Global Consensus through Local Synchronization (Technical Report)

Sung-Shik T.Q. Jongmans and Farhad Arbab

Centrum Wiskunde & Informatica, Amsterdam, Netherlands
[jongmans,farhad]@cwi.nl

Abstract. Coordination languages have emerged for the specification and implementation of interaction protocols among concurrent entities. Currently, we are developing a code generator for one such a language, based on the formalism of constraint automata (CA). As part of the compilation process, our tool computes the CA-specific synchronous product of a number of CA, each of which models a constituent of the protocol to generate code for. This ensures that implementations of those CA at run-time reach a consensus about their global behavior in every step. However, using the existing product operator on CA can be practically problematic. In this paper, we provide a solution by defining a new, local product operator on CA that avoids those problems. We then identify a sufficiently large class of CA for which using our new product instead of the existing one is semantics-preserving. Finally, we describe how to apply this result to code generation and also sketch how to use the same theory for projecting choreographies.

1 Introduction

Context. Coordination languages have emerged for the specification and implementation of interaction protocols among concurrent entities (services, threads, etc.). This class of languages includes Reo [1,2], a graphical dataflow language for compositional construction of *connectors*: communication media through which entities can interact with each other. Figure 1 shows example connectors in their usual graphical syntax. Briefly, connectors consist of one or more *channels*, through which data items flow, and a number of *nodes*, on which channel ends coincide. Through connector *composition* (the act of gluing connectors together on their common nodes), users can construct arbitrarily complex connectors.

To implement and use connectors in real applications, one must derive implementations from their graphical specification [8,13,14,15,19,20,21], as precompiled executable code or using a run-time interpretation engine. Roughly two implementation approaches currently exist. In the *distributed approach*, one implements the behavior of each of the k constituents of a connector and runs these k implementations concurrently as a distributed system; in the *centralized approach*, one computes the behavior of a connector as a whole, implements this behavior, and runs this implementation sequentially as a centralized system.

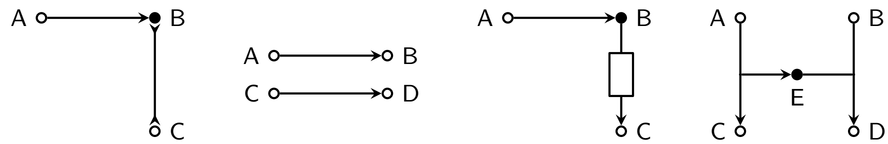

Fig. 1: Four example connectors. Open circles represent *boundary nodes*, on which entities perform I/O-operations; filled circles represent nodes for internal routing. Every connector in this figure consists of two *primitives* (i.e., minimal subconnectors); the pairs of primitives in the first, third, and fourth connector have one common node.

Currently, we are developing a Reo-to-Java code generator using the centralized approach based on the formalism of *constraint automata* (CA) [4]. On input of a graphical connector specification (as an XML file), our tool automatically generates code in four steps. *First*, it extracts from the specification a list of the channels constituting the specified connector. *Second*, it consults a database to find for every channel in the list a "small" CA that formally describes the behavior of that particular channel. *Third*, it computes the product of the CA in the constructed collection to obtain one "big" CA describing the behavior of the whole connector. *Fourth*, it feeds a data structure representing that big CA to a template. Essentially, this template is an incomplete Java class with "holes" that need be "filled" (with information from the data structure). The class generated in this way implements Java's `Runnable` interface. This means that a Java virtual machine can execute the implemented `run` method (declared in `Runnable` and generated by our tool), which simulates the big CA computed in the third step, sequentially in a separate thread (details appear elsewhere [13]).

Problem. Computing one big CA (the third step of the centralized approach) and afterward translating it to sequential code (the fourth step) can be problematic: at run-time, the generated implementation may unnecessarily restrict parallelism among independent transitions.[1] The problem is implementing such a big CA using exactly one thread: single-threaded programs cannot execute multiple independent transitions simultaneously, but instead, they force those transitions to execute one after the other (see Section 2 for details). Consequently, although formally sound, the generated implementation may run overly sequentially (e.g., if the first transition to execute takes a long time to complete, while other transitions could have fired manifold during that time).

One approach to this problem is to *not* compute one big CA but generate code directly for each of the small CA instead, essentially moving from the centralized approach to the distributed approach: the implementations of the small CA compute the product operators between them at run-time instead of at compile-time. Although this approach solves the stated problem—independent

[1] Independent transitions cannot disable each other by firing.

transitions can execute simultaneously—the necessary distributed algorithms for run-time product computation may inflict a substantial amount of overhead.

Contribution. This paper provides a better solution to the stated problem by offering a middle ground between centralized and distributed approaches, wherein some subsets of the constituent automata are statically composed to comprise a distributed system of locally centralized automata. Typically, each locally centralized automaton interacts/synchronizes with few other such automata for its transitions, while it represents the composition of a subset of the constitutent automata that interact/synchronize with each other relatively heavily.

Taking the purely distributed approach as our starting point, we define a new product operator whose computation at run-time requires only relatively simple distributed algorithms—CA need to communicate only locally (i.e., with "neighbors") instead of globally (i.e., with everybody)—while allowing independent transitions to execute simultaneously. We then characterize a class of product automata where substituting the existing product operator with our new product operator is semantics-preserving. This class includes product automata whose constituents communicate only *asynchronously* with each other, and so, the optimization technique based on the identification of synchronous and asynchronous regions of connectors can be combined with our results [20].

The rest of this paper looks as follows. In Section 2, we introduce the automata we work with in this paper, including their existing product operator. In Section 3, we introduce our new product operator. In Section 4, we define a class of automata for which substituting the existing product operator with our new product operator is semantics-preserving. In Section 5, we sketch how connector implementations can compute the new product operator at run-time. In Section 6, we discuss related work and conclude.

Although inspired by Reo, we can express our main results in a purely automata-theoretic setting. We therefore skip an introduction to Reo; interested readers may consult [1,2].

2 Preliminaries: Port Automata

Many formalisms exist for mathematically defining the semantics of connectors [12]; our code generator, for instance, relies on constraint automata (CA). In this paper, however, we adopt a simplification of CA, called *port automata* (PA) [16]. We prefer PA, because they allow us to focus on the core of our problem (synchronization of communication) without getting distracted by those details of CA (the data exchanged in communication) irrelevant to our present purpose. The results in this paper straightforwardly carry over from PA to CA.

A PA consists of a finite set of states and transitions between them, each of which has a set of *ports* as label. A transition represents an execution step of a connector, from one internal configuration to the next, where synchronous interaction occurs on the ports labeling that transition. Let $\mathbb{P}\text{ORT}$ and $\mathbb{S}\text{TATE}$ denote global sets of ports and states (see Definitions 13, 14 in Appendix A).

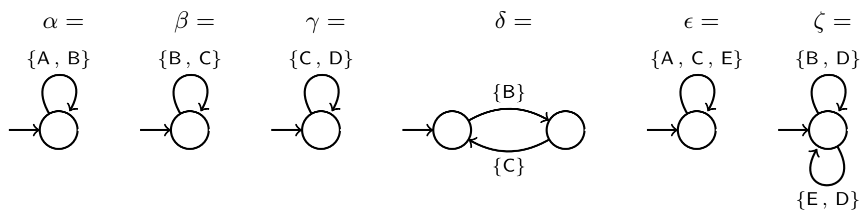

Fig. 2: Port automata, denoted by α, β, γ, δ, ϵ, and ζ, describing the behavior of the primitives constituting the example connectors in Figure 1: α and β model the primitives in the first connector, α and γ the primitives in the second, α and δ the primitives in the third, and ϵ and ζ the primitives in the fourth.

Definition 1 (Universe of port automata). *The universe of* PA*, denoted by* $\mathbb{P}$A *and typically ranged over by α, β, or γ, is the largest set of tuples $(Q\,,\mathcal{P}\,,\longrightarrow,\imath)$ where:*[2]

- $Q \subseteq \mathbb{S}\text{TATE}$; *(states)*
- $\mathcal{P} \subseteq \mathbb{P}\text{ORT}$; *(ports)*
- $\longrightarrow \subseteq Q \times \wp(\mathcal{P}) \times Q$; *(transitions)*
- *and* $\imath \in Q$. *(initial state)*

Figure 2 shows example PA. For instance, the $\{\mathsf{A}\,,\,\mathsf{B}\}$-transition of α describes the only (infinitely repeated) execution step of the horizontal primitive, say Prim, of the first connector in Figure 1. In that execution step, Prim has synchronous interaction on nodes A (a write of data d by the environment) and B (the flow of a copy of d from the horizontal to the vertical primitive). Similarly, the $\{\mathsf{A}\,,\,\mathsf{C}\,,\,\mathsf{E}\}$-transition of ϵ means that the left-hand primitive of the fourth connector in Figure 1 has synchronous interaction on nodes A (a write of data d by the environment), C (a take of a copy of d by the environment), and E (the flow of another copy of d from the left-hand to the right-hand primitive). Port automaton ζ, which models the right-hand primitive of the fourth connector in Figure 1, has two transitions. The right-hand primitive can repeatedly choose between two step: it has synchronous interaction either on nodes B (a write of data d by the environment) and D (a take of a copy of d by the environment) or on nodes E (the flow of data from the left-hand to the right-hand primitive) and D. It can choose the latter transition only if the left-hand primitive simultaneously does its $\{\mathsf{A}\,,\,\mathsf{C}\,,\,\mathsf{E}\}$-transition (otherwise, there is no data available on E).

If α denotes a PA, let $\mathsf{State}(\alpha)$, $\mathsf{Port}(\alpha)$, and $\mathsf{init}(\alpha)$ denote its states, ports, and initial state (see Definition 15 in Appendix A).

We adopt strong bisimilarity on PA as behavioral equivalence [16]: if α and β are bisimilar, denoted by $\alpha \approx \beta$, α can "simulate" every transition of β in every state and vice versa (see Definition 17 in Appendix A).

Individual PA describe the behavior of individual connectors; the application of the existing product operator to such PA models connector composition [16].

[2] Let $\wp(\_)$ denote the power set operator.

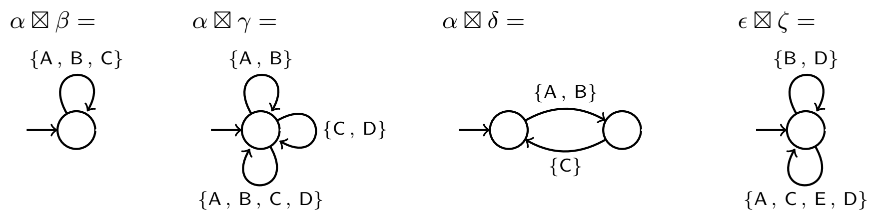

Fig. 3: Port automata describing the behavior of the example connectors in Figure 1, constructed using $\boxtimes$ (α, β, γ, δ, ϵ, and ζ denote the PA in Figure 2).

We define this operator in two steps.[3] First, we introduce a relation that defines when a transition of one PA, say Alice, and a transition of another PA, say Bob, represent execution steps in which Alice and Bob *weakly agree* on their behavior. In that case, Alice and Bob agree on which of their common ports to fire while allowing each other to simultaneously fire other ports. In the following definition, we represent a transition of Alice as a pair of port-sets: one for all Alice's ports ($\mathcal{P}_\alpha$) and one that labels a particular transition of hers (P_α). Likewise for Bob.

Definition 2 (Weak agreement relation). *The weak agreement relation, denoted by $\Diamond$, is the relation on $\wp(\mathbb{P}\text{ORT})^2 \times \wp(\mathbb{P}\text{ORT})^2$ defined as:*

$$(\mathcal{P}_\alpha \, , \, P_\alpha) \Diamond (\mathcal{P}_\beta \, , \, P_\beta) \ \textbf{iff} \ \begin{bmatrix} P_\alpha \subseteq \mathcal{P}_\alpha \ \textbf{and} \ P_\beta \subseteq \mathcal{P}_\beta \\ \textbf{and} \ \mathcal{P}_\alpha \cap P_\beta = \mathcal{P}_\beta \cap P_\alpha \end{bmatrix}$$

Next, we define the existing product operator on PA in terms of $\Diamond$.

Definition 3 (Product operator). *The product operator, denoted by $\_\boxtimes\_$, is the operator on $\mathbb{P}\text{A} \times \mathbb{P}\text{A}$ defined by the following equation:*

$$\alpha \boxtimes \beta = (\mathsf{State}(\alpha) \times \mathsf{State}(\beta) \, , \, \mathsf{Port}(\alpha) \cup \mathsf{Port}(\beta) \, , \, \longrightarrow , \, (\mathsf{init}(\alpha) \, , \, \mathsf{init}(\beta)))$$

where $\longrightarrow$ denotes the smallest relation induced by:

$$\frac{q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \ \textbf{and} \ q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \ \textbf{and} \ (\mathsf{Port}(\alpha) \, , \, P_\alpha) \Diamond (\mathsf{Port}(\beta) \, , \, P_\beta)}{(q_\alpha \, , \, q_\beta) \xrightarrow{P_\alpha \cup P_\beta} (q'_\alpha \, , \, q'_\beta)} \ (\textsc{WkAgr})$$

$$\frac{q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \ \textbf{and} \ q_\beta \in Q_\beta \quad \textbf{and} \ P_\alpha \cap \mathsf{Port}(\beta) = \emptyset}{(q_\alpha \, , \, q_\beta) \xrightarrow{P_\alpha} (q'_\alpha \, , \, q_\beta)} \ (\textsc{Indep}\mathrm{A}) \qquad \frac{q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \ \textbf{and} \ q_\alpha \in Q_\alpha \quad \textbf{and} \ P_\beta \cap \mathsf{Port}(\alpha) = \emptyset}{(q_\alpha \, , \, q_\beta) \xrightarrow{P_\beta} (q_\alpha \, , \, q'_\beta)} \ (\textsc{Indep}\mathrm{B})$$

The previous definition reformulates the product of PA in [16], which is a simplification of the product of CA in [4]. Figure 3 shows examples of the application of

[3] This simplifies relating this product operator to the product operator of Section 3.

$\boxtimes$. The $\{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$-transition in the second PA results from applying rule WK-AGR to disjoint sets of ports. This models that two independent transitions *coincidentally* can happen simultaneously (true concurrency). The following lemma states that bisimilarity is a congruence. See [16, Theorem 1] for a proof.

Lemma 1. $\big[\alpha \approx \beta$ **and** $\gamma \approx \delta\big]$ **implies** $\alpha \boxtimes \gamma \approx \beta \boxtimes \delta$

Furthermore, $\boxtimes$ is associative and commutative.

Interestingly, $\boxtimes$ *"transitively" propagates synchrony* over successive applications. We explain what this means with an example. Suppose Alice knows about ports $\{\mathsf{A}, \mathsf{B}\}$ and has one transition in which she fires exactly those ports. Similarly, suppose Bob knows about ports $\{\mathsf{B}, \mathsf{C}\}$ and has one transition in which he fires exactly those ports. Because these two transitions satisfy $\Diamond$, the product of Alice and Bob has one transition labeled by $\{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$. This means that Alice and Bob always synchronize on their common port B: Alice can perform her transition (i.e., is *willing* to fire B) only if Bob can perform his (i.e., is *ready* to fire B) and vice versa. Now, suppose Carol knows about ports $\{\mathsf{C}, \mathsf{D}\}$ and has one transition in which she fires exactly those ports. By the same reasoning as before, the product of $\big[$the product of Alice and Bob$\big]$[4] and Carol has one transition labeled by $\{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$. Thus, in the product of Alice, Bob, and Carol, Alice "transitively" synchronizes with Carol, through Bob.[5]

The problem addressed in this paper is that code generators using the centralized approach produce connector implementations that may unnecessarily restrict parallelism. To illustrate this problem, suppose Dave knows about ports $\{\mathsf{E}, \mathsf{F}\}$ and has one transition in which he fires exactly those ports. The product of Alice, Bob, Carol, and Dave computed by a tool using the centralized approach has three transitions: one labeled by $\{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$ (Alice, Bob, Carol make a transition), another labeled by $\{\mathsf{E}, \mathsf{F}\}$ (Dave makes a transition), and yet another labeled by $\{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}\}$ (Alice, Bob, Carol and Dave coincidentally make a transition at the same time by true concurrency). At run-time, in every iteration of its main loop, the thread simulating this big automaton nondeterministically picks one of those transitions, checks it for enabledness (in which case all ports are ready to fire), and if so, executes it. By this scheme, as soon as the automaton thread has selected the transition labeled by $\{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}\}$, the transition labeled by $\{\mathsf{E}, \mathsf{F}\}$ has to wait for the next iteration, even if it is enabled already in the current iteration. In other words, Dave cannot execute at its own pace despite being independent of Alice, Bob, and Carol.

Although the centralized approach may unnecessarily restrict parallelism, it guarantees high *throughput* compared to the alternative, distributed approach of generating code for Alice, Bob, Carol, and Dave individually. The problem with the distributed approach is the communication necessary for computing $\boxtimes$ at run-time. To see this, suppose that we indeed have separate threads simulating the automata of Alice, Bob, Carol, and Dave. Now, if Alice at some point becomes

[4] Square brackets for readability.

[5] This property of $\boxtimes$ models an important feature of Reo: compositional construction of globally synchronous protocol steps out of locally synchronous parts.

willing to execute her $\{\mathsf{A}, \mathsf{B}\}$ transition, she must ask Bob if he is ready to execute his $\{\mathsf{B}, \mathsf{C}\}$ transition. Before he can answer Alice's question, however, Bob in turn must ask Carol if she is ready to execute her $\{\mathsf{C}, \mathsf{D}\}$ transition. All this communication negatively affects throughput: it takes much longer for Alice, Bob, and Carol to agree on synchronously executing their individual transitions than for one big automaton to make and carry out such a decision by itself. Nevertheless, the distributed approach enhances parallelism: Dave can execute his transition *while* Alice, Bob, and Carol communicate to come to an agreement.

3 A New Local Product Operator

The approaches of the previous section force one to choose between two desirable properties: high throughput between *inter*dependent port automata (PA), at the cost of parallelism, and maximal parallelism between *in*dependent ones, at the cost of throughput. We need to find a middle ground between the purely centralized and fully distributed approaches that has both these desirable qualities.

Working toward such an approach, we start from the purely distributed approach of computing $\boxtimes$ at run-time through global, transitive communication between automaton threads (e.g., Alice talks to Bob, who in turn talks to Carol, etc.). The idea is to bound this transitivity: generally, when some Alice asks some Bob if he is ready to fire a transition involving common ports, Bob *should* immediately answer *without engaging others*. By doing so, Alice and Bob achieve a higher throughput, while independent others can still execute at their own pace.

In the proposed approach, automaton threads no longer compute $\boxtimes$: instead, they compute a new product operator whose run-time computation requires only local communication. Problematically, however, computing that new product operator instead of $\boxtimes$ can be unsound or incomplete, sometimes to the extent of deadlock. Which of those two happens depends on *how* Bob immediately answers Alice in cases where he actually should have consulted Carol (and possibly others). If Bob replies being ready, the firing of Alice's ports (including her ports common with Bob) incorrectly introduces asynchrony between Bob's two ports. However, if Bob always replies *not* being ready, he and Alice never interact on their common ports. In the rest of this section, we formalize the new product operator and make a first effort at studying under which circumstances substituting $\boxtimes$ with the new product operator is semantics-preserving.

First, we introduce a relation that defines when transitions of Alice and Bob represent execution steps in which they *strongly agree* on their behavior (cf. Definition 2 of $\Diamond$). In that case, they agree on which of their common ports to fire (possibly none), and either Alice forbids Bob to simultaneously fire any other port or vice versa. Afterward, we define our new product operator on PA.

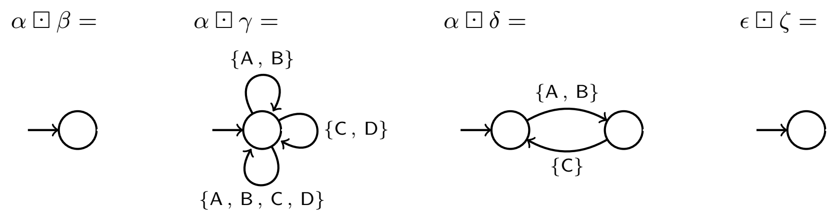

Fig. 4: Port automata constructed using $\boxdot$ (α, β, γ, δ, ϵ, and ζ denote the PA in Figure 2).

Definition 4 (Strong agreement relation). *The strong agreement relation, denoted by* $\blacklozenge$*, is the relation on* $\wp(\mathbb{P}\textsc{ort})^2 \times \wp(\mathbb{P}\textsc{ort})^2$ *defined as:*

$$(\mathcal{P}_\alpha\,,\,P_\alpha) \blacklozenge (\mathcal{P}_\beta\,,\,P_\beta) \;\;\textbf{iff}\;\; \begin{bmatrix} P_\alpha \subseteq \mathcal{P}_\alpha \;\textbf{ and }\; P_\beta \subseteq \mathcal{P}_\beta \;\textbf{ and} \\ \begin{bmatrix} P_\alpha = \mathcal{P}_\alpha \cap P_\beta \;\textbf{ or }\; P_\beta = \mathcal{P}_\beta \cap P_\alpha \\ \textbf{or }\; \mathcal{P}_\alpha \cap P_\beta = \emptyset = \mathcal{P}_\beta \cap P_\alpha \end{bmatrix} \end{bmatrix}$$

Definition 5 (Local product operator, l-product). *The local product operator, l-product, denoted by* $\_ \boxdot \_$*, is the operator on* $\mathbb{P}\textsc{a} \times \mathbb{P}\textsc{a}$ *defined by the following equation:*

$$\alpha \boxdot \beta = (\mathsf{State}(\alpha) \times \mathsf{State}(\beta)\,,\,\mathsf{Port}(\alpha) \cup \mathsf{Port}(\beta)\,,\,\longrightarrow\,,\,(\mathsf{init}(\alpha)\,,\,\mathsf{init}(\beta)))$$

where $\longrightarrow$ *denotes the smallest relation induced by* $\textsc{IndepA}$*,* $\textsc{IndepB}$*, and:*

$$\frac{q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \;\textbf{ and }\; q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \;\textbf{ and }\; (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta)}{(q_\alpha\,,\,q_\beta) \xrightarrow{P_\alpha \cup P_\beta} (q'_\alpha\,,\,q'_\beta)} \quad (\textsc{StAgr})$$

Figure 4 shows examples of the application of $\boxdot$. The following lemma states that bisimilarity is a congruence. See page 29 for a proof.

Lemma 2. $\big[\alpha \approx \beta$ **and** $\gamma \approx \delta\big]$ **implies** $\alpha \boxdot \gamma \approx \beta \boxdot \delta$

Furthermore, $\boxdot$ is commutative but generally *not* associative. This makes using $\boxdot$ for modeling purposes nontrivial. We address this issue in Section 5. To minimize numbers of parentheses in our notation, we assume right-associativity for $\boxdot$. For instance, we write $\alpha \boxdot \beta \boxdot \gamma \boxdot \delta$ for $\alpha \boxdot (\beta \boxdot (\gamma \boxdot \delta))$.

As informally explained earlier, substituting $\boxtimes$ with $\boxdot$ is not always semantics-preserving. It is, for instance, for the two l-products in the middle of Figure 4 (cf. the two products in the middle of Figure 3) but not for the l-products on the sides. To determine when substituting $\boxtimes$ with $\boxdot$ is semantics-preserving, we first define when Alice is a *subautomaton* of Bob. In that case, Bob has at least every transition that Alice has.

Definition 6 (Subautomaton relation). *The subautomaton relation, denoted by* $\sqsubseteq$*, is the relation on* $\mathbb{P}\textsc{a} \times \mathbb{P}\textsc{a}$ *defined as:*

$$(Q\,,\,\mathcal{P}\,,\,\longrightarrow_\alpha\,,\,\imath) \sqsubseteq (Q\,,\,\mathcal{P}\,,\,\longrightarrow_\beta\,,\,\imath) \;\;\textbf{iff}\;\; \longrightarrow_\alpha \subseteq \longrightarrow_\beta$$

The following proposition follows directly from the previous definition. In the rest of this section, we investigate under which circumstances its premise holds.

Proposition 1. $\big[\alpha \sqsubseteq \beta$ **and** $\beta \sqsubseteq \alpha\big]$ **implies** $\alpha = \beta$

Before showing that the l-product of Alice and Bob is a subautomaton of their product, the next lemma states that strong agreement implies weak agreement: if Alice fires exactly those common ports that Bob fires or vice versa, Alice and Bob agree on their common ports. See page 30 for a proof.

Lemma 3. $(\mathcal{P}_\alpha\,,\, P_\alpha) \blacklozenge (\mathcal{P}_\beta\,,\, P_\beta)$ **implies** $(\mathcal{P}_\alpha\,,\, P_\alpha) \lozenge (\mathcal{P}_\beta\,,\, P_\beta)$

The next lemma states that the l-product of Alice and Bob is a subautomaton of their product: the product of Alice and Bob can do *at least* the same as their l-product. See page 31 for a proof (which uses Lemma 3).

Lemma 4. $\alpha \boxdot \beta \sqsubseteq \alpha \boxtimes \beta$

The product of Alice and Bob is not necessarily a subautomaton of their l-product: if Alice and Bob agree on which of their common ports to fire, this does not necessarily mean that they fire no other ports. To characterize the cases in which they do, we define *conditional strong agreement* as a relation "in between" of $\blacklozenge$ and $\lozenge$ (and lifted from transitions to PA): Alice and Bob conditionally strongly agree iff, for each of their transitions, their weak agreement on which of their common ports to fire implies their strong agreement.

Definition 7 (Conditional strong agreement relation). *The conditional strong agreement relation, denoted by* $\lozenge\!\!\!\blacklozenge$*, is the relation on* $\mathbb{P}\text{A} \times \mathbb{P}\text{A}$ *defined as:*

$$\begin{array}{c}(Q_\alpha\,,\, \mathcal{P}_\alpha\,,\, \longrightarrow_\alpha\,,\, \imath_\alpha)\\ \lozenge\!\!\!\blacklozenge\, (Q_\beta\,,\, \mathcal{P}_\beta\,,\, \longrightarrow_\beta\,,\, \imath_\beta)\end{array} \;\textbf{iff}\; \left[\begin{array}{c}\left[\begin{array}{c}\left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and}\\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\, P_\beta)\end{array}\right]\\ \textbf{implies } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta)\end{array}\right]\\ \textbf{for all } q_\alpha\,,\, q_\beta\,,\, q'_\alpha\,,\, q'_\beta\,,\, P_\alpha\,,\, P_\beta\end{array}\right]$$

The next lemma states that if Alice and Bob conditionally strongly agree, their product is a subautomaton of their l-product (cf. Lemma 4). See page 32 for a proof.

