1 Interface Based Composition of Reactive Systems

1.1 Reactive Systems.

Large systems, in particular software embedded systems, are usually composed of (smaller) components. A component typically operates autonomously to some extent, and is equipped with an interface to establish some kind of cooperation with other components. Cooperating components may synchronously or asynchronously exchange messages or jointly perform steps. Systems consisting of this kind of co-operating components are frequently denoted as reactive. Typical examples include embedded systems, communication protocols, service-oriented architectures, multi-agent systems, and computerized business processes. The behavior of a reactive systems can not adequately be modelled, represented, or abstracted as a classical function that would start computing after receiving its input, and that eventually would finish computing with delivering the computed result. The two main reasons are:

1. A single meaningful run (computation, execution) of a reactive system usually does not terminate, whereas a non-terminating run of a classical function is flawed.
2. The algorithmic idea of a reactive system may be located not in the behavior of its components, but in their communication and co-operation. The realm of classical functions is not aware of the concepts of communication and cooperation.

As mentioned at the outset, cooperation of components is a pivotal structural principle of reactive systems. In a more technical setting, cooperation of two components $A$ and $B$ is then organized as some kind of composition, $A \cdot B$, such that $A \cdot B$ is again a component. A system is then just a component that may be composed of some (more elementary) components.

A lot of modelling techniques have been suggested to cope with this kind of systems. Pertaining examples include Petri Nets in the 1970ies; process algebras, statecharts and the ACTOR model in the 1980ies, multi-agent systems in the 1990ies, and service-oriented architectures in the 2000s.
1.2 The Quest for Associative Composition.

In general, more than two components are to be composed. For example, two services $S$ and $T$ may be linked by an adapter component $A$, resulting in the system

$$S \cdot A \cdot T.$$  \hfill (1)

A further example is a business supply chain

$$C_1 \cdots C_n$$  \hfill (2)

of producers, traders, retailers etc., for goods and services. The composed systems (1) and (2) already indicate a fundamental requirement for proper composition: composition should be associative. For example, in (1), $(S \cdot A) \cdot T$ and $S \cdot (A \cdot T)$ should be identical. In fact, brackets can be skipped only in case composition is associative. Associative composition is indeed intended or required in most areas where more than two components are composed. For example, composition of classical functions is associative (i.e. $((f \circ g) \circ h)(x) = (f \circ (g \circ h))(x)$ for functions $f,g,h$ and arguments $x$). But virtually all formalisms for reactive systems struggle with associative composition. Some formalisms do with composition operators that are associative in special cases only. This includes LOTOS, I/O-Automata and open Petri Nets (OPN). Others suggest somewhat unintuitive composition operators, in particular several versions of process algebras. Their parallel composition operator yields a lot of nondeterminism, where the intended synchronization is just one option. Synchronization can however be enforced by means of the hiding operator. Combining parallel composition and hiding into one operator, intuitively represents what “parallel composition” is intended to express. But this operator is not associative. So, the quest for adequate, in particular associative composition operators for reactive systems has found only non-satisfactory answers so far.

Associativity can be gained only at a price. In this paper we will show that this price can be kept low.

1.3 Interface-based associative composition.

We suggest a composition operator for any kind of components that come with interfaces, where an interface is a set of labelled elements. Interface-based composition of two components $C$ and $D$ then is achieved by “glueing” (overlying, unifying) identically labelled elements of the interfaces of $C$ and $D$. Two such glued interface elements then turn into one internal element of the composed system $C \cdot D$, as sketched in Figure 1.

Systems and components described by many modelling techniques exhibit this kind of interfaces, or they can easily and intuitively convincingly be equipped by such interfaces. Examples include various versions of process algebras, data flow graphs, bigraphs, control flow graphs, network layouts, automata, BPMN-models, open Petri Nets, etc. As outlined in Sec. 1.2 already, associativity of
composition of such systems is highly desirable, but difficult to achieve. The simple composition of components as in Figure 1 is in general not associative, as Fig. 2 shows.

In this paper we define a version of components with interfaces, together with a composition operator, such that

- the composition $C \cdot D$ of two components $C$ and $D$ is a component again,
- any two components can be composed,
- composition is associative, i.e.

$$ (C \cdot D) \cdot E = C \cdot (D \cdot E) $$

for all components $C$, $D$ and $E$.

