



HAL
open science

Routing Quantum Control of Causal Order

Maarten Grothus, Alastair A. Abbott, Augustin Vanrietvelde, Cyril Branciard

► **To cite this version:**

Maarten Grothus, Alastair A. Abbott, Augustin Vanrietvelde, Cyril Branciard. Routing Quantum Control of Causal Order. 2025. <hal-05266755>

HAL Id: hal-05266755

<https://inria.hal.science/hal-05266755v1>

Preprint submitted on 18 Sep 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Routing Quantum Control of Causal Order

Maarten Grothuis^{1,2}, Alastair A. Abbott¹, Augustin Vanrietvelde³, and Cyril Branciard²

¹Univ. Grenoble Alpes, Inria, 38000 Grenoble, France

²Univ. Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, 38000 Grenoble, France

³Télécom Paris-LTCI, Inria, Institut Polytechnique de Paris, 91120 Palaiseau, France

In recent years, various frameworks have been proposed for the study of quantum processes with indefinite causal order. In particular, quantum circuits with quantum control of causal order (QC-QCs) form a broad class of physical supermaps obtained from a bottom-up construction and are believed to represent all quantum processes physically realisable in a fixed spacetime. Complementarily, the formalism of routed quantum circuits introduces quantum operations constrained by “routes” to represent processes in terms of a more fine-grained routed circuit decomposition. This decomposition, formalised using a so-called routed graph, represents the information flow within the respective process. However, the existence of routed circuit decompositions has only been established for a small set of processes so far, including both certain specific QC-QCs and more exotic processes as examples.

In this work, we remedy this fact by connecting these two frameworks. We prove that for any given N , one can use a single routed graph to systematically obtain a routed circuit decomposition for any QC-QC with N parties. We detail this construction explicitly and contrast it with other routed circuit decompositions of QC-QCs, which we obtain from alternative routed graphs. We conclude by pointing out how this connection can be useful to tackle various open problems in the field of indefinite causal order, particularly establishing circuit representations of subclasses of QC-QCs.

Contents

1	Introduction	3
2	Consistent Routed Quantum Circuits	4
2.1	Preliminaries: relations	4
2.2	Routed graph	7
2.3	Branch graph	10
2.4	Routed linear maps	13
2.5	Skeletal supermap and fleshing out for a routed graph	14
2.6	Main theorem of Vanrietvelde <i>et al.</i> [39]	17
3	Quantum Circuits with Quantum Control of Causal Order	19
3.1	Choi picture and the link product	20
3.2	Formal construction of QC-QCs	20
4	QC-QCs as Routed Quantum Circuits	21
4.1	Generic routed graph for QC-QCs	22
4.2	Generic branch graph for QC-QCs	27
4.3	Obtaining any QC-QC: Fleshing out the skeletal supermap for $\mathcal{G}_{\text{QC-QC}(N)}$	30
5	Alternative Routed Quantum Circuit Descriptions	36
5.1	An alternative treatment of the QC-QC ancillary systems	37
5.2	Splitting the internal nodes according to their branch structure	38
5.3	Merging nodes in the routed graph	42
5.4	Removing arrows in the routed graph	46
6	Discussion	47
6.1	Contrasting routed circuit decompositions	47
6.2	Characterising classes of processes	48
6.3	Consequences for the routed quantum circuit framework	49
7	Acknowledgements	49
A	Derivation of the choice function for $\mathcal{G}_{\text{QC-QC}(N)}$	53
B	Obtaining a QC-QC by fleshing out a skeletal supermap	54
C	Proofs for Section 5	56
D	Reconstruction of the routed circuit decomposition of the Grenoble process in [39]	59
E	Restricting to agent-localised processing	60

1 Introduction

Over the course of the last fifteen years, there has been an explosion of interest in studying the quantum world through the eye of causality. While we are used to making sense of our everyday experiences in terms of cause and effect – a perspective which has been well-formalised in the field of classical statistics [1–3], certain quantum phenomena defy any attempt to be explained in terms of classical causation [4]. However, it recently has been realised that the notion of causality can be adapted into the quantum realm, shifting from classical random variables to quantum systems as causal relata [5, 6]. This has led to the increasing adoption of causal perspectives and techniques in the fields of quantum foundations and quantum information theory. In the wake of this movement, a particularly interesting field of research that has emerged is that of indefinite causal order (ICO) in quantum processes, suggesting the necessity of cyclic quantum causal models for their description [7–13].

The top-down approach of *quantum supermaps* [14] (also known as higher-order maps and often formalised in the *process matrix framework* [15]) has been extensively used to model ICO phenomena. In this framework, one considers a fixed number of agents who, as parties in a fixed process, each execute a freely chosen quantum operation precisely once. The only condition imposed on the process is that any probabilities observed by these agents are logically consistent, but no global causal structure is assumed – as such a structure might no longer be in place, e.g., in a theory of quantum gravity. Such processes can, equivalently, be understood as the most general “higher-order” transformations that transform the parties’ local operations into a valid quantum operation between a common global past and a global future. Quantum supermaps are therefore of interest both from an informational perspective and as an extension of quantum theory into general relativistic regimes [7, 11, 13, 15–20].

Over the course of the study of ICO, several particularly interesting processes have been studied in detail, such as the quantum switch [21], in which the two possible orders of two operations are placed in a quantum superposition with the aid of a quantum control system, as well as more exotic processes such as the Oreshkov-Costa-Brukner (OCB) process [15], for which no clear physical interpretation has been suggested so far (and which has in fact been argued to be non-physical, on the basis that it is not “purifiable” [22]), and the so-called Lugano (or “AF/BW”) process [23, 24], which features ICO on classical systems. These examples and their respective properties have been the subject of intense study, and for the quantum switch [21, 25–27] and its natural generalisation to more operations [28–31], as well as for the Lugano process [32], computational and communication advantages beyond standard causally ordered quantum information processing have been identified. However, the top-down perspective of the process matrix framework does not provide any straightforward way to systematically establish further processes with interesting properties, or easily assess their physical implementability. Accordingly, working towards a better understanding of the internal structure and physical realisability of such processes is one of the major challenges of the field.

With this goal in mind, various constructive frameworks have been proposed [33–40]. Refs. [34, 39] in particular establish formal connections with quantum supermaps, providing an avenue to categorise processes into structurally different classes. However, research into the relation and connections between these approaches and their synergies has been rather limited so far, with a notable exception being Refs. [41, 42] which relate the *causal box* framework [33] to the generalised circuit framework of [34] in an effort to gauge the physical realisability of processes within a fixed spacetime.

This work contributes to closing this gap by connecting the framework of *quantum circuits with quantum control of causal order* (QC-QCs) [34] – physically realisable processes which feature coherent control of the order of operations, and whose properties as process matrices have been well-characterised – with *routed quantum circuits* [39, 43], which provide a decomposition of ICO processes into more fine-grained “routed” circuits, with precise and intuitive diagrammatic representations in which the process validity can be checked from the circuit structure, despite the presence of feedback loops. While for some notable examples of causally indefinite QC-QCs, such as the quantum switch and the so-called Grenoble process [34], explicit constructions as routed quantum circuits have been formulated [39], no systematic procedure has so far been established. Indeed, it has thus far remained unknown whether routed quantum circuits can represent all QC-

QCs, even though they can also represent some more exotic purifiable processes, such as the Lugano process [39].

Here we establish a generic procedure for obtaining a routed quantum circuit representation of any QC-QC. Specifically, for any given number of parties, we specify a single routed graph which will allow one to obtain any QC-QC with that number of parties, by using an appropriate “fleshing out”. In Section 2, we provide a detailed review of the framework of routed quantum circuits and how it allows for consistent decompositions of ICO processes, which we will require to demonstrate the general validity of our construction. We then briefly review the QC-QC framework in Section 3, before presenting our main results in Section 4, representing any QC-QC as a routed circuit: in Section 4.1, we propose the backbone of this representation in terms of a routed graph; after confirming the validity of this routed graph in Section 4.2, we detail the “fleshing out” of a skeletal supermap associated with this routed graph to any desired QC-QC in Section 4.3, and prove the equivalence of the resulting routed supermap to the original QC-QC construction. We then discuss some possible alternative routed circuit decompositions for QC-QCs in Section 5, before concluding in Section 6.

2 Consistent Routed Quantum Circuits

In this section we review the routed quantum circuit framework for ICO processes of Ref. [39], making a significant effort to propose a simplified yet formal presentation of the framework that slightly deviates from that of Ref. [39] without affecting the main results.

The broader goal of *routed quantum circuits* [39, 43] is to generalise quantum circuits so as to represent not only the tensor product structure of the Hilbert spaces involved in a circuit, but also their potential direct sum structure. For instance, considering two Hilbert spaces $\mathcal{H}^A = \mathcal{H}_0^A \oplus \mathcal{H}_1^A$ and $\mathcal{H}^B = \mathcal{H}_0^B \oplus \mathcal{H}_1^B$, that each decomposes into the direct sum of two *sectors* $\mathcal{H}_k^A, \mathcal{H}_l^B$, it may be relevant (and sufficient) to only refer to the composite sectorised space

$$\tilde{\mathcal{H}}^{AB} = \mathcal{H}_0^A \otimes \mathcal{H}_1^B \oplus \mathcal{H}_1^A \otimes \mathcal{H}_0^B, \quad (2.1)$$

rather than to the full space $\mathcal{H}^A \otimes \mathcal{H}^B \supseteq \tilde{\mathcal{H}}^{AB}$. Sectorised spaces like this are of pivotal importance in quantum optics for instance, to describe the transmission of a photon in a superposition of two trajectories A and B [44]. In the framework, this is modelled by the introduction of *routes*, relations between the sectors specifying which sectors of $\mathcal{H}^A \otimes \mathcal{H}^B$ may be mapped to each other through a given *routed channel*, and possibly rendering certain sectors inaccessible, as in Eq. (2.1) above.

To affirm the validity of a routed quantum circuit, it is sufficient to study its route structure, which can be specified by an underlying *routed graph*, which may generally be cyclic. Accordingly, we will initially focus on establishing the route structure in isolation, before turning to the circuits themselves.

After presenting some preliminaries on relations in Section 2.1, we will formally introduce the routed graph in Section 2.2, followed by the more fine-grained *branch graph* it implies in Section 2.3. These directed graphs need to satisfy certain properties to ensure that one can build consistent routed quantum circuits on top of them: the routed graph needs to satisfy the condition of *bi-univocality* which, informally, ensures each operation is performed precisely once, while the branch graph is only allowed to contain so-called *weak* loops, ensuring that these do not introduce any paradoxical feedback mechanism. In Section 2.4, we then introduce formally the notion of *routed linear maps*, before connecting it with the routed graph structure to obtain a *skeletal (routed) supermap* in Section 2.5, by associating each arrow in the routed graph to a respective Hilbert space. This skeletal supermap will then be fleshed out to a routed or unrouted supermap by composition with compatible routed channels or superchannels at each node. We conclude this section by recalling the main result of [39] in Section 2.6.

2.1 Preliminaries: relations

The framework heavily relies on binary relations, also called *routes*. Let us start by introducing these.

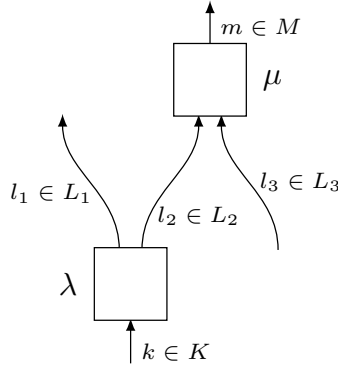


Figure 2.1: Two relations $\lambda : K \rightarrow L_1 \times L_2$ and $\mu : L_2 \times L_3 \rightarrow M$ can be composed through L_2 , resulting in a relation $\lambda * \mu : K \times L_3 \rightarrow L_1 \times M$ that can be written as $\lambda * \mu = \text{Tr}_{L_2}(\lambda \otimes \mu)$, as in Eq. (2.4).

A (binary) relation λ over some sets K and L will be denoted $\lambda : K \rightarrow L$. It can be represented as a Boolean matrix $(\lambda_k^l)_{k \in K}^{l \in L}$, where the lower and upper indices represent the inputs and outputs of the relation, respectively, and with $\lambda_k^l = 1$ if k and l are related through λ (which we write $k \overset{\lambda}{\sim} l$), and $\lambda_k^l = 0$ otherwise. Throughout, we will freely identify a relation with its Boolean matrix. A relation can also be defined by some constraint(s) on its inputs and outputs; e.g., the constraint “ $k = l$ ” would define the relation with matrix elements $\lambda_k^l = \delta_{k,l}$ (where δ is the Kronecker delta). Note also that the converse relation is represented by the transpose matrix λ^\top .

A relation $\lambda : K \rightarrow L$ can formally be seen as a function $K \rightarrow \mathcal{P}(L)$, where $\mathcal{P}(L)$ denotes the power set of L . We will thus use the notation $\lambda(k)$ to denote the set of all $l \in L$ that are related to k through λ . To characterise the subsets of K and L which are actually related by λ , we introduce the notion of a *practical (co)domain*.

Definition 2.1 (Practical (co)domain). The *practical domain* of a relation $\lambda : K \rightarrow L$ is defined as the subset $K_{\text{in}(\lambda), \text{prac}} := \{k \in K \mid \lambda(k) \neq \emptyset\}$. Similarly, the *practical codomain* of λ is defined as $L_{\text{out}(\lambda), \text{prac}} := \{l \in L \mid \lambda^\top(l) \neq \emptyset\}$.

2.1.1 Composing relations

To build up a routed graph we will need to compose relations with one another. Given two relations¹ $\lambda : K \rightarrow L_1 \times L_2$ and $\mu : L_2 \times L_3 \rightarrow M$, as illustrated in Fig. 2.1, we can define their composition through L_2 as the new relation $\lambda * \mu : K \times L_3 \rightarrow L_1 \times M$ that satisfies

$$(k, l_3) \overset{\lambda * \mu}{\sim} (l_1, m) \quad \text{if and only if} \quad \exists l_2 \in L_2 \text{ such that } k \overset{\lambda}{\sim} (l_1, l_2) \text{ and } (l_2, l_3) \overset{\mu}{\sim} m. \quad (2.2)$$

Representing the relations λ and μ as Boolean matrices $(\lambda_k^{l_1, l_2})_{k, l_1, l_2}$ and $(\mu_{l_2, l_3}^m)_{l_2, l_3, m}$, it is easy to verify² that $\lambda * \mu$ is represented by the Boolean matrix with elements

$$(\lambda * \mu)_{k, l_3}^{l_1, m} = \sum_{l_2} \lambda_k^{l_1, l_2} \mu_{l_2, l_3}^m, \quad (2.3)$$

where \sum denotes the Boolean sum, i.e. which evaluates to 1 if at least one of the summands is 1 (equivalent to the logical *or*). The composed relation can thus be written as a (Boolean) partial trace.³

$$\lambda * \mu = \text{Tr}_{L_2}(\lambda \otimes \mu). \quad (2.4)$$

¹The definition of composition that we present of course also applies when L_1 , L_2 and/or L_3 are trivial.

² $(\lambda * \mu)_{k, l_3}^{l_1, m} = 1 \iff \exists l_2, \lambda_k^{l_1, l_2} \mu_{l_2, l_3}^m = 1 \iff \sum_{l_2} \lambda_k^{l_1, l_2} \mu_{l_2, l_3}^m = 1$.

³This is somewhat similar to the “link product” [45, 46] (in its “pure version” [34]) used to compose quantum maps in their Choi representation.

Remark that composing two relations λ and μ in this manner may then generally result in a further restricted practical domain or co-domain in comparison to the individual relations. Specifically, in the case for instance where L_1 and L_3 are trivial, $K_{\text{in}(\lambda * \mu), \text{prac}}$ may be strictly contained in $K_{\text{in}(\lambda), \text{prac}}$ if $\lambda(k)$ does not relate to any $l_2 \in L_2$ within the practical domain of μ , and analogously for the practical co-domain $M_{\text{out}(\lambda * \mu), \text{prac}}$.

This way of composing relations can be generalised to more relations, connected through more common sets of values (i.e., more linking arrows in a graphical representations as in Fig. 2.1; see, e.g., Fig. 2.4 below). For the resulting relation to hold between a pair of its input and output values, it is required that there exists a value in each of the common sets, such that all individual relations hold. The resulting relation is still obtained by tensoring all relations and partially tracing out over all common sets of indices, thus defining a (commutative and associative) multi-term “link product”. We will in particular use this in defining the notion of a *choice relation* (Definition 2.7) in Section 2.2.2.

2.1.2 Branched relations

In the framework of routed circuits [39] we focus on a specific type of relations, the so-called *branched* relations.

Definition 2.2 (Branched relation). A relation $\lambda : K \rightarrow L$ is said to be *branched* if for any $k, k' \in K$, $\lambda(k)$ and $\lambda(k')$ are either the same, or disjoint.

Note that a relation λ is branched if and only if its converse relation λ^\top is branched. Branched relations define partitions of their (practical) domain and codomain that are related through so-called *branches*.

Definition 2.3 (Branch). A *branch* $\beta = (K^\beta \rightarrow L^\beta)$ of a branched relation $\lambda : K \rightarrow L$ is a pair of nonempty sets $K^\beta \subseteq K$ and $L^\beta \subseteq L$ such that for any $k \in K^\beta$, $\lambda(k) = L^\beta$ and for any $l \in L^\beta$, $\lambda^\top(l) = K^\beta$.

We denote by $\mathbf{Bran}(\lambda)$ the set of branches β of λ . Note that the corresponding partitions of K and L need not be complete, in the sense that $\bigsqcup_{\beta \in \mathbf{Bran}(\lambda)} K^\beta$ might not be equal to K and $\bigsqcup_{\beta \in \mathbf{Bran}(\lambda)} L^\beta$ might not be equal to L ; see Fig. 2.2. Rather, $\bigsqcup_{\beta \in \mathbf{Bran}(\lambda)} K^\beta$ is precisely the practical domain of λ , while $\bigsqcup_{\beta \in \mathbf{Bran}(\lambda)} L^\beta$ is its practical co-domain, as introduced in Definition 2.1 above.

Note that generally, if the partitions $\{L^\beta \mid \beta \in \mathbf{Bran}(\lambda)\}$ and $\{L^\gamma \mid \gamma \in \mathbf{Bran}(\mu)\}$ differ, the sequential composition of two branched relations $\lambda : K \rightarrow L$ and $\mu : L \rightarrow M$ will no longer be a branched relation.⁴

2.1.3 Augmenting relations

From a branched relation, we define its *augmented* version, which will be an important technical tool in determining if the composition of some relations can describe a valid circuit structure.

Definition 2.4 (Augmented relation⁵). Given a branched relation $\lambda : K \rightarrow L$, its *augmented* version is the partial function – written as a branched relation – $\lambda^{\text{aug}} : K \times \left(\times_{\beta \in \mathbf{Bran}(\lambda)} L^\beta \right) \rightarrow$

⁴However, this is not an if and only if condition, e.g., $\lambda * \mu$ is still branched if the partition of the domain of μ is a fine-graining of the partition of the co-domain of λ or vice versa.

⁵Note that we depart slightly here from the definition given in [39], which (equivalently) wrote the output of the partial function as $(l^{\vec{\beta}}, (\delta_{\beta, \vec{\beta}})_{\beta \in \mathbf{Bran}(\lambda)}) \in L \times \left(\times_{\beta \in \mathbf{Bran}(\lambda)} \mathbf{Happens}_\beta \right)$ (with each $\mathbf{Happens}_\beta \cong \{0, 1\}$). Since looking at all $\delta_{\beta, \vec{\beta}}$'s, for all $\beta \in \mathbf{Bran}(\lambda)$ individually, will not be relevant right away when we first use augmented relations (when constructing the “choice relation”, see Section 2.2.2), but will only be relevant later on when we need to clarify causal influences between branches (when constructing the “branch graph”, see Section 2.3), we will keep that alternative description of $\vec{\beta}$ for later and use the simpler expression of the augmented relations' outputs described here.

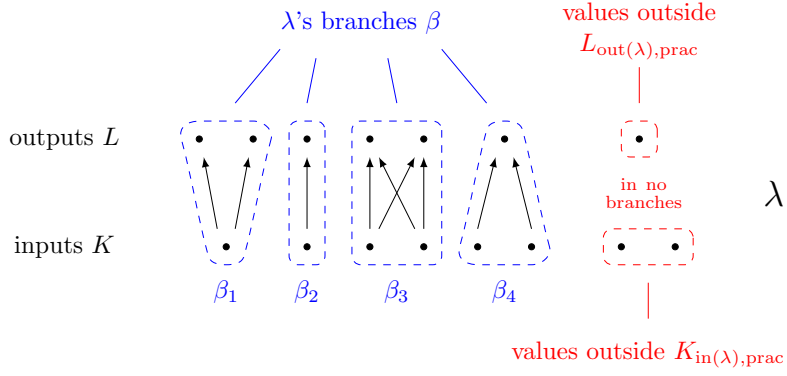


Figure 2.2: Example of a branched relation $\lambda : K \rightarrow L$, with its domain and codomain partitioned into branches (reproducing the example of Fig. 12 in [39]). If λ is not total or not surjective, then some values of K or L are outside of the practical domain or codomain of λ , and are in no branches.

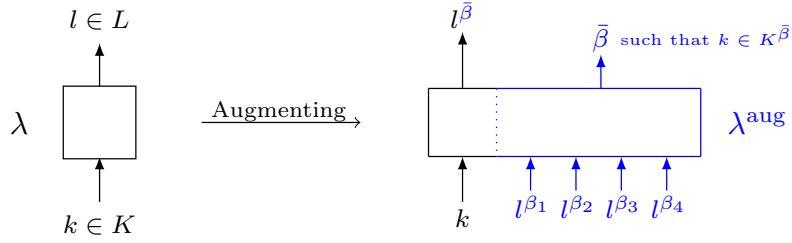


Figure 2.3: Augmenting the branched relation λ of Fig. 2.2. The augmented relation identifies in which branch $\bar{\beta}$ the input k is, and outputs the choice of output $l^{\bar{\beta}}$ – the “bifurcation choice” – within that branch provided among the auxiliary inputs, together with the name $\bar{\beta}$ of the branch that thus “happens”. If the input k is in no branch, the relation has no output.

$L \times \text{Bran}(\lambda)$ given by

$$\lambda^{\text{aug}}(k, (l^{\beta})_{\beta \in \text{Bran}(\lambda)}) := \begin{cases} (l^{\bar{\beta}}, \bar{\beta}) & \text{if } k \in K^{\bar{\beta}} \\ \emptyset & \text{if } k \notin K_{\text{in}(\lambda), \text{prac}} \end{cases} \quad (2.5)$$

That is, the augmented relation takes as input, in addition to the original input $k \in K$ of the initial branched relation, the choice of one output value l^{β} – that we call a “bifurcation choice” – for each branch β of the relation. It identifies whether the input k belongs to the input set $K^{\bar{\beta}}$ of some branch $\bar{\beta}$. If it does, it then outputs the corresponding bifurcation choice for that branch, together with the name of that branch; we then say that this branch “happens”. If it does not, the partial function is undefined – or, equivalently, the input is related to the empty set. See Fig. 2.3 for an illustration.

Note that some auxiliary inputs and/or outputs may be trivial, i.e. they may take only one possible value. In such a case, one may remove them from the definition of the augmented relation.⁶

2.2 Routed graph

2.2.1 Indexed and (branched) routed graphs

The route structure of a routed circuit will be obtained by composing (branched) relations together to obtain a so-called *routed graph*.

⁶For instance the branches β_2 and β_4 in Fig. 2.2 each have a single output value, so that the inputs l^{β_2} and l^{β_4} of the augmented relation can only take that single value. The auxiliary output on the other hand may be trivial if the relation has only one branch. One may choose, as e.g. in Fig. 13 of [39], not to represent these trivial inputs and outputs explicitly.

Definition 2.5 (Indexed graph⁷). An *indexed graph* Γ is a directed graph in which each arrow A is associated with a non-empty set of index values Ind_A .

The arrows in the graph can either connect two nodes (or possibly a node to itself) or be “open-ended” – i.e., an arrow that either points to a node but “comes from nowhere” (from no other node), or an arrow that leaves from a node but “goes nowhere” (to no other node).

We denote by Nodes_Γ the set of nodes of the graph Γ , and by Arr_Γ its set of arrows. For each node $\mathbf{N} \in \text{Nodes}_\Gamma$, we denote by $\text{in}(\mathbf{N}) \subseteq \text{Arr}_\Gamma$ the subset of arrows that point to \mathbf{N} , and by $\text{out}(\mathbf{N}) \subseteq \text{Arr}_\Gamma$ the subset of arrows that leave from \mathbf{N} . We then define $\text{Ind}_\mathbf{N}^{\text{in}} := \times_{A \in \text{in}(\mathbf{N})} \text{Ind}_A$ and $\text{Ind}_\mathbf{N}^{\text{out}} := \times_{A \in \text{out}(\mathbf{N})} \text{Ind}_A$.

Definition 2.6 ((Branched) Routed graph⁸). A *routed graph* $\mathcal{G} = (\Gamma, (\lambda_\mathbf{N})_{\mathbf{N} \in \text{Nodes}_\Gamma})$ consists of an indexed graph Γ and, for every node \mathbf{N} , a relation $\lambda_\mathbf{N} : \text{Ind}_\mathbf{N}^{\text{in}} \rightarrow \text{Ind}_\mathbf{N}^{\text{out}}$ called the *route* for node \mathbf{N} .

A routed graph $\mathcal{G} = (\Gamma, (\lambda_\mathbf{N})_{\mathbf{N} \in \text{Nodes}_\Gamma})$ is said to be *branched* if all relations $\lambda_\mathbf{N}$, for all $\mathbf{N} \in \text{Nodes}_\Gamma$, are branched.

We will often, for brevity, just write a (branched) routed graph as \mathcal{G} or $(\Gamma, (\lambda_\mathbf{N})_\mathbf{N})$. Graphically we write an index k_A (with the understanding that $k_A \in \text{Ind}_A$) next to each arrow A – although we may drop it when Ind_A is just a singleton, or simply to lighten the graphical representation – and we write the route $\lambda_\mathbf{N}$ next to each node \mathbf{N} ; see Fig. 2.4a.

In the context of a branched routed graph $(\Gamma, (\lambda_\mathbf{N})_\mathbf{N})$, the branches of a relation $\lambda_\mathbf{N}$ will generically be denoted \mathbf{N}^β (rather than just β as in the general preliminary section on relations); the corresponding sets of input and output values will be denoted $\text{Ind}_{\mathbf{N}^\beta}^{\text{in}} \subseteq \text{Ind}_\mathbf{N}^{\text{in}}$ and $\text{Ind}_{\mathbf{N}^\beta}^{\text{out}} \subseteq \text{Ind}_\mathbf{N}^{\text{out}}$ (instead of $K^\beta \subseteq K$ and $L^\beta \subseteq L$), respectively, so that $\mathbf{N}^\beta = (\text{Ind}_{\mathbf{N}^\beta}^{\text{in}} \rightarrow \text{Ind}_{\mathbf{N}^\beta}^{\text{out}})$. By abuse of language, the inputs, outputs and branches of the relations $\lambda_\mathbf{N}$ will also be referred to as the inputs, outputs and branches of the node \mathbf{N} directly.

Let us also mention the possibility to use *global index constraints* as a short-hand to specify the routes $(\lambda_\mathbf{N})_\mathbf{N}$ for a routed graph, which we will extensively use throughout this work. Indeed, just as the relation at a node can conveniently be specified by some constraint(s) on its inputs and outputs, the global routes (i.e., the multi-node relation $\times_{\mathbf{N} \in \text{Nodes}_\Gamma} \lambda_\mathbf{N}$) can be specified indirectly by giving some constraints relating all the indices appearing in the routed graph. Thereby, they specify the globally consistent assignments of index values across the graph.

2.2.2 Choice relation and (bi-)univocality

Instead of attaching the original relations $\lambda_\mathbf{N}$ to each node of the branched routed graph $(\Gamma, (\lambda_\mathbf{N})_\mathbf{N})$, one may “plug in” their augmented versions, representing the auxiliary inputs and outputs introduced in the augmenting procedure as additional open-ended arrows (similarly to Fig. 2.3), with for each node \mathbf{N} the input sets $\text{Ind}_{\mathbf{N}^\beta}^{\text{out}}$ (for all \mathbf{N}^β) and the output set $\text{Bran}(\lambda_\mathbf{N})$ attached to them. By composing the augmented relations through the arrows of the graph (see Section 2.1.1), this allows us to define a relation between all auxiliary inputs – together with the indices attached

⁷Again we slightly depart here from the definition given in [39], by not including the “dimension” of each index value in the definition of an indexed graph. We will only need to specify later on, when constructing the branch graph, which index values are considered to be “one-dimensional”; see Definition 2.11.

Furthermore, contrary to [39] we do not require *a priori* that for an open-ended arrow A , Ind_A must necessarily be a singleton; we will only impose this when this becomes necessary, namely when testing the property of “(bi-)univocality”, see Definition 2.8.

Also, for simplicity we define an indexed graph as a graph rather than a multigraph, which, as noted in [39], does not make any difference in the framework presented here. In fact, having several arrows from a node \mathbf{N} to a node \mathbf{M} , each with a set of index values Ind_{A_j} , is equivalent to having a single arrow with the cartesian product $\times_j \text{Ind}_{A_j}$ attached to it. Indeed, we allow index values to be of any type, including lists of values (or sets, as we will use in our construction for QC-QCs in Section 4).

⁸Once again we slightly depart and take a more general definition than that in [39], which required routed graphs to only involve branched relations. Here we will specify and talk about a “branched routed graph” when this is the case.

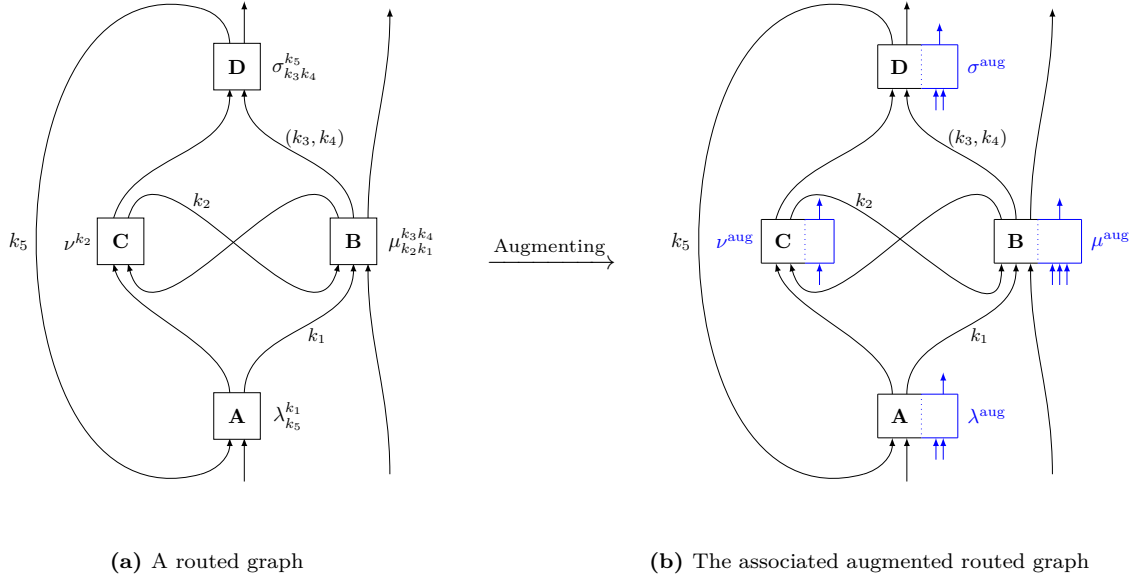


Figure 2.4: (a) Example of a routed graph (reproducing that of Fig. 11(b) in [39]). The arrows not bearing indices have a single index value attached to them (i.e. the corresponding Ind_A is a singleton). (b) Augmenting all of its relations (as in Fig. 2.3, with, for each node \mathbf{N} the set K being $\text{Ind}_{\mathbf{N}}^{\text{in}} = \times_{A \in \text{in}(\mathbf{N})} \text{Ind}_A$ and the set L being $\text{Ind}_{\mathbf{N}}^{\text{out}} = \times_{A \in \text{out}(\mathbf{N})} \text{Ind}_A$) and composing the relations through the arrows of the graph (as explained in Section 2.1.1) yields the *choice relation* – a relation between (the index values attached to) all arrows with open-ended tails and all arrows with open-ended heads.

to arrows that “come from nowhere” in the original indexed graph – and all auxiliary outputs – together with the indices attached to arrows that “go nowhere”; see Fig. 2.4b.

Formally, denoting by $\text{Arr}_{\Gamma}^{\text{int}}$ the set of all “internal” arrows of Γ (which connect two nodes in the graph), by $\text{Arr}_{\Gamma}^{\text{in}}$ and $\text{Arr}_{\Gamma}^{\text{out}}$ the sets of all (original) arrows of Γ that “come from nowhere” and that “go nowhere”, respectively (so that $\text{Arr}_{\Gamma} = \text{Arr}_{\Gamma}^{\text{int}} \sqcup \text{Arr}_{\Gamma}^{\text{in}} \sqcup \text{Arr}_{\Gamma}^{\text{out}}$), we introduce the following definition.

Definition 2.7 (Choice relation). Given a branched routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$, its *choice relation* is the relation $\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})} : \left(\times_{A \in \text{Arr}_{\Gamma}^{\text{in}}} \text{Ind}_A \right) \times \left(\times_{\substack{\mathbf{N} \in \text{Nodes}_{\Gamma}, \\ \mathbf{N}^{\beta} \in \text{Bran}(\lambda_{\mathbf{N}})}} \text{Ind}_{\mathbf{N}^{\beta}}^{\text{out}} \right) \rightarrow \left(\times_{A \in \text{Arr}_{\Gamma}^{\text{out}}} \text{Ind}_A \right) \times \left(\times_{\mathbf{N} \in \text{Nodes}_{\Gamma}} \text{Bran}(\lambda_{\mathbf{N}}) \right)$ defined as

$$\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})} := \text{Tr}_{(\text{Ind}_A)_{A \in \text{Arr}_{\Gamma}^{\text{int}}}} \left[\otimes_{\mathbf{N} \in \text{Nodes}_{\Gamma}} \lambda_{\mathbf{N}}^{\text{aug}} \right] = *_{\mathbf{N} \in \text{Nodes}_{\Gamma}} \lambda_{\mathbf{N}}^{\text{aug}} \quad (2.6)$$

(where $*$ denotes the multi-term “link product” that allows one to compose relations, generalising Eq. (2.4)).

Thus, together with some input in $\times_{A \in \text{Arr}_{\Gamma}^{\text{in}}} \text{Ind}_A$ and some output in $\times_{A \in \text{Arr}_{\Gamma}^{\text{out}}} \text{Ind}_A$, the choice relation relates some choice of output values $(l^{\mathbf{N}^{\beta}})_{\mathbf{N}, \mathbf{N}^{\beta}} \in \times_{\mathbf{N}, \mathbf{N}^{\beta}} \text{Ind}_{\mathbf{N}^{\beta}}^{\text{out}}$ – some “bifurcation choice” – for each of all branches of all routes in the graph, to the specific branches, for all nodes, that do “happen” for those bifurcation choices.

With all this in place, we are now ready to define the property of “(bi-)univocality” that we will require the branched routed graph to satisfy. This presupposes that the indexed graph has trivial input and output values attached to its open-ended arrows (i.e., that for all $A \in \text{Arr}_{\Gamma}^{\text{in}} \sqcup \text{Arr}_{\Gamma}^{\text{out}}$, Ind_A

is a singleton),⁹ and formalises the requirement that for each possible combination of bifurcation choices, one and only one branch should happen at each node (and similarly in the adjoint graph¹⁰).

Definition 2.8 ((Bi-)Univocality). A branched routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ with trivial Ind_A for all open-ended arrows A is said to be *univocal* if its choice relation $\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})}$ is a function – which is then accordingly called the *choice function*.

Such a $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ is said to be *bi-univocal* if both it and its adjoint $\mathcal{G}^\top = (\Gamma^\top, (\lambda_{\mathbf{N}}^\top)_{\mathbf{N}})$ are univocal.

Given that the definition of (bi-)univocality requires *a priori* that the routed graph is branched and that Ind_A is trivial for each open-ended arrow A , we will no longer explicitly specify these properties when referring to a (bi-)univocal routed graph. Furthermore, since $\times_{A \in \text{Arr}_\Gamma^{\text{in}}} \text{Ind}_A$ and $\times_{A \in \text{Arr}_\Gamma^{\text{out}}} \text{Ind}_A$ are taken to be trivial, we will make them implicit in the definition of the choice function, which will then be seen as a function

$$\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})} : \prod_{\substack{\mathbf{N} \in \text{Nodes}_\Gamma, \\ \mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})}} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}} \rightarrow \prod_{\mathbf{N} \in \text{Nodes}_\Gamma} \text{Bran}(\lambda_{\mathbf{N}}). \quad (2.7)$$

Remark 2.9. If $\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})}$ is not surjective, the choice function acts as a *global constraint* restricting which combination of branches in $\times_{\mathbf{N} \in \text{Nodes}_\Gamma} \text{Bran}(\lambda_{\mathbf{N}})$ may jointly happen. If, further, the projection of the choice function onto an individual node \mathbf{M} , i.e. $\times_{\mathbf{N} \in \text{Nodes}_\Gamma, \mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}} \rightarrow \text{Bran}(\lambda_{\mathbf{M}})$ is not surjective, some branches \mathbf{M}^γ will never happen and may be disregarded entirely.¹¹ This can be seen as a restriction of the practical input and output sets. In practice, we will restrict the combinations of incoming and outgoing indices $\text{Ind}_{\mathbf{M}}^{\text{in/out}}$ considered for each node \mathbf{M} *a priori*, so as not to be left with unnecessary irrelevant branches \mathbf{M}^γ .

2.3 Branch graph

Once we have checked that the routed graph under consideration is bi-univocal, we can build its *branch graph*, on which we will check some condition (the “only weak loops” property) on the potential loops it may contain. The branch graph connects the branches of the different nodes of the original routed graph, providing a more fine-grained picture of their connectivity, and differentiates between so-called *strong* and *weak parents*.¹²

2.3.1 One-dimensional values and strong parents

Roughly speaking, the notion of *strong parent* indicates whether there can be some direct transmission of information (in a way that will be specified later) from one branch at one node to another branch at some other node in the graph. For this, one first needs to characterise how – through which index values – the two branches under consideration are linked:

Definition 2.10 (Branch-linking values). Let $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ be a branched routed graph, and let $\mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})$, $\mathbf{M}^\gamma \in \text{Bran}(\lambda_{\mathbf{M}})$ be two branches of two nodes $\mathbf{N}, \mathbf{M} \in \text{Nodes}_\Gamma$. If there is an

⁹ In [39], this assumption is introduced to simplify the statement of bi-univocality, but the possibility of a direct generalisation is claimed by the authors.

¹⁰The adjoint of a routed graph is simply obtained by reversing all its arrows and taking the converse of all its routes. Beware that $\Lambda_{(\Gamma^\top, (\lambda_{\mathbf{N}}^\top)_{\mathbf{N}})} \neq \Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})}^\top$: the choice relation of the adjoint graph \mathcal{G}^\top does not even act on the same spaces as the adjoint of the choice relation on \mathcal{G} .

¹¹For the following section, one can easily deduce that any such branch will not partake in any strong or weak parent relation.

¹²For clarity: these are two independent notions. A branch may be a strong parent but not a weak parent of another branch, or vice versa.

arrow A from \mathbf{N} to \mathbf{M} in the graph Γ , we define the *set of values linking \mathbf{N}^β to \mathbf{M}^γ* as

$$\text{LinkVal}(\mathbf{N}^\beta, \mathbf{M}^\gamma) := \left\{ k_A \in \text{Ind}_A \left| \begin{array}{l} \exists (k_{A'})_{A' \in \text{Arr}_\Gamma \setminus \{A\}} \text{ such that } \forall \mathbf{N}' \in \text{Nodes}_\Gamma, (k_{A'})_{A' \in \text{in}(\mathbf{N}')} \stackrel{\lambda_{\mathbf{N}'}}{\sim} (k_{A'})_{A' \in \text{out}(\mathbf{N}')} \\ \text{and } (k_{A'})_{A' \in \text{out}(\mathbf{N})} \in \text{Ind}_{\mathbf{N}^\beta}^{\text{out}}, (k_{A'})_{A' \in \text{in}(\mathbf{M})} \in \text{Ind}_{\mathbf{M}^\gamma}^{\text{in}} \end{array} \right. \right\}. \quad (2.8)$$

That is, $\text{LinkVal}(\mathbf{N}^\beta, \mathbf{M}^\gamma)$ is the set of index values that can be attributed to the arrow (if any) from \mathbf{N} to \mathbf{M} , in order to satisfy the global index constraints, and belong to the branches $\mathbf{N}^\beta, \mathbf{M}^\gamma$ of the nodes \mathbf{N}, \mathbf{M} . By definition, these global index constraints are satisfied if there exists a consistent assignment of all other values on all other arrows of the graph, whose appropriate combinations (corresponding to the arrows that point to, or leave from a given node) satisfy all the relations specified at all nodes in the graph. In this case, these index combinations are related by the global routes.

In fact not all index values that thus link two branches will be useful to transmit information. Indeed we shall specify and single out certain index values that will play a specific role, in not being able to transmit information by themselves.

Definition 2.11 (One-dimensional index values). Certain index values are singled out and said to be “*one-dimensional*”.¹³

Given an arrow A of an indexed graph Γ , we denote by $1\text{Dim}_A \subseteq \text{Ind}_A$ the set of one-dimensional index values attached to it; $(1\text{Dim}_A)_{A \in \text{Arr}_\Gamma}$ (which we shall abbreviate as $(1\text{Dim}_A)_A$) thus denotes the set of all one-dimensional index values of Γ .

Once the set of one-dimensional index values is specified, the *strong parents* are then characterised as follows:

Definition 2.12 (Strong parent). Given a branched routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$, its set of one-dimensional index values $(1\text{Dim}_A)_A$, and two nodes $\mathbf{N}, \mathbf{M} \in \text{Nodes}_\Gamma$ as well as two branches $\mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})$, $\mathbf{M}^\gamma \in \text{Bran}(\lambda_{\mathbf{M}})$, we say that \mathbf{N}^β is a *strong parent* of \mathbf{M}^γ with respect to $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}}, (1\text{Dim}_A)_A)$ if there is an arrow A from \mathbf{N} to \mathbf{M} in the graph Γ , and $\text{LinkVal}(\mathbf{N}^\beta, \mathbf{M}^\gamma)$ contains

- (at least) one element that is *not* one-dimensional – i.e., not in 1Dim_A ;
- *or* strictly more than one element.

2.3.2 Weak parents

Recall that, according to [Definition 2.4](#), the augmented version of a relation outputs the branch $\bar{\beta}$ (if any) that the input k belongs to – the branch that “happens” when the input is k . One may then focus on a given branch β of the relation, and ask specifically if that branch happens or not – i.e., ask for the Boolean value of $\delta_{\beta, \bar{\beta}}$ (see [Footnote 5](#)).

Similarly, considering now the choice function $\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})} : \times_{\mathbf{N}, \mathbf{N}^\beta} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}} \rightarrow \times_{\mathbf{N}} \text{Bran}(\lambda_{\mathbf{N}})$ of a univocal routed graph, one may focus on a given branch $\mathbf{M}^\gamma \in \text{Bran}(\lambda_{\mathbf{M}})$ of a given node \mathbf{M} , and ask specifically if that branch happens for some given inputs in $\times_{\mathbf{N}, \mathbf{N}^\beta} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}}$. To formalise this, we introduce the following:

Definition 2.13 (\mathbf{M}^γ -Happens function). Given a univocal routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ and a specific node $\mathbf{M} \in \text{Nodes}_\Gamma$ with a specific branch $\mathbf{M}^\gamma \in \text{Bran}(\lambda_{\mathbf{M}})$, the \mathbf{M}^γ -*Happens function* is

¹³With the original definitions of [\[39\]](#), all index values are attributed a non-zero natural number called its “dimension”. For now we actually only need to identify those whose dimension is 1.

the Boolean function¹⁴

$$\begin{aligned} \text{Happ}_{\mathbf{M}^\gamma} : \prod_{\mathbf{N} \in \text{Nodes}_\Gamma, \mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}} &\rightarrow \{0, 1\}, \\ (l^{\mathbf{N}^\beta})_{\mathbf{N} \in \text{Nodes}_\Gamma, \mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}})} &\mapsto \delta_{\mathbf{M}^\gamma, \overline{\mathbf{M}^\gamma}}, \end{aligned} \quad (2.9)$$

where $\overline{\mathbf{M}^\gamma} \in \text{Bran}(\lambda_{\mathbf{M}})$ is the output of the choice function $\Lambda_{(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})} : \prod_{\mathbf{N}, \mathbf{N}^\beta} \text{Ind}_{\mathbf{N}^\beta}^{\text{out}} \rightarrow \prod_{\mathbf{N}} \text{Bran}(\lambda_{\mathbf{N}})$ of the graph corresponding to the node \mathbf{M} , for the same input $(l^{\mathbf{N}^\beta})_{\mathbf{N}, \mathbf{N}^\beta}$.

The notion of a *weak parent* then characterises whether a bifurcation choice made within a given branch influences whether another branch happens or not.

Definition 2.14 (Weak parent). Given a univocal routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ and two nodes $\mathbf{N}, \mathbf{M} \in \text{Nodes}_\Gamma$ with two branches $\mathbf{N}^\beta \in \text{Bran}(\lambda_{\mathbf{N}}), \mathbf{M}^\gamma \in \text{Bran}(\lambda_{\mathbf{M}})$, we say that \mathbf{N}^β is a *weak parent* of \mathbf{M}^γ with respect to $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ if the Boolean output value of the \mathbf{M}^γ -Happens function $\text{Happ}_{\mathbf{M}^\gamma}$, as defined in Eq. (2.9), depends on the input value of $l^{\mathbf{N}^\beta}$ – i.e., the bifurcation choice – attached to the branch \mathbf{N}^β .

Note that if \mathbf{N}^β has a single output value (i.e. there is no bifurcation choice to be made within that branch: $|\text{Ind}_{\mathbf{N}^\beta}^{\text{out}}| = 1$), then it cannot be the weak parent of any other branch. On the other hand, if the node \mathbf{M} has a single possible branch \mathbf{M}^γ , i.e. $\text{Bran}(\lambda_{\mathbf{M}}) = \{\mathbf{M}^\gamma\}$, then \mathbf{M}^γ happens for sure ($\text{Happ}_{\mathbf{M}^\gamma}$ returns the constant output 1, independently of any input values at any node of the routed graph), so that \mathbf{M}^γ has no weak parent.

2.3.3 Branch graph and weak loops

Definition 2.15 (Branch graph). Given a bi-univocal routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ and the specification of its set of one-dimensional index values $(\mathbf{1Dim}_A)_A$, the *branch graph* $\mathcal{G}^{\text{Bran}}$ is the directed (multi)graph in which

- the nodes are the branches of $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$;
- there is a solid arrow from a branch \mathbf{N}^β to a branch \mathbf{M}^γ if \mathbf{N}^β is a strong parent of \mathbf{M}^γ with respect to $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}}, (\mathbf{1Dim}_A)_A)$;
- there is a green dashed arrow from \mathbf{N}^β to \mathbf{M}^γ if \mathbf{N}^β is a weak parent of \mathbf{M}^γ with respect to $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$;
- there is a red dashed arrow from \mathbf{N}^β to \mathbf{M}^γ if \mathbf{M}^γ is a weak parent of \mathbf{N}^β with respect to the adjoint graph $(\Gamma^\top, (\lambda_{\mathbf{N}}^\top)_{\mathbf{N}})$.

Considering all types of arrows, a branch graph may contain cycles, consisting of arrows of potentially different colours. The framework distinguishes a certain type of loops, namely:

Definition 2.16 (Weak loop). A loop in a branch graph $\mathcal{G}^{\text{Bran}}$ is said to be *weak* if it only contains green dashed arrows, or if it only contains red dashed arrows.

As we will see below, such loops are not “harmful”: routed graphs that only contain weak loops can be used to build valid quantum processes. This motivates the following:

Definition 2.17 (Valid graph). We take a routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ with one-dimensional index values given by $(\mathbf{1Dim}_A)_A$ to be *valid* if it satisfies the principle of *bi-univocality* (cf. Definition 2.8) and every loop in its associated branch graph is *weak* (cf. Definition 2.16).

In particular, a bi-univocal routed graph will be valid if its associated branch graph features no loops at all.

¹⁴ $\text{Happ}_{\mathbf{M}^\gamma}$ here is not to be confused with the notation Happens_β from [39], which was denoting the Boolean output set, $\text{Happens}_\beta \cong \{0, 1\}$ (see Footnote 5).

2.4 Routed linear maps

With the routed graph, we have established a graph-theoretic structure to encode – potentially cyclic, yet consistent – connectivity between nodes. To bridge the gap to quantum processes, we now shift our attention to *routed linear maps*, which we will use to describe quantum operations to associate with these nodes. Introduced in [43], routed linear maps also use relations to restrict linear maps as well as the Hilbert spaces they act upon, splitting them into multiple independent sectors. In this work, we will only consider Hilbert spaces \mathcal{H} of finite dimension $\dim(\mathcal{H})$, and refrain from mentioning this fact explicitly going forward.

Consider two Hilbert spaces \mathcal{H}^K and \mathcal{H}^L , which are sectorised into orthogonal subspaces $\mathcal{H}^K = \bigoplus_{k \in K} \mathcal{H}_k^K$ and $\mathcal{H}^L = \bigoplus_{l \in L} \mathcal{H}_l^L$, with K and L denoting finite index sets. Then, for linear maps $A : \mathcal{H}^K \rightarrow \mathcal{H}^L$ (which we will also write as $A \in \mathcal{L}(\mathcal{H}^K, \mathcal{H}^L)$), we may introduce *sectorial constraints* restricting the form of A . More precisely, we specify a *route* λ , given as a relation $\lambda : K \rightarrow L$ (identified with a Boolean matrix $(\lambda_k^l)_{k \in K}^{l \in L}$). We then say that A *follows the route* λ if it satisfies

$$\neg(k \overset{\lambda}{\sim} l) \implies \Pi_l^L A \Pi_k^K = 0, \quad \text{or equivalently,} \quad A = \sum_{k,l} \lambda_k^l \cdot \Pi_l^L A \Pi_k^K, \quad (2.10)$$

where Π_k^K is the projector from \mathcal{H}^K to \mathcal{H}_k^K and Π_l^L is the projector from \mathcal{H}^L to \mathcal{H}_l^L . We will then write $A : \mathcal{H}^K \xrightarrow{\lambda} \mathcal{H}^L$ and denote the tuple (λ, A) a *routed (linear) map*. The condition above captures the idea that the image of a state in \mathcal{H}_k^K has support only on those sectors \mathcal{H}_l^L such that $k \overset{\lambda}{\sim} l$, i.e. on $\bigoplus_{l \in \lambda(k)} \mathcal{H}_l^L$.

These routed maps can be composed analogously to the linear maps themselves: their composition will follow the composition of the respective routes. Consider $\lambda : K \rightarrow L_1 \times L_2$, $\mu : L_2 \times L_3 \rightarrow M$ (as in Section 2.1.1), and $A : \mathcal{H}^K \xrightarrow{\lambda} \mathcal{H}^{L_1 L_2}$, $B : \mathcal{H}^{L_2 L_3} \xrightarrow{\mu} \mathcal{H}^M$, with $\mathcal{H}^{L_1 L_2} := \mathcal{H}^{L_1} \otimes \mathcal{H}^{L_2}$ and $\mathcal{H}^{L_2 L_3} := \mathcal{H}^{L_2} \otimes \mathcal{H}^{L_3}$ being (naturally) sectorised according to $L_1 \times L_2$ and $L_2 \times L_3$, respectively. These then compose as¹⁵

$$(\lambda, A) * (\mu, B) = (\lambda * \mu, BA) \in (K \times L_3 \rightarrow L_1 \times M) \times \mathcal{L}(\mathcal{H}^K \otimes \mathcal{H}^{L_3}, \mathcal{H}^{L_1} \otimes \mathcal{H}^M). \quad (2.11)$$

Here, we employ the composition of relations as considered in Section 2.1.1, and for matrix multiplication, assume that the involved matrices are extended with identities on the respectively missing spaces. In the same way that the composition of unitary quantum transformations can be represented using circuit diagrams [47], Theorem 1 of [43] then proves that entire routed circuits may be constructed using elementary routed maps as building blocks.

One important consequence of this construction is that λ may restrict the *practical* input and output spaces of A in contrast to its *formal* input and output spaces \mathcal{H}^K and \mathcal{H}^L . This reflects the notion of practical (co-)domain for the relation λ , as introduced in Definition 2.1: \mathcal{H}_k^K is in the practical input space of (λ, A) if and only if $k \in K_{\text{in}(\lambda), \text{prac}}$, and analogously for the practical output space. We then define the practical in- and output spaces for A as

$$\begin{aligned} \mathcal{H}_{\text{in}(\lambda), \text{prac}}^K &:= \bigoplus_{k \in K_{\text{in}(\lambda), \text{prac}}} \mathcal{H}_k^K = \bigoplus_{k \in K} \left(\sum_l \lambda_k^l \right) \mathcal{H}_k^K \subseteq \mathcal{H}^K \\ \mathcal{H}_{\text{out}(\lambda), \text{prac}}^L &:= \bigoplus_{l \in L_{\text{out}(\lambda), \text{prac}}} \mathcal{H}_l^L = \bigoplus_{l \in L} \left(\sum_k \lambda_k^l \right) \mathcal{H}_l^L \subseteq \mathcal{H}^L. \end{aligned} \quad (2.12)$$

This captures the idea that A must annihilate sectors \mathcal{H}_k^K outside of its practical input space (with $\sum_l \lambda_k^l = 0$): the route λ does not relate these sectors to anything. Similarly, unrelated sectors

¹⁵If A and B follow the routes λ and μ , resp., then $\lambda_k^{l_1, l_2} = 0$ implies $(\Pi_{l_1}^{L_1} \otimes \Pi_{l_2}^{L_2}) A \Pi_k^K = 0$ and $\mu_{l_2, l_3}^m = 0$ implies $\Pi_m^M B (\Pi_{l_2}^{L_2} \otimes \Pi_{l_3}^{L_3}) = 0$. Then using Eq. (2.3), $(\lambda * \mu)_{k, l_3}^{l_1, m} = 0 \iff \sum_{l_2} \lambda_k^{l_1, l_2} \mu_{l_2, l_3}^m = 0 \iff \forall l_2, \lambda_k^{l_1, l_2} = 0$ or $\mu_{l_2, l_3}^m = 0$, from which it follows that $(\Pi_{l_1}^{L_1} \otimes \Pi_m^M) (\mathbb{1}^{L_1} \otimes B) (A \otimes \mathbb{1}^{L_3}) (\Pi_k^K \otimes \Pi_{l_3}^{L_3}) = \sum_{l_2} (\Pi_{l_1}^{L_1} \otimes \Pi_m^M B) (\mathbb{1}^{L_1} \otimes \Pi_{l_2}^{L_2} \otimes \mathbb{1}^{L_3}) (A \Pi_k^K \otimes \Pi_{l_3}^{L_3}) = \sum_{l_2} \left(\underbrace{\mathbb{1}^{L_1} \otimes \Pi_m^M B (\Pi_{l_2}^{L_2} \otimes \Pi_{l_3}^{L_3})}_0 \right) \left(\underbrace{(\Pi_{l_1}^{L_1} \otimes \Pi_{l_2}^{L_2}) A \Pi_k^K \otimes \mathbb{1}^{L_3}}_0 \right) = 0$, meaning that $(\mathbb{1}^{L_1} \otimes B) (A \otimes \mathbb{1}^{L_3})$ follows the route $\lambda * \mu$.

\mathcal{H}_l^L at the output must be outside the image of A (as $\sum_k \lambda_k^l = 0$). Note that under sequential composition of routed maps, the practical input and output spaces may be further reduced to smaller subspaces of the respective spaces of the individual routed maps. This is analogous to the behaviour of the practical input and output sets of relations under composition, as outlined in [Section 2.1.1](#).

For a *branched relation* λ in particular, we specify *branch spaces* $\mathcal{H}_{\text{in}}^{K^\beta}$ and $\mathcal{H}_{\text{out}}^{L^\beta}$ for a given branch $\beta = (K^\beta \rightarrow L^\beta) \in \text{Bran}(\lambda)$. We define

$$\mathcal{H}_{\text{in}}^{K^\beta} := \bigoplus_{k \in K^\beta} \mathcal{H}_k^K \quad \text{and} \quad \mathcal{H}_{\text{out}}^{L^\beta} := \bigoplus_{l \in L^\beta} \mathcal{H}_l^L. \quad (2.13)$$

Any routed map $M : \mathcal{H}^K \xrightarrow{\lambda} \mathcal{H}^L$ will then (uniquely) decompose into a set of routed maps $M = \bigoplus_{\beta \in \text{Bran}(\lambda)} M^\beta$, with $M^\beta : \mathcal{H}_{\text{in}}^{K^\beta} \rightarrow \mathcal{H}_{\text{out}}^{L^\beta}$ for each branch β .

We will consider a routed map a *routed unitary or isometry* if it acts as a unitary or isometry, respectively, when restricted to its practical input and output space. (Due to [Eq. \(2.10\)](#), they will be 0 outside these spaces.¹⁶)

2.5 Skeletal supermap and fleshing out for a routed graph

Having introduced routed linear maps, we now shift our attention to *higher-order (linear) operations*, which are commonly called *supermaps* [14]. These (routed) supermaps take a tuple of (routed) linear maps as an argument and yield another (routed) linear map. Accordingly, in circuit diagrams, they are usually depicted as boxes with “slots”, allowing one to “plug in” some linear operations inside. Anticipating our intention to associate these slots with a node in a graph later in this section, for a given slot associated with a node \mathbf{N} , we will generally denote its input and output space as $\mathcal{H}^{\text{in}(\mathbf{N})}$ and $\mathcal{H}^{\text{out}(\mathbf{N})}$ respectively, rather than the potentially more familiar syntax $\mathcal{H}^{A^{I/O}}$ used e.g. in [34]. For the unrouted case, supermaps are then defined as follows.

Definition 2.18 (Supermap). A *supermap* of the type

$$\bigtimes_{\mathbf{N}} \left(\mathcal{H}^{\text{in}(\mathbf{N})} \rightarrow \mathcal{H}^{\text{out}(\mathbf{N})} \right) \longrightarrow (\mathcal{H}^P \rightarrow \mathcal{H}^F), \quad (2.14)$$

is a linear map $\bigotimes_{\mathbf{N}} \mathcal{L}(\mathcal{H}^{\text{in}(\mathbf{N})}, \mathcal{H}^{\text{out}(\mathbf{N})}) \longrightarrow \mathcal{L}(\mathcal{H}^P, \mathcal{H}^F)$.¹⁷

For a given slot in the supermap, associated with a node \mathbf{N} , we then say it is of type $\mathcal{H}^{\text{in}(\mathbf{N})} \rightarrow \mathcal{H}^{\text{out}(\mathbf{N})}$. Generally, the linear operations we compose with the supermap may feature additional input and output systems, which will denote as $\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}$ and $\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$. In circuit diagrams, these systems will be “sticking out” as “dangling wires”, after an operation extended in this manner has been plugged into a slot.

Being interested in quantum theory, supermaps which preserve unitarity and isometricity are of particular interest.

Definition 2.19 (Superunitary, superisometry). A *superunitary* of the type

$$\bigtimes_{\mathbf{N}} \left(\mathcal{H}^{\text{in}(\mathbf{N})} \rightarrow \mathcal{H}^{\text{out}(\mathbf{N})} \right) \longrightarrow (\mathcal{H}^P \rightarrow \mathcal{H}^F) \quad (2.15)$$

¹⁶For a routed unitary $U : \mathcal{H}^K \xrightarrow{\lambda} \mathcal{H}^L$, this implies that $\dim(\mathcal{H}_{\text{prac,in}}^{K,\lambda}) = \dim(\mathcal{H}_{\text{prac,out}}^{L,\lambda})$, while for a routed isometry V on the same spaces, $\dim(\mathcal{H}_{\text{prac,in}}^{K,\lambda}) \leq \dim(\mathcal{H}_{\text{prac,out}}^{L,\lambda})$ is sufficient. If the relation is branched, then additionally for a given branch $\beta = (K^\beta \rightarrow L^\beta)$, $\dim(\mathcal{H}_{\text{in}}^{K^\beta}) = \dim(\mathcal{H}_{\text{out}}^{L^\beta})$ holds for unitaries and $\dim(\mathcal{H}_{\text{in}}^{K^\beta}) \leq \dim(\mathcal{H}_{\text{out}}^{L^\beta})$ for isometries.

¹⁷In the literature, supermaps are usually expressed as higher-order operations of the type $\bigtimes_{\mathbf{N}} (\mathcal{L}(\mathcal{H}^{\text{in}(\mathbf{N})}) \rightarrow \mathcal{L}(\mathcal{H}^{\text{out}(\mathbf{N})})) \longrightarrow (\mathcal{L}(\mathcal{H}^P) \rightarrow \mathcal{L}(\mathcal{H}^F))$, mapping channels to channels at the level of the “mixed” description of quantum states and operations. As in this work we focus on the actions of supermaps onto “pure” operations, we use the simpler pure representations instead – as was also considered in [39].

is a supermap of the same type, such that for any choice of ancillary input and output spaces $\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}$ and $\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$ and any choice of unitary maps $U_{\mathbf{N}} : \mathcal{H}^{\text{in}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})} \rightarrow \mathcal{H}^{\text{out}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$ for every \mathbf{N} , the linear map

$$M : \mathcal{H}^P \otimes \left(\bigotimes_{\mathbf{N}} \mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})} \right) \longrightarrow \mathcal{H}^F \otimes \left(\bigotimes_{\mathbf{N}} \mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})} \right) \quad (2.16)$$

resulting from plugging the $U_{\mathbf{N}}$ into the slots of the supermap is unitary.¹⁸ Analogously, if for any such choice of ancillary spaces and any corresponding choice of isometries $V_{\mathbf{N}}$ for every \mathbf{N} , the resulting linear map M is an isometry, the supermap is a *superisometry*.¹⁹

Clearly, if we plug other superunitaries rather than unitaries into a superunitary, the resulting supermap will again be a superunitary, with an analogous statement holding for superisometries.

We continue by formally generalising the notions of supermaps, superunitaries and superisometries to their respective *routed* analogues.

Definition 2.20 (Routed supermap). Let $\lambda_{\mathbf{N}}$ and μ be relations with $\lambda_{\mathbf{N}} : \text{Ind}_{\mathbf{N}}^{\text{in}} \rightarrow \text{Ind}_{\mathbf{N}}^{\text{out}}$, and all Hilbert spaces \mathcal{H} be sectorised accordingly. Then, a routed supermap of type

$$\bigtimes_{\mathbf{N}} \left(\mathcal{H}^{\text{in}(\mathbf{N})} \xrightarrow{\lambda_{\mathbf{N}}} \mathcal{H}^{\text{out}(\mathbf{N})} \right) \longrightarrow (\mathcal{H}^P \xrightarrow{\mu} \mathcal{H}^F), \quad (2.17)$$

is a supermap on the respective spaces, which is restricted to input maps in $\bigotimes_{\mathbf{N}} \mathcal{L}(\mathcal{H}^{\text{in}(\mathbf{N})}, \mathcal{H}^{\text{out}(\mathbf{N})})$, where each tensor factor follows the associated route $\lambda_{\mathbf{N}}$, and whose output map follows the route μ . To denote the practical input and output spaces for a node \mathbf{N} , we write $\mathcal{H}_{\text{prac}}^{\text{in/out}(\mathbf{N})}$, while for individual branches \mathbf{N}^{β} (if applicable), we write $\mathcal{H}_{\beta}^{\text{in/out}(\mathbf{N})}$.

We call a routed supermap a *routed superunitary* / *superisometry* whenever it acts as a superunitary / superisometry when restricted to any routed unitaries $(U_{\mathbf{N}}, \lambda_{\mathbf{N}})$ / routed isometries $(V_{\mathbf{N}}, \lambda_{\mathbf{N}})$ following the route $\lambda_{\mathbf{N}}$ for each node \mathbf{N} , potentially involving unsectorised ancillary systems $\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}$ and $\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$.²⁰

For a given slot in the routed supermap, associated with a node \mathbf{N} , we then say it is of type $\mathcal{H}^{\text{in}(\mathbf{N})} \xrightarrow{\lambda_{\mathbf{N}}} \mathcal{H}^{\text{out}(\mathbf{N})}$.

We continue by associating the routed graph we have constructed in Section 2.2 with a particular routed supermap, which we will call the *skeletal supermap*. Here, the arrows will be translated into wires in a circuit diagram as shown in Fig. 2.5, representing sectorised Hilbert spaces, while the nodes will be understood as slots for operations following their associated routes.²¹ Thereby,

¹⁸In order to be able to plug in unitaries $U_{\mathbf{N}}$ at each node \mathbf{N} , the dimensions of input and output systems need to match: $\dim(\mathcal{H}^{\text{in}(\mathbf{N})}) \cdot \dim(\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}) = \dim(\mathcal{H}^{\text{out}(\mathbf{N})}) \cdot \dim(\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})})$. In order for the resulting map to be a unitary, one further needs to have $\dim(\mathcal{H}^P) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}) = \dim(\mathcal{H}^F) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})})$. Altogether, this requires $\dim(\mathcal{H}^P) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}^{\text{out}(\mathbf{N})}) = \dim(\mathcal{H}^F) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}^{\text{in}(\mathbf{N})})$.

¹⁹Here, the ancillary systems need to be chosen such that $\dim(\mathcal{H}^{\text{in}(\mathbf{N})}) \cdot \dim(\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}) \leq \dim(\mathcal{H}^{\text{out}(\mathbf{N})}) \cdot \dim(\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})})$ and $\dim(\mathcal{H}^P) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}) \leq \dim(\mathcal{H}^F) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})})$.

²⁰While for unrouted superunitaries, we can adjust the dimensions of $\mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}$ and $\mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$ for the input and output dimension of each unitary $U_{\mathbf{N}}$ to match whenever $\dim(\mathcal{H}^P) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}^{\text{out}(\mathbf{N})}) = \dim(\mathcal{H}^F) \cdot \prod_{\mathbf{N}} \dim(\mathcal{H}^{\text{in}(\mathbf{N})})$ (see Footnote 18), this is generally not the case for a routed superunitary. Considering for instance a branched node \mathbf{N} with sectorised practical input/output spaces $\mathcal{H}_{\text{prac}}^{\text{in/out}(\mathbf{N})} = \bigoplus_{\mathbf{N}^{\beta} \in \text{Bran}(\lambda_{\mathbf{N}})} \mathcal{H}_{\mathbf{N}^{\beta}}^{\text{in/out}(\mathbf{N})}$, to which one attaches (unsectorised) ancillary systems, the extended input/output spaces are $\mathcal{H}_{\text{prac}}^{\text{in/out}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{in/out}(\mathbf{N})} = \bigoplus_{\mathbf{N}^{\beta}} (\mathcal{H}_{\mathbf{N}^{\beta}}^{\text{in/out}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{in/out}(\mathbf{N})})$. In order to be able to plug in unitaries that follow the branched relations, one then needs $\dim(\mathcal{H}_{\mathbf{N}^{\beta}}^{\text{in}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})}) = \dim(\mathcal{H}_{\mathbf{N}^{\beta}}^{\text{out}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})})$ for each \mathbf{N}^{β} , which requires the ratio $\dim(\mathcal{H}_{\mathbf{N}^{\beta}}^{\text{in}(\mathbf{N})}) / \dim(\mathcal{H}_{\mathbf{N}^{\beta}}^{\text{out}(\mathbf{N})})$ to be the same for all \mathbf{N}^{β} . Any routed supermap which does not satisfy this condition for all \mathbf{N} will therefore *trivially* be a routed superunitary, as it is unable to act on routed unitaries *at all* (not even using ancillas). For routed superisometries, no similar challenges arise.

²¹While supermaps are often conveniently described in the Choi picture (cf. Section 3.1) we will refrain from doing so for the skeletal supermap, as this would require introducing two copies of each Hilbert space, attached to each

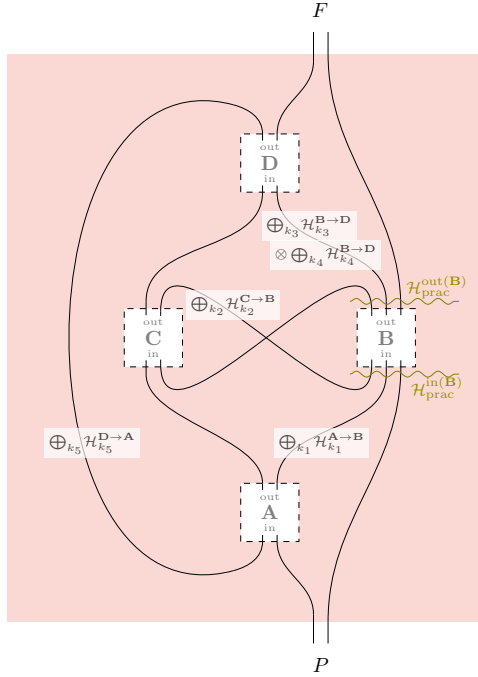


Figure 2.5: Depiction of the skeletal supermap (cf. Definition 2.21) associated to the routed graph shown in Fig. 2.4a. Each wire linking nodes \mathbf{N} to \mathbf{M} , corresponding to an arrow A from $\mathbf{N} \rightarrow \mathbf{M}$ in the routed graph, is associated with a sectorised Hilbert space $\mathcal{H}^A = \bigoplus_{k_A \in \text{Ind}_A} \mathcal{H}_{k_A}^A$. The dimension of these sectors is specified via a dimension assignment dim_Γ . Due to correlations between the indices mediated through the index constraints, the physically relevant input/output space of each node is only given by a practical subspace (cf. Section 2.4) of the tensor product of the incoming/outgoing wires. E.g., for the node \mathbf{B} , we have $\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{B})} \subset \mathcal{H}^{\text{in}(\mathbf{B})}$.

To reduce clutter, we do not make explicit Hilbert spaces with a trivial sectorisation on the wires in the figure, e.g. $\mathcal{H}^{\mathbf{B} \rightarrow \mathbf{C}}$, and omit an explicit representation of the routes in this figure.

a single routed graph can lead to a whole class of processes by specifying different (dimensional) Hilbert spaces and choosing different routed maps or supermaps for the fleshing out, which we will formally introduce in Definition 2.22 below.

Definition 2.21 (Skeletal supermap associated to a routed graph). Given a routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ and a dimension assignment $\text{dim}_\Gamma : \bigsqcup_{A \in \text{Arr}_A} \text{Ind}_A \rightarrow \mathbb{N}^*$, their associated *skeletal supermap* is the routed supermap obtained by interpreting

- each arrow A as a sectorised Hilbert space $\mathcal{H}^A = \bigoplus_{k_A} \mathcal{H}_{k_A}^A$, whose sectors are labelled by the different index values $k_A \in \text{Ind}_A$ attached to this arrow and satisfy $\dim(\mathcal{H}_{k_A}^A) = \text{dim}_\Gamma(k_A) \forall k_A \in \text{Ind}_A$; and
- each node \mathbf{N} as an open slot for a routed linear map M , going from the tensor product of the Hilbert spaces associated to its incoming arrows in the set $\text{in}(\mathbf{N})$, to that of the Hilbert spaces associated to its outgoing arrows in the set $\text{out}(\mathbf{N})$, and following the route $\lambda_{\mathbf{N}} : \text{Ind}_{\mathbf{N}}^{\text{in}} \rightarrow \text{Ind}_{\mathbf{N}}^{\text{out}}$ associated to that node

$$M : \bigotimes_{A \in \text{in}(\mathbf{N})} \bigoplus_{k_A \in \text{Ind}_A} \mathcal{H}_{k_A}^A \xrightarrow{\lambda_{\mathbf{N}}} \bigotimes_{A \in \text{out}(\mathbf{N})} \bigoplus_{k_A \in \text{Ind}_A} \mathcal{H}_{k_A}^A. \quad (2.18)$$

The skeletal supermap acts on the respective routed linear maps M as a routed supermap, composing them as specified by the routed graph Γ , interpreting the arrows as identity channels. The

end of their supporting arrows (so as to describe the identity channels by the Choi vectors of the form $|\mathbb{1}\rangle^{XX'}$). We will use the Choi picture later on, when explicitly “fleshing out” the skeletal supermap of our generic routed graph for QC-QCs in Section 4.3.

open-ended arrows that “come from nowhere” and that “go nowhere” will (altogether) be identified with the “global past” and “global future” spaces \mathcal{H}^P and \mathcal{H}^F that appear the definition of a routed supermap, respectively: i.e., $\mathcal{H}^P := \bigotimes_{A \in \text{Arr}_\Gamma^{\text{in}}} \mathcal{H}^A$ and $\mathcal{H}^F := \bigotimes_{A \in \text{Arr}_\Gamma^{\text{out}}} \mathcal{H}^A$.

Furthermore, when specifying the set of one-dimensional index values $(1\text{Dim}_A)_A$ of the routed graph, one imposes that any Hilbert space sector associated to an index value in $(1\text{Dim}_A)_A$ must be 1-dimensional – i.e. $\forall A, \forall k_A \in 1\text{Dim}_A, \dim(\mathcal{H}_{k_A}^A) = \text{dim}_\Gamma(k_A) = 1$.²² If $(1\text{Dim}_A)_A$ is not specified explicitly, then $(1\text{Dim}_A)_A$ is implicitly taken to contain precisely the indices k for which $\text{dim}_\Gamma(k) = 1$.

If a skeletal supermap is associated with a bi-univocal routed graph (Definition 2.8), the open arrows A of the routed graph must be trivially indexed. Nonetheless, we may also consider a generalised notion of bi-univocality (see Footnote 9), which does feature such indices, and correspondingly, yields a non-trivially routed map $\mathcal{H}^P \xrightarrow{\mu} \mathcal{H}^F$.

Note that when stating the practical input and output space $\mathcal{H}_{\text{prac}}^{\text{in/out}(\mathbf{N})}$ for each node \mathbf{N} , we can exclude index combinations which will never be realised due to global constraints, as outlined in Remark 2.9. With respect to the associated global route, these are not part of the practical input and output spaces of the skeletal supermap slots. Therefore, some tuples $k_{\text{in/out}} \in \text{Ind}_{\mathbf{N}}^{\text{in/out}}$ are not practically realised due to $(\lambda_{\mathbf{N}})_{\mathbf{N}}$ restricting index combinations, and we can effectively omit the respective spaces.

With this in place, we can specify compatible routed linear (super)maps M to fill the respective slots of the skeletal supermap, thereby *fleshing out* the skeletal supermap.

Definition 2.22 (Fleshing out). Given a skeletal supermap associated with a routed graph, its *fleshing out* is specified by a collection of routed maps or supermaps (with any number of slots) $\{M_{\mathbf{N}}\}_{\mathbf{N} \in \text{Nodes}_\Gamma}$, matching the type $\mathcal{H}^{\text{in}(\mathbf{N})} \xrightarrow{\lambda_{\mathbf{N}}} \mathcal{H}^{\text{out}(\mathbf{N})}$ for each \mathbf{N} , while allowing for ancillary systems.²³

If a slot is fleshed out with a routed supermap, it may feature an arbitrary number of (sectorised or unsectorised) slots.

2.6 Main theorem of Vanrietvelde *et al.* [39]

To conclude this review of routed quantum circuits, we state the main theorems of Ref. [39], which certify that when the underlying routed graph is valid (according to Definition 2.17), then the construction outlined above actually yields valid quantum processes.

Theorem 2.23 (Main theorem of [39]). Let $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ be a valid routed graph, with a dimension assignment dim_Γ . Then their associated skeletal supermap is a routed superunitary.

Corollary 2.24. Let $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ be a valid routed graph with a dimension assignment dim_Γ . Then, any supermap built from their associated skeletal supermap by fleshing it out with routed unitaries at some of its nodes and routed superunitaries²⁴ at the other nodes is a routed superunitary. In particular, if the routed superunitaries used in the fleshing out only have unsectorised slots, then the resulting supermap is a conventional (non-routed) superunitary.

Actually, this corollary can be extended to routed isometries and superisometries, even if these were not explicitly studied in [43]. Note, however, that we restrict ourselves to fleshing out with *monopartite* superisometries for this statement.

Corollary 2.25. Let $(\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$ be a valid routed graph with a dimension assignment dim_Γ . Then, any supermap built from their associated skeletal supermap by fleshing it out with routed

²²While we will usually not consider this case, further sectors could, in general, be of dimension 1 as well.

²³That is, the routed maps are of type $\mathcal{H}^{\text{in}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{in}(\mathbf{N})} \xrightarrow{\lambda_{\mathbf{N}}} \mathcal{H}^{\text{out}(\mathbf{N})} \otimes \mathcal{H}_{\text{anc}}^{\text{out}(\mathbf{N})}$ for each \mathbf{N} , while the routed supermaps have this type for their resulting output map, but may have (any number of) arbitrarily typed slots.

²⁴The restriction to monopartite superunitaries made in [39] is clearly not necessary for the corollary to hold.

isometries at some of its nodes and monopartite routed superisometries at other nodes is a routed superisometry. In particular, if the routed superisometries used in the fleshing out only have unsectorised slots, then the resulting supermap is a conventional (non-routed) superisometry.

Proof. Any isometry $V : \mathcal{H}^A \rightarrow \mathcal{H}^B$ can be extended to a unitary by introducing additional ancillary input and output systems $\mathcal{H}_{\text{anc}}^{\text{in}}$ and $\mathcal{H}_{\text{anc}}^{\text{out}}$, such that $\dim(\mathcal{H}_{\text{anc}}^{\text{in}}) = \dim(\mathcal{H}^B)$ and $\dim(\mathcal{H}_{\text{anc}}^{\text{out}}) = \dim(\mathcal{H}^A)$. We obtain a unitary $U : \mathcal{H}^A \otimes \mathcal{H}_{\text{anc}}^{\text{in}} \rightarrow \mathcal{H}^B \otimes \mathcal{H}_{\text{anc}}^{\text{out}}$, from which we can recover V in two steps. By plugging in a suitable ancillary input state into $\mathcal{H}_{\text{anc}}^{\text{in}}$, we obtain a map $V' : \mathcal{H}^A \rightarrow \mathcal{H}^B \otimes \mathcal{H}_{\text{anc}}^{\text{out}}$ that recovers the action of the isometry onto \mathcal{H}^B . As this restricts the image of V' , when projected on $\mathcal{H}_{\text{anc}}^{\text{out}}$, to be one-dimensional, we then recover V from V' by simply tracing out $\mathcal{H}_{\text{anc}}^{\text{out}}$.

Analogously, we can extend any monopartite superisometry, which may map unitaries to isometries, to a superunitary, by decomposing it into isometries first.²⁵ In a routed graph, these ancillary input / output systems correspond to additional unindexed open arrows (i.e. wires “coming from nowhere”) going into / coming out of the individual nodes \mathbf{N} corresponding to the skeletal supermap slots fleshed out by the respective maps $U_{\mathbf{N}}$. As introducing these arrows impacts neither the presence of loops nor the index assignments, the (valid) routed graph still gives rise to a superunitary.

We may then again compose the additional input wires with suitable input states, while tracing out the additional output wires, to obtain a routed superisometry (via the action of the respective isometries and monopartite superisometries). \square

Inspired by this corollary, we specify when we consider a routed quantum circuit to be a satisfactory representation of the connectivity of a supermap.

Definition 2.26 (Routed circuit decomposition). A *routed circuit decomposition of a routed superisometry* is given by a routed graph $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N}})$, a dimension assignment \dim_{Γ} – which, together, define a skeletal supermap – and a fleshing out, given by a collection of routed isometries for some nodes \mathbf{V} and monopartite routed superisometries for the remaining nodes \mathbf{A} , that do not feature any ancillary systems $\mathcal{H}_{\text{anc}}^{\text{in/out}(\mathbf{A})}$ and $\mathcal{H}_{\text{anc}}^{\text{in/out}(\mathbf{V})}$, respectively.

Two routed circuit decompositions are considered *equivalent* if they give rise to the same routed superisometry.

If none of the remaining slots after the fleshing out features non-trivial routes, we simply call the decomposition a *routed circuit decomposition of a superisometry*.

By disallowing the use of ancillary systems within a routed circuit decomposition, we ensure that indeed all relevant (routed) information processing relevant to the process is encoded into the skeletal supermap. As an alternative, additional unindexed arrows may be added to some nodes in the routed graph in their stead, recovering the auxiliary systems from these arrows in a modified skeletal supermap.

Routed circuit decompositions have been given for several supermaps, including causally indefinite ones [39, 48]. Indeed, thus far all the examples of purifiable supermaps [22] that have been explicitly studied have been able to be given such decompositions, but it remains an open question whether this is possible for all such supermaps. Note further that in a routed circuit decomposition of a supermap, only some of the slots in the skeletal supermap will be associated with slots in the decomposed supermap, while others will be fleshed out with “internal” operations with no open slots. That is, the routed graph used in a routed circuit decomposition may, in general, have more nodes than there are slots in the resulting supermap.

Regarding the following section, Corollary 2.25 allows for an independent proof for the validity of QC-QCs based on the routed quantum circuit framework. However, this will not be our main point of attention in this work. Rather, we will demonstrate a systematic construction that makes

²⁵Each monopartite superisometry is a (monopartite) quantum comb [45], which can be decomposed into two isometries V_1 and V_2 , with a memory system α and an open slot in-between. Extending both isometries to unitaries, we obtain a superunitary with two additional input and output wires $\mathcal{H}_{1,\text{anc}}^{\text{in/out}}$ and $\mathcal{H}_{2,\text{anc}}^{\text{in/out}}$, respectively. While this reasoning generalises to quantum combs and QC-QCs with more parties, it may not be applicable to general superisometries, for which no decomposition into isometries is established.

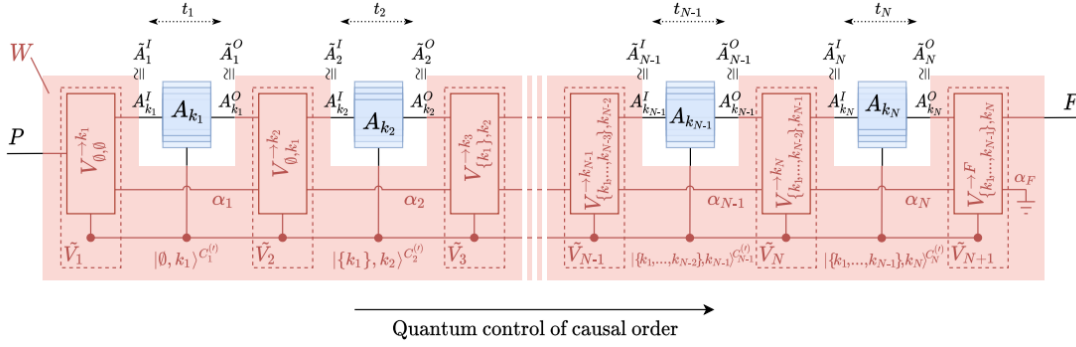


Figure 3.1: Generic depiction of a QC-QC. Figure reproduced from Ref. [34].

QC-QCs fit into the framework of [39] in terms of a suitable routed circuit decomposition, which will then allow us to see how their properties translate into diagrammatic features.

3 Quantum Circuits with Quantum Control of Causal Order

Quantum circuits with quantum control of causal order (QC-QCs) [34] are a class of supermaps with concrete physical descriptions that includes many relevant causally indefinite processes. In an N -slot QC-QC, as illustrated in Fig. 3.1, the action of N agents A_k , happening precisely once, alternate with “internal” transformations \tilde{V}_n , which control which agent A_k acts at which time t_1, \dots, t_N , dependent both on the agents who have acted before and on the operations they have performed. Reflecting this, in the remainder of this work, k (and ℓ) will consistently be used to index (time-delocalised) agents or their operations, while n indexes will refer to well-defined time slots. Various results indicate that under certain conditions, the class of QC-QCs is precisely the set of processes realisable within a fixed spacetime [11, 13, 41, 42]. It is worth noting also that QC-QCs do not allow for violations of causal inequalities [34].

For the formal description of QC-QCs, we write $\mathcal{N} = \{1, \dots, N\}$, consistently take $k, \ell \in \mathcal{N}, \mathcal{K}_n \subseteq \mathcal{N}$ and implicitly take \mathcal{K}_n such that $|\mathcal{K}_n| = n$, with $0 \leq n \leq N$. Furthermore, we identify a singleton $\{k\}$ with its unique element, and write for instance $\mathcal{K}_n \setminus k$ rather than $\mathcal{K}_n \setminus \{k\}$ or $\mathcal{K}_{n-1} \cup k$ rather than $\mathcal{K}_{n-1} \cup \{k\}$.

The QC-QC pattern is then realised using a quantum control system C_n with Hilbert space \mathcal{H}^{C_n} , spanned by states of the form $|\mathcal{K}_n \setminus k, k\rangle^{C_n}$. This system coherently controls, for each time slot t_n , which agent A_k 's operation is applied, as well as the “internal operations” $V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}$ applied in-between two agent's operations. Here, each internal $V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}$ sends its input from agent A_k with $k \in \mathcal{K}_n$ to agent A_ℓ with $\ell \notin \mathcal{K}_n$, for a given set of prior agents $\{A_j\}_{j \in \mathcal{K}_n \setminus k}$. Jointly, they constitute an isometry \tilde{V}_{n+1} , specifying both the transfer $k \rightarrow \ell$ between the agents and potential further transformations applied in-between. In addition to the input/output systems $A_k^{I/O}$, which are passed to the respective agents A_k to act upon, the isometries may also act on an ancillary system α_n , which allows one to pass arbitrary information between the internal isometries without making it available to the agents.

The control system C_n serves to keep track of the set of operations \mathcal{K}_n (including k , for A_k being applied at time t_n) already used, ensuring that each operation is used exactly once. Altogether, the QC-QC construction yields a superisometry, which is why we will call the resulting circuits “pure” QC-QCs. These have an open past P and an open future F system. However, by setting the dimension of P to 1^{26} , we may also close the past. Similarly, by partially tracing out the future F , we may obtain mixed supermaps with partially or fully closed future, which are no longer superisometries, but still purifiable superchannels.

²⁶This can be done by absorbing incoming quantum states into the isometry \tilde{V}_1 .

3.1 Choi picture and the link product

We continue by introducing the (pure) *Choi picture* for linear maps [49] (on finite-dimensional spaces), which we will rely on heavily to lay out the formal QC-QC construction. It allows one to conveniently express linear maps in the form of vectors. To do so, for a given Hilbert space \mathcal{H}^X , we fix an orthonormal basis $\{|i\rangle^X\}_i$ as the computational basis, and for any isomorphic space $\mathcal{H}^{X'}$, take its computational basis states $|i\rangle^{X'}$ to be in 1-to-1 correspondence. Then, we define

$$|\mathbb{1}\rangle\rangle^{XX'} = \sum_i |i\rangle^X \otimes |i\rangle^{X'} \in \mathcal{H}^X \otimes \mathcal{H}^{X'}. \quad (3.1)$$

Going forward, as a general rule, we will denote a $|\psi\rangle \in \mathcal{H}^X$ as $|\psi\rangle^X$, using just the label X to denote explicitly the state space. We may omit this label if it is obvious from context. With this, we may represent a (routed) linear map $A : \mathcal{H}^K \rightarrow \mathcal{H}^L$ by its *Choi vector*,

$$|A\rangle\rangle := (\mathbb{1}^K \otimes A) |\mathbb{1}\rangle\rangle^{KK'} = \sum_i |i\rangle^K \otimes A|i\rangle^{K'} \in \mathcal{H}^{KL}. \quad (3.2)$$

Notice that by definition of the Choi vector, $A = |A\rangle\rangle^{\top K}$ (where $\top K$ denotes the partial transpose on the space \mathcal{H}^K).

Given two Choi vectors $|V_1\rangle\rangle \in \mathcal{H}^{XY}$ and $|V_2\rangle\rangle \in \mathcal{H}^{YZ}$, we may introduce their (pure) *link product*, denoted by $*$, by using the partial trace over their shared systems $\{Y\}$:

$$|V_1\rangle\rangle * |V_2\rangle\rangle = \text{Tr}_Y \left[|V_1\rangle\rangle^\top \otimes |V_2\rangle\rangle \right] \in \mathcal{H}^{XZ}. \quad (3.3)$$

This is analogous to the composition of relations as introduced in Section 2.1.1, and provides the Choi vector for the composition of the two linear maps V_1, V_2 [34, 45, 46].

3.2 Formal construction of QC-QCs

We continue by outlining the formal construction of the notion outlined in the beginning of the section. For more specific details we direct the reader to Ref. [34].

Without loss of generality, we assume that for all agents A_k , their associated Hilbert spaces $\mathcal{H}^{A_k^I}$ and $\mathcal{H}^{A_k^O}$ are isomorphic to one another. Therefore, they are taken to be of the same dimension d_{A^I} and d_{A^O} .²⁷ If not given by default, this can be achieved by extending the dimensions of all operations appropriately.²⁸

To translate from the spaces of the individual agent operations to the space of all operations applied for a given time slot t_n , we introduce a further set of spaces $\mathcal{H}^{\tilde{A}_n^{I/O}}$, which are isomorphic to $\mathcal{H}^{A_k^{I/O}}$. Going forward, we will generally use a $\tilde{\cdot}$ to indicate that an operation is expressed with respect to $\mathcal{H}^{\tilde{A}_n^{I/O}}$ rather than $\mathcal{H}^{A_k^{I/O}}$. The internal isometries V then act upon these spaces, as well as ancillary spaces \mathcal{H}^{α_n} , passing information amongst themselves rather than making it accessible to the agents. Accordingly, for each internal control operator $V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} : \mathcal{H}^{A_k^O \alpha_n} \rightarrow \mathcal{H}^{A_\ell^I \alpha_{n+1}}$ applied between two agents A_k and A_ℓ in n^{th} and $(n+1)^{\text{th}}$ positions (with $k \in \mathcal{K}_n, \ell \notin \mathcal{K}_n$), we obtain its tilded description $\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} : \mathcal{H}^{\tilde{A}_n^O \alpha_n} \rightarrow \mathcal{H}^{\tilde{A}_n^I \alpha_{n+1}}$ as follows:

$$\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} = \mathbb{1}^{A_k^O \rightarrow \tilde{A}_n^O} V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} \mathbb{1}^{\tilde{A}_n^I \rightarrow A_\ell^I}, \quad (3.4)$$

where $\mathbb{1}^{X \rightarrow Y}$ is to be understood as the identity from a space \mathcal{H}^X to an isomorphic space \mathcal{H}^Y . Similarly, we may use these spaces to describe the joint input and output spaces of the superposed operations $A_k : \mathcal{H}^{A_k^I} \rightarrow \mathcal{H}^{A_k^O}$ being in n^{th} position respectively. Using the control system, we define a joint operation $\tilde{A}_n : \mathcal{H}^{\tilde{A}_n^I C_n} \rightarrow \mathcal{H}^{\tilde{A}_n^O C'_n}$.

²⁷These do not necessarily have to be equal.

²⁸Specifically, the respective operations can be extended by adding identities as tensor factors whose additional inputs and outputs are then traced out in the process.

This gives the following formal description of the operations in the tilded spaces in terms of Choi vectors:

$$|\tilde{A}_n\rangle\rangle = \sum_{\mathcal{K}_n, k \in \mathcal{K}_n} (|\mathbb{1}_{\mathcal{K}_n}^{A_k^I \tilde{A}_n^I}\rangle\rangle \otimes |\mathbb{1}_{\mathcal{K}_n}^{A_k^O \tilde{A}_n^O}\rangle\rangle) * |A_k\rangle\rangle^{A_k^I A_k^O} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n} \in \mathcal{H}^{\tilde{A}_n^I C_n \tilde{A}_n^O C'_n}, \quad (3.5)$$

$$\begin{aligned} |\tilde{V}_1\rangle\rangle &= \sum_{\ell \in \mathcal{N}} |\tilde{V}_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle\rangle^{P \tilde{A}_1^I \alpha_1} \otimes |\emptyset, \ell\rangle^{C_1} \in \mathcal{H}^{P \tilde{A}_1^I \alpha_1 C_1}, \\ |\tilde{V}_{n+1}\rangle\rangle &= \sum_{\mathcal{K}_n, k \in \mathcal{K}_n, \ell \notin \mathcal{K}_n} |\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle^{\tilde{A}_n^O \alpha_n \tilde{A}_{n+1}^I \alpha_{n+1}} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n} \otimes |\mathcal{K}_n, \ell\rangle^{C_{n+1}} \in \mathcal{H}^{\tilde{A}_n^O \alpha_n C'_n \tilde{A}_{n+1}^I \alpha_{n+1} C_{n+1}}, \\ |\tilde{V}_{N+1}\rangle\rangle &= \sum_{k \in \mathcal{N}} |\tilde{V}_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle\rangle^{\tilde{A}_N^O \alpha_N F} \otimes |\mathcal{N} \setminus k, k\rangle^{C'_N} \in \mathcal{H}^{\tilde{A}_N^O \alpha_N C'_N F}. \end{aligned} \quad (3.6)$$

We then compose all of these operations to a circuit $|V_{P \rightarrow F}\rangle\rangle$ which decomposes into the (pure) QC-QC $|w_{\text{QC-QC}}\rangle\rangle$ and the N agent operations $|A_k\rangle\rangle$. We compose these operations as follows, relabelling $\mathcal{K}_{n-1} \rightarrow \{k_1, \dots, k_{n-1}\}$, $k \rightarrow k_n$, $\ell \rightarrow k_{n+1}$ to be able to combine the respective sums:

$$\begin{aligned} |V_{P \rightarrow F}\rangle\rangle &= |\tilde{V}_1\rangle\rangle * |\tilde{A}_1\rangle\rangle * |\tilde{V}_2\rangle\rangle * \dots * |\tilde{V}_N\rangle\rangle * |\tilde{A}_N\rangle\rangle * |\tilde{V}_{N+1}\rangle\rangle \\ &= \sum_{(k_1, \dots, k_N)} |\tilde{V}_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle\rangle * |\tilde{A}_{k_1}\rangle\rangle * |\tilde{V}_{\emptyset, k_1}^{\rightarrow k_2}\rangle\rangle * \dots * |\tilde{V}_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N}\rangle\rangle * |\tilde{A}_{k_N}\rangle\rangle * |\tilde{V}_{\{k_1, \dots, k_{N-1}\}, k_N}^{\rightarrow F}\rangle\rangle \\ &= \sum_{(k_1, \dots, k_N)} (|A_1\rangle\rangle \otimes \dots \otimes |A_N\rangle\rangle) * |w_{(k_1, \dots, k_N, F)}\rangle\rangle \\ &= (|A_1\rangle\rangle \otimes \dots \otimes |A_N\rangle\rangle) * |w_{\text{QC-QC}}\rangle\rangle \in \mathcal{H}^{PF}, \end{aligned} \quad (3.7)$$

where

$$|w_{(k_1, \dots, k_N, F)}\rangle\rangle := |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle\rangle * |V_{\emptyset, k_1}^{\rightarrow k_2}\rangle\rangle * |V_{\{k_1\}, k_2}^{\rightarrow k_3}\rangle\rangle * \dots * |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N}\rangle\rangle * |V_{\{k_1, \dots, k_{N-1}\}, k_N}^{\rightarrow F}\rangle\rangle, \quad (3.8)$$

and obtaining

$$|w_{\text{QC-QC}}\rangle\rangle := \sum_{(k_1, \dots, k_N)} |w_{(k_1, \dots, k_N, F)}\rangle\rangle \quad (3.9)$$

as the process vector description for a pure QC-QC. Here, the sums are over all permutations (k_1, \dots, k_N) of $\{1, \dots, N\}$ (and, more generally, when we write (k_1, \dots, k_n) throughout the paper we implicitly assume that all k_i 's are distinct).

Finally, let us mention the generalisation from pure to mixed QC-QCs. In the QC-QC framework, mixed processes are captured by tracing out a final ancillary system α_F , which, in the description above, would be a subsystem of F . Therefore, only this single modification is required to obtain general QC-QCs from the pure case, leaving the construction of the process vector $|w_{\text{QC-QC}}\rangle\rangle$ unchanged otherwise. We obtain the process matrix representation of mixed circuits directly by splitting $\mathcal{H}^F = \mathcal{H}^{F'} \otimes \mathcal{H}^{\alpha_F}$:

$$W_{\text{QC-QC}} = \text{Tr}_{\alpha_F} |w_{\text{QC-QC}}\rangle\rangle \langle w_{\text{QC-QC}}|. \quad (3.10)$$

4 QC-QCs as Routed Quantum Circuits

In this section, we will present our main result. For each number of agents N , we will provide a single “generic” routed graph, whose associated skeletal supermap can be fleshed out to obtain any N -slot QC-QC, detailing their construction explicitly and proving the equivalence to the original QC-QC construction. All in all, this will amount to proving the following theorem:

Theorem 4.1. Any N -slot quantum circuit with quantum control of causal order can be implemented by fleshing out a skeletal supermap obtained from the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ introduced in [Definition 4.2](#).

In this construction, many structural features of QC-QCs will be recovered: both the internal isometries \tilde{V}_{n+1} and the agent operations A_k will be featured as nodes \mathbf{V}_{n+1} and \mathbf{A}_k in the routed graph, and the control system $C_n^{(\prime)}$ will emanate both the indices \mathcal{X}_n^k featured in the routed graph and the branch structure these indices induce. To make these correspondences as clear as possible, we choose our notation accordingly.

4.1 Generic routed graph for QC-QCs

We begin by describing in detail how the “generic routed graph” for a QC-QC will be defined. To ease understanding of the construction, we will provide an intuitive interpretations of various elements in italic sentences.

4.1.1 Definition of the routed graph

Definition 4.2 ($\mathcal{G}_{\text{QC-QC}(N)}$). The *generic routed graph* $\mathcal{G}_{\text{QC-QC}(N)}$ for an N -slot QC-QC is defined as follows:

- There are 2 types of nodes: “**A**-nodes” \mathbf{A}_k , for $k = 1, \dots, N$, and “**V**-nodes” \mathbf{V}_{n+1} , for $n = 0, \dots, N$.

*(As we will see, the external operations A_k , on which a QC-QC acts, will be plugged into the **A**-nodes, while the internal operations \tilde{V}_{n+1} of a QC-QC will be plugged into the **V**-nodes.)*

- The indexed graph involves the following indexed arrows:
 - An open-ended arrow $\xrightarrow{\emptyset} \mathbf{V}_1$ that “comes from nowhere” and points to \mathbf{V}_1 , with the single index value \emptyset attached to it (which we will henceforth drop).
 - $\forall n = 1, \dots, N, \forall k = 1, \dots, N$,

$$\mathbf{V}_n \xrightarrow{\mathcal{X}_n^k} \mathbf{A}_k \quad \text{and} \quad \mathbf{A}_k \xrightarrow{\mathcal{X}_n^{k'}} \mathbf{V}_{n+1}. \quad (4.1)$$

The indices $\mathcal{X}_n^{k(\prime)}$ take values among the n -element subsets of \mathcal{N} that contain k , or the “null value” \emptyset : either $\mathcal{X}_n^{k(\prime)} = \mathcal{K}_n$ for some $\mathcal{K}_n \ni k$ (still with the convention that $|\mathcal{K}_n| = n$), or $\mathcal{X}_n^{k(\prime)} = \emptyset$.

- An open-ended arrow $\mathbf{V}_{N+1} \xrightarrow{\mathcal{N} \cup F}$ that leaves from \mathbf{V}_{N+1} and “goes nowhere”, with the single index value $\mathcal{N} \cup F$ attached to it (which we will henceforth drop).

Thus, each node \mathbf{A}_k has N incoming arrows from the N nodes \mathbf{V}_1 to \mathbf{V}_N , and N outgoing arrows to the N nodes \mathbf{V}_2 to \mathbf{V}_{N+1} ; its input and output index values are of the form $(\mathcal{X}_n^k)_{n=1}^N \in \text{Ind}_{\mathbf{A}_k}^{\text{in}}$ and $(\mathcal{X}_n^{k'})_{n=1}^N \in \text{Ind}_{\mathbf{A}_k}^{\text{out}}$, resp. Each node \mathbf{V}_{n+1} on the other hand has N incoming and N outgoing arrows, from and to the N nodes \mathbf{A}_1 to \mathbf{A}_N ; its input and output index values are of the forms $(\mathcal{X}_n^k)_{k=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{in}}$ and $(\mathcal{X}_{n+1}^k)_{k=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{out}}$, respectively. The neighbourhood of each type of node is represented in Figs. 4.1 and 4.2.

(The intuitive interpretation for the values of the indices $\mathcal{X}_n^{k(\prime)}$ is that $\mathcal{X}_n^{k(\prime)} = \mathcal{K}_n$ indicates that operation A_k is applied in n^{th} position, with all other operations in $\mathcal{K}_n \setminus k$ being applied before; while $\mathcal{X}_n^{k(\prime)} = \emptyset$ indicates that A_k does not come in n^{th} position.)

- The routes are specified as follows (and as also stated on Fig. 4.1):
 - For each node \mathbf{A}_k , the lists of all input and output index values $(\mathcal{X}_n^k)_{n=1}^N \in \text{Ind}_{\mathbf{A}_k}^{\text{in}}$ and $(\mathcal{X}_n^{k'})_{n=1}^N \in \text{Ind}_{\mathbf{A}_k}^{\text{out}}$ must be equal – from now on we will then simply drop the primes – and must further satisfy

$$\exists! n, \mathcal{X}_n^k \neq \emptyset. \quad (4.2)$$

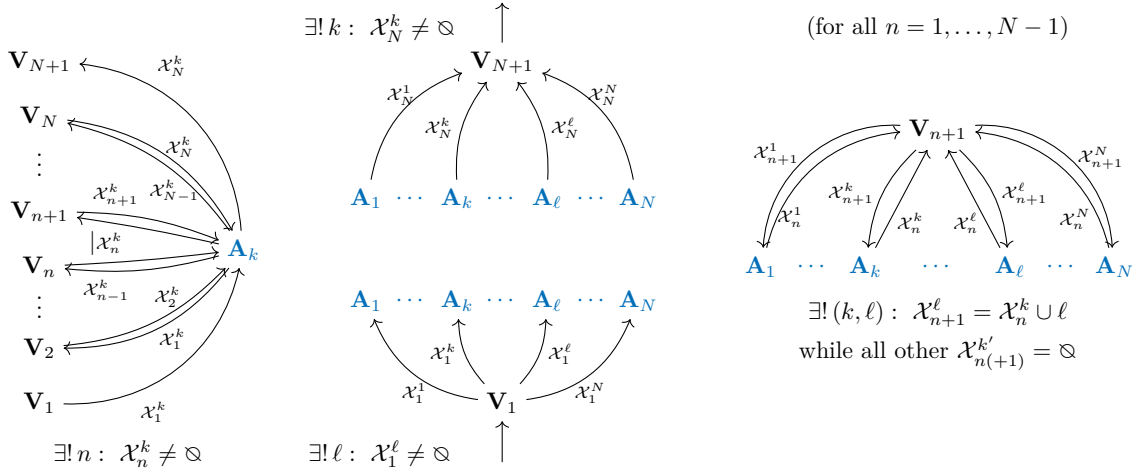


Figure 4.1: Neighbourhood of each type of node in the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ of a QC-QC: \mathbf{A}_k , \mathbf{V}_1 , \mathbf{V}_{n+1} (for $n = 1, \dots, N-1$), and \mathbf{V}_{N+1} .

Hence, the practical input and output sets of each node \mathbf{A}_k consist of (the same) lists of index values $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, for some set $\mathcal{K}_n \ni k$ at the n^{th} position,²⁹ attached to the arrow coming from node \mathbf{V}_n and to the arrow going to node \mathbf{V}_{n+1} .

(The fact that in each admissible combination of input and output index values to the \mathbf{A} -nodes there must be one and only one non-null-value index translates the fact that each operation A_k must be applied once and only once, at a single position n .)

- For each node \mathbf{V}_{n+1} , the lists of all input and output index values $(\mathcal{X}_n^k)_{k=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{in}}$ and $(\mathcal{X}_{n+1}^\ell)_{\ell=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{out}}$ must satisfy

$$\text{for } n = 0 \text{ and } n = N : \exists! \ell, \mathcal{X}_1^\ell \neq \emptyset \text{ and } \exists! k, \mathcal{X}_N^k \neq \emptyset, \text{ resp.} \quad (4.3)$$

$$\text{for } 1 \leq n \leq N-1 : \exists! k, \mathcal{X}_n^k \neq \emptyset, \exists! \ell, \mathcal{X}_{n+1}^\ell \neq \emptyset; \text{ and } \mathcal{X}_{n+1}^\ell = \mathcal{X}_n^k \cup \ell. \quad (4.4)$$

Hence, the practical input sets of each node \mathbf{V}_{n+1} , for $0 < n \leq N$, consist of lists of index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, for some set \mathcal{K}_n at the k^{th} position, such that $k \in \mathcal{K}_n$. Similarly, the practical output sets of each node \mathbf{V}_{n+1} , for $0 \leq n < N$, consist of lists of index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_{n+1}, \emptyset, \dots, \emptyset)$, for some set \mathcal{K}_{n+1} at the ℓ^{th} position, such that $\ell \in \mathcal{K}_{n+1}$. Such input and output index values are related (for $1 \leq n \leq N-1$) if and only if $\mathcal{K}_{n+1} = \mathcal{K}_n \cup \ell$.

(These constraints translate the fact that along each route, between any two operations \tilde{V}_n and \tilde{V}_{n+1} , one and only one operation A_k is applied.)

Combining the different constraints at the different nodes, we get the *global index constraint* (as expressed for the cases $N = 2, 3$ on Fig. 4.2) that³⁰

$$\begin{aligned} &\exists! (k_1, k_2, \dots, k_N), \\ &\mathcal{X}_1^{k_1} = \{k_1\}, \mathcal{X}_2^{k_2} = \{k_1, k_2\}, \dots, \mathcal{X}_N^{k_N} = \{k_1, k_2, \dots, k_N\} = \mathcal{N}, \text{ and all other } \mathcal{X}_n^k = \emptyset. \end{aligned} \quad (4.5)$$

(Here the understanding is that along each global path, all operations are applied, in a given order.)

²⁹Elements in the lists $(\mathcal{X}_n^k)_{n=1}^N \in \text{Ind}_{\mathbf{A}_k}^{\text{in/out}}$ are ordered from $n = 1$ to $n = N$. Further below, elements in the lists $(\mathcal{X}_n^k)_{k=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{in}}$ and $(\mathcal{X}_{n+1}^\ell)_{\ell=1}^N \in \text{Ind}_{\mathbf{V}_{n+1}}^{\text{out}}$ will be ordered from $k = 1$ to $k = N$.

³⁰It can indeed be seen that Eqs. (4.3) and (4.4) for all n imply Eq. (4.5); and vice versa, that Eq. (4.5) implies Eqs. (4.2) to (4.4).

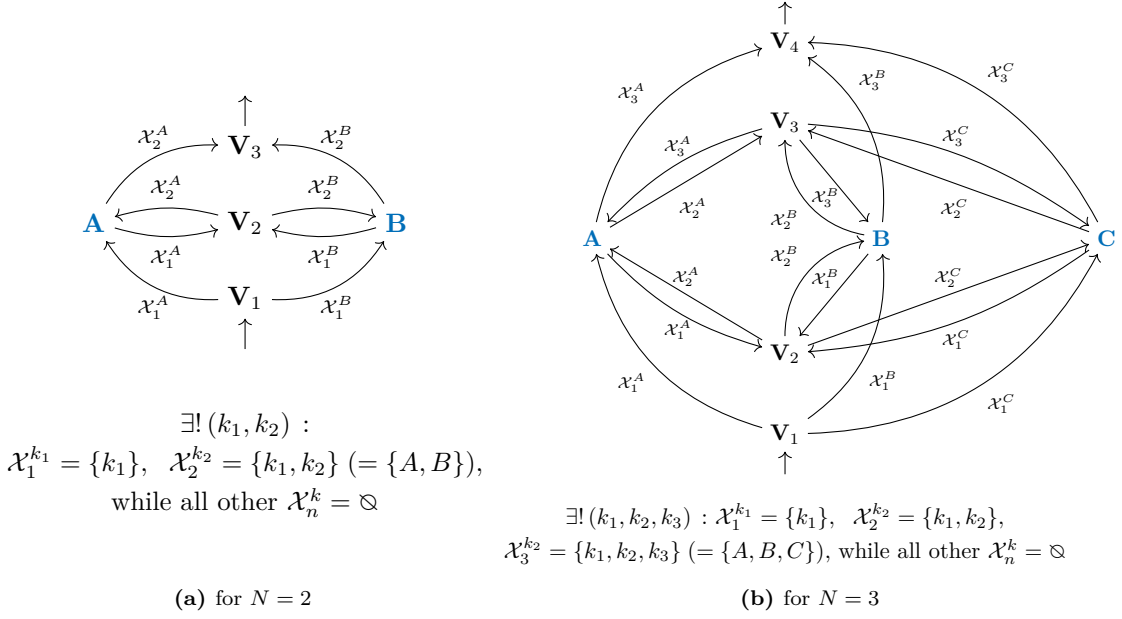


Figure 4.2: The generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ for $N = 2$ (left) and $N = 3$ (right). Here, $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ are denoted as $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

4.1.2 Identifying the branches

A-nodes. The lists of index values $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$ in the practical input and output sets of each node \mathbf{A}_k are in one-to-one relation. The relation $\lambda_{\mathbf{A}_k}$ is therefore (rather trivially) branched; to help relate its branches to the QC-QC construction later on, we label them by $\mathcal{K}_n \setminus k$ rather than by \mathcal{K}_n directly, i.e. we denote the branches as

$$\mathbf{A}_k^{\mathcal{K}_n \setminus k} := \left(\{(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)\} \rightarrow \{(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)\} \right). \quad (4.6)$$

Each node \mathbf{A}_k has 2^{N-1} such branches, one for each subset $\mathcal{K}_n \subseteq \mathcal{N}$ that includes k . (Each branch $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ relates to the set of parties $\mathcal{K}_n \setminus k$ that came before \mathbf{A}_k .)

V-nodes. According to the previous description, the route of each node \mathbf{V}_{n+1} (for $1 \leq n \leq N-1$) relates index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$ in the practical input set to index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_{n+1}, \emptyset, \dots, \emptyset)$ in the practical output set, with $\mathcal{K}_{n+1} = \mathcal{K}_n \cup \ell$ for some $\ell \notin \mathcal{K}_n$. One can verify that $\lambda_{\mathbf{V}_{n+1}}$ is also branched, with one branch corresponding to each subset \mathcal{K}_n , namely:³¹

$$\mathbf{V}_{n+1}^{\mathcal{K}_n} := \left(\{(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset) \mid \mathcal{K}_n \text{ at the } k^{\text{th}} \text{ position, } \forall k \in \mathcal{K}_n\} \right. \\ \left. \rightarrow \{(\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell, \emptyset, \dots, \emptyset) \mid \mathcal{K}_n \cup \ell \text{ at the } \ell^{\text{th}} \text{ position, } \forall \ell \in \mathcal{N} \setminus \mathcal{K}_n\} \right). \quad (4.7)$$

³¹To see that $\lambda_{\mathbf{V}_{n+1}}$ is branched, recall that its practical input set consists of lists of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, for some set \mathcal{K}_n at some position $k \in \mathcal{K}_n$ (as we denote by $\uparrow k$ below it). From Eq. (4.4) it can then be seen that $\lambda_{\mathbf{V}_{n+1}}((\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)) = \{(\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell, \emptyset, \dots, \emptyset) \mid \ell \in \mathcal{N} \setminus \mathcal{K}_n\}$, from which one can see that $\lambda_{\mathbf{V}_{n+1}}((\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset))$ and $\lambda_{\mathbf{V}_{n+1}}((\emptyset, \dots, \emptyset, \mathcal{K}'_n, \emptyset, \dots, \emptyset))$ are either equal if $\mathcal{K}_n = \mathcal{K}'_n$ (while $k, k' \in \mathcal{K}_n^{(\ell)}$ may differ), or disjoint if $\mathcal{K}_n \neq \mathcal{K}'_n$. Similarly, we find $\lambda_{\mathbf{V}_{n+1}}^\top((\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell, \emptyset, \dots, \emptyset)) = \{(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset) \mid k \in \mathcal{K}_n\}$. From this one can indeed identify the branches of $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ as in Eq. (4.7).

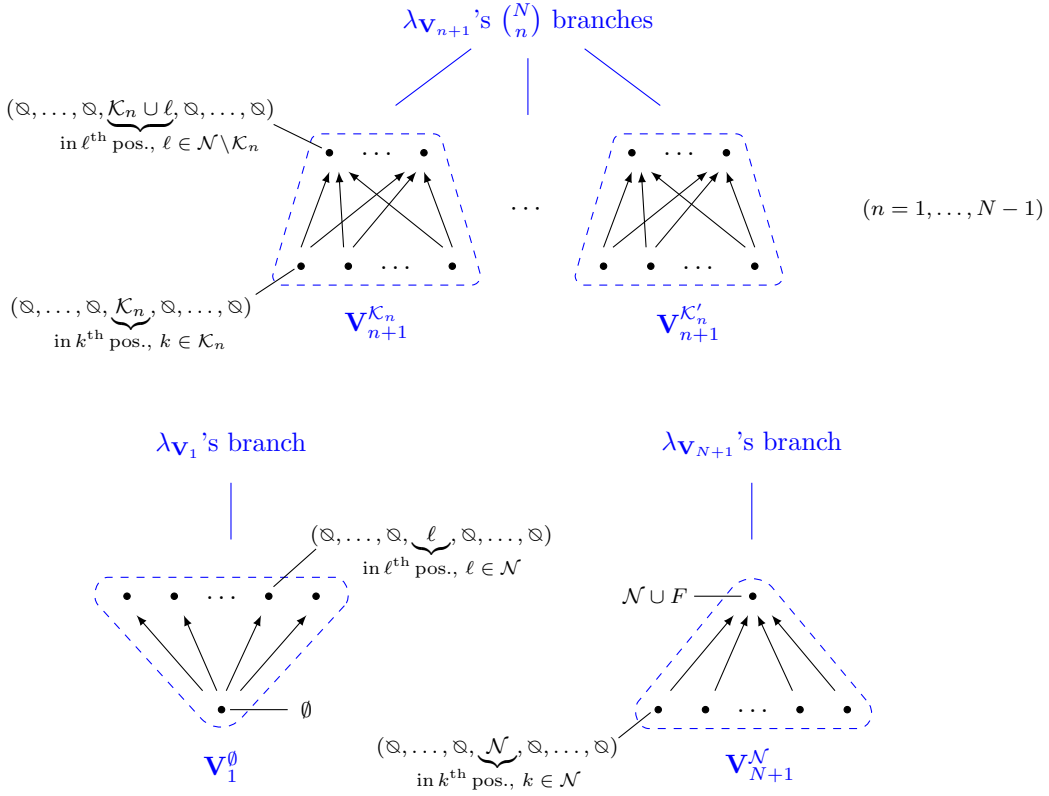


Figure 4.3: Branches of the \mathbf{V} -nodes. For each node \mathbf{V}_{n+1} , for $n = 0, \dots, N$, $\lambda_{\mathbf{V}_{n+1}} : \text{Ind}_{\mathbf{V}_{n+1}}^{\text{in}} \rightarrow \text{Ind}_{\mathbf{V}_{n+1}}^{\text{out}}$ features $\binom{N}{n}$ branches, as well as values outside its practical input and output sets (not shown). Each branch $V_{n+1}^{\mathcal{K}_n}$ is associated with n input values (for $n \geq 1$), each taking the form $(\mathcal{X}_n^k)_{k=1}^n$, where the position k of the single non-null index value indicates the agent A_k last visited. Analogously, it is associated with $N - n$ output values of the same form (for $n \leq N - 1$), with the position ℓ of the single non-null index value indicating the agent A_ℓ to be visited next.

For the nodes \mathbf{V}_1 and \mathbf{V}_{N+1} , we find that these each have a single branch

$$\mathbf{V}_1^\emptyset := \left(\{\emptyset\} \rightarrow \{(\emptyset, \dots, \emptyset, \ell, \emptyset, \dots, \emptyset) \mid \ell \text{ at the } \ell^{\text{th}} \text{ position}, \forall \ell \in \mathcal{N}\} \right)$$

and $\mathbf{V}_{N+1}^{\mathcal{N}} := \left(\{(\emptyset, \dots, \emptyset, \mathcal{N}, \emptyset, \dots, \emptyset) \mid \mathcal{N} \text{ at the } k^{\text{th}} \text{ position}, \forall k \in \mathcal{N}\} \rightarrow \{\mathcal{N} \cup F\} \right), \quad (4.8)$

respectively.

The branches of the \mathbf{V} -nodes are illustrated on Fig. 4.3. Each node \mathbf{V}_{n+1} has $\binom{N}{n}$ branches $\mathbf{V}_{n+1}^{\mathcal{K}_n}$, one for each n -element subset $\mathcal{K}_n \subseteq \mathcal{N}$.

(Each branch $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ relates to the set of parties \mathcal{K}_n that came before \tilde{V}_{n+1} .)

4.1.3 Augmented relations

Now that we have identified the branches of the relations $\lambda_{\mathbf{A}_k}$ and $\lambda_{\mathbf{V}_{n+1}}$, we continue by specifying their augmented versions $\lambda_{\mathbf{A}_k}^{\text{aug}}$ and $\lambda_{\mathbf{V}_{n+1}}^{\text{aug}}$ according to Definition 2.4.

A-nodes. There are no bifurcation choices to be made, hence no need to introduce auxiliary inputs (as these would anyway be trivial). The auxiliary output of some node \mathbf{A}_k just corresponds to the label of the branch that the input belongs to, namely $\mathcal{K}_n \setminus k$ as in Eq. (4.6). The augmented relation $\lambda_{\mathbf{A}_k}^{\text{aug}}$, as illustrated on Fig. 4.4, is then defined as the (partial) function

$$\lambda_{\mathbf{A}_k}^{\text{aug}} : (\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset) \mapsto \left((\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset), \mathcal{K}_n \setminus k \right). \quad (4.9)$$

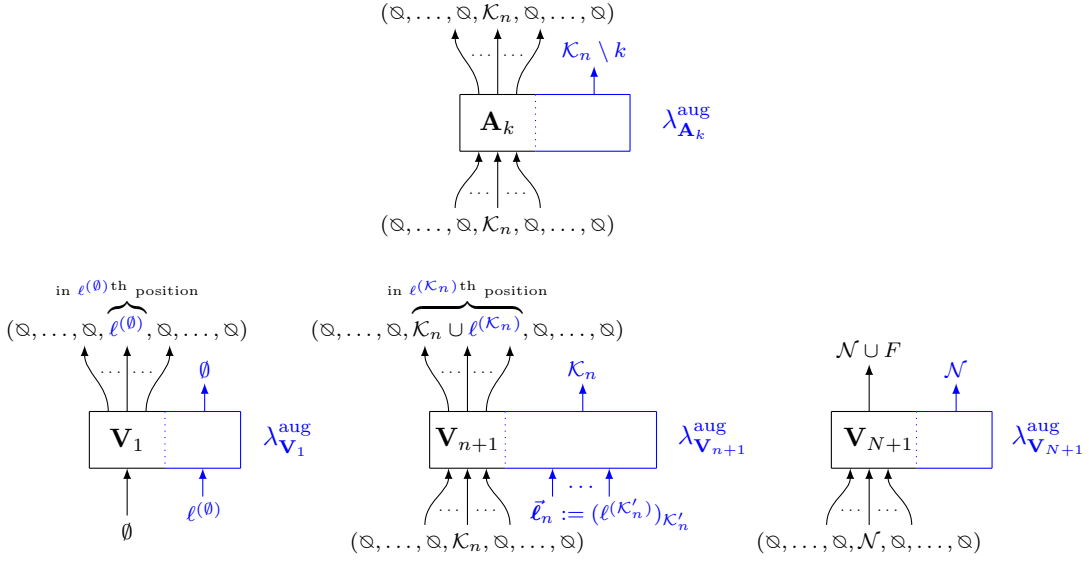


Figure 4.4: Augmented relations for the \mathbf{A} - and \mathbf{V} -nodes of the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$. We do not show any auxiliary input for the nodes \mathbf{A}_k and \mathbf{V}_{N+1} , as there are no bifurcation choices to be made. Notice also that the auxiliary output of \mathbf{V}_1 , as well as both outputs of \mathbf{V}_{N+1} , are constant and could therefore be ignored.

V-nodes. For each branch $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ of each node \mathbf{V}_{n+1} , for $n = 1, \dots, N-1$, there is a bifurcation choice to be made, corresponding to the choice of the value $\ell \in \mathcal{N} \setminus \mathcal{K}_n$ in the output set of that branch (cf. Eq. (4.7)).³² Hence to define the augmented relation, we add some auxiliary inputs $(\ell^{(\mathcal{K}'_n)})_{\mathcal{K}'_n} \in \times_{\mathcal{K}'_n} \mathcal{N} \setminus \mathcal{K}'_n$ that specify these bifurcation choices, for all possible branches. Upon receiving an input (to the original relation $\lambda_{\mathbf{V}_{n+1}}$) of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, the augmented relation $\lambda_{\mathbf{V}_{n+1}}^{\text{aug}}$ returns the element $(\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell^{(\mathcal{K}_n)}, \emptyset, \dots, \emptyset)$ of the output set that is determined by the value $\ell^{(\mathcal{K}_n)}$ in the list of auxiliary inputs, together with the value of \mathcal{K}_n indicating the branch that happens.

The augmented relation, as illustrated on Fig. 4.4, can thus be written as the (partial) function

$$\lambda_{\mathbf{V}_{n+1}}^{\text{aug}} : \left((\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset), (\ell^{(\mathcal{K}'_n)})_{\mathcal{K}'_n} \right) \mapsto \left((\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell^{(\mathcal{K}_n)}, \emptyset, \dots, \emptyset), \mathcal{K}_n \right) \\ \text{with } \mathcal{K}_n \cup \ell^{(\mathcal{K}_n)} \text{ at the } \ell^{(\mathcal{K}_n)\text{th}} \text{ position.} \quad (4.10)$$

The situation is similar for \mathbf{V}_1 , whose augmented relation can be written as

$$\lambda_{\mathbf{V}_1}^{\text{aug}} : (\emptyset, \ell^{(\emptyset)}) \mapsto ((\emptyset, \dots, \emptyset, \ell^{(\emptyset)}, \emptyset, \dots, \emptyset), \emptyset) \text{ with } \ell^{(\emptyset)} \text{ at the } \ell^{(\emptyset)\text{th}} \text{ position.} \quad (4.11)$$

For \mathbf{V}_{N+1} on the other hand, no bifurcation choice remains to be made, and the output of the (augmented) relation is constant:

$$\lambda_{\mathbf{V}_{N+1}}^{\text{aug}} : (\emptyset, \dots, \emptyset, \mathcal{N}, \emptyset, \dots, \emptyset) \mapsto (\mathcal{N} \cup F, \mathcal{N}) \quad (4.12)$$

(see again Fig. 4.4).

4.1.4 The choice function and bi-univocality

By Definition 2.7, the choice relation of $\mathcal{G}_{\text{QC-QC}(N)}$ is obtained by composing all augmented relations according to the routed graph, and is thus given, as in Eq. (2.6), by

$$\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}} = \lambda_{\mathbf{A}_1}^{\text{aug}} * \dots * \lambda_{\mathbf{A}_N}^{\text{aug}} * \lambda_{\mathbf{V}_1}^{\text{aug}} * \dots * \lambda_{\mathbf{V}_{N+1}}^{\text{aug}}. \quad (4.13)$$

³²For $n = N-1$, the bifurcation choice within each branch $\mathbf{V}_N^{\mathcal{K}_{N-1}}$ is trivial, as there remains only one $\ell \in \mathcal{N} \setminus \mathcal{K}_{N-1}$. In that case we may ignore the auxiliary inputs of the augmented relation.

The detailed calculations, starting from this expression, are presented in [Appendix A](#). Let us describe here the result on a more intuitive level.

The choice relation $\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}}$ takes as inputs all the various bifurcation choices for all possible branches $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ of all \mathbf{V} -nodes (recalling that the \mathbf{A} -nodes have no bifurcation choices), and outputs the possible branches that may happen at each of the \mathbf{A} - and \mathbf{V} -nodes. For convenience let us denote by $\vec{\ell}_n := (\ell^{(\mathcal{K}'_n)})_{\mathcal{K}'_n}$ the bifurcation choices for the different branches at the node \mathbf{V}_{n+1} (i.e. the auxiliary inputs of $\lambda_{\mathbf{V}_{n+1}}^{\text{aug}}$), and by $\vec{\ell} := (\vec{\ell}_n)_{n=0}^{N-1}$ the collection of all these bifurcation choices, at all \mathbf{V} -nodes. Let us then recursively define

$$\begin{aligned} \mathcal{K}_1^{\vec{\ell}} &:= \ell^{(\emptyset)}, \quad \mathcal{K}_{n+1}^{\vec{\ell}} := \mathcal{K}_n^{\vec{\ell}} \cup \ell^{(\mathcal{K}'_n)} \quad \text{for } n = 1, \dots, N-2, \\ \text{and for each } k \in \mathcal{N}, \text{ let } n_k^{\vec{\ell}} &\text{ be the (single) value of } n \text{ such that } \ell^{(\mathcal{K}'_{n-1})} = k. \end{aligned} \quad (4.14)$$

Starting from the first \mathbf{V} -node \mathbf{V}_1 , its bifurcation choice $\ell^{(\emptyset)}$ determines the output value $(\emptyset, \dots, \emptyset, \ell^{(\emptyset)}, \emptyset, \dots, \emptyset) = (\emptyset, \dots, \emptyset, \mathcal{K}_1^{\vec{\ell}}, \emptyset, \dots, \emptyset)$ of the augmented relation $\lambda_{\mathbf{V}_1}^{\text{aug}}$ – which coincides with the input of node \mathbf{V}_2 ,³³ and which also determines the branch $\mathbf{V}_2^{\mathcal{K}_1^{\vec{\ell}}}$ that happens at that node. The corresponding component $\ell^{(\mathcal{K}'_1)}$ of the bifurcation choice $\vec{\ell}_1$ at that second \mathbf{V} -node \mathbf{V}_2 then determines the output value $(\emptyset, \dots, \emptyset, \mathcal{K}_1^{\vec{\ell}} \cup \ell^{(\mathcal{K}'_1)}, \emptyset, \dots, \emptyset) = (\emptyset, \dots, \emptyset, \mathcal{K}_2^{\vec{\ell}}, \emptyset, \dots, \emptyset)$ of the augmented relation $\lambda_{\mathbf{V}_2}^{\text{aug}}$ – which coincides with the input of node \mathbf{V}_3 , and which also determines the branch $\mathbf{V}_3^{\mathcal{K}_2^{\vec{\ell}}}$ that happens at that node. Continuing accordingly up to $\mathcal{K}_N^{\vec{\ell}}$, the bifurcation choices in $\vec{\ell}$ thus determine the sequence $(\mathcal{K}_1^{\vec{\ell}}, \mathcal{K}_2^{\vec{\ell}}, \dots, \mathcal{K}_{N-1}^{\vec{\ell}}, \mathcal{K}_N^{\vec{\ell}})$ of (the labels of) all branches that successively happen at the \mathbf{V} -nodes.

Note, furthermore, that each $\mathcal{K}_n^{\vec{\ell}} = \mathcal{K}_{n-1}^{\vec{\ell}} \cup \ell^{(\mathcal{K}'_{n-1})}$ in the output values $(\emptyset, \dots, \emptyset, \mathcal{K}_{n-1}^{\vec{\ell}} \cup \ell^{(\mathcal{K}'_{n-1})}, \emptyset, \dots, \emptyset)$ in the procedure above appears at a “new” position $k = \ell^{(\mathcal{K}'_{n-1})}$ in the tuple of index values, “unvisited” before. For each k there is therefore (as required) a single value of n such that $\mathcal{K}_n^k \neq \emptyset$, for which $\mathcal{X}_n^k = \mathcal{K}_n^{\vec{\ell}}$ – that value being $n_k^{\vec{\ell}}$, as introduced above. This determines the branch that happens at the node \mathbf{A}_k to be the one labelled by $\mathcal{K}_{n_k^{\vec{\ell}}}^{\vec{\ell}} \setminus k = \mathcal{K}_{n_k^{\vec{\ell}}-1}^{\vec{\ell}}$.

All in all, one thus finds that the collection of all bifurcation choices $\vec{\ell}$ determines in a unique manner all the branches that happen at all nodes. The choice relation $\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}}$ therefore defines a proper function, which can be written as

$$\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}} : \underbrace{\left(\vec{\ell} = (\ell^{(\mathcal{K}'_n)})_{n, \mathcal{K}'_n} \right)}_{\text{auxiliary inputs of the } \mathbf{V}\text{-nodes}} \mapsto \left(\underbrace{\left(\mathcal{K}_{n_k^{\vec{\ell}}-1}^{\vec{\ell}} \right)_{k \in \mathcal{N}}}_{\text{auxiliary outputs of the } \mathbf{A}\text{-nodes}}, \underbrace{\left(\mathcal{K}_n^{\vec{\ell}} \right)_{n=1}^{N-1}}_{\text{auxiliary outputs of the } \mathbf{V}\text{-nodes}} \right) \quad (4.15)$$

(see also [Eq. \(A.10\)](#) in [Appendix A](#)). This means that the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ is univocal.

Conceptually, each of the inputs of the choice function says ‘if the set of operations up to n was \mathcal{K}_n , then here is which operation should be the $(n+1)^{\text{th}}$ ’, which provides an algorithm to fully determine the global order.)

Furthermore, since $\mathcal{G}_{\text{QC-QC}(N)}$ is self-adjoint – in the sense that reversing the arrows and the routes gives the same routed graph, up to some relabelling $[\mathbf{V}_{n+1} \leftrightarrow \mathbf{V}_{N-n+1}, \mathcal{X}_n^k \leftrightarrow \mathcal{X}_{N-n+1}^k]$ (taking values $\emptyset \leftrightarrow \emptyset, \mathcal{K}_n \leftrightarrow (\mathcal{N} \setminus \mathcal{K}_n) \cup k$) – it readily follows that $\mathcal{G}_{\text{QC-QC}(N)}^\top$ is also univocal, and therefore that $\mathcal{G}_{\text{QC-QC}(N)}$ is bi-univocal.

4.2 Generic branch graph for QC-QCs

Having constructed a bi-univocal routed graph for QC-QCs, we will now derive its branch graph. To do so, we need to study the information flow between the branches by determining all strong and weak parent relations.

³³Recalling that the input and output index values of the \mathbf{A} -nodes are the same (when these are in the practical input/output sets), the routes at the \mathbf{A} -nodes that come between \mathbf{V}_n and \mathbf{V}_{n+1} are “transparent” for the index values of the \mathbf{V} -nodes.

4.2.1 Strong parents

To define strong parents we first need to specify the 1-dimensional values in the routed graph. We take these to be the index values \otimes (on all arrows where these may appear), while all other index values of the form \mathcal{K}_n are considered (a priori) to be higher-dimensional. Hence, for all internal arrows $A \in \text{Arr}_{\Gamma}^{\text{int}}$, we take $1\text{Dim}_A = \{\otimes\}$.

Recalling [Definition 2.12](#), given that each arrow has at most one possible 1-dimensional index value, then one can see that in order for some branch $\mathbf{V}_n^{\mathcal{K}_{n-1}}$ to be a strong parent of some branch $\mathbf{A}_k^{\mathcal{K}_{m-1}}$ (with \mathcal{K}_{m-1} , this requires that $\mathcal{K}_m \setminus k = \mathcal{K}_{n-1}$ (and hence $m = n$). Similarly, for some branch $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ to be a strong parent of some branch $\mathbf{V}_{m+1}^{\mathcal{K}'_m}$, this requires that $\mathcal{K}'_m = \mathcal{K}_n$ (hence $m = n$). In both cases, $\text{LinkVal}(\mathbf{V}_n^{\mathcal{K}_n \setminus k}, \mathbf{A}_k^{\mathcal{K}_n \setminus k})$ and $\text{LinkVal}(\mathbf{A}_k^{\mathcal{K}_n \setminus k}, \mathbf{V}_{n+1}^{\mathcal{K}_n})$ contain \mathcal{K}_n – which is not taken a priori to be 1-dimensional. We thus find that each $\mathbf{V}_n^{\mathcal{K}_n \setminus k}$ is a strong parent of each $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ (for all $\mathcal{K}_n, k \in \mathcal{K}_n$), and each $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ is a strong parent of $\mathbf{V}_{n+1}^{\mathcal{K}_n}$, and that these are the only “strong parent” relationships between branches.

4.2.2 Weak parents

Recall from [Section 2.3.2](#) that in order to determine whether a branch \mathbf{N}^β is a weak parent of another branch \mathbf{M}^γ , we need to analyse how the bifurcation choices for \mathbf{N}^β influence whether \mathbf{M}^γ happens or not. To do so, we consider the “ \mathbf{M}^γ -Happens functions” (cf. [Definition 2.13](#)) derived from the choice function of [Eq. \(4.15\)](#).

For the branches $\mathbf{A}_k^{\mathcal{K}_{n-1}}$ (with \mathcal{K}_{n-1} of the form $\mathcal{K}_n \setminus k$) of the \mathbf{A} -nodes, by considering the k^{th} element in the first tuple of outputs of the choice function, we obtain

$$\text{Happ}_{\mathbf{A}_k^{\mathcal{K}_{n-1}}} : \vec{\ell} \mapsto \delta_{\mathcal{K}_n^{\vec{\ell}}, \mathcal{K}_{n-1}}, \quad (4.16)$$

where $\mathcal{K}_n^{\vec{\ell}}$ and $n_k^{\vec{\ell}}$ are defined in [Eq. \(4.14\)](#), as a function of the input (the whole collection of bifurcation choices at the \mathbf{V} -nodes) $\vec{\ell} = (\ell^{(\mathcal{K}'_n)})_{n, \mathcal{K}'_n}$. It can be seen that $\mathcal{K}_n^{\vec{\ell}}$, and therefore the Boolean output value of $\text{Happ}_{\mathbf{A}_k^{\mathcal{K}_{n-1}}}$, depends non-trivially only on³⁴ $(\ell^{(\mathcal{K}_{m-1})})_{m \leq n, \mathcal{K}_{m-1} \subseteq \mathcal{K}_{n-1}, m < N$ – i.e., on the bifurcation choices made within the branches $\mathbf{V}_m^{\mathcal{K}_{m-1}}$, for all $m \leq \min(n, N-1)$ and all subsets $\mathcal{K}_{m-1} \subseteq \mathcal{K}_{n-1}$, so that these branches are precisely the weak parents of $\mathbf{A}_k^{\mathcal{K}_{n-1}}$.

For the branches $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ of the \mathbf{V} -nodes, by considering now the n^{th} element (for $n = 1, \dots, N-1$) in the second tuple of outputs of the choice function, we obtain

$$\text{Happ}_{\mathbf{V}_{n+1}^{\mathcal{K}_n}} : \vec{\ell} \mapsto \delta_{\mathcal{K}_n^{\vec{\ell}}, \mathcal{K}_n}. \quad (4.17)$$

Here it can be seen that $\mathcal{K}_n^{\vec{\ell}}$ (again defined as in [Eq. \(4.14\)](#)) depends non-trivially only on $(\ell^{(\mathcal{K}_m)})_{m < n, \mathcal{K}_m \subsetneq \mathcal{K}_n}$. The weak parents of $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ are thus the branches $\mathbf{V}_{m+1}^{\mathcal{K}_m}$ for all $m < n$ and $\mathcal{K}_m \subsetneq \mathcal{K}_n$.

As for the branches \mathbf{V}_1^\emptyset and $\mathbf{V}_{N+1}^{\mathcal{N}}$, since these are the only branches of their respective nodes \mathbf{V}_1 and \mathbf{V}_{N+1} , they do not have any weak parents.

(One may notice, for completeness, that since no bifurcation choices are made at the \mathbf{A} -nodes, their branches are not weak parents of any branch of any other node.)

To construct the branch graph below, we also need to characterise the weak parent relations in the adjoint graph. Here again, we use the fact that the graph $\mathcal{G}_{\text{QC-QC}(N)}$ is self-adjoint, as already mentioned to justify its bi-univocality (end of [Section 4.1.4](#)). By symmetry (through $\mathbf{A}_k^{\mathcal{K}_n \setminus k} \leftrightarrow \mathbf{A}_k^{\mathcal{N} \setminus \mathcal{K}_n}, \mathbf{V}_{n+1}^{\mathcal{K}_n} \leftrightarrow \mathbf{V}_{N-n+1}^{\mathcal{N} \setminus \mathcal{K}_n}$) we readily find that in the adjoint graph $\mathcal{G}_{\text{QC-QC}(N)}^\top$, the weak parents of the branches $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ are the branches $\mathbf{V}_{m+1}^{\mathcal{K}_m}$, for all $m \geq \max(n, 2)$ and all $\mathcal{K}_m \supseteq \mathcal{K}_n$, and that the weak parents of the branches $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ are the branches $\mathbf{V}_{m+1}^{\mathcal{K}_m}$, for all $m > n$ and all $\mathcal{K}_m \supsetneq \mathcal{K}_n$ (and for $n = 1, \dots, N-1$; \mathbf{V}_1^\emptyset and $\mathbf{V}_{N+1}^{\mathcal{N}}$ also do not have any weak parents in the adjoint graph).

³⁴Notice that the $\ell^{(\mathcal{K}_{N-1})}$'s are excluded here, because these are trivial bifurcation choices (with only one possible values, $\ell^{(\mathcal{K}_{N-1})} = \mathcal{N} \setminus \mathcal{K}_{N-1}$).

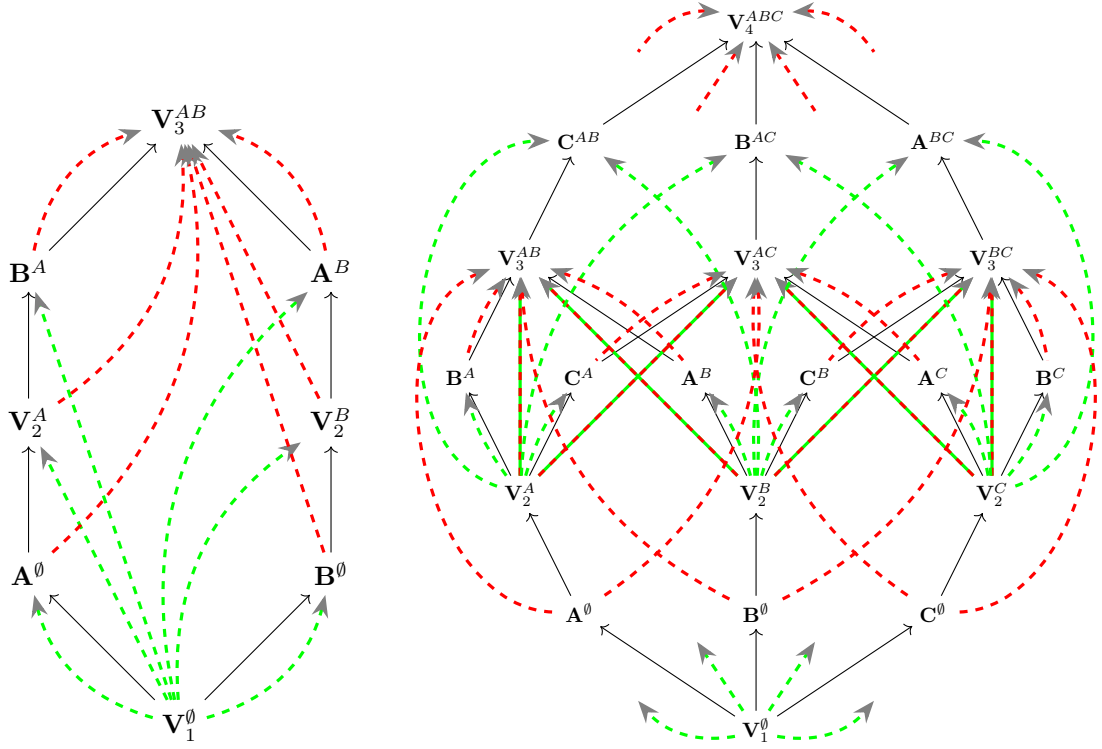


Figure 4.5: Generic branch graphs for $\mathcal{G}_{\text{QC-QC}(N)}$ with $N = 2$ (left) and with $N = 3$ (right). For simplicity of notation, $\{A, B\}$ and similar in the superscripts are written as AB . In the graph on the right, the unconnected green dashed arrows leaving from \mathbf{V}_1^0 are understood as being connected to all other nodes, except for \mathbf{V}_4^{ABC} ; similarly the unconnected red dashed arrows pointing to \mathbf{V}_4^{ABC} are understood as being connected to all other nodes, except for \mathbf{V}_1^0 . As discussed in Section 5.3, these branch graphs are closely related to Figures 9 and 24 in [39], and may be obtained by identifying P with \mathbf{V}_1^0 and F with $\mathbf{V}_3^{AB}/\mathbf{V}_4^{ABC}$, merging all nodes $\mathbf{V}_2^{k_1}$ with their unique parent and all nodes $\mathbf{V}_3^{k_2}$ with their unique child.

4.2.3 Constructing the branch graph and verifying the absence of loops

The nodes of the branch graph are the various branches $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ and $\mathbf{V}_{n+1}^{\mathcal{K}_n}$. According to Definition 2.15 and to our identification of the strong and weak parent relations above, these are connected by the following arrows:

- solid arrows $\mathbf{V}_n^{\mathcal{K}_n \setminus k} \rightarrow \mathbf{A}_k^{\mathcal{K}_n \setminus k}$ and $\mathbf{A}_k^{\mathcal{K}_n \setminus k} \rightarrow \mathbf{V}_{n+1}^{\mathcal{K}_n}$ for all $n, k = 1, \dots, N$ and all $\mathcal{K}_n \ni k$;
- green dashed arrows $\mathbf{V}_m^{\mathcal{K}_m \setminus k} \rightarrow \mathbf{A}_k^{\mathcal{K}_n \setminus k}$ for $1 \leq m \leq n \leq N$, $m < N$ and all $\mathcal{K}_m \supseteq \mathcal{K}_n \cup k$;
- red dashed arrows $\mathbf{A}_k^{\mathcal{K}_n \setminus k} \rightarrow \mathbf{V}_{m+1}^{\mathcal{K}_m}$ for $1 \leq n \leq m \leq N$, $m > 1$ and all $k, \mathcal{K}_n \ni k, \mathcal{K}_m \supseteq \mathcal{K}_n$;
- green dashed arrows $\mathbf{V}_{m+1}^{\mathcal{K}_m} \rightarrow \mathbf{V}_{n+1}^{\mathcal{K}_n}$ for $0 \leq m < n < N$ and all $\mathcal{K}_m \subsetneq \mathcal{K}_n$;
- red dashed arrows $\mathbf{V}_{n+1}^{\mathcal{K}_n} \rightarrow \mathbf{V}_{m+1}^{\mathcal{K}_m}$ for $0 < n < m \leq N$ and all $\mathcal{K}_n \subsetneq \mathcal{K}_m$.

Fig. 4.5 shows the generic branch graphs for $\mathcal{G}_{\text{QC-QC}(N)}$ with $N = 2$ and $N = 3$. One can clearly see the symmetry of the graph, due to the above-mentioned self-adjointness of $\mathcal{G}_{\text{QC-QC}(N)}$: flipping the branch graph upside-down, relabelling $\mathbf{A}_k^{\mathcal{K}_n \setminus k} \leftrightarrow \mathbf{A}_k^{\mathcal{N} \setminus \mathcal{K}_n}$ and $\mathbf{V}_{n+1}^{\mathcal{K}_n} \leftrightarrow \mathbf{V}_{N-n+1}^{\mathcal{N} \setminus \mathcal{K}_n}$, flipping the orientation of the arrows and swapping the green and red colours gives the same branch graph.

Notice that along each (solid or dashed) arrow described above, either the size of the branch indicator $\mathcal{K}_n(\setminus k)$ strictly increases, or it stays the same and the arrow goes from from a \mathbf{V} -node to a \mathbf{A} -node. This implies that there cannot be any loop in the graph: indeed, since the cardinality of the branch indicators cannot decrease, a loop would also require at least one arrow between

two nodes of the same type, or from an **A**-node to a **V**-node, with the same branch indicator cardinality.

It then follows that the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, with its 1-dimensional index values taken to be \mathbb{Q} , is valid (according to [Definition 2.17](#)). According to [Theorem 2.23](#), its skeletal supermap is therefore a routed superunitary. We will now show how to flesh it out and obtain any QC-QC.

4.3 Obtaining any QC-QC: Fleshing out the skeletal supermap for $\mathcal{G}_{\text{QC-QC}(N)}$

Having demonstrated the validity of the routed graph that we defined in [Section 4.1](#), we now finally demonstrate that and how we can use it to represent QC-QCs. We begin by obtaining a skeletal supermap for the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, associating each arrow with a Hilbert space of suitable dimension.

Then, we provide a fleshing out of both the **V**- and the **A**-nodes. The general technique we will use for this purpose is to specify general isomorphisms $J_{\mathbf{V}_{n+1}}^{\text{in/out}}$ and $J_{\mathbf{A}_k}^{\text{in/out}}$, which will be wrapped around all transformations performed in the slots of the skeletal supermap. They will serve to translate between spaces and representations natural to routed quantum circuits, e.g. $\mathcal{H}^{\text{in/out}(\mathbf{V}_{n+1})}$ or $\mathcal{H}^{\text{in/out}(\mathbf{A}_k)}$, and the spaces known and used in the QC-QC framework, e.g. $\mathcal{H}^{A_k^{I/O}}$, $\mathcal{H}^{\bar{A}_n^{I/O}}$, \mathcal{H}^{α_n} or $\mathcal{H}^{C_n^{(')}}$.

This will enable the use of the original QC-QC construction in the slots of the skeletal supermap, and leave us with a fleshing out for the **V**-nodes by plugging in the (routed) isometries \bar{V}_{n+1} for the “internal” nodes, dependent on the particular QC-QC, as well as with a fleshing out for the **A**-nodes, with monopartite superunitaries independent of the particular QC-QC, maintaining an unsectorised slot for each agent operation A_k .

We conclude this section by demonstrating that the fleshing out of the skeletal supermap indeed yields the same supermap as the QC-QC construction stated in [Section 3](#), and straightforwardly generalise to mixed QC-QCs.

4.3.1 The skeletal supermap

As a first step to obtain a supermap associated with the generic routed graph stated in [Definition 4.2](#), we use [Definition 2.21](#) to obtain a skeletal supermap. Here, each arrow A connecting two nodes $\mathbf{M} \rightarrow \mathbf{N}$ is associated with a sectorised Hilbert space $\mathcal{H}^{\mathbf{M} \rightarrow \mathbf{N}} = \bigoplus_{k_A} \mathcal{H}_{k_A}^{\mathbf{M} \rightarrow \mathbf{N}}$.

As established in [Section 4.1](#), for each internal (i.e. not open-ended) arrow, its index \mathcal{X}_n^k takes values among the n -element subsets \mathcal{K}_n of \mathcal{N} that contain k , or the null value \mathbb{Q} . As stated in [Section 4.2](#), we take $1\text{Dim}_{\mathbf{M} \rightarrow \mathbf{N}} = \{\mathbb{Q}\}$, implying the associated sector to be one-dimensional and be denoted as $\mathcal{H}_{\mathbb{Q}}^{\mathbf{M} \rightarrow \mathbf{N}} \cong \mathbb{C}$. While for the higher-dimensional sectors, our generic notation for an index $\mathcal{X}_n^k = \mathcal{K}_n \in \text{Ind}_{\mathbf{M} \rightarrow \mathbf{N}}$ would be $\mathcal{H}_{\mathcal{K}_n}^{\mathbf{M} \rightarrow \mathbf{N}}$, we will deviate from this notation going forward, writing $\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{M} \rightarrow \mathbf{N}}$. Thereby, we reintroduce and single out k (which disappears when writing $\mathcal{X}_n^k = \mathcal{K}_n$) and relate more directly to the notations of the control state $|\mathcal{K}_n \setminus k, k\rangle^{C_n^{(')}} for QC-QCs. Altogether, we endow the arrows with a sectorised Hilbert space as follows for $n, k = 1, \dots, N$:$

$$\begin{aligned} \mathbf{V}_n \xrightarrow{\mathcal{X}_n^k} \mathbf{A}_k : \quad \mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} &:= \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, \\ \mathbf{A}_k \xrightarrow{\mathcal{X}_n^k} \mathbf{V}_{n+1} : \quad \mathcal{H}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} &:= \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}. \end{aligned} \quad (4.18)$$

Specifically, we choose these spaces (and thereby, specify a dimension assignment) such that $\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \cong \mathcal{H}^{A_k^I} \otimes \mathcal{H}^{\alpha_n} \cong \mathcal{H}^{\bar{A}_n^I} \otimes \mathcal{H}^{\alpha_n}$, $\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \cong \mathcal{H}^{A_k^O} \otimes \mathcal{H}^{\alpha_n} \cong \mathcal{H}^{\bar{A}_n^O} \otimes \mathcal{H}^{\alpha_n}$ where the respective sectors each are a copy of their respective isomorphic space introduced in [Section 3.2](#).

To the two open-ended arrows we attach non-sectorised spaces:

$$\emptyset \rightarrow \mathbf{V}_1 : \quad \mathcal{H}^P, \quad \mathbf{V}_{N+1} \xrightarrow{\mathcal{N} \cup F} : \quad \mathcal{H}^F. \quad (4.19)$$

Thereby, we obtain the following “formal” input/output spaces, as introduced in [Section 2.4](#), for each node (for $n, k = 1, \dots, N$):

$$\begin{aligned}
\mathcal{H}^{\text{in}(\mathbf{V}_{n+1})} &= \bigotimes_{k=1}^N \mathcal{H}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} = \bigotimes_{k=1}^N \left(\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \right), & \mathcal{H}^{\text{in}(\mathbf{V}_1)} &=: \mathcal{H}^P, \\
\mathcal{H}^{\text{out}(\mathbf{V}_n)} &= \bigotimes_{k=1}^N \mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} = \bigotimes_{k=1}^N \left(\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \right), & \mathcal{H}^{\text{out}(\mathbf{V}_{N+1})} &=: \mathcal{H}^F, \\
\mathcal{H}^{\text{in}(\mathbf{A}_k)} &= \bigotimes_{n=1}^N \mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} = \bigotimes_{n=1}^N \left(\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \right), \\
\mathcal{H}^{\text{out}(\mathbf{A}_k)} &= \bigotimes_{n=1}^N \mathcal{H}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} = \bigotimes_{n=1}^N \left(\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \oplus \mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \right). \tag{4.20}
\end{aligned}$$

These spaces are restricted by index constraints on the routed graph, which imply sectorial correlations between the incoming/outgoing spaces at each slot of the skeletal supermap. Specifically, for each valid index value assignment, the parties A_k act in a certain order. For each space $\mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}$ (or $\mathcal{H}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}$), its higher-dimensional sector³⁵ $\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}$ (or $\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}$) is associated with to a party A_k being in n^{th} position. For each $n' \neq n$, we then find a sectorial correlation between $\mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}$ and $\mathcal{H}^{\mathbf{V}_{n'} \rightarrow \mathbf{A}_k}$, selecting the respective 1-dimensional sector $\mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_{n'} \rightarrow \mathbf{A}_k}$ (or $\mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n'+1}}$) for $\mathcal{H}^{\mathbf{V}_{n'} \rightarrow \mathbf{A}_k}$ (or $\mathcal{H}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n'+1}}$). For each $k' \neq k$ and $\mathcal{H}^{\mathbf{V}_n \rightarrow \mathbf{A}_{k'}}$ (or $\mathcal{H}^{\mathbf{A}_{k'} \rightarrow \mathbf{V}_{n+1}}$), we find analogous results. In a slight abuse of notation, we will drop these 1-dimensional tensor factors when denoting the practical subspaces.³⁶ Therefore, the practical input/output subspaces for each node are given by:

$$\begin{aligned}
\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{V}_{n+1})} &= \bigoplus_{k=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, & \mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{V}_1)} &= \mathcal{H}^P, \\
\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_n)} &= \bigoplus_{k=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, & \mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_{N+1})} &= \mathcal{H}^F \\
\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)} &= \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, \\
\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{A}_k)} &= \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}. \tag{4.22}
\end{aligned}$$

Remark 4.3. The QC-QC control systems C_n can be seen as providing an isomorphic embedding of $\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_n)}$ into a joint space, as $\mathcal{H}^{C_n} \cong \bigoplus_{k=1}^N \bigoplus_{\mathcal{K}_n \ni k} \text{span}\{|\mathcal{K}_n \setminus k, k\rangle^{C_n}\}$. Similarly, the control systems C'_n correspond to $\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{V}_{n+1})}$.

With these, we have introduced (and simplified) the Hilbert spaces needed to construct the skeletal supermap, as explained in [Section 2.5](#), with the nodes replaced by empty slots connected by identities following the respective routes. In order to obtain the desired QC-QC, we will then flesh out the slots for both the \mathbf{V} - and \mathbf{A} -nodes, as specified in the following subsections.

³⁵Composed of multiple subsectors, for each set of “previous parties” $\mathcal{K}_n \setminus k$.

³⁶Specifically, we will identify a Hilbert space \mathcal{H} with its tensor product with 1-dimensional Hilbert spaces. For instance, we more explicitly have

$$\mathcal{H}^{\text{in}(\mathbf{A}_k)} \supset \tilde{\mathcal{H}}_{\text{prac}}^{\text{in}(\mathbf{A}_k)} = \bigoplus_{n=1}^N \left[\bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \otimes \bigotimes_{n' \neq n} \mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_{n'} \rightarrow \mathbf{A}_k} \right] \cong \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} = \mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)}. \tag{4.21}$$

4.3.2 Fleshing out for the \mathbf{V} -nodes

Having constructed the skeletal supermap associated with the generic routed graph, we first proceed by fleshing out the \mathbf{V} -nodes with routed isometries. In order to obtain the desired QC-QC, the idea of this fleshing out is to use precisely the internal isometries \tilde{V}_{n+1} of the QC-QC in question, acting on appropriate subspaces of the Hilbert spaces associated with the \mathbf{V} -nodes. To this end, we first introduce two isomorphisms $J_{\mathbf{V}_{n+1}}^{\text{in}}$ and $J_{\mathbf{V}_{n+1}}^{\text{out}}$ for each node \mathbf{V}_{n+1} . They identify explicitly the subspaces of the practical spaces onto which the internal isometries \tilde{V}_{n+1} can be applied – recalling that each $\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \cong \mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_n^O} \otimes \mathcal{H}^{\alpha_n}$ and each³⁷ $\mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_k} \cong \mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_{n+1}^I} \otimes \mathcal{H}^{\alpha_{n+1}}$:

$$\begin{aligned}
(\text{for } n = 1, \dots, N) \quad J_{\mathbf{V}_{n+1}}^{\text{in}} : \mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{V}_{n+1})} &= \bigoplus_{k=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} \rightarrow \mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_n^O} \otimes \mathcal{H}^{\alpha_n} \otimes \mathcal{H}^{C'_n}, \\
|\psi\rangle \in \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}} &\mapsto |\psi\rangle^{\tilde{A}_n^O \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n}, \\
(\text{for } n = 0, \dots, N-1) \quad J_{\mathbf{V}_{n+1}}^{\text{out}} : \mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_{n+1}^I} \otimes \mathcal{H}^{\alpha_{n+1}} \otimes \mathcal{H}^{C_{n+1}} &\rightarrow \mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_{n+1})} = \bigoplus_{\ell=1}^N \bigoplus_{\mathcal{K}_n \not\ni \ell} \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}, \\
|\psi\rangle^{\tilde{A}_{n+1}^I \alpha_{n+1}} \otimes |\mathcal{K}_n, \ell\rangle^{C_{n+1}} &\mapsto |\psi\rangle \in \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}. \tag{4.23}
\end{aligned}$$

These isomorphisms are represented on Fig. 4.6. As Choi vectors, they can be written as

$$\begin{aligned}
|J_{\mathbf{V}_{n+1}}^{\text{in}}\rangle\rangle &= \bigoplus_{k=1}^N \bigoplus_{\mathcal{K}_n \ni k} |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \tilde{A}_n^O \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n}, \\
|J_{\mathbf{V}_{n+1}}^{\text{out}}\rangle\rangle &= \bigoplus_{\ell=1}^N \bigoplus_{\mathcal{K}_n \not\ni \ell} |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_{n+1}^I \alpha_{n+1}}, \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}} \otimes |\mathcal{K}_n, \ell\rangle^{C_{n+1}}. \tag{4.24}
\end{aligned}$$

Here, we have kept the whole sectorised Hilbert spaces explicit in the superscripts of $|\mathbb{1}\rangle\rangle$ to be clear with regard to the upper and lower indices. Using Eq. (3.6) for $|\tilde{V}_{n+1}\rangle\rangle$, the Choi vector of the overall operation that fleshes out the slot for \mathbf{V}_{n+1} (where we abuse slightly the notation by also calling this operation \mathbf{V}_{n+1}), obtained by composing the above two isomorphisms with \tilde{V}_{n+1} (for $n = 1, \dots, N-1$), is then given by (cf. Fig. 4.6)

$$\begin{aligned}
|\mathbf{V}_{n+1}\rangle\rangle &= |J_{\mathbf{V}_{n+1}}^{\text{in}}\rangle\rangle * |\tilde{V}_{n+1}\rangle\rangle * |J_{\mathbf{V}_{n+1}}^{\text{out}}\rangle\rangle \\
&= |J_{\mathbf{V}_{n+1}}^{\text{in}}\rangle\rangle * \left(\sum_{\substack{\mathcal{K}_n, k \in \mathcal{K}_n, \\ \ell \notin \mathcal{K}_n}} |\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle^{\tilde{A}_n^O \alpha_n \tilde{A}_{n+1}^I \alpha_{n+1}} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n} \otimes |\mathcal{K}_n, \ell\rangle^{C_{n+1}} \right) * |J_{\mathbf{V}_{n+1}}^{\text{out}}\rangle\rangle \\
&= \bigoplus_{\substack{\mathcal{K}_n, k \in \mathcal{K}_n, \\ \ell \notin \mathcal{K}_n}} |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \tilde{A}_n^O \alpha_n} * |\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle^{\tilde{A}_n^O \alpha_n \tilde{A}_{n+1}^I \alpha_{n+1}} * |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}} \\
&= \bigoplus_{\substack{\mathcal{K}_n, k \in \mathcal{K}_n, \\ \ell \notin \mathcal{K}_n}} |V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}}, \tag{4.25}
\end{aligned}$$

where $|V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}}$ is formally the same as $|\tilde{V}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle^{\tilde{A}_n^O \alpha_n \tilde{A}_{n+1}^I \alpha_{n+1}}$ (or as $|V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle\rangle^{\tilde{A}_k^O \alpha_n \tilde{A}_\ell^I \alpha_{n+1}}$), just written in the (isomorphic) spaces $\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}$, $\mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}$ instead of $\mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_n^O \alpha_n}$, $\mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_{n+1}^I \alpha_{n+1}}$ (or of $\mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_k^O \alpha_n}$, $\mathcal{H}_{\mathcal{K}_n}^{\tilde{A}_\ell^I \alpha_{n+1}}$), as introduced in Section 3.2. For the nodes \mathbf{V}_1 and

³⁷As we consider here the node \mathbf{V}_{n+1} , then rather than considering $\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_n)}$ as in Eq. (4.22), we will consider $\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{V}_{n+1})}$ – hence the change in the notations: $n \rightarrow n+1$, $(\mathcal{K}_n \setminus k, k) \rightarrow (\mathcal{K}_n, \ell)$.

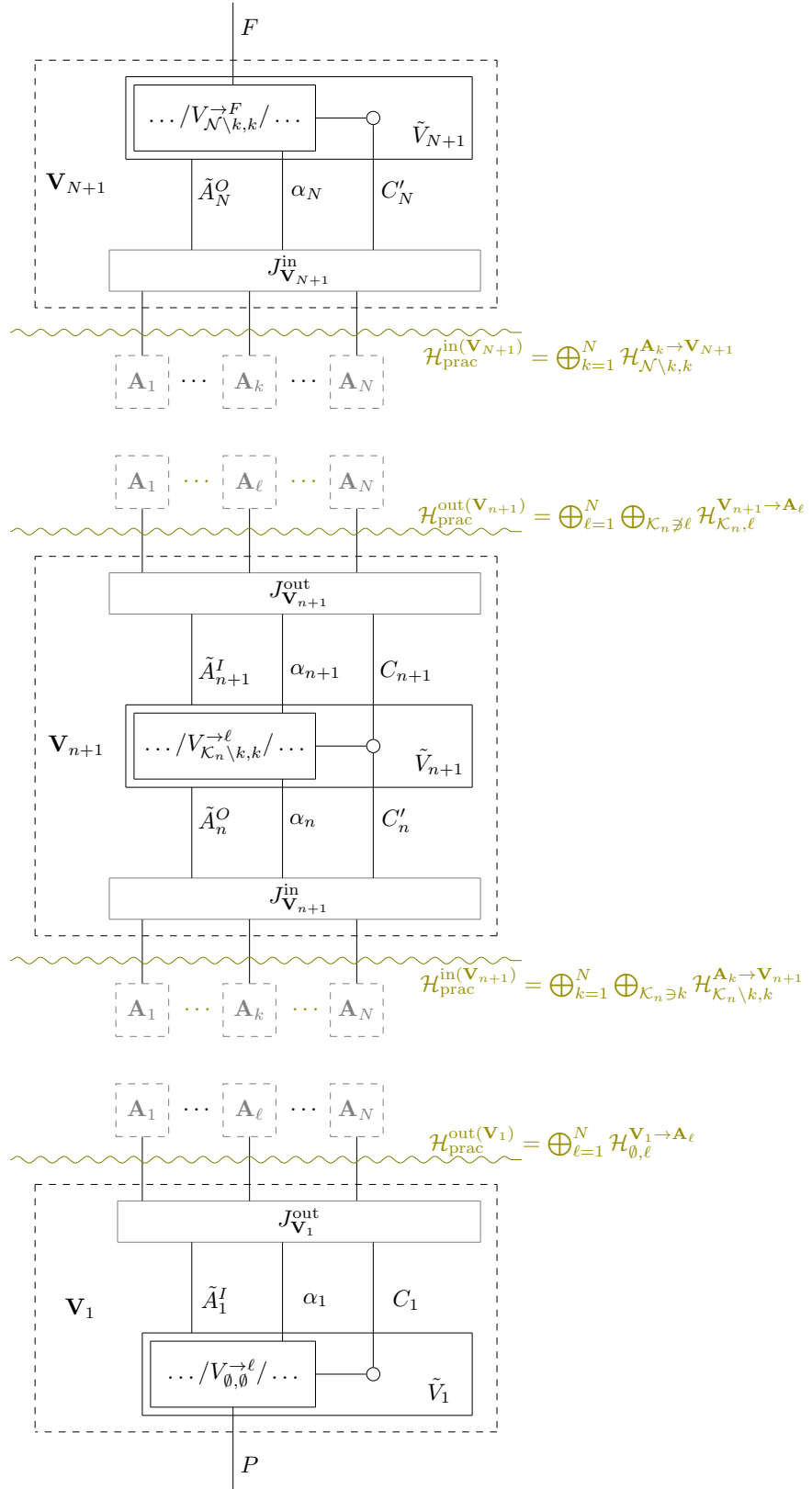


Figure 4.6: Fleshing out for the \mathbf{V} -nodes (with a channel). For each slot \mathbf{V}_{n+1} , isomorphisms $J_{\mathbf{V}_{n+1}}^{\text{in/out}}$ translate between the skeletal supermap and its associated spaces in the neighbourhood of the slot, and the QC-QC systems $\tilde{A}_n^{I/O}$, α_n and C_n within the slot. In-between these isomorphisms, the QC-QC isometries \tilde{V}_n control whose agent \mathbf{A}_ℓ acts next dependent on the previous agents.

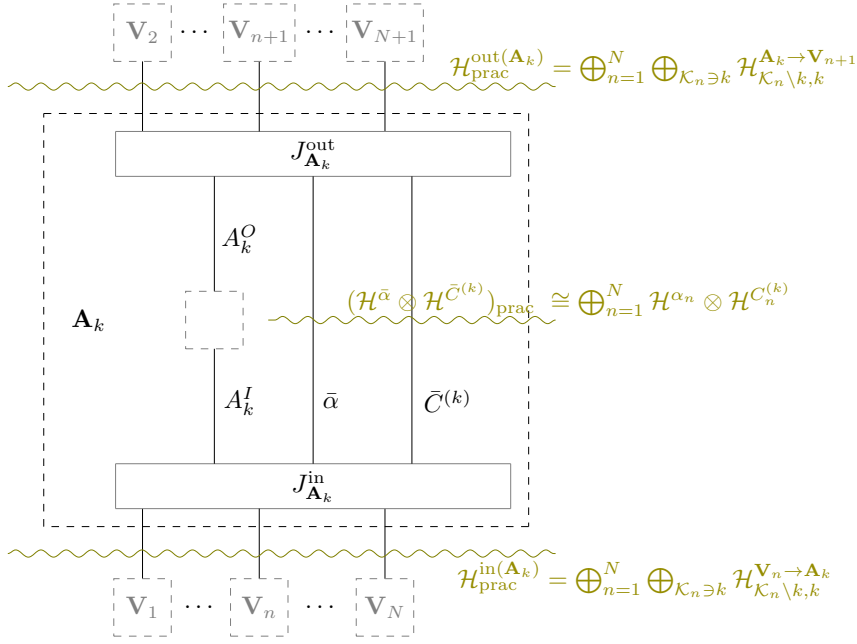


Figure 4.7: Fleshing out for the \mathbf{A} -nodes (with a 1-slot comb). After the fleshing out, an empty (unsectorised) slot $A_k^I \rightarrow A_k^O$ remains, associated with the free operation of an agent A_k , while in parallel, system $\bar{C}^{(k)}$ maintains the control system of the QC-QC. Two isomorphisms $J_{\mathbf{A}_k}^{\text{in/out}}$ translate between the skeletal supermap and its associated spaces in the neighbourhood of the slot, and the QC-QC systems $A_k^{I/O}$, α_n and $C_n^{(k)}$ within the slot. Altogether, the fleshing out of \mathbf{A}_k is independent of the QC-QC under consideration.

\mathbf{V}_{N+1} , the fleshing-out operations are

$$|\mathbf{V}_1\rangle\rangle = |\tilde{V}_1\rangle\rangle * |J_{\mathbf{V}_1}^{\text{out}}\rangle\rangle = \bigoplus_{\ell \in \mathcal{N}} |V_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle\rangle^{P, \mathcal{H}_{\emptyset, \ell}^{V_1 \rightarrow A_\ell}}, \quad (4.26)$$

$$|\mathbf{V}_{N+1}\rangle\rangle = |J_{\mathbf{V}_{N+1}}^{\text{in}}\rangle\rangle * |\tilde{V}_{N+1}\rangle\rangle = \bigoplus_{k \in \mathcal{N}} |V_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle\rangle^{\mathcal{H}_{\mathcal{N} \setminus k, k}^{A_k \rightarrow V_{N+1}, F}}, \quad (4.27)$$

respectively. Analogously, $|V_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle\rangle^{P, \mathcal{H}_{\emptyset, \ell}^{V_1 \rightarrow A_\ell}}$ and $|V_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle\rangle^{\mathcal{H}_{\mathcal{N} \setminus k, k}^{A_k \rightarrow V_{N+1}, F}}$ are formally the same as $|\tilde{V}_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle\rangle^{P, \tilde{A}_1^I \alpha_1}$ and $|\tilde{V}_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle\rangle^{\tilde{A}_N^O \alpha_N, F}$ (or as $|V_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle\rangle^{P, A_\ell^I \alpha_1}$ and $|V_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle\rangle^{A_k^O \alpha_N, F}$), respectively.

Note that these operations are indeed routed isometries on the (practical) spaces: they explicitly follow the routes at each node, and they are isometries due to the fact that the \tilde{V}_{n+1} of QC-QCs are isometries (on the “practical” spaces of the QC-QC). In fact, *every* fleshing out for the \mathbf{V} -nodes specifies a QC-QC, given that the decomposition in Eq. (3.6) precisely ensures that each \tilde{V}_n follows the relevant routes.

4.3.3 Fleshing out for the \mathbf{A} -nodes

As the next step, as illustrated in Fig. 4.7, we flesh out the slots for the \mathbf{A} -nodes using 1-slot superunitaries of the type

$$(\mathcal{H}^{A_k^I} \rightarrow \mathcal{H}^{A_k^O}) \longrightarrow (\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)} \rightarrow \mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{A}_k)}), \quad (4.28)$$

which will take the routed slot of type $\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)} \rightarrow \mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{A}_k)}$ to a non-routed slot of type $\mathcal{H}^{A_k^I} \rightarrow \mathcal{H}^{A_k^O}$.

For this purpose, we need to consider the subspace associated with the specific agent node \mathbf{A}_k , rather than the tilded spaces $\mathcal{H}^{\tilde{A}_n^I/O}$, which are employed in Eq. (3.5) to obtain a QC-QC implementation. To do so, we introduce a Hilbert space $\mathcal{H}^{\bar{C}^{(k)}}$, serving as the control space for the

causal order, reorganised to be grouped by the ‘‘current’’ agent A_k :

$$\mathcal{H}^{\bar{C}^{(k)}} := \bigoplus_{n=1}^N \mathcal{H}^{C_n^{(k)}} \quad \text{with} \quad \mathcal{H}^{C_n^{(k)}} := \bigoplus_{\mathcal{K}_n \ni k} \text{span}\{|\mathcal{K}_n \setminus k, k\rangle^{C_n}\}. \quad (4.29)$$

Due to the tensor product with the ancillary space, this decomposition by k carries over to the ancillary system α_n , which takes different values for n dependent on the system $C_n^{(k)}$ we consider. Accordingly, we introduce $\mathcal{H}^{\bar{\alpha}} := \bigoplus_{n=1}^N \mathcal{H}^{\alpha_n}$. Following the treatment of indexed wires in the skeletal supermap picture, we differentiate between the formal tensor product of $\mathcal{H}^{\bar{C}^{(k)}}$ and $\mathcal{H}^{\bar{\alpha}}$ and their more restricted practical space

$$(\mathcal{H}^{\bar{\alpha}} \otimes \mathcal{H}^{\bar{C}^{(k)}})_{\text{prac}} \cong \bigoplus_{n=1}^N \mathcal{H}^{\alpha_n} \otimes \mathcal{H}^{C_n^{(k)}}. \quad (4.30)$$

With this in place, we introduce two isomorphisms $J_{\mathbf{A}_k}^{\text{in}}$ and $J_{\mathbf{A}_k}^{\text{out}}$ for each node \mathbf{A}_k , transforming the incoming and outgoing spaces $\mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)}$ and $\mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{A}_k)}$ to the spaces used by the QC-QC formalism, as introduced in [Section 3](#):

$$\begin{aligned} J_{\mathbf{A}_k}^{\text{in}} : \quad & \mathcal{H}_{\text{prac}}^{\text{in}(\mathbf{A}_k)} = \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \rightarrow \mathcal{H}^{A_k^I} \otimes (\mathcal{H}^{\bar{\alpha}} \otimes \mathcal{H}^{\bar{C}^{(k)}})_{\text{prac}}, \\ & |\psi\rangle \in \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k} \mapsto |\psi\rangle^{A_k^I \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C_n^{(k)}}, \\ J_{\mathbf{A}_k}^{\text{out}} : \quad & \mathcal{H}^{A_k^O} \otimes (\mathcal{H}^{\bar{\alpha}} \otimes \mathcal{H}^{\bar{C}^{(k)}})_{\text{prac}} \rightarrow \mathcal{H}_{\text{prac}}^{\text{out}(\mathbf{A}_k)} = \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \\ & |\psi\rangle^{A_k^O \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C_n^{(k)}} \mapsto |\psi\rangle \in \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}. \end{aligned} \quad (4.31)$$

As Choi vectors, these isomorphisms can be represented as:

$$\begin{aligned} |J_{\mathbf{A}_k}^{\text{in}}\rangle\rangle &= \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, A_k^I \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C_n^{(k)}}, \\ |J_{\mathbf{A}_k}^{\text{out}}\rangle\rangle &= \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} |\mathbb{1}\rangle\rangle_{A_k^O \alpha_n, \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C_n^{(k)}}, \end{aligned} \quad (4.32)$$

using the same notational conventions as in the previous section. So the Choi vector of the overall 1-slot superunitary that fleshes out the slot for \mathbf{A}_k (where again, we abuse slightly the notation by also calling this operation \mathbf{A}_k), as pictured in [Fig. 4.7](#), is directly given as

$$|\mathbf{A}_k\rangle\rangle = |J_{\mathbf{A}_k}^{\text{in}}\rangle\rangle * |J_{\mathbf{A}_k}^{\text{out}}\rangle\rangle = \bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} |\mathbb{1}\rangle\rangle_{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, A_k^I \alpha_n} * |\mathbb{1}\rangle\rangle_{A_k^O \alpha_n, \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}}. \quad (4.33)$$

4.3.4 Composing the process vector and obtaining any QC-QC

Altogether, fleshing out the slots for all \mathbf{V} -nodes and all \mathbf{A} -nodes, with isometries and 1-slot superunitaries respectively, for the supermap we obtain the Choi vector

$$\begin{aligned}
& \left(\bigotimes_{n=0}^N |\mathbf{V}_{n+1}\rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) \\
&= \left(\left(\bigoplus_{\ell \in \mathcal{N}} |V_{\emptyset, \emptyset}^{\rightarrow \ell}\rangle \right)^{P, \mathcal{H}_{\emptyset, \ell}^{\mathbf{V}_1 \rightarrow \mathbf{A}_\ell}} \otimes \bigotimes_{n=1}^{N-1} \left(\bigoplus_{\substack{\mathcal{K}_n, \\ k \in \mathcal{K}_n, \\ \ell \notin \mathcal{K}_n}} |V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\rangle \right)^{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}, \mathcal{H}_{\mathcal{K}_n, \ell}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}} \right) \otimes \left(\bigoplus_{k \in \mathcal{N}} |V_{\mathcal{N} \setminus k, k}^{\rightarrow F}\rangle \right)^{\mathcal{H}_{\mathcal{N} \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{N+1}, F}} \\
& \quad * \left(\bigotimes_{k \in \mathcal{N}} \left(\bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k} |\mathbb{1}\rangle \right)^{\mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{V}_n \rightarrow \mathbf{A}_k}, A_k^I \alpha_n} * |\mathbb{1}\rangle \right)^{A_k^O \alpha_n, \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{n+1}}} \\
&= \sum_{(k_1, k_2, \dots, k_N)} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2}\rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * |V_{\{k_1\}, k_2}^{\rightarrow k_3}\rangle^{A_{k_2}^O \alpha_2, A_{k_3}^I \alpha_3} * \dots \\
& \quad \dots * |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N}\rangle^{A_{k_{N-1}}^O \alpha_{N-1}, A_{k_N}^I \alpha_N} * |V_{\{k_1, \dots, k_{N-1}\}, k_N}^{\rightarrow F}\rangle^{A_{k_N}^O \alpha_N, F} = |w_{\text{QC-QC}}\rangle, \tag{4.34}
\end{aligned}$$

composing Eqs. (4.25) to (4.27) and (4.33) so that we indeed recover the process vector of the QC-QC under consideration, as given in Eq. (3.7). For the detailed calculation, see Appendix B.

Having successfully reconstructed the process vector $|w_{\text{QC-QC}}\rangle$ representing a pure QC-QC, we still need to generalise to the mixed case to capture generic QC-QCs. As discussed in Section 3.2, we obtain the generalisation to mixed circuits directly by splitting $\mathcal{H}^F = \mathcal{H}^{F'} \otimes \mathcal{H}^{\alpha_F}$:

$$W = \text{Tr}_{\alpha_F} |w\rangle\langle w|. \tag{4.35}$$

As in the respective skeletal supermap, the output space of \mathbf{V}_{N+1} is not sectorised, discarding α_F does not require any special care in comparison to conventional quantum circuits.

We have thus proven our main result, which we recall here:

Theorem 4.1. Any N -slot quantum circuit with quantum control of causal order can be implemented by fleshing out a skeletal supermap obtained from the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ introduced in Definition 4.2.

5 Alternative Routed Quantum Circuit Descriptions

The construction of Section 4 provides a systematic approach to obtaining any N -slot QC-QC by fleshing out the skeletal supermap associated with the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$. It is important to note, however, that this construction does not provide a *unique* way to obtain any given QC-QC as a routed quantum circuit. Not only can any given QC-QC supermap have different decompositions into internal operations $V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}$, but the same routed supermap can also be obtained by fleshing out skeletal supermaps associated with different routed graphs. Indeed, as a case in point, Ref. [39] gave a routed circuit decomposition of the quantum switch different from what one would obtain using our construction from the graph $\mathcal{G}_{\text{QC-QC}(2)}$ (compare Fig. 6 from Ref. [39] to Fig. 4.2a here), and similarly for what the authors of [39] called the ‘‘Grenoble process’’ (introduced in [34]) in contrast to $\mathcal{G}_{\text{QC-QC}(3)}$ (compare Fig. 21 from Ref. [39] to Fig. 4.2b here).

In this section, we introduce some alternative routed quantum circuit decompositions for QC-QCs, demonstrating that they indeed yield the same sets of processes. First, we provide an alternative treatment of the QC-QC ancillary systems α_n in Section 5.1. In Section 5.2, we split apart the \mathbf{V} -nodes according to their branch structure, before rewriting them as pre- and post-processings for individual agents in Section 5.3. In certain cases, this allows for a less verbose representation that focuses on the connectivity between the individual agents. We then apply these alternatives to reconstruct previous routed quantum circuit decompositions for both the quantum switch and the Grenoble process. Finally, in Section 5.4, we sketch how certain restrictions regarding the index values practically attained allow one, for certain QC-QCs, to omit arrows from the routed graph.

5.1 An alternative treatment of the QC-QC ancillary systems

The first such variation concerns a more distinctive treatment of the ancillary systems, which is perhaps closer to how they are treated in the QC-QC framework. Recall that in that framework, an ancillary memory system α_n is passed between the internal isometries \tilde{V}_n and \tilde{V}_{n+1} , to which the agents' operations A_k do not have access, see Fig. 3.1. However, for the skeletal supermap associated with the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, in Section 4.3.1 we attach Hilbert spaces to the wires such that the ancillary system α_n is passed to the agent nodes \mathbf{A}_k from each node \mathbf{V}_n . The requirement that the agents' operations A_k cannot act on these ancillary degrees of freedom was instead enforced by the choice of fleshing out for the \mathbf{A} -nodes (see Fig. 4.7).

A natural, alternative definition of the routed graph is also possible, in which the separation of these ancillary systems from the agents' nodes is reflected in the structure of the graph and enforced by the skeletal supermap. Matching more closely the original formulation of QC-QCs, one could add further direct arrows $\mathbf{V}_n \rightarrow \mathbf{V}_{n+1}$ (for $1 \leq n \leq N$) to the routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, each associated with a single index value.³⁸ Going forward, we will drop these indices, just as for the open unindexed arrows in the routed graph, and denote the resulting (valid) routed graph as $\mathcal{G}_{\text{QC-QC}(N)}^\alpha$. The additional arrows can then be associated directly with the Hilbert spaces \mathcal{H}^{α_n} in the skeletal supermap. For the $N = 2$ and $N = 3$ cases, whose routed graphs are shown explicitly for our initial construction in Fig. 4.2, this would correspond simply to adding the direct arrows $\mathbf{V}_1 \rightarrow \mathbf{V}_2$, $\mathbf{V}_2 \rightarrow \mathbf{V}_3$ and (for $N = 3$) $\mathbf{V}_3 \rightarrow \mathbf{V}_4$. The associated branch graph is then similarly augmented with additional solid arrows, connecting compatible branches of \mathbf{V}_n and \mathbf{V}_{n+1} directly.³⁹

With this alternative treatment of the ancillary systems, the $\bar{\alpha}$ wire is removed from the fleshing out for the nodes \mathbf{A}_k , given in Eq. (4.31) and illustrated in Fig. 4.7. The fleshing out is thereby reduced to withholding the routing information encoded in $C_n^{(i)}$ from the operations A_k , while still producing an unsectorised slot. Considering this, we note that one could also refrain from fleshing out the \mathbf{A} -node slots entirely, as the agent's access to the control system $\bar{C}^{(k)}$ is restricted due to sectorial constraints. Specifically, we would obtain a generalisation of any QC-QC as a *routed supermap* – with its agents A_k performing routed operations dependent on the state of $\bar{C}^{(k)}$, which may not be modified (by e.g. a measurement in a basis incompatible with the sectorisation).⁴⁰

Ultimately, both representations are valid and allow one to construct any QC-QC via appropriate fleshing outs.⁴¹ However, as is generally the case, certain representations may be more natural depending on the physical situation, e.g. the specific implementation being considered. For example, in a quantum optics experiment where ancillary information is stored in a photonic degree of freedom (presumed to be) inaccessible to the agent (even though it passes through their labs), the first construction might seem more appropriate. By contrast, the second picture would more closely reflect a setup where the respective degree of freedom is stored elsewhere and never physically coincident with the agent, for instance in a physical system directly transferred between the internal isometries \tilde{V}_n .⁴² A similar distinction is also discussed in [51], where the authors argue for the importance of accounting for all physical transformations *in principle possible* that an agent might perform when characterising the causal structure of an experiment, rather than imposing artificial restrictions on their operations. Nonetheless, as our generic routed graph is not intended to represent any particular physical realisation of the indefinite causal order in a QC-QC, and as we do not analyse any specific causal structure here, these distinctions are of no immediate importance to this work.

³⁸For instance, $\text{Ind}_{\mathbf{V}_n \rightarrow \mathbf{V}_{n+1}} = \{\alpha_n\}$, the choice of value bearing no further relevance – it just in general should not be one-dimensional (so as to allow for non-trivial ancillary systems).

³⁹This follows as we obtain additional strong parent relations via the branch-linking values $\text{LinkVal}(\mathbf{V}_n^{\mathcal{K}_{n-1}}, \mathbf{V}_{n+1}^{\mathcal{K}_n}) = \{\alpha_n\}$ if $\mathcal{K}_{n-1} \subsetneq \mathcal{K}_n$, with $\text{LinkVal}(\mathbf{V}_n^{\mathcal{K}_{n-1}}, \mathbf{V}_{n+1}^{\mathcal{K}_n}) = \emptyset$ otherwise, with the arrow $\mathbf{V}_n \xrightarrow{\alpha_n} \mathbf{V}_{n+1}$ being of non-trivial dimension. The weak parent relations remain unchanged.

⁴⁰We may then consider $\bigotimes_{n=0}^N |\mathbf{V}_{n+1}\rangle\rangle$ as the Choi representation of this *routed supermap* (cf. Definition 2.20) generalisation of QC-QCs, while additionally bearing in mind the routes for the open slots associated with \mathbf{A}_k .

⁴¹Furthermore, both routed graphs generate the same set of routed supermaps, obtained from fleshing out a skeletal supermap associated with the respective routed graph.

⁴²A similar idea to physically prevent agents from accessing the control qubit is proposed in [50].

5.2 Splitting the internal nodes according to their branch structure

We continue by introducing an alternative representation to the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ of Definition 4.2, which splits apart the internal nodes \mathbf{V}_{n+1} according to their respective branch structure. Intuitively, the idea is to split each node \mathbf{V}_{n+1} into one individual node per branch $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$. Thus, the connectivity of the agents via the internal nodes becomes more apparent in the routed graph picture than for the original generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, at the cost of increasing the number of internal nodes exponentially.

Definition 5.1 ($\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$). The *split-node generic routed graph* $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ for an N -slot QC-QC is defined as follows:

- There are 2 types of nodes: “**A**-nodes” \mathbf{A}_k , for $k = 1, \dots, N$, and “ **$\hat{\mathbf{V}}$** -nodes” $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, for $n = 0, \dots, N$, with $\mathcal{K}_n \subseteq \mathcal{N}$ and $|\mathcal{K}_n| = n$. (In total, there are $N + 2^N$ nodes.)
- The indexed graph involves the following indexed arrows:

- open-ended arrows $\xrightarrow{\emptyset} \hat{\mathbf{V}}_1^\emptyset$ and $\hat{\mathbf{V}}_{N+1}^{\mathcal{N}} \xrightarrow{\mathcal{N} \cup F}$ (whose indices which we will usually drop in practice),
- $\forall n = 1, \dots, N, \forall k = 1, \dots, N, \forall \mathcal{K}_n \ni k$,

$$\hat{\mathbf{V}}_n^{\mathcal{K}_n \setminus k} \xrightarrow{\mathcal{X}_{\mathcal{K}_n \setminus k}^k} \mathbf{A}_k \quad \text{and} \quad \mathbf{A}_k \xrightarrow{\mathcal{X}_{\mathcal{K}_n \setminus k}^k} \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}, \quad (5.1)$$

with a given index $\mathcal{X}_{\mathcal{K}_n \setminus k}^k$ taking either $\mathcal{X}_{\mathcal{K}_n \setminus k}^k = \mathcal{K}_n$ as its value, or $\mathcal{X}_{\mathcal{K}_n \setminus k}^k = \emptyset$.

- The routes are specified such that
 - For each node \mathbf{A}_k , the lists of all input and output index values $(\mathcal{X}_{\mathcal{K}_n \setminus k}^k)_{\mathcal{K}_n \ni k} \in \text{Ind}_{\mathbf{A}_k}^{\text{in}}$ and $(\mathcal{X}_{\mathcal{K}_n \setminus k}^k)_{\mathcal{K}_n \ni k} \in \text{Ind}_{\mathbf{A}_k}^{\text{out}}$ must be equal – from now on we will then simply drop the primes – and must further satisfy $\exists! n, \exists! \mathcal{K}_n, \mathcal{X}_{\mathcal{K}_n \setminus k}^k \neq \emptyset$. Hence, the practical input and output sets of each node \mathbf{A}_k consist of (the same) lists of index values $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, for some n and some set $\mathcal{K}_n \ni k$ at the position corresponding to the arrow coming from node $\hat{\mathbf{V}}_n^{\mathcal{K}_n \setminus k}$ and to the arrow going to node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$.
 - For each node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, the lists of all input and output index values $(\mathcal{X}_{\mathcal{K}_n \setminus k}^k)_{k \in \mathcal{K}_n} \in \text{Ind}_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{in}}$ and $(\mathcal{X}_{\mathcal{K}_n}^\ell)_{\ell \notin \mathcal{K}_n} \in \text{Ind}_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{out}}$ must satisfy

$$\text{for } n = 0 \text{ and } n = N : \exists! \ell, \mathcal{X}_\emptyset^\ell \neq \emptyset \text{ and } \exists! k, \mathcal{X}_{\mathcal{N} \setminus k}^k \neq \emptyset, \text{ respectively.} \quad (5.2)$$

$$\text{for } \emptyset \subsetneq \mathcal{K}_n \subsetneq \mathcal{N} : \begin{cases} \text{either } \forall k, \mathcal{X}_{\mathcal{K}_n \setminus k}^k = \emptyset, \forall \ell, \mathcal{X}_{\mathcal{K}_n}^\ell = \emptyset \\ \text{or } \exists! k, \mathcal{X}_{\mathcal{K}_n \setminus k}^k \neq \emptyset, \exists! \ell, \mathcal{X}_{\mathcal{K}_n}^\ell \neq \emptyset \end{cases} \quad (5.3)$$

Hence, the practical input sets of each node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, for $0 < n \leq N$ and $\emptyset \subsetneq \mathcal{K}_n \subseteq \mathcal{N}$, consist either of lists of all- \emptyset index values, or of lists of index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$, with the set \mathcal{K}_n attached to the arrow coming from some node \mathbf{A}_k , with $k \in \mathcal{K}_n$. Similarly, the practical output sets of each node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, for $0 \leq n < N$ and $\emptyset \subseteq \mathcal{K}_n \subsetneq \mathcal{N}$, consist either of lists of all- \emptyset index values, or of lists of index values of the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell, \emptyset, \dots, \emptyset)$, with the set $\mathcal{K}_n \cup \ell$ attached to the arrow going to some node \mathbf{A}_ℓ , such that $\ell \notin \mathcal{K}_n$. Lists of all- \emptyset index values at the input are related to lists of all- \emptyset index values at the output; lists of index values the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n, \emptyset, \dots, \emptyset)$ at the input are related to lists of index values the form $(\emptyset, \dots, \emptyset, \mathcal{K}_n \cup \ell, \emptyset, \dots, \emptyset)$ at the output.

Both conditions are analogous to the non-split case (Definition 4.2), when replacing the subscripts n of the index labels with some $\mathcal{K}_n \setminus k$ and adapting the constraints in an appropriate

manner. They can be condensed into the global index constraint

$$\begin{aligned} \exists! (k_1, k_2, \dots, k_N), \mathcal{X}_\emptyset^{k_1} = \{k_1\}, \mathcal{X}_{\{k_1\}}^{k_2} = \{k_1, k_2\}, \dots, \mathcal{X}_{\mathcal{N} \setminus k_N}^{k_N} = \{k_1, k_2, \dots, k_N\} = \mathcal{N}, \\ \text{and all other } \mathcal{X}_{\mathcal{K}_n \setminus k}^k = \emptyset. \end{aligned} \quad (5.4)$$

Just like Eq. (4.5), this again captures the understanding that, along each global path, all operations A_k are applied once and only once, in a given order.

Contrasting this with the definition of the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ in Section 4.1.1, we see that splitting the \mathbf{V} -nodes forces the index *labels* to carry additional information to ensure their uniqueness across the graph. By contrast, we see that the index *values* remain virtually unchanged, such that the value \mathcal{K}_n does not encode any additional information other than being non- \emptyset . With the values being binary, we could alternatively pick $\mathcal{X}_{\mathcal{K}_n}^k \in \{0, 1\}$, and write the global index constraints as follows:

$$\exists! (k_1, k_2, \dots, k_N) : \quad \mathcal{X}_\emptyset^{k_1} = \mathcal{X}_{\{k_1\}}^{k_2} = \dots = \mathcal{X}_{\mathcal{N} \setminus k_N}^{k_N} = 1, \quad (5.5)$$

while all other $\mathcal{X}_{\mathcal{K}_n}^k = 0$. This would reproduce index values closer to the presentation of [39]. However, to keep representations as close to Section 4.1.1 as possible, we will stick to the more redundant convention in this work.

The routed graphs $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ for $N = 2$ and $N = 3$ are shown in Figs. 5.1a and 5.1b.

The branches $\mathbf{A}_k^{\mathcal{K}_n \setminus k}$ of the \mathbf{A} -nodes maintain the form given in Eq. (4.6), albeit with a larger number of indices (stemming from the larger number of $\hat{\mathbf{V}}$ -nodes) taking \emptyset as value. For the $\hat{\mathbf{V}}$ -nodes, we obtain a branch structure $\text{Bran}(\lambda_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}) = \{\mathbf{V}_{n+1}^{\mathcal{K}_n}, \bar{\mathbf{V}}_{n+1}^{\mathcal{K}_n}\}$, encoding precisely whether a given branch $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ happens or not, with $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ of the same form as in Eqs. (4.7) and (4.8) and with $\bar{\mathbf{V}}_{n+1}^{\mathcal{K}_n} = (\{\emptyset, \dots, \emptyset\} \rightarrow \{\emptyset, \dots, \emptyset\})$. Considering the index values \emptyset to be 1-dimensional, we may then construct the branch graph for $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$. It is closely related to the branch graph for \mathcal{G} , with the addition of the branches $\bar{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ being subject of no strong parent/child relations as well as the same weak parent/child relations as $\mathbf{V}_{n+1}^{\mathcal{K}_n}$. For the instance of $N = 2$, the respective branch graph is shown in Fig. 5.2. (A general comment on how node splitting affects the branch graph is presented in Fig. 5.3b.)

Lemma 5.2. