A formal algebraic approach for the quantitative modeling of connectors in architectures

Christina Chrysovalanti Fountoukidou\textsuperscript{a} and Maria Pittou\textsuperscript{b}

Department of Mathematics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
\textsuperscript{a}christinafountoukidou1@gmail.com, \textsuperscript{b}mpittou@math.auth.gr

Abstract

In this paper we propose an algebraic formalization of connectors in the quantitative setting, in order to address their non-functional features in architectures of component-based systems. We firstly present a weighted Algebra of Interactions over a set of ports and a commutative and idempotent semiring, which is proved sufficient for modeling well-known coordination schemes in the weighted setup. In turn, we study a weighted Algebra of Connectors over a set of ports and a commutative and idempotent semiring, which extends the weighted Algebra of Interactions with types that encode Rendezvous and Broadcast synchronization. We show the expressiveness of the algebra by modeling the weighted connectors of several coordination schemes. Moreover, we derive two subalgebras, namely the weighted Algebra of Synchrons and the weighted Algebra of Triggers, and study their properties. Finally, we introduce a concept of congruence relation for connectors in the weighted setup and we provide conditions for proving such a congruence.

Keywords: Weighted Algebra of Interactions. Weighted Algebra of Connectors. Coordination schemes. Architectures. Component-based systems.

1 Introduction

Software proliferation has induced the development of massively communicating systems with increasing growth in size and complexity. Well-founded design of such systems can be achieved by component-based techniques which allow the separation of concerns between computation and coordination [BLM06, BS08a, Sif13]. Component-based systems are constructed by multiple individual components which interact according to a software architecture. Architectures have been proved fine-grained models for defining the communication of components, that is implemented by the concept of the so-called connectors [AG94, BCD00, MBBS17, Sif13]. Connectors are architectural entities that regulate the synchronization mode among the permissible interactions of components, where the interactions are specified by the imposed coordination scheme [BMM11, RHJ18]. For instance, in an architecture with a sender and two receiver components, a coordination scheme may forbid any interaction between the receivers. In turn, a connector may impose Rendezvous synchronization mode requiring that all the components should interact simultaneously or Broadcast mode.
where a component, namely the sender, should initiate the interactions with some of the receivers.

Connectors are distinguished in two main categories, namely the stateless and stateful connectors [BMM11]. A connector is called stateless when the interaction constraints that it imposes on the ports stay the same at each round, and it is called stateful otherwise, i.e., when it supports dynamic interactions. Applications of connectors occur in cloud and grid computing technologies with several functionalities including shared variable accesses, buffers, networking protocols, pipes etc. Rigorous formalization of connectors has been proved crucial for the efficient modeling and analysis of coordinated software systems [BMM11, SZ18]. Connectors have been studied mainly in the qualitative setting with alternative frameworks and expressiveness, including process algebras [BCD00, BMM11] and category theory [BLM06, BMM11], while they have been also supported by several architectural description languages in order to facilitate the specification of coordination among components [NS19, Ozk18, SZ18].

However, well-founded design of systems communication should incorporate not only the required qualitative properties but also the related non-functional aspects (cf. [NS19, Sif13, SZ18]). Such features include available resources, energy consumption, probabilities, etc., for implementing the interactions among the components. In this paper, we propose an algebraic formalization of the quantitative aspects of connectors in architectures. The quantitative modeling of connectors has only been addressed in the setting of architectural description languages and mainly deals with their probabilistic behavior (cf. [NS19, SZ18]). On the other hand, there is lack of a formal algebraic framework for connectors in the weighted setup, which is the main contribution of this work.

In particular, we extend the results of [BS08a] for stateless connectors in the weighted setting. In [BS08a], the authors introduced two algebras for modeling the interactions and connectors in component based-systems, and studied their properties. In this paper, we extend these algebras in the weighted framework, and we show that the key results from [BS08a] still hold. Our weighted algebras do not require knowledge on the behavior of components, and the only necessary information lies on the ports that serve for performing the communication. For the quantitative setup, we associate each port with a weight from a commutative and idempotent semiring $K$, expressing the “cost” of its participation in the interactions, which are described as sets of ports. In turn, we study two algebras that encode the weight of the interactions and of connectors, respectively. Also, we define an equivalence relation for the elements of the algebras and by their equivalence classes we derive several properties. In turn, we introduce a concept of congruence relation for weighted connectors and we extend the respective results from [BS08a] in the weighted setup. Congruences are important for connectors since they allow to use them interchangeably without affecting the architecture [BCD00, BMM11, BS08a]. Specifically, the contributions of the current paper are the following:

(i) We introduce the \textit{weighted Algebra of Interactions} over a set of ports $P$ and a commutative and idempotent semiring $K$ ($wAI(P)$ for short). The syntax of the algebra is built over $P$, contains the symbols “0” and “1” such that $0, 1 \notin P$, and allows two operators, namely the weighted union operator “$\oplus$” and the weighted synchronization operator “$\otimes$”, that encode the weight of independent and simultaneous interactions, respectively. We refer to the elements of the algebra simply as $wAI(P)$ elements, and we denote them by $z$. Given a set of interactions, we interpret the semantics of $z \in wAI(P)$ as polynomials over $P$ and $K$. Moreover, we denote by “$\equiv$” the equivalence relation of two $wAI(P)$ elements, i.e., elements that return the same weight on the same set of interactions over $P$. We define the quotient set $wAI(P)/\equiv$ of “$\equiv$” on $wAI(P)$ and for every $z \in wAI(P)$ we denote its equivalence class by $\overline{z}$. Then we show that the structure $(wAI(P)/\equiv, \oplus, \otimes, 0, 1)$ is a commutative and idempotent semiring with the binary operations $\oplus$.
and $\otimes$ and constant elements 0 and 1, i.e., the equivalence class of 0 and 1, respectively. In turn, we apply this result for the computation of the semantics of our second algebra that encodes weighted connectors. Furthermore, we apply $wAI(P)$ in order to model several coordination schemes in the weighted setup, and specifically for Rendezvous, Broadcast, Atomic Broadcast and Causality Chain [BS08a].

(ii) We introduce the weighted Algebra of Connectors over a set of ports $P$ and a commutative and idempotent semiring $K$ ($wAC(P)$ for short) which extends $wAI(P)$ with two unary typing operators, that characterize the type of synchronization applied to ports, namely triggers that can initiate an interaction and synchrons which need synchronization with other ports in order to interact. Hence, the syntax of $wAC(P)$ allows two unary typing operators, trigger “$[\cdot]$” and synchron “$[\cdot]$”, and two binary operators “$\oplus$” and “$\otimes$”, called weighted union and weighted fusion operator, respectively. Weighted union has the same meaning as in $wAI(P)$, while weighted fusion is a generalization of the weighted synchronization in $wAI(P)$. We define the semantics of $wAC(P)$ connectors as $wAI(P)$ elements, and then, applying the semantics of $wAI(P)$ we derive the weight of the connectors over a concrete set of interactions in $P$. We write $[\cdot]^{\alpha}$ for $\alpha \in \{0,1\}$ to denote a typed $wAC(P)$ connector. When $\alpha = 0$, the $wAC(P)$ connector is a synchron otherwise for $\alpha = 1$ it is a trigger. Then we call $\alpha$ a fusion-$wAC(P)$ connector when $\alpha = [\cdot]^{\alpha_1} \otimes \ldots \otimes [\cdot]^{\alpha_n}$, where $\alpha_1, \ldots, \alpha_n \in wAC(P)$ and $\alpha_1, \ldots, \alpha_n \in \{0,1\}$. We obtain several nice properties for $wAC(P)$ and we show the expressiveness of $wAC(P)$ by encoding the weight for the connectors of the coordination schemes Rendezvous, Broadcast, Atomic Broadcast and Causality Chain using fusion-$wAC(P)$ connectors. Furthermore, we consider two subalgebras of $wAC(P)$, the weighted Algebra of Synchrons ($wAS(P)$ for short) and of Triggers ($wAT(P)$ for short) over $P$ and $K$, where the former restricts to synchron elements and the latter to trigger elements, and study their properties.

(iii) We show that due to the weighted fusion operator, equivalent $wAC(P)$ connectors, i.e., connectors with the same $wAI(P)$ elements, are not in general interchangeable. For this, we are interested in a congruence relation for $wAC(P)$ connectors. We explain why we cannot derive a congruence relation between $wAC(P)$ connectors in general, and in turn, we introduce a concept of congruence applied to fusion-$wAC(P)$ connectors. Finally, we derive two theorems that provide conditions for proving such a congruence for fusion-$wAC(P)$ connectors, by extending the results of [BS08a] in our weighted framework. The first theorem shows that similarly typed equivalent fusion-$wAC(P)$ connectors are congruent. The second theorem shows that congruence relation between two fusion-$wAC(P)$ connectors is ensured when the following three conditions hold: (i) they are equivalent, (ii) the equivalence is preserved under the weighted fusion with the trigger $[1]^n$, and (iii) both connectors either contain only synchrons or some triggers.

2 Related work

The concept of connectors has been extensively studied in software engineering by versatile formal frameworks and architectural modeling languages. Most of the existing work addresses the qualitative aspects of connectors, while there is a few work investigating the quantitative setting. In the present paper, we propose an algebraic formalization of the interactions and (stateless) connectors over a commutative and idempotent semiring. In the sequel, we discuss some of the modeling theories and languages on connectors that are comparable with our methodology. The work presented below differs with our framework in at least one of the following aspects: (i) they deal with stateful connectors and dynamic interactions, (ii) the modeling process incorporates the behavior of components and connectors, and (iii) they do not address the quantitative features of connectors, which
is the main novelty of the current paper.

Representative work in the algebraic formalization of connectors includes process algebras, contracts, and category theory. Process algebras are algebraic languages which support the compositional description of concurrent and distributed systems, whose basic elements are their actions. In [BCD00], the authors formalized architectural types by a process algebra based on an architectural description language that supported dynamic interactions. Architectural types were described by components types and stateful connectors. Connectors were defined by a function of their behavior, specified as a family of process algebra terms, and their interactions, specified as a set of process algebra actions. The process algebra was build over a set of actions and contained a hiding, a relabeling, a composition and parallel composition operator, while the semantics was given by state transition graphs. In turn, the authors presented a weak bisimulation equivalence based technique for verifying architectural compatibility and conformity.

In [Pah01], contracts were used to model components and connectors in UML language. Specifically, p-calculus was combined with first-order modal reasoning in order to model connectors. The extended calculus was interpreted in state-based algebraic structures (called objects) and captured dynamic interactions. Reasoning and refinement were also studied for component composition.

In [RHJ18], the authors proposed an architectural metamodel for describing stateful connectors focusing on the communication styles of message passing and remote procedure call mechanisms, which are common in distributed systems. The semantics of the styles were expressed by finite state machine models. Then Alloy was chosen for formalizing those communication styles and for verifying conformance of the communication style at the model level.

In contrast to [BCD00, Pah01, RHJ18], our framework studies stateless connectors, and hence does not deal with dynamic interactions. On the other hand, our algebra models the quantitative aspects of connectors that were not addressed in [BCD00, Pah01, RHJ18]. The investigation of dynamic interactions and stateful connectors in the weighted setup is future work.

In the work of [BLM06], the authors developed an algebra for stateless connectors based on category theory. Their algebra supported symmetry, synchronization, mutual exclusion, hiding and inaction connectors. The authors provided the operational, observational, and denotational semantics of connectors, and then, showed that the latter two coincide. Finally, a complete normal-form axiomatization was presented for the algebra and the proposed framework was applied for modeling architectural and coordination connectors in CommUnity and Reo languages, respectively. In contrast to our framework, the work of [BLM06] studied the qualitative modeling of connectors. Also, our algebras encode sufficiently the weight of independent and synchronized actions, though do not involve hiding connectors. On the other hand, in the work of [BLM06], connectors were specified as entities with behavior and interactions, where concurrent actions, i.e., multiple actions that are executed in parallel, were permitted. Studying the behavior of components and their connectors as well as the application of connectors in concurrent systems is future work.

In [SG03], the authors introduced and studied connector wrappers in order to repair or augment communication-related properties of a system. A wrapper was defined as a new code interposed between component interfaces and infrastructure support. The intent of the code was to modify the behavior of the component with respect to the other components in the system, without actually modifying the component or the infrastructure of the system. Having applied a wrapper to a faulty connector, the authors studied whether the result was sound and if the wrapper was transparent to the caller role without changing the interface. The goal of wrappers was to avoid directly modifying the components in a system. In our approach, we introduce a concept of congruence relation for weighted connectors and we provide conditions that allow checking congruence between two fusion-
\(wAC(P)\) connectors. In contrast to [SG03], this allows one to interchange directly fusion-\(wAC(P)\) connectors without affecting the communication pattern of the architecture.

We point the reader to [BMM11] for a nice survey on some well-known theories of connectors. In that paper, the authors studied the formal approaches of Reo, BIP, nets with boundaries, the algebra of stateless connectors, the tile model, and wire calculus, for the modeling, composing and analyzing connectors. The authors presented a comparison framework for those methodologies and discussed possible enhancements. In comparison to our work, the presented theories studied connectors in the qualitative setting.

Apart from the algebraic methods, the modeling of connectors has been supported by several architectural description languages. In the recent work of [Ozk18] can be found an extensive survey on several architectural description languages that support the modeling of several types of connectors. Among them, a well-known example is Reo, a channel-based coordination model, that served as a language for coordination of concurrent processes or for compositional construction of connectors among component-based systems [APR06, Arb04]. In [APR06, Arb04], components performed input/output operations through connectors that do not have knowledge on the behavior of components. Coordination of components was achieved through channels, which were considered as atomic connectors in Reo. Channels were used to transfer data and there were assumed two types of channel ends: sources and sinks. Complex connectors were compositionally built out of simpler ones. Hence, a connector was a set of channel ends and their connecting channels organized in a graph. The topology of connectors in [APR06, Arb04] was inherently dynamic and it allowed mobility in components connections. Some work has also investigated Reo connectors with probabilistic behavior (cf. [NS19, SZ18]). In [SZ18], the authors studied Reo connectors with probabilistic behavior as timed data distribution streams implemented in Coq. Moreover, in [NS19], the authors formalized Reo connectors with random and probabilistic behavior in PVS. In contrast to [NS19, SZ18], we propose a general algebraic modeling framework in the weighted setup. Another difference is that Reo treats input and output ports separately, while in our framework, we use bidirectional ports.

In [BS08a], the authors introduced the Algebra of Interactions over a set of ports \(P\) (\(AI(P)\) for short) for modeling the interactions of components. The syntax of the algebra was build over the set of ports \(P\), two special symbols encoding empty interaction and the set of empty interactions, a union operator and a synchronization operator. In turn, for their second algebra, the Algebra of Connectors (\(AC(P)\) for short), each port was characterized by a type of synchronization, called trigger when initiating an interaction and synchron when synchronizing with other ports in order to interact. The authors considered also two subalgebras, where all of the connectors had the same type (synchron or trigger), and discussed the relation of the presented algebras. In turn, they introduced and studied a congruence relation for connectors, expressed as the biggest equivalence relation that allowed using connectors interchangeably. Finally, the authors presented applications of their algebras for improving the language and the execution engine of BIP framework. We also point the reader to [GS03] and [GS05] for related methodologies on the characterization of connectors.

Our paper is closely related to the work [BS08a], and in particular extends the presented results in the weighted setup. An important difference from [BS08a] is that there was a clear distinction between syntactic equality and semantic equivalence. Moreover, most of the presented results in [BS08a] were proved by syntactic equality and in particular by considering several axioms corresponding to important properties for the algebras. On the contrary, in the weighted framework in general, we cannot state results by syntactic equality. For this, in our framework we explain that
the notion of equivalence for the \(wAI(P)\) elements and \(wAC(P)\) connectors induces an equivalence relation, respectively. In turn, we derive the equivalence classes for \(wAI(P)\) and \(wAC(P)\). Then we formalize the required properties by equalities and we prove them using the respective equivalence classes. Due to this difference with the work of [BS08a], solving the congruence problem in the weighted setup is more difficult. Specifically, the investigation of a congruence relation for any \(wAC(P)\) connector is an open problem (see also Section 7).

3 Preliminaries

3.1 Notations

For every natural number \(n \geq 1\) we denote by \([n]\) the set \(\{1, \ldots, n\}\). Hence, in the sequel, whenever we use the notation \([n]\) we always assume that \(n \geq 1\).

Next, we recall the basic notions for semirings and series [DKV09, Sak].

3.2 Semiring and series

A monoid \((K, +, \hat{0})\) is a non-empty set \(K\) which is equipped with an associative operation \(+\) and a neutral element \(\hat{0}\) such that \(\hat{0} + k = k + \hat{0} = k\) for every \(k \in K\). A monoid is called commutative if \(+\) is commutative.

A semiring \((K, +, \cdot, \hat{0}, \hat{1})\) consists of a set \(K\), two binary operations \(+\) and \(\cdot\), and two constant elements \(\hat{0}\) and \(\hat{1}\) in \(K\), such that:

(i) \((K, +, \hat{0})\) is a commutative monoid,
(ii) \((K, \cdot, \hat{1})\) is a monoid,
(iii) \(\cdot\) distributes over \(+\), i.e., \((k_1 + k_2) \cdot k_3 = (k_1 \cdot k_3) + (k_2 \cdot k_3)\) and \(k_1 \cdot (k_2 + k_3) = (k_1 \cdot k_2) + (k_1 \cdot k_3)\) for every \(k_1, k_2, k_3 \in K\), and
(iv) \(\hat{0} \cdot k = k \cdot \hat{0} = \hat{0}\) for every \(k \in K\).

A semiring \(K\) is called commutative if \((K, \cdot, \hat{1})\) is commutative. The semiring is denoted simply by \(K\) if the operations and the constant elements are understood. Further, \(K\) is called additively idempotent if \((K, +, \hat{0})\) is an idempotent monoid, i.e., \(k + k = k\) for every \(k \in K\). By the distributivity law, this holds iff \(\hat{1} + \hat{1} = \hat{1}\). In the sequel, we call an additively idempotent semiring simply an idempotent semiring. The following algebraic structures are well-known semirings:

- the semiring \((\mathbb{N}, +, \cdot, 0, 1)\) of natural numbers,
- the Boolean semiring \(B = (\{0, 1\}, +, \cdot, 0, 1)\),
- the tropical or min-plus semiring \(\mathbb{R}_{\min} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)\) where \(\mathbb{R}_{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}\),
- the arctical or max-plus semiring \(\mathbb{R}_{\max} = (\mathbb{R}_{\leq 0} \cup \{-\infty\}, \max, +, -\infty, 0)\),
- the semiring \((\mathcal{PA}, \cup, \cap, \emptyset, A)\) for every non-empty set \(A\),
- the Viterbi semiring \(([0, 1], \max, \cdot, 0, 1)\) used in probability theory, and
the Fuzzy semiring \( F = ([0,1], \max, \min, 0, 1) \), and in general every bounded distributive lattice with the operations sup and inf.

All the above semirings are commutative, and all but the first one are idempotent.

Let \( K \) be a semiring and \( P \) be a non-empty set. A formal series (or simply series) over \( P \) and \( K \) is a mapping \( s : P \to K \). The support of \( s \) is the set \( \text{supp}(s) = \{ p \in P \mid s(p) \neq \hat{0} \} \). A series with finite support is called a polynomial. We denote by \( K \langle P \rangle \) the class of all polynomials over \( P \) and \( K \). Let \( s, r \in K \langle P \rangle \) and \( k \in K \). The sum \( s \oplus r \), the product with scalars \( ks \) and \( sk \), and the Hadamard product \( s \otimes r \) are polynomials in \( K \langle P \rangle \), and are defined elementwise, by

\[
\begin{align*}
\text{(sum)} & : s(p) + r(p), \\
\text{(product with scalars)} & : k \cdot s(p), \\
\text{(Hadamard product)} & : s(p) \cdot r(p)
\end{align*}
\]

Throughout the paper, \((K, +, \cdot, \hat{0}, \hat{1})\) denotes a commutative and idempotent semiring.

