} : (K'_2, INS'_2) \to (K'_2, INS'_2)$$

Then from the fact that

$$x_N \circ x'_N = id_{K'_2} \text{ and } x'_N \circ x_N = id_{K_2}$$

and the uniqueness of instantiation set morphisms in Fact 4.26 follows that

$$x' \circ x = id_{(K_2,INS_2)} \text{ and } x \circ x' = id_{(K'_2,INS'_2)}$$

and hence

$$x_I \circ x'_I = id_{INS_2} \text{ and } x'_I \circ x_I = id_{INS'_2}$$

i.e. x_I and x'_I are isomorphisms.

Let (X, g, f) be a pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$\begin{tikzcd}
INS_I & INS_1 \arrow[l, "f_{1,I}"'] \\
X \arrow[u, "f"] \arrow[ur, phantom, "(\mathbf{7})"] & INS \arrow[l, "g"] \arrow[u, "g_{1,I}"']
\end{tikzcd}$$

Since KI'_2 is a cominimal decomposition of KI w.r.t. $I \overset{f_{1,N}}{\to} K_1 \overset{g_{1,N}}{\to} K$ due to Theorem 6.25 INS'_2 is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

$$
\begin{tikzcd}
INS_I & INS_1 \arrow[l, "f_{1,I}"'] \\
INS'_2 \arrow[u, "f'_{2,I}"] \arrow[ur, phantom, "(\mathbf{8})"] & INS \arrow[l, "g'_{2,I}"] \arrow[u, "g_{1,I}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
INS_I & INS'_2 \arrow[l, "f'_{2,I}"'] \arrow[dl, "y"] \\
INS'_2 \arrow[u, "f"] & INS \arrow[l, "g"] \arrow[u, "g'_{2,I}"']
\end{tikzcd}
$$

Due to the fact that (X, g, f) is a pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET** there is a unique morphism $y : INS'_2 \to X$ such that

$$y \circ g'_{2,I} = g \text{ and } f \circ y = f'_{2,I}$$

$$
\begin{tikzcd}
& INS_I & INS_2 \arrow[l, "f_{2,I}"'] \arrow[dl, bend left, "x'_I"] \\
& INS'_2 \arrow[u, "f'_{2,I}"] \arrow[ur, bend left, "x_I"] \arrow[dl, "y"] & INS \arrow[l, "g'_{2,I}"] \arrow[u, "g_{2,i}"'] \\
X \arrow[uur, bend left, "f"] \arrow[urr, bend right, "g"'] & &
\end{tikzcd}
$$

Let $z := y \circ x'_I : INS_2 \to X$. Then there is

$$z \circ g_{2,I} = y \circ x'_I \circ g_{2,I} = y \circ g'_{2,I} = g$$

and

$$f \circ z = f \circ y \circ x'_I = f'_{2,I} \circ x'_I = f_{2,I}$$

Let $z' : INS'_2 \to X$ such that

$$z' \circ g_{2,I} = g \text{ and } f \circ z' = f_{2,I}$$

Then there is

$$z' \circ x_I \circ g'_{2,I} = z' \circ g_{2,I} = g$$

and

$$f \circ z' \circ x_I = f_{2,I} \circ x_I = f'_{2,I}$$

which due to the uniqueness of y implies that

$$z' \circ x_I = y$$

This means

$$z' = z' \circ id_{INS_2} = z' \circ x_I \circ x'_I = y \circ x'_I = z$$

i.e. z is unique and hence INS_2 is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET**.

"$\Leftarrow$":

By Fact 4.27 there is pushout (3) in **AHLNet** and pullback (2) in **SET**.

$$\begin{tikzcd} I \arrow[r, "f_{1,N}"] \arrow[d, "f_{2,N}"'] \arrow[dr, phantom, "(\mathbf{3})"] & K_1 \arrow[d, "g_{1,N}"] \\ K_2 \arrow[r, "g_{2,N}"'] & K \end{tikzcd} \qquad \begin{tikzcd} INS_I \arrow[dr, phantom, "(\mathbf{2})"] & INS_1 \arrow[l, "f_{1,I}"'] \\ INS_2 \arrow[u, "f_{2,I}"] & INS \arrow[l, "g_{2,I}"] \arrow[u, "g_{1,I}"'] \end{tikzcd}$$

Let $((K_2', INS_2'), f_2', g_2')$ a pushout complement in **AHLNetI**. Due to the uniqueness of pushout complements in **AHLNet** there is an isomorphism $x_N : K_2' \to K_2$ such that

$$x_N \circ f_{2,N}' = f_{2,N} \text{ and } g_{2,N} \circ x_N = g_{2,N}'$$

Since INS_2 is a minimal pullback complement of $INS \overset{g_{1,I}}{\to} INS_1 \overset{f_{1,I}}{\to} INS_I$ in **SET** there is a unique morphism $x_I : INS_2 \to INS_2'$ such that

$$x_I \circ g_{2,I} = g_{2,I}' \text{ and } f_{2,I}' \circ x_I = f_{2,I}$$

We define $x = (x_N, x_I)$ and there is

$$x \circ f_2' = (x_N, x_I) \circ (f_{2,N}', f_{2,I}') = (x_N \circ f_{2,N}', f_{2,I}' \circ x_I) = (f_{2,N}, f_{2,I}) = f_2$$

and

$$g_2 \circ x = (g_{2,N}, g_{2,I}) \circ (x_N, x_I) = (g_{2,N} \circ x_N, x_I \circ g_{2,I}) = (g_{2,N}', g_{2,I}') = g_2'$$

Let $x' : (K_2', INS_2') \to (K_2, INS_2)$ such that

$$x' \circ f_2' = f_2 \text{ and } g_2 \circ x' = g_2'$$

Since this means that also the components commute there is

$$g_{2,N} \circ x_N' = g_{2,N}' = g_{2,N} \circ x_N$$

Pushouts along monomorphisms in **AHLNet** are pullbacks. Therefore pushout (3) is also a pullback because by assumption the morphism $f_{1,N}$ is injective which means that it is a monomorphism. Since pullbacks preserve monomorphisms also $g_{2,N}$ is a monomorphism and hence $x_N' = x_N$.
Furthermore

$$x_I' \circ f_{2,I}' = f_{2,I} \text{ and } g_{2,I} \circ x_I' = g_{2,I}'$$

implies $x_I' = x_I$ because x_I is unique due to the property of minimal pullbacks.
Hence

$$x = (x_N, x_I) = (x_N', x_I') = x'$$

which means that $((K_2, INS_2), f_2, g_2)$ a maximal pushout complement in **AHLNetI**.

□

Detailed Proof C.41 (Pushout Complement of AHL-Processes with Instantiations)
See Corollary 6.32.

Proof. ”$\Rightarrow$”:

Given the maximal pushout complement (K_2, INS_2) of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLNetI** and **AHLNetI**-morphisms mp_0, mp_1, mp_2 and $mp = (mp_N, mp_I)$ such that mp_2 is a pushout complement of $mp_0 \overset{(f_1,f_{1,I})}{\to} mp_1 \overset{(g_1,g_{1,I})}{\to} mp$ in **AHLProcI**(**AN**), i.e. (1) is pushout in **AHLNetI** and in **AHLProcI**(**AN**).

$$\begin{tikzcd}
(I, INS_I) \arrow[r, "{(f_1,f_{1,I})}"] \arrow[d, "{(f_2,f_{2,I})}"'] \arrow[dr, phantom, "(\mathbf{1})"] \arrow[ddrr, bend right, "mp_0"'] & (K_1, INS_1) \arrow[d, "{(g_1,g_{1,I})}"] \arrow[ddr, bend left, "mp_1"] & \\
(K_2, INS_2) \arrow[r, "{(g_2,g_{2,I})}"'] \arrow[drr, "mp_2"'] & (K, INS) \arrow[dr, "mp"] & \\
 & & (AN, \emptyset)
\end{tikzcd}$$

Then diagram (1) is also a diagram in **AHLONetI**.
Let (X, INS_X) be an **AHLONetI**-object together with **AHLONetI**-morphisms $x_1 : (K_1, INS_1) \to (X, INS_X)$ and $x_2 : (K_2, INS_2) \to (X, INS_X)$ such that

$$x_1 \circ (f_1, f_{1,I}) = x_2 \circ (f_2, f_{2,I})$$

Since **AHLONetI** is a subcategory of **AHLNetI** the object (X, INS_X) and morphisms x_1 and x_2 are also in **AHLNetI** which due to the pushout property of (1) implies that there is a unique morphism $x : (K, INS) \to (X, INS_X)$ such that

$$x \circ (g_1, g_{1,I}) = x_1 \text{ and } x \circ (g_2, g_{2,I}) = x_2$$

Since **AHLONetI** is a full subcategory of **AHLNetI** the morphism x between AHL-occurrence nets with instantiations (K, INS) and (X, INS) is also an **AHLONetI**-morphism. Hence (1) is also pushout in **AHLONetI** which means that (K_2, INS_2) is a pushout complement of $(I, INS_I) \overset{(f_1,f_{1,I})}{\longrightarrow} (K_1, INS_1) \overset{(g_1,g_{1,I})}{\longrightarrow} (K, INS)$ in **AHLONetI**.
So Corollary 6.29 implies that KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$.