To motivate the forthcoming constructs, we start with the observation that associativity is about three or more components. In this case, each "inner" components (e.g. $A$ in (1), and $C_2, \cdots, C_{n-1}$ in (2), and $D$ in (3) has a left and a right partner (e.g. $C$ and $E$, respectively, in (3)). These partners frequently play different roles. For example, in a composition $f \circ g \circ h$ of functions, the left partner $f$ of $g$ provides the arguments for $g$, and the right partner $h$ of $g$ receives the result of $g$. In a business supply chain $C_1, \cdots, C_n$, each components $C_i$ has a sell side (e.g. to its left) and a buy side (to its right). The buy side of $C_{i-1}$ is "glued" with the sell side of $C_i$. So, it is overly intuitive to assume the interface of a component $C$ to be partitioned into two ports, the left port $^*C$ and the right port, $C^*$. Composition $C \cdot D$ of two components $C$ and $D$ then means to glue the right part $C_r$ of $C$ with the left port $D_l$ of $D$.  

III
Fig. 3 exemplifies this idea. In fact, this kind of composition is associative! Two problems remain: Firstly, the ports to be glued do not always fit, as the ports $D_r' = \{c, d\}$ in Figure 4, and $E_l' = \{d\}$ (Figure 3).

In this case, the remaining element $c$ of $D_r'$ goes to the right port of $D \cdot E$. Likewise, Figure 4(b), the remaining element $e$ of $E_l'$ goes to the left port of $D \cdot E'$. Both effects can mix, as in Figure 4(c). This construct, however, yields another problem, exemplified in Figure 5:

A port may contain equally labeled nodes! Composition is not uniquely defined in this case. We solve this problem by indexing equally labeled nodes. Two nodes are glued only if they coincide in both, their label and their index. More details follow in the next section. In most applications, ports do not contain equally labeled nodes.

Fig. 3. The components of Fig. 1 and Fig. 2 with interfaces partitioned into left and right ports, and composition along ports.

Section 3 will discuss a choice of quite different applications of this formalism. The above examples in Figs. 1 to 5 show components with disjoint ports. Composition of components with disjoint ports yields a component with disjoint ports, again. Nevertheless, disjointness of ports is not essential: The formalism functions perfect also in case the left and the right port of a component intersect. This yields further, most useful applications, as will be discussed in Sect. 4.

2 Interface Graphs and their Composition

The examples in Figs. 1–5 of Sect. 1 suggest that a component can be abstracted as a graph: Just skip the box that surrounds each components’ name. A port of a component is then just a subset of its nodes, together with a specific labelling of the nodes. The identity of a node is irrelevant; just its links to other nodes and its label count. So, we consider graphs $G$ with two distinguished labelled subsets of nodes, called the left and the right port of $G$. To achieve a unique identification of nodes within a port, equally labeled nodes of a port are indexed.
Graphs of this kind will be denoted as *interface graphs*. We define a notion of *composition* of interface graphs, as already sketched in Sect. 1. The central Theorem of this paper states that composition of interface graphs is associative. Proof of this Theorem is lengthy, and postponed to the Appendix. It requires a smart distinction of cases, as well as detailed arguments on carefully chosen notions.

### 2.1 Index labelled sets

As mentioned above, we consider graphs with two distinguished subsets of labelled nodes ("ports"). The distinguished structure of the labelling of ports is denoted as *index labelling*. To cope with ports, this section considers index labelled sets. In the rest of this paper we assume

\[ A \]  

(a finite alphabet).
Each element of an index labelled set will be labelled by a symbol in $A$. Furthermore, equally labelled nodes are distinguished by indices $i \in \mathbb{N}$.

**Definition 1.** Let $P$ be a set, let $\lambda : P \rightarrow A$ and $\vartheta : P \rightarrow \mathbb{N}$ be mappings.

(i) For each $a \in A$ let $\#(a, P) = \{q \in P | \lambda(q) = a\}.$

(ii) For each $a \in A$ and for each $1 \leq i \leq \#(a, P)$ assume there exists exactly one $p \in P$ with $\lambda(p) = a$ and $\vartheta(p) = i$. Then $(\lambda, \vartheta)$ is an index label of $P$.

**Lemma 1.** Let $P$ be a set, let $\lambda : P \rightarrow A$ and $\vartheta : P \rightarrow \mathbb{N}$ be mappings. For each $a \in A$, let $P_a = \{p \in P | \lambda(p) = a\}$ and let $\lambda_a : P_a \rightarrow A$ and $\vartheta_a : P_a \rightarrow \mathbb{N}$ be the restrictions of $\lambda$ and $\vartheta$ to $P_a$. Then $(\lambda_a, \vartheta_a)$ is an index label of $P$ iff for each $a \in A$, $(\lambda_a, \vartheta_a)$ is an index label of $P_a$.