### 3.3 Interactions

In this work, we are interested in the coordination patterns of the systems, and specifically on the algebraic characterization of the concept of connectors in the quantitative setting. For this, we develop no theory about the semantics of component-based systems. The investigation of the weighted behavior of a system, where communication is expressed with our weighted algebras, is part of future work.

Next, we use the basic notions and definitions of BIP for the communication of components in qualitative setting [BS08a, Sif13]. Though, our results can be applied to every component-based framework where the interface of the system can be described by a set of ports.

In our setting, communication in architectures is performed by a set of ports and is defined by interactions. In turn, the permissible set of interactions is specified by the coordination scheme implemented in the architecture.

**Definition 1.** Let \( P \) be a finite non-empty set of ports. Then an interaction \( a \) is a set of ports over \( P \), i.e., \( a \in 2^P \).

Let \( a = \{p_1, \ldots, p_n\} \subseteq P \). We often simplify the notation of \( a \) by writing \( p_1 \ldots p_n \) instead of the set \( \{p_1, \ldots, p_n\} \). Then an *interactions set* \( \gamma \) is a set of interactions over \( P \), i.e., \( \gamma \in 2^{2^P} \). We let \( I(P) = 2^P \) denote the set of interactions over \( P \), i.e., \( \gamma \in 2^{I(P)} \). We let \( \Gamma(P) = 2^{I(P)} \) denote the set of all subsets of \( I(P) \).

It should be clear that we are not practically interested in the empty interaction \( a = \emptyset \), and in turn the empty interactions set \( \gamma = \{\emptyset\} \) since they correspond to the case that there is no communication among any components of a system. Still, we need to consider \( a = \emptyset \) and \( \gamma = \{\emptyset\} \) in the construction of our weighted algebras in order to be well-defined.

In order to encode the weight of the interactions, we introduce a weighted Algebra of Interactions over a set of ports \( P \) and a commutative and idempotent semiring \( K \), presented in the next section. The proposed algebra is applied for modeling well-known coordination schemes in the weighted setup, and in turn serves as the basis for constructing our second algebra, the weighted Algebra of Connectors, that formalizes of the concept of weighted connectors.
4 The Weighted Algebra of Interactions

Coordination schemes capture the permissible interactions among the components of an architecture. In order to address the quantitative aspects of coordination schemes, we study the weighted Algebra of Interactions over a set of ports $P$ and a commutative and idempotent semiring $K$. The algebra is build over $P$, two further symbols “0” and “1” that do not belong in $P$, and permits two operators, the weighted union “$\oplus$” and weighted synchronization “$\otimes$” operators. Weighted union serves to compute the weight of interactions executed independently by the involved terms, while weighted synchronization captures the weight of interactions applied simultaneously. The symbols 0 and 1 serve as the neutral elements of the operators $\oplus$ and $\otimes$, respectively. We prove several nice properties for the algebra and we provide concrete examples of coordination schemes in the weighted setup, described by our algebra.

Let $P$ be a set of ports. Then we assign to each port $p \in P$ a unique weight from $K$, denoted by $k_p$.

**Definition 2.** Let $P$ be a set of ports such that $0, 1 \notin P$. The syntax of the weighted Algebra of Interactions (wAI($P$) for short) over $P$ and $K$ is given by

$$z ::= 0 \mid 1 \mid p \mid z \oplus z \mid z \otimes z \mid (z)$$

where $p \in P$, “$\oplus$” is the weighted union operator and “$\otimes$” is the weighted synchronization operator that binds stronger than “$\oplus$”.

It should be clear that “0” and “1” occurring in the syntax of the $wAI(P)$ are distinct from the respective elements $\hat{0}$ and $\hat{1}$ in semiring $K$. Indeed, the former serve to encode the weight on a specific $\gamma \in \Gamma(P)$, as presented in Definition 3 below, while the latter correspond to the neutral elements of the operations “$+$” and “$\cdot$” in $K$, respectively.

We call $z$ a $wAI(P)$ element over $P$ and $K$. Whenever the latter are understood we simply refer to $z$ as a $wAI(P)$ element. The semantics of a $wAI(P)$ element $z$ over the set of ports $P$ and the semiring $K$ is defined by the function $\|\cdot\|: \Gamma(P) \to K$. Therefore, we represent the semantics of $z$ as polynomials $\|z\| \in K \langle \Gamma(P) \rangle$. Hence, given an interactions set $\gamma \in \Gamma(P)$, we can derive the weight of implementing $\gamma$ in a given architecture.

Note that for different instantiations of the semiring $K$, the resulting weight is interpreted as a particular quantitative property. For example, if we are interested in ensuring a sufficient level of trustworthiness, then we opt for the interactions with the maximal probability to be executed. Hence, we may choose to work in the Viterbi semiring in order to address this issue. On the contrary, if the weights range over $\mathbb{R}_{\min}$, then we may be interested in the minimal cost, where the cost may refer to the energy consumption of implementing the communications. In such a setting, the derived weight returns the minimal cost of the interactions that could improve the efficiency of the architecture.

**Definition 3.** Let $z$ be a $wAI(P)$ element over $P$ and $K$. The semantics of $z$ is a polynomial $\|z\| \in K \langle \Gamma(P) \rangle$. For every interactions set $\gamma \in \Gamma(P)$, the value $\|z\| (\gamma)$ is defined inductively on $z$ as follows:

- $\|0\| (\gamma) = \hat{0}$,

- $\|1\| (\gamma) = 1$ if $\emptyset \in \gamma$

otherwise, $\hat{0}$.


Remark 1. According to the above semantics, the \( wAI(P) \) element “0” returns zero weight on any \( \gamma \) while the \( wAI(P) \) element “1” returns non-zero value, equal to 1, only if \( \gamma \) contains \( a = \emptyset \), i.e., the empty interaction. Also, the \( wAI(P) \) element \( p \) returns its weight \( k_p \) whenever it occurs in some interaction \( a \in \gamma \), which implies that the port is “activated” with its assigned weight. Moreover, the semantics of weighted union and synchronization are justified by the fact that they encode the weight of independent and synchronous interactions, respectively. Hence, for the former operator we consider the weight on the same set \( \{a\} \) for \( z_1 \) and \( z_2 \) for every \( a \in \gamma \), while for the latter we analyze the interaction \( a \) to the union of \( a_1 \) and \( a_2 \) and we compute the weight of \( z_1 \) and \( z_2 \) on \( \{a_1\} \) and \( \{a_2\} \), respectively, for every such analysis of \( a \) and for every \( a \in \gamma \). Finally, the application of the parenthesis construct on a \( wAI(P) \) element \( z \) does not affect its weight, since parenthesis only serves for imposing the common order restrictions among the occurring operators. For this, \( z \) and \( z \) return the same semantics.

Remark 2. Recall that an element of the Algebra of Interactions from [BS08a] served to encode a specific interactions set \( \gamma \in \Gamma(P) \). On the other hand, a \( wAI(P) \) element can be interpreted for any \( \gamma \in \Gamma(P) \), inducing a weight from \( K \). This difference results from the semantics given to ports \( p \in P \). In [BS08a], a term \( p \) was associated only with \( \gamma = \{\{p\}\} \), while in \( wAI(P) \), a weighted term \( p \) returns its weight \( k_p \) whenever it occurs in some interaction \( a \in \gamma \). This is a natural choice, and as a consequence, the algebra can sufficiently encode the expected overall weight of a coordination scheme, as it is shown by the examples presented later in this section.

Remark 3. Observe that if \( \gamma = \emptyset \), then by the above definition we get that \( \|z\| (\gamma) = 0 \) for every \( z \in wAI(P) \).

Remark 4. In case that \( \gamma \in \Gamma(P) \) is a singleton, i.e., \( \gamma = \{a'\} \), then for the semantics of the weighted union operator we have

\[
\|z_1 \oplus z_2\| (\gamma) = \sum_{a \in \gamma} \left( \|z_1\| (\{a\}) + \|z_2\| (\{a\}) \right)
\]

\[
= \sum_{\{a'\}} \left( \|z_1\| (\{a\}) + \|z_2\| (\{a\}) \right)
\]

\[
= \|z_1\| (\{a'\}) + \|z_2\| (\{a'\}).
\]

Moreover, for the semantics of the weighted synchronization operator we have

\[
\|z_1 \otimes z_2\| (\gamma) = \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} \left( \|z_1\| (\{a_1\}) \cdot \|z_2\| (\{a_2\}) \right) \right)
\]
\[
\sum_{a \in \{a'\}} \left( \sum_{a = a_1 \cup a_2} \left( \|z_1\| (\{a_1\}) \cdot \|z_2\| (\{a_2\}) \right) \right)
= \sum_{a' = a_1 \cup a_2} \left( \|z_1\| (\{a_1\}) \cdot \|z_2\| (\{a_2\}) \right).
\]

Hence, in what follows, when \( \gamma \) is a singleton, i.e., \( \gamma = \{a'\} \in \Gamma(P) \) we often write

- \( \|z_1 \oplus z_2\| (\gamma) = \|z_1\| (\{a'\}) + \|z_2\| (\{a'\}) \), and
- \( \|z_1 \otimes z_2\| (\gamma) = \sum_{a' = a_1 \cup a_2} \left( \|z_1\| (\{a_1\}) \cdot \|z_2\| (\{a_2\}) \right) \).

In [BS08a], the authors derived several properties in their Algebra of Interactions, as axioms obtained directly by syntactic equality. On the contrary, in the weighted setup, we derive an equivalence relation induced by the semantics of \( wAI(P) \). In turn, we use the equivalence classes of \( wAI(P) \) for proving the properties that hold for the weighted algebra. We say that \( z_1, z_2 \in wAI(P) \) are equivalent and we write \( z_1 \equiv z_2 \), when they return the same weight on the same set of interactions, i.e., when \( \|z_1\| (\gamma) = \|z_2\| (\gamma) \) for every \( \gamma \in \Gamma(P) \).

Obviously the relation \( \equiv \) is an equivalence relation. We define the quotient set \( wAI(P)/ \equiv \) of \( \equiv \) on \( wAI(P) \). For every \( z \in wAI(P) \), we simply denote by \( z \) its equivalence class. We define on \( wAI(P)/ \equiv \) two operations as follows: \( \overline{z_1} \oplus \overline{z_2} = \overline{z_1 \oplus z_2} \) and \( \overline{z_1} \otimes \overline{z_2} = \overline{z_1 \otimes z_2} \) for every \( z_1, z_2 \in wAI(P) \). It can be easily proved that these operations depend only on the classes and not on their representatives, i.e., they are well-defined. Thus, if \( \overline{z_1} = \overline{z_3} \) and \( \overline{z_2} = \overline{z_4} \), then \( \overline{z_1 \oplus z_2} = \overline{z_3 \oplus z_4} \) and \( \overline{z_1} \otimes \overline{z_2} = \overline{z_3} \otimes \overline{z_4} \) for every \( z_1, z_2, z_3, z_4 \in wAI(P) \).

In the next proposition, we prove several properties satisfied by the \( wAI(P) \) elements over \( P \) and \( K \). As a consequence we obtain that \( (wAI(P)/ \equiv, \oplus, \otimes, \overline{0}, \overline{1}) \) is a commutative and idempotent semiring with two binary operations \( \oplus \), i.e., weighted union and \( \otimes \), i.e., weighted synchronization, and two constant elements \( \overline{0} \) and \( \overline{1} \), respectively. In turn, we apply this result to our weighted algebra for connectors, presented in the following section. Firstly, we consider the subsequent lemma which is needed for proving the equalities iii), iv) and vii) of Proposition 1. The lemma actually formalizes the intuitive result that the weight of a \( wAI(P) \) element \( z \) on a given set of interactions \( \gamma \in \Gamma(P) \) equals the sum of the weight of \( z \) on \( \{a\} \) for every interaction \( a \in \gamma \).

**Lemma 1.** Let \( z \in wAI(P) \). Then for every interactions set \( \gamma \in \Gamma(P) \) it holds that

\[
\sum_{a \in \gamma} (\|z\| (\{a\})) = \|z\| (\gamma).
\]

**Proof.** We prove the lemma by induction on the structure of \( z \in wAI(P) \). By Remark 3, we assume that \( \gamma \in \Gamma(P) \setminus \{\emptyset\} \).

- If \( z = 0 \), then \( \sum_{a \in \gamma} (\|0\| (\{a\})) = \sum_{a \in \gamma} (\overline{0}) = \overline{0} \) and \( \|0\| (\gamma) = \overline{0} \).

- If \( z = 1 \) and \( \emptyset \in \gamma \), then \( \sum_{a \in \gamma} (\|1\| (\{a\})) = \|1\| (\{\emptyset\}) + \sum_{a \in \gamma, a \notin \emptyset} (\|1\| (\{a\})) = \overline{1} + \sum_{a \in \gamma, a \notin \emptyset} (\overline{0}) = \overline{1} + \overline{0} \) and \( \|1\| (\gamma) = \overline{1} \). On the other hand, if \( z = 1 \) and \( \emptyset \notin \gamma \), then \( \sum_{a \in \gamma} (\|1\| (\{a\})) = \sum_{a \in \gamma} (\overline{0}) = \overline{0} \) and \( \|1\| (\gamma) = \overline{0} \). Hence, in each case we obtain that \( \sum_{a \in \gamma} (\|1\| (\{a\})) = \|1\| (\gamma) \).

10
Therefore, for any $z \in wAI(P)$, we proved that $\sum_{a \in \gamma} (\|z\| (\{a\})) = \|z\| (\{a\})$.

Proposition 1. Let $\overline{z_1}, \overline{z_2}, \overline{z_3} \in wAI(P)/ \equiv$. Then
\textbf{Proof.} Let $\gamma \in \Gamma(P)$. By Remark 3, we prove the above equalities for $\gamma \in \Gamma(P) \setminus \{0\}$.

\begin{itemize}
  \item[i)] $(z_1 \oplus z_2) \oplus z_3 = z_1 \oplus (z_2 \oplus z_3)$
  \item[ii)] $z_1 \oplus z_3 = z_3 \oplus z_1$
  \item[iii)] $z_1 \oplus z_1 = z_1$
  \item[iv)] $z_1 \oplus 0 = z_1$
  \item[v)] $(z_1 \otimes z_2) \otimes z_3 = z_1 \otimes (z_2 \otimes z_3)$
\end{itemize}

\begin{align*}
  \{ z \in wAI(P) \mid z \equiv (z_1 \oplus z_2) \oplus z_3 \} & = \{ z \in wAI(P) \mid \|z\| (\gamma) = \|z_1 \oplus z_2 \oplus z_3\| (\gamma) \text{ for every } \gamma \in \Gamma(P) \} \\
  \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} \left( \sum_{a' \in \{a\}} (\|z_1\| (\{a\}) + \|z_2\| (\{a\})) + \|z_3\| (\{a\}) \right) \text{ for every } \gamma \in \Gamma(P) \} & = \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} \left( \sum_{a' \in \{a\}} (\|z_1\| (\{a\}) + \|z_2\| (\{a\}) + \|z_3\| (\{a\}) \right) \text{ for every } \gamma \in \Gamma(P) \} \\
  \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2\| (\{a\}) + \|z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} & = \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2 \oplus z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} \\
  \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2 \oplus z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} & = \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2 \oplus z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} \\
  \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2 \oplus z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} & = \{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2 \oplus z_3\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \} \\
  \{ z \in wAI(P) \mid z \equiv z_1 \oplus (z_2 \oplus z_3) \} & = \{ z \in wAI(P) \mid z \equiv z_1 \oplus (z_2 \oplus z_3) \}
\end{align*}
where the eighth equality holds since \( \{a\} \) is a singleton and hence \( a' \in \{a\} \) implies that \( a' = a \).

The ninth equality holds by the associativity property of “+” in semiring \( K \).

\[
\begin{align*}
ii) & \quad \overline{z_1} + \overline{z_2} = \overline{z_1 + z_2} = \{z \in wAI(P) \mid z \equiv z_1 \oplus z_2\} \\
& = \{z \in wAI(P) \mid \|z\| (\gamma) = \|z_1 \oplus z_2\| (\gamma) \text{ for every } \gamma \in \Gamma(P)\} \\
& = \left\{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_2\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \right\} \\
& = \{z \in wAI(P) \mid \|z\| (\gamma) = \|z_2 \oplus z_1\| (\gamma) \text{ for every } \gamma \in \Gamma(P)\} \\
& = \{z \in wAI(P) \mid z \equiv z_2 \oplus z_1\} \\
& = \overline{z_2 \oplus z_1} \\
& = \overline{z_2} \oplus \overline{z_1}
\end{align*}
\]

where the fifth equality holds by the commutativity property of “+” in semiring \( K \).

\[
\begin{align*}
iii) & \quad \overline{z_1} \oplus \overline{z_1} = \overline{z_1 + z_1} = \{z \in wAI(P) \mid z \equiv z_1 \oplus z_1\} \\
& = \{z \in wAI(P) \mid \|z\| (\gamma) = \|z_1 \oplus z_1\| (\gamma) \text{ for every } \gamma \in \Gamma(P)\} \\
& = \left\{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|z_1\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \right\} \\
& = \{z \in wAI(P) \mid \|z\| (\gamma) = \|z_1\| (\gamma) \text{ for every } \gamma \in \Gamma(P)\} \\
& = \{z \in wAI(P) \mid z \equiv z_1\} \\
& = \overline{z_1}
\end{align*}
\]

where the fifth and sixth equality hold since the semiring \( K \) is idempotent and by a direct application of Lemma 1, respectively.

\[
\begin{align*}
iv) & \quad \overline{z_1} \oplus 0 = \overline{z_1 \oplus 0} = \{z \in wAI(P) \mid z \equiv z_1 \oplus 0\} \\
& = \{z \in wAI(P) \mid \|z\| (\gamma) = \|z_1 \oplus 0\| (\gamma) \text{ for every } \gamma \in \Gamma(P)\} \\
& = \left\{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} (\|z_1\| (\{a\}) + \|0\| (\{a\})) \text{ for every } \gamma \in \Gamma(P) \right\}
\end{align*}
\]
\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \|z_1\|_\gamma (\{a\}) + \hat{0} \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \|z_1\|_\gamma \{a\} \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid z \equiv z_1 \]

where the sixth and the seventh equality hold since \( \hat{0} \) is the neutral element of \( \{\, +\, \} \) in semiring \( K \) and by applying Lemma 1, respectively.

\[ \{z \in wAI(P) \mid z \equiv (z_1 \circ z_2) \circ z_3 \} \]

\[ \{z \in wAI(P) \mid \|z\|_\gamma \equiv (\|z_1\|_\gamma \circ \|z_2\|_\gamma) \circ \|z_3\|_\gamma \text{ for every } \gamma \in \Gamma(P) \} \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' \in \{a\}} \left( \sum_{a'' = a_1 \cup a_2} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \right) \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' = a_1 \cup a_2} \left( \sum_{a'' = a_3} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \right) \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' = a_1 \cup a_2 \cup a_3} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \cdot \|z_3\|_\gamma (\{a_3\}) \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' = a_1 \cup a_2} \left( \sum_{a'' = a_3} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \right) \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' = a_1 \cup a_2} \left( \sum_{a'' = a_3} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \right) \right) \text{ for every } \gamma \in \Gamma(P) \]

\[ z \in wAI(P) \mid \|z\|_\gamma = \sum_{a \in \gamma} \left( \sum_{a' = a_1 \cup a_2} \left( \sum_{a'' = a_3} \|z_1\|_\gamma (\{a_1\}) \cdot \|z_2\|_\gamma (\{a_2\}) \right) \right) \text{ for every } \gamma \in \Gamma(P) \]
where the fifth equality holds since \( \cdot \) is commutative in semiring \( K \).

\[ \]
\[
\{ z \in wAI(P) \mid \|z\| (\gamma) = \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\|z_1\| (\{a_1\}) \cdot 1 \| (\{a_2\})) \right) \text{ for every } \gamma \in \Gamma(P) \}
\]

where the fifth equality holds since the only interactions set with non-zero weight for “1” is \( \{a_2\} = \{\emptyset\} \). Hence, we obtain that \( a_1 = a \) and \( a_2 = \emptyset \). Also, the seventh equality holds since “1” is the neutral element for “·” in semiring \( K \), while the eighth equality results by an application of Lemma 1.

\[
\begin{align*}
\overline{z_1} \otimes \overline{0} &= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} \|z_1\| (\{a_1\}) \cdot 1 \| (\{a_2\}) \right) \text{ for every } \gamma \in \Gamma(P) \\
\end{align*}
\]

where the sixth equality holds since “\( \overline{0} \)” is the absorbing element for “·” in semiring \( K \).

\[
\overline{z_1} \otimes (\overline{z_2} \oplus \overline{z_3}) = \overline{z_1} \otimes \overline{z_2} \oplus \overline{z_1} \otimes \overline{z_3} = \overline{z_1} \otimes (\overline{z_2} \oplus \overline{z_3}) = \overline{z_1} \otimes (\overline{z_2} \oplus \overline{z_3})
\]

\[16\]
Thus, our proof is completed.

x) The equality is proved as in the previous case.