”$\Leftarrow$”:

Since KI is decomposable w.r.t. $I \overset{f_1}{\to} K_1 \overset{g_1}{\to} K$ and $g_{1,I}$ Corollary 6.29 implies that (K_2, INS_2) is also a pushout complement in **AHLONetI**.
Furthermore the decomposability implies that the dangling condition is satisfied and hence due to Corollary 6.13 there exist morphisms $mp_{0,N} : I \to AN, mp_{1,N} : K_1 \to AN, mp_{2,N} : K_2 \to AN$ such that mp_2 is the pushout complement of $mp_{0,N} \overset{f_1}{\to} mp_{1,N} \overset{g_1}{\to} mp_N$ in **AHLProc**(**AN**).
The fact that $(AN, \emptyset) = Inst(AN)$ and $Inst$ is a cofree functor implies that there are unique morphisms mp_0, mp_1, mp_2 and mp such that $Net(mp_0) = mp_{0,N}$, $Net(mp_1) = mp_{1,N}$, $Net(mp_2) = mp_{2,N}$ and $Net(mp) = mp_N$, i.e. diagram (1) is a diagram in **AHLProcI**(**AN**).
Let $mp_X : (X, INS_X) \to (AN, \emptyset)$ in **AHLProcI**(**AN**) together with **AHLProcI**(**AN**)-morphisms $x_1 : (K_1, INS_1) \to (X, INS_X)$ and $x_2 : (K_2, INS_2) \to (X, INS_X)$ such that

$$x_1 \circ (f_1, f_{1,I}) = x_2 \circ (f_2, f_{2,I})$$

Then (X, INS_X) is an AHL-occurrence net which implies that x_1 and x_2 are **AHLONetI**-morphisms. Since (1) is pushout in **AHLONetI** there is a unique morphism $x : (K, INS) \to (X, INS_X)$ such that

$$x \circ (g_1, g_{1,I}) = x_1 \text{ and } x \circ (g_2, g_{2,I}) = x_2$$

This means that there is

$$x_N \circ g_1 = x_{1,N} \text{ and } x_N \circ g_2 = x_{2,N}$$

which by the fact that

$$x_{1,N} \circ f_1 = x_{2,N} \circ f_2$$

implies that x_N is the unique morphism induced by pushout object mp_N in **AHLProc**(**AN**), i.e. $mp_{X,N} \circ x_N = mp_N$.
Due to the initiality of $\emptyset$ in **SET** there is

$$x_{1,I} \circ mp_{X,I} = mp_I$$

which means that x is an **AHLProcI**(**AN**)-morphism. Hence (1) is a pushout in **AHLProcI**(**AN**).

□

C.7 Rule-based Transformation of Algebraic High-Level Processes

Detailed Proof C.42 (Gluing Relation Lemma)
See Lemma 7.6.

Proof. *Part 1*:
We define a relation

$$\prec_C \subseteq (P_C \times T_C) \uplus (T_C \times P_C)$$
$$\prec_C = \{(x, y) \mid x \in \bullet y\}$$

The relation $\prec_C$ describes the direct causal relationship of the elements in C, i.e. the causal relation $<_C$ is the transitive closure of $\prec_C$.
We show that $\prec_{(K,m)} = \prec_C$:

"$\subseteq$":
Let $x, y \in P_K \uplus (T_K \setminus m_T(T_L))$ with $x \prec_{(K,m)} y$. Due to the structure of petri nets there are two possible cases:

Case 1: $x \in P_K, y \in T_K \setminus m_T(T_L)$
Due to the pushout complement construction of C there is $y \in T_C$. Furthermore there is $term \in T_{OP}(X)_{type_K(x)}$ such that

$$\begin{aligned} & (term, x) \leq pre_K(y) \\ \Leftrightarrow\ & (term, x) \leq pre_K|_{T_C}(y) \\ \Leftrightarrow\ & (term, x) \leq pre_C(y) \end{aligned}$$

and hence $x \prec_C y$.

<u>*Case 2:*</u> $x \in T_K \setminus m_T(T_L), y \in P_K$

In this case is $x \in T_C$ and there is $term \in T_{OP}(X)_{type_K(x)}$ such that

$$\begin{aligned} & (term, y) \leq post_K(x) \\ \Leftrightarrow\ & (term, y) \leq post_K|_{T_C}(x) \\ \Leftrightarrow\ & (term, y) \leq post_C(x) \end{aligned}$$

and hence $x \prec_C y$.

"$\supseteq$":

Let $x, y \in P_C \uplus T_C$ with $x \prec_C y$. Again we distinguish the two possible cases:

<u>*Case 1:*</u> $x \in P_C, y \in T_C$

Then there is $term \in T_{OP}(X)_{type_C(x)}$ such that

$$(term, x) \leq pre_C(y)$$

Since AHL-morphisms preserve pre conditions and d is an inclusion we have

$$\begin{aligned} & (term, x) \leq pre_C(y) \\ \Leftrightarrow\ & (term, x) \leq d^{\oplus} \circ pre_C(y) \\ \Leftrightarrow\ & (term, x) \leq pre_K(d(y)) \\ \Leftrightarrow\ & (term, x) \leq pre_K(y) \end{aligned}$$

So the fact that $T_C = T_K \setminus m_T(T_L)$ implies $x \prec_{(K,m)} y$.

<u>*Case 2:*</u> $x \in T_C, y \in P_C$

Then there is $term \in T_{OP}(X)_{type_C(x)}$ such that

$$(term, x) \leq post_C(y)$$

Since AHL-morphisms preserve not only pre but also post conditions we obtain analogously to Case 1 that $x \prec_{(K,m)} y$.

So we have that $\prec_{(K,m)} = \prec_C$ and since $<_{(K,m)}$ is the transitive closure of $\prec_{(K,m)}$ and $<_C$ is the transitive closure of $\prec_C$ it follows that $<_{(K,m)} = <_C$.
Furthermore we can use the inclusion d to obtain from the commutativity of (1):

$$m \circ l(x) = d \circ c(x) = c(x)$$

and hence

$$\begin{aligned} & m \circ l(x) <_{(K,m)} m \circ l(y) \\ \Leftrightarrow\ & c(x) <_{(K,m)} c(y) \\ \Leftrightarrow\ & c(x) <_C c(y) \end{aligned}$$

Part 2:

Follows from Part 1 and the definitions of $<_{(p,m)}$ and $<_{(c,r)}$.

$\square$

Detailed Proof C.43 (Construction of Direct Transformations of AHL-Occurrence Nets)
See Theorem 7.9.

Proof.

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \\
K & C \arrow[l, "d"] \arrow[r, "e"'] & K'
\end{tikzcd}$$

"$\Rightarrow$":

Let production p satisfy the extended gluing condition under match m.

decomposition:

Since the dangling condition is satisfied from Theorem 6.8 follows that the decomposition C of K w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$ exists leading to composition diagram (1).

composition:

We have to show that (C, R) are composable w.r.t. (I, c, r), i.e.

1. c is injective
2. for all $x \in P_I$ there is
   $c(x) \notin IN(C) \Rightarrow r(x) \in IN(R)$ and
   $c(x) \notin OUT(C) \Rightarrow r(x) \in OUT(R)$
3. the induced causal relation $<_{(c,r)}$ is a finitary strict partial order

Part 1:

The injectivity of c follows from the fact that (L, C) are composable w.r.t. (I, l, c) because C is the decomposition of K w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$.

Part 2:

Let $x \in P_I$.

$c(x) \notin IN(C) \Rightarrow r(x) \in IN(R)$:

Let

$$c(x) \notin IN(C)$$

then by the composability of (L, C) w.r.t. (I, l, c) follows that

$$l(x) \in IN(L)$$

The fact that $c(x) \notin IN(C)$ implies $t \in T_C$ with

$$c(x) \in t\bullet$$

and

$$g_2 \circ c(x) \in g_2(t)\bullet$$

because AHL-morphisms preserve post conditions. Due to the commutativity of (1) there is

$$m \circ l(x) = g_2 \circ c(x)$$

which means that

$$m \circ l(x) \notin IN(K)$$

because $m \circ l(x) \in g_2(t)\bullet$.
So there is $x \in InP$ and the fact that p satisfies the extended gluing condition implies that $r(x) \in IN(R)$.

$c(x) \notin OUT(C) \Rightarrow r(x) \in OUT(R)$:
Let

$$c(x) \notin OUT(C)$$

then by the composability of (L, C) w.r.t. (I, l, c) follows that

$$l(x) \in OUT(L)$$

The fact that $c(x) \notin OUT(C)$ implies $t \in T_C$ with

$$c(x) \in \bullet t$$

and

$$g_2 \circ c(x) \in \bullet g_2(t)$$

because AHL-morphisms preserve pre conditions. Due to the commutativity of (1) there is

$$m \circ l(x) = g_2 \circ c(x)$$

which means that

$$m \circ l(x) \notin OUT(K)$$

because $m \circ l(x) \in \bullet g_2(t)$.
So there is $x \in OutP$ and the fact that p satisfies the extended gluing condition implies that $r(x) \in OUT(R)$.

Part 3:
The fact that the gluing relation $<_{(p,m)}$ of p und m is a finitary strict partial order implies that the induced causal relation $<_{(c,r)}$ is a finitary strict partial order because by Lemma 7.6 there is

$$x <_{(p,m)} y \;\Leftrightarrow\; x <_{(c,r)} y$$

Thus (C, R) are composable w.r.t. (I, c, r) leading to an AHL-occurrence net $K' = C \circ_{(I,c,r)} R$ with composition diagram (2).