**Proof.** By construction of $\lambda_a$ and $\vartheta_a$ holds for all $p \in P$ and all $a \in A : \lambda(p) = \lambda_a(p)$ iff $\lambda(p) = a$ and $\vartheta(p) = \vartheta_a(p)$ iff $\lambda(p) = a$. Then the Lemma follows from Definition 1(ii). \hfill \Box

**Lemma 2.** Let $P$ be a set, index labelled by $(\lambda, \vartheta)$.

(i) For each $p \in P$, $\#(\lambda(p), P)$ is the number of elements of $P$ with their $\lambda$-label identical to $p$’s $\lambda$-label.

(ii) As a consequence of (i) and Definition 1(ii), $\vartheta(p) \leq \#(\lambda(p), P)$ for each $p \in P$.

(iii) If $\lambda$ is injective, $\vartheta(p) = 1$ for all $p \in P$.

An index labelled set is graphically represented by multiple instances of the $\lambda$-labels of its elements. Equally labelled elements are vertically ordered: Elements with higher $\vartheta$-value are on top of elements with lower $\vartheta$-value. For example, the index labelled set $(CD)'$ in Figure 5 contains three elements. We may call them $\alpha$, $\beta$ and $\gamma$. Arcs link $\alpha$ and $\beta$ to $D'$, and $\gamma$ to $C$. Furthermore, $\lambda(\alpha) = d, \lambda(\beta) = \lambda(\gamma) = c, \vartheta(\alpha) = \vartheta(\beta) = 1$ and $\vartheta(\gamma) = 2$. Fig. 6 shows another example. Again, the $\vartheta$-value of the nodes in this set is implicitly given by the layout of their $\lambda$-labels.

In the sequel, subset, complement, and disjoint union of sets is extended to index labelled sets. We start with the notion of subset. Intuitively formulated, both the $\lambda$-label and the $\vartheta$-label of the elements of an index labelled subset coincide with the labelling of the superset. This way, the notion of subset, extends to index labelled sets (then denoted as initial subset) as follows:

**Definition 2.** For $i = 1, 2$ let $P_i$ be two sets, index labelled by $(\lambda_i, \vartheta_i)$. Let $P_2 \subseteq P_1$. For all $p \in P_2$, assume $\lambda_2(p) = \lambda_1(p)$ and $\vartheta_2(p) = \vartheta_1(p)$. Then $P_2$ is an initial subset of $P_1$, written $P_2 \subseteq P_1$.

Figure 7 shows an example and a counterexample for initial subsets.

**Lemma 3.** For $i = 1, 2$ let $P_i$ be two sets, index labelled by $(\lambda_i, \vartheta_i)$. Let $P_2 \subseteq P_1$. Then for each $p \in P_1 \setminus P_2$ holds: $\vartheta_1(p) > \#(\lambda_2(p), P_2)$. 

VI
Let $P = \{p_1, \ldots, p_7\}$, index labelled by $(\lambda, \vartheta)$, with

\[
\begin{align*}
\lambda(p_i) &= a & \vartheta(p_i) &= i & (i = 1, \ldots, 4), \\
\lambda(p_5) &= \lambda(p_6) &= b & \vartheta(p_5) &= \vartheta(p_6) = 1, \\
\lambda(p_7) &= c & \vartheta(p_7) &= 2
\end{align*}
\]

Graphical representation of $P$:

```
\begin{array}{c}
a \\
a \\
a \ b \\
a \ b \ c
\end{array}
```

**Fig. 6.** Representation of a set $P$ with index label $(\lambda, \vartheta)$.

Let $Q = \{p_1, p_2, p_3, p_7\}$ and $R = \{p_4, p_5, p_6\}$. Let $\lambda$ as in Figure 6 and

\[
\begin{align*}
\vartheta'(p_i) &= 1 & \text{for } i = 1, 4, 5, 7, \\
\vartheta'(p_2) &= \vartheta'(p_6) &= 2, \\
\vartheta'(p_3) &= 3
\end{align*}
\]

Then $Q$ and $R$ are index labelled sets. With $P$ be as in Figure 6 holds: $Q \subseteq P$, whereas $R$ is no initial subset of $P$, because e.g. $\vartheta(p_4)$.

**Fig. 7.** Index labelled set $Q$ and $R$.

**Proof.** Let $p \in P_1 \setminus P_2$. For each $n \leq \#(\lambda(p), P_2)$ there exists some $q \in P_2$ with $\vartheta_2(q) = n$, because $P_2$ is index labelled. Hence for each $n \leq \#(\lambda(p), P_2)$ there exists some $q \in P_2$ with $\vartheta_1(q) = n$ by Definition 1. Hence for each $q \in P_1 \setminus P_2$, $\vartheta_1(q) \leq \#(\lambda(p), P_2)$, by Lemma 2 (ii). Hence the proposition follows by simple arithmetic. \hfill \square

Notice that not each subset of an index labelled set is itself an index labelled set, and that not each index labelled subset is initial.