Lemma 5. $\alpha \lozenge\!\!\!\blacklozenge \beta$ **implies** $\alpha \boxtimes \beta \sqsubseteq \alpha \boxdot \beta$

We end this section with the following theorem: if Alice and Bob conditionally strongly agree, substituting $\boxtimes$ with $\boxdot$ is semantics-preserving (in fact, not just under bisimilarity but even under structural equality). See page 34 for a proof (which uses Proposition 1 and Lemmas 4, 5).

Theorem 1. $\alpha \lozenge\!\!\!\blacklozenge \beta$ **implies** $\alpha \boxdot \beta = \alpha \boxtimes \beta$

4 Substituting $\boxtimes$ with $\boxdot$, a Cheaper Characterization

To test if Alice and Bob conditionally strongly agree, one must pairwise compare their transitions. This can be computationally expensive (i.e., $\mathcal{O}(n_1 n_2)$, where n_1 and n_2 denote the numbers of transitions), and it makes the $\lozenge$-based characterization, although (conjectured to be) complete, hard to apply in practice. In this section, we therefore study a cheaper characterization of (a subset of) conditionally strongly agreeing port automata (PA) without restricting the applicability of $\boxdot$ for our present purpose.

In Section 2, we explained reduction of parallelism in terms of independent PA. Therefore, substituting $\boxtimes$ with $\boxdot$ should be semantics-preserving at least when applied to such PA. We start by formally defining when Alice and Bob are independent: in that case, they have no common ports.

Definition 8 (Independence relation). *The independence relation, denoted by $\asymp$, is the relation on $\mathbb{P}\text{A} \times \mathbb{P}\text{A}$ defined as:*

$$\alpha \asymp \beta \;\; \textbf{iff} \;\; \mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset$$

The next lemma states that if Alice and Bob are independent, they conditionally strongly agree (because their independence means that Alice and Bob have no common ports). See page 34 for a proof.

Lemma 6. $\alpha \asymp \beta$ **implies** $\alpha \lozenge \beta$

Lemma 6 and Theorem 1 imply that substituting $\boxtimes$ with $\boxdot$ is semantics-preserving, if their operands satisfy the independence relation. Moreover, checking $\asymp$ costs less than checking whether PA conditionally strongly agree: $\mathcal{O}(1)$ versus $\mathcal{O}(n_1 n_2)$. The next lemma states another important property, namely that $\boxdot$ preserves independence: if Alice is independent of Bob and Carol individually, she is independent of Bob and Carol together. See page 35 for a proof.

Lemma 7. $\big[\alpha \asymp \beta$ **and** $\alpha \asymp \gamma\big]$ **implies** $\alpha \asymp \beta \boxdot \gamma$

Although checking PA for independence is cheap, the result implied by Lemma 6 and Theorem 1 in its present form has limited practical value: total independence is a condition rarely satisified by the PA encountered in code generation of a composite system. To get a more useful similar result, we now introduce the notion of *slavery* and afterward combine it with independence. We start by formally defining when Bob is a slave of Alice: in that case, every transition of Bob that involves *some* ports common with Alice, involves *only* ports common with Alice. In other words, if common ports are involved, Alice completely dictates what Bob does. Our notion of slavery does not prevent Bob from freely executing transitions involving only ports that Alice does not know about.

Definition 9 (Slave relation). *The slave relation, denoted by $\mapsto$, is the relation on $\mathbb{P}\text{A} \times \mathbb{P}\text{A}$ defined as:*

$$\begin{array}{l} (Q_\beta\,,\, \mathcal{P}_\beta\,,\, \longrightarrow_\beta\,,\, \imath_\beta) \\ \mapsto \alpha \end{array} \;\; \textbf{iff} \;\; \left[\begin{array}{c} \Big[\begin{bmatrix} q_\beta \xrightarrow{P_\beta} q'_\beta \;\; \textbf{and} \\ P_\beta \cap \mathsf{Port}(\alpha) \neq \emptyset \end{bmatrix} \;\; \textbf{implies} \;\; P_\beta \subseteq \mathsf{Port}(\alpha)\Big] \\ \textbf{for all} \;\; q_\beta\,,\, q'_\beta\,,\, P_\beta \end{array}\right]$$

The next lemma states that if Bob is a slave of Alice, they conditionally strongly agree (i.e., Alice forces her will upon Bob). See page 35 for a proof.

Lemma 8. $\beta \mapsto \alpha$ **implies** $\beta \mathrel{\langle\!\blacklozenge} \alpha$

Lemma 8 and Theorem 1 imply that substituting $\boxtimes$ with $\boxdot$ is semantics-preserving, if their operands satisfy the slave relation. Moreover, checking $\mapsto$ costs less than checking whether PA conditionally strongly agree: $\mathcal{O}(n_1)$ versus $\mathcal{O}(n_1 n_2)$. The next lemma states another important property, namely that $\boxdot$ preserves slavery: if Bob is a slave of Alice, he is a slave of Alice and Carol together. See page 37 for a proof.

Lemma 9. $\beta \mapsto \alpha$ **implies** $\beta \mapsto \alpha \boxdot \gamma$

By combining independence and slavery, we obtain the notion of *conditional slavery*: Bob is a conditional slave of Alice iff Alice and Bob not being independent implies that Bob is a slave of Alice.

Definition 10 (Conditional slave relation). *The conditional slave relation, denoted by* $\Mapsto$*, is the relation on* $\mathbb{P}\text{A} \times \mathbb{P}\text{A}$ *defined as:*

$$\beta \Mapsto \alpha \ \textbf{iff} \ \big[\beta \asymp \alpha \ \textbf{or} \ \beta \mapsto \alpha\big]$$

The next lemma states that if Bob is a conditional slave of Alice, they conditionally strongly agree (i.e., Alice and Bob are independent or Alice forces her will upon Bob). See page 38 for a proof (which uses Lemmas 6, 8).

Lemma 10. $\beta \Mapsto \alpha$ **implies** $\beta \mathrel{\langle\!\blacklozenge} \alpha$

The combination of Lemma 10 and Theorem 1 implies that substituting $\boxtimes$ with $\boxdot$ is semantics-preserving, if the PA involved satisfy the conditional slave relation. Moreover, checking the conditional slave relation costs the same as checking the slave relation (i.e., less than checking whether PA conditionally strongly agree). The next lemma states another important property, namely that $\boxdot$ preserves conditional slavery: if Bob is a conditional slave of Alice and Carol individually, he is a conditional slave of Alice and Carol together. The corollary following this lemma generalizes this result from 2 to k individuals. See page 38 for a proof (which uses Lemmas 7, 9).

Lemma 11. $\big[\beta \Mapsto \alpha$ **and** $\beta \Mapsto \gamma\big]$ **implies** $\beta \Mapsto \alpha \boxdot \gamma$

Corollary 1. $\big[\beta \Mapsto \alpha_1$ **and** $\cdots$ **and** $\beta \Mapsto \alpha_k\big]$ **implies** $\beta \Mapsto (\alpha_1 \boxdot \cdots \boxdot \alpha_k)$

With conditional slavery, in contrast to independence alone, one can characterize a sufficiently large class of PA that satisfies the premise of Theorem 1 (i.e., for which substituting $\boxtimes$ with $\boxdot$ is semantics-preserving), as follows. Suppose that we have a list of k PA such that every i-th PA in the list is a conditional slave of all PA in a higher position. Then, the l-product of all PA in this list, starting from the ones with the highest positions and working our way down, is in the class. The following definition formalizes this (recall that $\boxdot$ is right-associative).

Definition 11. *$\mathcal{A}$ denotes the smallest set induced by the following rule:*

$$\frac{\big[i \neq j \ \ \textbf{implies} \ \ \alpha_i \not\Rightarrow \alpha_j\big] \ \ \textbf{for all} \ \ 1 \leq i < j \leq k}{\alpha_1 \boxdot \cdots \boxdot \alpha_k \in \mathcal{A}}$$

Strictly, $\mathcal{A}$ contains terms over ($\mathbb{P}\text{A}$, $\boxdot$), which represent PA, rather than actual elements from $\mathbb{P}\text{A}$. Nevertheless, we often call the elements from $\mathcal{A}$ "PA" for simplicity. Also, instead of writing $\alpha_1 \boxdot \cdots \boxdot \alpha_k$, we sometimes write $\alpha_1 \cdots \alpha_k$or, even more compactly, $[\alpha]_1^k$.[6]

The following theorem states that for every PA in $\mathcal{A}$, substituting $\boxtimes$ for $\boxdot$ is semantics-preserving. See page 38 for a proof (which uses Lemma 10 and Corollary 1).

Theorem 2. $\alpha_1 \boxdot \cdots \boxdot \alpha_k \in \mathcal{A}$ **implies** $\alpha_1 \boxdot \cdots \boxdot \alpha_k = \alpha_1 \boxtimes \cdots \boxtimes \alpha_k$

Although $\alpha_1 \boxdot \cdots \boxdot \alpha_k = \alpha_1 \boxtimes \cdots \boxtimes \alpha_k$ generally does not imply $\alpha_1 \boxdot \cdots \boxdot \alpha_k \in \mathcal{A}$, it does for the examples considerd in this paper. For instance, Figures 3, 4 show that $\beta \boxdot \delta = \beta \boxtimes \delta$ (Figure 2 defines β and δ). By the commutativity of $\boxdot$ and $\boxtimes$, we have also $\delta \boxdot \beta = \delta \boxtimes \beta$. Now, because δ is a slave of β, we conclude that $\delta \boxdot \beta$ is an element of $\mathcal{A}$: indeed, if δ makes a transition involving ports common with β (only B), it fires no other ports (β, in contrast, does fire another port in that case, namely C).

Previously, we claimed that the subclass of PA characterized in this section (i.e., $\mathcal{A}$ in Definition 11) does not restrict the applicability of $\boxdot$ for our purpose. We end this section by substantiating that claim. We start by introducing a further restricted class of PA with a more natural interpretation in our context.

Definition 12. *$\mathcal{B}$ denotes the smallest set induced by the following rule:*

$$\frac{\begin{array}{c}\Big[\big[i_1 \neq i_2 \ \ \textbf{implies} \ \ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}\big] \ \ \textbf{for all} \ \ 1 \leq i_1 \, , \, i_2 \leq k\Big] \ \ \textbf{and} \\ \Big[\big[j_1 \neq j_2 \ \ \textbf{implies} \ \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \ \textbf{for all} \ \ 1 \leq j_1 \, , \, j_2 \leq l\Big] \ \ \textbf{and} \\ \big[\alpha_i \not\Rightarrow \beta_j \ \ \textbf{for all} \ \ 1 \leq i \leq k \, , \, 1 \leq j \leq l\big]\end{array}}{\alpha_1 \boxdot \cdots \boxdot \alpha_k \boxdot \beta_1 \boxdot \cdots \boxdot \beta_l \in \mathcal{B}}$$

The following proposition follows directly from the previous definition.

Proposition 2. $\mathcal{B} \subseteq \mathcal{A}$

The combination of Proposition 2 and Theorem 2 implies that substituting $\boxtimes$ with $\boxdot$ is semantics-preserving for every PA in $\mathcal{B}$.

Informally, every PA in $\mathcal{B}$ is the l-product of (i) k PA that are conditional slaves of *all* other PA in the term and (ii) l pairwise independent PA that are "masters" of the k conditional slaves. The masters, being pairwise independent, do not *directly* communicate with each other. However, when two or more masters share the same slave (the definition of $\mathcal{B}$ allows this), communication between those

[6] Mixing these notations does not induce parentheses: right-associativity is preserved. For instance, $[\alpha]_1^3\beta$ stands for $\alpha_1 \boxdot (\alpha_2 \boxdot (\alpha_3 \boxdot \beta))$—not for $(\alpha_1 \boxdot (\alpha_2 \boxdot \alpha_3)) \boxdot \beta$.

masters occurs *indirectly* through that slave. Such indirect communication is always asynchronous: if it were synchronous, the slave involved would fire ports of more than one of its masters in the same transition, which slavery forbids.

The previous interpretation of masters and slaves corresponds exactly to the notion of synchronous and asynchronous *regions* in the Reo literature [14,20]. Roughly, one can always split a connector into subconnectors—the regions—such that firings of ports in such a subconnector are either purely independent (i.e., always, only one port fires at a time) or require some synchronization (i.e., at least once, more than one port fires). Furthermore, the synchronous regions of a connector are maximal in the sense that no two synchronous regions have common ports: all synchronous regions are, by definition, pairwise independent. Consequently, the PA describing the l synchronous regions of a connector can act as the l masters in a PA term from $\mathcal{B}$.

To actually obtain those PA, for every synchronous region, a code generator during compilation computes the *existing* product of the PA describing the constituents of that particular region (finding the synchronous regions of a connector is trivial). At compile-time, this resembles the purely centralized approach, while at run-time, it ensures high throughput between interdependent "small" PA for constituents of the same synchronous region (i.e., no run-time computation of product operators within synchronous regions). The asynchronous regions then form the "glue" between the synchronous regions: the PA for every asynchronous region has the same shape as δ in Figure 2,[7] and consequently, they can act as the k conditional slaves in a PA term from $\mathcal{B}$. Finally, at run-time, the automaton threads executing the generated code compute the l-product operators.

In summary: a code generator can always process the set of PA describing a connector to a form that satisfies $\mathcal{B}$, by computing $\boxtimes$ between interdependent PA belonging to the same synchronous region at compile-time (for the sake of throughput), and by computing $\boxdot$ between the resulting "medium" PA plus the PA for the asynchronous regions at run-time (for the sake of parallelism). Proposition 2 and Theorem 2 ensure that this is semantics-preserving.

5 Note on Associativity

The associativity of $\boxtimes$ plays a role in the centralized approach and is even more important in the distributed approach. In the centralized approach, it guarantees that it does not matter in which particular order a code generator computes the product of the port automata (PA) for the constituents of a connector—all have

[7] Port automaton δ in Figure 2 describes the behavior of an asynchronous Reo primitive, called Fifo [1,2], with a buffer (of capacity 1) that accepts data on one port (i.e., B), buffers it, and at a later time dispenses that same data on another port (i.e. C). Of the currently common Reo primitives, only Fifo is asynchronous, and so, only Fifo instances induce asynchronous regions in the current practice. In general, a PA modeling an asynchronous region can have more than two states or ports but, crucially, each of its transitions has a singleton set of ports as label (as does δ), which guarantees that that PA can act as a conditional slave in a $\mathcal{B}$-term.

the same semantics. In the distributed approach, it guarantees that it does not matter in which order PA threads communicate with each other: the PA term corresponding to a particular communication order is always bisimilar to the original (because one can freely move parentheses).

Now, recall from Section 3 that $\boxdot$ is generally *not* associative. The structure of the PA terms from $\mathcal{A}$ also reflects this (and the proof of Theorem 2 exploits this structure). This means that the PA constituting such terms must communicate in a particular order at run-time for the substitution of $\boxtimes$ with $\boxdot$ to be semantics-preserving. This can kill performance and seems a serious practical problem. Below, we sketch a solution applicable to terms from $\mathcal{B}$. For reasons of space, we postpone a full exposition to a future paper; interested readers may consult Appendix C for a dense formal overview.

Suppose that we have a PA term $\alpha_1 \cdots \alpha_k \beta_{k+1} \cdots \beta_{k+l}$ from $\mathcal{B}$ (we juxtapose instead of writing $\boxdot$). Because $\boxdot$ is right-associative, at run-time, the PA indexed j *must* communicate with all PA between i and j before it may communicate with the PA indexed $1 \leq i < j$. If those intermediate PA are independent of j, however, their communication is effectively redundant and should be avoided for the sake of performance: at run-time, j *should* communicate directly with i "skipping" the PA between them. One can formally model this skipping as *reordering* the original PA term: i moves as far as possible to the right, while j moves as far as possible to the left, until $i = j-1$. Although generally not semantics-preserving (because $\boxdot$ is not associative), one can prove that if masters initiate communication with slaves and never the other way around (i.e., $1 \leq i \leq k$ and $k+1 \leq j \leq k+l$), the corresponding reordering of $\alpha_1 \cdots \alpha_k \beta_{k+1} \cdots \beta_{k+l}$ is always bisimilar to the original, just as in the purely distributed approach. Moreover, one can prove that $\mathcal{B}$ is closed under such reordering. Thus, as long as masters initiate communication—a trivial constraint—*not* imposing a particular communication order is still semantics-preserving.

6 Related Work & Conclusion

Related work. Closest to ours is the work on splitting connectors into (a)synchronous regions for better performance. Proença developed the first implementation based on these ideas, demonstrated its merit through benchmarks, and invented an automaton model—*behavioral automata*—to reason about split connectors in his PhD thesis and associated publications [19,20,21]. Furthermore, Clarke and Proença explored connector splitting in the context of the connector coloring semantics [8]. They discovered that the standard version of that semantics has undesirable properties in the context of splitting: some split connectors that intuitively *should* be equivalent to the original connector are not equivalent under the standard version. To address this problem, Clarke and Proença propose a new variant—*partial connector coloring*—which allows one to better model locality and independencies between different parts of a connector. Recently, Jongmans et al. studied a formal justification of connector splitting in a process algebraic setting [14]. Although, as shown in Section 4, one can use the notion of (a)synchronous regions to apply our results to code generation for connectors, our results go beyond that. (They can, for instance, also be applied to code generation for Web service proxies in Reo-based orchestrations [15].)

Also related to the work presented in this paper is the work of Kokash et al. on *action constraint automata* (ACA) [17]. Kokash et al. argue that ordinary port/constraint automata describe the behavior of Reo connectors too coarsely, which makes it impossible to express certain fine parallel behavior. In contrast, ACA have more flexible transition labels which, for instance, allow one to explicitly model the start and end of interaction on a particular port (one cannot make this distinction using port/constraint automata). Consequently, ACA better describe the behavior of existing connector implementations (under certain assumptions). However, the increased granularity of ACA comes at the price of substantially larger models. This makes them less suitable for code generation.

Conclusion. Existing approaches to implementing connectors force one to make a choice between high throughput (at the cost of parallelism) and maximal parallelism (at the cost of throughput). In this paper, we proposed a formal basis to support a solution for this problem. We found and formalized a middle ground between those approaches by defining a new product operator on port automata (PA) and by showing that in all practically relevant cases (with respect to code generation for connectors), one can use this new operator instead of the existing one to get both high throughput and maximal parallelism in a semantics-preserving way.

Although we developed our results for PA, they generalize straightforwardly to the more powerful constraint automata (CA) [4]: the problem dealt with in this paper is essentially about synchronization, while the actual data exchanged play no role. More concretely, the premises of rules WkAgr and StAgr do not change when defining $\boxtimes$ and $\boxdot$ for CA. Thus, whenever those rules are applicable to PA transitions, they are applicable also to the corresponding CA transitions. Although the conclusions of WkAgr and StAgr, in contrast, change when defining $\boxtimes$ and $\boxdot$ for CA (because CA have richer transition labels), those changes are exactly the same for both WkAgr and StAgr. Thus, whenever those rules are both applicable, they yield exactly the same composite transition, as in the PA case. Consequently, all our proofs directly carry over from PA to CA.

While inspired by Reo, our results apply to every language whose programs can be described by automata satisfying the characterizations in Section 4. For instance, a possible application of our results outside Reo is *projection* in choreography languages [5,6,9,10,7,11]. A projection maps a global protocol specification among k parties, called *choreography*, to k local specifications of per-party observable behavior, called *contracts* [5,6] (or *peers* [9,10] or *end-point processes* [7,11]). The challenge is to project such that the collective behavior of the resulting contracts *conforms* with the projected choreography. Interestingly, for some choreographies, without adding extra communication actions to their original specifications, no projection to contracts exists that satisfies the conformance requirement. The theory presented in this paper constitutes a step in a process that may alleviate this problem by automatically inferring which communication actions need be added to otherwise unprojectable choreographies. Below, we make a first sketch.

Choreographies are commonly formally modeled as *labeled transition systems* (LTS) or automata. To compute a projection of a choreography involving k parties, we take such an LTS as our starting point (i.e., our approach builds on top of existing choreography models). If this LTS is finite, we translate it to a *choreography* PA (by mapping transition labels in the LTS to ports).[8] Afterward, we *decompose* the resulting "big" PA into a number of "small" PA [3]. Essentially, by recovering the internal structure of the big PA, this step reveals the previously "hidden" communication actions necessary to make the original choreography projectable. Next, we recombine the small PA into a number of *contract* PA such that for each of those PA, its input ports represent communication actions of only one party. To do this, we first apply the theory of masters, slaves, and (a)synchronous regions for computing a number of "medium" PA from the small PA using $\boxtimes$ (see Section 4). Subsequently, we iteratively compute $\boxtimes$ of every two medium PA whose input ports belong to the same party. Finally, we construct a number of sets of PA, each of which contains: (i) a contract PA resulting from the previous step and (ii) a number of Fifo PA such that the output port of every Fifo PA is the input port of the contract PA. Those Fifo PA essentially represent incoming message buffers of parties.

Generally, the sketched process yields l sets of PA. We conjecture that, with some extra steps skipped here for simplicity, $l = k$: we have a PA set for every party. Every such as PA set can then be compiled into the implementation of a party. Communication between PA of different sets (i.e., between different parties) has to satisfy only the local synchronization requirements imposed by $\boxdot$, which can be done relatively efficiently. The previous process is applicable also to choreographies represented as UML sequence diagrams using a translation by Meng et al. [18].

References

1. Farhad Arbab. Reo: a channel-based coordination model for component composition. *MSCS*, 14(3):329–366, 2004.
2. Farhad Arbab. Puff, The Magic Protocol. In *Talcott Festschrift*, volume 7000 of *LNCS*, pages 169–206. Springer, 2011.
3. Farhad Arbab, Christel Baier, Frank De Boer, Jan Rutten, and Marjan Sirjani. Synthesis of Reo Circuits for Implementation of Component-Connector Automata Specifications. In *Proceedings of COORDINATION 2005*, volume 3454 of *LNCS*, pages 236–251. Springer, 2005.
4. Christel Baier, Marjan Sirjani, Farhad Arbab, and Jan Rutten. Modeling component connectors in Reo by constraint automata. *SCP*, 61(2):75–113, 2006.
5. Mario Bravetti and Gianluigi Zavattaro. Towards a Unifying Theory for Choreography Conformance and Contract Compliance. In *Proceedings of SC 2007*, volume 4829 of *LNCS*, pages 34–50. Springer, 2007.

[8] If the model assumes synchronous communication, we should also "desynchronize" communication actions while constructing the PA from the LTS (in a semantics-preserving way, under some equivalence).

6. Mario Bravetti and Gianluigi Zavattaro. Contract Compliance and Choreography Conformance in the Presence of Message Queues. In *Proceedings of WS-FM 2008*, volume 5387 of *LNCS*, pages 37–45. Springer, 2009.
7. Marco Carbone, Kohei Honda, and Nobuko Yoshida. Structured Communication-Centered Programming for Web Services. *TOPLAS*, 34(2):8:1–8:78, 2012.
8. Dave Clarke and José Proença. Partial Connector Colouring. In *Proceedings of COORDINATION 2012*, volume 7274 of *LNCS*, pages 59–73. Springer, 2012.
9. Xiang Fu, Tevfik Bultan, and Jianwen Su. Conversation protocols: a formalism for specification and verification of reactive electronic services. *TCS*, 328(1–2):19–37, 2004.
10. Xiang Fu, Tevfik Bultan, and Jianwen Su. Realizability of Conversation Protocols with Message Contents. *IJWSR*, 2(4):68–93, 2005.
11. Kohei Honda, Nobuko Yoshida, and Marco Carbone. Multiparty Asynchronous Session Types. In *Proceedings of POPL 2008*, pages 273–284. ACM, 2008.
12. Sung-Shik Jongmans and Farhad Arbab. Overview of Thirty Semantic Formalisms for Reo. *SACS*, 22(1):201–251, 2012.
13. Sung-Shik Jongmans and Farhad Arbab. Modularizing and Specifying Protocols among Threads. In *Proceedings of PLACES 2012*, volume 109 of *EPTCS*, pages 34–45. CoRR, 2013.
14. Sung-Shik Jongmans, Dave Clarke, and José Proença. A Procedure for Splitting Processes and its Application to Coordination. In *Proceedings of FOCLASA 2012*, volume 91 of *EPTCS*, pages 79–96. CoRR, 2012.
15. Sung-Shik Jongmans, Francesco Santini, Mahdi Sargolzaei, Farhad Arbab, and Hamideh Afsarmanesh. Automatic Code Generation for the Orchestration of Web Services with Reo. In *Proceedings of ESOCC 2012*, volume 7592 of *LNCS*, pages 1–16. Springer, 2012.
16. Christian Koehler and Dave Clarke. Decomposing Port Automata. In *Proceedings of SAC 2009*, pages 1369–1373. ACM, 2009.
17. Natallia Kokash, Behnaz Changizi, and Farhad Arbab. A Semantic Model for Service Composition with Coordination Time Delays. In *Proceedings of ICFEM*, volume 6447 of *LNCS*, pages 106–121. Springer, 2010.
18. Sun Meng, Farhad Arbab, and Christel Baier. Synthesis of Reo circuits from scenario-based interaction specifications. *SCP*, 76(8):651–680, 2011.
19. José Proença, Dave Clarke, Erik De Vink, and Farhad Arbab. Dreams: a framework for distributed synchronous coordination. In *Proceedings of SAC 2012*, pages 1510–1515. ACM, 2012.
20. José Proença. *Synchronous Coordination of Distributed Components*. PhD thesis, Leiden University, 2011.
21. José Proença, Dave Clarke, Erik De Vink, and Farhad Arbab. Decoupled execution of synchronous coordination models via behavioural automata. In *Proceedings of FOCLASA 2011*, volume 58 of *EPTCS*, pages 65–79. CoRR, 2011.