"$\Leftarrow$":
Let the composition diagrams (1) and (2) exist.

dangling condition:
We have that C is the decomposition of K w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$ which by Theorem 6.8 implies that the dangling condition is satisfied.

gluing relation:
By Theorem 5.5 composition diagram (1) implies that (C, R) are composable w.r.t. (I, c, r) which means that the incuded causal relation $<_{(c,r)}$ is a finitary strict

partial order.
Due to the gluing relation lemma there is for $x, y \in P_I$ that

$$x <_{(c,r)} y \;\Leftrightarrow\; x <_{(p,m)} y$$

leading to the fact that also $<_{(p,m)}$ is a finitary strict partial order.

conflict-freeness condition:
Since (1) is a composition diagram it is a pushout in **AHLNet** and therefore from the uniqueness of pushout complements of AHL-nets follows that C can be constructed in the way given in Fact 6.6.

$r(InP) \subseteq IN(R)$:
Let $x \in InP$, i.e. $x \in P_I$ with

$$l_P(x) \in IN(L) \text{ and } m_P \circ l_P(x) \notin IN(K)$$

$m_P \circ l_P(x) \notin IN(K)$ implies that there is $t \in T_K$ with

$$m_P \circ l_P(x) \in t\bullet$$

Let us assume that there is $t' \in T_L$ with $m_T(t') = t$. Then from the fact that m is an AHL-morphism follows that $l_P(x) \in t'\bullet$ because AHL-morphisms preserve post conditions.
This contradicts the fact that $l_P(x) \in IN(K)$ and hence $t \notin m_T(T_L)$, i.e. $t \in T_K \setminus m_T(T_L)$.
Then by the construction of T_C follows that $t \in T_C$.
Furthermore there is

$$c_P(x) = m_P \circ l_P(x) \in t\bullet$$

which means that

$$c_P(x) \notin IN(C)$$

This implies that $r(x) \in IN(R)$ due to the composability of (C, R) w.r.t. (I, c, r) given by composition diagram (2).

$r(OutP) \subseteq OUT(R)$:
Let $x \in OutP$, i.e. $x \in P_I$ with

$$l_P(x) \in OUT(L) \text{ and } m_P \circ l_P(x) \notin OUT(K)$$

$m_P \circ l_P(x) \notin OUT(K)$ implies that there is $t \in T_K$ with

$$m_P \circ l_P(x) \in \bullet t$$

Let us assume that there is $t' \in T_L$ with $m_T(t') = t$. Then from the fact that m is an AHL-morphism follows that $l_P(x) \in \bullet t'$ because AHL-morphisms preserve pre conditions.
This contradicts the fact that $l_P(x) \in OUT(K)$ and hence $t \notin m_T(T_L)$, i.e. $t \in T_K \setminus m_T(T_L)$.
Then by the construction of T_C follows that $t \in T_C$.
Furthermore there is

$$c_P(x) = m_P \circ l_P(x) \in \bullet t$$

which means that

$$c_P(x) \notin OUT(C)$$

This implies that $r(x) \in OUT(R)$ due to the composability of (C, R) w.r.t. (I, c, r) given by composition diagram (2).

□

Detailed Proof C.44 (Double Pushout Transformation of AHL-Processes)
See Theorem 7.15.

Proof. "⇒":

Since p satisfies the extended gluing condition under m from Theorem 7.9 follows that the composition diagrams (1) and (2) exists and from Corollary 7.11 follows that $K \overset{(p,m)}{\Rightarrow} K'$ is a double pushout transformation in **AHLONet**, i.e. (1) and (2) are pushouts in **AHLONet**.

Since AHL-morphisms reflect AHL-processes there are process morphisms $ip : I \to AN$, $lp : L \to AN$ such that the morphisms in diagram (1) and (2) are **AHLProc**(**AN**) morphisms.

Then there is

$$lp \circ l = ip = cp \circ c \text{ and } rp \circ r = ip = cp \circ c$$

i.e. (lp, cp) are composable w.r.t. (I, l, c) and (rp, cp) are composable w.r.t. (I, r, c). This implies by Corollary 5.11 that there are morphisms $\overline{mp} : K \to AN$ and $\overline{mp}' : K' \to AN$ such that $\overline{mp}$ is pushout object of (1) and $\overline{mp}'$ is pushout object of (2) are pushouts in **AHLProc**(**AN**).

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[ddrr, bend right=40, "lp"'] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"'] \arrow[dl, phantom, "(\mathbf{1})"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"'] \arrow[dd, bend left=50, "rp"] \\
K \arrow[drr, "\overline{mp}"'] & C \arrow[l, "d"'] \arrow[r, "e"] \arrow[dr, "cp"] & N' \arrow[d, "\overline{mp}'"] \\
 & & AN
\end{tikzcd}$$

Then there is

$$mp \circ m = lp \text{ and } mp \circ d = cp$$

and

$$\overline{mp} \circ m = lp \text{ and } \overline{mp} \circ d = cp$$

which due to the pushout property of (1) implies that $mp = \overline{mp}$. Analogously the pushout property of (2) implie $mp' = \overline{mp}'$.

Hence (1) and (2) are pushouts in **AHLProc**(**AN**), i.e. $mp \overset{(p,m)}{\Rightarrow} mp'$ is a double pushout transformation in **AHLProc**(**AN**).

"⇐":

$mp \overset{(p,m)}{\Rightarrow} mp'$ is a double pushout transformation in **AHLProc**(**AN**) means that (1) and (2) are pushouts in **AHLProc**(**AN**) which by Corollary 5.11 implies that (lp, cp)

are composable w.r.t. (I, l, c) and (rp, cp) are composable w.r.t. (I, r, c).
This means that (L, C) are composable w.r.t. (I, l, c) and (R, C) are composable w.r.t. (I, r, c). By Corollary 5.7 pushouts (1) and (2) in **AHLNet** are also pushouts in **AHLONet**, i.e. $K \overset{(p,m)}{\Rightarrow} K'$ is double pushout transformation in **AHLONet**.
Hence by Corollary 7.11 p satisfies the extended gluing condition under match m.
Furthermore from the given commutativities follows that

$$\begin{aligned} mp \circ m \circ l &= mp \circ d \circ c \\ &= cp \circ c \\ &= mp' \circ e \circ c \\ &= mp' \circ n \circ r \\ &= mp' \circ rp \end{aligned}$$

Hence pp satisfies the extended gluing condition for AHL-processes under match m.

□

Detailed Proof C.45 (Construction of Direct Transformations of AHL-Occurrence Nets with Instantiations)
See Theorem 7.19

Proof.

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \\ K & C \arrow[l, "a"] \arrow[r, "b"'] & K' \end{tikzcd}$$

”⇒”:

Let $(p, INIT_R, INS_R)$ satisfy the extended gluing condition under match m.
Then the induced preimage $LI = (L, INIT_L, INS_L)$ of KI and m exists and KI is decomposable w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$ leading to cominimal decomposition $CI = (C, INIT_C, INS_C)$ by Theorem 6.17.
Furthermore p satisfies the extended gluing condition under match m implying composition diagrams (1) and (2) which implies that (C, R) are composable w.r.t. (I, c, r).
Let $RI = (R, INIT_R, INS_R)$. Due to the fact that $(p, INIT_R, INS_R)$ satisfies the extended gluing condition there is

$$|RetInit_{(I,c,r)}(INS_C, INS_R)| = |Composable_{(I,c,r)}(INS_C, INS_R)|$$

leading to composition $KI' = CI \circ_{(I,c,r)} RI$.
Hence there is a direct process preserving transformation of AHL-occurrence nets with instantiations $KI \overset{pi,m}{\Rightarrow} KI'$.

”⇐”:

Let $KI \overset{pi,m}{\Rightarrow} KI'$ be a direct process preserving transformation of AHL-occurrence nets with instantiations. Then there are composition diagrams (1) and (2) which by theorem 7.9 implies that p satisfies the extended gluing condition under match m.

Furthermore there is the cominimal decomposition $CI = (C, INIT_C, INS_C)$ of KI w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$ which implies an AHL-occurrence net with instantiations $LI = (L, INIT_L, INS_L)$ together with a surjection $s : INS \to INS_L$ such that for every $L_{init} \in INS$ there is an **INet**-morphism $(m_L, m) : (s(L_{init}), L) \to (L_{init}, K)$. This implies by Lemma 6.21 that LI is the induced preimage of KI and m and by Theorem 6.17 KI is decomposable w.r.t. $I \xrightarrow{l} L \xrightarrow{m} K$.
Finally there is $KI' = CI \circ_{(I,c,r)} RI$ which by theorem 5.25 implies that

$$|RetInit_{(I,c,r)}(INS_C, INS_R)| = |Composable_{(I,c,r)}(INS_C, INS_R)|$$

□

Detailed Proof C.46 (Double Pushout Transformation of AHL-Occurrence Nets with Instantiations)
See Theorem 7.23.

Proof. "$\Rightarrow$":
Due to the fact that $(p, INIT_R, INS_R)$ satisfies the extended gluing condition for AHL-occurrence nets with instantiations under match $m_N : L \to K$ the AHL-occurrence net with instantiations KI is decomposable w.r.t. $I \xrightarrow{l_N} L \xrightarrow{m_N} K$ leading to a cominimal decomposition $CI' = (C', INIT'_C, INS'_C)$.
By Theorem 6.27 there is (C', INS'_C) a maximal pushout complement in **AHLNetI** which by the uniqueness of maximal pushout complements implies that (C, INS_C) and (C', INS'_C) are equal up to isomorphism. Hence for

$$INIT_C = \{IN(L) \mid L \in INS_C\}$$

there is also $CI = (C, INIT_C, INS_C)$ a cominimal decomposition of KI w.r.t. $I \xrightarrow{l_N} L \xrightarrow{m_N} K$.
Then by Corollary 6.19 there is (C, INS_C) also a pushout complement in **AHLONetI**. Furthermore due to Fact 4.27 there are pushouts (3) and (4) in **AHLNet**.

$$\begin{tikzcd}
L \arrow[d, "m_N"'] \arrow[dr, phantom, "(3)"] & I \arrow[l, "l_N"'] \arrow[r, "r_N"] \arrow[d, "c_N"] \arrow[dr, phantom, "(4)"] & R \arrow[d, "n_N"] \\
K & C \arrow[l, "a_N"] \arrow[r, "b_N"'] & K'
\end{tikzcd}$$

Since the extended gluing condition for AHL-occurrence nets with instantiations includes the extended gluing condition for AHL-occurrence nets by Theorem 7.9 pushouts (3) and (4) are composition diagrams.
Theorem 5.5 implies that (C, R) are composable w.r.t. (I, c, r) which together with the fact that

$$|RetInit_{(I,c,r)}(INS_C, INS_R)| = |Composable_{(I,c,r)}(INS_C, INS_R)|$$

implies that $(CI, (R, INIT_R, INS_R))$ are composable w.r.t. (I, c, r). So by Corollary 5.7 there is (2) a pushout in **AHLONetI** and hence $(K, INS) \overset{(pi,m)}{\Rightarrow} (K', INS')$ is a double pushout transformation in **AHLONetI**.