The complement for initial subsets is defined as follows:

**Definition 3.** For $i = 1, 2$ let $P_i$ be two sets, index labelled by $(\lambda_i, \vartheta_i)$, and assume $P_2 \subseteq P_1$. Let $P_1 \setminus P_2$ be index labelled by $(\lambda, \vartheta)$, where for each $p \in P_1 \setminus P_2$:

\[
\begin{align*}
\lambda(p) &= \mathrel{\overset{\text{def}}{=}} \lambda_1(p), & \lambda(p) &= \mathrel{\overset{\text{def}}{=}} \lambda_2(p), \\
\vartheta(p) &= \mathrel{\overset{\text{def}}{=}} \vartheta_1(p) - \#(\lambda_2(p), P_2), & \vartheta(p) &= \mathrel{\overset{\text{def}}{=}} \vartheta_1(p) - \#(\lambda(p), P_2).
\end{align*}
\]
With the index label \((\lambda, \vartheta)\), the set \(P_1 \setminus P_2\) is the index labelled complement of \(P_2\) w.r.t. \(P_1\), written \(P_1 \downarrow P_2\).

Notice that \(\vartheta(p)\) is well-defined only if \(#(\lambda(p), P_2) < \vartheta(p)\) for each \(p \in P\). This, however, has been proven in Lemma 1. Fig. 8 shows an example.

Finally we define index labelled union. Intuitively formulated, the union \(P_1 \uparrow P_2\) of two disjoint labelled sets \(P_1\) and \(P_2\) puts \(P_2\) "on-top" of \(P_1\):

**Definition 4.** For \(i = 1, 2\) let \(P_i\) be disjoint sets, index labelled by \((\lambda_i, \vartheta_i)\). Let \(P_1 \cup P_2\) be index labelled by \((\lambda, \vartheta)\), with

\[
\lambda(p) = \lambda_i(p) \text{ for } p \in P_i \quad \text{ and } \quad \vartheta(p) = \begin{cases} \vartheta_i(p) & \text{if } p \in P_1 \\ 
\#(\lambda(p), P_1) + \vartheta_2(p) & \text{if } p \in P_2 
\end{cases}
\]

With the index label \((\lambda, \vartheta)\), the set \(P_1 \cup P_2\) is the index labelled union of \(P_1\) with \(P_2\), written \(P_1 \uparrow P_2\).

Elements of two index labelled sets accord iff both, their \(\lambda\)-labelling and their \(\vartheta\)-numbering coincides. Any two index labelled sets yield a maximal set of according pairs of elements. This set is itself index labelled. Furthermore, projection of this set yields initial subsets of the respective sets.

**Definition 5.** For \(i = 1, 2\) let \(P_i\) be disjoint sets, index labelled by \((\lambda_i, \vartheta_i)\), and let \(p_i \in P_i\).

(i) The two elements \(p_1\) and \(p_2\) accord \(((p_1, p_2)\) is an according pair of \(P_1\) and \(P_2\)) iff \(\lambda_1(p_1) = \lambda_2(p_2)\) and \(\vartheta_1(p_1) = \vartheta_2(p_2)\).

(ii) The set of all according pairs of \(P_1\) and \(P_2\) is the accord of \(P_1\) and \(P_2\)

For example, the accord of \(C_i\) and \(D_i\) as in Figure 3 contains two pairs \((p_1, p_2)\) and \((q_1, q_2)\) with \(\lambda(p_i) = c\) and \(\lambda(q_i) = d\), for \(i = 1, 2\). As a further example, \(Q\) and \(R\) have one according pair, \((p_1, p_4)\).

**Lemma 4.** Let \(P_i\) as in Definition 5. Let \(P\) be the accord of \(P_1\) and \(P_2\).

(i) \(P\) itself is an index ordered set.

(ii) \(pr_i(P) \subseteq P_i\).

**Proof.** Let \(a \in A_i\), let \(i = 1, 2\).

(i) For \(i = 1, 2\), let \(n\) be the maximal index with \(\lambda_i(p_i) = a\) and \(\vartheta_i(p_i) = n\), for \((p_1, p_2) \in P\). For all \(k = 1, \ldots, n\), \(P_i\) contains elements \(q_i^k\) with \(\lambda_i(q_i^k) = a\) and \(\vartheta_i(q_i^k) = k\), because \(P_i\) is index ordered. Then for all \(k = 1, \ldots, n\), \((q_i^k, q_i^k)\) is an according pair of \(P_i \cup P_2\) by Definition 5 (i).