Thus, our proof is completed.
Corollary 1. The structure \((wAI(P)/\equiv, \oplus, \otimes, 0, 1)\) is a commutative and idempotent semiring.

4.1 Examples of coordination schemes encoded in \(wAI(P)\)

Next we present several coordination schemes in the weighted setup and we encode the cost of their implementation by the weighted Algebra of Interactions.

Let \(z\) be the \(wAI(P)\) element describing a scheme in the weighted setup and consider an interactions set \(\gamma \in \Gamma(P)\). In order to derive the semantics of the \(z\), we use for every \(a \in \gamma\) two tables, namely the primary and the auxiliary tables. In the primary tables, we derive the overall weight of element \(z\) using the corresponding auxiliary tables. Primary and auxiliary tables contain in the first row the \(wAI(P)\) element \(z\) and the \(wAI(P)\) terms of \(z\), respectively, whose weight is calculated on a given interaction \(a \in \gamma\) as indicated by the label of the tables. The primary tables, referred to as tables, have four columns. The first one includes all the possible analyses \(a_1 \cup a_2\) of the given \(a \in \gamma\), while the second and third column contain the semantics of the respective \(wAI(P)\) terms comprising \(z\). In the fourth column we compute the weight of the weighted synchronization applied on the derived \(wAI(P)\) terms. Then in the last row of the tables we obtain the overall weight of \(z\) on the particular \(\{a\}\). The auxiliary tables, are structured as the primary ones with the difference that they return the weight on the \(wAI(P)\) terms of \(z\) appearing on the third column of the primary tables. The auxiliary tables are also referred to as tables and are identified by their labels.

For representation reasons, all the tables used in our examples, are found in the Appendix 9. There we present firstly the primary tables of the \(wAI(P)\) elements for every interaction \(a \in \gamma\), and then, the auxiliary tables which are necessary for our calculations. Also, when a \(wAI(P)\) element contains the “\(\oplus\)” operator and its set of interactions is singleton, we compute directly its corresponding weight within its table, without considering further tables.

It should be clear that the analysis of \(z\) in the respective \(wAI(P)\) elements is not unique, and hence it is chosen arbitrarily. In what follows, we usually split \(z\) in the terms occurring at the left and right of the first weighted synchronization operator. Obviously, for any other analyses of \(z\) we derive the same overall weight on the given interactions set \(\gamma\). Also, we choose to compute the weight of each \(wAI(P)\) element on the interactions set \(\gamma \in \Gamma(P)\) that contains only the interactions of the scheme. Obviously, by Definition 3, for any other \(\gamma \in \Gamma(P)\) our \(wAI(P)\) algebra returns the expected weight.

Example 1. Consider an architecture with a sender and two receivers, each having a single port \(s\) and \(r_1, r_2\), for sending and receiving messages, respectively. Hence, \(P = \{s, r_1, r_2\}\), and let \(k_s, k_{r_1}, k_{r_2}\), denote the weights of the ports \(s, r_1, r_2\), respectively. We consider the coordination schemes of Rendezvous, Broadcast, Atomic Broadcast and Causality Chain for defining the communication among the three components of the architecture. Next, we formalize each of the aforementioned schemes in the weighted setup by a \(wAI(P)\) element \(z\).

**Weighted Rendezvous:** This coordination scheme requires strong synchronization between the sender and each of the two receivers. Hence, it consists of a single interaction involving all ports, namely \(sr_1r_2\). The \(wAI(P)\) element describing the weighted Rendezvous is

\[ z = s \otimes r_1 \otimes r_2. \]

We let \(\gamma = \{\{s, r_1, r_2\}\} \in \Gamma(P)\) and we compute the weight of \(z\) on \(\gamma\) by applying the semantics of \(wAI(P)\). In Table 1, we list in the first column all the possible analyses for \(a = \{s, r_1, r_2\} \in \gamma\)
such that \( a = a_1 \cup a_2 \), in the second and third column we compute the semantics of the \( \text{wAI}(P) \) elements \( s \) and \( r_1 \otimes r_2 \), respectively, and in the last column we derive the cost of the respective weighted synchronization. The semantics for \( r_1 \otimes r_2 \) are obtained by the auxiliary Tables 3-10, for each set \( a_2 \) where \( a_2 = a_{2,1} \cup a_{2,2} \). Then the last row in each table serves for computing the sum of the weights occurring in the last column. Specifically, in Table 1, the resulting value \( k_s \cdot k_{r_1} \cdot k_{r_2} \), which is highlighted, corresponds to the overall weight of Rendezvous scheme on the given \( \gamma \). The weight of \( z \in \text{wAI}(P) \) on \( \gamma \) is computed as follows:

\[
\|s \otimes r_1 \otimes r_2\|(\gamma) = \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\|s\|\{a_1\}) \cdot \|r_1 \otimes r_2\|\{a_2\}) \right)
\]

For instance, let us consider the Fuzzy semiring \( F = ([0,1], \max, \min, 0, 1) \). Then the value

\[
\|s \otimes r_1 \otimes r_2\|(\gamma) = \max_{a \in \gamma} \left( \max_{a = a_1 \cup a_2} \left( \min_{a = a_1 \cup a_2} (\|s\|\{a_1\}) \right) \cdot \left( \max_{a = a_1 \cup a_2} \left( \min_{a = a_1 \cup a_2} (\|r_1 \otimes r_2\|\{a_2\}) \right) \right) \right)
\]

returns for the Rendezvous scheme the maximum of the minimum weights associated with each port.

Let now \( \gamma = \{\{s, r_1, r_2\}, \{s, r_2\}\} \in \Gamma(P) \). We use the primary Tables 1 and 2 as well as the auxiliary Tables 3-10. Then the weight of \( z \in \text{wAI}(P) \) on \( \gamma \) occurs as follows:

\[
\|s \otimes r_1 \otimes r_2\|(\gamma) = \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\|s\|\{a_1\}) \cdot \|r_1 \otimes r_2\|\{a_2\}) \right)
\]
Hence, in the case that \( \gamma \) includes further interactions than the expected ones, then the given \( wAI(P) \) element returns the expected weight for the scheme. Moreover, let us consider the case that \( \gamma \) consists only of interactions that do not result from the given Rendezvous scheme, and let \( \gamma = \{\{s, r_2\}\} \in \Gamma(P) \). Then by Table 2 the weight of \( z \in wAI(P) \) on \( \gamma \) equals to 0. The cases for \( \gamma = \{\{s, r_1, r_2\}, \{s, r_2\}\} \) and \( \gamma = \{\{s, r_2\}\} \) verify the robustness of our \( wAI(P) \) and justify the interpretation of the presented semantics.

**Weighted Broadcast:** This coordination scheme allows executing all interactions involving the sender and any subset of receivers, possibly the empty one. Hence, each permissible interaction should contain the port \( s \). Thus, all the possible interactions are \( s, sr_1, sr_2, sr_1r_2 \). The \( wAI(P) \) element \( z \) describing weighted Broadcast is

\[
z = s \otimes (1 + r_1) \otimes (1 + r_2).
\]

We set \( \gamma = \{\{s\}, \{s, r_1\}, \{s, r_2\}, \{s, r_1, r_2\}\} \in \Gamma(P) \) and we compute the weight of \( z \) on \( \gamma \). For this, we need to obtain the analyses and the respective weight for each of the four interactions occurring in \( \gamma \). We firstly consider the interaction \( a = \{s\} \in \gamma \) for which we obtain Table 11. Then we derive the weight of \( z \) for \( a = \{s, r_1\} \in \gamma \) and \( a = \{s, r_2\} \in \gamma \), presented in Tables 12 and 13, respectively. Finally, we get the weight of \( z \) for \( a = \{s, r_1, r_2\} \in \gamma \), as shown in Table 14. The above weights are derived using the auxiliary Tables 15-22. The overall weight of \( z \) on \( \gamma \) is computed as follows:

\[
\begin{align*}
&= \sum_{a \in \gamma} \left( \sum_{a = 1 \cup a_2} \left( \|s\| \left( \{a_1\}\right) \cdot \left( \sum_{a' = 1 \cup a_2} \left( \|1 + r_1\| \left( \{a_2, 1\}\right) \right) \right) \right) \right)
\end{align*}
\]

\[
\begin{align*}
= \sum_{a \in \gamma} \left( \sum_{a = 1 \cup a_2} \left( \|s\| \left( \{a_1\}\right) \cdot \left( \sum_{a' = 1 \cup a_2} \left( \|1 + r_1\| \left( \{a_2, 1\}\right) \cdot \|1 + r_2\| \left( \{a_2, 2\}\right)\right) \right) \right) \right)
\end{align*}
\]

\[
\begin{align*}
= \sum_{a \in \gamma} \left( \sum_{a = 1 \cup a_2} \left( \|s\| \left( \{a_1\}\right) \cdot \left( \sum_{a' = 1 \cup a_2} \left( \|1 + r_1\| \left( \{a_2, 1\}\right) \cdot \|1 + r_2\| \left( \{a_2, 2\}\right)\right) \right) \right) \right)
\end{align*}
\]

\[
\begin{align*}
= \sum_{a \in \gamma} \left( \sum_{a = 1 \cup a_2} \left( \|s\| \left( \{a_1\}\right) \cdot \left( \sum_{a' = 1 \cup a_2} \left( \|1 + r_1\| \left( \{a_2, 1\}\right) \cdot \|1 + r_2\| \left( \{a_2, 2\}\right)\right) \right) \right) \right)
\end{align*}
\]

\[
\begin{align*}
= k_s + (k_s + (k_s \cdot k_{r_1})) + (k_s + (k_s \cdot k_{r_2})) + (k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_2}) +
\end{align*}
\]
\((k_s \cdot k_{r_1} \cdot k_{r_2})\)
\[= k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_2}) + (k_s \cdot k_{r_1} \cdot k_{r_2}).\]

Consider for instance the \(\mathbb{F}_{\text{max}}\) semiring. Then the resulting value represents for weighted Broadcast scheme the maximum sum of the weights associated with the ports occurring in the interactions of \(\gamma\).

**Weighted Atomic Broadcast:** In the Atomic Broadcast scheme a message is either received by all receivers or by none of them, which implies that we have only two interactions, \(s\) and \(sr_1r_2\). The \(wAI(P)\) element \(z\) describing the weighted Atomic Broadcast is
\[z = s \otimes (1 \oplus r_1 \otimes r_2).\]

We let \(\gamma = \{\{s\}, \{s, r_1, r_2\}\} \in \Gamma(P)\). We firstly compute the weight of the above \(wAI(P)\) element for \(a = \{s\} \in \gamma\), and hence we derive Table 23. Afterwards, we obtain the weight of \(z\) for \(a = \{s, r_1, r_2\} \in \gamma\), as shown in Table 24. For this, we also need the auxiliary Tables 3-10 that we obtained for the weighted Rendezvous scheme. Then the weight of the Atomic Broadcast on \(\gamma\) is obtained as follows:
\[
\|s \otimes (1 \oplus r_1 \otimes r_2)\|\!(\gamma) = \sum_{s \in \gamma} \left( \sum_{a \in a_1 \cup a_2} (\|s\|\{a_1\}) \cdot |1 \oplus r_1 \otimes r_2|\{a_2\} \right)
\]
\[
= \sum_{a \in \gamma} \left( \sum_{s \in a_1 \cup a_2} (\|s\|\{a_1\}) \cdot \left( \sum_{a' \in a_2} (\|1\{a'\}) + \|r_1 \otimes r_2\|\{a'\}) \right) \right)
\]
\[
= \sum_{a \in \gamma} \left( \sum_{s \in a_1 \cup a_2} (\|s\|\{a_1\}) \cdot (\|1\{a_2\}) + \|r_1 \otimes r_2\|\{a_2\}) \right)
\]
\[
= \sum_{a \in \gamma} \left( \sum_{s \in a_1 \cup a_2} (\|s\|\{a_1\}) \cdot (\|1\{a_2\}) + \sum_{a'' \in a_2} (\|r_1\{a_2, 1\}) \cdot \|r_2\|\{a_2, 2\}) \right)
\]
\[
= \sum_{a \in \gamma} \left( \sum_{s \in a_1 \cup a_2} (\|s\|\{a_1\}) \cdot (\|1\{a_2\}) + \sum_{a_2 \in a_2, 1 \cup a_2, 2} (\|r_1\{a_2, 1\}) \cdot \|r_2\|\{a_2, 2\}) \right)
\]
\[
= k_s + (k_s \cdot k_{r_1} \cdot k_{r_2})
\]
In the Viterbi semiring for instance, the above value represents the maximum weight between \(k_s\) and \(k_s \cdot k_{r_1} \cdot k_{r_2}\) for the Atomic Broadcast scheme.

**Weighted Causality Chain:** In a Causality Chain scheme, a message is either not received by none of the two receivers, or it is received by the first one of them, or by both of them. Thus, there are three possible interactions, namely \(s, sr_1, sr_1r_2\). The \(wAI(P)\) element describing the weighted Causality Chain is
\[z = s \otimes (1 \oplus r_1 \otimes (1 \oplus r_2)).\]
We let $\gamma = \{\{s\}, \{s, r_1\}, \{s, r_1, r_2\}\} \in \Gamma(P)$. In order to compute the weight of $z$ on $\gamma$ we firstly obtain the weight of $z$ for $a = \{s\} \in \gamma$ derived in Table 25. Then we compute the weight of $z$ for $a = \{s, r_1\} \in \gamma$, shown in Table 26, and finally, in Table 27 is derived the resulting weight for $a = \{s, r_1, r_2\} \in \gamma$. For these computations we use the auxiliary Tables 28-35. Therefore, the weight for implementing the Causality Chain scheme in $\gamma$ is computed as follows:

$$
\| s \otimes (1 \oplus r_1 \otimes (1 \oplus r_2)) \| (\gamma)
$$

$$
= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\| s \| \{a_1\} \cdot \| 1 \oplus r_1 \otimes (1 \oplus r_2) \| \{a_2\})) \right)
$$

$$
= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\| s \| \{a_1\} \cdot \left( \sum_{a' \in \{a_2\}} (\| 1 \| \{a'\}) + \| r_1 \otimes (1 \oplus r_2) \| \{a'\})) \right)
$$

$$
= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\| s \| \{a_1\} \cdot (\| 1 \| \{a_2\}) + \sum_{a'' \in \{a_2\}} (\sum_{a'' = a_{2,1} \cup a_{2,2}} (\| r_1 \| \{a_{2,1}\} + \| 1 \oplus r_2 \| \{a_{2,2}\})) \right)
$$

$$
= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\| s \| \{a_1\} \cdot (\| 1 \| \{a_2\}) + \sum_{a_2 = a_{2,1} \cup a_{2,2}} (\| r_1 \| \{a_{2,1}\})) \right)
$$

$$
= \sum_{a \in \gamma} \left( \sum_{a = a_1 \cup a_2} (\| s \| \{a_1\} \cdot (\| 1 \| \{a_2\}) + \sum_{a'' = a_{2,1} \cup a_{2,2}} (\| r_1 \| \{a_{2,1}\}) \right)
$$

$$
= k_s + (k_s + (k_s \cdot k_{r_1})) + (k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_1} \cdot k_{r_2}))
$$

$$
= k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_1} \cdot k_{r_2}).
$$

Consider for instance the $\mathbb{R}_{\text{min}}$ semiring. Then the above value represents for weighted Causality Chain the minimum sum of the weights associated with the ports occurring in the interactions of $\gamma$. 

22
5 The Weighted Algebra of Connectors

In architectures, the components communicate through their ports and their allowed interactions are defined by the imposed coordination scheme. In turn, connectors specify the synchronization constraints among these interactions by relating a set of typed ports. Types extend ports with synchronization modes, and specifically in this work, with Rendezvous and Broadcast mode [BS08a]. Rendezvous requires that all the components should interact simultaneously, while in Broadcast, a component initiates the interactions with some of the rest components.

In this section, we are interested in encoding the weight of connectors in architectures. For this, we study the weighted Algebra of Connectors over \( P \) and \( K \), that extends \( wAI(P) \) with two typing operators, namely triggers and synchrons that correspond to Rendezvous and Broadcast mode, respectively. We prove several properties for \( wAC(P) \) and we show that it can encode sufficiently several connectors in the weighted setup.

Let \( P \) be a set of ports. Similarly to \( wAI(P) \), we assign to each port \( p \in P \) a unique weight from \( K \), denoted by \( k_p \).

**Definition 4.** Let \( P \) be a set of ports, such that \( 0,1 \notin P \). The syntax of the weighted Algebra of Connectors (\( wAC(P) \) for short) over \( P \) and \( K \) is given by

\[
\begin{align*}
\sigma &:= [0] | [1] | [p] | [[\zeta]] \quad \text{(synchron)} \\
\tau &:= [0]’ | [1]’ | [p]’ | [[\zeta]’] \quad \text{(trigger)} \\
\zeta &:= \sigma \mid \tau \mid \zeta \oplus \zeta \mid \zeta \odot \zeta
\end{align*}
\]

where \( p \in P \), “ \( \oplus \) ” denotes the weighted union operator, “ \( \odot \) ” denotes the weighted fusion operator, and “ \( [\_] \) ”, “ \( [\_]’ \) ” are the unary synchron and trigger typing operators, respectively.

Similarly to the Algebra of Connectors introduced in [BS08a], the new operators in \( wAC(P) \), specifically, “ \( [\_] \) ” and “ \( [\_]’ \) ”, are assigned the characterization “typing operators” since they encode the type of synchronization mode applied to the respective ports. Particularly, a trigger is responsible for initiating an interaction, while a synchron requires to interact simultaneously with other ports. It should be clear that the typing operators in \( wAC(P) \) coincide with the ones from the work of [BS08a], since the synchronization mode that they encode is a qualitative feature. The difference is that in this work, the typing operators are applied among connectors in the weighted setup.

Weighted union has the same meaning both in \( wAC(P) \) and \( wAI(P) \), while weighted fusion is a generalization of weighted synchronization in \( wAI(P) \). This is clarified by the semantics of \( wAC(P) \) presented in Definition 5.

We call \( \zeta \) a \( wAC(P) \) connector over \( P \) and \( K \). Whenever the latter are understood, we simply refer to \( \zeta \) as a \( wAC(P) \) connector. Also, we write \( [\zeta]^\alpha \) for \( \alpha \in \{0,1\} \) to denote a typed \( wAC(P) \) connector. When \( \alpha = 0 \), the \( wAC(P) \) connector is a synchron, otherwise for \( \alpha = 1 \) it is a trigger. When the type is irrelevant we write \( [\_]’ \). Moreover, we call \( \zeta \) a fusion-\( wAC(P) \) connector when \( \zeta = [\zeta_1]^\alpha_1 \odot \ldots \odot [\zeta_n]^\alpha_n \), where \( \zeta_1, \ldots, \zeta_n \in wAC(P) \) and \( \alpha_1, \ldots, \alpha_n \in \{0,1\} \).