"$\Leftarrow$":

$(K, INS) \overset{(pi,m)}{\Rightarrow} (K', INS')$ is a double pushout transformation in **AHLNetI** and **AHLONetI** means that (1) and (2) are pushouts in **AHLNetI** and **AHLONetI**.

Due to Corollary 6.19 there is $(C, INIT_C, INS_C)$ a decomposition of KI w.r.t. $I \overset{l_N}{\to} L \overset{m_N}{\to} K$ where

$$INIT_C = \{IN(L) \mid L \in INS_C\}$$

So the fact that (C, INS_C) is a maximal pushout complement implies by Theorem 6.27 that $(C, INIT_C, INS_C)$ is a cominimal decomposition of KI w.r.t. $I \overset{l_N}{\to} L \overset{m_N}{\to} K$.

Furthermore the fact that (2) is pushout in **AHLNetI** and **AHLONetI** by Corollary 5.28 implies that

$$KI' = (R, INIT_R, INS_R) \circ_{(I,r,c)} CI$$

From Theorem 7.19 follows that $(p, INIT_R, INS_R)$ satisfies the extended gluing relation for AHL-occurrence nets with instantiations.

□

Detailed Proof C.47 (Double Pushout Transformation of AHL-Processes with Instantiations)
See Theorem 7.28.

Proof.

$$\begin{tikzcd}
(L, INS_L) \arrow[d, "m"] \arrow[ddrr, bend right=60, "lp"'] & (I, INS_I) \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dl, phantom, "(1)"] \arrow[dr, phantom, "(2)"] & (R, INS_R) \arrow[d, "n"] \arrow[dd, bend left, "rp"] \\
(K, INS) \arrow[drr, "mp"'] & (C, INS_C) \arrow[l, "d"'] \arrow[r, "e"] \arrow[dr, "cp"] & (K', INS') \arrow[d, "mp'"] \\
 & & Inst(AN)
\end{tikzcd}$$

"$\Rightarrow$":

Since $(p, INIT_R, INS_R)$ satisfies the extended gluing condition under m from Corollary 7.23 follows that $(K, INS) \overset{(p,m)}{\Rightarrow} (K', INS')$ is a double pushout transformation in **AHLONetI**, i.e. (1) and (2) are pushouts in **AHLONetI**.

Since AHL-morphisms reflect AHL-processes there are process morphisms $ip_N : I \to AN$, $lp_N : L \to AN$ such that the morphisms in diagram (1) and (2) are **AHLProcI**(**AN**) morphisms where ip and lp are the unique morphisms induced by the cofree construction $(Inst(AN), id_{AN})$.

Then there is

$$lp_N \circ l_N = ip_N = cp_N \circ c_N \text{ and } rp_N \circ r_N = ip_N = cp_N \circ c_N$$

i.e. (lp_N, cp_N) are composable w.r.t. (I, l_N, c_N) and (rp_N, cp_N) are composable w.r.t. (I, r_N, c_N). This implies by Corollary 5.34 that there are morphisms $\overline{mp} : (K, INS) \to (AN, \emptyset)$ and $\overline{mp}' : K' \to AN$ such that $\overline{mp}$ is pushout object of (1) and $\overline{mp}'$ is pushout object of (2) in **AHLProcI**(**AN**).

$$\begin{tikzcd}
(L,INS_L) \arrow[d, "m"'] \arrow[ddrr, bend right, "lp"'] \arrow[dr, phantom, "(\mathbf{1})"] & (I,INS_I) \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{1})"] & (R,INS_R) \arrow[d, "n"] \arrow[dd, bend left, "rp"] \\
(K,INS) \arrow[drr, "\overline{mp}"'] & (C,INS_C) \arrow[l, "d"'] \arrow[r, "e"] \arrow[dr, "cp"] & (K',INS') \arrow[d, "\overline{mp}'"] \\
 & & (AN,\emptyset)
\end{tikzcd}$$

Then there is

$$mp \circ m = lp \text{ and } mp \circ d = cp$$

and

$$\overline{mp} \circ m = lp \text{ and } \overline{mp} \circ d = cp$$

which due to the pushout property of (1) implies that $mp = \overline{mp}$. Analogously the pushout property of (2) implie $mp' = \overline{mp}'$.
Hence (1) and (2) with morphisms mp and mp' are pushouts in $\mathbf{AHLProcI}(\mathbf{AN})$, i.e. $mp \overset{(pi,m)}{\Rightarrow} mp'$ is a double pushout transformation in $\mathbf{AHLProcI}(\mathbf{AN})$.

"$\Leftarrow$":
$mp \overset{(pi,m)}{\Rightarrow} mp'$ is a double pushout transformation in $\mathbf{AHLProcI}(\mathbf{AN})$ means that (1) and (2) are pushouts in $\mathbf{AHLProcI}(\mathbf{AN})$ which by Corollary 5.34 implies that the AHL-processes with instantiations (lp_N, cp_N) are composable w.r.t. (I, l_N, c_N) and (rp_N, cp_N) are composable w.r.t. (I, r_N, c_N).
This means that (L, C) are composable w.r.t. (I, l_N, c_N) and (R, C) are composable w.r.t. (I, r_N, c_N) and there is

$$|RetInit_{(I,l_N,c_N)}(INS_L, INS_C)| = |Composable_{(I,l_N,c_N)}(INS_L, INS_C)|$$

and

$$|RetInit_{(I,r_N,c_N)}(INS_R, INS_C)| = |Composable_{(I,r_N,c_N)}(INS_R, INS_C)|$$

i.e. (LI, CI) are composable w.r.t. (I, l_N, c_N) and (RI, CI) are composable w.r.t. (I, r_N, c_N) where $LI = (L, INIT_L, INS_L)$, $CI = (C, INIT_C, INS_C)$ and $RI = (R, INIT_R, INS_R)$ and the sets of initializations are given by the fact that we have AHL-occurrence nets with instantiations.
Then by Corollary 5.29 pushouts (1) and (2) in **AHLNetI** are also pushouts in **AHLONetI**, i.e. $K \overset{(p,m)}{\Rightarrow} K'$ is double pushout transformation in **AHLONetI**.
Hence by Corollary 7.23 $(p, INIT_R, INS_R)$ satisfies the extended gluing condition under match m.

□

C.8 Amalgamation of Algebraic High-Level Processes

Detailed Proof C.48 (Amalgamation Theorem for AHL-Processes)
See Theorem 8.7.

Proof. We define

$$Comp\left(\left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]\right) = [mp_3]$$

where mp_3 is the amalgamation $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$ defined in Theorem 8.4. The amalgamation is defined as the unique morphism induced by the pushout $K_3 = K_1 \circ_{(K_0,\phi_1,\phi_2)} K_2$ in **AHLNet** and since the category **AHLNet** has pushouts and pushouts are unique up to isomorphism the function $Comp$ is well-defined.
Furthermore we define

$$Decomp([mp_3]) = \left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]$$

where $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$ is the amalgamation decomposition of mp_3 defined in Theorem 8.5. The amalgamation decomposition is defined via pullbacks (1)-(4) in **AHLNet** and since the category **AHLNet** has pullbacks and pullbacks are unique up to isomorphisms the function $Decomp$ is well-defined.

$$\begin{tikzcd}
 & & K_0 \arrow[d, "mp_0"] \arrow[ddll, "\phi_1"'] \arrow[ddrr, "\phi_2"] & & \\
 & (\mathbf{3}) & AN_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & (\mathbf{4}) & \\
K_1 \arrow[r, "mp_1"'] \arrow[ddrr, "\psi_1"'] & AN_1 \arrow[dr, "g_1"] & (\mathbf{PO}) & AN_2 \arrow[dl, "g_2"'] & K_2 \arrow[l, "mp_2"] \arrow[ddll, "\psi_2"] \\
 & (\mathbf{1}) & AN_3 & (\mathbf{2}) & \\
 & & K_3 \arrow[u, "mp_3"] & &
\end{tikzcd}$$

Given an agreeing span $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$ with respect to pushout (PO). Then by the properties of agreement projections diagrams (3) and (4) above are pullbacks. The composition $mp_3 : K_3 \to AN_3$ is constructed via the pushout which is the outer square in the diagram above. Then the pushout (PO) is a weak Van Kampen square implying that (1) and (2) are pullbacks in **AHLNet**.
Since the decomposition of mp_3 is constructed via pullback (1)-(4) and pullbacks are unique up to isomorphism the result is isomorphic to $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$, i.e.

$$Decomp\left(Comp\left(\left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]\right)\right) = \left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]$$

Vice versa given an AHL-process $mp_3 : K_3 \to AN_3$ the amalgamation decomposition $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$ of mp_3 is defined via pullbacks (1)-(4) leading to the fact that (PO) is a weak Van Kampen square. This implies that the outer square is a pushout which defines exactly the composition of $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$. Since pushouts are unique up to isomorphism there is

$$Comp(Decomp([mp_3])) = [mp_3]$$

Hence $Comp$ and $Decomp$ are inverse to each other which means that they are bijections. □

Detailed Proof C.49 (Mutual Decomposability of AHL-Occurrence Nets with Instantiations)
See Lemma 8.13.