A More Definitions

Definition 13 (Universe of ports). *The universe of ports, denoted by* $\mathbb{P}\text{ORT}$ *and typically ranged over by* p*, is a set.*

Definition 14 (Universe of states). *The universe of states, denoted by* $\mathbb{S}\text{TATE}$ *and typically ranged over by* q*, is a set induced by the following rule:*

$$\frac{q_\alpha \in \mathbb{S}\text{TATE} \;\; \textbf{and} \;\; q_\beta \in \mathbb{S}\text{TATE}}{(q_\alpha \,,\, q_\beta) \in \mathbb{S}\text{TATE}}$$

Definition 15 (Accessor functions on port automata). *The accessor functions on port automata, denoted by* State*,* Port*, and* init*, are functions from* $\mathbb{P}\text{A}$ *to* $\wp(\mathbb{S}\text{TATE})$*,* $\wp(\mathbb{P}\text{ORT})$*, and* $\mathbb{S}\text{TATE}$ *defined as:*

$$\begin{aligned} \mathsf{State}((Q \,,\, \mathcal{P} \,,\, \longrightarrow \,,\, \imath)) &= Q \\ \mathsf{Port}((Q \,,\, \mathcal{P} \,,\, \longrightarrow \,,\, \imath)) &= \mathcal{P} \\ \mathsf{init}((Q \,,\, \mathcal{P} \,,\, \longrightarrow \,,\, \imath)) &= \imath \end{aligned}$$

Definition 16 (Similarity relation). *The similarity relation, denoted by* $\preceq$*, is the relation on* $\mathbb{P}\text{A} \times \wp(\mathbb{S}\text{TATE}^2) \times \mathbb{P}\text{A}$ *defined as:*

$$(Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha) \preceq^R (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \;\; \textbf{iff}$$
$$\left[\begin{array}{c} R \subseteq Q_\alpha \times Q_\beta \;\; \textbf{and} \;\; \mathcal{P}_\alpha = \mathcal{P}_\beta \;\; \textbf{and} \;\; \imath_\alpha \; R \; \imath_\beta \;\; \textbf{and} \\ \left[\left[\begin{array}{c} q_\alpha \xrightarrow{P}_\alpha q'_\alpha \\ \textbf{and} \;\; q_\alpha \; R \; q_\beta \end{array}\right] \;\; \textbf{implies} \;\; \left[\left[\begin{array}{c} q_\beta \xrightarrow{P}_\beta q'_\beta \\ \textbf{and} \;\; q'_\alpha \; R \; q'_\beta \end{array}\right] \;\; \textbf{for some} \;\; q'_\beta\right]\right] \end{array}\right]$$

Definition 17 (Bisimilarity relation). *The bisimilarity relation, denoted by* $\approx$*, is the relation on* $(\mathbb{P}\text{A} \times \wp(\mathbb{S}\text{TATE}^2) \times \mathbb{P}\text{A}) \cup (\mathbb{P}\text{A} \times \mathbb{P}\text{A})$ *defined as:*

$$\begin{aligned} \alpha \approx^R \beta \;\; &\textbf{iff} \;\; \left[\alpha \preceq^R \beta \;\; \textbf{and} \;\; \beta \preceq^{R^{-1}} \alpha\right] \\ \alpha \approx \beta \;\; &\textbf{iff} \;\; \left[\alpha \approx^R \beta \;\; \textbf{for some} \;\; R\right] \end{aligned}$$

B More Results

The following proposition follows directly from Definition 17 of $\approx$: if Alice and Bob are equal, they are bisimilar.

Proposition 3. $\alpha = \beta$ **implies** $\alpha \approx \beta$

The following proposition follows directly from Definition 3 of $\boxtimes$: the ports of Alice and Bob together equal the union of the ports of Alice and Bob individually.

Proposition 4. $\mathsf{Port}(\alpha \boxtimes \beta) = \mathsf{Port}(\alpha) \cup \mathsf{Port}(\beta)$

Also the following proposition follows directly from Definition 3 of $\boxtimes$: the universe of port automata is closed under product, product is associative, and product is commutative.

Proposition 5. $(\mathbb{P}\mathrm{A}\,,\,\boxtimes)$ *is a commutative semigroup:*

1. $\big[\alpha \boxtimes \beta \approx \gamma$ **and** $\gamma \in \mathbb{P}\mathrm{A}\big]$ **for some** γ
2. $(\alpha \boxtimes \beta) \boxtimes \gamma \approx \alpha \boxtimes (\beta \boxtimes \gamma)$
3. $\alpha \boxtimes \beta \approx \beta \boxtimes \alpha$

The following proposition follow directly from Definition 5 of $\boxtimes$: the ports of Alice and Bob together equal the union of the ports of Alice and Bob individually.

Proposition 6. $\mathsf{Port}(\alpha \boxdot \beta) = \mathsf{Port}(\alpha) \cup \mathsf{Port}(\beta)$

Also the following proposition follows directly from Definition 5 of $\boxdot$: the universe of port automata is closed under product and product is associative.

Proposition 7. $(\mathbb{P}\mathrm{A}\,,\,\boxdot)$ *is a commutative magma:*

1. $\big[\alpha \boxdot \beta \approx \gamma$ **and** $\gamma \in \mathbb{P}\mathrm{A}\big]$ **for some** γ
2. $\alpha \boxdot \beta \approx \beta \boxdot \alpha$

The following lemma states that similarity is a congruence. See page 22 for a proof.

Lemma 12.

$$\begin{bmatrix} \alpha \preceq^{R_1} \beta \ \textbf{ and } \ \gamma \preceq^{R_2} \delta \ \textbf{ and} \\ \begin{bmatrix} (q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta) \ \textbf{ iff} \\ \big[q_\alpha\ R_1\ q_\beta \ \textbf{ and } \ q_\gamma\ R_2\ q_\delta\big] \end{bmatrix} \end{bmatrix} \ \textbf{implies} \ \alpha \boxdot \gamma \preceq^{R} \beta \boxdot \delta$$

The following proposition follows directly from Definition 8 of $\asymp$ (plus Definition 17 of $\approx$): if Alice and Bob are independent and Bob and Carol are bisimilar, Alice and Carol are independent.

Proposition 8. $\big[\alpha \asymp \beta$ **and** $\beta \approx \gamma\big]$ **implies** $\alpha \asymp \gamma$

The following corollary generalizes Lemma 7 from 2 to k individuals.

Corollary 2. $\big[\alpha \asymp \beta_1$ **and** $\cdots$ **and** $\alpha \asymp \beta_k\big]$ **implies** $\alpha \asymp (\beta_1 \boxdot \cdots \boxdot \beta_k)$

The following proposition follows directly from Definition 9 of $\mapsto$ (plus Definition 17 of $\approx$): if Bob is a slave of Alice and Alice is bisimilar to Carol, Bob is a slave of Carol.

Proposition 9. $\big[\beta \mapsto \alpha$ **and** $\alpha \approx \gamma\big]$ **implies** $\beta \mapsto \gamma$

The following proposition follows directly from Definition 10 of $\Mapsto$ (plus Definition 17 of $\approx$): if Bob is a conditional slave of Alice and Alice is bisimilar to Carol, Bob is a conditional slave of Carol.

Proposition 10. $\big[\beta \Mapsto \alpha$ **and** $\alpha \approx \gamma\big]$ **implies** $\beta \Mapsto \gamma$

C Reordering Communication, Formally

Lemma 13. $\left[\alpha \Rrightarrow \beta \textbf{ and } \alpha\,,\,\beta \asymp \gamma\right]$ **implies** $\alpha \boxdot (\beta \boxdot \gamma) \approx (\alpha \boxdot \beta) \boxdot \gamma$

Proof. See page 39. $\square$

Lemma 14. $\left[\alpha \Rrightarrow \beta\,,\,\gamma \textbf{ and } \beta \Rrightarrow \alpha\,,\,\gamma\right]$ **implies** $\alpha \boxdot (\beta \boxdot \gamma) \approx \beta \boxdot (\alpha \boxdot \gamma)$

Proof. See page 40. $\square$

Lemma 15. $\left[\alpha \asymp \beta\,,\,\gamma \textbf{ and } \beta \asymp \gamma\right]$ **implies** $\alpha \boxdot (\beta \boxdot \gamma) \approx \beta \boxdot (\alpha \boxdot \gamma)$

Proof. See page 41. $\square$

Lemma 16.

$$\left[1 < j \leq l \textbf{ and } [\alpha]_1^k[\beta]_1^l \in \mathcal{B}\right] \textbf{ implies } \begin{bmatrix} [\alpha]_1^k[\beta]_1^{j-2}(\beta_{j-1}\beta_j)[\beta]_{j+1}^l \in \mathcal{B} \textbf{ and} \\ [\alpha]_1^k[\beta]_1^{j-2}(\beta_{j-1}\beta_j)[\beta]_{j+1}^l \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Proof. See page 42. $\square$

Corollary 3.

$$\left[1 \leq j \leq l \textbf{ and } [\alpha]_1^k[\beta]_1^l \in \mathcal{B}\right] \textbf{ implies } \begin{bmatrix} [\alpha]_1^k([\beta]_1^j)[\beta]_{j+1}^l \in \mathcal{B} \textbf{ and} \\ [\alpha]_1^k([\beta]_1^j)[\beta]_{j+1}^l \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Lemma 17.

$$\left[[\alpha]_1^k[\beta]_1^l \in \mathcal{B} \textbf{ and } \alpha_k \asymp \beta_2\,,\,\ldots\,,\,\beta_l\right] \textbf{ implies } \begin{bmatrix} [\alpha]_1^{k-1}(\alpha_k\beta_1)[\beta]_2^l \in \mathcal{B} \textbf{ and} \\ [\alpha]_1^{k-1}(\alpha_k\beta_1)[\beta]_2^l \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Proof. See page 45. $\square$

Corollary 4.

$$\begin{bmatrix} i \leq k \textbf{ and } [\alpha]_1^k[\beta]_1^l \in \mathcal{B} \\ \textbf{and } \begin{bmatrix} \alpha_{i'} \asymp \beta_2\,,\,\ldots\,,\,\beta_l \\ \textbf{for all} \\ k-i+1 \leq i' \leq k \end{bmatrix} \end{bmatrix} \textbf{ implies } \begin{bmatrix} [\alpha]_1^{k-i}([\alpha]_{k-i+1}^k\beta_1)[\beta]_2^l \in \mathcal{B} \textbf{ and} \\ [\alpha]_1^{k-i}([\alpha]_{k-i+1}^k\beta_1)[\beta]_2^l \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Definition 18 (Move-to-the-left functions). *The move-to-the-left function, denoted by* $\Leftarrow$*, is the function from* $\mathbb{P}\mathrm{A} \times \mathcal{B}$ *to* $\mathbb{P}\mathrm{A}$ *defined by the following equation:*

$$\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = \begin{cases} \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^{j-2}\beta_j\beta_{j-1}[\beta]_{j+1}^l) & \textbf{if } 1 < j \leq l \\ [\alpha]_1^k[\beta]_1^l & \textbf{otherwise} \end{cases}$$

The move-all-to-the-left function, denoted by $\Lleftarrow$*, is the function from* $\wp(\mathbb{P}\mathrm{A}) \times \mathcal{B}$ *to* $\mathbb{P}\mathrm{A}$ *defined by the following equation:*

$$\Lleftarrow(B\,,\,[\alpha]_1^k[\beta]_1^l) = \begin{cases} \Lleftarrow(B - B(1)\,,\,\Leftarrow(B(1)\,,\,[\alpha]_1^k[\beta]_1^l)) & \textbf{if } B \subseteq \{\beta_1\,,\,\ldots\,,\,\beta_l\} \\ [\alpha]_1^k[\beta]_1^l & \textbf{otherwise} \end{cases}$$

Lemma 18.

$$\begin{bmatrix} 1 \leq j \leq l \ \textbf{and} \\ [\alpha]_1^k[\beta]_1^l \in \mathcal{B} \end{bmatrix} \ \textbf{implies} \ \begin{bmatrix} \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k\beta_j[\beta]_1^{j-1}[\beta]_{j+1}^l \\ \textbf{and} \ \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B} \\ \textbf{and} \ \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Proof. See page 47.

Corollary 5.

$$B \subseteq \{\beta_1\,,\,\ldots\,,\,\beta_l\} \ \textbf{implies}$$
$$\begin{bmatrix} \Lleftarrow(B\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k B(|B|)\cdots B(1)[\beta]_1^k{\downarrow}_{\{\beta_1,\ldots,\beta_k\}\setminus B} \\ \textbf{and} \ \Lleftarrow(B\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B} \\ \textbf{and} \ \Lleftarrow(B\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Definition 19 (Move-to-the-right functions). *The move-to-the-right function, denoted by $\Rightarrow$, is the function from $\mathbb{P}_{\mathrm{A}} \times \mathcal{B}$ to $\mathbb{P}_{\mathrm{A}}$ defined by the following equation:*

$$\Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) = \begin{cases} \Rightarrow(\alpha_i\,,\,[\alpha]_1^{i-1}\alpha_{i+1}\alpha_i[\alpha]_{i+2}^k[\beta]_1^l) & \textbf{if} \ 1 \leq i < k \\ [\alpha]_1^k[\beta]_1^l & \textbf{otherwise} \end{cases}$$

The move-all-to-the-right function, denoted by $\Rrightarrow$, is the function from $\wp(\mathbb{P}_{\mathrm{A}}) \times \mathcal{B}$ to $\mathbb{P}_{\mathrm{A}}$ defined by the following equation:

$$\Rrightarrow(A\,,\,[\alpha]_1^k[\beta]_1^l) = \begin{cases} \Rrightarrow(A - A(1)\,,\,\Rightarrow(A(1)\,,\,[\alpha]_1^k[\beta]_1^l)) & \textbf{if} \ A \subseteq \{\alpha_1\,,\,\ldots\,,\,\alpha_k\} \\ [\alpha]_1^k[\beta]_1^l & \textbf{otherwise} \end{cases}$$

Lemma 19.

$$\begin{bmatrix} 1 \leq i \leq k \ \textbf{and} \\ [\alpha]_1^k[\beta]_1^l \in \mathcal{B} \end{bmatrix} \ \textbf{implies} \ \begin{bmatrix} \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^{i-1}[\alpha]_{i+1}^k\alpha_i[\beta]_1^l \\ \textbf{and} \ \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B} \\ \textbf{and} \ \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Proof. See page 50.

Corollary 6.

$$A \subseteq \{\alpha_1\,,\,\ldots\,,\,\alpha_k\} \ \textbf{implies}$$
$$\begin{bmatrix} \Rrightarrow(A\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k{\downarrow}_{\{\alpha_1,\ldots,\alpha_k\}\setminus A} A(|A|)\cdots A(1)[\beta]_1^l \\ \textbf{and} \ \Rrightarrow(A\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B} \\ \textbf{and} \ \Rrightarrow(A\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Definition 20 (Role functions). *The role functions, denoted by* Slave *and* Master*, are functions from $\mathbb{P}_{\mathrm{A}} \times \wp(\mathbb{P}_{\mathrm{A}})$ to $\wp(\mathbb{P}_{\mathrm{A}})$ defined by the following equations:*

$$\begin{aligned} \mathsf{Slave}(\beta\,,\,A) &= \{\alpha \mid \alpha \in A \ \textbf{and} \ \alpha \mapsto \beta\} \\ \mathsf{Master}(\alpha\,,\,B) &= \{\beta \mid \beta \in B \ \textbf{and} \ \alpha \mapsto \beta\} \end{aligned}$$

Proposition 11. $\mathsf{Slave}(\beta\,,\,A) \subseteq A$ **and** $\mathsf{Master}(\alpha\,,\,B) \subseteq B$

Definition 21 (Reorder function). *The reorder function, denoted by* $\Downarrow$*, is the function from* $\mathbb{P}_{\mathrm{A}} \times \mathcal{B}$ *to* $\mathbb{P}_{\mathrm{A}}$ *defined by the following equation:*

$$\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = \begin{cases} [\tilde{\alpha}]_1^{k-|A|}\left([\tilde{\alpha}]_{k-|A|+1}^k[\tilde{\beta}]_1^{|B|+1}\right)[\tilde{\beta}]_{|B|+2}^l & \textbf{if } 1 \leq j \leq l \\ [\alpha]_1^k[\beta]_1^l & \textbf{otherwise} \end{cases}$$

$$\textbf{for} \begin{bmatrix} A = \mathsf{Slave}(\beta_j\,,\,\{\alpha_1\,,\,\ldots\,,\,\alpha_k\}) \\ B = (\bigcup_{\alpha \in A} \mathsf{Master}(\alpha, \{\beta_1\,,\,\ldots\,,\,\beta_l\})) \setminus \{\beta_j\} \\ [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l = \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l))) \end{bmatrix}$$

Theorem 3.

$$\left[1 \leq j \leq l \;\textbf{ and }\; [\alpha]_1^k[\beta]_1^l \in \mathcal{B}\right] \;\textbf{ implies }\; \begin{bmatrix} \Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B} \;\textbf{ and} \\ \Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l \end{bmatrix}$$

Proof. See page 53.

D Proofs

D.1 Proofs of Section 3

Proof (of Lemma 12). Assume:

(A1) $\alpha \preceq^{R_1} \beta$

(A2) $\gamma \preceq^{R_2} \delta$

(A3) $(q_\alpha\,,\,q_\gamma)\;R\;(q_\beta\,,\,q_\delta)$ **iff** $\left[q_\alpha\;R_1\;q_\beta \;\textbf{ and }\; q_\gamma\;R_2\;q_\delta\right]$

(A4) $\alpha = (Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha)$

(A5) $\beta = (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$

(A6) $\gamma = (Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma)$

(A7) $\delta = (Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta)$

(A8) $\longrightarrow_\dagger$ denotes the smallest relation induced by the rules StAgr, IndepA, and IndepB under α and γ.

(A9) $\longrightarrow_\ddagger$ denotes the smallest relation induced by the rules StAgr, IndepA, and IndepB under β and δ.

Observe:

(Z1) Recall $\alpha \preceq^{R_1} \beta$ from (A1). Then, by applying (A4), conclude $(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} \beta$. Then, by applying (A5), conclude $(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$. Then, by applying Definition 16 of $\preceq$, conclude $\mathcal{P}_\alpha = \mathcal{P}_\beta$. Then, by applying Definition 15 of Port, conclude $\mathsf{Port}((Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha)) = \mathsf{Port}((Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta))$. Then, by applying (A4), conclude $\mathsf{Port}(\alpha) = \mathsf{Port}((Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta))$. Then, by applying (A5), conclude $\mathsf{Port}(\alpha) = \mathsf{Port}(\beta)$.

Z2 Recall $\gamma \preceq^{R_2} \delta$ from A2. Then, by applying A6, conclude $(Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} \delta$. Then, by applying A7, conclude $(Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} (Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta)$. Then, by applying Definition 16 of $\preceq$, conclude $\mathcal{P}_\gamma = \mathcal{P}_\delta$. Then, by applying Definition 15 of Port, conclude $\mathsf{Port}((Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma)) = \mathsf{Port}((Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta))$. Then, by applying A6, conclude $\mathsf{Port}(\gamma) = \mathsf{Port}((Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta))$. Then, by applying A7, conclude $\mathsf{Port}(\gamma) = \mathsf{Port}(\delta)$.

Z3 Recall $\gamma \preceq^{R_2} \delta$ from A2. Then, by introducing A1, conclude $\big[\alpha \preceq^{R_1} \beta$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A4, conclude $\big[(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} \beta$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A5, conclude $\big[(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A6, conclude:

$$\begin{gathered}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)\\ \textbf{and}\;\; (Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} \delta\end{gathered}$$

Then, by applying A7, conclude:

$$\begin{gathered}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)\\ \textbf{and}\;\; (Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} (Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta)\end{gathered}$$

Then, by applying Definition 16 of $\preceq$, conclude $\big[\imath_\alpha \; R \; \imath_\beta$ **and** $\imath_\gamma \; R \; \imath_\delta\big]$. Then, by applying A3, conclude $(\imath_\alpha\,,\,\imath_\gamma) \; R \; (\imath_\beta\,,\,\imath_\delta)$.

Z4 Recall $\gamma \preceq^{R_2} \delta$ from A2. Then, by introducing A1, conclude $\big[\alpha \preceq^{R_1} \beta$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A4, conclude $\big[(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} \beta$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A5, conclude $\big[(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$ **and** $\gamma \preceq^{R_2} \delta\big]$. Then, by applying A6, conclude:

$$\begin{gathered}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)\\ \textbf{and}\;\; (Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} \delta\end{gathered}$$

Then, by applying A7, conclude:

$$\begin{gathered}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)\\ \textbf{and}\;\; (Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} (Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta)\end{gathered}$$

Then, by applying Definition 16 of $\preceq$, conclude $\big[R_1 \subseteq Q_\alpha \times Q_\beta$ **and** $R_2 \subseteq Q_\gamma \times Q_\delta\big]$. Then, by rewriting under ZFC, conclude:

$$\begin{gathered}\big[\big[q_\alpha \; R_1 \; q_\beta \;\; \textbf{implies} \;\; \big[q_\alpha \in Q_\alpha \;\; \textbf{and} \;\; q_\beta \in Q_\beta\big]\big] \;\; \textbf{for all} \;\; q_\alpha\,,\,q_\beta\big]\\ \textbf{and} \;\; \big[\big[q_\gamma \; R_2 \; q_\delta \;\; \textbf{implies} \;\; \big[q_\gamma \in Q_\gamma \;\; \textbf{and} \;\; q_\delta \in Q_\delta\big]\big] \;\; \textbf{for all} \;\; q_\gamma\,,\,q_\delta\big]\end{gathered}$$

Then, by basic rewriting, conclude:

$$\begin{gathered}\big[\big[q_\alpha \; R_1 \; q_\beta \;\; \textbf{and} \;\; q_\gamma \; R_2 \; q_\delta\big] \;\; \textbf{implies} \;\; \begin{bmatrix} q_\alpha \in Q_\alpha \;\; \textbf{and} \;\; q_\beta \in Q_\beta \\ \textbf{and} \;\; q_\gamma \in Q_\gamma \;\; \textbf{and} \;\; q_\delta \in Q_\delta \end{bmatrix}\big]\\ \textbf{for all} \;\; q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\end{gathered}$$

Then, by rewriting under ZFC, conclude:

$$\big[\big[q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\big]\ \textbf{ implies }\ \begin{bmatrix} (q_\alpha\,,\,q_\gamma) \in Q_\alpha \times Q_\gamma \\ \textbf{and }\ (q_\beta\,,\,q_\delta) \in Q_\beta \times Q_\delta \end{bmatrix}\big]$$
$$\textbf{for all }\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta$$

Then, by applying A3, conclude:

$$\big[(q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta)\ \textbf{ implies }\ \begin{bmatrix} (q_\alpha\,,\,q_\gamma) \in Q_\alpha \times Q_\gamma \\ \textbf{and }\ (q_\beta\,,\,q_\delta) \in Q_\beta \times Q_\delta \end{bmatrix}\big]$$
$$\textbf{for all }\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta$$

Then, by rewriting under ZFC, conclude $R \subseteq (Q_\alpha \times Q_\gamma) \times (Q_\beta \times Q_\delta)$.

Reasoning to a generalization, suppose:

$$\big[(q_\alpha\,,\,q_\gamma) \xrightarrow{P}_\dagger (q'_\alpha\,,\,q'_\gamma)\ \textbf{ and }\ (q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta)\big]$$
$$\textbf{for some }\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\,,\,q'_\alpha\,,\,q'_\gamma\,,\,P$$

Then, by applying A3, conclude $\big[(q_\alpha\,,\,q_\gamma) \xrightarrow{P}_\dagger (q'_\alpha\,,\,q'_\gamma)$ **and** $q_\alpha\ R_1\ q_\beta$ **and** $q_\gamma\ R_2\ q_\delta\big]$. Then, by applying A8, conclude:

$$\big[\big[\textsc{StAgr}\text{ applies}\big]\ \textbf{ or }\ \big[\textsc{Indep}\text{A applies}\big]\ \textbf{ or }\ \big[\textsc{Indep}\text{B applies}\big]\big]$$
$$\textbf{and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta$$

Then, by basic rewriting, conclude:

$$\begin{aligned} &\big[\big[\textsc{StAgr}\text{ applies}\big]\ \textbf{ and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\big] \\ \textbf{or }\ &\big[\big[\textsc{Indep}\text{A applies}\big]\ \textbf{ and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\big] \\ \textbf{or }\ &\big[\big[\textsc{Indep}\text{B applies}\big]\ \textbf{ and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\big] \end{aligned}$$

Proceed by case distinction.