Next we introduce the notations relating to the degree of a \( wAC(P) \) connector \( \zeta \). In particular, for a fusion-\( wAC(P) \) connector \( \zeta = [\zeta_1]^\alpha_1 \odot \ldots \odot [\zeta_n]^\alpha_n \), where \( \zeta_1, \ldots, \zeta_n \in wAC(P) \), we denote by \( \#T\zeta \) the number of its trigger elements, which we call the degree of \( \zeta \). Then for \( \zeta = \bigoplus_{i \in [n]} \zeta_i \), where all \( \zeta_i \) are fusion-\( wAC(P) \) connectors, we let \( \#T\zeta = \max \{ \#T\zeta_i \mid i \in [n] \} \). We say that \( \zeta \) has a
strictly positive degree iff \( \min \{ \#\, \zeta_i \mid i \in [n]\} > 0 \). We use the notion of the degree in the semantics of \( wAC(P) \) and in Section 7, for proving a concept of congruence relation between fusion-\( wAC(P) \) connectors.

The intuition behind the semantics of \( wAC(P) \), presented in Definition 5, is to encode the weight of a \( wAC(P) \) connector according to the coordination scheme imposed on an architecture. This implies that given an interactions set \( \gamma \), we would like to get for each \( wAC(P) \) connector a value from \( K \). For this, we firstly relate each \( wAC(P) \) connector with a \( wAI(P) \) element. In turn, we obtain the weight of the \( wAC(P) \) connector through the semantics of \( wAI(P) \). Formally, the semantics of a \( wAC(P) \) connector \( \zeta \) over a set of ports \( P \) and the semiring \( K \) is defined by the function \( |\cdot| : wAC(P) \rightarrow wAI(P) \), which is presented below in Definition 5. For every \( \zeta \in wAC(P) \) we firstly compute its \( wAI(P) \) element \( |\zeta| \), and then, applying the semantics of \( wAI(P) \), we compute its weight over an interactions set \( \gamma \in \Gamma(P) \) by the polynomial \(||\zeta||| \in K(\Gamma(P))\).

**Definition 5.** Let \( \zeta \) be a \( wAC(P) \) connector over \( P \) and \( K \). The semantics of \( \zeta \) is a \( wAI(P) \) element defined by the function \( |\cdot| : wAC(P) \rightarrow wAI(P) \) as follows:

- \(|p| = p\), for \( p \in P \cup \{0, 1\}\),
- \(|p'p| = p\), for \( p \in P \cup \{0, 1\}\),
- \(|\zeta| = |\zeta|\),
- \(|\zeta'\zeta| = |\zeta|\),
- \(|\zeta_1 \oplus \zeta_2| = |\zeta_1| \oplus |\zeta_2|\),
- \(|\zeta_1 \otimes \zeta_2| = |\zeta_1| \otimes |\zeta_2|\),
- \(|\zeta_1^{\alpha_1} \otimes \ldots \otimes \zeta_n^{\alpha_n}| = \bigoplus_{\substack{i \in [n], \alpha_i = 1 \atop k \neq i}} \left( |\zeta_i| \otimes \bigotimes_{\substack{\alpha_k \in \{0, 1\} \atop k \neq i}} (1 \oplus |\zeta_k|) \right)\), where \( \#\, \zeta_1^{\alpha_1} \otimes \ldots \otimes \zeta_n^{\alpha_n} > 0 \)

and \( \alpha_1, \ldots, \alpha_n \in \{0, 1\}\).

Observe that in the last case of the \( wAC(P) \) semantics, the index \( k \) refers both to connectors which act as triggers and to connectors which act as synchrons.

Obviously, we can extend the application of the weighted fusion operator between synchrons as well as the union operator for more than two \( wAC(P) \) connectors. Specifically, for \( \zeta_1, \ldots, \zeta_n \in wAC(P) \) it holds that

\[ |\otimes_{i \in [n]} \zeta_i| = |\otimes_{i \in [n]} |\zeta_i|\] and

\[ |\oplus_{i \in [n]} \zeta_i| = |\oplus_{i \in [n]} |\zeta_i|\].

**Remark 5.** Note that the weighted union operator between \( wAC(P) \) connectors does not require any further typing operator. On the other hand, the weighted fusion operator serves to encode a \( wAI(P) \) element for a Rendezvous or Broadcast type of synchronization, which necessitates the typing of the involved \( wAC(P) \) connectors. Specifically, we need to distinguish the following cases, relating to the last two semantics of \( wAC(P) \), respectively.

1. No component initiates the communication and all of them interact simultaneously. This is a case of synchronization where each one of the respective \( wAC(P) \) connectors is typed with a synchron. In turn, by applying the weighted fusion operator we derive the \( wAI(P) \) elements for synchronizing all of the involved \( wAC(P) \) connectors.

24
2. At least one of involved components eagers to initiate the communication, i.e., at least one of the \( wAC(P) \) connectors is typed with a trigger. Then the idea behind these fusion semantics is to fix at each time the \( wAC(P) \) connector that will act as a trigger and to compute the \( wAI(P) \) element by treating the rest of the triggers as synchrons, along with the actual synchrons.

Applying the semantics of \( wAI(P) \) we can derive in turn, the weight of the \( wAC(P) \) connectors for the above modes of synchronization. To simplify the semantics of \( wAI(P) \), we restricted the computation on two \( wAC(P) \) connectors for the above cases but the last one, where we obviously need to consider a finite but arbitrary number of triggers and synchrons.

Next, for simplicity we write 0, 1, \( p \), for \([0],[1],[p]\), respectively, and \( 0', 1', p' \), for \([0'],[1'],[p']\), respectively. For instance, \([p]''\) is written \([p]'\), \( [p] \oplus [q]'\) is written \([p] \oplus q'\), \([p] \oplus [q]'\) is written \([p \oplus q]'\), and \([p]' \otimes [q]\) is written \([p] \otimes [q]\).

It should be clear that given a \( \zeta \in wAC(P) \), in order to proceed the computations on \(|\zeta| \in wAI(P)\), we need to consider the corresponding equivalence class and apply Corollary 1, i.e., that \((wAI(P)/ \equiv, \oplus, \otimes, 0, 1)\) is a commutative and idempotent semiring. Though, for simplicity, in the rest of the paper, we identify \( [\zeta] \) with the representative \(|\zeta|\).

In the next example, we clarify the above conventions in our notations.

**Example 2.** Consider the ports \( p,q,r \in P \). We apply the weighted fusion operator to \( wAC(P) \) connectors \( p \) and \( q \oplus r \). For this, we need to specify a typing operator on them. We choose to synchronize the connectors \( p \) and \( q \oplus r \). Therefore, the resulting connector is \( [p] \otimes [q \oplus r] \) and its \( wAI(P) \) element is computed as follows:

\[
| [p] \otimes [q \oplus r] | = |p| \otimes |q \oplus r| \\
= |p| \otimes (|q| \oplus |r|) \\
= p \otimes (q \oplus r) \\
= (p \otimes q) \oplus (p \otimes r).
\]

On the other hand, if we let \( p \) serve as a trigger, then we obtain the connector \( [p]'' \otimes [q \oplus r] \). In turn, its \( wAI(P) \) element is computed as follows:

\[
| [p]' \otimes [q \oplus r] | = |p| \otimes (1 \oplus |q \oplus r|) \\
= |p| \otimes (1 \oplus (|q| \oplus |r|)) \\
= p \otimes (1 \oplus (q \oplus r)) \\
= p \otimes (1 \oplus q \oplus r) \\
= p \oplus (p \otimes q) \oplus (p \otimes r).
\]

Next we introduce a notion of equivalence for \( wAC(P) \) connectors and by their equivalence classes we derive several nice properties for \( wAC(P) \).

Two connectors \( \zeta_1, \zeta_2 \in wAC(P) \) are equivalent, and we write \( \zeta_1 \equiv \zeta_2 \) when \( |\zeta_1| = |\zeta_2| \), i.e., when they return the same \( wAI(P) \) elements. Obviously, this in turn implies that \( |\zeta_1| \|(\gamma) = |\zeta_2| \|(\gamma) \) for every \( \gamma \in \Gamma(P) \), i.e., equivalent \( wAC(P) \) connectors return the same weight on the same interactions set \( \gamma \). Clearly “\( \equiv \)” is an equivalence relation. We define the quotient set \( wAC(P)/ \equiv \) of “\( \equiv \)” on \( wAC(P) \). For every \( \zeta \in wAC(P) \) we simply denote by \( \overline{\zeta} \) its equivalence class. We define on \( wAC(P)/ \equiv \) the operations

\[
\overline{\zeta_1} \oplus \overline{\zeta_2} = \overline{\zeta_1 \oplus \zeta_2} \tag{05}
\]
Let \( \alpha, \beta \in \{0, 1\} \). The above two operations are well-defined, since, if \( \zeta_1 = \zeta_3 \) and \( \zeta_2 = \zeta_4 \), then we get \( \zeta_1 \otimes \zeta_2 = \zeta_3 \otimes \zeta_4 \) and \( [\zeta_1]^{\alpha} \otimes [\zeta_2]^{\beta} = [\zeta_3]^{\alpha} \otimes [\zeta_4]^{\beta} \) for every \( \zeta_1, \zeta_2, \zeta_3, \zeta_4 \in wAC(P) \) and \( \alpha, \beta \in \{0, 1\} \).

In [BS08a], the authors proved several properties for the Algebra of Connectors using axioms. In the sequel, we show that \( wAC(P) \) acknowledges the respective properties in the weighted setup. For this, we use the equivalence classes of \( wAC(P) \).

Next proposition states that connector \([1]\) is the neutral element of weighted fusion operator over the quotient set \( wAC(P)/\equiv \).

**Proposition 2.** Let \( \zeta \in wAC(P) \) and \( \alpha \in \{0, 1\} \). Then

\[
[\zeta]^{\alpha} \otimes [1] = [\zeta]^{\alpha} = [1] \otimes [\zeta]^{\alpha}.
\]

**Proof.** We use the equivalence classes of \( wAC(P) \) and we apply Definition 5 and Corollary 1. \( \blacksquare \)

In the following proposition, we show that \( wAC(P) \) satisfies the associativity property of weighted fusion operator when a specific type, synchron or trigger, is applied for simple grouping. Associativity property is useful since it indicates that independently of the order in which we apply the typing operator, the resulting synchronization remains the same.

**Proposition 3.** Let \( \zeta_1, \zeta_2, \zeta_3 \in wAC(P) \). Then

i) \( [\zeta_1 \otimes [\zeta_2] \otimes [\zeta_3] = \zeta_1 \otimes [\zeta_2 \otimes [\zeta_3] \right]

ii) \( [\zeta_1 \otimes [\zeta_2'] \otimes [\zeta_3'] \right] = \zeta_1 \otimes [\zeta_2' \otimes [\zeta_3'] \right].

**Proof.** In order to prove the above equalities we apply Corollary 1.

i) We compute the left part of the first equality as follows:

\[
[\zeta_1 \otimes [\zeta_2] \otimes [\zeta_3] = [\zeta_1 \otimes [\zeta_2] \otimes [\zeta_3] = \{ \zeta \in wAC(P) \mid [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\[
= \{ \zeta \in wAC(P) \mid [\zeta] = [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\[
= \{ \zeta \in wAC(P) \mid [\zeta] = [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\[
= \{ \zeta \in wAC(P) \mid [\zeta] = [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\]

Now, we compute the right part of the first equality as follows:

\[
[\zeta_1 \otimes [\zeta_2] \otimes [\zeta_3] = [\zeta_1 \otimes [\zeta_2] \otimes [\zeta_3] = \{ \zeta \in wAC(P) \mid [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\[
= \{ \zeta \in wAC(P) \mid [\zeta] = [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\[
= \{ \zeta \in wAC(P) \mid [\zeta] = [\zeta_1] \otimes [\zeta_2] \otimes [\zeta_3] \}

\]

26
Finally, for the right part in the second equality, we have:

\[
\{ \zeta \in wAC(P) \mid \zeta = |\zeta_1| \otimes (|\zeta_2| \otimes |\zeta_3|) \}\n\]

Consequently, \([|\zeta_1| \otimes |\zeta_2|] \otimes |\zeta_3|\) and \([|\zeta_2| \otimes |\zeta_3|]\) are equal.

ii) For the left part of the second equality we have:

\[
\left[ [\zeta_1'] \otimes [\zeta_2'] \right]' \otimes [\zeta_3]' = \left[ [\zeta_1'] \otimes [\zeta_2'] \right]' \otimes [\zeta_3]' \]

\[
= \left\{ \zeta \in wAC(P) \mid \zeta \equiv \left[ [\zeta_1'] \otimes [\zeta_2'] \right]' \otimes [\zeta_3]' \right\}
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left[ [\zeta_1'] \otimes [\zeta_2'] \right]' \otimes [\zeta_3]' \right\}
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left( (|\zeta_1| \otimes (1 \oplus |\zeta_3|)) \oplus (|\zeta_2| \otimes (1 \oplus |\zeta_3|)) \right) \otimes (1 \oplus |\zeta_3|)
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left( (|\zeta_1| \otimes (1 \oplus |\zeta_3|)) \oplus (|\zeta_2| \otimes (1 \oplus |\zeta_3|)) \right) \otimes (1 \oplus |\zeta_3|)
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left( (|\zeta_1| \otimes (1 \oplus |\zeta_3|)) \oplus (|\zeta_2| \otimes (1 \oplus |\zeta_3|)) \right) \otimes (1 \oplus |\zeta_3|)
\]

Finally, for the right part in the second equality, we have:

\[
\left[ [\zeta_1'] \otimes [\zeta_2'] \otimes [\zeta_3]' \right] = \left[ [\zeta_1'] \otimes [\zeta_2'] \otimes [\zeta_3]' \right]'
\]

\[
= \left\{ \zeta \in wAC(P) \mid \zeta = [\zeta_1'] \otimes [\zeta_2'] \otimes [\zeta_3]' \right\}
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = [\zeta_1'] \otimes [\zeta_2'] \otimes [\zeta_3]' \right\}
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left( (|\zeta_1| \otimes (1 \oplus [\zeta_2'] \oplus [\zeta_3]')) \oplus (|\zeta_2| \otimes [\zeta_3]') \oplus (1 \oplus |\zeta_1|) \right) \right\}
\]

\[
= \left\{ \zeta \in wAC(P) \mid |\zeta| = \left( (|\zeta_1| \otimes (1 \oplus [\zeta_2'] \oplus [\zeta_3]')) \oplus (|\zeta_2| \otimes [\zeta_3]') \oplus (1 \oplus |\zeta_1|) \right) \right\}
\]
Hence, \( \zeta \in \text{wAC}(P) \mid |\zeta| = \left( |\zeta_1| \otimes \left( 1 \oplus |\zeta_2| \oplus (|\zeta_1| \otimes |\zeta_3|) \right) \right) \oplus \left( |\zeta_2| \oplus (|\zeta_2| \otimes |\zeta_3|) \right) \oplus (1 \oplus |\zeta_1|) \right) \right) \)}

\[ = \{ \zeta \in \text{wAC}(P) \mid |\zeta| = \left( |\zeta_1| \otimes \left( 1 \oplus |\zeta_2| \oplus (|\zeta_1| \otimes |\zeta_3|) \right) \right) \oplus \left( |\zeta_2| \oplus (|\zeta_2| \otimes |\zeta_3|) \right) \oplus (1 \oplus |\zeta_1|) \right) \right) \}

\[ = \{ \zeta \in \text{wAC}(P) \mid |\zeta| = \left( |\zeta_1| \oplus |\zeta_2| \oplus (|\zeta_1| \otimes |\zeta_3|) \right) \oplus \left( |\zeta_2| \oplus (|\zeta_2| \otimes |\zeta_3|) \right) \oplus (1 \oplus |\zeta_1|) \right) \}

\[ = \{ \zeta \in \text{wAC}(P) \mid |\zeta| = |\zeta_1| \oplus |\zeta_2| \oplus (|\zeta_1| \otimes |\zeta_3|) \oplus (1 \oplus |\zeta_1|) \right) \}

Proof. We apply Corollary 1 and we compute the equivalence classes of the involved \text{wAC}(P)\) connectors.

Proposition 4. Let \( \zeta_1, \zeta_2 \in \text{wAC}(P) \) and \( \alpha, \beta \in \{0, 1\} \). Then

\[ a) \left[ \left[ \zeta_1 \right]^\alpha \right]^\beta = \left[ \zeta_1 \right]^\beta \]

\[ b) \left[ \zeta_1 \oplus \zeta_2 \right]^\alpha = \left[ \zeta_1 \right]^\alpha \oplus \left[ \zeta_2 \right]^\alpha \]

\[ c) \left[ \zeta_1 \right]^\alpha \oplus \left[ \zeta_2 \right]^\beta = \left[ \zeta_2 \right]^\beta \oplus \left[ \zeta_1 \right]^\alpha \]

\[ d) \left[ \zeta_1 \right]^\alpha \otimes \left[ \zeta_2 \right]^\beta = \left[ \zeta_2 \right]^\beta \otimes \left[ \zeta_1 \right]^\alpha \].

Proof. We apply Corollary 1 and we compute the equivalence classes of the involved \text{wAC}(P)\) connectors. ■

Next proposition serves for simplifying the computation of the semantics for some of the \text{wAC}(P)/ = \equiv elements. Specifically, part a) indicates that in case we apply two typing operators to a \text{wAC}(P)\) connector, then the inner typing can be omitted, part b) expresses that the typing of weighted union of two \text{wAC}(P)\) connectors coincides with the weighted union of their respective typing, while part c) and d) verify the commutativity property of the weighted union and the weighted fusion operator with respect to the typing operator when a specific type, synchron or trigger, is applied for simple grouping.

In the sequel, we apply \text{wAC}(P)\) for modeling several connectors in the weighted setup. Then we apply the semantics of \text{wAC}(P)\) and in turn by the semantics of the respective \text{wAI}(P)\) elements, we are able to derive the weight of implementing each \text{wAC}(P)\) connector for a given interactions set.

28
For the $wAC(P)$ connectors presented in the sequel, we follow the representation considered in [BS08a]. In particular, we use triangles and circles in order to represent triggers and synchrons, respectively. Then we connect the involved $wAC(P)$ connectors with lines which are labeled by the respective weight from $K$. In turn, we draw the resulting $wAC(P)$ connector incrementally, as it described in the following examples.

Example 3. We present the $wAC(P)$ connectors of the coordination schemes Rendezvous, Broadcast, Atomic Broadcast and Causality Chain, in the weighted setup, described in Example 1. Recall that we considered a sender and two receivers with ports $s, r_1, r_2$, respectively, and hence, $P = \{s, r_1, r_2\}$. Also, we denoted by $k_s, k_{r_1}, k_{r_2}$, the weight associated with the ports $s, r_1, r_2$, respectively.

**Weighted Rendezvous**: In Rendezvous coordination scheme, the involved components should be strongly synchronized. Hence, the respective connector should not contain any trigger operator, but it would rather apply the synchron typing in each of the terms. Then in the weighted setting the connector should encode the “cost” for this synchronization. Specifically, the $wAC(P)$ connector for weighted Rendezvous in our example is given by

$$\zeta = [s] \otimes [r_1] \otimes [r_2].$$

The $wAC(P)$ connector is shown in Figure 1a. In particular, the $wAC(P)$ connectors $s, r_1, r_2$, are represented with circles, since they are all typed with synchrons. Moreover, the connection is performed simultaneously, and hence the three $wAC(P)$ connectors occur at the same level. Then we obtain the $wAI(P)$ element of the connector as follows:

$$|\zeta| = |[s] \otimes [r_1] \otimes [r_2]|$$

Figure 1: The $wAC(P)$ connectors of four coordination schemes.
The weight of the above $wAI(P)$ element was computed in the corresponding part of Example 1 for $\gamma = \{\{s, r_1, r_2\}\}$, and we showed that it equals to $k_s \cdot k_{r_1} \cdot k_{r_2}$.