Proof. Due to the mutual decomposability of KI w.r.t. composition (1) the functions

$$IN_1 : PreIns(g_1)(INS) \to PreInit(g_1)(INS)$$

with

$$IN_1(PreIns(g_1)(L_{init})) = PreInit(g_1)(L_{init})$$

is injective which by Theorem 4.13 implies that the induced preimage $KI_1 = (K_1, INIT_1, INS_1)$ of KI and g_1 exists.
Therefore we can define a function $g_{1,I} : INS \to INS_1$ with

$$g_{1,I}(L_{init}) = PreIns(g_1)(L_{init})$$

which by the fact that for every $L_{init} \in INS$ there is an **INet**-morphism $(g_{1,L}, g_1) : (g_{1,I}(L_{init}), K_1) \to (L_{init}, K)$ implies that $(g_1, g_{1,I})$ is an **AHLNetI**-morphism.
It remains to show that KI is decomposable w.r.t. $K_0 \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$. The existence of the decomposition K_2 of K w.r.t. $K_0 \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ follows from composition (1).
The function

$$IN_2 : PreIns(g_2)(INS) \to PreInit(g_2)(INS)$$

with

$$IN_2(PreIns(g_2)(L_{init})) = PreInit(g_2)(L_{init})$$

is injective due to the mutual decomposability of KI.
Let $L_{init_1} \in INS_1$ and $L_{init} \in INS$. Then there is an instantiation $L_{init'} \in INS$ such that

$$L_{init_1} = PreIns(g_1)(L_{init'}) = g_{1,I}(L_{init'})$$

Let

$$PreIns(f_1)(L_{init_1}) = PreIns(f_1)(g_{1,I}(L_{init}))$$

From the uniqueness of instantiation preimages we obtain

$$\begin{aligned}
PreIns(f_1)(L_{init_1}) &= PreIns(f_1)(g_{1,I}(L_{init})) \\
\Leftrightarrow PreIns(f_1)(g_{1,I}(L_{init'})) &= PreIns(f_1)(g_{1,I}(L_{init})) \\
\Leftrightarrow PreIns(f_1)(PreIns(g_1)(L_{init'})) &= PreIns(f_1)(PreIns(g_1)(L_{init})) \\
\Leftrightarrow PreIns(f_1)(PreIns(g_1)(L_{init'})) &= PreIns(g_1 \circ f_1)(L_{init}) \\
\Leftrightarrow PreIns(f_1)(PreIns(g_1)(L_{init'})) &= PreIns(g_2 \circ f_2)(L_{init}) \\
\Leftrightarrow PreIns(f_1)(PreIns(g_1)(L_{init'})) &= PreIns(f_2)(PreIns(g_2)(L_{init}))
\end{aligned}$$

which means that $PreIns(g_1)(L_{init'})$ and $PreIns(g_2)(L_{init})$ have a common preimage. By Theorem 4.18 the instantiations are compatible with (f_1, f_2) which by Lemma 5.15 implies that

$$\begin{aligned}
&(PreIns(g_1)(L_{init'}), PreIns(g_2)(L_{init})) \in Composable_{(K_0,f_1,f_2)} \\
\Leftrightarrow\ &(L_{init_1}, PreIns(g_2)(L_{init})) \in Composable_{(K_0,f_1,f_2)}
\end{aligned}$$

So from the mutual decomposability of KI follows that there is

$$L_{init_1} \circ_{(J,j_1,j_2)} PreIns(g_2)(L_{init}) \in INS$$

where (J, j_1, j_2) is the instantiation interface induced by (K_0, f_1, f_2).
Hence KI is decomposable w.r.t. $K_0 \xrightarrow{f_1} K_1 \xrightarrow{g_1} K$ and $g_{1,I}$. □

C.9 Process Grammars for Low-Level Petri Nets

Detailed Proof C.50 (Well-definedness of Atomic Transition Process)
See Definition B.6.

Well-definedness. Due to the definition of N_t there is $p \in P$ for $p_i \in P_t$ or $p'_i \in P_t$. Together with the fact that $t \in T$ the functions ap_P and ap_T are well-defined.
Let

$$pre(t) = \sum_{p \in P} \lambda_p p$$

and

$$post(t) = \sum_{p \in P} \mu_p p$$

Then there is

$$\begin{aligned} ap_P^{\oplus} \circ pre_t(t) &= ap_P^{\oplus}(\sum_{p \in P} \sum_{i=1}^{\lambda_p} p_i) \\ &= \sum_{p \in P} \sum_{i=1}^{\lambda_p} ap_P(p_i) \\ &= \sum_{p \in P} \sum_{i=1}^{\lambda_p} p \\ &= \sum_{p \in P} \lambda_p p \\ &= pre(t) \\ &= pre \circ ap_T(t) \end{aligned}$$

and

$$\begin{aligned}
ap_P^{\oplus} \circ post_t(t) &= ap_P^{\oplus}(\sum_{p\in P}\sum_{i=1}^{\mu_p} p'_i) \\
&= \sum_{p\in P}\sum_{i=1}^{\mu_p} ap_P(p'_i) \\
&= \sum_{p\in P}\sum_{i=1}^{\mu_p} p \\
&= \sum_{p\in P} \mu_p p \\
&= post(t) \\
&= post \circ ap_T(t)
\end{aligned}$$

which means that ap is a well-defined PT-net morphism. □

Detailed Proof C.51 (Atomic Processes are Finite Processes)
See Fact B.12.

Proof. Let $mp \in AtomProcs(N)$.

Case 1: $mp = \emptyset : N_\emptyset \to N$
The empty net obviously is a finite occurrence net because it has no places and no transitions. Therefore there is $mp \in Procs(N)$.

Case 2: $mp \in AtomProcs_P(N)$
Then there is $p \in P$ and $mp : N_p \to N$. Since an atomic process net N_p has only one place and no transition it also is a finite occurrence net and hence $mp \in Procs(N)$.

Case 3: $mp \in AtomProcs_T(N)$
Then there is $t \in T$ and $mp : N_t \to N$.
Let us assume that N_t is not finite. Since t is the only transition in N_t this implies that there is an infinite set P_t of places which means that there is an infinite number of places in the pre or post domain of t. Without any loss of generality we assume that there is an infinite set P' of places in the pre domain of t.
Due to the fact that P' is infinte there is also $\sum_{p\in P'} p$ an infinte sum and therefore $mp_P^{\oplus}(\sum_{p\in P'} p)$ an infinite sum of places. So together with the fact that P/T-morphisms preserve pre and post domains we have

$$\begin{aligned}
mp_P^{\oplus}(\sum_{p\in P'} p) &= mp_P^{\oplus} \circ pre_t(t) \\
&= pre_N(mp_T(t))
\end{aligned}$$

Since the arc descriptions in P/T-nets are natural numbers it follows that there is an infinite set of places in the pre domain of $mp_T(t)$ which contradicts the fact that N is finite.
It remains to show that N_t is unary, it has no conflicts and the causal relation is a finitary strict partial order.

unarity:
The only transition in N_t is the transition t. Due to the definitions of pre_t and $post_t$ there is exactly one arc between t and every place $p \in P_t$. So N_t is unary.

conflicts:
Due to the fact that N_t has exactly one transition it is not possible that N_t has a conflict.

irreflexivity of the causal relation:
Let us assume that the causal relation $<_{N_t}$ is not irreflexive, i.e. there is $x \in P_t \uplus T_t$ with $x <_{N_t} x$. Since t is the only transition in N_t this implies that the pre and post domains of t are not disjoint, i.e. there is $x \in P_t$ such that

$$x \leq pre_t(t)$$

which means that x has the form $x = p_i$ with $p \in P$ and $1 \leq i \leq \lambda_p$ and also

$$x \leq post_t(t)$$

which means that x has the form $x = p'_i$ with $p \in P$ and $1 \leq i \leq \mu_p$. This is a contradiction because x has either the one form or the other.
Hence $<_{N_t}$ is irreflexive.

finitarity of the causal relation:
Due to the irreflexivity of $<_{N_t}$ together with the fact that N_t contains only one transition an element in N_t has at most two predecessor elements.
Hence $<_{N_t}$ is finitary.

□

Detailed Proof C.52 (Generation of Finite Process Semantics by Composition)
See Theorem B.13.

Proof. $GenProcs(N) \subseteq FinProcs(N)$:
Let $mp \in GenProcs(N)$. We have to show that $mp \in FinProcs(N)$. The recursive definition of $GenProcs(N)$ allows us to do a structural induction.

Induction Basis: $mp = \emptyset : N_\emptyset \to N$
Then there is $mp \in AtomProcs(N)$ and hence $mp \in FinProcs(N)$ due to Fact B.12.

Induction Hypothesis: $mp_2 : K_1 \to N \in FinProcs(N)$

Induction Step: $mp = mp_1 \circ_{(I,i_1,i_2)} mp_2$, $mp_1 \in AtomProcs^+(N)$. From Fact B.12 follows that $mp_1 \in FinProcs(N)$ which together with the induction hypothesis means that mp_1 and mp_2 are finite processes of mp. Since the composition of processes is a process there is $mp \in Procs(N)$.
mp is finite because it has at most as many places and transitions as mp_1 and mp_2 together.