– **Case:** $\big[\big[\textsc{StAgr}\text{ applies}\big]$ **and** $q_\alpha\ R_1\ q_\beta$ **and** $q_\gamma\ R_2\ q_\delta\big]$.
  Then, by applying Definition 5 of StAgr, conclude:

$$\begin{bmatrix} P = P_\alpha \cup P_\gamma\ \textbf{ and }\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha\ \textbf{ and }\ q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma \\ \textbf{and }\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \\ \textbf{and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta \end{bmatrix}\ \textbf{ for some }\ P_\alpha\,,\,P_\gamma$$

  Then, by introducing A2, conclude:

$$\gamma \preceq^{R_2} \delta\ \textbf{ and }\ P = P_\alpha \cup P_\gamma\ \textbf{ and }\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha\ \textbf{ and }\ q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma$$
$$\textbf{and }\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma)\ \textbf{ and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta$$

  Then, by applying A6, conclude:

$$(Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} \delta\ \textbf{ and }$$
$$P = P_\alpha \cup P_\gamma\ \textbf{ and }\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha\ \textbf{ and }\ q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma\ \textbf{ and }$$
$$(\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma)\ \textbf{ and }\ q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta$$

Then, by applying (A7), conclude:

$$\begin{array}{c}(Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} (Q_\delta\,,\,\mathcal{P}_\delta\,,\,\longrightarrow_\delta\,,\,\imath_\delta) \text{ and} \\ P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and } q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta \text{ and } q_\gamma\ R_2\ q_\delta\end{array}$$

Then, by applying Definition 16 of $\preceq$, conclude:

$$\left[\begin{bmatrix} q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma \\ \text{and } q_\gamma\ R_2\ q_\delta\end{bmatrix} \text{ implies } \left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \text{and } q'_\gamma\ R_2\ q'_\delta\end{bmatrix} \text{ for some } q'_\delta\right]\right]$$

$$\begin{array}{c}\text{and } P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and } q_\gamma \xrightarrow{P_\gamma}_\gamma q'_\gamma \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta \text{ and } q_\gamma\ R_2\ q_\delta\end{array}$$

Then, by basic rewriting, conclude:

$$\left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \text{and } q'_\gamma\ R_1\ q'_\delta\end{bmatrix} \text{ for some } q'_\delta\right]$$

$$\begin{array}{c}\text{and } P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta\end{array}$$

Then, by introducing (A1), conclude:

$$\alpha \preceq^{R_1} \beta \text{ and } \left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \text{and } q'_\gamma\ R_2\ q'_\delta\end{bmatrix} \text{ for some } q'_\delta\right]$$

$$\begin{array}{c}\text{and } P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta\end{array}$$

Then, by applying (A4), conclude:

$$(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} \beta \text{ and}$$

$$\left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \text{and } q'_\gamma\ R_2\ q'_\delta\end{bmatrix} \text{ for some } q'_\delta\right]$$

$$\begin{array}{c}\text{and } P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta\end{array}$$

Then, by applying (A5), conclude:

$$(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta) \text{ and}$$

$$\left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \text{and } q'_\gamma\ R_2\ q'_\delta\end{bmatrix} \text{ for some } q'_\delta\right]$$

$$\begin{array}{c}\text{and } P = P_\alpha \cup P_\gamma \text{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \text{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\,P_\gamma) \text{ and } q_\alpha\ R_1\ q_\beta\end{array}$$

Then, by applying Definition 16 of $\preceq$, conclude:

$$\left[\begin{bmatrix} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \\ \textbf{and } q_\alpha \; R_1 \; q_\beta \end{bmatrix} \textbf{ implies } \left[\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \\ \textbf{and } q'_\alpha \; R_1 \; q'_\beta \end{bmatrix} \textbf{ for some } q'_\beta\right]\right]$$

$$\left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \textbf{and } q'_\gamma \; R_2 \; q'_\delta \end{bmatrix} \textbf{ for some } q'_\delta\right]$$

$$\textbf{and } P = P_\alpha \cup P_\gamma \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\, P_\gamma) \textbf{ and } q_\alpha \; R_1 \; q_\beta$$

Then, by basic rewriting, conclude:

$$\left[\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \\ \textbf{and } q'_\alpha \; R_1 \; q'_\beta \end{bmatrix} \textbf{ for some } q'_\beta\right] \textbf{ and } \left[\begin{bmatrix} q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \\ \textbf{and } q'_\gamma \; R_2 \; q'_\delta \end{bmatrix} \textbf{ for some } q'_\delta\right]$$
$$\textbf{and } P = P_\alpha \cup P_\gamma \textbf{ and } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\, P_\gamma)$$

Then, by basic rewriting, conclude:

$$\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha \; R_1 \; q'_\beta \textbf{ and} \\ q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \textbf{ and } q'_\gamma \; R_2 \; q'_\delta \textbf{ and} \\ P = P_\alpha \cup P_\gamma \textbf{ and } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\, P_\gamma) \end{bmatrix} \textbf{ for some } q'_\beta\,,\, q'_\delta$$

Then, by applying Z1, conclude:

$$q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha \; R_1 \; q'_\beta \textbf{ and } q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \textbf{ and } q'_\gamma \; R_2 \; q'_\delta \\ \textbf{and } P = P_\alpha \cup P_\gamma \textbf{ and } (\mathsf{Port}(\beta)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\gamma)\,,\, P_\gamma)$$

Then, by applying Z2, conclude:

$$q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha \; R_1 \; q'_\beta \textbf{ and } q_\delta \xrightarrow{P_\gamma}_\delta q'_\delta \textbf{ and } q'_\gamma \; R_2 \; q'_\delta \\ \textbf{and } P = P_\alpha \cup P_\gamma \textbf{ and } (\mathsf{Port}(\beta)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\delta)\,,\, P_\gamma)$$

Then, by applying Definition 5 of StAgr, conclude $\big[[\text{StAgr applies}]$ **and** $q'_\alpha \; R_1 \; q'_\beta$ **and** $q'_\gamma \; R_2 \; q'_\delta\big]$. Then, by applying A9, conclude $\big[(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta)$ **and** $q'_\alpha \; R_1 \; q'_\beta$ **and** $q'_\gamma \; R_2 \; q'_\delta\big]$.

– **Case:** $\big[[\text{IndepA applies}]$ **and** $q_\alpha \; R_1 \; q_\beta$ **and** $q_\gamma \; R_2 \; q_\delta\big]$.
Then, by applying Definition 3 of IndepA, conclude:

$$\begin{bmatrix} P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and} \\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \\ \textbf{and } q_\alpha \; R_1 \; q_\beta \textbf{ and } q_\gamma \; R_2 \; q_\delta \end{bmatrix} \textbf{ for some } P_\alpha$$

Then, by introducing A1, conclude:

$$\alpha \preceq^{R_1} \beta \textbf{ and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and} \\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \\ \textbf{and } q_\alpha \; R_1 \; q_\beta \textbf{ and } q_\gamma \; R_2 \; q_\delta$$

Then, by applying A4, conclude:

$$(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} \beta \textbf{ and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and}$$
$$q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset$$
$$\textbf{and } q_\alpha\ R_1\ q_\beta \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by applying A5, conclude:

$$(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \preceq^{R_1} (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$$
$$\textbf{and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and}$$
$$q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset$$
$$\textbf{and } q_\alpha\ R_1\ q_\beta \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by applying Definition 16 of $\preceq$, conclude:

$$\left[\begin{bmatrix} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \\ \textbf{and } q_\alpha\ R_1\ q_\beta \end{bmatrix} \textbf{ implies } \left[\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \\ \textbf{and } q'_\alpha\ R_1\ q'_\beta \end{bmatrix} \textbf{ for some } q'_\beta\right]\right]$$
$$\textbf{and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and}$$
$$q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset$$
$$\textbf{and } q_\alpha\ R_1\ q_\beta \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by basic rewriting, conclude:

$$\left[\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \\ \textbf{and } q'_\alpha\ R_1\ q'_\beta \end{bmatrix} \textbf{ for some } q'_\beta\right] \textbf{ and}$$
$$P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by basic rewriting, conclude:

$$\left[q_\delta = q'_\delta \textbf{ for some } q'_\delta\right] \textbf{ and } \left[\begin{bmatrix} q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \\ \textbf{and } q'_\alpha\ R_1\ q'_\beta \end{bmatrix} \textbf{ for some } q'_\beta\right]$$
$$\textbf{and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by basic rewriting, conclude:

$$\begin{bmatrix} q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and } P = P_\alpha \\ \textbf{and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta \end{bmatrix} \textbf{ for some } q'_\beta\,,\,q'_\delta$$

Then, by introducing A2, conclude:

$$\gamma \preceq^{R_2} \delta \textbf{ and } q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}$$
$$P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by applying A6, conclude:

$$(Q_\gamma\,,\,\mathcal{P}_\gamma\,,\,\longrightarrow_\gamma\,,\,\imath_\gamma) \preceq^{R_2} \delta \textbf{ and } q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta$$
$$\textbf{and } P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta$$

Then, by applying (A7), conclude:

$$\begin{gathered}(Q_\gamma\,,\, \mathcal{P}_\gamma\,,\, \longrightarrow_\gamma\,,\, \imath_\gamma) \preceq^{R_2} (Q_\delta\,,\, \mathcal{P}_\delta\,,\, \longrightarrow_\delta\,,\, \imath_\delta) \textbf{ and}\\ q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}\\ P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta\end{gathered}$$

Then, by applying Definition 16 of $\preceq$, conclude:

$$\begin{gathered}R_2 \subseteq Q_\gamma \times Q_\delta \textbf{ and } q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}\\ P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta\end{gathered}$$

Then, by rewriting under ZFC, conclude:

$$\begin{gathered}q_\delta \in Q_\delta \textbf{ and } q_\delta = q'_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}\\ P = P_\alpha \textbf{ and } q_\gamma = q'_\gamma \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q_\gamma\ R_2\ q_\delta\end{gathered}$$

Then, by basic rewriting, conclude:

$$\begin{gathered}q_\delta \in Q_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}\\ P = P_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\gamma) = \emptyset \textbf{ and } q'_\gamma\ R_2\ q'_\delta\end{gathered}$$

Then, by applying (Z2), conclude:

$$\begin{gathered}q_\delta \in Q_\delta \textbf{ and } q_\beta \xrightarrow{P_\alpha}_\beta q'_\beta \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and}\\ P = P_\alpha \textbf{ and } P_\alpha \cap \mathsf{Port}(\delta) = \emptyset \textbf{ and } q'_\gamma\ R_2\ q'_\delta\end{gathered}$$

Then, by applying Definition 3 of IndepA, conclude $\big[\big[$IndepA applies$\big]$ **and** $q'_\alpha\ R_1\ q'_\beta$ **and** $q'_\gamma\ R_2\ q'_\delta\big]$. Then, by applying (A9), conclude $\big[(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta)$ **and** $q'_\alpha\ R_1\ q'_\beta$ **and** $q'_\gamma\ R_2\ q'_\delta\big]$.

– **Case:** $\big[\big[$IndepB applies$\big]$ **and** $q_\alpha\ R_1\ q_\beta$ **and** $q_\gamma\ R_2\ q_\delta\big]$. Symmetrically.

Hence, after considering all cases, conclude:

$$\big[(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta) \textbf{ and } q'_\alpha\ R_1\ q'_\beta \textbf{ and } q'_\gamma\ R_2\ q'_\delta\big] \textbf{ for some } q'_\beta\,,\, q'_\delta$$

Then, by applying (A3), conclude $\big[(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta)$ **and** $(q'_\alpha\,,\, q'_\gamma)\ R\ (q'_\beta\,,\, q'_\delta)\big]$. Then, by generalizing the premise, conclude:

$$\begin{gathered}\Big[\begin{bmatrix}(q_\alpha\,,\, q_\gamma) \xrightarrow{P}_\dagger (q'_\alpha\,,\, q'_\gamma)\\ \textbf{and } (q_\alpha\,,\, q_\gamma)\ R\ (q_\beta\,,\, q_\delta)\end{bmatrix} \textbf{ implies } \begin{bmatrix}(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta)\\ \textbf{and } (q'_\alpha\,,\, q'_\gamma)\ R\ (q'_\beta\,,\, q'_\delta)\end{bmatrix}\Big]\\ \textbf{for all } q_\alpha\,,\, q_\beta\,,\, q_\gamma\,,\, q_\delta\,,\, q'_\alpha\,,\, q'_\gamma\,,\, P\end{gathered}$$

Then, by introducing (Z3), conclude:

$$\begin{gathered}(\imath_\alpha\,,\, \imath_\gamma)\ R\ (\imath_\beta\,,\, \imath_\delta) \textbf{ and}\\ \left[\begin{gathered}\Big[\begin{bmatrix}(q_\alpha\,,\, q_\gamma) \xrightarrow{P}_\dagger (q'_\alpha\,,\, q'_\gamma)\\ \textbf{and } (q_\alpha\,,\, q_\gamma)\ R\ (q_\beta\,,\, q_\delta)\end{bmatrix} \textbf{ implies } \begin{bmatrix}(q_\beta\,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta\,,\, q'_\delta)\\ \textbf{and } (q'_\alpha\,,\, q'_\gamma)\ R\ (q'_\beta\,,\, q'_\delta)\end{bmatrix}\Big]\\ \textbf{for all } q_\alpha\,,\, q_\beta\,,\, q_\gamma\,,\, q_\delta\,,\, q'_\alpha\,,\, q'_\gamma\,,\, P\end{gathered}\right]\end{gathered}$$

Then, by introducing Z4, conclude:

$$
\begin{array}{c}
R \subseteq (Q_\alpha \times Q_\gamma) \times (Q_\beta \times Q_\delta) \ \textbf{and} \ (\imath_\alpha \,,\, \imath_\gamma) \ R \ (\imath_\beta \,,\, \imath_\delta) \ \textbf{and} \\
\left[\begin{array}{c}
[\left[\begin{array}{c} (q_\alpha \,,\, q_\gamma) \xrightarrow{P}_\dagger (q'_\alpha \,,\, q'_\gamma) \\ \textbf{and} \ (q_\alpha \,,\, q_\gamma) \ R \ (q_\beta \,,\, q_\delta) \end{array}\right] \ \textbf{implies} \ \left[\begin{array}{c} (q_\beta \,,\, q_\delta) \xrightarrow{P}_\ddagger (q'_\beta \,,\, q'_\delta) \\ \textbf{and} \ (q'_\alpha \,,\, q'_\gamma) \ R \ (q'_\beta \,,\, q'_\delta) \end{array}\right]] \\
\textbf{for all} \ q_\alpha \,,\, q_\beta \,,\, q_\gamma \,,\, q_\delta \,,\, q'_\alpha \,,\, q'_\gamma \,,\, P
\end{array}\right]
\end{array}
$$

Then, by applying Definition 16 of $\preceq$, conclude:

$$
\begin{array}{l}
(Q_\alpha \times Q_\gamma \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\gamma \,,\, \longrightarrow_\dagger \,,\, (\imath_\alpha \,,\, \imath_\gamma)) \\
\ \preceq^R (Q_\beta \times Q_\delta \,,\, \mathcal{P}_\beta \cup \mathcal{P}_\delta \,,\, \longrightarrow_\ddagger \,,\, (\imath_\beta \,,\, \imath_\delta))
\end{array}
$$

Then, by introducing A9, conclude:

$$
\text{A9} \ \textbf{and} \ \begin{array}{l}
(Q_\alpha \times Q_\gamma \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\gamma \,,\, \longrightarrow_\dagger \,,\, (\imath_\alpha \,,\, \imath_\gamma)) \\
\ \preceq^R (Q_\beta \times Q_\delta \,,\, \mathcal{P}_\beta \cup \mathcal{P}_\delta \,,\, \longrightarrow_\ddagger \,,\, (\imath_\beta \,,\, \imath_\delta))
\end{array}
$$

Then, by introducing A8, conclude:

$$
\text{A8} \ \textbf{and} \ \text{A9} \ \textbf{and} \ \begin{array}{l}
(Q_\alpha \times Q_\gamma \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\gamma \,,\, \longrightarrow_\dagger \,,\, (\imath_\alpha \,,\, \imath_\gamma)) \\
\ \preceq^R (Q_\beta \times Q_\delta \,,\, \mathcal{P}_\beta \cup \mathcal{P}_\delta \,,\, \longrightarrow_\ddagger \,,\, (\imath_\beta \,,\, \imath_\delta))
\end{array}
$$

Then, by applying Definition 3 of $\boxdot$, conclude:

$$
\begin{array}{l}
(Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha) \boxdot (Q_\gamma \,,\, \mathcal{P}_\gamma \,,\, \longrightarrow_\gamma \,,\, \imath_\gamma) \\
\ \preceq^R (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \boxdot (Q_\delta \,,\, \mathcal{P}_\delta \,,\, \longrightarrow_\delta \,,\, \imath_\delta)
\end{array}
$$

Then, by applying A4, conclude $\alpha \boxdot (Q_\gamma \,,\, \mathcal{P}_\gamma \,,\, \longrightarrow_\gamma \,,\, \imath_\gamma) \preceq^R (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \boxdot (Q_\delta \,,\, \mathcal{P}_\delta \,,\, \longrightarrow_\delta \,,\, \imath_\delta)$. Then, by applying A6, conclude $\alpha \boxdot \gamma \preceq^R (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \boxdot (Q_\delta \,,\, \mathcal{P}_\delta \,,\, \longrightarrow_\delta \,,\, \imath_\delta)$. Then, by applying A5, conclude $\alpha \boxdot \gamma \preceq^R \beta \boxdot (Q_\delta \,,\, \mathcal{P}_\delta \,,\, \longrightarrow_\delta \,,\, \imath_\delta)$. Then, by applying A7, conclude $\alpha \boxdot \gamma \preceq^R \beta \boxdot \delta$. □

Proof (of Lemma 2). Assume:

A1 $\alpha \approx \beta$

A2 $\gamma \approx \delta$

Recall $\gamma \approx \delta$ from A2. Then, by introducing A1, conclude $\big[\alpha \approx \beta \ \textbf{and} \ \gamma \approx \delta\big]$. Then, by applying Definition 17 of $\approx$, conclude:

$$
\big[\alpha \approx^{R_1} \beta \ \textbf{and} \ \gamma \approx^{R_2} \delta\big] \ \textbf{for some} \ R_1 \,,\, R_2
$$

Then, by basic rewriting, conclude:

$$
\begin{array}{c}
[[\left[\begin{array}{c} (q_\alpha \,,\, q_\gamma) \ R \ (q_\beta \,,\, q_\delta) \ \textbf{iff} \\ [q_\alpha \ R_1 \ q_\beta \ \textbf{and} \ q_\gamma \ R_2 \ q_\delta] \end{array}\right] \ \textbf{for all} \ q_\alpha \,,\, q_\beta \,,\, q_\gamma \,,\, q_\delta] \ \textbf{for some} \ R] \\
\textbf{and} \ \alpha \approx^{R_1} \beta \ \textbf{and} \ \gamma \approx^{R_2} \delta
\end{array}
$$

Then, by basic rewriting, conclude:

$$\left[\begin{array}{c}\left[\left[\begin{array}{c}(q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta)\ \textbf{iff}\\ \left[q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\right]\end{array}\right]\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\right]\\ \textbf{and}\ \alpha \approx^{R_1} \beta\ \textbf{ and }\ \gamma \approx^{R_2} \delta\end{array}\right]\ \textbf{for some}\ R$$

Then, by rewriting under ZFC, conclude:

$$\begin{array}{c}\left[\left[\begin{array}{c}(q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta)\ \textbf{iff}\\ \left[q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\right]\end{array}\right]\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\right]\ \textbf{and}\\ \left[\left[\begin{array}{c}(q_\beta\,,\,q_\delta)\ R^{-1}\ (q_\alpha\,,\,q_\gamma)\ \textbf{iff}\\ \left[q_\beta\ R_1^{-1}\ q_\alpha\ \textbf{ and }\ q_\delta\ R_2^{-1}\ q_\gamma\right]\end{array}\right]\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\right]\\ \textbf{and}\ \alpha \approx^{R_1} \beta\ \textbf{ and }\ \gamma \approx^{R_2} \delta\end{array}$$

Then, by applying Definition 17 of $\approx$, conclude:

$$\begin{array}{c}\left[\left[\begin{array}{c}(q_\alpha\,,\,q_\gamma)\ R\ (q_\beta\,,\,q_\delta)\ \textbf{iff}\\ \left[q_\alpha\ R_1\ q_\beta\ \textbf{ and }\ q_\gamma\ R_2\ q_\delta\right]\end{array}\right]\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\right]\ \textbf{and}\\ \left[\left[\begin{array}{c}(q_\beta\,,\,q_\delta)\ R^{-1}\ (q_\alpha\,,\,q_\gamma)\ \textbf{iff}\\ \left[q_\beta\ R_1^{-1}\ q_\alpha\ \textbf{ and }\ q_\delta\ R_2^{-1}\ q_\gamma\right]\end{array}\right]\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q_\gamma\,,\,q_\delta\right]\\ \textbf{and}\ \alpha \preceq^{R_1} \beta\ \textbf{ and }\ \beta \preceq^{R_1^{-1}} \alpha\ \textbf{ and }\ \gamma \preceq^{R_2} \delta\ \textbf{ and }\ \delta \preceq^{R_2^{-1}} \gamma\end{array}$$

Then, by applying Lemma 12, conclude $\left[\alpha \boxdot \gamma \preceq^{R} \beta \boxdot \delta\ \textbf{ and }\ \beta \boxdot \delta \preceq^{R^{-1}} \alpha \boxdot \gamma\right]$. Then, by applying Definition 17 of $\approx$, conclude $\alpha \boxdot \gamma \approx^{R} \beta \boxdot \delta$. Then, by applying Definition 17 of $\approx$, conclude $\alpha \boxdot \gamma \approx \beta \boxdot \delta$. □

D.2 Proofs of Section 3

Proof (of Lemma 3). Suppose $(\mathcal{P}_\alpha\,,\,P_\alpha) \blacklozenge (\mathcal{P}_\beta\,,\,P_\beta)$. Then, by applying Definition 4 of $\blacklozenge$, conclude:

$$P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ \left[\begin{array}{c}P_\alpha = \mathcal{P}_\alpha \cap P_\beta\ \textbf{ or }\ P_\beta = \mathcal{P}_\beta \cap P_\alpha\\ \textbf{or}\ \mathcal{P}_\alpha \cap P_\beta = \emptyset = \mathcal{P}_\beta \cap P_\alpha\end{array}\right]$$

Then, by rewriting under ZFC, conclude:

$$\begin{array}{c}\left[P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ P_\alpha = \mathcal{P}_\alpha \cap P_\beta\right]\ \textbf{or}\\ \left[P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ P_\beta = \mathcal{P}_\beta \cap P_\alpha\right]\ \textbf{or}\\ \left[P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ \mathcal{P}_\alpha \cap P_\beta = \emptyset = \mathcal{P}_\beta \cap P_\alpha\right]\end{array}$$

Proceed by case distinction.

- **Case:** $\left[P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ P_\alpha = \mathcal{P}_\alpha \cap P_\beta\right]$.
  Then, by rewriting under ZFC, conclude $\left[P_\alpha \subseteq P_\beta\ \textbf{ and }\ P_\alpha \subseteq \mathcal{P}_\alpha\ \textbf{ and }\ P_\beta \subseteq \mathcal{P}_\beta\ \textbf{ and }\ P_\alpha = \mathcal{P}_\alpha \cap P_\beta\right]$. Then, by rewriting under ZFC, conclude

$\big[P_\alpha \subseteq \mathcal{P}_\beta$ **and** $P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $P_\alpha = \mathcal{P}_\alpha \cap P_\beta\big]$. Then, by rewriting under ZFC, conclude $\big[P_\alpha = \mathcal{P}_\beta \cap P_\alpha$ **and** $P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $P_\alpha = \mathcal{P}_\alpha \cap P_\beta\big]$. Then, by basic rewriting, conclude $\big[P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $\mathcal{P}_\alpha \cap P_\beta = \mathcal{P}_\beta \cap P_\alpha\big]$.

- **Case:** $\big[P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $P_\beta = \mathcal{P}_\beta \cap P_\alpha\big]$. Symmetrically.
- **Case:** $\big[P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $\mathcal{P}_\alpha \cap P_\beta = \emptyset = \mathcal{P}_\beta \cap P_\alpha\big]$.
  Then, by rewriting under ZFC, conclude $\big[P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $\mathcal{P}_\alpha \cap P_\beta = \mathcal{P}_\beta \cap P_\alpha\big]$.

Hence, after considering all cases, conclude $\big[P_\alpha \subseteq \mathcal{P}_\alpha$ **and** $P_\beta \subseteq \mathcal{P}_\beta$ **and** $\mathcal{P}_\alpha \cap P_\beta = \mathcal{P}_\beta \cap P_\alpha\big]$. Then, by applying Definition 2 of $\Diamond$, conclude $(\mathcal{P}_\alpha\,,\, P_\alpha) \Diamond (\mathcal{P}_\beta\,,\, P_\beta)$. □

Proof (of Lemma 4). Assume:

(A1) $\alpha = (Q_\alpha\,,\, \mathcal{P}_\alpha\,,\, \longrightarrow_\alpha\,,\, \imath_\alpha)$

(A2) $\beta = (Q_\beta\,,\, \mathcal{P}_\beta\,,\, \longrightarrow_\beta\,,\, \imath_\beta)$

(A3) $\longrightarrow_\boxtimes$ denotes the smallest relation induced by the rules WkAgr, IndepA, and IndepB under α and β.

(A4) $\longrightarrow_\boxdot$ denotes the smallest relation induced by the rules StAgr, IndepA, and IndepB under α and β.

Reasoning to a generalization, suppose:

$$(q_\alpha\,,\, q_\beta) \xrightarrow{P}_\boxdot (q'_\alpha\,,\, q'_\beta) \ \textbf{for some}\ q_\alpha\,,\, q_\beta\,,\, q'_\alpha\,,\, q'_\beta\,,\, P$$

Then, by applying (A4), $\big[\big[$StAgr applies$\big]$ **or** $\big[$IndepA applies$\big]$ **or** $\big[$IndepB applies$\big]\big]$. Proceed by case distinction.

- **Case:** $\big[$StAgr applies$\big]$.
  Then, by applying Definition 3 of StAgr, conclude:

$$\begin{bmatrix} P = P_\alpha \cup P_\beta \ \textbf{and}\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \ \textbf{and}\ q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and}\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta) \end{bmatrix} \ \textbf{for some}\ P_\alpha\,,\, P_\beta$$

  Then, by applying Lemma 3, conclude

$$\begin{gathered} P = P_\alpha \cup P_\beta \ \textbf{and}\ q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \ \textbf{and}\ q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and}\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \Diamond (\mathsf{Port}(\beta)\,,\, P_\beta) \end{gathered}$$

  Then, by applying Definition 5 of WkAgr, conclude $\big[P = P_\alpha \cup P_\beta$ **and** $\big[$WkAgr applies$\big]\big]$. Then, by applying (A3), conclude $\big[P = P_\alpha \cup P_\beta$ **and** $(q_\alpha\,,\, q_\beta) \xrightarrow{P_\alpha \cup P_\beta}_\boxtimes (q'_\alpha\,,\, q'_\beta)\big]$. Then, by basic rewriting, conclude $(q_\alpha\,,\, q_\beta) \xrightarrow{P}_\boxtimes (q'_\alpha\,,\, q'_\beta)$.

- **Case:** [Indep A applies].
  Then, by applying (A3), conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta)$.
- **Case:** [Indep B applies].
  Then, by applying (A3), conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta)$.