**Weighted Broadcast:** In Broadcast communication, a component should trigger the interaction with the rest components. As a result, we should allow a trigger typing along with some synchrons in the respective connector. For our example, the sender initiates the interactions with some of the two receivers, and hence we consider the trigger typing for $s$, while $r_1$ and $r_2$ are typed with synchrons. In turn, the $wAC(P)$ connector for Broadcast scheme is

$$\zeta = \pi s \otimes \pi r_1 \otimes \pi r_2.$$ 

The connector is shown in Figure 1b, where the trigger for $s$ and the synchrons for $r_1$ and $r_2$ are indicated by a triangle and two circles, respectively, all occurring at the same level. Then the $wAI(P)$ element of the above connector is computed as follows:

$$|\zeta| = |\pi s \otimes \pi r_1 \otimes \pi r_2|$$

$$= |\pi s \otimes (1 \oplus |\pi r_1| \otimes |\pi r_2|)$$

$$= \pi s \otimes (1 \oplus |r_1| \otimes |r_2|).$$

The weight of the latter was obtained in the weighted Broadcast scheme of Example 1 for $\gamma = \{\{s\}, \{s, r_1\}, \{s, r_2\}, \{s, r_1, r_2\}\}$, and it was computed equal to $k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_2}) + (k_s \cdot k_{r_1} \cdot k_{r_2})$. Then in $\mathbb{P}_{\min}$ semiring the resulting value corresponds to the minimum sum of the weights associated with the ports occurring in the interactions of $\gamma$. In turn, if we would interpret the weight as the energy consumption associated with each of the allowed interactions, then the architecture could opt for the most efficient one.

**Weighted Atomic Broadcast:** Similarly to Broadcast, in the Atomic Broadcast scheme there is a component that initiates the interactions with the other components. The difference is that it requires the communication with all the other components or with none of them, and hence we should obtain the overall weight for the two cases. For our example, the unique sender is assigned a trigger typing, while we compute the weight of the simultaneous presence or absence of both receivers, by the application of the synchron typing to the weighted fusion of $[r_1]$ and $[r_2]$. The $wAC(P)$ connector for the weighted Atomic Broadcast is given by

$$\zeta = \pi s \otimes \pi [r_1] \otimes \pi [r_2]$$

and is shown in Figure 1c. The synchrons for $r_1$ and $r_2$ are denoted with circles and occur at the same level, the trigger for $s$ is represented with a triangle, and a circle is used for the synchronization of the latter with $[r_1] \otimes [r_2]$. The corresponding $wAI(P)$ element of $\zeta$ is obtained as follows:

$$|\zeta| = |\pi s \otimes \pi [r_1] \otimes \pi [r_2]|$$

$$= |\pi s \otimes (1 \oplus |r_1| \otimes |r_2|)|$$

$$= |\pi s \otimes (1 \oplus |r_1| \otimes |r_2|)|$$
\[ = s \otimes (1 \oplus r_1 \otimes r_2). \]

The weight of the above \( wAI(P) \) element was computed in the corresponding scheme in Example 1 for \( \gamma = \{\{s\}, \{s, r_1, r_2\}\} \), and equals to \( k_s + (k_s \cdot k_{r_1} \cdot k_{r_2}) \). Then in Viterbi semiring the resulting value corresponds to executing the interaction with the maximum probability.

**Weighted Causality Chain:** A \( wAC(P) \) connector for the Causality Chain, should apply a trigger typing to the sender, i.e., to the component that initiates the interaction as well as to any of the involved receivers but the “last” one. Then we encode the weight of the resulting synchronization by the \( wAC(P) \) connector

\[ \zeta = [s]' \otimes ([r_1]' \otimes [r_2]). \]

The connector is shown in Figure 1d where triangles are used to denote the trigger for \( s \) and \( r_1 \), while the synchron for \( r_2 \) is denoted by a circle. Then we synchronize \([r_1]'\) with \([r_2] \) and in turn, we synchronize \([s]' \) with \([r_1]' \otimes [r_2] \). The two synchronizations are depicted with a circle in Fig. 1d, at lower and higher level, respectively. Hence, the \( wAI(P) \) element is computed as follows:

\[
|\zeta| = |[s]' \otimes ([r_1]' \otimes [r_2])|
= |s| \otimes \left( 1 \oplus |[r_1]' \otimes [r_2]| \right)
= |s| \otimes \left( 1 \oplus |r_1| \otimes (1 \oplus |r_2|) \right)
= s \otimes (1 \oplus r_1 \otimes (1 \oplus r_2)).
\]

The weight of the above \( wAI(P) \) element was computed in Example 1 for \( \gamma = \{\{s\}, \{s, r_1\}, \{s, r_1, r_2\}\} \), and equals to \( k_s + (k_s \cdot k_{r_1}) + (k_s \cdot k_{r_1} \cdot k_{r_2}) \). Then in \( \mathbb{R}_{\max} \) semiring the resulting value corresponds to the maximum sum of the weights associated with the ports occurring in the interactions of \( \gamma \).

6 Weighted subalgebras

Next we present two subalgebras of \( wAC(P) \), namely the weighted Algebra of Synchrons and the weighted Algebra of Triggers over \( P \) and \( K \), and consider their properties. The former algebra restricts to synchrons, while the latter involves only triggers.

6.1 The Weighted Algebra of Synchrons

Let \( P \) be a set of ports. We assign to each port \( p \in P \) a unique weight from \( K \), denoted by \( k_p \). Then we consider the subalgebra of \( wAC(P) \) generated by restricting its syntax to synchrons. The resulting algebra over \( P \) and \( K \) is called weighted Algebra of Synchrons and is denoted by \( wAS(P) \).

**Definition 6.** Given a set of ports \( P \), the syntax of the weighted Algebra of Synchrons (\( wAS(P) \) for short) over \( P \) and \( K \) is defined by:

\[
\sigma ::= [0] \mid [1] \mid [p] \mid [\zeta]
\]

\[
\zeta ::= \sigma \mid \zeta \oplus \zeta \mid \zeta \otimes \zeta
\]

where \( p \in P \), \( \sigma \) denotes a synchron element, and \( \zeta \in wAS(P) \).
The weighted operators “⊕” and “⊗” are the weighted union and the weighted fusion operator, respectively, from the syntax of $wAC(P)$. Obviously, we get that Proposition 3i) holds for $wAS(P)$, hence the subalgebra satisfies the associativity property of weighted fusion with respect to the synchron typing operator. Furthermore, $wAS(P)$ satisfies Proposition 4, for $\alpha = 0$ and $\beta = 0$. Intuitively, part a) implies that when we apply the synchron typing operator twice to a $wAS(P)$ connector, then the inner typing can be omitted, part b) expresses that the synchron of the weighted union of two $wAS(P)$ connectors coincides with the weighted union of their respective synchrons, and parts c) and d) indicate that the weighted union and weighted fusion operators, respectively, satisfy the commutativity property for the synchron typing operator.

6.2 The Weighted Algebra of Triggers

Let $P$ be a set of ports. We assign to each port $p \in P$ a unique weight from $K$, denoted by $k_p$.

Then we consider the subalgebra of $wAC(P)$ generated by restricting its syntax to triggers. The resulting algebra over $P$ and $K$ is called weighted Algebra of Triggers and is denoted by $wAT(P)$.

Definition 7. Given a set of ports $P$, the syntax of the weighted Algebra of Triggers ($wAT(P)$ for short) over $P$ and $K$ is defined by:

$$
\begin{align*}
\tau &::= [0]' | [1]' | [p]' | [\zeta]' \\
\zeta &::= \tau | \zeta \oplus \zeta | \zeta \otimes \zeta
\end{align*}
$$

where $p \in P$, $\tau$ denotes a trigger element, and $\zeta \in wAT(P)$.

The operators “⊕” and “⊗” are the weighted union and the weighted fusion operator, respectively, from the syntax of the $wAC(P)$. Moreover, by Proposition 3ii) of the previous section we obtain that $wAT(P)$ satisfies the associativity property of weighted fusion with respect to trigger typing operator. Furthermore, $wAT(P)$ satisfies Proposition 4, for $\alpha = 1$ and $\beta = 1$, where part a) implies that we doubly type a $wAT(P)$ connector with a trigger, then the inner one can be omitted, part b) expresses that the trigger typing of the weighted union of two $wAT(P)$ connectors coincides with the weighted union of their respective triggers, and parts c) and d) imply that weighted union and fusion operators, respectively, satisfy the commutativity property for the trigger typing operator.

Recall that the $wAC(P)$ connector [1] served as the neutral element of weighted fusion in $wAC(P)$. It should be clear though, that by construction, the weighted Algebra of Triggers does not contain [1], and hence weighted fusion loses its neutral element. To tackle this, we restrict the equivalence relation “≡” defined in $wAC(P)$ (cf. Section 5) only to the trigger elements, i.e., to $\zeta \in wAT(P)$. Hence, we define the quotient set $wAT(P)/ \equiv$ of “≡” on $wAT(P)$. Then we trivially get that $[0]'$ can serve as an alternative neutral element for the weighted fusion operator in $wAT(P)$, i.e., $[\zeta]' \otimes [0]' = [\zeta]' = [0]' \otimes [\zeta]'$ for $\zeta \in wAT(P)$.

7 On congruence relation for fusion-$wAC(P)$ connectors

In Section 5, we defined the semantics of a connector $\zeta \in wAC(P)$ as an element $z \in wAI(P)$. Then the semantics of the latter was interpreted as a polynomial $\|z\| \in K(\Gamma(P))$. Moreover, we stated that two $wAC(P)$ connectors $\zeta_1, \zeta_2$, are equivalent, i.e., $\zeta_1 \equiv \zeta_2$ when $|\zeta_1| = |\zeta_2|$. By the
semantics of \(wAI(P)\) this implied that \(\| \cdot \|_\gamma(\cdot) = \| \cdot \|_\gamma'\|_\gamma(\cdot)\) for every \(\gamma \in \Gamma(P)\), i.e., equivalent \(wAC(P)\) connectors return the same weight on the same interactions set.

In this section, we are interested in the congruence problem of \(wAC(P)\) connectors. In [BS08a], the authors introduced a congruence relation for the Algebra of Connectors and provided conditions for proving their congruence. Congruence relation is important because in contrast to equivalence relation, it allows to use connectors interchangeably whenever it is required and without causing undesirable alterations in the architecture.

However, it occurs that providing a congruence relation in the weighted setup, it is not an easy task. Congruence relation implies that given two equivalent elements, if we apply any operator from the algebra on them, then we obtain again two equivalent elements [BCD00, BLM06, DKV09]. In our setting, applying the weighted fusion operator to \(wAC(P)\) connectors would require specifying a typing. In [BS08a], the authors resolved this issue using the syntactic equality resulting by the several axioms defined for the connectors. In the weighted framework we can only derive results by semantic equivalence, and hence we cannot follow a similar method. For this, we restrict the congruence problem on fusion-\(wAC(P)\) connectors which are typed by definition. In particular, we show that two equivalent fusion-\(wAC(P)\) connectors are not in general interchangeable. In turn, we define a concept of congruence relation for fusion-\(wAC(P)\) connectors, and we provide two theorems for proving such a congruence by extending the respective results from [BS08a] in the weighted setup.

Note that our concept of congruence relation is not a “true” congruence in terms that (i) it does not apply to any \(wAC(P)\) connector and (ii) it takes equivalent fusion-\(wAC(P)\) connectors and returns equivalent \(wAC(P)\) connectors. Extending our results for studying a congruence relation for \(wAC(P)\) connectors in general, is an interesting open problem that is left as future work.

**Example 4.** Consider the set of ports \(P = \{p, q\}\) and let \(k_p, k_q\), denote the weight of ports \(p, q\), respectively. It can be easily verified that the fusion-\(wAC(P)\) connectors \([p]_p'\) and \([p]\) are equivalent, i.e., \([p]_p' \equiv [p]\). In order to prove the equivalence, we compute the \(wAI(P)\) elements of the above connectors and we have \(\| [p]_p' \| = p\) and \(\| [p] \| = p\), respectively.

Though, we show that the \(wAC(P)\) connectors \([p]_p' \otimes [q]_p\) and \([p] \otimes [q]\) are not equivalent. Indeed, the \(wAI(P)\) elements for each one of the above \(wAC(P)\) connectors is \(\| [p]_p' \otimes [q]_p \| = [p] \otimes (1 \oplus [q]) = p \oplus (p \otimes q)\) and \(\| [p] \otimes [q] \| = [p] \otimes [q] = p \otimes q\), respectively. We let \(\gamma = \{\{p\}, \{p, q\}\} \in \Gamma(P)\). Then we compute the weight of the former \(wAI(P)\) element on \(\gamma\) and we get \(\| p \otimes (1 \oplus q) \| (\gamma) = k_p + (k_p \cdot k_q)\). On the other hand, the weight of the latter \(wAI(P)\) element on \(\gamma\) is \(\| p \otimes q \| (\gamma) = k_p \cdot k_q\). Consequently, we infer that \([p]_p' \otimes [q]_p\) and \([p] \otimes [q]\) are not equivalent.

By the previous example it occurs that when equivalent fusion-\(wAC(P)\) connectors are differently typed, then the application of the weighted fusion operator does not preserve the equivalence. For this, next we introduce a concept of congruence relation for fusion-\(wAC(P)\) connectors.

**Definition 8.** We denote by \(\equiv\) the largest congruence relation for fusion-\(wAC(P)\) connectors contained in \(\equiv\) of \(wAC(P)\), i.e., the largest relation satisfying the following: For fusion-\(wAC(P)\) connectors \(\zeta_1, \zeta_2\) and \(r \notin P\),

\[
\zeta_1 \equiv \zeta_2 \Rightarrow \forall E \in wAC(P \cup \{r\}), E(\zeta_1/r) \equiv E(\zeta_2/r)
\]

where \(E(\zeta/r)\) denotes the expression obtained from \(E\) by replacing all occurrences of \(wAC(P)\) connector \(r\) by \(\zeta\).
Since the weighted fusion operator does not preserve the equivalence of fusion-$wAC(P)$ connectors, we need further conditions for proving their congruence. According to our first result presented below, it suffices to assign identical typing on equivalent fusion-$wAC(P)$ connectors. Hence, we show that two equivalent fusion-$wAC(P)$ connectors which are similarly typed, they are also congruent.

**Theorem 1.** Let $\zeta_1, \zeta_2$ be fusion-$wAC(P)$ connectors. Then

$$\zeta_1 \equiv \zeta_2 \Leftrightarrow |\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha} \text{ for any } \alpha \in \{0, 1\}.$$ 

**Proof.** Firstly we prove the right-to-left implication. Assume that $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha}$, i.e., for every $E \in wAC(P \cup \{r\})$ we have $E(\zeta_1^{\alpha}/r) \equiv E(\zeta_2^{\alpha}/r)$, and hence $E(\zeta_1^{\alpha}/r) = E(\zeta_2^{\alpha}/r)$.

For $E = r \in wAC(P \cup \{r\})$ we get that $E(\zeta_1^{\alpha}/r) = |\zeta_1|^{\alpha}$ and $E(\zeta_2^{\alpha}/r) = |\zeta_2|^{\alpha}$. By assumption $|E(\zeta_1^{\alpha}/r)| = |E(\zeta_2^{\alpha}/r)|$, i.e., $|\zeta_1|^{\alpha} = |\zeta_2|^{\alpha}$. We consider the following two cases:

- For $\alpha = 0$ we have that $|\zeta_1| = |\zeta_2| \Rightarrow |\zeta_1| = |\zeta_2|$.
- For $\alpha = 1$ we have that $|\zeta_1'| = |\zeta_2'| \Rightarrow |\zeta_1| = |\zeta_2|$.

In any case, we conclude that $|\zeta_1| = |\zeta_2|$.

Now we prove the left-to-right implication. We assume that $\zeta_1 \equiv \zeta_2$, i.e., for the fusion-$wAC(P)$ connectors $\zeta_1$ and $\zeta_2$ it holds that $|\zeta_1| = |\zeta_2|$. We have to prove that $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha}$. Therefore, we have to prove that for any expression $E \in wAC(P \cup \{r\})$ it holds that $E(\zeta_1^{\alpha}/r) \equiv E(\zeta_2^{\alpha}/r)$. We assume that $r$ occurs only once in $E$. Otherwise, we apply the proof iteratively. By our assumption for $r$ it suffices to consider and prove the following equivalences:

1. $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha}$. For $\alpha = 0$, we have that $|\zeta_1| = |\zeta_1| = |\zeta_2| = |\zeta_2|$. On the other hand, for $\alpha = 1$, we have that $|\zeta_1'| = |\zeta_1| = |\zeta_2| = |\zeta_2'|$. Hence, $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha}$ for $\alpha \in \{0, 1\}$.

2. $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha} \equiv \zeta$ where $\zeta \in wAC(P)$. For $\alpha = 0$, we have that $|\zeta_1| \equiv |\zeta_2| \equiv |\zeta_1| \equiv |\zeta_2|$. On the other hand, for $\alpha = 1$, we have that $|\zeta_1'| \equiv |\zeta_2'| \equiv |\zeta_1| \equiv |\zeta_2|$. Hence, $|\zeta_1|^{\alpha} \equiv |\zeta_2|^{\alpha} \equiv \zeta$ for $\alpha \in \{0, 1\}$.

3. $|\zeta_1|^{\alpha} \equiv |\zeta_1|^{\alpha_1} \equiv \cdots \equiv |\zeta_n|^{\alpha_n} \equiv |\zeta_2|^{\alpha_1} \equiv \cdots \equiv |\zeta_n|^{\alpha_n}$ where $\zeta_1, \ldots, \zeta_n \in wAC(P)$ and $\alpha_1, \ldots, \alpha_n \in \{0, 1\}$.

   - For $\alpha = 0$, we consider the following cases:
     - **Case 1:** $\#_T(\zeta_1^{\alpha_1} \cdots \zeta_n^{\alpha_n}) = 0$. Then
       $$|\zeta_1| \equiv |\zeta_2| \equiv |\zeta_1| \equiv |\zeta_2|$$

       and
       $$|\zeta_2| \equiv |\zeta_1| \equiv |\zeta_2|$$
- **Case 2**: \( \#T([\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n}) > 0 \). Then

\[
| [\xi_1] \otimes [\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n} |
= \bigoplus_{i \in [n], \alpha_i = 1} (|\xi_i| \otimes \bigotimes_{k \neq i, \alpha_k \in \{0,1\}} (1 \oplus |\xi_k|) \otimes (1 \oplus |\xi_1|))
\]

and

\[
| [\xi_2] \otimes [\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n} |
= \bigoplus_{i \in [n], \alpha_i = 1} (|\xi_i| \otimes \bigotimes_{k \neq i, \alpha_k \in \{0,1\}} (1 \oplus |\xi_k|) \otimes (1 \oplus |\xi_2|))
\]

- **Case 1**: \( \#T([\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n}) = 0 \). Then

\[
| [\xi_1]' \otimes [\xi_1] \otimes \ldots \otimes [\xi_n] |
= |\xi_1| \otimes \bigotimes_{l \in [n]} (1 \oplus |\xi_l|)
\]

and

\[
| [\xi_2]' \otimes [\xi_1] \otimes \ldots \otimes [\xi_n] |
= |\xi_2| \otimes \bigotimes_{l \in [n]} (1 \oplus |\xi_l|).
\]

- **Case 2**: \( \#T([\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n}) > 0 \). Then

\[
| [\xi_1]' \otimes [\xi_1]^{\alpha_1} \otimes \ldots \otimes [\xi_n]^{\alpha_n} |
= \bigoplus_{i \in [n], \alpha_i = 1} (|\xi_1| \otimes \bigotimes_{k \neq i, \alpha_k \in \{0,1\}} (1 \oplus |\xi_k|) \otimes (1 \oplus |\xi_1|))
\]
Proof. We prove the above equality by induction on $n$. Let $\zeta = [\xi_1]_{\alpha_1} \otimes \ldots \otimes [\xi_n]_{\alpha_n}$, where $\alpha_1, \ldots, \alpha_n \in \{0, 1\}$. Then we get that $\zeta$ can be congruent if they are equivalent, their respective weighted fusion with the $wAC(P)$ connector $[1]'$ preserves their equivalence, and their degree is simultaneously zero or strictly positive. The second condition of the theorem actually serves as the least condition required in order to maintain the equivalence under the weighted fusion operator. Hence, as it is shown by the proof of the theorem, when equivalence is preserved under the weighted fusion for each one of the equivalent connectors with the trivial $wAC(P)$ connector $[1]'$, then it is preserved for its application with any other $wAC(P)$ connector. Regarding the other two conditions of the theorem, they are obviously necessary, since fusion-$wAC(P)$ connectors which are not equivalent cannot be congruent, and similarly for the case that the degree of only one of the two given fusion-$wAC(P)$ connectors is zero or strictly positive. For the proof of the theorem we need the following proposition.