Then \(Q_a = \{(q_1^1, q_2^1), \ldots, (q_i^n, q_i^n)\}\) is index ordered.

As this goes for all \(a \in A, P = \bigcup_{a \in A} Q_a\) is index ordered according to Lemma 1, because each \(Q_a\) is index ordered.
(ii) With $Q_a$ as in (i), $pr_i(Q_a) = \{a_1^i, \ldots, a_k^i\}$ is index ordered, because $Q_a$ is. Then $pr_i(P) = \bigcup_{a \in A} pr_i(Q_a)$ is index ordered by Lemma 1, because each $Q_a$ is index ordered.

The decisive notion of this Section is the composition of interface graphs. Fig. 8 sketches the central idea:

First the according pairs of $G_r$ and $H_l$ are identified and corresponding nodes are "glued". The remaining nodes (upper section) of $G_r$ moves on top of $H_r$. Both together yield $(G \cdot H)_l$. The remaining nodes of $H_l$ likewise contribute to $(G \cdot H)_r$.

**Fig. 8. Outline of composition.**

- The composition of two interface graphs $G$ and $H$ is an interface graph again, written $G \cdot H$.
- The left port $G_l$ of $G$, and the right port $H_r$ of $H$, are contained in the left and the right port $(G \cdot H)_l$ and $(G \cdot H)_r$ of $G \cdot H$, respectively.
- The according pairs of nodes of $G_r$ and $H_l$ merge into inner elements of $G \cdot H$.
- The remaining nodes of $H_l$ and $G_r$ go to $(G \cdot H)_l$ and $(G \cdot H)_r$, respectively.

This idea yields the following Definition:

**Definition 6.**
Let $P$ be a finite set,
let $E \subseteq P \times P$,
let $G = (P, E)$
let $G_l, G_r \subseteq P$ be index labelled.

Then $G$ together with $G_l$ and $G_r$ is an interface graph. $P$ and $E$ are its sets of nodes and edges. $G_l$ and $G_r$ are its left and right ports.

inner($G$) =_{def} P \setminus (G_l \cup G_r)$ is the set of inner nodes of $G$. 

IX
Definition 7. Let \( G = (P_G, E_G) \) and \( H = (P_H, E_H) \) be two disjoint interface graphs. Let \( R \) be the accord of the ports \( G_r \) and \( H_l \). The composition of \( G \) and \( H \) is the interface graph \( G \cdot H = \text{def} (P, E) \), with

\[
P = ((P_G \cup P_H) \setminus (pr_1(R) \cup pr_2(R))) \cup R;
\]

\( E \) results from \( E_G \cup E_H \), where for each \( (x, y) \in R \)

- each arc \((z, x) \in E_G \) or \((z, y) \in E_H \) is replaced by \((z, (x, y))\);
- each arc \((x, z) \in E_G \) or \((y, z) \in E_H \) is replaced by \(((x, y), z)\).

\( (G \cdot H)_l = \text{def} G_l \uparrow (H_l \downarrow pr_2(R)) \);
\( (G \cdot H)_r = \text{def} H_r \uparrow (G_r \downarrow pr_1(R)) \).

Figure 3 and Figure 4 depict examples of composed interface graphs. Particularly interesting is the composition of \((C \cdot D')\) (Figure 5) with \( F \) (Figure 9).

![Diagram](image-url)

**Fig. 9.** Interface graphs \( F \) and \((C \cdot D') \cdot F\), with \( C \) as in Figure 3, \( D' \) as in Figure 4. (C. f. also \( C \cdot D' \) in Figure 5).

In graphical representations of composed interface graphs \( G \cdot H \), a pair \((x, y)\) of nodes in the accord of the ports \( G_r \) and \( H_l \) is represented by their common \( \lambda \)-label. Figure 3 and Figure 4 depict examples of composed interface graphs. Particularly interesting is the composition of \( C \cdot D' \) in Figure 5 with \( F \) in Figure 9, yielding two equally labelled nodes in \((C \cdot D') \cdot F)_r\).

Lemma 5. Let \( G, H \) and \( R \) be as in Definition 7.

\[
\text{inner}(G \cdot H) = \text{inner}(G) \cup \text{inner}(H) \cup R.
\]

We conclude this section with the fundamental property of associativity of composition of interface graphs.

Theorem 1. Let \( G, H, K \) be three disjoint interface graphs. Then

\[
(G \cdot H) \cdot K = G \cdot (H \cdot K)
\]

Proof of this Theorem is given in the Appendix.
2.2 Interface graphs and their composition

As outlined above, a component can be abstracted as an interface graph $G$, viz. a graph with two distinguished, index labelled subsets of nodes, the right port $G_r$ and the left port $G_l$ of $G$, respectively.

Figures 3, 4 and 5 showed already graphical representations of interface graphs: The inner nodes remain in a dotted box. The elements of its left and right port are placed left and right of this box. The ports are depicted as described after Definition 1.

3 Appendix

Proof of the Theorem 1.

In the sequel we apply the following notions and shorthands:

$$GH \text{ for } G \cdot H$$

(5)

![Fig. 10. Structure of the proof of Theorem 1.](image-url)
To prove the Theorem, with Definition 6 we have to show for nodes \( e, f \) of \( G, H \) and \( K \):

The nodes of \((GH)K\) and of \(G(HK)\) coincide. \(\text{(6)}\)

The arcs of \((GH)K\) and of \(G(HK)\) coincide, \(\text{(7)}\)

\[ ((GH)K)_l = (G(HK))_l \] \(\text{(8)}\)

\[ ((GH)K)_r = (G(HK))_r \] \(\text{(9)}\)

\[ \lambda_{((GH)K)_l} = \lambda_{(G(HK))_l} \] \(\text{(10)}\)

\[ \lambda_{((GH)K)_r} = \lambda_{(G(HK))_r} \] \(\text{(11)}\)

\[ \vartheta_{((GH)K)_l} = \vartheta_{(G(HK))_l} \] \(\text{(12)}\)

\[ \vartheta_{((GH)K)_r} = \vartheta_{(G(HK))_r} \] \(\text{(13)}\)

Proof of \(\text{(6)}\).

Definition 6 and Definition 7 immediately imply for interface graphs \(X, Y\) and \(f \in \text{inner}(X)\):

\[ f \in \text{inner}(XY) \text{ and } f \in \text{inner}(YX) \]

This in turn implies f.a. \( f \in \text{inner}(G) \cup \text{inner}(H) \cup \text{inner}(K) \) and \( f \in \text{inner}(G(HK)) \). The proposition then follows with forthcoming properties \((8)\) and \((9)\), and Definition 6.

Proof of \(\text{(7)}\).

Property \(\text{(6)}\) implies that the according pairs of \((GH)K\) and of \(G(HK)\) coincide. Hence, the replacements of arcs in the arc sets of \((GH)K\) and of \(G(HK)\), as in Definition 7, coincide.

Proof of \(\text{(10)}\). The \(\lambda\)-labels of elements of ports of interface graphs \(X, Y\) remain when \(x\) is moved to \((XY)_l\) and \((XY)_r\) in Definition 7.

Proof of \(\text{(11)}\). Analog to Proof of \(\text{(10)}\).

\(\text{(8)}\) and \(\text{(12)}\) are proven in the sequel. \(\text{(9)}\) and \(\text{(13)}\) follow analogous to \(\text{(8)}\) and \(\text{(12)}\) and are left to the reader.

Proof of \(\text{(8)}\).

Throughout the proof we assume a node

\[ f \in G \cup H \cup K \] \(\text{(14)}\)

With respect to a port \(P\) and \(f \in P\) we write

\[ \max(P) \text{ for } \#(\lambda_P(f), P), \text{ and} \]

\[ P(f) \text{ for } \vartheta_P(f) \] \(\text{(15)}\)

\(\text{(16)}\)

Intuitively formulated, \(\max(P)\) is the number of nodes in \(P\) with labels coinciding with the label of \(f\). These nodes are numbered, and \(P(f)\) is the corresponding number of \(f\). Obviously holds
Lemma 6. Let P be a port, let p ∈ P. Then P(p) ≤ max(P).

By Definition 7, ((GH)K)_l and (G(HK))_l contain only nodes of G_l, H_l and K_l. Therefore it suffices to show either of the two properties

\[ f \in ((GH)K)_l \text{ and } f \in (G(HK))_l, \text{ with } ((GH)K)_l(f) = (G(HK))_l(f) \]  \hspace{1cm} (17)

or for some node g,

\[ (q, f) \in \text{inner}((GH)K) \text{ and } (g, f) \in \text{inner}(G(HK)). \]  \hspace{1cm} (18)

For the proof we distinguish seven cases, structured as shown in Figure 10. Property (18) applies to cases 2.1, 3.1 and 3.2.2.1 for all other cases holds property (17).

\[ \square \]

Lemma 7. Let X, Y be interface graphs.

(i) Let x ∈ X_l. Then x ∈ (XY)_l and (XY)_l(x) = X_l(x).

(ii) Let y ∈ Y_l. Then y ∈ (XY)_r and (XY)_r(y) = Y_l(y).

(iii) Let y ∈ Y_l and max(X_r) ≥ Y_l(y). Then there exists a unique x ∈ X_r with X_r(x) = Y_l(y) and (x, y) \in \text{inner}(XY).

(iv) Let y ∈ Y_l and let max(X_r) < Y_l(y). Then y \in (XY)_l and (XY)_l(y) = max(X_l) + Y_l(f) - max(X_r).

(v) Let x ∈ X_r and X_r(f) > max(Y_r). Then x \in (XY)_r and (XY)_r(x) = X_r(x) + max(Y_r) - max(Y_l).

(vi) Let max(X_r) > max(Y_l). Then max(XY)_l = max(X_l).

(vii) Let max(X_r) > max(Y_l). Then max(XY)_r = max(Y_r) + max(X_r) - max(X_l).

(viii) Let max(X_r) < max(Y_l). Then max(XY)_l = max(X_l) + max(Y_l) - max(X_r).

(ix) Let max(X_r) < max(Y_l). Then max(XY)_r = max(Y_r)

Case 1 : f ∈ G_l.

a) f ∈ (G(HK))_l and (G(HK))_l(f) = G_l(f), by Case 1 and Lemma 7(ii).

b) f ∈ (GH)_l and (GH)_l(f) = G_l(f), by Case 1 and Lemma 7(ii).

Then f ∈ ((GH)K)_l and ((GH)K)_l(f) = G_l(f), by (*)

\[
\begin{array}{cccccc}
G_l & G_l & H_l & H_l & K_l & K_l \\
\end{array}
\]

Fig. 11. case 1

Case 2 : f ∈ H_l.
\textbf{Case 2.1} : \( \max(G_r) \geq H_l(f) \).

There exists a unique \( g \in G_r \) with \( G_r(g) = H_l(f) \), by Case 2.1 and Lemma 7(iv).

a) \( f \in (HK)_l \) and \( (HK)_l(f) = H_l(f) \), by Case 2 and Lemma 7(ii).

Then \( \max(G_r) \geq (HK)_l(f) \), by Case 2.1 Then \((g, f) \in \text{inner}(G(HK))\), by Case 2 and Lemma 7(iv).

b) \((g, f) \in \text{inner}(GH)\), by Case 2, Case 2.1 and Lemma 7(iv).

Then \((g, f) \in \text{inner}((GH)K)\), by Lemma 5.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig12.png}
\caption{case 2.1}
\end{figure}

\textbf{Case 2.2} : \( \max(G_r) < H_l(f) \).

As a shorthand, let

\[ n = \text{def} \ \max(G_l) + H_l(f) - \max(G_r) \quad (19) \]

a) \( f \in (HK)_l \) and \( (HK)_l(f) = H_l(f) \) \( (20) \)

by Case 2 and Lemma 7(ii).

Then \( \max(G_r) < (HK)_l(f) \), by Case 2.2. Then \( f \in (G(HK))_l \) and

\[
(G(HK))_l(f) \\
= \max(G_l) + (HK)_l(f) - \max(G_r) \quad \text{, by (20) and Lemma 7(v)} \\
= \max(G_l) + H_l(f) - \max(G_r) \quad \text{, by (20)} \\
= n \quad \text{, by (19)}
\]

b) \( f \in (GH)_l \) and

\[
(GH)_l(f) = \max(G_l) + H_l(f) - \max(G_r) \quad (21)
\]

by Case 2 and Lemma 7(v).

Then \( f \in ((GH)K)_l \) and

\[
((GH)K)_l(f) \\
= (GH)_l(f) \quad \text{, by Lemma 7(ii)} \\
= \max(G_l) + H_l(f) - \max(G_r) \quad \text{, by (21)} \\
= n \quad \text{, by (19)}.
\]
Case 3 : \( f \in K_i \)

Case 3.1 : \( \max(H_r) \geq K_i(f) \) There exists a unique \( h \in H_r \) with

\[
H_r(h) = K_i(f) \tag{22}
\]

by case 3, case 3.1 and Lemma 7(iv).

a) \((h, f) \in \text{inner}(HK)\) by case 3, case 3.1 and Lemma 7(iv). Then \((h, f) \in \text{inner}(G(HK))\) by Lemma 5.

b) \( h \in (GH)_r \) and \((GH)_r(h) = H_r(h)\), by (22) and Lemma 7(iii). Then \( h \in (GH)_r \) and \((GH)_r(h) = K_i(f)\), by (22). Then \( \max(GH)_r \geq K_i(f)\) and \((GH)_r(h) = K_i(f)\). Then there exists a unique \( g \in (GH)_r\) with \((GH)_r(g) = K_i(f)\) and \( (g, f) \in \text{inner}(GH)K\), by case 3 and Lemma 7(iv), and \((GH)_r(g) = K_i(f)\).

Furthermore, \((GH)_r(g) = (GH)_r(h)\). Then \( (g, f) \in \text{inner}(GH)K\) and \( g = h\), because \( v_p \) is injective. Then \((b, f) \in \text{inner}(GH)K\).

Case 3.2 : \( \max(H_r) < K_i(f) \). In the rest of this proof, let

\[
n = \text{def} \max(G_l) + \max(H_l) + K_i(f) - \max(G_r) - \max(H_r) \tag{23}
\]

Furthermore it holds:

\[
f \in (HK) \text{ and } (HK)_l(f) = \max(H_l) + K_i(f) - \max(H_r) \tag{24}
\]

by 3.2 and Lemma 7(v).
Case 3.2.1 : \( \max(G_r) < \max(H_l) \)

a) \( \max(G_r) < \max(H_l) + K_l(f) - \max(H_r) \), by 3.2, 3.2.1 and elementary arithmetic.

Then \( \max(G_r) < \max(HK_l) \), by case 3, 3.2.1 and Lemma 7(v).

Then \( f \in (G(HK)_l) \) and \( (G(HK)_l)(f) \)

\[
= \max(G_l) + (HK)_l(f) - \max(G_r) \), by (24) and Lemma 7(v)
\]

\[
= \max(G_l) + \max(H_l) + K_l(f) - \max(H_r) - \max(G_r) \), by (24)
\]

\[
= n \), by (23)
\]

\[
\text{Fig. 15. case 3.2.1 (a)}
\]

b) \( \max((GH)_r) = \max(H_r) \), by 3.2.1 and Lemma 7(xiii). Then \( \max((GH)_r) < K_l(f) \) by 3.2. Then \( f \in ((GH)K)_l \) and

\[
((GH)K)_l(f) = \max((GH)_l) + K_l(f) - \max((GH)_r) \) by case 3 and Lemma 7 (v).

Then \( ((GH)K)_l(f) \)

\[
= \max((GH)_l) + K_l(f) - \max(H_r) \), by (24)
\]

\[
= \max(G_l) + \max(H_l) - \max(G_r) + K_l(f) - \max(H_r) \), by Lem. 7 (xii)
\]

\[
= n \), by (23) and elementary arithmetic.
\]

XVI
Case 3.2.2.1: \( \max(G_r) > \max(H_l) + K_I(f) - \max(H_r) \)
There exists a unique \( g \in G_r \) with
\[
G_r(g) = \max(H_l) + K_I(f) - \max(H_r), \quad \text{by 3.2.2 and Lemma 7(v)} \tag{25}
\]
a) \( f \in (HK)_l \) and \( (HK)_l(f) = \max(H_l) + K_I(f) - \max(H_r) \)
by case 3.2 and Lemma 7(v). Then \( \max(G_r) \geq (HK)_l(f) \), by 3.2.2.1 Then \( (g, f) \in \text{inner}((GHK)_l) \), by (25) and Lemma 7(iv).

Case 3.2.2.2: \( \max(G_r) < \max(H_l) + K_I(f) - \max(H_r) \)
a) \( f \in (HK)_l \) and \( (HK)_l(f) = \max(H_l) + K_I(f) - \max(H_r) \)
by case 3.2 and Lemma 7(v). Then \( f \in (HK)_l \) and \( \max(G_r) < (HK)_l(f) \), by 3.2.2.2. Then \( f \in (G(HK))_l \) and \( \text{inner}((GHK)_l) \)
\[
= \max(G_l) + (HK)_l(f) - \max(G_r), \quad \text{by Lemma 7(v)} \tag{25}
\]
\[
= \max(G_l) + \max(H_l) + K_I(f) - \max(H_r) - \max(G_r), \quad \text{by (23)}
\]
b) \( \max(G_r) + \max(H_r) - \max(H_l) < K_i(f) \), by 3.2.2 and elementary arithmetic.

Then \( \max((GH)_r) < K_i(f) \) by 3.2.2 and Lemma 7(a). Then

\[
f \in ((GH)K)_i \text{ and } ((GH)K)_i(f) = \max((GH)l) + K_i(f) - \max((GH)r), \text{ by 3.2.2 and Lemma 7(x)}
\]

\[
= \max((GH)_l) + K_i(f) - \max((GH)_r), \text{ by 3. and Lemma 7(v)}
\]

\[
= \max(G_l) + K_i(f) - (\max(G_r) + \max(H_r) - \max(H_l)), \text{ by 3.2.2 and Lemma 7(x)}
\]

\[
= n, \text{ by elementary arithmetic.}
\]