Hence, after considering all cases, conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta)$. Then, by generalizing the premise, conclude:

$$\big[(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta) \ \textbf{implies}\ (q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta)\big] \\ \textbf{for all}\ q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P$$

Then, by rewriting under ZFC, conclude $\longrightarrow_{\boxdot} \subseteq \longrightarrow_{\boxtimes}$. Then, by introducing (A4), conclude [(A4) **and** $\longrightarrow_{\boxdot} \subseteq \longrightarrow_{\boxtimes}$]. Then, by introducing (A3), conclude [(A3) **and** (A4) **and** $\longrightarrow_{\boxdot} \subseteq \longrightarrow_{\boxtimes}$]. Then, by applying Definition 6 of $\sqsubseteq$, conclude:

$$\text{(A3)}\ \textbf{and}\ \text{(A4)}\ \textbf{and}\ \begin{array}{l}(Q_\alpha \times Q_\beta\,,\,\mathcal{P}_\alpha \cup \mathcal{P}_\beta\,,\,\longrightarrow_{\boxdot}\,,\,(\imath_\alpha\,,\,\imath_\beta)) \\ \sqsubseteq (Q_\alpha \times Q_\beta\,,\,\mathcal{P}_\alpha \cup \mathcal{P}_\beta\,,\,\longrightarrow_{\boxtimes}\,,\,(\imath_\alpha\,,\,\imath_\beta))\end{array}$$

Then, by applying Definition 5 of $\boxdot$, conclude:

$$\text{(A3)}\ \textbf{and}\ \begin{array}{l}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \boxdot (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta) \\ \sqsubseteq (Q_\alpha \times Q_\beta\,,\,\mathcal{P}_\alpha \cup \mathcal{P}_\beta\,,\,\longrightarrow_{\boxtimes}\,,\,(\imath_\alpha\,,\,\imath_\beta))\end{array}$$

Then, by applying Definition 3 of $\boxtimes$, conclude:

$$\begin{array}{l}(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \boxdot (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta) \\ \sqsubseteq (Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \boxtimes (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)\end{array}$$

Then, by applying (A1), conclude $\alpha \boxdot (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta) \sqsubseteq \alpha \boxtimes (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$. Then, by applying (A2), conclude $\alpha \boxdot \beta \sqsubseteq \alpha \boxtimes \beta$. $\square$

Proof (of Lemma 5). Assume:

(A1) $\alpha \diamondsuit \beta$

(A2) $\alpha = (Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha)$

(A3) $\beta = (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$

(A4) $\longrightarrow_{\boxtimes}$ denotes the smallest relation induced by the rules WkAgr, Indep A, and Indep B under α and β.

(A5) $\longrightarrow_{\boxdot}$ denotes the smallest relation induced by the rules StAgr, Indep A, and Indep B under α and β.

Reasoning to a generalization, suppose:

$$(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta)\ \textbf{for some}\ q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P$$

Then, by applying (A4), [[WkAgr applies] **or** [Indep A applies] **or** [Indep B applies]]. Proceed by case distinction.

- **Case:** $\big[$WkAgr applies$\big]$.
  Then, by applying Definition 3 of WkAgr, conclude:

$$\left[\begin{array}{c} P = P_\alpha \cup P_\beta \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\,P_\alpha) \between (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}\right] \textbf{ for some } P_\alpha\,,\,P_\beta$$

  Then, by introducing (A1), conclude:

$$\alpha \,\blacklozenge\, \beta \textbf{ and } \left[\begin{array}{c} P = P_\alpha \cup P_\beta \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\,P_\alpha) \between (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}\right]$$

  Then, by applying Definition 7 of $\blacklozenge$, conclude:

$$\left[\begin{array}{c} \left[\begin{array}{c} \left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \between (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}\right] \\ \textbf{implies } (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}\right] \\ \textbf{for all } q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P_\alpha\,,\,P_\beta \end{array}\right]$$
$$\textbf{and}$$
$$\left[\begin{array}{c} P = P_\alpha \cup P_\beta \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\,P_\alpha) \between (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}\right]$$

  Then, by rewriting under ZFC, conclude:

$$\begin{array}{c} P = P_\alpha \cup P_\beta \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta) \end{array}$$

  Then, by applying Definition 5 of StAgr, conclude $\big[P = P_\alpha \cup P_\beta$ **and** $\big[$ StAgr applies$\big]\big]$. Then, by applying (A5), conclude $\big[P = P_\alpha \cup P_\beta$ **and** $(q_\alpha\,,\,q_\beta) \xrightarrow{P_\alpha \cup P_\beta}_{\boxdot} (q'_\alpha\,,\,q'_\beta)\big]$. Then, by basic rewriting, conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta)$.

- **Case:** $\big[$IndepA applies$\big]$.
  Then, by applying (A5), conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta)$.

- **Case:** $\big[$IndepB applies$\big]$.
  Then, by applying (A5), conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta)$.

Hence, after considering all cases, conclude $(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta)$. Then, by generalizing the premise, conclude:

$$\begin{array}{c} \big[(q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxtimes} (q'_\alpha\,,\,q'_\beta) \textbf{ implies } (q_\alpha\,,\,q_\beta) \xrightarrow{P}_{\boxdot} (q'_\alpha\,,\,q'_\beta)\big] \\ \textbf{for all } q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P \end{array}$$

Then, by rewriting under ZFC, conclude $\longrightarrow_{\boxtimes} \subseteq \longrightarrow_{\boxdot}$. Then, by introducing (A5), conclude $\big[$(A5) **and** $\longrightarrow_{\boxtimes} \subseteq \longrightarrow_{\boxdot}\big]$. Then, by introducing (A4), conclude $\big[$(A4) **and** (A5) **and** $\longrightarrow_{\boxtimes} \subseteq \longrightarrow_{\boxdot}\big]$. Then, by applying Definition 6 of $\sqsubseteq$, conclude:

$$\text{(A4) } \textbf{and} \text{ (A5) } \textbf{and} \begin{array}{l}(Q_\alpha \times Q_\beta \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\beta \,,\, \longrightarrow_{\boxtimes} \,,\, (\imath_\alpha \,,\, \imath_\beta)) \\ \sqsubseteq (Q_\alpha \times Q_\beta \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\beta \,,\, \longrightarrow_{\boxdot} \,,\, (\imath_\alpha \,,\, \imath_\beta))\end{array}$$

Then, by applying Definition 3 of $\boxtimes$, conclude:

$$\text{(A5) } \textbf{and} \begin{array}{l}(Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha) \boxtimes (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \\ \sqsubseteq (Q_\alpha \times Q_\beta \,,\, \mathcal{P}_\alpha \cup \mathcal{P}_\beta \,,\, \longrightarrow_{\boxdot} \,,\, (\imath_\alpha \,,\, \imath_\beta))\end{array}$$

Then, by applying Definition 5 of $\boxdot$, conclude:

$$\begin{array}{l}(Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha) \boxtimes (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \\ \sqsubseteq (Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha) \boxdot (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta)\end{array}$$

Then, by applying (A2), conclude $\alpha \boxtimes (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta) \sqsubseteq \alpha \boxdot (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta)$. Then, by applying (A3), conclude $\alpha \boxtimes \beta \sqsubseteq \alpha \boxdot \beta$. □

Proof (of Theorem 1). Recall $\alpha \boxtimes \beta \sqsubseteq \alpha \boxdot \beta$ from Lemma 5. Then, by introducing Lemma 4, conclude $\big[\alpha \boxdot \beta \sqsubseteq \alpha \boxtimes \beta$ **and** $\alpha \boxtimes \beta \sqsubseteq \alpha \boxdot \beta\big]$. Then, by applying Proposition 1, conclude $\alpha \boxdot \beta = \alpha \boxtimes \beta$. □

D.3 Proofs of Section 4

Proof (of Lemma 6). Assume:

(A1) $\alpha \asymp \beta$.

(A2) $\alpha = (Q_\alpha \,,\, \mathcal{P}_\alpha \,,\, \longrightarrow_\alpha \,,\, \imath_\alpha)$

(A3) $\beta = (Q_\beta \,,\, \mathcal{P}_\beta \,,\, \longrightarrow_\beta \,,\, \imath_\beta)$

Reasoning to a generalization, suppose:

$$(\mathsf{Port}(\alpha) \,,\, P_\alpha) \between (\mathsf{Port}(\beta) \,,\, P_\beta) \ \textbf{for some}\ P_\alpha \,,\, P_\beta$$

Then, by introducing (A1), conclude $\big[\alpha \asymp \beta$ **and** $(\mathsf{Port}(\alpha) \,,\, P_\alpha) \between (\mathsf{Port}(\beta) \,,\, P_\beta)\big]$. Then, by applying Definition 8 of $\asymp$, conclude $\big[\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset$ **and** $(\mathsf{Port}(\alpha) \,,\, P_\alpha) \between (\mathsf{Port}(\beta) \,,\, P_\beta)\big]$. Then, by applying Definition 2 of $\between$, conclude:

$$\begin{array}{c}\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset \ \textbf{and}\ P_\alpha \subseteq \mathsf{Port}(\alpha) \ \textbf{and}\ P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and}\ \mathsf{Port}(\alpha) \cap P_\beta = \mathsf{Port}(\beta) \cap P_\alpha\end{array}$$

Then, by rewriting under ZFC, conclude:

$$\begin{array}{c}\mathsf{Port}(\alpha) \cap P_\beta = \emptyset \ \textbf{and}\ P_\alpha \subseteq \mathsf{Port}(\alpha) \ \textbf{and}\ P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and}\ \mathsf{Port}(\alpha) \cap P_\beta = \mathsf{Port}(\beta) \cap P_\alpha\end{array}$$

Then, by rewriting under ZFC, conclude $\big[P_\alpha \subseteq \mathsf{Port}(\alpha)$ **and** $P_\beta \subseteq \mathsf{Port}(\beta)$ **and** $\mathsf{Port}(\alpha) \cap P_\beta = \emptyset = \mathsf{Port}(\beta) \cap P_\alpha\big]$. Then, by applying Definition 4 of $\blacklozenge$, conclude $(\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta)$. Then, by generalizing the premise, conclude:

$$\left[\begin{array}{c}(\mathsf{Port}(\alpha)\,,\,P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\,P_\beta) \\ \textbf{implies}\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right] \textbf{ for all } P_\alpha\,,\,P_\beta$$

Then, by basic rewriting, conclude:

$$\left[\begin{array}{l}\left[\begin{array}{c}q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right] \\ \textbf{implies}\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right] \textbf{ for all } q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P_\alpha\,,\,P_\beta$$

Then, by applying Definition 7 of $\langle\!\blacklozenge\rangle$, conclude $(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \langle\!\blacklozenge\rangle (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$. Then, by applying (A2), conclude $\alpha \langle\!\blacklozenge\rangle (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$. Then, by applying (A3), conclude $\alpha \langle\!\blacklozenge\rangle \beta$. ☐

Proof (of Lemma 7). Suppose $\alpha \asymp \beta\,,\,\gamma$. Then, by applying Definition 8 of $\asymp$, conclude $\big[\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset$ **and** $\mathsf{Port}(\alpha) \cap \mathsf{Port}(\gamma) = \emptyset\big]$. Then, by rewriting under ZFC, conclude $\mathsf{Port}(\alpha) \cap (\mathsf{Port}(\beta) \cup \mathsf{Port}(\gamma)) = \emptyset$. Then, by applying Proposition 6, conclude $\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta \boxdot \gamma) = \emptyset$. Then, by applying Definition 8 of $\asymp$, conclude $\alpha \asymp \beta \boxdot \gamma$. ☐

Proof (of Lemma 8). Assume:

(A1) $\alpha \mapsto \beta$

(A2) $\alpha = (Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha)$

(A3) $\beta = (Q_\beta\,,\,\mathcal{P}_\beta\,,\,\longrightarrow_\beta\,,\,\imath_\beta)$

Reasoning to a generalization, suppose:

$$\left[\begin{array}{c}q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right] \textbf{ for some } q_\alpha\,,\,q_\beta\,,\,q'_\alpha\,,\,q'_\beta\,,\,P_\alpha\,,\,P_\beta$$

Then, by introducing (A1), conclude:

$$\alpha \mapsto \beta \textbf{ and } \left[\begin{array}{c}q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right]$$

Then, by applying (A2), conclude:

$$(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \mapsto \beta \textbf{ and } \left[\begin{array}{c}q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\,P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\,P_\beta)\end{array}\right]$$

Then, applying Definition 9 of $\mapsto$, conclude:

$$\left[\begin{array}{c}\left[\begin{array}{c}\left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta) \neq \emptyset \end{array}\right] \\ \textbf{implies } P_\alpha \subseteq \mathsf{Port}(\beta) \end{array}\right] \\ \textbf{for all } q_\alpha \,,\, q'_\alpha \,,\, P_\alpha \end{array}\right] \textbf{ and } \left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right]$$

Then, by basic rewriting, conclude:

$$\left[\begin{array}{c}\left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta) \neq \emptyset \end{array}\right] \\ \textbf{implies } P_\alpha \subseteq \mathsf{Port}(\beta) \end{array}\right] \textbf{ and } \left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right]$$

Then, by basic rewriting, conclude:

$$\left[\begin{array}{c} (\textbf{not } q_\alpha \xrightarrow{P_\alpha} q'_\alpha) \\ \textbf{or } P_\alpha \cap \mathsf{Port}(\beta) = \emptyset \\ \textbf{or } P_\alpha \subseteq \mathsf{Port}(\beta) \end{array}\right] \textbf{ and } \left[\begin{array}{c} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right]$$

Then, by basic rewriting, conclude:

$$\begin{array}{c} \left[\begin{array}{c} (\textbf{not } q_\alpha \xrightarrow{P_\alpha} q'_\alpha) \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right] \\ \textbf{or } \left[\begin{array}{c} P_\alpha \cap \mathsf{Port}(\beta) = \emptyset \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right] \\ \textbf{or } \left[\begin{array}{c} P_\alpha \subseteq \mathsf{Port}(\beta) \textbf{ and } q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \textbf{ and } q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \\ \textbf{and } (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right] \end{array}$$

Then, by basic rewriting, conclude:

$$\textbf{false or } \left[\begin{array}{c} P_\alpha \cap \mathsf{Port}(\beta) = \emptyset \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right] \textbf{ or } \left[\begin{array}{c} P_\alpha \subseteq \mathsf{Port}(\beta) \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right]$$

Then, by basic rewriting, conclude:

$$\left[\begin{array}{c} P_\alpha \cap \mathsf{Port}(\beta) = \emptyset \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right] \textbf{ or } \left[\begin{array}{c} P_\alpha \subseteq \mathsf{Port}(\beta) \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \between (\mathsf{Port}(\beta)\,,\, P_\beta) \end{array}\right]$$

Then, by applying Definition 2 of $\between$, conclude:

$$\left[\begin{array}{c} P_\alpha \cap \mathsf{Port}(\beta) = \emptyset \textbf{ and} \\ P_\alpha \subseteq \mathsf{Port}(\alpha) \textbf{ and } P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and } \mathsf{Port}(\alpha) \cap P_\beta = \mathsf{Port}(\beta) \cap P_\alpha \end{array}\right] \textbf{ or } \left[\begin{array}{c} P_\alpha \subseteq \mathsf{Port}(\beta) \textbf{ and} \\ P_\alpha \subseteq \mathsf{Port}(\alpha) \textbf{ and } P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and } \mathsf{Port}(\alpha) \cap P_\beta = \mathsf{Port}(\beta) \cap P_\alpha \end{array}\right]$$

Then, by rewriting under ZFC, conclude:

$$\begin{bmatrix} P_\alpha \subseteq \mathsf{Port}(\alpha) \ \textbf{ and } \ P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and} \\ \mathsf{Port}(\alpha) \cap P_\beta = \emptyset = \mathsf{Port}(\beta) \cap P_\alpha \end{bmatrix} \ \textbf{ or } \ \begin{bmatrix} P_\alpha \subseteq \mathsf{Port}(\alpha) \ \textbf{ and } \ P_\beta \subseteq \mathsf{Port}(\beta) \\ \textbf{and } \ \mathsf{Port}(\alpha) \cap P_\beta = \mathsf{Port}(\beta) \cap P_\alpha \end{bmatrix}$$

Then, by applying Definition 4 of $\blacklozenge$, conclude $\big[(\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta)$ **or** $(\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta)\big]$. Then, by basic rewriting, conclude $(\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta)$. Then, by generalizing the premise, conclude:

$$\begin{bmatrix} \begin{bmatrix} q_\alpha \xrightarrow{P_\alpha}_\alpha q'_\alpha \ \textbf{ and } \ q_\beta \xrightarrow{P_\beta}_\beta q'_\beta \ \textbf{ and} \\ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \lozenge (\mathsf{Port}(\beta)\,,\, P_\beta) \end{bmatrix} \\ \textbf{implies } \ (\mathsf{Port}(\alpha)\,,\, P_\alpha) \blacklozenge (\mathsf{Port}(\beta)\,,\, P_\beta) \end{bmatrix} \ \textbf{ for all } \ q_\alpha\,,\, q_\beta\,,\, q'_\alpha\,,\, q'_\beta\,,\, P_\alpha\,,\, P_\beta$$

Then, by applying Definition 7 of $\langle\!\blacklozenge\!\rangle$, conclude $(Q_\alpha\,,\, \mathcal{P}_\alpha\,,\, \longrightarrow_\alpha\,,\, \imath_\alpha) \langle\!\blacklozenge\!\rangle (Q_\beta\,,\, \mathcal{P}_\beta\,,\, \longrightarrow_\beta\,,\, \imath_\beta)$. Then, by applying (A2), conclude $\alpha \langle\!\blacklozenge\!\rangle (Q_\beta\,,\, \mathcal{P}_\beta\,,\, \longrightarrow_\beta\,,\, \imath_\beta)$. Then, by applying (A3), conclude $\alpha \langle\!\blacklozenge\!\rangle \beta$. □

Proof (of Lemma 9). Assume:

(A1) $\alpha \mapsto \beta$

(A2) $\alpha = (Q_\alpha\,,\, \mathcal{P}_\alpha\,,\, \longrightarrow_\alpha\,,\, \imath_\alpha)$

Recall $\alpha \mapsto \beta$ from (A1). Then, by applying (A2), conclude $(Q_\alpha\,,\, \mathcal{P}_\alpha\,,\, \longrightarrow_\alpha\,,\, \imath_\alpha) \mapsto \beta$. Then, applying Definition 9 of $\mapsto$, conclude:

$$\left[\begin{bmatrix} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \ \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta) \neq \emptyset \end{bmatrix} \ \textbf{ implies } \ P_\alpha \subseteq \mathsf{Port}(\beta) \right] \ \textbf{ for all } \ q_\alpha\,,\, q'_\alpha\,,\, P_\alpha$$

Then, by introducing Proposition 6, conclude

$$\mathsf{Port}(\beta \boxdot \gamma) = \mathsf{Port}(\beta) \cup \mathsf{Port}(\gamma) \ \textbf{ and } \ \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \ \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta) \neq \emptyset \end{bmatrix} \\ \textbf{implies } \ P_\alpha \subseteq \mathsf{Port}(\beta) \end{bmatrix} \\ \textbf{for all } \ q_\alpha\,,\, q'_\alpha\,,\, P_\alpha \end{bmatrix}$$

Then, by rewriting under ZFC, conclude:

$$\mathsf{Port}(\beta) \subseteq \mathsf{Port}(\beta \boxdot \gamma) \ \textbf{ and } \ \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \ \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta) \neq \emptyset \end{bmatrix} \\ \textbf{implies } \ P_\alpha \subseteq \mathsf{Port}(\beta) \end{bmatrix} \\ \textbf{for all } \ q_\alpha\,,\, q'_\alpha\,,\, P_\alpha \end{bmatrix}$$

Then, by rewriting under ZFC, conclude:

$$\left[\begin{bmatrix} q_\alpha \xrightarrow{P_\alpha} q'_\alpha \textbf{ and} \\ P_\alpha \cap \mathsf{Port}(\beta \boxdot \gamma) \neq \emptyset \end{bmatrix} \textbf{ implies } P_\alpha \subseteq \mathsf{Port}(\beta \boxdot \gamma)\right] \textbf{ for all } q_\alpha\,,\, q'_\alpha\,,\, P_\alpha$$

Then, by applying Definition 9 of $\mapsto$, conclude $(Q_\alpha\,,\,\mathcal{P}_\alpha\,,\,\longrightarrow_\alpha\,,\,\imath_\alpha) \mapsto \beta \boxdot \gamma$. Then, by applying (A2), conclude $\alpha \mapsto \beta \boxdot \gamma$. $\square$

Proof (of Lemma 10). Suppose $\alpha \Mapsto \beta$. Then, by Definition 10 of $\Mapsto$, conclude $\big[\alpha \asymp \beta \textbf{ or } \alpha \mapsto \beta\big]$. Then, by applying Lemma 6, conclude $\big[\alpha \blacklozenge \beta \textbf{ or } \alpha \mapsto \beta\big]$. Then, by applying Lemma 8, conclude $\big[\alpha \blacklozenge \beta \textbf{ or } \alpha \blacklozenge \beta\big]$. Then, by basic rewriting, conclude $\alpha \blacklozenge \beta$. $\square$

Proof (of Lemma 11). Suppose $\alpha \Mapsto \beta\,,\,\gamma$. Then, by applying Definition 10 of $\Mapsto$, conclude $\big[\big[\alpha \asymp \beta \textbf{ or } \alpha \mapsto \beta\big] \textbf{ and } \big[\alpha \asymp \gamma \textbf{ or } \alpha \mapsto \gamma\big]\big]$. Then, by basic rewriting, conclude:

$$\begin{bmatrix} \big[\alpha \asymp \beta \textbf{ and } \alpha \asymp \gamma\big] \textbf{ or } \big[\alpha \asymp \beta \textbf{ and } \alpha \mapsto \gamma\big] \\ \textbf{or } \big[\alpha \mapsto \beta \textbf{ and } \alpha \asymp \gamma\big] \textbf{ or } \big[\alpha \mapsto \beta \textbf{ and } \alpha \mapsto \gamma\big] \end{bmatrix}$$

Proceed by case distinction.

- **Case:** $\big[\alpha \asymp \beta \textbf{ and } \alpha \asymp \gamma\big]$.
  Then, by applying Lemma 7, conclude $\alpha \asymp \beta \boxdot \gamma$. Then, by applying Definition 10 of $\Mapsto$, conclude $\alpha \Mapsto \beta \boxdot \gamma$.
- **Case:** $\big[\alpha \asymp \beta \textbf{ and } \alpha \mapsto \gamma\big]$.
  Then, by applying Lemma 9, conclude $\alpha \mapsto \gamma \boxdot \beta$. Then, by introducing Proposition 7:2, conclude $\big[\beta \boxdot \gamma \approx \gamma \boxdot \beta \textbf{ and } \alpha \mapsto \gamma \boxdot \beta\big]$. Then, by applying Proposition 9, conclude $\alpha \mapsto \beta \boxdot \gamma$. Then, by applying Definition 10 of $\Mapsto$, conclude $\alpha \Mapsto \beta \boxdot \gamma$.
- **Case:** $\big[\alpha \mapsto \beta \textbf{ and } \alpha \asymp \gamma\big]$.
  Then, by applying Lemma 9, conclude $\alpha \mapsto \beta \boxdot \gamma$. Then, by applying Definition 10 of $\Mapsto$, conclude $\alpha \Mapsto \beta \boxdot \gamma$.
- **Case:** $\big[\alpha \mapsto \beta \textbf{ and } \alpha \mapsto \gamma\big]$.
  Likewise.

Hence, after considering all cases, conclude $\alpha \Mapsto \beta \boxdot \gamma$. $\square$

Proof (of Theorem 2). Assume:

(A1) $\alpha_1 \boxdot \cdots \boxdot \alpha_k \in \mathcal{A}$

Proceed by induction on k.

- **Base:** $k = 1$. Immediate.
- **IH:** $\begin{bmatrix} \hat{\alpha}_1 \boxdot \cdots \boxdot \hat{\alpha}_{\hat{k}} \in \mathcal{A} \textbf{ implies} \\ \hat{\alpha}_1 \boxdot \cdots \boxdot \hat{\alpha}_{\hat{k}} = \hat{\alpha}_1 \boxtimes \cdots \boxtimes \hat{\alpha}_{\hat{k}} \end{bmatrix}$ **for all** $\hat{\alpha}_1 , \ldots , \hat{\alpha}_{\hat{k}} , \hat{k} < k$
- **Step:** $k > 1$. Assume:

  B1 $\hat{\alpha}_i = \alpha_{i+1}$ **for all** $1 \leq i \leq k-1$

  B2 $\hat{k} = k - 1$

  Recall $\alpha_1 \boxdot \cdots \boxdot \alpha_k \in \mathcal{A}$ from A1. Then, by applying Definition 11 of $\mathcal{A}$, conclude $\big[[i \neq j \textbf{ implies } \alpha_i \Mapsto \alpha_j] \textbf{ for all } 1 \leq i < j \leq k\big]$. Then, by basic rewriting, conclude $\big[\alpha_1 \Mapsto \alpha_2 , \ldots , \alpha_k \textbf{ and } \big[[i \neq j \textbf{ implies } \alpha_i \Mapsto \alpha_j] \textbf{ for all } 2 \leq i < j \leq k\big]\big]$. Then, by Corollary 1, conclude:

  $$\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \big[[i \neq j \textbf{ implies } \alpha_i \Mapsto \alpha_j] \textbf{ for all } 2 \leq i < j \leq k\big]$$

  Then, by applying B1, conclude:

  $$\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \begin{bmatrix} [i \neq j \textbf{ implies } \hat{\alpha}_i \Mapsto \hat{\alpha}_j] \\ \textbf{for all } 1 \leq i < j \leq k-1 \end{bmatrix}$$

  Then, by applying B2, conclude:

  $$\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \begin{bmatrix} [i \neq j \textbf{ implies } \hat{\alpha}_i \Mapsto \hat{\alpha}_j] \\ \textbf{for all } 1 \leq i < j \leq \hat{k} \end{bmatrix} \textbf{ and } \hat{k} < k$$

  Then, by applying Definition 11 of $\mathcal{A}$, conclude $\big[\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \hat{\alpha}_1 \boxdot \cdots \boxdot \hat{\alpha}_{\hat{k}} \in \mathcal{A} \textbf{ and } \hat{k} < k\big]$. Then, by applying $\boxed{\textbf{IH}}$, conclude $\big[\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \hat{\alpha}_1 \boxdot \cdots \boxdot \hat{\alpha}_{\hat{k}} = \hat{\alpha}_1 \boxtimes \cdots \boxtimes \hat{\alpha}_{\hat{k}}\big]$. Then, by applying B2, conclude $\big[\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \hat{\alpha}_1 \boxdot \cdots \boxdot \hat{\alpha}_{k-1} = \hat{\alpha}_1 \boxtimes \cdots \boxtimes \hat{\alpha}_{k-1}\big]$. Then, by applying B1, conclude $\big[\alpha_1 \Mapsto (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \alpha_2 \boxdot \cdots \boxdot \alpha_k = \alpha_2 \boxtimes \cdots \boxtimes \alpha_k\big]$. Then, by applying Lemma 10, $\big[\alpha_1 \blacklozenge (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \textbf{ and } \alpha_2 \boxdot \cdots \boxdot \alpha_k = \alpha_2 \boxtimes \cdots \boxtimes \alpha_k\big]$. Then, by Theorem 1, conclude:

  $$\begin{gathered} \alpha_1 \boxdot \alpha_2 \boxdot \cdots \boxdot \alpha_k = \alpha_1 \boxtimes (\alpha_2 \boxdot \cdots \boxdot \alpha_k) \\ \textbf{and } \alpha_2 \boxdot \cdots \boxdot \alpha_k = \alpha_2 \boxtimes \cdots \boxtimes \alpha_k \end{gathered}$$

  Then, by basic rewriting, conclude $\alpha_1 \boxdot \alpha_2 \boxdot \cdots \boxdot \alpha_k = \alpha_1 \boxtimes \alpha_2 \boxtimes \cdots \boxtimes \alpha_k$.