**Proposition 5.** Let $\zeta = [\xi_1]_{\alpha_1} \otimes \ldots \otimes [\xi_n]_{\alpha_n}$, where $\alpha_i \in \{0, 1\}$ for $i \in [n]$. Then

\[
\left| [\zeta] \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_i|) \right| = \bigotimes_{i \in [n]} (1 \oplus |\zeta_i|).
\]

**Proof.** We prove the above equality by induction on $n$ and by Corollary 1.

**Theorem 2.** Let $\zeta_1, \zeta_2$ be fusion-$wAC(P)$ connectors. Then

\[
\zeta_1 \equiv \zeta_2 \Leftrightarrow \begin{cases} 
\zeta_1 \equiv \zeta_2 \\
\zeta_1 \otimes [1]' \equiv \zeta_2 \otimes [1]' \\
\#_r \zeta_1 > 0 \Leftrightarrow \#_r \zeta_2 > 0.
\end{cases}
\]

**Proof.** We prove the left-to-right implication. We assume that $\zeta_1 \equiv \zeta_2$. Hence, for every expression $E \in wAC(P \cup \{r\})$ it holds that $E(\zeta_1/r) \equiv E(\zeta_2/r)$. Let $E_1 = r \in wAC(P \cup \{r\})$. Then we get that $E_1(\zeta_1/r) \equiv E_1(\zeta_2/r)$, i.e., $\zeta_1 \equiv \zeta_2$. Furthermore, for $E_2 = r \otimes [1]' \in wAC(P \cup \{r\})$, it holds that

\[
\left| \zeta_1 \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_i|) \right| = \bigotimes_{i \in [n]} (1 \oplus |\zeta_i|).
\]
$E_2(\xi/r) \equiv E_2(\eta/r)$ and hence $\zeta \otimes [1]' \equiv \zeta_2 \otimes [1]'$. Now let $\xi = [\xi_1]^\alpha_1 \otimes \ldots \otimes [\xi_n]^\alpha_n \in wAC(P)$ for $
abla \alpha_1, \ldots, \alpha_n \in \{0, 1\}$. We claim that $\#_T \zeta_1 > 0 \Leftrightarrow \#_T \zeta_2 > 0$. On the contrary, we let $\#_T \zeta_1 > 0$ and $\#_T \zeta_2 = 0$. Specifically, let $\zeta_1 = [p]'$, $\zeta_2 = [p]$ and $\xi = [q]$. Then it holds that $|\zeta_1 \otimes \xi| = |[p]' \otimes [q]| = |p| \otimes (1 \oplus |q|) = p \otimes (1 \oplus q) = p \otimes (p \otimes q)$ and $|\zeta_2 \otimes \xi| = |[p] \otimes [q]| = |p| \otimes |q| = p \otimes q$. That is, for $E_3 = r \otimes \xi \in wAC(P \cup \{r\}$ it does not hold that $E(\zeta_1/r) \equiv E(\zeta_2/r)$, which is a contradiction, since $\zeta_1 \equiv \zeta_2$. Hence, $\#_T \zeta_1 > 0 \Leftrightarrow \#_T \zeta_2 > 0$.

Now we prove the right-to-left implication. Since $\zeta_1$ and $\zeta_2$ are fusion-$wAC(P)$ connectors, we let $\zeta_1 = [\zeta_1, 1]^\alpha_1 \otimes \ldots \otimes [\zeta_n, n]^\alpha_n$ and $\zeta_2 = [\zeta_1, 2]^\beta_1 \otimes \ldots \otimes [\zeta_m, m]^\beta_m$, where $\zeta_1, \alpha_1, \zeta_2, \beta_1 \in wAC(P)$ and $\alpha_1, \beta_1 \in \{0, 1\}$, for $i \in [n]$ and $j \in [m]$. We assume that (i) $\zeta_1 \equiv \zeta_2$, i.e., $|\zeta_1| = |\zeta_2|$, (ii) $\zeta_1 \otimes [1]' \equiv \zeta_2 \otimes [1]'$ which implies that $|\zeta_1 \otimes [1]'| = |\zeta_2 \otimes [1]'|$, and (iii) $\#_T \zeta_1 > 0 \Leftrightarrow \#_T \zeta_2 > 0$. We have to prove that $\zeta_1 \equiv \zeta_2$, thus we have to prove that for any expression $E \in wAC(P \cup \{r\}$, it holds that $E(\zeta_1/r) \equiv E(\zeta_2/r)$.

We assume that $r$ occurs only once in $E$, and otherwise we apply the proof iteratively. By this assumption it suffices to consider and prove the following:

1. $\zeta_1 \equiv \zeta_2$, which holds directly since $|\zeta_1| = |\zeta_2|$.
2. $\zeta_1 \oplus \zeta \equiv \zeta_2 \oplus \zeta$, where $\zeta \in wAC(P)$. The equivalence holds since $|\zeta_1 \oplus \zeta| = |\zeta_1| \oplus |\zeta| = |\zeta_2| \oplus |\zeta|$ and $|\zeta_2 \oplus \zeta| = |\zeta_2| \oplus |\zeta|$.
3. $\zeta \otimes [\xi, 1]^\delta_1 \otimes \ldots \otimes [\xi_r]^\delta_r \equiv \zeta_2 \otimes [\xi, 1]^\delta_1 \otimes \ldots \otimes [\xi_r]^\delta_r$, where $\xi_1, \ldots, \xi_r \in wAC(P)$ and $\delta_1, \ldots, \delta_r \in \{0, 1\}$. We consider the following cases:

- **Case 1**: ($\#_T \zeta_1 = 0, \#_T \zeta_2 = 0$)
  It holds that
  
  $$|\zeta_1| = |\zeta_2| \Rightarrow |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n]| = |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m]|$$
  $$\Rightarrow |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n]| = |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m]|.$$  
  \[\text{(Sigma}_1)\]

  and

  $$|\zeta_1 \otimes [1]'| = |\zeta_2 \otimes [1]'| \Rightarrow |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n] \otimes [1]'| = |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m] \otimes [1]'|$$
  $$\Rightarrow |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n] \otimes [1]'| = |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m] \otimes [1]'|$$
  $$\Rightarrow \bigotimes_{i \in [n]} (1 \oplus |\zeta_1, i|) = \bigotimes_{j \in [m]} (1 \oplus |\zeta_2, j|).$$  
  \[\text{(Sigma}_2)\]

  Then we consider the following cases:

  - If $\#_T([\xi, 1]^\delta_1 \otimes \ldots \otimes [\xi_r]^\delta_r) = 0$, then we have that
    $$|\zeta_1 \otimes \xi| = |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n] \otimes [\xi_1] \otimes \ldots \otimes [\xi_r]|$$
    $$= |[\zeta_1, 1] \otimes \ldots \otimes [\zeta_1, n] \otimes [\xi_1] \otimes \ldots \otimes [\xi_r]|$$
    \[\text{(Sigma}_1)\]
    $$\Rightarrow |\zeta_2, 1| \otimes \ldots \otimes [\zeta_2, m] \otimes [\zeta, 1] \otimes \ldots \otimes [\zeta, r]|$$
    $$\Rightarrow |\zeta_2 \otimes \xi| = |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m] \otimes [\xi_1] \otimes \ldots \otimes [\xi_r]|$$
    $$= |[\zeta_2, 1] \otimes \ldots \otimes [\zeta_2, m] \otimes [\xi_1] \otimes \ldots \otimes [\xi_r]|,$$
If \( \#_T([\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r}) > 0 \), then we have that
\[
|\zeta_1 \otimes \xi| = |\zeta_{1,1} \otimes \ldots \otimes [\zeta_{1,n}] \otimes [\zeta_1]^{\delta_1} \otimes \ldots \otimes [\zeta_r]^{\delta_r}|
\]
\[
= \bigoplus_{\lambda \in [r], \delta_\lambda = 1} \left( |\xi_\lambda| \otimes \bigotimes_{\mu \neq \lambda, \delta_\mu \in \{0,1\}} (1 \oplus |\xi_\mu|) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{1,i}|) \right)
\]
\[
= \left( \bigoplus_{\lambda \in [r], \delta_\lambda = 1} \left( |\xi_\lambda| \otimes \bigotimes_{\mu \neq \lambda, \delta_\mu \in \{0,1\}} (1 \oplus |\xi_\mu|) \right) \right) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{1,i}|)
\]
\[
= \left( \bigoplus_{\lambda \in [r], \delta_\lambda = 1} \left( |\xi_\lambda| \otimes \bigotimes_{\mu \neq \lambda, \delta_\mu \in \{0,1\}} (1 \oplus |\xi_\mu|) \right) \right) \otimes \bigotimes_{j \in [m]} (1 \oplus |\zeta_{2,j}|).
\]

Hence, we obtain that \( \zeta_1 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r} \equiv \zeta_2 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r} \) for Case 1.

Case 2: \((\#_T\zeta_1 > 0, \#_T\zeta_2 > 0)\)
It holds that
\[
|\zeta_1| = |\zeta_2| \Rightarrow |\zeta_{1,1}^{\alpha_{1,1}} \otimes \ldots \otimes [\zeta_{1,n}]^{\alpha_{1,n}}| = |\zeta_{2,1}^{\delta_1} \otimes \ldots \otimes [\zeta_{2,m}]^{\delta_m}|
\]
\[
\Rightarrow \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\zeta_{1,i_1}| \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1\}} (1 \oplus |\zeta_{1,k_1}|) \right)
\]
\[
= \bigoplus_{i_2 \in [m], \beta_{i_2} = 1} \left( |\zeta_{2,i_2}| \otimes \bigotimes_{k_2 \neq i_2, \alpha_{k_2} \in \{0,1\}} (1 \oplus |\zeta_{2,k_2}|) \right).
\]

and
\[
|\zeta_1 \otimes [1]'| = |\zeta_2 \otimes [1]'|
\]
\[
\Rightarrow |\zeta_{1,1}^{\alpha_{1,1}} \otimes \ldots \otimes [\zeta_{1,n}]^{\alpha_{1,n}} \otimes [1]'| = |\zeta_{2,1}^{\delta_1} \otimes \ldots \otimes [\zeta_{2,m}]^{\delta_m} \otimes [1]'|
\]
\[
\Rightarrow \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\zeta_{1,i_1}| \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1\}} (1 \oplus |\zeta_{1,k_1}|) \otimes (1 \oplus [1]) \right) \oplus |1| \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{1,i}|)
\]

38
where the last step holds by Proposition 5. Then we consider the following cases:

- If \( \#(\xi_1^{\delta_1} \otimes \ldots \otimes \xi_r^{\delta_r}) = 0 \), then we have that
  \[
  |\xi_1 \otimes \xi| = |\xi_1,1|^{\alpha_1} \otimes \ldots \otimes |\xi_1,n|^{\alpha_n} \otimes |\xi_1 \otimes \ldots \otimes |\xi_r| |
  \]
  \[
  = \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\xi_{1,i_1} | \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1,\}} (1 + |\xi_{1,k_1} |) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right)
  \]
  \[
  = \left( \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\xi_{1,i_1} | \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1,\}} (1 + |\xi_{1,k_1} |) \right) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right)
  \]
  \[
  (\Sigma_3) \left( \bigoplus_{i_2 \in [m], \beta_{i_2} = 1} \left( |\xi_{2,i_2} | \otimes \bigotimes_{j \in [m]} (1 + |\xi_{2,j} |) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right) \right)
  \]
  and
  \[
  |\xi_2 \otimes \xi| = |\xi_{2,1} |^{\beta_1} \otimes \ldots \otimes |\xi_{2,m} |^{\beta_m} \otimes |\xi_1 \otimes \ldots \otimes |\xi_r| |
  \]
  \[
  = \bigoplus_{i_2 \in [m], \beta_{i_2} = 1} \left( |\xi_{2,i_2} | \otimes \bigotimes_{j \in [m]} (1 + |\xi_{2,j} |) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right)
  \]
  \[
  = \left( \bigoplus_{i_2 \in [m], \beta_{i_2} = 1} \left( |\xi_{2,i_2} | \otimes \bigotimes_{j \in [m]} (1 + |\xi_{2,j} |) \right) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right).
  \]

- If \( \#(\xi_1^{\delta_1} \otimes \ldots \otimes \xi_r^{\delta_r}) > 0 \), then we have that
  \[
  |\xi_1 \otimes \xi| = |\xi_1,1|^{\alpha_1} \otimes \ldots \otimes |\xi_1,n|^{\alpha_n} \otimes |\xi_1 \otimes \ldots \otimes |\xi_r| |
  \]
  \[
  = \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\xi_{1,i_1} | \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1,\}} (1 + |\xi_{1,k_1} |) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right)
  \]
  \[
  = \left( \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1} \left( |\xi_{1,i_1} | \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1,\}} (1 + |\xi_{1,k_1} |) \right) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right)
  \]
  \[
  (\Sigma_3) \left( \bigoplus_{i_2 \in [m], \beta_{i_2} = 1} \left( |\xi_{2,i_2} | \otimes \bigotimes_{j \in [m]} (1 + |\xi_{2,j} |) \bigotimes_{l \in [r]} (1 + |\xi_l |) \right) \right).
  \]

39
Next proposition is an application of Theorem 2 for proving congruence relation of fusion-$wAC(P)$ connectors.

Finally, for any other form of the expression $E$ we apply the presented cases iteratively, and our proof is completed.

4. The symmetric case of 2 and the cases for any other position of $\zeta_1, \zeta_2$, in 3. For these cases the proof is analogous.

\[\begin{align*}
\bigoplus_{\lambda \in [r], \delta_{\lambda} = 1} \left( |\xi_{\lambda}| \otimes \bigotimes_{\mu \neq \lambda, \delta_{\mu} \in \{0,1\}} (1 \oplus |\xi_{\mu}|) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{1,i}|) \right) \\
= \biggl( \bigoplus_{i_1 \in [n], \alpha_{i_1} = 1, \zeta_{1,i_1} \neq 1} \left( |\zeta_{1,i_1}| \otimes \bigotimes_{k_1 \neq i_1, \alpha_{k_1} \in \{0,1\}} (1 \oplus |\zeta_{1,k_1}|) \right) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{i}|) \biggr) \\
\quad \bigg( \bigoplus_{\lambda \in [r], \delta_{\lambda} = 1} \left( |\xi_{\lambda}| \otimes \bigotimes_{\mu \neq \lambda, \delta_{\mu} \in \{0,1\}} (1 \oplus |\xi_{\mu}|) \right) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{1,i}|) \biggr) \\
\quad \bigg( \bigoplus_{i_2 \in [m], \beta_{i_2} = 1, \zeta_{2,i_2} \neq 1} \left( |\zeta_{2,i_2}| \otimes \bigotimes_{k_2 \neq i_2, \beta_{k_2} \in \{0,1\}} (1 \oplus |\zeta_{2,k_2}|) \right) \otimes \bigotimes_{i \in [n]} (1 \oplus |\zeta_{i}|) \biggr) \\
\quad \bigg( \bigoplus_{\lambda \in [r], \delta_{\lambda} = 1} \left( |\xi_{\lambda}| \otimes \bigotimes_{\mu \neq \lambda, \delta_{\mu} \in \{0,1\}} (1 \oplus |\xi_{\mu}|) \right) \otimes \bigotimes_{j \in [m]} (1 \oplus |\zeta_{2,j}|) \biggr) \\
\end{align*}\]

Hence, we get that $\zeta_1 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r} \equiv \zeta_2 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r}$ for Case 2. Therefore, in any case it holds that $\zeta_1 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r} \equiv \zeta_2 \otimes [\xi_1]^{\delta_1} \otimes \ldots \otimes [\xi_r]^{\delta_r}$.
Proposition 6. Let $\zeta, \zeta_1, \zeta_2, \zeta_3$ be fusion-\textit{wAC}(P) connectors where $\#\tau\zeta > 0$. Then we have

i) $\zeta \otimes [0]' \cong \zeta$

ii) $[\zeta_1]' \otimes [\zeta_2] \otimes [\zeta_3] \cong [\zeta_1]' \otimes ([\zeta_2]' \otimes [\zeta_3]')$

iii) $[\zeta_1]' \otimes [\zeta_2]' \cong ([\zeta_1]' \otimes [\zeta_2]')'$.

Proof. We apply Theorem 2 and Corollary 1, and we are done. \qed

Discussion

In [BS08a], the authors proved the soundness of their algebras and investigated the conditions under which completeness also holds. Proving such results in the weighted setting, is in general, much harder. In particular, according to [PR22], soundness has been only defined for multi-valued logics, with values in the bounded distributive lattice $[0,1]$ with the usual max and min operations (cf. [Haj98]). In turn, in [PR22], the authors introduced a notion of soundness in the context of weighted propositional configuration logic formulas, with weights ranging over a commutative semiring. The formulas of that logic served for encoding the quantitative features of architectures styles.

Following the work of [PR22], we could provide an analogous definition of soundness for our weighted algebras. In this case, it occurs that proving soundness would require semiring $K$ to be idempotent with respect to its first and second operation. Idempotency for the second operation of $K$ is required by the weighted synchronization and fusion operators in $\textit{wAI}(P)$ and $\textit{wAC}(P)$, respectively. However, in this paper, $K$ is idempotent only with respect to its first operation. A further investigation of soundness for our algebras along with the consideration of other algebraic structures is left as future work.

On the other hand, the notion of completeness does not comply in general, in the weighted setup. Indeed, due to the presence of weights we cannot ensure that two arbitrary constructs with the same weight have also the same syntax. In our setting, let for instance $z_1, z_2 \in \textit{wAI}(P)$. Then $z_1$ and $z_2$ can return the same weight, while they encode different coordination schemes.

8 Conclusion

In this paper, we developed an algebraic framework for the formal characterization of the quantitative aspects of connectors in architectures of component-based systems. In particular, we firstly studied a weighted Algebra of Interactions, $\textit{wAI}(P)$, over a set of ports $P$ and a commutative and idempotent semiring $K$. Then we interpreted $\textit{wAI}(P)$ by polynomials in $K(\Gamma(P))$ and using the equivalence classes of the algebra we proved several properties for the $\textit{wAI}(P)$ elements. Specifically, we showed that the structure $(\textit{wAI}(P)/ \equiv, \oplus, \otimes, 0, 1)$ is a commutative and idempotent semiring, an important result that was used for the computation of the semantics of the $\textit{wAC}(P)$ connectors. In turn, we applied the $\textit{wAI}(P)$ algebra for encoding the weight of well-known coordination schemes.