$FinProcs(N) \subseteq GenProcs(N)$:
Let $mp : K \to N \in FinProcs(N)$. For $mp = \emptyset : N_\emptyset \to N$ there obviously is $mp \in GenProcs(N)$.

So let us consider the case that $mp \neq \emptyset : N_\emptyset \to N$. Then we have to show that mp can be generated by a sequence of compositions as shown below where $K_n = N_\emptyset$, $A_1, \ldots A_n \in AtomProcs(N)$, $I_x = AtomInterface(N, A_x)$, i_x inclusions for $1 \leq x \leq n$ and there are corresponding process morphisms for every net.

$$\begin{tikzcd}
I_n \arrow[r, "i_n"] \arrow[d, "j_n"] & A_n \arrow[dr, "f_n"] & I_{n-1} \arrow[r, "i_{n-1}"] \arrow[d, "j_{n-1}"] & \cdots & {} \arrow[r, "i_2"] & A_2 \arrow[dr, "f_2"] & I_1 \arrow[r, "i_1"] \arrow[d, "j_1"] & A_1 \arrow[dr, "f_1"] & \\
K_n \arrow[rr, "g_n"'] & & K_{n-1} \arrow[r, "g_{n-1}"'] & \cdots \arrow[rrr, "g_2"'] & & & K_1 \arrow[rr, "g_1"'] & & K
\end{tikzcd}$$

Since composition and decomposition are inverse to each other alternively we can show that every non-empty process mp can be stepwise decomposed with respect to atomic processes. Deleting an atomic non-empty part of the process in every step it is sufficient to show that for an arbitrary process mp there exists one single decomposition step (1) where mp_1 is an atomic process. That leads to smaller processes in every step finally resulting in the empty process.

$$\begin{tikzcd}
I \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "(\mathbf{1})"] & K_1 \arrow[d, "g_1"] \arrow[ddr, bend left, "mp_1"] & \\
K_2 \arrow[r, "g_2"'] \arrow[drr, bend right, "mp_2"'] & K \arrow[dr, "mp"] & \\
& & N
\end{tikzcd}$$

<u>*Case 1*</u>: $T_K = \emptyset$

Considering only non-empty processes we know that the net K is not empty.
So we can choose an arbitrary place $\tilde{p} \in P_K$ with $mp_P(\tilde{p}) = p$.
Let $K_1 = N_p$ and $mp_1 = ap : N_p \to N$. Furthermore let $g_1 : K_1 \to K$ with $g_1(\tilde{p}) = p$. g_1 obviously is a well-defined P/T-morphism since $T_{K_1} = \emptyset$ and it is injective because $|P_{K_1}| = 1$.
For

$$I = AtomInterface(N, N_p) = N_\emptyset$$

the dangling condition is satisfied because due to $T_K = \emptyset$ there is

$$DP = \emptyset \subseteq GP$$

Hence there exists a decomposition K_2 of K w.r.t. $I \xrightarrow{\emptyset} K_1 \xrightarrow{g_1} K$ together with a process morphism $mp_2 : K_2 \to N$ with $mp_2 = mp \circ g_2$.

<u>*Case 2*</u>: $T_K \neq \emptyset$

In order to avoid confusion between the sums of the commutative monoid $\oplus$ and sums of natural numbers we will in the following use the symbol $\bigoplus$ instead of $\sum$ for sums of the commutative monoid.
We choose an arbitrary transition $\tilde{t} \in T_K$ with $mp_T(\tilde{t}) = t$ and

$$pre(t) = \bigoplus_{p \in P} \lambda_p p$$

$$post(t) = \bigoplus_{p \in P} \mu_p p$$

Let $K_1 = N_t$ and $mp_1 = ap : N_t \to N$.
Due to the fact that N_t is an occurrence net it is unary and hence for

$$pre_K(\tilde{t}) = \bigoplus_{p \in P_K} \eta_p p$$

there is $\eta_i \leq 1$ for every $p \in P_t$. This means that there is a set

$$J \subseteq P_K$$

such that

$$pre_K(\tilde{t}) = \bigoplus_{p \in J} p$$

Since P/T-net morphisms preserve pre domains we have

$$\begin{aligned}
\bigoplus_{p \in P} \lambda_p p &= pre(t) \\
&= pre(mp_T(\tilde{t})) \\
&= mp_P^{\oplus}(pre_K(\tilde{t})) \\
&= mp_P^{\oplus}(\bigoplus_{p \in J} p) \\
&= \bigoplus_{p \in J} mp_P(p)
\end{aligned}$$

and therefore

$$|\bigoplus_{p \in P} \lambda_p p| = |\bigoplus_{p \in J} mp_P(p)| = |\bigoplus_{p \in J} p|$$

which means that

$$\sum_{p \in P} \lambda_p = |J|$$

and hence there exists a bijection

$$b : \{1, \ldots, \sum_{p \in P} \lambda_p\} \xrightarrow{\sim} J$$

In order to address specific places by numbers let there be a bijection

$$a : \{1, \ldots, |P|\} \xrightarrow{\sim} P$$

and furthermore let

$$(c_{i,j})_{i \in \{1,\ldots,|P|\}, j \in \{1,\ldots,\lambda_{a(i)}\}}$$

with

$$c_{i,j} = \sum_{k=1}^{i-1} \lambda_{a(k)} + j$$

This allows us to address the elements in $pre_K(\tilde{t})$ by

$$pre_K(\tilde{t}) = \bigoplus_{p \in J} p = \bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_{a(i)}} b(c_{i,j})$$

The fact that

$$\begin{aligned}
\bigoplus_{i=1}^{|P|} mp_P^{\oplus}(\bigoplus_{j=1}^{\lambda_{a(i)}} b(c_{i,j})) &= mp_P^{\oplus}(\bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_{a(i)}} b(c_{i,j})) \\
&= mp_P^{\oplus}(\bigoplus_{p \in J} p) \\
&= mp_P^{\oplus}(pre_K(\tilde{t})) \\
&= pre(t) \\
&= \bigoplus_{p \in P} \lambda_p p \\
&= \bigoplus_{i=1}^{|P|} \lambda_{a(i)} a(i)
\end{aligned}$$

implies that for $i \in \{1, \ldots, |P|\}$ there are $\lambda_{a(i)}$ places in the pre domain of $\tilde{t}$ which are mapped to $a(i)$ by mp_P. Therefore we can choose b to be in the way that for $i \in \{1, \ldots, |P|\}, j \in \{1, \ldots \lambda_{a(i)}\}$ there is

$$mp_P(b(c_{i,j})) = a(i)$$

Analogously to the pre domain of $\tilde{t}$ in K we obtain for the post domain

$$post_K(\tilde{t}) = \bigoplus_{p \in P_K} \sigma_p p$$

a set $J' \subseteq P_K$ such that

$$post_K(\tilde{t}) = \bigoplus_{p \in J'} p$$

and for

$$post(t) = \bigoplus_{p \in P} \mu_p p$$

there is a bijection

$$b' : \{1, \ldots, \sum_{p \in P} \mu_p\} \xrightarrow{\sim} J'$$

Furthermore let

$$(c'_{i,j})_{i \in \{1,\ldots,|P|\}, j \in \{1,\ldots,\mu_{a(i)}\}}$$

with

$$c'_{i,j} = \sum_{k=1}^{i-1} \mu_{a(k)} + j$$

which allows us to address the elements in $post_K(\tilde{t})$ by

$$post_K(\tilde{t}) = \bigoplus_{p \in J'} p = \bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_{a(i)}} b'(c'_{i,j})$$

The fact that

$$\begin{aligned}
\bigoplus_{i=1}^{|P|} mp_P^{\oplus}(\bigoplus_{j=1}^{\mu_{a(i)}} b'(c'_{i,j})) &= mp_P^{\oplus}(\bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\mu_{a(i)}} b'(c'_{i,j})) \\
&= mp_P^{\oplus}(\bigoplus_{p \in J'} p) \\
&= mp_P^{\oplus}(post_K(\tilde{t})) \\
&= post(t) \\
&= \bigoplus_{p \in P} \mu_p p \\
&= \bigoplus_{i=1}^{|P|} \mu_{a(i)} a(i)
\end{aligned}$$

implies that for $i \in \{1, \ldots, |P|\}$ there are $\mu_{a(i)}$ places in the post domain of $\tilde{t}$ which are mapped to $a(i)$ by mp_P. Therefore we can choose b' to be in the way that for $i \in \{1, \ldots, |P|\}, j \in \{1, \ldots \mu_{a(i)}\}$ there is

$$mp_P(b'(c'_{i,j})) = a(i)$$

Now we have everything that we need to define a P/T-net morphism $g_1 : K_1 \to K$ with

$$g_{1,T}(t) = \tilde{t}$$

and

$$g_{1,P}(a(i)_j) = b(c_{i,j})$$
$$g_{1,P}(a(i)'_j) = b'(c'_{i,j})$$

well-definedness of $g_{1,T}$:
The function $g_{1,T}$ is well-defined because $T_t = \{t\}$.

well-definedness of $g_{1,P}$:
The function $g_{1,P}$ is well-defined because a is a bijection and thus surjective. Furthermore $p_j \in P_t$ implies $1 \leq j \leq \lambda_p$ which means that

$$g_{1,P}(a(i)_j) = b(c_{i,j}) = b(\sum_{k=1}^{i-1} \lambda_{a(k)} + j) \in J \subseteq P_K$$

and $p'_j \in P_t$ implies $1 \leq j \leq \mu_p$ which means that

$$g_{1,P}(a(i)'_j) = b'(c'_{i,j}) = b'(\sum_{k=1}^{i-1} \mu_{a(k)} + j) \in J' \subseteq P_K$$

well-definedness of g_1:

In order that g_1 is a P/T-morphism it remains to show that it preserves pre and post domains:

Let

$$S = \{1, \ldots, \sum_{i=1}^{|P|} \lambda_a(i)\}$$

and

$$S' = \{1, \ldots, \sum_{i=1}^{|P|} \mu_a(i)\}$$

Then there is

$$\begin{aligned}
g_{1,P}^{\oplus}(pre_t(t)) &= g_{1,P}^{\oplus}(\bigoplus_{p \in P} \bigoplus_{j=1}^{\lambda_p} p_j) \\
&= g_{1,P}^{\oplus}(\bigoplus_{i=1}^{|P|} \bigoplus_{i=1}^{\lambda_a(i)} a(i)_j) \\
&= \bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_a(i)} g_{1,P}(a(i)_j) \\
&= \bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_a(i)} b(c_{i,j}) \\
&= \bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_a(i)} b(\sum_{k=1}^{i-1} \lambda_{a(k)} + j) \\
&= b^{\oplus}(\bigoplus_{i=1}^{|P|} \bigoplus_{j=1}^{\lambda_a(i)} \sum_{k=1}^{i-1} \lambda_{a(k)} + j) \\
&= b^{\oplus}(\bigoplus_{i \in S} i) \\
&= \bigoplus_{p \in J} p \\
&= pre_K(\tilde{t}) \\
&= pre_K(g_{1,T}(t))
\end{aligned}$$

and

$$\begin{aligned}
g_{1,P}^{\oplus}(post_t(t)) &= g_{1,P}^{\oplus}(\bigoplus_{p\in P}\bigoplus_{j=1}^{\mu_p} p'_j) \\
&= g_{1,P}^{\oplus}(\bigoplus_{i=1}^{|P|}\bigoplus_{i=1}^{\mu_a(i)} a(i)'_j) \\
&= \bigoplus_{i=1}^{|P|}\bigoplus_{j=1}^{\mu_a(i)} g_{1,P}(a(i)'_j) \\
&= \bigoplus_{i=1}^{|P|}\bigoplus_{j=1}^{\mu_a(i)} b'(c'_{i,j}) \\
&= \bigoplus_{i=1}^{|P|}\bigoplus_{j=1}^{\mu_a(i)} b'(\sum_{k=1}^{i-1}\mu_{a(k)} + j) \\
&= b'^{\oplus}(\bigoplus_{i=1}^{|P|}\bigoplus_{j=1}^{\mu_a(i)}\sum_{k=1}^{i-1}\mu_{a(k)} + j) \\
&= b'^{\oplus}(\bigoplus_{i\in S'} i) \\
&= \bigoplus_{p\in J} p \\
&= post_K(\tilde{t}) \\
&= post_K(g_{1,T}(t))
\end{aligned}$$

injectivity of $g_{1,T}$: The function $g_{1,T}$ obviously is injective because $|T_t| = 1$.

injectivity of $g_{1,P}$: Let $p_1, p_2 \in P_t$ with

$$g_{1,P}(p_1) = g_{1,P}(p_2)$$

Since all places of N_t are either in pre_t or $post_t$ there are the following three cases possible:

<u>*Case 1:*</u> $p_1, p_2 \leq pre_t(t)$

In this case there are $i_1, i_2 \in \{1, \ldots, |P|\}$ and $j_1 \in \{1, \ldots, \lambda_{a(i_1)}\}$, $j_2 \in \{1, \ldots, \lambda_{a(i_2)}\}$ such that

$$p_1 = a(i_1)_{j_1}$$

$$p_2 = a(i_2)_{j_2}$$

The fact that

$$b(c_{i_1,j_1}) = g_{1,P}(p_1) = g_{1,P}(p_2) = b(c_{i_2,j_2})$$

implies

$$c_{i_1,j_1} = c_{i_2,j_2}$$

because b is a bijection and hence injective. This implies that

$$\sum_{k=1}^{i_1-1} \lambda_{a(k)} + j_1 = \sum_{k=1}^{i_2-1} \lambda_{a(k)} + j_2$$

Let us assume that $i_1 \neq i_2$, i.e. $i_1 < i_2$ or $i_1 > i_2$. Without any loss of generality let $i_1 < i_2$, i.e. $i_1 + 1 + d = i_2$ with $d \in \mathbb{N}$.
Since $1 \leq j_1 \leq \lambda_{a(i_1)}$ and $1 \leq j_2$ there is

$$\begin{aligned} c_{i_1,j_1} &= \sum_{k=1}^{i_1-1} \lambda_{a(k)} + j_1 \\ &\leq \sum_{k=1}^{i_1} \lambda_{a(k)} \\ &< \sum_{k=1}^{i_1} \lambda_{a(k)} + 1 \\ &\leq \sum_{k=1}^{i_1+d} \lambda_{a(k)} + j_2 \\ &= c_{i_2,j_2} \end{aligned}$$

contradicting the fact that $c_{i_1,j_1} = c_{i_2,j_2}$. So we have that $i_1 = i_2$ and hence

$$\begin{aligned} & c_{i_1,j_1} = c_{i_2,j_2} \\ \Leftrightarrow\ & \sum_{k=1}^{i_1-1} \lambda_{a(k)} + j_1 = \sum_{k=1}^{i_2-1} \lambda_{a(k)} + j_2 \\ \Rightarrow\ & j_1 = j_2 \end{aligned}$$

Thus $p_1 = p_2$.

Case 2: $p_1, p_2 \leq post_t(t)$

In this case there are $i_1, i_2 \in \{1, \ldots, |P|\}$ and $j_1 \in \{1, \ldots, \mu_{a(i_1)}\}$, $j_2 \in \{1, \ldots, \mu_{a(i_2)}\}$ such that

$$p_1 = a(i_1)'_{j_1}$$

$$p_2 = a(i_2)'_{j_2}$$

The proof that also in this case $i_1 = i_2$ and $j_1 = j_2$ works completely analogously to case 1 because also b' is a bijection and c'_{i_1,i_2} is defined analogously to c_{i_1,i_2}.

Case 3: $p_1 \leq pre_t(t)$ and $p_2 \leq post_t(t)$

Since P/T-net morphisms preserve pre and post domains it follows that

$$g_{1,P}(p_1) \leq pre_K(g_{1,T}(t))$$

and

$$g_{1,P}(p_2) \leq post_K(g_{1,T}(t))$$

which means that $g_{1,P}(p_1) <_K g_{1,P}(p_2)$. By the assumption that $g_{1,P}(p_1) = g_{1,P}(p_2)$ this contradicts the fact that $<_K$ is irreflexive because K is an occurrence net.

injectivity of g_1:
Due to the injectivity of the functions $g_{1,P}$ and $g_{2,P}$ the P/T-net morphism $g_1 = (g_{1,P}, g_{1,T})$ is injective.

interface:

$$I = AtomInterface(N, N_t)$$

Since I is a subnet of $K_1 = P_t$ there is an inclusion $f_1 : I \to K_1$.

dangling condition:
Since $P_I = P_t$ there is

$$GP = f_{1,P}(P_I) = P_t$$

which means that

$$DP \subseteq P_t = GP$$

i.e. the dangling condition is satisfied.

$mp \circ g_1 = mp_1$:

$$\begin{array}{rcl} mp_T \circ g_{1,T}(t) & = & mp_T(\tilde{t}) \\ & = & t \\ & = & mp_{1,T}(t) \end{array}$$

$$\begin{array}{rcl} mp_P \circ g_{1,P}(a(i)_j) & = & mp_P(b(c_{i,j})) \\ & = & a(i) \\ & = & mp_{1,P}(a(i)_j) \end{array}$$

$$\begin{array}{rcl} mp_P \circ g_{1,P}(a(i)'_j) & = & mp_P(b'(c'_{i,j})) \\ & = & a(i) \\ & = & mp_{1,P}(a(i)'_j) \end{array}$$

So there exists a decomposition K_2 of $I \stackrel{f_1}{\to} K_1 \stackrel{g_1}{\to} K$ together with a process morphism $mp_2 : K_2 \to N$ with $mp_2 = mp \circ g_2$.

□

Detailed Proof C.53 (Generation of Process Semantics by Grammar)
See Theorem B.19.

Proof. ”$\subseteq$”:
From fact B.16 follows that $PL(PSG(N)) \subseteq Procs(N)$.
Every transformation step consists of two compositions (1) and (2).

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(\mathbf{1})"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(\mathbf{2})"] & R \arrow[d, "n"] \\
N & C \arrow[l, "d"] \arrow[r, "e"'] & N'
\end{tikzcd}$$

Since a decomposition means the deletion of places or transitions it obviously again leads to a finite process. By Fact B.13 also the composition leads to a finite process.

"$\supseteq$":

Let $mp : K \to N \in FinProcs(N)$ and $PSG(N) = (sp, PR)$.
We have to show that there is a sequence of transformations $sp \overset{*}{\Rightarrow}_{PR} mp$.
This means that there are composition diagrams $(1), (1'), \ldots, (n), (n')$ where $K_n = N_\emptyset$, $(I_x \overset{id_{I_x}}{\leftarrow} I_x \overset{i_x}{\to} A_x, ap_x : A_x \to N) \in PR$ for $1 \le x \le n$ and there are corresponding process morphisms for every net.

$$\begin{tikzcd}
I_n \arrow[d, "m_n"] \arrow[dr, phantom, "(\mathbf{n})"] & I_n \arrow[l, "id_{I_n}"'] \arrow[r, "i_n"] \arrow[d, "j_n"] & A_n \arrow[dr, "f_n"] \arrow[d, phantom, "(\mathbf{n'})"] & I_{n-1} \arrow[d, "m_{n-1}"] \arrow[dr, phantom, "(\mathbf{n-1})"] & I_{n-1} \arrow[l, "id_{I_{n-1}}"'] \arrow[r, "i_{n-1}"] \arrow[d, "j_{n-1}"] \arrow[dr, phantom, "(\mathbf{n-1'})"] & \cdots \\
K_n & K_n \arrow[l, "h_n"] \arrow[rr, "g_n"'] & & K_{n-1} & K_{n-1} \arrow[l, "h_{n-1}"] \arrow[r, "g_{n-1}"'] & \cdots
\end{tikzcd}$$

$$\begin{tikzcd}
\cdots \arrow[r, "i_3"] & A_3 \arrow[dr, "f_3"] \arrow[d, phantom, "(\mathbf{3'})"] & I_2 \arrow[d, "m_2"] \arrow[dr, phantom, "(\mathbf{2})"] & I_2 \arrow[l, "id_{I_2}"'] \arrow[r, "i_2"] \arrow[d, "j_2"] & A_2 \arrow[dr, "f_2"] \arrow[d, phantom, "(\mathbf{2'})"] & I_1 \arrow[d, "m_1"] \arrow[dr, phantom, "(\mathbf{1})"] & I_1 \arrow[l, "id_{I_1}"'] \arrow[r, "i_1"] \arrow[d, "j_1"] & A_1 \arrow[dr, "f_1"] \arrow[d, phantom, "(\mathbf{1'})"] & \\
\cdots \arrow[rr, "g_3"'] & & K_2 & K_2 \arrow[l, "h_2"] \arrow[rr, "g_2"'] & & K_1 & K_1 \arrow[l, "h_1"] \arrow[rr, "g_1"'] & & K
\end{tikzcd}$$

Let $K_0 = K$. Then from Theorem B.13 we obtain composition diagrams (x') for $1 \le x \le n$ with corresponding process morphisms ap_x, mp_x where

$$mp_x \circ f_x = ap_x \text{ and } mp_{x-1} \circ g_x = mp_x \quad (*)$$

$$\begin{tikzcd}
I_x \arrow[r, "i_x"] \arrow[d, "j_x"] \arrow[dr, phantom, "(\mathbf{x'})"] & A_x \arrow[d, "f_x"] \arrow[ddr, bend left, "ap_x"] & \\
K_x \arrow[r, "g_x"] \arrow[drr, bend right, "mp_x"'] & K_{x-1} \arrow[dr, "mp_{x-1}"] & \\
& & N
\end{tikzcd}$$

For $1 \le x \le n$ we choose $m_x = j_x$ and $h_x = id_{K_x}$ which leads to the fact that diagram (x) is a pushout diagram.

$$\begin{tikzcd}
I_x \arrow[d, "j_x"] \arrow[dr, phantom, "(\mathbf{x})"] & I_x \arrow[l, "id_{I_x}"'] \arrow[d, "j_x"] \\
K_x & K_x \arrow[l, "id_{K_x}"]
\end{tikzcd}$$

Since for every $A_x \in AtomProcNets(N)$ and $I_x = AtomInterface(N, A_x)$ there is $T_{I_x} = \emptyset$ it follows that for every $p \in P_{I_x}$ there is $p \in IN(I_x)$ and $p \in OUT(I_x)$. Furthermore from composition diagram (x') follows that j_x is injective.
Hence (I_x, K_x) are composable w.r.t. (I_x, id_{I_x}, j_x) which means that pushout diagrams (x) are composition diagrams.
So for $1 \leq x \leq n$ with composition diagrams $(x), (x')$ we have a direct process preserving production

$$K_x \overset{(p_x, j_x)}{\Rightarrow} K_{x-1}$$

via production

$$p_x : I_x \overset{id_{I_x}}{\longleftarrow} I_x \overset{i_x}{\longrightarrow} A_x$$

It remains to show that the production is also a process preserving production of processes which follows from the equation:

$$\begin{aligned} mp_x \circ j_x \circ id_{I_x} &= mp_x \circ j_x \\ &\overset{(*)}{=} mp_{x-1} \circ g_x \circ j_x \\ &\overset{(x')}{=} mp_{x-1} \circ f_x \circ i_x \\ &\overset{(*)}{=} ap_x \circ i_x \end{aligned}$$

$\square$

List of Figures

1 AHL-net $Alarm$ . . . 8
2 AHL-occurrence net $K_{Heat/Light}$ . . . 11
3 AHL-net GE . . . 15
4 Part of the Flattening $Flat(GE)$ of the Net GE . . . 15
5 Skeleton $Skel(GE)$ of the Net GE . . . 16
6 Instantiation L_{init_1} of $K_{Heat/Light}$ . . . 22
7 Sequential Composition of AHL-Occurrence Nets $K = K_1 \circ_{(I,i_1,i_2)} K_2$ . . . 26
8 Sequential Composition of Instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ . . . 28
9 Instantiations of KI_1 (Sequential Composition) . . . 29
10 Instantiations of KI_2 (Sequential Composition) . . . 30
11 AHL-Occurrence Net with Instantiations $KI_{Heat/Light}$ . . . 32
12 AHL-Occurrence Net with Instantiations $KI_{Light/Heat}$ . . . 33
13 AHL-Occurrence Net with Instantiations KI_{Heat} . . . 35
14 AHL-Occurrence Net with Instantiations KI_{Light} . . . 35
15 AHL-morphism f_H and corresponding instantiation morphism f_L . . . 39
16 Pullback diagram in **PTNet** . . . 41
17 Pullback diagram in **SET** . . . 44
18 AHL-occurrence net with instantiations KI_2 . . . 47
19 AHL-occurrence net with instantiations KI_1 . . . 48
20 AHL-morphisms $f_{1,H}, f_{2,H}$ with epi-mono-factorization . . . 50
21 Compatibility of (L_1, L_2) with $(f_{1,H}, f_{2,H})$ . . . 50
22 Instantiation interface . . . 51
23 Pushout in **INet** . . . 53
24 Instantiations of I . . . 57
25 Composition of AHL-Occurrence Nets $K = K_1 \circ_{(I,i_1,i_2)} K_2$ . . . 65
26 Composition of Instantiations $L_{init} = L_{init_1} \circ_{(J,j_1,j_2)} L_{init_2}$ . . . 72
27 Instantiations of KI_1 (Composition) . . . 78
28 Instantiations of KI_2 (Composition) . . . 79
29 Production for AHL-Occurrence Nets $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ . . . 106
30 Direct Transformation $K \overset{(p,m)}{\Rightarrow} K'$ . . . 110
31 Instantiations of K, R and K', respectively . . . 116
32 Amalgamation of AHL-Processes . . . 126
33 Overview of the main results of this thesis . . . 132
34 Extension diagram of AHL-processes . . . 135
35 Local Church Rosser diagram of processes . . . 135
36 Independence of transformations . . . 136

References

[BCEH01] Paolo Baldan, Andrea Corradini, Hartmut Ehrig, and Reiko Heckel. Compositional modeling of reactive systems using open nets. In *CONCUR '01: Proceedings of the 12th International Conference on Concurrency Theory*, pages 502–518, London, UK, 2001. Springer-Verlag.

[DMM89] Pierpaolo Degano, José Meseguer, and Ugo Montanari. Axiomatizing Net Computations and Processes. In *Proc. on Logic in Computer Science (LICS)*, pages 175–185. IEEE Computer Society, 1989.

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

[EHGP09] H. Ehrig, K. Hoffmann, K. Gabriel, and J. Padberg. Composition and independence of high-level net processes. *Electronic Notes in Theoretical Computer Science*, 242(2):59 – 71, 2009. Proceedings of the First Workshop on Formal Methods for Wireless Systems (FMWS 2008).

[EHP+02] Hartmut Ehrig, Kathrin Hoffmann, Julia Padberg, Paolo Baldan, and Reiko Heckel. High-level net processes. In *Formal and Natural Computing*, volume 2300 of *LNCS*, pages 191–219. Springer, 2002.

[Ehr05] H. Ehrig. Behaviour and Instantiation of High-Level Petri Net Processes. *Fundamenta Informaticae*, 65(3):211–247, 2005.

[Eng91] J. Engelfriet. Branching Processes of Petri Nets. *Acta Informatica*, 28(6):575–591, 1991.

[GR83] Ursula Goltz and Wolfgang Reisig. The Non-sequential Behavior of Petri Nets. *Information and Control*, 57(2/3):125–147, 1983.

[KMO+05] Hans-Jörg Kreowski, Ugo Montanari, Fernando Orejas, Grzegorz Rozenberg, and Gabriele Taentzer, editors. *Formal Methods in Software and Systems Modeling.* Springer-Verlag, 2005.

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

[MMS97] J. Meseguer, U. Montanari, and V. Sassone. On the Semantics of Place/Transition Petri Nets. *Mathematical Structures in Computer Science*, 7(4):359–397, 1997.

[RON09] RON Editor Website. *http://tfs.cs.tu-berlin.de/roneditor/*, December 2009.

[Roz87] Grzegorz Rozenberg. Behaviour of Elementary Net Systems. In *Petri Nets: Central Models and Their Properties, Advances in Petri Nets*, volume 254 of *LNCS*, pages 60–94. Springer, 1987.