$\square$

D.4 Proofs of Section C

Proof (of Lemma 13). Assume:

A1 $\alpha \Mapsto \beta$

A2 $\alpha\,,\,\beta \asymp \gamma$

Observe:

Z1 Recall $\alpha \asymp \gamma$ from A2. Then, by introducing A1, conclude $\big[\alpha \mapsto \beta$ **and** $\alpha \asymp \gamma\big]$. Then, by applying Definition 10 of $\Mapsto$, conclude $\alpha \Mapsto \beta\,,\,\gamma$. Then, by applying Lemma 11, conclude $\alpha \Mapsto \beta \boxdot \gamma$. Then, by applying Lemma 10, conclude $\alpha \mathrel{\blacklozenge} \beta \boxdot \gamma$. Then, by applying Theorem 1, conclude $\alpha \boxdot (\beta \boxdot \gamma) = \alpha \boxtimes (\beta \boxdot \gamma)$.

Z2 Recall $\beta \asymp \gamma$ from A2. Then, by applying Definition 10 of $\Mapsto$, conclude $\beta \Mapsto \gamma$. Then, by applying Lemma 10, conclude $\beta \mathrel{\blacklozenge} \gamma$. Then, by applying Theorem 1, conclude $\beta \boxdot \gamma = \beta \boxtimes \gamma$.

Z3 Recall $\alpha \Mapsto \beta$ from A1. Then, by applying Lemma 10, conclude $\alpha \mathrel{\blacklozenge} \beta$. Then, by applying Theorem 1, conclude $\alpha \boxdot \beta = \alpha \boxtimes \beta$.

Z4 Recall $\alpha\,,\,\beta \asymp \gamma$ from A2. Then, by applying Definition 8 of $\asymp$, conclude $\big[\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset$ **and** $\mathsf{Port}(\beta) \cap \mathsf{Port}(\gamma) = \emptyset\big]$. Then, by rewriting under ZFC, conclude $\big[\mathsf{Port}(\beta) \cap \mathsf{Port}(\alpha) = \emptyset$ **and** $\mathsf{Port}(\gamma) \cap \mathsf{Port}(\beta) = \emptyset\big]$. Then, by applying Definition 8 of $\asymp$, conclude $\gamma \asymp \alpha\,,\,\beta$. Then, by applying Definition 10 of $\Mapsto$, conclude $\gamma \Mapsto \alpha\,,\,\beta$. Then, by applying Lemma 11, conclude $\gamma \Mapsto \alpha \boxdot \beta$. Then, by applying Lemma 10, conclude $\gamma \mathrel{\blacklozenge} \alpha \boxdot \beta$. Then, by applying Theorem 1, conclude $\gamma \boxdot (\alpha \boxdot \beta) = \gamma \boxtimes (\alpha \boxdot \beta)$.

Proceed by equational reasoning.

$\alpha \boxdot (\beta \boxdot \gamma)$

= /* *By applying Z1:* */

$\alpha \boxtimes (\beta \boxdot \gamma)$

= /* *By applying Z2:* */

$\alpha \boxtimes (\beta \boxtimes \gamma)$

= /* *By applying Proposition 5:2:* */

$(\alpha \boxtimes \beta) \boxtimes \gamma$

= /* *By applying Z3:* */

$(\alpha \boxdot \beta) \boxtimes \gamma$

= /* *By applying Proposition 5:3:* */

$\gamma \boxtimes (\alpha \boxdot \beta)$

= /* *By applying Z4:* */

$\gamma \boxdot (\alpha \boxdot \beta)$

= /* *By applying Proposition 5:3:* */

$(\alpha \boxdot \beta) \boxdot \gamma$

□

Proof (of Lemma 14). Assume:

(A1) $\alpha \Mapsto \beta\,,\, \gamma$

(A2) $\beta \Mapsto \alpha\,,\, \gamma$

Observe:

(Z1) Recall $\alpha \Mapsto \beta\,,\, \gamma$ from (A1). Then, by applying Lemma 11, conclude $\alpha \Mapsto \beta \boxdot \gamma$. Then, by applying Lemma 10, conclude $\alpha \blacklozenge \beta \boxdot \gamma$. Then, by applying Theorem 1, conclude $\alpha \boxdot (\beta \boxdot \gamma) = \alpha \boxtimes (\beta \boxdot \gamma)$.

(Z2) Recall $\beta \Mapsto \gamma$ from (A2). Then, by applying Lemma 10, conclude $\beta \blacklozenge \gamma$. Then, by applying Theorem 1, conclude $\beta \boxdot \gamma = \beta \boxtimes \gamma$.

(Z3) Recall $\alpha \Mapsto \gamma$ from (A1). Then, by applying Lemma 10, conclude $\alpha \blacklozenge \gamma$. Then, by applying Theorem 1, conclude $\alpha \boxdot \gamma = \alpha \boxtimes \gamma$.

(Z4) Recall $\beta \Mapsto \alpha\,,\, \gamma$ from (A2). Then, by applying Lemma 11, conclude $\beta \Mapsto \alpha \boxdot \gamma$. Then, by applying Lemma 10, conclude $\beta \blacklozenge \alpha \boxdot \gamma$. Then, by applying Theorem 1, conclude $\beta \boxdot (\alpha \boxdot \gamma) = \beta \boxtimes (\alpha \boxdot \gamma)$.

Proceed by equational reasoning.

$$
\begin{array}{ll}
 & \alpha \boxdot (\beta \boxdot \gamma) \\
= & \textit{/* By applying (Z1): */} \\
 & \alpha \boxtimes (\beta \boxdot \gamma) \\
= & \textit{/* By applying (Z2): */} \\
 & \alpha \boxtimes (\beta \boxtimes \gamma) \\
= & \textit{/* By applying Proposition 5:2: */} \\
 & (\alpha \boxtimes \beta) \boxtimes \gamma \\
= & \textit{/* By applying Proposition 5:3: */} \\
 & (\beta \boxtimes \alpha) \boxtimes \gamma \\
= & \textit{/* By applying Proposition 5:2: */} \\
 & \beta \boxtimes (\alpha \boxtimes \gamma) \\
= & \textit{/* By applying (Z3): */} \\
 & \beta \boxtimes (\alpha \boxdot \gamma) \\
= & \textit{/* By applying (Z4): */} \\
 & \beta \boxdot (\alpha \boxdot \gamma)
\end{array}
$$

□

Proof (of Lemma 15). Assume:

(A1) $\alpha \asymp \beta\,,\, \gamma$

(A2) $\beta \asymp \gamma$

Recall $\alpha \asymp \beta$ from (A1). Then, by applying Definition 8 of $\asymp$, conclude $\mathsf{Port}(\alpha) \cap \mathsf{Port}(\beta) = \emptyset$. Then, by rewriting under ZFC, conclude $\mathsf{Port}(\beta) \cap \mathsf{Port}(\alpha) = \emptyset$. Then, by applying Definition 8 of $\asymp$, conclude $\beta \asymp \alpha$. Then, by introducing (A2), conclude $\beta \asymp \alpha\,,\,\gamma$. Then, by introducing (A1), conclude $\big[\alpha \asymp \beta\,,\,\gamma$ **and** $\beta \asymp \alpha\,,\,\gamma\big]$. Then, by applying Definition 10 of $\Rrightarrow$, conclude $\big[\alpha \Rrightarrow \beta\,,\,\gamma$ **and** $\beta \Rrightarrow \alpha\,,\,\gamma\big]$. Then, by applying Lemma 14, conclude $\alpha \boxdot (\beta \boxdot \gamma) \approx \beta \boxdot (\alpha \boxdot \gamma)$. □

Proof (of Lemma 16). Assume:

(A1) $1 < j \leq l$

(A2) $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$

Observe:

(Z1) Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from (A2). Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$\begin{gathered}\big[\big[i_1 \neq i_2 \ \textbf{implies}\ \alpha_{i_1} \Rrightarrow \alpha_{i_2}\big] \ \textbf{for all}\ 1 \leq i_1\,,\,i_2 \leq k\big] \ \textbf{and} \\ \big[\big[j_1 \neq j_2 \ \textbf{implies}\ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1\,,\,j_2 \leq l\big] \ \textbf{and} \\ \big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\,1 \leq j_2 \leq l\big]\end{gathered}$$

(Z2) Recall $\big[\big[j_1 \neq j_2$ **implies** $\beta_{j_1} \asymp \beta_{j_2}\big]$ **for all** $1 \leq j_1\,,\,j_2 \leq l\big]$ from (Z1). Then, by introducing (A1), conclude $\big[1 < j \leq l$ **and** $\big[\big[j_1 \neq j_2$ **implies** $\beta_{j_1} \asymp \beta_{j_2}\big]$ **for all** $1 \leq j_1\,,\,j_2 \leq l\big]\big]$. Then, by basic rewriting, conclude $\big[\beta_{j-1} \asymp \beta_j$ **and** $\beta_{j-1} \asymp \beta_{j+1}\,,\,\ldots\,,\,\beta_l$ **and** $\beta_j \asymp \beta_{j+1}\,,\,\ldots\,,\,\beta_l\big]$. Then, by applying Corollary 2, conclude $\big[\beta_{j-1} \asymp \beta_j$ **and** $\beta_{j-1} \asymp \beta_{j+1} \cdots \beta_l$ **and** $\beta_j \asymp \beta_{j+1} \cdots \beta_l\big]$. Then, by applying Definition 10 of $\Rrightarrow$, conclude $\big[\beta_{j-1} \Rrightarrow \beta_j$ **and** $\beta_{j-1} \asymp \beta_{j+1} \cdots \beta_l$ **and** $\beta_j \asymp \beta_{j+1} \cdots \beta_l\big]$. Then, by applying Lemma 13, conclude $\beta_{j-1}\beta_j[\beta]_{j+1}^l \approx (\beta_{j-1}\beta_j)[\beta]_{j+1}^l$. Then, by applying Lemma 2, conclude $[\alpha]_1^k[\beta]_1^l \approx [\alpha]_1^k[\beta]_1^{j-2}(\beta_{j-1}\beta_j)[\beta]_{j+1}^l$.

Assume:

(B1) $[\tilde{\beta}]_1^{j-2} = [\beta]_1^{j-2}$ **and** $\tilde{\beta}_{j-1} = \beta_{j-1}\beta_j$ **and** $[\tilde{\beta}]_j^{l-1} = [\beta]_{j+1}^l$

Observe:

(Y1) Recall $\big[\big[j_1 \neq j_2$ **implies** $\beta_{j_1} \asymp \beta_{j_2}\big]$ **for all** $1 \leq j_1\,,\,j_2 \leq l\big]$ from (Z1). Then, by introducing (A1), conclude $\big[1 < j \leq l$ **and** $\big[\big[j_1 \neq j_2$ **implies** $\beta_{j_1} \asymp \beta_{j_2}\big]$ **for all** $1 \leq j_1\,,\,j_2 \leq l\big]\big]$. Then, by basic rewriting, conclude:

$$\begin{aligned}&\big[\big[j_1 \neq j_2 \ \textbf{implies}\ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1 \leq j-2\,,\,1 \leq j_2 \leq l\big] \\ &\textbf{and}\ \big[\big[j-1 \neq j_2 \ \textbf{implies}\ \beta_{j-1} \asymp \beta_{j_2}\big] \ \textbf{for all}\ 1 \leq j_2 \leq l\big] \\ &\textbf{and}\ \big[\big[j \neq j_2 \ \textbf{implies}\ \beta_j \asymp \beta_{j_2}\big] \ \textbf{for all}\ 1 \leq j_2 \leq l\big] \\ &\textbf{and}\ \big[\big[j_1 \neq j_2 \ \textbf{implies}\ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all}\ j+1 \leq j_1 \leq l\,,\,1 \leq j_2 \leq l\big]\end{aligned}$$

Then, by applying Definition 8 of $\asymp$, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \big[[j-1 \neq j_2 \ \textbf{implies} \ \mathsf{Port}(\beta_{j-1}) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset] \ \textbf{for all} \ 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \big[[j \neq j_2 \ \textbf{implies} \ \mathsf{Port}(\beta_j) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset] \ \textbf{for all} \ 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by basic rewriting, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \left[\begin{array}{c}\left[\begin{array}{l}[j-1 \neq j_2 \ \textbf{implies} \ \mathsf{Port}(\beta_{j-1}) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset] \\ \textbf{and} \ [j \neq j_2 \ \textbf{implies} \ \mathsf{Port}(\beta_j) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset]\end{array}\right] \\ \textbf{for all} \ 1 \leq j_2 \leq l\end{array}\right] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by basic rewriting, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \left[\begin{array}{c}\left[\left[\begin{array}{l}j-1 \neq j_2 \\ \textbf{and} \ j \neq j_2\end{array}\right] \ \textbf{implies} \ \left[\begin{array}{l}\mathsf{Port}(\beta_{j-1}) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \\ \textbf{and} \ \mathsf{Port}(\beta_j) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset\end{array}\right]\right] \\ \textbf{for all} \ 1 \leq j_2 \leq l\end{array}\right] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by rewriting under ZFC, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \left[\begin{array}{c}\left[\left[\begin{array}{l}j-1 \neq j_2 \\ \textbf{and} \ j \neq j_2\end{array}\right] \ \textbf{implies} \ (\mathsf{Port}(\beta_{j-1}) \cup \mathsf{Port}(\beta_j)) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset\right] \\ \textbf{for all} \ 1 \leq j_2 \leq l\end{array}\right] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by applying Proposition 6, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \left[\begin{array}{c}\left[\left[\begin{array}{l}j-1 \neq j_2 \\ \textbf{and} \ j \neq j_2\end{array}\right] \ \textbf{implies} \ \mathsf{Port}(\beta_{j-1}\beta_j) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset\right] \\ \textbf{for all} \ 1 \leq j_2 \leq l\end{array}\right] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by applying Definition 8 of $\asymp$, conclude:

$$\begin{array}{l}\big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l\big] \\ \textbf{and} \ \left[\left[\left[\begin{array}{l}j-1 \neq j_2 \\ \textbf{and} \ j \neq j_2\end{array}\right] \ \textbf{implies} \ (\beta_{j-1}\beta_j) \asymp \beta_{j_2}\right] \ \textbf{for all} \ 1 \leq j_2 \leq l\right] \\ \textbf{and} \ \big[[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}] \ \textbf{for all} \ j+1 \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by applying (B1), conclude:

$$\begin{aligned}&\big[[j_1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1 \leq j-2\,,\, 1 \leq j_2 \leq l-1\big]\\ \textbf{and}\ &\big[[j-1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j-1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ 1 \leq j_2 \leq l-1\big]\\ \textbf{and}\ &\big[[j_1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ j \leq j_1 \leq l\,,\, 1 \leq j_2 \leq l-1\big]\end{aligned}$$

Then, by basic rewriting, conclude $\big[[j_1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1\,,\, j_2 \leq l-1\big]$.

(Y2) Recall $\big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l\big]$ from (Z1). Then, by introducing (A1), conclude $\big[1 < j \leq l \ \textbf{and}\ \big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l\big]\big]$. Then, by basic rewriting, conclude:

$$\begin{aligned}&\big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq j-2\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \beta_{j-1}\,,\, \beta_j \ \textbf{for all}\ 1 \leq i \leq k\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, j+1 \leq j_2 \leq l\big]\end{aligned}$$

Then, by applying Lemma 11, conclude:

$$\begin{aligned}&\big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq j-2\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \beta_{j-1}\beta_j \ \textbf{for all}\ 1 \leq i \leq k\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \beta_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, j+1 \leq j_2 \leq l\big]\end{aligned}$$

Then, by applying (B1), conclude:

$$\begin{aligned}&\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq j-2\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \tilde{\beta}_{j-1} \ \textbf{for all}\ 1 \leq i \leq k\big] \ \textbf{and}\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, j \leq j_2 \leq l-1\big]\end{aligned}$$

Then, by basic rewriting, conclude $\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l-1\big]$.

(Y3) Recall $\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l-1\big]$ from (Y2). Then, by introducing (Y1), conclude:

$$\begin{aligned}&\big[[j_1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j-1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1\,,\, j_2 \leq l-1\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l-1\big]\end{aligned}$$

Then, by introducing (Z1), conclude:

$$\begin{aligned}&\big[[i_1 \neq i_2 \ \textbf{implies}\ \alpha_{i_1} \Rrightarrow \alpha_{i_2}\big] \ \textbf{for all}\ 1 \leq i_1\,,\, i_2 \leq k\big]\\ \textbf{and}\ &\big[[j_1 \neq j_2 \ \textbf{implies}\ \tilde{\beta}_{j-1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all}\ 1 \leq j_1\,,\, j_2 \leq l-1\big]\\ \textbf{and}\ &\big[\alpha_i \Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all}\ 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l-1\big]\end{aligned}$$

Then, by applying Definition 12 of $\mathcal{B}$, conclude $[\alpha]_1^k[\tilde{\beta}]_1^{l-1} \in \mathcal{B}$. Then, by applying (B1), conclude $[\alpha]_1^k[\beta]_1^{j-2}(\beta_{j-1}\beta_j)[\beta]_{j+1}^l \in \mathcal{B}$.

Conclude the consequent of this lemma by **and**-ing the results from Y3 and Z2
. $\square$

Proof (of Lemma 17). Assume:

A1 $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$

A2 $\alpha_k \asymp \beta_2 \,, \ldots \,, \beta_l$

Observe:

Z1 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from A2. Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$\big[\big[i_1 \neq i_2 \ \textbf{implies} \ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}\big] \ \textbf{for all} \ 1 \leq i_1 \,, i_2 \leq k\big] \ \textbf{and}$$
$$\big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 1 \leq j_1 \,, j_2 \leq l\big] \ \textbf{and}$$
$$\big[\alpha_i \not\Rightarrow \beta_{j_2} \ \textbf{for all} \ 1 \leq i \leq k \,, 1 \leq j_2 \leq l\big]$$

Z2 Recall from Z1:

$$\big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 1 \leq j_1 \,, j_2 \leq l\big]$$
$$\textbf{and} \ \big[\alpha_i \not\Rightarrow \beta_{j_2} \ \textbf{for all} \ 1 \leq i \leq k \,, 1 \leq j_2 \leq l\big]$$

Then, by basic rewriting, conclude $\big[\beta_1 \asymp \beta_2 \,, \ldots \,, \beta_l \ \textbf{and} \ \alpha_k \not\Rightarrow \beta_1\big]$. Then, by introducing A2, conclude $\big[\alpha_k \asymp \beta_2 \,, \ldots \,, \beta_l \ \textbf{and} \ \beta_1 \asymp \beta_2 \,, \ldots \,, \beta_l \ \textbf{and} \ \alpha_k \not\Rightarrow \beta_1\big]$. Then, by applying Corollary 2, conclude $\big[\alpha_k \,, \beta_1 \asymp \beta_2 \cdots \beta_l \ \textbf{and} \ \alpha_k \not\Rightarrow \beta_1\big]$. Then, by applying Lemma 13, conclude $\alpha_k\beta_1[\beta]_2^l \approx (\alpha_k\beta_1)[\beta]_2^l$. Then, by applying Lemma 2, conclude $[\alpha]_1^{k-1}(\alpha_k\beta_1)[\beta]_2^l \approx [\alpha]_1^k[\beta]_1^l$.

Assume:

B1 $\tilde{\beta}_1 = \alpha_k\beta_1 \ \textbf{and} \ [\tilde{\beta}]_2^l = [\beta]_2^l$

Observe:

Y1 Recall $\big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 1 \leq j_1 \,, j_2 \leq l\big]$ from Z1. Then, by basic rewriting, conclude:

$$\beta_1 \asymp \beta_2 \,, \ldots \,, \beta_l$$
$$\textbf{and} \ \big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 2 \leq j_1 \leq l \,, 2 \leq j_2 \leq l\big]$$

Then, by introducing A2, conclude:

$$\alpha_k \asymp \beta_2 \,, \ldots \,, \beta_l \ \textbf{and} \ \beta_1 \asymp \beta_2 \,, \ldots \,, \beta_l$$
$$\textbf{and} \ \big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 2 \leq j_1 \leq l \,, 2 \leq j_2 \leq l\big]$$

Then, by applying Definition 8 of $\asymp$, conclude:

$$\big[\mathsf{Port}(\alpha_k) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \ \textbf{for all} \ 2 \leq j_2 \leq l\big]$$
$$\textbf{and} \ \big[\mathsf{Port}(\beta_1) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \ \textbf{for all} \ 2 \leq j_2 \leq l\big]$$
$$\textbf{and} \ \big[\big[j_1 \neq j_2 \ \textbf{implies} \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{for all} \ 2 \leq j_1 \leq l \,, 2 \leq j_2 \leq l\big]$$

Then, by basic rewriting, conclude:

$$\left[\begin{bmatrix} \mathsf{Port}(\alpha_k) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \\ \textbf{and } \mathsf{Port}(\beta_1) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \end{bmatrix} \textbf{ for all } 2 \leq j_2 \leq l\right]$$
$$\textbf{and } \left[\left[j_1 \neq j_2 \textbf{ implies } \beta_{j_1} \asymp \beta_{j_2}\right] \textbf{ for all } 2 \leq j_1 \leq l\,,\, 2 \leq j_2 \leq l\right]$$

Then, by rewriting under ZFC, conclude:

$$\left[(\mathsf{Port}(\alpha_k) \cup \mathsf{Port}(\beta_1)) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \textbf{ for all } 2 \leq j_2 \leq l\right]$$
$$\textbf{and } \left[\left[j_1 \neq j_2 \textbf{ implies } \beta_{j_1} \asymp \beta_{j_2}\right] \textbf{ for all } 2 \leq j_1 \leq l\,,\, 2 \leq j_2 \leq l\right]$$

Then, by applying Proposition 6, conclude:

$$\left[\mathsf{Port}(\alpha_k\beta_1) \cap \mathsf{Port}(\beta_{j_2}) = \emptyset \textbf{ for all } 2 \leq j_2 \leq l\right]$$
$$\textbf{and } \left[\left[j_1 \neq j_2 \textbf{ implies } \beta_{j_1} \asymp \beta_{j_2}\right] \textbf{ for all } 2 \leq j_1 \leq l\,,\, 2 \leq j_2 \leq l\right]$$

Then, by applying Definition 8 of $\asymp$, conclude:

$$\left[\alpha_k\beta_1 \asymp \beta_{j_2} \textbf{ for all } 2 \leq j_2 \leq l\right]$$
$$\textbf{and } \left[\left[j_1 \neq j_2 \textbf{ implies } \beta_{j_1} \asymp \beta_{j_2}\right] \textbf{ for all } 2 \leq j_1 \leq l\,,\, 2 \leq j_2 \leq l\right]$$

Then, by applying B1, conclude:

$$\left[\tilde{\beta}_1 \asymp \tilde{\beta}_{j_2} \textbf{ for all } 2 \leq j_2 \leq l\right]$$
$$\textbf{and } \left[\left[j_1 \neq j_2 \textbf{ implies } \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\right] \textbf{ for all } 2 \leq j_1 \leq l\,,\, 2 \leq j_2 \leq l\right]$$

Then, by basic rewriting, conclude $\left[\left[j_1 \neq j_2 \textbf{ implies } \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\right] \textbf{ for all } 1 \leq j_1\,,\, j_2 \leq l\right]$.

Y2 Recall from Z1:

$$\left[\left[i_1 \neq i_2 \textbf{ implies } \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\right] \textbf{ for all } 1 \leq i_1\,,\, i_2 \leq k\right]$$
$$\textbf{and } \left[\alpha_i \not\Rrightarrow \beta_{j_2} \textbf{ for all } 1 \leq i \leq k\,,\, 1 \leq j_2 \leq l\right]$$

Then, by basic rewriting, conclude:

$$\alpha_1\,,\,\ldots\,,\,\alpha_{k-1} \not\Rrightarrow \alpha_k \textbf{ and } \alpha_1\,,\,\ldots\,,\,\alpha_{k-1} \not\Rrightarrow \beta_1$$
$$\textbf{and } \left[\left[i_1 \neq i_2 \textbf{ implies } \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\right] \textbf{ for all } 1 \leq i_1\,,\, i_2 \leq k-1\right]$$
$$\textbf{and } \left[\alpha_i \not\Rrightarrow \beta_{j_2} \textbf{ for all } 1 \leq i \leq k-1\,,\, 2 \leq j_2 \leq l\right]$$

Then, by applying Lemma 11, conclude:

$$\alpha_1\,,\,\ldots\,,\,\alpha_{k-1} \not\Rrightarrow \alpha_k\beta_1$$
$$\textbf{and } \left[\left[i_1 \neq i_2 \textbf{ implies } \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\right] \textbf{ for all } 1 \leq i_1\,,\, i_2 \leq k-1\right]$$
$$\textbf{and } \left[\alpha_i \not\Rrightarrow \beta_{j_2} \textbf{ for all } 1 \leq i \leq k-1\,,\, 2 \leq j_2 \leq l\right]$$

Then, by applying B1, conclude:

$$\alpha_1\,,\,\ldots\,,\,\alpha_{k-1} \not\Rrightarrow \tilde{\beta}_1$$
$$\textbf{and } \left[\left[i_1 \neq i_2 \textbf{ implies } \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\right] \textbf{ for all } 1 \leq i_1\,,\, i_2 \leq k-1\right]$$
$$\textbf{and } \left[\alpha_i \not\Rrightarrow \tilde{\beta}_{j_2} \textbf{ for all } 1 \leq i \leq k-1\,,\, 2 \leq j_2 \leq l\right]$$

Then, by basic rewriting, conclude:

$$\begin{array}{l}\big[\big[i_1 \neq i_2 \ \textbf{implies} \ \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\big] \ \textbf{for all} \ 1 \leq i_1 \,,\, i_2 \leq k-1\big] \\ \textbf{and} \ \big[\alpha_i \not\Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all} \ 1 \leq i \leq k-1 \,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by introducing Y1, conclude:

$$\begin{array}{l}\big[\big[j_1 \neq j_2 \ \textbf{implies} \ \tilde{\beta}_{j_1} \asymp \tilde{\beta}_{j_2}\big] \ \textbf{for all} \ 1 \leq j_1 \,,\, j_2 \leq l\big] \\ \textbf{and} \ \big[\big[i_1 \neq i_2 \ \textbf{implies} \ \alpha_{i_1} \not\Rrightarrow \alpha_{i_2}\big] \ \textbf{for all} \ 1 \leq i_1 \,,\, i_2 \leq k-1\big] \\ \textbf{and} \ \big[\alpha_i \not\Rrightarrow \tilde{\beta}_{j_2} \ \textbf{for all} \ 1 \leq i \leq k-1 \,,\, 1 \leq j_2 \leq l\big]\end{array}$$

Then, by applying Definition 12 of $\mathcal{B}$, conclude $[\alpha]_1^{k-1}[\tilde{\beta}]_1^l \in \mathcal{B}$. Then, by applying B1, conclude $[\alpha]_1^{k-1}(\alpha_k\beta_1)[\beta]_2^l \in \mathcal{B}$.

Conclude the consequent of this lemma by **and**-ing the results from Y2 and Z2 . $\square$

Proof (of Lemma 18). Assume:

A1 $1 \leq j \leq l$

A2 $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$

Proceed by induction on $1 \leq j \leq l$.

– **Base:** $j = 1$. Observe:

Z1 Recall $\big[\big[\textbf{not} \ 1 < j \leq l\big] \ \textbf{implies} \ \Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$ from Definition 18. Then, by applying $\boxed{\textbf{Base}}$, conclude $\big[\big[\textbf{not} \ 1 < 1 \leq l\big] \ \textbf{implies} \ \Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$. Then, by basic rewriting, conclude $\big[\textbf{false} \ \textbf{implies} \ \Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$. Then, by basic rewriting, conclude $\Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l$.

Z2 By equational reasoning, conclude:

$$\begin{array}{cl} & \Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) \\ = & /^* \ \textit{by applying Z1} \ ^*/ \\ & [\alpha]_1^k[\beta]_1^l \\ = & /^* \ \textit{by unfolding} \ ^*/ \\ & [\alpha]_1^k\beta_j[\beta]_2^l \\ = & /^* \ \textit{by inserting an ``empty''} \ [\_]_-^- \ ^*/ \\ & [\alpha]_1^k\beta_j[\beta]_1^0[\beta]_2^l \\ = & /^* \ \textit{by basic rewriting} \ ^*/ \\ & [\alpha]_1^k\beta_j[\beta]_1^{1-1}[\beta]_{1+1}^l \\ = & /^* \ \textit{by applying} \ \boxed{\textbf{Base}} \ ^*/ \\ & [\alpha]_1^k\beta_j[\beta]_1^{j-1}[\beta]_{j+1}^l \end{array}$$

Z3 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from A2. Then, by applying Z1, conclude $\Leftarrow(\beta_j\,,\ [\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$.

Z4 Recall $\Leftarrow(\beta_j\,,\ [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l$ from Z1. Then, by applying Proposition 3, conclude $\Leftarrow(\beta_j\,,\ [\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l$.

Conclude the consequent of this lemma by **and**-ing the results from Z2, Z3, and Z4.

- **IH:**

$$\left[\begin{bmatrix}1 \leq \hat{j} \leq l \ \textbf{ and} \\ [\alpha]_1^k[\hat{\beta}]_1^l \in \mathcal{B}\end{bmatrix} \ \textbf{implies} \ \begin{bmatrix}\Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) = [\alpha]_1^k\hat{\beta}_{\hat{j}}[\hat{\beta}]_1^{\hat{j}-1}[\hat{\beta}]_{\hat{j}+1}^l \\ \textbf{and} \ \Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) \in \mathcal{B} \\ \textbf{and} \ \Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) \approx [\alpha]_1^k[\hat{\beta}]_1^l\end{bmatrix}\right]$$
$$\textbf{for all} \ \hat{\beta}_1\,,\ \ldots\,,\ \hat{\beta}_l\,,\ \hat{j} < j$$

- **Step:** $1 < j \leq l$. Assume:

B1 $$\left[\hat{\beta}_{j'} = \beta_j \ \textbf{ for all} \ \left[1 \leq j' \leq j-2 \ \textbf{ and } \ j+1 \leq j' \leq l\right]\right]$$
$$\textbf{and} \ \hat{\beta}_{j-1} = \beta_j \ \textbf{ and } \ \hat{\beta}_j = \beta_{j-1}$$

B2 $\hat{j} = j - 1$

Observe:

Y1 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$. Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$\left[\left[i_1 \neq i_2 \ \textbf{ implies } \ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}\right] \ \textbf{for all} \ 1 \leq i_1\,,\ i_2 \leq k\right] \ \textbf{and}$$
$$\left[\left[j_1 \neq j_2 \ \textbf{ implies } \ \beta_{j_1} \asymp \beta_{j_2}\right] \ \textbf{for all} \ 1 \leq j_1\,,\ j_2 \leq l\right] \ \textbf{and}$$
$$\left[\alpha_i \not\Rightarrow \beta_{j'} \ \textbf{ for all } \ 1 \leq i \leq k\,,\ 1 \leq j' \leq l\right]$$

Then, by applying B1, conclude:

$$\left[\left[i_1 \neq i_2 \ \textbf{ implies } \ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}\right] \ \textbf{for all} \ 1 \leq i_1\,,\ i_2 \leq k\right] \ \textbf{and}$$
$$\left[\left[j_1 \neq j_2 \ \textbf{ implies } \ \hat{\beta}_{j_1} \asymp \hat{\beta}_{j_2}\right] \ \textbf{for all} \ 1 \leq j_1\,,\ j_2 \leq l\right] \ \textbf{and}$$
$$\left[\alpha_i \not\Rightarrow \hat{\beta}_{j'} \ \textbf{ for all } \ 1 \leq i \leq k\,,\ 1 \leq j' \leq l\right]$$

Then, by applying Definition 12 of $\mathcal{B}$, conclude $[\alpha]_1^k[\hat{\beta}]_1^l \in \mathcal{B}$. Then, by introducing $\boxed{\textbf{Step}}$, conclude $\left[1 < j \leq l \ \textbf{ and } \ [\alpha]_1^k[\hat{\beta}]_1^l \in \mathcal{B}\right]$. Then, by basic rewriting, conclude $\left[1 \leq j-1 \leq l \ \textbf{ and } \ [\alpha]_1^k[\hat{\beta}]_1^l \in \mathcal{B}\right]$. Then, by applying B2, conclude $\left[\hat{j} < j \ \textbf{ and } \ 1 \leq \hat{j} \leq l \ \textbf{ and } \ [\alpha]_1^k[\hat{\beta}]_1^l \in \mathcal{B}\right]$. Then, by applying $\boxed{\textbf{IH}}$, conclude:

$$\Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) = [\alpha]_1^k\hat{\beta}_{\hat{j}}[\hat{\beta}]_1^{\hat{j}-1}[\hat{\beta}]_{\hat{j}+1}^l$$
$$\textbf{and} \ \Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) \in \mathcal{B}$$
$$\textbf{and} \ \Leftarrow(\hat{\beta}_{\hat{j}}\,,\ [\alpha]_1^k[\hat{\beta}]_1^l) \approx [\alpha]_1^k[\hat{\beta}]_1^l$$

(Y2) Recall $\left[1 < j \leq l \text{ **implies** } \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^{j-2}\beta_j\beta_{j-1}[\beta]_{j+1}^l)\right]$ from Definition 18. Then, by applying $\boxed{\textbf{Step}}$, conclude $\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^{j-2}\beta_j\beta_{j-1}[\beta]_{j+1}^l)$.

(Y3) By equational reasoning, conclude:

$$\begin{aligned}
& \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \\
= \quad & \text{/* \textit{by applying} (Y2) */} \\
& \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^{j-2}\beta_j\beta_{j-1}[\beta]_{j+1}^l) \\
= \quad & \text{/* \textit{by applying} (B1) */} \\
& \Leftarrow(\hat{\beta}_{j-1}\,,\,[\alpha]_1^k[\hat{\beta}]_1^{j-2}\hat{\beta}_{j-1}\hat{\beta}_j[\hat{\beta}]_{j+1}^l) \\
= \quad & \text{/* \textit{by applying} (B2) */} \\
& \Leftarrow(\hat{\beta}_{\hat{j}}\,,\,[\alpha]_1^k[\hat{\beta}]_1^{\hat{j}-1}\hat{\beta}_{\hat{j}}\hat{\beta}_{\hat{j}+1}[\hat{\beta}]_{\hat{j}+2}^l) \\
= \quad & \text{/* \textit{by collapsing} */} \\
& \Leftarrow(\hat{\beta}_{\hat{j}}\,,\,[\alpha]_1^k[\hat{\beta}]_1^l)
\end{aligned}$$

(Y4) By equational reasoning, conclude:

$$\begin{aligned}
& \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \\
= \quad & \text{/* \textit{by applying} (Y3) */} \\
& \Leftarrow(\hat{\beta}_{\hat{j}}\,,\,[\alpha]_1^k[\hat{\beta}]_1^l) \\
= \quad & \text{/* \textit{by applying} (Y1) */} \\
& [\alpha]_1^k\hat{\beta}_{\hat{j}}[\hat{\beta}]_1^{\hat{j}-1}[\hat{\beta}]_{\hat{j}+1}^l \\
= \quad & \text{/* \textit{by applying} (B2) */} \\
& [\alpha]_1^k\hat{\beta}_{j-1}[\hat{\beta}]_1^{j-2}[\hat{\beta}]_j^l \\
= \quad & \text{/* \textit{by unfolding} */} \\
& [\alpha]_1^k\hat{\beta}_{j-1}[\hat{\beta}]_1^{j-2}\hat{\beta}_j[\hat{\beta}]_{j+1}^l \\
= \quad & \text{/* \textit{by applying} (B1) */} \\
& [\alpha]_1^k\beta_j[\beta]_1^{j-2}\beta_{j-1}[\beta]_{j+1}^l \\
= \quad & \text{/* \textit{by collapsing} */} \\
& [\alpha]_1^k\beta_j[\beta]_1^{j-1}[\beta]_{j+1}^l
\end{aligned}$$

(Y5) Recall $\Leftarrow(\hat{\beta}_{\hat{j}}\,,\,[\alpha]_1^k[\hat{\beta}]_1^l) \in \mathcal{B}$ from (Y1). Then, by applying (Y3), conclude $\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$.

(Y6) Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$. Then, by applying Definition 12 of $\mathcal{B}$, conclude $\left[\left[j_1 \neq j_2 \text{ **implies** } \beta_{j_1} \asymp \beta_{j_2}\right] \text{ **for all** } 1 \leq j_1\,,\,j_2 \leq l\right]$. Then, by basic rewriting, conclude $\left[\beta_{j-1} \asymp \beta_j\,,\,\beta_{j+1}\,,\,\ldots\,,\,\beta_l \text{ **and** } \beta_j \asymp \beta_{j+1}\,,\,\ldots\,,\,\beta_l\right]$. Then, by applying Corollary 2, conclude $\left[\beta_{j-1} \asymp \beta_j\,,\,[\beta]_{j+1}^l \text{ **and** } \beta_j \asymp [\beta]_{j+1}^l\right]$. Then, by applying Lemma 15, conclude $\beta_{j-1}\beta_j[\beta]_{j+1}^l \approx \beta_j\beta_{j-1}[\beta]_{j+1}^l$.

Y7 By equational reasoning, conclude:

$$
\begin{aligned}
& \Leftarrow(\beta_j \,,\, [\alpha]_1^k[\beta]_1^l) \\
= \;& \text{/* } \textit{by applying } \text{Y3} \text{ */} \\
& \Leftarrow(\hat{\beta}_{\hat{j}} \,,\, [\alpha]_1^k[\hat{\beta}]_1^l) \\
\approx \;& \text{/* } \textit{by applying } \text{Y1} \text{ */} \\
& [\alpha]_1^k[\hat{\beta}]_1^l \\
= \;& \text{/* } \textit{by unfolding} \text{ */} \\
& [\alpha]_1^k[\hat{\beta}]_1^{j-2}\hat{\beta}_{j-1}\hat{\beta}_j[\hat{\beta}]_{j+1}^l \\
= \;& \text{/* } \textit{by applying } \text{B1} \text{ */} \\
& [\alpha]_1^k[\beta]_1^{j-2}\beta_j\beta_{j-1}[\beta]_{j+1}^l \\
\approx \;& \text{/* } \textit{by applying } \text{Y6} \text{ */} \\
& [\alpha]_1^k[\beta]_1^{j-2}\beta_{j-1}\beta_j[\beta]_{j+1}^l \\
= \;& \text{/* } \textit{by collapsing} \text{ */} \\
& [\alpha]_1^k[\beta]_1^l
\end{aligned}
$$

Conclude the consequent of this lemma by **and**-ing the results from Y4, Y5 , and Y7.

□

Proof (of Lemma 19). Assume:

A1 $1 \leq i \leq k$

A2 $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$

Proceed by induction on $1 \leq i \leq k$.

- **Base:** $i = k$. Observe:

  Z1 Recall $\big[[$**not** $1 \leq i < k]$ **implies** $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$ from Definition 19. Then, by applying $\boxed{\textbf{Base}}$, conclude $\big[[$**not** $1 \leq k < k]$ **implies** $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$. Then, by basic rewriting, conclude $\big[$**false implies** $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l\big]$. Then, by basic rewriting, conclude $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l$.

  Z2 By equational reasoning, conclude:

$$
\begin{aligned}
& \Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) \\
= \;& \textit{/* by applying Z1 */} \\
& [\alpha]_1^k[\beta]_1^l \\
= \;& \textit{/* by unfolding */} \\
& [\alpha]_1^{k-1}\alpha_i[\beta]_1^l \\
= \;& \textit{/* by inserting an ``empty'' } [\_]_{\_}^{\_} \textit{ */} \\
& [\alpha]_1^{k-1}[\alpha]_{k+1}^k\alpha_i[\beta]_1^l \\
= \;& \textit{/* by applying } \boxed{\textbf{Base}} \textit{ */} \\
& [\alpha]_1^{i-1}[\alpha]_{i+1}^k\alpha_i[\beta]_1^l
\end{aligned}
$$

Z3 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from A2. Then, by applying Z1, conclude $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$.

Z4 Recall $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta]_1^l$ from Z1. Then, by applying Proposition 3, conclude $\Rightarrow(\alpha_i \,,\, [\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l$.

Conclude the consequent of this lemma by **and**-ing the results from Z2, Z3 , and Z4.

– **IH:**

$$
\left[\begin{bmatrix} 1 \le \hat{i} \le k \ \textbf{ and} \\ [\hat{\alpha}]_1^k[\beta]_1^l \in \mathcal{B} \end{bmatrix} \ \textbf{implies} \ \begin{bmatrix} \Rightarrow(\hat{\alpha}_{\hat{i}} \,,\, [\hat{\alpha}]_1^k[\beta]_1^l) = [\hat{\alpha}]_1^{\hat{i}-1}[\hat{\alpha}]_{\hat{i}+1}^k\hat{\alpha}_{\hat{i}}[\beta]_1^l \\ \textbf{and } \Rightarrow(\hat{\alpha}_{\hat{i}} \,,\, [\hat{\alpha}]_1^k[\beta]_1^l) \in \mathcal{B} \\ \textbf{and } \Rightarrow(\hat{\alpha}_{\hat{i}} \,,\, [\hat{\alpha}]_1^k[\beta]_1^l) \approx [\hat{\alpha}]_1^k[\beta]_1^l \end{bmatrix}\right]
$$

$$
\textbf{for all } \hat{\alpha}_1 \,,\, \ldots \,,\, \hat{\alpha}_l \,,\, \hat{i} > i
$$

– **Step:** $1 \le i < k$. Assume:

B1 $\big[\hat{\alpha}_{i'} = \alpha_i \ \textbf{ for all } \big[1 \le i' \le i-1 \ \textbf{ and } \ i+2 \le i' \le k\big]\big]$ $\textbf{ and } \hat{\alpha}_i = \alpha_{i+1} \ \textbf{ and } \ \hat{\alpha}_{i+1} = \alpha_i$

B2 $\hat{i} = i + 1$

Observe:

Y1 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$. Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$
\begin{gathered}
\big[\big[i_1 \ne i_2 \ \textbf{ implies } \ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}\big] \ \textbf{ for all } \ 1 \le i_1 \,,\, i_2 \le k\big] \ \textbf{ and} \\
\big[\big[j_1 \ne j_2 \ \textbf{ implies } \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{ for all } \ 1 \le j_1 \,,\, j_2 \le l\big] \ \textbf{ and} \\
\big[\alpha_{i'} \not\Rrightarrow \beta_j \ \textbf{ for all } \ 1 \le i' \le k \,,\, 1 \le j \le l\big]
\end{gathered}
$$

Then, by applying B1, conclude:

$$
\begin{gathered}
\big[\big[i_1 \ne i_2 \ \textbf{ implies } \ \hat{\alpha}_{i_1} \not\Rightarrow \hat{\alpha}_{i_2}\big] \ \textbf{ for all } \ 1 \le i_1 \,,\, i_2 \le k\big] \ \textbf{ and} \\
\big[\big[j_1 \ne j_2 \ \textbf{ implies } \ \beta_{j_1} \asymp \beta_{j_2}\big] \ \textbf{ for all } \ 1 \le j_1 \,,\, j_2 \le l\big] \ \textbf{ and} \\
\big[\hat{\alpha}_{i'} \not\Rrightarrow \beta_j \ \textbf{ for all } \ 1 \le i' \le k \,,\, 1 \le j \le l\big]
\end{gathered}
$$

Then, by applying Definition 12 of $\mathcal{B}$, conclude $[\hat{\alpha}]_1^k[\beta]_1^l \in \mathcal{B}$. Then, by introducing $\boxed{\textbf{Step}}$, conclude $\big[1 \le i < k \ \textbf{ and } \ [\hat{\alpha}]_1^k[\beta]_1^l \in \mathcal{B}\big]$. Then, by basic rewriting, conclude $\left[1 \leq i+1 \leq k \textbf{ and } [\hat{\alpha}]_1^k[\beta]_1^l \in \mathcal{B}\right]$. Then, by applying B2, conclude $\left[\hat{i} > i \textbf{ and } 1 \leq \hat{i} \leq l \textbf{ and } [\hat{\alpha}]_1^k[\beta]_1^l \in \mathcal{B}\right]$. Then, by applying $\boxed{\textbf{IH}}$, conclude:

$$\begin{aligned}
&\Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l) = [\hat{\alpha}]_1^{\hat{i}-1}[\hat{\alpha}]_{\hat{i}+1}^k\hat{\alpha}_{\hat{i}}[\beta]_1^l\\
&\textbf{and } \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l) \in \mathcal{B}\\
&\textbf{and } \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l) \approx [\hat{\alpha}]_1^k[\beta]_1^l
\end{aligned}$$

Y2 Recall $\left[1 \leq i < k \textbf{ implies } \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) = \Rightarrow(\alpha_i\,,\,[\alpha]_1^{i-1}\alpha_{i+1}\alpha_i[\alpha]_{i+2}^k[\beta]_1^l)\right]$ from Definition 19. Then, by applying $\boxed{\textbf{Step}}$, conclude $\Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) = \Rightarrow(\alpha_i\,,\,[\alpha]_1^{i-1}\alpha_{i+1}\alpha_i[\alpha]_{i+2}^k[\beta]_1^l)$.

Y3 By equational reasoning, conclude:

$$\begin{aligned}
& \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l)\\
= \;& \textit{/* by applying Y2 */}\\
& \Rightarrow(\alpha_i\,,\,[\alpha]_1^{i-1}\alpha_{i+1}\alpha_i[\alpha]_{i+2}^k[\beta]_1^l)\\
= \;& \textit{/* by applying B1 */}\\
& \Rightarrow(\hat{\alpha}_{i+1}\,,\,[\hat{\alpha}]_1^{i-1}\hat{\alpha}_i\hat{\alpha}_{i+1}[\hat{\alpha}]_{i+2}^k[\beta]_1^l)\\
= \;& \textit{/* by applying B2 */}\\
& \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^{\hat{i}-2}\hat{\alpha}_{\hat{i}-1}\hat{\alpha}_{\hat{i}}[\hat{\alpha}]_{\hat{i}+1}^k[\beta]_1^l)\\
= \;& \textit{/* by collapsing */}\\
& \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l)
\end{aligned}$$

Y4 By equational reasoning, conclude:

$$\begin{aligned}
& \Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l)\\
= \;& \textit{/* by applying Y3 */}\\
& \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l)\\
= \;& \textit{/* by applying Y1 */}\\
& [\hat{\alpha}]_1^{\hat{i}-1}[\hat{\alpha}]_{\hat{i}+1}^k\hat{\alpha}_{\hat{i}}[\beta]_1^l\\
= \;& \textit{/* by applying B2 */}\\
& [\hat{\alpha}]_1^i[\hat{\alpha}]_{i+2}^k\hat{\alpha}_{i+1}[\beta]_1^l\\
= \;& \textit{/* by unfolding */}\\
& [\hat{\alpha}]_1^{i-1}\hat{\alpha}_i[\hat{\alpha}]_{i+2}^k\hat{\alpha}_{i+1}[\beta]_1^l\\
= \;& \textit{/* by applying B1 */}\\
& [\alpha]_1^{i-1}\alpha_{i+1}[\alpha]_{i+2}^k\alpha_i[\beta]_1^l\\
= \;& \textit{/* by collapsing */}\\
& [\alpha]_1^{i-1}[\alpha]_{i+1}^k\alpha_i[\beta]_1^l
\end{aligned}$$

Y5 Recall $\Rightarrow(\hat{\alpha}_{\hat{i}}\,,\,[\hat{\alpha}]_1^k[\beta]_1^l) \in \mathcal{B}$ from Y1. Then, by applying Y3, conclude $\Rightarrow(\alpha_i\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$.

Y6 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$. Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$\begin{array}{c}\big[[i_1 \neq i_2 \ \textbf{implies}\ \alpha_{i_1} \not\Rightarrow \alpha_{i_2}] \ \textbf{for all}\ 1 \leq i_1\,,\, i_2 \leq k\big] \\ \textbf{and}\ \big[\alpha_{i'} \not\Rightarrow \beta_j \ \textbf{for all}\ 1 \leq i' \leq k\,,\, 1 \leq j \leq l\big]\end{array}$$

Then, by basic rewriting, conclude:

$$\begin{array}{c}\alpha_i \not\Rightarrow \alpha_{i+1}\,,\, \alpha_{i+2}\,,\, \ldots\,,\, \alpha_k\,,\, \beta_1\,,\, \ldots\,,\, \beta_l \\ \textbf{and}\ \alpha_{i+1} \not\Rightarrow \alpha_i\,,\, \alpha_{i+2}\,,\, \ldots\,,\, \alpha_k\,,\, \beta_1\,,\, \ldots\,,\, \beta_l\end{array}$$

Then, by Corollary 1, conclude $\big[\alpha_i \not\Rightarrow \alpha_{i+1}\,,\, [\alpha]_{i+1}^k[\beta]_1^l$ **and** $\alpha_{i+1} \not\Rightarrow \alpha_i\,,\, [\alpha]_{i+1}^k[\beta]_1^l\big]$. Then, by applying Lemma 14, conclude $\alpha_i\alpha_{i+1}[\alpha]_{i+1}^k[\beta]_1^l = \alpha_{i+1}\alpha_i[\alpha]_{i+1}^k[\beta]_1^l$.

Y7 By equational reasoning, conclude:

$$\begin{array}{ll} & \Rightarrow(\alpha_i\,,\, [\alpha]_1^k[\beta]_1^l) \\ = & \text{/* } \textit{by applying Y3} \text{ */} \\ & \Rightarrow(\hat{\alpha}_{\hat{i}}\,,\, [\hat{\alpha}]_1^k[\beta]_1^l) \\ \approx & \text{/* } \textit{by applying Y1} \text{ */} \\ & [\hat{\alpha}]_1^k[\beta]_1^l \\ = & \text{/* } \textit{by unfolding} \text{ */} \\ & [\hat{\alpha}]_1^{i-1}\hat{\alpha}_i\hat{\alpha}_{i+1}[\hat{\alpha}]_{i+2}^k[\beta]_1^l \\ = & \text{/* } \textit{by applying B1} \text{ */} \\ & [\alpha]_1^{i-1}\alpha_{i+1}\alpha_i[\alpha]_{i+2}^k[\beta]_1^l \\ \approx & \text{/* } \textit{by applying Y6} \text{ */} \\ & [\alpha]_1^{i-1}\alpha_i\alpha_{i+1}[\alpha]_{i+2}^k[\beta]_1^l \\ = & \text{/* } \textit{by collapsing} \text{ */} \\ & [\alpha]_1^k[\beta]_1^l \end{array}$$

Conclude the consequent of this lemma by **and**-ing the results from Y4, Y5, and Y7.

□

Proof (of Theorem 3). Assume:

A1 $1 \leq j \leq l$

A2 $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$

A3 $A = \mathsf{Slave}(\beta_j\,,\, \{\alpha_1\,,\, \ldots\,,\, \alpha_k\})$

A4 $B = (\bigcup_{\alpha \in A} \mathsf{Master}(\alpha, \{\beta_1\,,\, \ldots\,,\, \beta_l\})) \setminus \{\beta_j\}$

A5 $A_0 = \{\alpha_1\,,\, \ldots\,,\, \alpha_k\}$

(A6) $B_0 = \{\beta_1, \dots, \beta_l\}$

Observe:

(Z1) Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from (A2). Then, by applying Definition 12 of $\mathcal{B}$, conclude $\big[\alpha_i \Rrightarrow \beta_j$ **for all** $1 \le i \le k\,, 1 \le j \le l\big]$. Then, by applying Definition 10 of $\Rrightarrow$, conclude $\big[\alpha_i\,, \beta_j \in \mathbb{P}\mathrm{A}$ **for all** $1 \le i \le k\,, 1 \le j \le l\big]$. Then, by rewriting under ZFC, conclude $\{\alpha_1, \dots, \alpha_k\} \cup \{\beta_1, \dots, \beta_l\} \subseteq \mathbb{P}\mathrm{A}$. Then, by applying (A5), conclude $A_0 \cup \{\beta_1 \dots, \beta_l\} \subseteq \mathbb{P}\mathrm{A}$. Then, by applying (A6), conclude $A_0 \cup B_0 \subseteq \mathbb{P}\mathrm{A}$. Then, by rewriting under ZFC, conclude $\big[A_0 \subseteq \mathbb{P}\mathrm{A}$ **and** $B_0 \subseteq \mathbb{P}\mathrm{A}\big]$.

(Z2) Recall $\big[A_0 \subseteq \mathbb{P}\mathrm{A}$ **and** $B_0 \subseteq \mathbb{P}\mathrm{A}\big]$. from (Z1). Then, by applying (A6), conclude $\big[A_0 \subseteq \mathbb{P}\mathrm{A}$ **and** $\{\beta_1, \dots, \beta_l\} \subseteq \mathbb{P}\mathrm{A}\big]$. Then, by introducing (A1), conclude $\big[1 \le j \le l$ **and** $A_0 \subseteq \mathbb{P}\mathrm{A}$ **and** $\{\beta_1, \dots, \beta_l\} \subseteq \mathbb{P}\mathrm{A}\big]$. Then, by rewriting under ZFC, conclude $\big[A_0 \subseteq \mathbb{P}\mathrm{A}$ **and** $\beta_j \in \mathbb{P}\mathrm{A}\big]$. Then, by applying Proposition 11, conclude $\mathsf{Slave}(\beta_j\,, A_0) \subseteq A_0$. Then, by applying (A3), conclude $A \subseteq A_0$.

(Z3) Recall $A \subseteq A_0$ from (Z2). Then, by applying (Z1), conclude $A \subseteq \mathbb{P}\mathrm{A}$. Then, by rewriting under ZFC, conclude $\big[\big[\alpha \in A$ **implies** $\alpha \in \mathbb{P}\mathrm{A}\big]$ **for all** $\alpha\big]$. Then, by introducing (Z1), conclude $\big[\big[\alpha \in A$ **implies** $\big[\alpha \in \mathbb{P}\mathrm{A}$ **and** $B_0 \subseteq \mathbb{P}\mathrm{A}\big]\big]$ **for all** $\alpha\big]$. Then, by applying Proposition 11, conclude $\big[\big[\alpha \in A$ **implies** $\mathsf{Master}(\alpha\,, B_0) \subseteq B_0\big]$ **for all** $\alpha\big]$. Then, by rewriting under ZFC, conclude $\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0) \subseteq B_0$. Then, by applying (A6), conclude $\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0) \subseteq \{\beta_1, \dots, \beta_l\}$. Then, by rewriting under ZFC, conclude $(\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0)) \setminus \{\beta_j\} \subseteq \{\beta_1, \dots, \beta_l\}$. Then, by introducing (A1), conclude $\big[1 \le j \le l$ **and** $(\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0)) \setminus \{\beta_j\} \subseteq \{\beta_1, \dots, \beta_l\}\big]$. Then, by rewriting under ZFC, conclude $(\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0)) \setminus \{\beta_j\} \subset \{\beta_1, \dots, \beta_l\}$. Then, by applying (A4), conclude $B \subset B_0$.

(Z4) Reasoning to a generalization, suppose:

$$\big[\alpha \in A \text{ **and** } \beta \in B_0 \setminus (B \cup \{\beta_j\})\big] \textbf{ for some } \alpha\,, \beta$$

Then, by rewriting under ZFC, conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $\beta \notin (B \cup \{\beta_j\})\big]$. Then, by applying (A4), conclude:

$$\begin{gathered} \alpha \in A \textbf{ and } \beta \in B_0 \textbf{ and} \\ \beta \notin ((\textstyle\bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, \{\beta_1, \dots, \beta_l\})) \setminus \{\beta_j\}) \cup \{\beta_j\} \end{gathered}$$

Then, by rewriting under ZFC, conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $\beta \notin \bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, \{\beta_1, \dots, \beta_l\})\big]$. Then, by applying (A6), conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $\beta \notin \bigcup_{\alpha\in A} \mathsf{Master}(\alpha\,, B_0)\big]$. Then, by rewriting under ZFC, conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $\beta \notin \mathsf{Master}(\alpha\,, B_0)\big]$. Then, by Definition 20 of Master, conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $\big[\beta \notin B_0$ **or** $\big[$**not** $\alpha \mapsto \beta\big]\big]\big]$. Then, by basic rewriting, conclude:

$$\begin{gathered} \big[\alpha \in A \textbf{ and } \beta \in B_0 \textbf{ and } \beta \notin B_0\big] \\ \textbf{or } \big[\alpha \in A \textbf{ and } \beta \in B_0 \textbf{ and } \big[\textbf{not } \alpha \mapsto \beta\big]\big] \end{gathered}$$

Then, by rewriting under ZFC, conclude $\big[[\alpha \in A$ **and** **false**$]$ **or** $[\alpha \in A$ **and** $\beta \in B_0$ **and** $[$**not** $\alpha \mapsto \beta]]\big]$. Then, by basic rewriting, conclude $\big[$**false** **or** $[\alpha \in A$ **and** $\beta \in B_0$ **and** $[$**not** $\alpha \mapsto \beta]]\big]$. Then, by basic rewriting, conclude $\big[\alpha \in A$ **and** $\beta \in B_0$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by introducing (A2), conclude $\big[[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ **and** $\alpha \in A$ **and** $\beta \in B_0$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by applying Definition 12 of $\mathcal{B}$, conclude:

$$\begin{gathered}\big[\alpha_i \Mapsto \beta_j \ \textbf{for all}\ 1 \le i \le k\,,\, 1 \le j \le l\big] \ \textbf{and} \\ \alpha \in A \ \textbf{and}\ \beta \in B_0 \ \textbf{and}\ \big[\textbf{not}\ \alpha \mapsto \beta\big]\end{gathered}$$

Then, by applying (A5), conclude

$$\begin{gathered}\big[\alpha_i \Mapsto \beta_{j'} \ \textbf{for all}\ 1 \le i \le k\,,\, 1 \le j' \le l\big] \ \textbf{and} \\ \alpha \in \{\alpha_1\,,\, \ldots \alpha_k\} \ \textbf{and}\ \beta \in B_0 \ \textbf{and}\ \big[\textbf{not}\ \alpha \mapsto \beta\big]\end{gathered}$$

Then, by rewriting under ZFC, conclude $\big[[\alpha \Mapsto \beta_{j'}$ **for all** $1 \le j' \le l]$ **and** $\beta \in B_0$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by applying (A6), conclude $\big[[\alpha \Mapsto \beta_{j'}$ **for all** $1 \le j' \le l]$ **and** $\beta \in \{\beta_1\,,\, \ldots\,,\, \beta_l\}$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by basic rewriting, conclude $\big[\alpha \Mapsto \beta$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by applying Definition 10 of $\Mapsto$, conclude $\big[[\alpha \asymp \beta$ **or** $\alpha \mapsto \beta]$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by basic rewriting, conclude $\big[[\alpha \asymp \beta$ **and** $[$**not** $\alpha \mapsto \beta]]$ **or** $[\alpha \mapsto \beta$ **and** $[$**not** $\alpha \mapsto \beta]]\big]$. Then, by basic rewriting, conclude $\big[[\alpha \asymp \beta$ **and** $[$**not** $\alpha \mapsto \beta]]$ **or** **false**$\big]$. Then, by basic rewriting, conclude $\big[\alpha \asymp \beta$ **and** $[$**not** $\alpha \mapsto \beta]\big]$. Then, by basic rewriting, conclude $\alpha \asymp \beta$. Then, by generalizing the premise, conclude $\big[[[\alpha \in A$ **and** $\beta \in B_0 \setminus (B \cup \{\beta_j\})]$ **implies** $\alpha \asymp \beta]$ **for all** $\alpha\,,\, \beta\big]$.

(Z5) Recall $A \subseteq A_0$ from (Z2). Then, by rewriting under ZFC, conclude $|A| \le |A_0|$. Then, by applying (A5), conclude $|A| \le |\{\alpha_1\,,\, \ldots\,,\, \alpha_k\}|$. Then, by rewriting under ZFC, conclude $|A| \le k$.

(Z6) Recall $B \subset B_0$ from (Z3). Then, by rewriting under ZFC, conclude $|B| < |B_0|$. Then, by applying (A6), conclude $|B| < |\{\beta_1\,,\, \ldots\,,\, \beta_l\}|$. Then, by rewriting under ZFC, conclude $|B| < l$. Then, by basic rewriting, conclude $1 \le |B| + 1 \le l$.

Assume:

(B1) $[\alpha']_1^{k-|A|} = [\alpha]_1^k{\downarrow_{\{\alpha_1,\ldots,\alpha_k\}\setminus A}}$ **and** $[\alpha']_{k-|A|+1}^k = A(|A|)\cdots A(1)$

(B2) $\beta'_1 = \beta_j$ **and** $[\beta']_2^j = [\beta]_1^{j-1}$ **and** $[\beta']_{j+1}^l = [\beta]_{j+1}^l$

(B3) $[\beta'']_1^{|B|} = B(|B|)\cdots B(1)$ **and** $[\beta'']_{|B|+1}^l = [\beta']_1^l{\downarrow_{\{\beta'_1,\ldots,\beta'_l\}\setminus B}}$

(B4) $A'_0 = \{\alpha'_1\,,\, \ldots\,,\, \alpha'_k\}$

(B5) $B'_0 = \{\beta'_1\,,\, \ldots\,,\, \beta'_l\}$

(B6) $B''_0 = \{\beta''_1\,,\, \ldots\,,\, \beta''_l\}$

Observe:

Y1 Recall $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}$ from A2. Then, by introducing A1, conclude $\big[1 \leq j \leq l$ **and** $[\alpha]_1^k[\beta]_1^l \in \mathcal{B}\big]$. Then, by applying Lemma 18, conclude:

$$\begin{aligned}&\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k\beta_j[\beta]_1^{j-1}[\beta]_{j+1}^l\\ \textbf{and}\ &\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}\\ \textbf{and}\ &\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l\end{aligned}$$

Then, by applying B2, conclude:

$$\begin{aligned}&\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = [\alpha]_1^k[\beta']_1^l\\ \textbf{and}\ &\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}\\ \textbf{and}\ &\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l\end{aligned}$$

Y2 By equational reasoning, conclude:

$$\begin{aligned}&B_0\\ =\ &\text{/* \textit{by applying} A6 */}\\ &\{\beta_1\,,\,\ldots\,,\,\beta_l\}\\ =\ &\text{/* \textit{by applying} B2 */}\\ &\{\beta'_1\,,\,\ldots\,,\,\beta'_l\}\\ =\ &\text{/* \textit{by applying} B5 */}\\ &B'_0\end{aligned}$$

Y3 Recall $\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$ from Y1. Then, by applying Y1, conclude $[\alpha]_1^k[\beta']_1^l \in \mathcal{B}$. Then, by introducing Z3, conclude $\big[B \subset B_0$ **and** $[\alpha]_1^k[\beta']_1^l \in \mathcal{B}\big]$. Then, by rewriting under ZFC, conclude $\big[B \subseteq B_0$ **and** $[\alpha]_1^k[\beta']_1^l \in \mathcal{B}\big]$. Then, by applying Y2, conclude $\big[B \subseteq B'_0$ **and** $[\alpha]_1^k[\beta']_1^l \in \mathcal{B}\big]$. Then, by applying B5, conclude $\big[B \subseteq \{\beta'_1\,,\,\ldots\,,\,\beta'_l\}$ **and** $[\alpha]_1^k[\beta']_1^l \in \mathcal{B}\big]$. Then, by applying Corollary 5, conclude:

$$\begin{aligned}&\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) = [\alpha]_1^k B(|B|)\cdots B(1)[\beta']_1^k{\downarrow_{\{\beta'_1,\ldots,\beta'_k\}\setminus B}}\\ \textbf{and}\ &\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \in \mathcal{B}\\ \textbf{and}\ &\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \approx [\alpha]_1^k[\beta']_1^l\end{aligned}$$

Then, by applying B3, conclude:

$$\begin{aligned}&\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) = [\alpha]_1^k[\beta'']_1^l\\ \textbf{and}\ &\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \in \mathcal{B}\\ \textbf{and}\ &\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \approx [\alpha]_1^k[\beta']_1^l\end{aligned}$$

Y4 By equational reasoning, conclude:

$$
\begin{array}{cl}
& B'_0 \\
= & \textit{/* by applying (B5) */} \\
& \{\beta'_1\,,\, \ldots\,,\, \beta'_l\} \\
= & \textit{/* by applying (B3) */} \\
& \{\beta''_1\,,\, \ldots\,,\, \beta''_l\} \\
= & \textit{/* by applying (B6) */} \\
& B''_0
\end{array}
$$

(Y5) Recall $\Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \in \mathcal{B}$ from (Y3). Then, by applying (Y3), conclude $[\alpha]_1^k[\beta'']_1^l \in \mathcal{B}$. Then, by introducing (Z2), conclude $\big[A \subseteq A_0$ **and** $[\alpha]_1^k[\beta'']_1^l \in \mathcal{B}\big]$. Then, by applying (A5), conclude $\big[A \subseteq \{\alpha_1\,,\, \ldots\,,\, \alpha_k\}$ **and** $[\alpha]_1^k[\beta'']_1^l \in \mathcal{B}\big]$. Then, by applying Corollary 6, conclude:

$$
\begin{array}{l}
\Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) = [\alpha]_1^k\!\downarrow_{\{\alpha_1,\ldots,\alpha_k\}\setminus A} A(|A|)\cdots A(1)[\beta'']_1^l \\
\textbf{and } \Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \in \mathcal{B} \\
\textbf{and } \Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \approx [\alpha]_1^k[\beta'']_1^l
\end{array}
$$

Then, by applying (B1), conclude:

$$
\begin{array}{l}
\Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) = [\alpha']_1^k[\beta'']_1^l \\
\textbf{and } \Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \in \mathcal{B} \\
\textbf{and } \Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \approx [\alpha]_1^k[\beta'']_1^l
\end{array}
$$

(Y6) By equational reasoning, conclude:

$$
\begin{array}{cl}
& A_0 \\
= & \textit{/* by applying (A5) */} \\
& \{\alpha_1\,,\, \ldots\,,\, \alpha_k\} \\
= & \textit{/* by applying (B1) */} \\
& \{\alpha'_1\,,\, \ldots\,,\, \alpha'_k\} \\
= & \textit{/* by applying (B4) */} \\
& A'_0
\end{array}
$$

Assume:

(C1) $[\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l = \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l)))$

Observe:

(X1) Recall $[\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l = \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l)))$ from (C1). Then, by introducing (A4), conclude:

$$
\begin{array}{c}
B = \bigcup_{\alpha\in A} \mathsf{Master}(\alpha, \{\beta_1\,,\, \ldots\,,\, \beta_l\}) \\
[\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l = \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l)))
\end{array}
$$

Then, by introducing A3, conclude:

$$\begin{aligned} A &= \mathsf{Slave}(\beta_j\,,\,\{\alpha_1\,,\,\ldots\,,\,\alpha_k\}) \\ B &= \textstyle\bigcup_{\alpha\in A}\mathsf{Master}(\alpha,\{\beta_1\,,\,\ldots\,,\,\beta_l\}) \\ [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l &= \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l))) \end{aligned}$$

Then, by introducing A2, conclude:

$$[\alpha]_1^k[\beta]_1^l \in \mathcal{B} \ \ \textbf{and} \ \ \left[\begin{aligned} A &= \mathsf{Slave}(\beta_j\,,\,\{\alpha_1\,,\,\ldots\,,\,\alpha_k\}) \\ B &= \textstyle\bigcup_{\alpha\in A}\mathsf{Master}(\alpha,\{\beta_1\,,\,\ldots\,,\,\beta_l\}) \\ [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l &= \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l))) \end{aligned}\right]$$

Then, by introducing A1, conclude:

$$\left[\begin{array}{c} 1 \le j \le l \ \ \textbf{and} \\ [\alpha]_1^k[\beta]_1^l \in \mathcal{B} \end{array}\right] \ \ \textbf{and} \ \ \left[\begin{aligned} A &= \mathsf{Slave}(\beta_j\,,\,\{\alpha_1\,,\,\ldots\,,\,\alpha_k\}) \\ B &= \textstyle\bigcup_{\alpha\in A}\mathsf{Master}(\alpha,\{\beta_1\,,\,\ldots\,,\,\beta_l\}) \\ [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l &= \Rrightarrow(A\,,\,\Lleftarrow(B\,,\,\Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l))) \end{aligned}\right]$$

Then, by applying Definition 21 of $\Downarrow$, conclude $\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) = [\tilde{\alpha}]_1^{k-|A|}$ $([\tilde{\alpha}]_{k-|A|+1}^k[\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l$.

X2 By equational reasoning, conclude:

$$\begin{aligned} & [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l \\ = \ & \textit{/* by applying X3 */} \\ & [\alpha']_1^k[\beta'']_1^l \\ = \ & \textit{/* by applying Y5 */} \\ & \Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \\ \approx \ & \textit{/* by applying Y5 */} \\ & [\alpha]_1^k[\beta'']_1^l \\ = \ & \textit{/* by applying Y3 */} \\ & \Lleftarrow(B\,,\,[\alpha]_1^k[\beta']_1^l) \\ \approx \ & \textit{/* by applying Y3 */} \\ & [\alpha]_1^k[\beta']_1^l \\ = \ & \textit{/* by applying Y1 */} \\ & \Leftarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \\ \approx \ & \textit{/* by applying Y1 */} \\ & [\alpha]_1^k[\beta]_1^l \end{aligned}$$

X3 By equational reasoning, conclude:

$$
\begin{aligned}
& [\tilde{\alpha}]_1^k [\tilde{\beta}]_1^l \\
= \;& \text{/* \textit{by applying} C1 */} \\
& \Rrightarrow(A\,,\, \Lleftarrow(B\,,\, \Leftarrow(\beta_j\,,\, [\alpha]_1^k [\beta]_1^l))) \\
= \;& \text{/* \textit{by applying} Y1 */} \\
& \Rrightarrow(A\,,\, \Lleftarrow(B\,,\, [\alpha]_1^k [\beta']_1^l)) \\
= \;& \text{/* \textit{by applying} Y3 */} \\
& \Rrightarrow(A\,,\, [\alpha]_1^k [\beta'']_1^l) \\
= \;& \text{/* \textit{by applying} Y5 */} \\
& [\alpha']_1^k [\beta'']_1^l
\end{aligned}
$$

X4 Recall $\Rrightarrow(A\,,\,[\alpha]_1^k[\beta'']_1^l) \in \mathcal{B}$ from Y5. Then, by applying Y5, conclude $[\alpha']_1^k[\beta'']_1^l \in \mathcal{B}$. Then, by applying X3, conclude $[\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l \in \mathcal{B}$. Then, by introducing Z6, conclude $\big[1 \leq |B|+1 \leq l \;\;\textbf{and}\;\; [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l \in \mathcal{B}\big]$. Then, by applying Corollary 3, conclude $\big[[\tilde{\alpha}]_1^k([\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \in \mathcal{B} \;\;\textbf{and}\;\; [\tilde{\alpha}]_1^k([\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \approx [\alpha]_1^k[\beta]_1^l\big]$.

Recall $\big[\big[\big[\alpha \in A \;\;\textbf{and}\;\; \beta \in B_0 \setminus (B \cup \{\beta_j\})\big] \;\;\textbf{implies}\;\; \alpha \asymp \beta\big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta\big]$ from Z4. Then, by rewriting under ZFC, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{A(1)\,,\,\dots\,,\,A(|A|)\} \\ \textbf{and}\;\; \beta \in B_0 \setminus (B \cup \{\beta_j\}) \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by applying B1, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \\ \textbf{and}\;\; \beta \in B_0 \setminus (B \cup \{\beta_j\}) \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by applying Y2, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \\ \textbf{and}\;\; \beta \in B'_0 \setminus (B \cup \{\beta_j\}) \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by rewriting under ZFC, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \\ \textbf{and}\;\; \beta \in (B'_0 \setminus \{\beta_j\}) \setminus B \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by applying B5, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \;\;\textbf{and} \\ \beta \in (\{\beta'_1\,,\,\dots\,,\,\beta'_l\} \setminus \{\beta_j\}) \setminus B \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by applying B2, conclude:

$$\Big[\begin{bmatrix} \alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \;\;\textbf{and} \\ \beta \in (\{\beta'_1\,,\,\dots\,,\,\beta'_l\} \setminus \{\beta'_1\}) \setminus B \end{bmatrix} \;\;\textbf{implies}\;\; \alpha \asymp \beta\Big] \;\;\textbf{for all}\;\; \alpha\,,\,\beta$$

Then, by rewriting under ZFC, conclude:

$$\Big[\begin{bmatrix}\alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \;\textbf{and}\\ \beta \in \{\beta'_2\,,\,\dots\,,\,\beta'_l\} \setminus B\end{bmatrix} \;\textbf{implies}\; \alpha \asymp \beta\Big] \;\textbf{for all}\; \alpha\,,\,\beta$$

Then, by basic rewriting, conclude:

$$\Big[\begin{bmatrix}\alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \;\textbf{and}\\ \beta \in \{\beta'_2{\downarrow}_{\{\beta'_2\,,\,\dots\,,\,\beta'_l\}\setminus B}\,,\,\dots\,,\,\beta'_l{\downarrow}_{\{\beta'_2\,,\,\dots\,,\,\beta'_l\}\setminus B}\}\end{bmatrix} \;\textbf{implies}\; \alpha \asymp \beta\Big] \;\textbf{for all}\; \alpha\,,\,\beta$$

Then, by rewriting under ZFC, conclude:

$$\Big[\begin{bmatrix}\alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\} \;\textbf{and}\\ \beta \in \{\beta'_2{\downarrow}_{\{\beta'_1\,,\,\dots\,,\,\beta'_l\}\setminus B}\,,\,\dots\,,\,\beta'_l{\downarrow}_{\{\beta'_1\,,\,\dots\,,\,\beta'_l\}\setminus B}\}\end{bmatrix} \;\textbf{implies}\; \alpha \asymp \beta\Big] \;\textbf{for all}\; \alpha\,,\,\beta$$

Then, by applying B3, conclude:

$$\Big[\begin{bmatrix}\alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\}\\ \textbf{and}\;\beta \in \{\beta''_{|B|+2}\,,\,\dots\,,\,\beta''_l\}\end{bmatrix} \;\textbf{implies}\; \alpha \asymp \beta\Big] \;\textbf{for all}\; \alpha\,,\,\beta$$

Then, by basic rewriting, conclude $\big[\big[\alpha \in \{\alpha'_{k-|A|+1}\,,\,\dots\,,\,\alpha'_k\}$ **implies** $\alpha \asymp \beta''_{|B|+2}\,,\,\dots\,,\,\beta''_l\big]$ **for all** $\alpha\big]$. Then, by basic rewriting, conclude $\big[\alpha'_{i'} \asymp \beta''_{|B|+2}\,,\,\dots\,,\,\beta''_l$ **for all** $k-|A|+1 \le i' \le k\big]$. Then, by applying X3, conclude $\big[\tilde{\alpha}_{i'} \asymp \tilde{\beta}_{|B|+2}\,,\,\dots\,,\,\tilde{\beta}_l$ **for all** $k-|A|+1 \le i' \le k\big]$. Then, by introducing X4, conclude:

$$[\tilde{\alpha}]_1^k([\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \in \mathcal{B} \;\textbf{and}\; \begin{bmatrix}\tilde{\alpha}_{i'} \asymp \tilde{\beta}_{|B|+2}\,,\,\dots\,,\,\tilde{\beta}_l\\ \textbf{for all}\; k-|A|+1 \le i' \le k\end{bmatrix}$$

Then, by introducing Z5, conclude:

$$|A| \le k \;\textbf{and}\; [\tilde{\alpha}]_1^k([\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \in \mathcal{B} \;\textbf{and}\; \begin{bmatrix}\tilde{\alpha}_{i'} \asymp \tilde{\beta}_{|B|+2}\,,\,\dots\,,\,\tilde{\beta}_l\\ \textbf{for all}\; k-|A|+1 \le i' \le k\end{bmatrix}$$

Then, by applying Corollary 4, conclude:

$$\begin{array}{l}[\tilde{\alpha}]_1^{k-|A|}([\tilde{\alpha}]_{k-|A|+1}^k[\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \in \mathcal{B} \;\textbf{and}\\ [\tilde{\alpha}]_1^{k-|A|}([\tilde{\alpha}]_{k-|A|+1}^k[\tilde{\beta}]_1^{|B|+1})[\tilde{\beta}]_{|B|+2}^l \approx [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l\end{array}$$

Then, by applying X1, conclude $\big[\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$ **and** $\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\tilde{\alpha}]_1^k[\tilde{\beta}]_1^l\big]$. Then, by applying X2, conclude $\big[\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \in \mathcal{B}$ **and** $\Downarrow(\beta_j\,,\,[\alpha]_1^k[\beta]_1^l) \approx [\alpha]_1^k[\beta]_1^l\big]$. $\square$