In the sequel, we studied the weighted Algebra of Connectors over $P$ and $K$, $\textit{wAC}(P)$, that extended $\textit{wAI}(P)$ with two typing operators, namely triggers "$[\cdot]'$ that initiate an interaction and synchrons "$[\cdot]''$ that need synchronization with other ports in order to interact. We expressed the semantics of $\textit{wAC}(P)$ connectors as $\textit{wAI}(P)$ elements, and then, applying the semantics of
$wAI(P)$, we obtained the corresponding weight of the connectors over a concrete interactions set over $P$. We proved several nice properties for $wAC(P)$ and we showed the expressiveness of our algebra by modeling several connectors in the weighted setup. Moreover, we studied two subalgebras of $wAC(P)$, the weighted Algebra of Synchrons $wAS(P)$ and the weighted Algebra of Triggers $wAT(P)$, over $P$ and $K$, where the former restricted to synchron elements and the latter to trigger elements. Finally, we defined a concept of congruence relation for fusion-$wAC(P)$ connectors and we proved two theorems for checking such a congruence.

There are several directions for future work. An important open problem is providing a congruence relation for $wAC(P)$ connectors in general, as well as investigating a different modeling methodology for the weighted framework of connectors in order to solve their congruence problem. Future work is also studying our weighted algebras over alternative structures than $K$, in order to prove their soundness. Another work direction includes investigating the concept of $wAC(P)$ connectors over more general structures than semirings, that are used in practical applications, for instance valuation monoids (cf. [DM11, KP20]).

Moreover, in [BS08b], the authors studied the concept of glue operators as composition operators, in order to formalize the coordinated behavior in component-based systems, while in [BS11], they alternatively expressed glue operators as boolean constraints between interactions and the state of the coordinated components. Therefore, future research includes developing a framework for modeling the coordination and behavior of component-based systems in the weighted setting [BLM06, BMM11, NS19, SZ18]. On the other hand, in several approaches, connectors have been modeled as entities whose interactions may be modified during the operation of the system [BCD00, Pah01]. In other words, it would be interesting to extend our results for connectors with dynamic interactions. In addition to these theoretical directions, ongoing work includes an implementation of the presented formal framework.
References

[AG94] R. J. Allen and D. Garlan. Formalizing architectural connection. In B. Fadini, L. J. Osterweil, and A. van Lamsweerde, editors, *Proceedings of the ICSE*, pages 71–80. IEEE Computer Society / ACM Press, 1994.

[APR06] S. Amaro, E. Pimentel, and A.M. Roldán. Reo based interaction model. *Electron. Notes Theor. Comput. Sci.*, 160:3–14, 2006.

[Arb04] F. Arbab. Reo: a channel-based coordination model for component composition. *Math. Struct. Comput. Sci.*, 14(3):329–366, 2004.

[BCD00] M. Bernardo, P. Ciancarini, and L. Donatiello. On the formalization of architectural types with process algebras. In J. C. Knight and D. S. Rosenblum, editors, *Proceedings of SIGSOFT FSE*, pages 140–148. ACM, 2000.

[BLM06] R. Bruni, I. Lanese, and U. Montanari. A basic algebra of stateless connectors. *Theor. Comput. Sci.*, 366(1-2):98–120, 2006.

[BMM11] R. Bruni, H. C. Melgratti, and U. Montanari. A survey on basic connectors and buffers. In B. Beckert, F. Damiani, F. S. de Boer, and M. M. Bonsangue, editors, *Proceedings of FMCO, Revised Selected Papers*, volume 7542 of LNCS, pages 49–68. Springer, 2011.

[BS08a] S. Bliudze and J. Sifakis. The algebra of connectors - structuring interaction in BIP. *IEEE Trans. Computers*, 57(10):1315–1330, 2008.

[BS08b] S. Bliudze and J. Sifakis. A notion of glue expressiveness for component-based systems. In F. van Breugel and M. Chechik, editors, *Proceedings of CONCUR*, volume 5201, pages 508–522. Springer, 2008.

[BS11] S. Bliudze and J. Sifakis. Synthesizing glue operators from glue constraints for the construction of component-based systems. In S. Apel and E. K. Jackson, editors, *Proceedings of SC@TOOLS*, LNCS, pages 51–67. Springer, 2011.

[DKV09] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Springer-Verlag, Berlin-Heidelberg, 2009.

[DM11] M. Droste and I. Meinecke. Weighted automata and regular expressions over valuation monoids. *Int. J. Found. Comput. Sci.*, 22(8):1829–1844, 2011.

[GS03] G. Gössler and J. Sifakis. Component-based construction of deadlock-free systems: Extended abstract. In P. K. Pandya and J. Radhakrishnan, editors, *Proceedings of FSTTCS*, volume 2914 of LNCS, pages 420–433. Springer, 2003.

[GS05] G. Gössler and J. Sifakis. Composition for component-based modeling. *Sci. Comput. Program.*, 55(1-3):161–183, 2005.

[Háj98] P. Hájek, editor. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, 1998.

[KP20] V. Karyoti and P. Paraponiari. Weighted PCL over product valuation monoids. In S. Bliudze and L. Becchi, editors, *Proceedings of COORDINATION*, volume 12134 of LNCS, pages 301–319. Springer, 2020.

43
[MBBS17] A. Mavridou, E. Baranov, S. Bliudze, and J. Sifakis. Configuration logics: Modeling architecture styles. *J. Log. Algebr. Methods Program.*, 86(1):2–29, 2017.

[NS19] M. S. Nawaz and M. Sun. Using PVS for modeling and verification of probabilistic connectors. In H. Hojjat and M. Massink, editors, *Proceedings of FSEN, Revised Selected Papers*, volume 11761 of *LNCS*, pages 61–76. Springer, 2019.

[Ozk18] M. Ozkaya. Architectural languages’ connector support for modeling various component interactions: A review. In H. Fujita and E. Herrera-Viedma, editors, *Proceedings of SoMeT*, volume 303 of *Frontiers in Artificial Intelligence and Applications*, pages 474–489. IOS Press, 2018.

[Pah01] C. Pahl. Components, contracts, and connectors for the unified modelling language UML. In J. N. Oliveira and P. Zave, editors, *Proceedings of FME*, volume 2021 of *LNCS*, pages 259–277. Springer, 2001.

[PR22] P. Paraponiari and G. Rahonis. Weighted propositional configuration logics: A specification language for architectures with quantitative features. *Information and Computation*, 282, 2022.

[RHJ18] Q. Rouland, B. Hamid, and J. Jaskolka. Formalizing reusable communication models for distributed systems architecture. In E. H. Abdelwahed, L. Bellatreche, M. Golfarelli, D. Mery, and C. Ordonez, editors, *Proceedings of MEDI*, volume 11163 of *LNCS*, pages 198–216. Springer, 2018.

[Sak] J. Sakarovitch. *Rational and recognisable power series*, chapter in [DKV09].

[SG03] B. Spitznagel and D. Garlan. A compositional formalization of connector wrappers. In L. A. Clarke, L. Dillon, and W. F. Tichy, editors, *Proceedings of ICSE*, volume 11533, pages 374–384. IEEE Computer Society, 2003.

[Sif13] J. Sifakis. Rigorous systems design. *Found. Trends Signal Process.*, 6(4):293–362, 2013.

[SZ18] M. Sun and X. Zhang. A relational model for probabilistic connectors based on timed data distribution streams. In D. N. Jansen and P. Prabhakar, editors, *Proceedings of FORMATS*, volume 11022 of *LNCS*, pages 125–141. Springer, 2018.
9 Appendix

9.1 Weighted Rendezvous

Next we present the tables that we used for computing the weight of the $wAI(P)$ element $z = s \otimes r_1 \otimes r_2$ on a concrete interactions set. Specifically, for $\gamma = \{\{s, r_1, r_2\}\} \in \Gamma(P)$ we obtained the primary Table 1, while for $\gamma = \{\{s, r_1, r_2\}, \{s, r_2\}\} \in \Gamma(P)$ we also used the primary Table 2. Moreover, we used the primary Table 2 for computing the weight of $z = s \otimes r_1 \otimes r_2$ on $\gamma = \{\{s, r_2\}\}$. Finally, in all of the three cases we used the auxiliary Tables 3-10.

| $a = a_1 \cup a_2$ | $||a|| \langle \{a_1\} \rangle$ | $||r_1 \otimes r_2|| \langle \{a_2\} \rangle$ | $s \otimes r_1 \otimes r_2$ | $s \otimes r_1 \otimes r_2$ |
|----------------------|---------------------------------|---------------------------------|----------------------|----------------------|
| $a_1 = \emptyset, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \emptyset$ | $k_s$ | 0 | 0 | 0 |
| $a_1 = \{s\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_1, r_2\}, a_2 = \{s\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1\}, a_2 = \{r_2\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_2\}, a_2 = \{r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1\}, a_2 = \{s, r_2\}$ | 0 | 0 | 0 |
| $a_1 = \{s\}, a_2 = \{s, r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1\}, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_2\}, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_2\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{s, r_1\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1\}, a_2 = \{s, r_1, r_2\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_2\}, a_2 = \{s, r_1, r_2\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 | 0 |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1, r_2\}$ | 0 | $k_{r_1} \cdot k_{r_2}$ | 0 |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | $k_s$ | $k_{r_1} \cdot k_{r_2}$ | $k_s \cdot k_{r_1} \cdot k_{r_2}$ |
| $a_1 = \{r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 | 0 |

Table 1: Weighted Rendezvous and $a = \{s, r_1, r_2\}$. 

45
Table 2: Weighted Rendezvous and $a = \{s, r_2\}$.

| $a = a_1 \cup a_2$ | $\|s\| (\{a_1\})$ | $\|r_1 \otimes r_2\| (\{a_2\})$ | $\cdot a_1$ |
|---------------------|---------------------|---------------------|---------|
| $a_1 = \emptyset$, $a_2 = \{s, r_2\}$ | 0                  | 0                    | $\emptyset$ |
| $a_1 = \{s, r_2\}$, $a_2 = \emptyset$ | $k_s$              | 0                    | 0        |
| $a_1 = \emptyset$, $a_2 = \{r_2\}$ | 0                  | 0                    | 0        |
| $a_1 = \{s, r_2\}$, $a_2 = \{r_2\}$ | $k_s$              | 0                    | 0        |
| $a_1 = \emptyset$, $a_2 = \{s, r_2\}$ | $k_s$              | 0                    | 0        |
| $a_1 = \{s, r_2\}$, $a_2 = \{s, r_2\}$ | 0                  | 0                    | 0        |
| $a_1 = \emptyset$, $a_2 = \{s, r_2\}$ | $k_s$              | 0                    | 0        |
| $a_1 = \emptyset$, $a_2 = \emptyset$ | 0                  | 0                    | 0        |

Table 3: $r_1 \otimes r_2$ and $a_2 = \emptyset$.

| $a_2 = a_2,1 \cup a_2,2$ | $\|r_1\| (\{a_2\})$ | $\|r_2\| (\{a_2\})$ | $\cdot a_2,1$ |
|--------------------------|---------------------|---------------------|---------|
| $a_2,1 = \emptyset$, $a_2,2 = \emptyset$ | 0                  | 0                    | 0        |
| $a_2,1 = \emptyset$, $a_2,2 = \emptyset$ | 0                  | 0                    | 0        |
| $a_2,1 = \emptyset$, $a_2,2 = \emptyset$ | 0                  | 0                    | 0        |
| $a_2,1 = \emptyset$, $a_2,2 = \emptyset$ | 0                  | 0                    | 0        |

Table 4: $r_1 \otimes r_2$ and $a_2 = \{s\}$.

| $a_2 = a_2,1 \cup a_2,2$ | $\|r_1\| (\{a_2\})$ | $\|r_2\| (\{a_2\})$ | $\cdot a_2,1$ |
|--------------------------|---------------------|---------------------|---------|
| $a_2,1 = \emptyset$, $a_2,2 = \{r_1\}$ | 0                  | 0                    | 0        |
| $a_2,1 = \{r_1\}$, $a_2,2 = \emptyset$ | $k_{r_1}$          | 0                    | 0        |
| $a_2,1 = \{r_1\}$, $a_2,2 = \{r_1\}$ | $k_{r_1}$          | 0                    | 0        |

Table 5: $r_1 \otimes r_2$ and $a_2 = \{r_1\}$.

46
\[
\begin{array}{|c|c|c|}
\hline
a_2 &= a_{2,1} \cup a_{2,2} & ||r_1||(||\{a_{2,1}\}||) \quad ||r_2||(||\{a_{2,2}\}||) \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
a_{2,1} = \{r_2\}, a_{2,2} = \emptyset & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{r_2\}, a_{2,2} = \{r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
\end{array}
\]

Table 6: \( r_1 \otimes r_2 \) and \( a_2 = \{r_2\} \).

\[
\begin{array}{|c|c|c|}
\hline
a_2 &= a_{2,1} \cup a_{2,2} & ||r_1||(||\{a_{2,1}\}||) \quad ||r_2||(||\{a_{2,2}\}||) \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s, r_1\} & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{s, r_1\}, a_{2,2} = \emptyset & k_{r_1} & 0 \quad 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \{r_1\} & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{r_1\}, a_{2,2} = \{s\} & k_{r_1} & 0 \quad 0 \\
\hline
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1\} & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s\} & k_{r_1} & 0 \quad 0 \\
\hline
a_{2,1} = \{r_1\}, a_{2,2} = \{s, r_1\} & k_{r_1} & 0 \quad 0 \\
\hline
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1\} & k_{r_1} & 0 \quad 0 \\
\hline
\end{array}
\]

Table 7: \( r_1 \otimes r_2 \) and \( a_2 = \{s, r_1\} \).

\[
\begin{array}{|c|c|c|}
\hline
a_2 &= a_{2,1} \cup a_{2,2} & ||r_1||(||\{a_{2,1}\}||) \quad ||r_2||(||\{a_{2,2}\}||) \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s, r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
a_{2,1} = \{s, r_2\}, a_{2,2} = \emptyset & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \{r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
a_{2,1} = \{r_2\}, a_{2,2} = \{s\} & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \{s, r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
a_{2,1} = \{s, r_2\}, a_{2,2} = \{s\} & 0 & 0 \quad 0 \\
\hline
a_{2,1} = \{r_2\}, a_{2,2} = \{s, r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
a_{2,1} = \{s, r_2\}, a_{2,2} = \{s, r_2\} & 0 & k_{r_2} \quad 0 \\
\hline
\hline
\end{array}
\]

Table 8: \( r_1 \otimes r_2 \) and \( a_2 = \{s, r_2\} \).
\[ \|r_1 \otimes r_2\| (\{a_2\}) \]

| \(a_2 = a_{2,1} \cup a_{2,2}\) | \(\|r_1\| (\{a_{2,1}\})\) | \(\|r_2\| (\{a_{2,2}\})\) | \(\cdot\) |
|-----------------|-----------------|-----------------|-----------------|
| \(a_{2,1} = \emptyset, a_{2,2} = \{r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{r_1, r_2\}, a_{2,2} = \emptyset\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{r_1\}, a_{2,2} = \{r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{r_2\}, a_{2,2} = \{r_1\}\) | 0 | 0 | 0 |
| \(a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1, r_2\}\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |

Table 9: \(r_1 \otimes r_2\) and \(a_2 = \{r_1, r_2\}\).

\[ \|r_1 \otimes r_2\| (\{a_2\}) \]

| \(a_2 = a_{2,1} \cup a_{2,2}\) | \(\|r_1\| (\{a_{2,1}\})\) | \(\|r_2\| (\{a_{2,2}\})\) | \(\cdot\) |
|-----------------|-----------------|-----------------|-----------------|
| \(a_{2,1} = \emptyset, a_{2,2} = \{s, r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \emptyset\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{s\}, a_{2,2} = \{r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s\}\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{s\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1, r_2\}\) | 0 | 0 | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | 0 | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1, r_2\}\) | 0 | \(k_{r_2}\) | 0 |
| \(a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |
| \(a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\}\) | \(k_{r_1}\) | \(k_{r_2}\) | \(k_{r_1 \cdot k_{r_2}}\) |

Table 10: \(r_1 \otimes r_2\) and \(a_2 = \{s, r_1, r_2\}\).
9.2 Weighted Broadcast

The following tables were used for computing the weight of the $wAlI(P)$ element $z = s \otimes (1 \oplus r_1) \otimes (1 \oplus r_2)$ for the Broadcast scheme on the interactions set $\gamma = \{\{s\}, \{s, r_1\}, \{s, r_2\}, \{s, r_1, r_2\}\} \in \Gamma(P)$. The primary tables are Tables 11-14, while Tables 15-22 are the auxiliary ones.

| $a = a_1 \cup a_2$ | $\|s \otimes (1 \oplus r_1) \otimes (1 \oplus r_2)\| \{a_1\}$ | $\| (1 \oplus r_1) \otimes (1 \oplus r_2) \| \{a_2\}$ | $+$ |
|-------------------|-----------------|-----------------|---|
| $a_1 = 0, \ a_2 = \{s\}$ | 0               | 0               | 0 |
| $a_1 = \{s\}, \ a_2 = \emptyset$ | $k_s$           | 1               | $k_s$ |
| $a_1 = \{s\}, \ a_2 = \{s\}$ | $k_s$           | 0               | 0 |
| $a_1 = \emptyset, \ a_2 = \{s\}$ | 0               | $k_{r_1}$       | 0 |

Table 11: Weighted Broadcast and $a = \{s\}$.

| $a = a_1 \cup a_2$ | $\|s \otimes (1 \oplus r_1) \otimes (1 \oplus r_2)\| \{a_1\}$ | $\| (1 \oplus r_1) \otimes (1 \oplus r_2) \| \{a_2\}$ | $+$ |
|-------------------|-----------------|-----------------|---|
| $a_1 = \emptyset, \ a_2 = \{s, r_1\}$ | 0               | $k_{r_1}$       | 0 |
| $a_1 = \{s, r_1\}, \ a_2 = \emptyset$ | $k_s$           | 1               | $k_s$ |
| $a_1 = \{s\}, \ a_2 = \{r_1\}$ | $k_s$           | $k_{r_1}$       | $k_s \cdot k_{r_1}$ |
| $a_1 = \{r_1\}, \ a_2 = \{s\}$ | 0               | 0               | 0 |
| $a_1 = \{s\}, \ a_2 = \{s, r_1\}$ | $k_s$           | $k_{r_1}$       | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1\}, \ a_2 = \{s\}$ | $k_s$           | 0               | 0 |
| $a_1 = \{r_1\}, \ a_2 = \{s, r_1\}$ | 0               | $k_{r_1}$       | 0 |
| $a_1 = \{s, r_1\}, \ a_2 = \{r_1\}$ | $k_s$           | $k_{r_1}$       | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1\}, \ a_2 = \{s, r_1\}$ | $k_s$           | $k_{r_1}$       | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1\}, \ a_2 = \{s\}$ | $k_s$           | 0               | 0 |
| $a_1 = \{r_1\}, \ a_2 = \{s, r_1\}$ | 0               | $k_{r_1}$       | 0 |
| $a_1 = \{s, r_2\}, \ a_2 = \{r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{r_2\}, \ a_2 = \{s, r_2\}$ | 0               | $k_{r_2}$       | 0 |
| $a_1 = \{s, r_2\}, \ a_2 = \{r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_2\}, \ a_2 = \{s, r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{r_2\}, \ a_2 = \{s, r_2\}$ | 0               | $k_{r_2}$       | 0 |

Table 12: Weighted Broadcast and $a = \{s, r_1\}$.

| $a = a_1 \cup a_2$ | $\|s \otimes (1 \oplus r_1) \otimes (1 \oplus r_2)\| \{a_1\}$ | $\| (1 \oplus r_1) \otimes (1 \oplus r_2) \| \{a_2\}$ | $+$ |
|-------------------|-----------------|-----------------|---|
| $a_1 = \emptyset, \ a_2 = \{s, r_2\}$ | 0               | $k_{r_2}$       | 0 |
| $a_1 = \{s, r_2\}, \ a_2 = \emptyset$ | $k_s$           | 1               | $k_s$ |
| $a_1 = \{s\}, \ a_2 = \{r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{r_2\}, \ a_2 = \{s\}$ | 0               | 0               | 0 |
| $a_1 = \{s\}, \ a_2 = \{s, r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_2\}, \ a_2 = \{s\}$ | $k_s$           | 0               | 0 |
| $a_1 = \{r_2\}, \ a_2 = \{s, r_2\}$ | 0               | $k_{r_2}$       | 0 |
| $a_1 = \{s, r_2\}, \ a_2 = \{r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_2\}, \ a_2 = \{s, r_2\}$ | $k_s$           | $k_{r_2}$       | $k_s \cdot k_{r_2}$ |
| $a_1 = \{r_2\}, \ a_2 = \{s, r_2\}$ | 0               | $k_{r_2}$       | 0 |

Table 13: Weighted Broadcast and $a = \{s, r_2\}$. 

49
| $a = a_1 \cup a_2$ | $\|s\|\{(a_1)\}$ | $\|(1 \oplus r_1) \otimes (1 \oplus r_2)\|\{(a_2)\}$ | $\|a\|$ |
|-------------------|-----------------|-----------------|-----|
| $a_1 = \emptyset$, $a_2 = \{s, r_1, r_2\}$ | 0 | $k_s + k_r + (k_r \cdot k_s)$ | 0 |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | 1 | $k_s$ |
| $a_1 = \{s\}$, $a_2 = \{r_1, r_2\}$ | $k_s$ | $k_r + k_s + (k_r \cdot k_s)$ | $(k_s \cdot k_r) + (k_s \cdot k_r) + (k_r \cdot k_r \cdot k_s)$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1\}$, $a_2 = \{r_2\}$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{r_2\}$, $a_2 = \{s, r_1\}$ | 0 | $k_{r_1}$ | 0 |
| $a_1 = \{s, r_2\}$, $a_2 = \{r_1\}$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{r_1\}$, $a_2 = \{s, r_2\}$ | 0 | $k_{r_2}$ | 0 |
| $a_1 = \{s\}$, $a_2 = \{s, r_1, r_2\}$ | $k_s$ | $k_r + k_s + (k_r \cdot k_s)$ | $(k_s \cdot k_r) + (k_s \cdot k_r) + (k_r \cdot k_s \cdot k_r)$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s\}$ | 0 | 0 | 0 |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{r_1\}$ | $k_s$ | $k_r + k_s + (k_r \cdot k_s)$ | $(k_s \cdot k_r) + (k_s \cdot k_r) + (k_r \cdot k_s)$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{r_2\}$ | $k_s$ | $k_r + k_s + (k_r \cdot k_s)$ | $(k_s \cdot k_r) + (k_s \cdot k_r) + (k_r \cdot k_s)$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_1\}$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_2\}$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_1, r_2\}$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_1\}$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_2\}$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \{s, r_1, r_2\}$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_1}$ | $k_s \cdot k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_r$ | 0 | $k_r$ |
| $a_1 = \{s, r_1, r_2\}$, $a_2 = \emptyset$ | $k_s$ | $k_{r_2}$ | $k_s \cdot k_{r_2}$ |

Table 14: Weighted Broadcast and $a = \{s, r_1, r_2\}$.

| $a_2 = a_{2,1} \cup a_{2,2}$ | $\|1 \oplus r_1\|\{(a_{2,1})\}$ | $\|1 \oplus r_2\|\{(a_{2,2})\}$ | $\|a_{2}\|$ |
|-------------------|-----------------|-----------------|-----|
| $a_{2,1} = \emptyset$, $a_{2,2} = \emptyset$ | 0 | 0 | 0 |
| $a_{2,1} = \emptyset$, $a_{2,2} = \emptyset$ | $1 + 0$ | $1 + 0$ | 1 |

Table 15: $(1 \oplus r_1) \otimes (1 \oplus r_2)$ and $a_2 = \emptyset$.  

50
\[
\begin{array}{c|ccc}
| a_2 = a_{2,1} \cup a_{2,2} | & \|1 \oplus r_1\|(|a_{2,1}|) & \|1 \oplus r_2\|(|a_{2,2}|) |
| \hline
| a_{2,1} = \emptyset, a_{2,2} = \{s\} | & 1 + 0 & 0 + 0 & 0 \\
| a_{2,1} = \{s\}, a_{2,2} = \emptyset | & 0 + 0 & 1 + 0 & 0 \\
| a_{2,1} = \{s\}, a_{2,2} = \{s\} | & 0 + 0 & 0 + 0 & 0 \\
+ & & & 0 \\
\end{array}
\]

Table 16: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{s\}\).

\[
\begin{array}{c|ccc}
| a_2 = a_{2,1} \cup a_{2,2} | & \|1 \oplus r_1\|(|a_{2,1}|) & \|1 \oplus r_2\|(|a_{2,2}|) |
| \hline
| a_{2,1} = \emptyset, a_{2,2} = \{r_1\} | & 1 + 0 & 0 + 0 & 0 \\
| a_{2,1} = \{r_1\}, a_{2,2} = \emptyset | & 0 + k_{r_1} & 1 + 0 | k_{r_1} |
| a_{2,1} = \{r_1\}, a_{2,2} = \{r_1\} | & 0 + k_{r_1} & 0 + 0 & 0 \\
+ & & & k_{r_1} \\
\end{array}
\]

Table 17: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{r_1\}\).

\[
\begin{array}{c|ccc}
| a_2 = a_{2,1} \cup a_{2,2} | & \|1 \oplus r_1\|(|a_{2,1}|) & \|1 \oplus r_2\|(|a_{2,2}|) |
| \hline
| a_{2,1} = \emptyset, a_{2,2} = \{r_2\} | & 1 + 0 & 0 + k_{r_2} & k_{r_2} \\
| a_{2,1} = \{r_2\}, a_{2,2} = \emptyset | & 0 + 0 & 1 + 0 & 0 \\
| a_{2,1} = \{r_2\}, a_{2,2} = \{r_2\} | & 0 + 0 & 0 + k_{r_2} & 0 \\
+ & & & k_{r_2} \\
\end{array}
\]

Table 18: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{r_2\}\).

\[
\begin{array}{c|ccc}
| a_2 = a_{2,1} \cup a_{2,2} | & \|1 \oplus r_1\|(|a_{2,1}|) & \|1 \oplus r_2\|(|a_{2,2}|) |
| \hline
| a_{2,1} = \emptyset, a_{2,2} = \{s, r_1\} | & 1 + 0 & 0 + 0 & 0 \\
| a_{2,1} = \{s, r_1\}, a_{2,2} = \emptyset | & 0 + k_{s, r_1} & 1 + 0 | k_{s, r_1} |
| a_{2,1} = \{s, r_1\}, a_{2,2} = \{s\} | & 0 + k_{s, r_1} & 0 + 0 & 0 \\
| a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1\} | & 0 + k_{s, r_1} & 0 + 0 & 0 \\
| a_{2,1} = \{s\}, a_{2,2} = \{s, r_1\} | & 0 + 0 & 0 + 0 & 0 \\
| a_{2,1} = \{s\}, a_{2,2} = \{s, r_1\} | & 0 + k_{s, r_1} & 0 + 0 & 0 \\
| a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1\} | & 0 + k_{s, r_1} & 0 + 0 & 0 \\
+ & & & k_{s, r_1} \\
\end{array}
\]

Table 19: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{s, r_1\}\).
\[
\begin{array}{|c|c|c|}
\hline
a_2 = a_{2,1} \cup a_{2,2} & \|1 \oplus r_1\| \|\{a_{2,1}\}\| & \|1 \oplus r_2\| \|\{a_{2,2}\}\| \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s, r_2\} & 1 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \emptyset & 0 + 0 & 1 + 0 \\
a_{2,1} = s, a_{2,2} = \{r_2\} & 0 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{r_2\}, a_{2,2} = \{s\} & 0 + 0 & 0 + 0 \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \{s\} & 0 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \{r_2\} & 0 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \{s, r_2\} & 0 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \{r_2\} & 0 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{s, r_2\}, a_{2,2} = \{s, r_2\} & 0 + 0 & 0 + k_{r_2} \\
+ & & k_{r_2} \\
\hline
\end{array}
\]

Table 20: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{s, r_2\}\).

\[
\begin{array}{|c|c|c|}
\hline
a_2 = a_{2,1} \cup a_{2,2} & \|1 \oplus r_1\| \|\{a_{2,1}\}\| & \|1 \oplus r_2\| \|\{a_{2,2}\}\| \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{r_1, r_2\} & 1 + 0 & 0 + k_{r_2} \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \emptyset & 0 + k_{r_1} & 1 + 0 \\
a_{2,1} = \{r_1\}, a_{2,2} = \{r_2\} & 0 + k_{r_1} & 0 + k_{r_2} \\
a_{2,1} = \{r_2\}, a_{2,2} = \{r_1\} & 0 + 0 & 0 + 0 \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1\} & 0 + k_{r_1} & 0 + k_{r_2} \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_2\} & 0 + k_{r_1} & 0 + 0 \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1, r_2\} & 0 + k_{r_1} & 0 + k_{r_2} \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_2\} & 0 + k_{r_1} & 0 + k_{r_2} \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1, r_2\} & 0 + k_{r_1} & 0 + k_{r_2} \\
+ & & k_{r_1} + k_{r_2} \\
\hline
\end{array}
\]

Table 21: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{r_1, r_2\}\).
### 9.3 Weighted Atomic Broadcast

The weight of the \(wAf(P)\) element \(z = s \otimes (1 \oplus r_1 \otimes r_2)\) on the interactions set \(\gamma = \{\{s\}, \{s, r_1, r_2\}\} \in \Gamma(P)\) was computed using the tables presented below. Specifically, we used the primary Tables 23-24 as well as the auxiliary Tables 3-10 presented in Subsection 9.1 for the weighted Rendezvous scheme.

![Table 22](image)

Table 22: \((1 \oplus r_1) \otimes (1 \oplus r_2)\) and \(a_2 = \{s, r_1, r_2\}\).

![Table 23](image)

Table 23: Weighted Atomic Broadcast and \(a = \{s\}\).
Table 24: Weighted Atomic Broadcast and \( a = \{s, r_1, r_2\} \).

### 9.4 Weighted Causality Chain

The weight of the \( wAI(P) \) element \( z = s \otimes (1 \oplus r_1 \otimes (1 \oplus r_2)) \) on the interactions set \( \gamma = \{\{s\}, \{s, r_1\}, \{s, r_1, r_2\}\} \in \Gamma(P) \) was computed by using the following tables. The primary tables are Tables 25-27, while Tables 28-35 are the auxiliary ones.

Table 25: Weighted Causality Chain and \( a = \{s\} \).
Table 26: Weighted Causality Chain and $a = \{s, r_1\}$.

| $a = a_1 \cup a_2$ | $||s(\{a_1\})||$ | $||1 \otimes r_1 \otimes (1 \otimes r_2)|| (\{a_2\})$ |
|---------------------|------------------|----------------------------------|
| $a_1 = \emptyset, a_2 = \{s, r_1\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \emptyset$ | $k_s$ | 1 + 0 |
| $a_1 = \{s\}, a_2 = \{r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{r_1\}, a_2 = \{s\}$ | 0 | 0 = 0 |
| $a_1 = \{s\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1\}, a_2 = \{r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{r_1\}, a_2 = \{s, r_1\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1\}, a_2 = \{r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{r_1\}, a_2 = \{s, r_1\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{s\}, a_2 = \{s, r_1, r_2\}$ | $k_s$ | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_1\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_2\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_2\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{r_1, r_2\}$ | 0 | 0 + $k_{r_1}$ |
| $a_1 = \{s, r_1, r_2\}, a_2 = \{s, r_2\}$ | 0 | 0 + $k_{r_1}$ |

Table 27: Weighted Causality Chain and $a = \{s, r_1, r_2\}$.

55
\[
\begin{array}{|c|c|c|}
\hline
\| r_1 \otimes (1 \oplus r_2) \| (\{a_2\}) & r_1 \| (\{a_1\}) \| 1 + r_2 \| (\{a_2\}) \\
\hline
a_2 = a_{2,1} \cup a_{2,2} & r_1 \| (\{a_1\}) & 1 + 0 & 0 \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
+a & 0 & 1 + 0 & 0 \\
\hline
\end{array}
\]

Table 28: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \emptyset \).

\[
\begin{array}{|c|c|c|}
\hline
\| r_1 \otimes (1 \oplus r_2) \| (\{a_2\}) & r_1 \| (\{a_1\}) \| 1 + r_2 \| (\{a_2\}) \\
\hline
a_2 = a_{2,1} \cup a_{2,2} & r_1 \| (\{a_1\}) & 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
+a & 0 & 0 + 0 & 0 \\
\hline
\hline
a_{2,1} = \{s\}, a_{2,2} = \{r_1\} & 0 & 0 & 0 \\
\hline
\hline
a_{2,1} = \{r_1\}, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
+a & 0 & 0 + k_{r_1} & 0 \\
\hline
\end{array}
\]

Table 29: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \{s\} \).

\[
\begin{array}{|c|c|c|}
\hline
\| r_1 \otimes (1 \oplus r_2) \| (\{a_2\}) & r_1 \| (\{a_1\}) \| 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_2 = a_{2,1} \cup a_{2,2} & r_1 \| (\{a_1\}) & 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{r_1\}, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
+a & 0 & 0 + k_{r_1} & 0 \\
\hline
\end{array}
\]

Table 30: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \{r_1\} \).

\[
\begin{array}{|c|c|c|}
\hline
\| r_1 \otimes (1 \oplus r_2) \| (\{a_2\}) & r_1 \| (\{a_1\}) \| 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_2 = a_{2,1} \cup a_{2,2} & r_1 \| (\{a_1\}) & 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{r_2\} & 0 & 0 + k_{r_1} & 0 \\
\hline
a_{2,1} = \{r_2\}, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
+a & 0 & 0 + k_{r_1} & 0 \\
\hline
\end{array}
\]

Table 31: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \{r_2\} \).

\[
\begin{array}{|c|c|c|}
\hline
\| r_1 \otimes (1 \oplus r_2) \| (\{a_2\}) & r_1 \| (\{a_1\}) \| 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_2 = a_{2,1} \cup a_{2,2} & r_1 \| (\{a_1\}) & 1 + r_2 \| (\{a_2\}) \| k_{r_1} \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s,r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s,r_1\}, a_{2,2} = \emptyset & 0 & 1 + 0 & 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \{r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s\}, a_{2,2} = \{s,r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s,r_1\}, a_{2,2} = \{s\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s,r_1\}, a_{2,2} = \{s,r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s,r_1\}, a_{2,2} = \{r_1\} & 0 & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s,r_1\}, a_{2,2} = \{s,r_1\} & 0 & 0 + 0 & 0 \\
\hline
+a & 0 & 0 + k_{r_1} & 0 \\
\hline
\end{array}
\]

Table 32: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \{s,r_1\} \).

56
Table 33: $r_1 \otimes (1 \oplus r_2)$ and $a_2 = \{s, r_2\}$.

| $a_2 = a_{2,1} \cup a_{2,2}$ | $|r_1| (\{a_{2,1}\})$ | $|1 \oplus r_2| (\{a_{2,2}\})$ |  |
|-----------------------------|-----------------|-----------------|---|
| $a_{2,1} = \emptyset, a_{2,2} = \{s, r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{s, r_2\}, a_{2,2} = \emptyset$ | $0$ | $1 + 0$ | $0$ |
| $a_{2,1} = \{s\}, a_{2,2} = \{r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{r_2\}, a_{2,2} = \{s\}$ | $0$ | $0 + 0$ | $0$ |
| $a_{2,1} = \{s\}, a_{2,2} = \{s, r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{s, r_2\}, a_{2,2} = \{s\}$ | $0$ | $0 + 0$ | $0$ |
| $a_{2,1} = \{r_2\}, a_{2,2} = \{s, r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{s, r_2\}, a_{2,2} = \{s, r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |

Table 34: $r_1 \otimes (1 \oplus r_2)$ and $a_2 = \{r_1, r_2\}$.

| $a_2 = a_{2,1} \cup a_{2,2}$ | $|r_1| (\{a_{2,1}\})$ | $|1 \oplus r_2| (\{a_{2,2}\})$ |  |
|-----------------------------|-----------------|-----------------|---|
| $a_{2,1} = \emptyset, a_{2,2} = \{r_1, r_2\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{r_1, r_2\}, a_{2,2} = \emptyset$ | $k_{r_1}$ | $1 + 0$ | $k_{r_1}$ |
| $a_{2,1} = \{r_1\}, a_{2,2} = \{r_2\}$ | $k_{r_1}$ | $0 + k_{r_2}$ | $k_{r_1} \cdot k_{r_2}$ |
| $a_{2,1} = \{r_2\}, a_{2,2} = \{r_1\}$ | $0$ | $0 + 0$ | $0$ |
| $a_{2,1} = \{r_1\}, a_{2,2} = \{r_1, r_2\}$ | $k_{r_1}$ | $0 + k_{r_2}$ | $k_{r_1} \cdot k_{r_2}$ |
| $a_{2,1} = \{r_2\}, a_{2,2} = \{r_1, r_2\}$ | $k_{r_2}$ | $0 + 0$ | $0$ |
| $a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1\}$ | $0$ | $0 + k_{r_2}$ | $0$ |
| $a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_2\}$ | $k_{r_1}$ | $0 + k_{r_2}$ | $k_{r_1} \cdot k_{r_2}$ |
| $a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{r_1, r_2\}$ | $k_{r_1}$ | $0 + k_{r_2}$ | $k_{r_1} \cdot k_{r_2}$ |

+ $k_{r_1} \cdot (k_{r_1} \cdot k_{r_2})$
\[
\begin{array}{|c|c|c|c|}
\hline
a_2 = a_{2,1} \cup a_{2,2} & ||r_1 \otimes (1 \oplus r_2)|| (\{a_2\}) & ||1 \oplus r_2|| (\{a_2\}) & - \\
\hline
a_{2,1} = \emptyset, a_{2,2} = \{s, r_1, r_2\} & 0 & 0 + k_{r_2} & 0 \\
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \emptyset & k_{r_1} & 1 + 0 & k_{r_1} \\
a_{2,1} = \{s\}, a_{2,2} = \{r_1, r_2\} & 0 & 0 + k_{r_2} & 0 \\
a_{2,1} = \{r_1, r_2\}, a_{2,2} = \{s\} & k_{r_1} & 0 + 0 & 0 \\
a_{2,1} = \{s, r_1\}, a_{2,2} = \{r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{r_2\}, a_{2,2} = \{s, r_1\} & 0 & 0 + 0 & 0 \\
a_{2,1} = \{r_1\}, a_{2,2} = \{r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{s\}, a_{2,2} = \{s, r_1, r_2\} & 0 & 0 + k_{r_2} & 0 \\
\hline
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\} & k_{r_1} & 0 + 0 & 0 \\
a_{2,1} = \{r_1\}, a_{2,2} = \{s, r_1, r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{r_1\} & 0 & 0 + 0 & 0 \\
a_{2,1} = \{r_1\}, a_{2,2} = \{s, r_1, r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s, r_1\} & 0 & 0 + 0 & 0 \\
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\} & k_{r_1} & 0 + 0 & 0 \\
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s\} & k_{r_1} & 0 + 0 & 0 \\
\hline
a_{2,1} = \{s, r_1, r_2\}, a_{2,2} = \{s\} & k_{r_1} & 0 + 0 & 0 \\
a_{2,1} = \{s, r_1\}, a_{2,2} = \{s, r_1, r_2\} & k_{r_1} & 0 + k_{r_2} & k_{r_1} \cdot k_{r_2} \\
\hline
\end{array}
\]

Table 35: \( r_1 \otimes (1 \oplus r_2) \) and \( a_2 = \{s, r_1, r_2\} \).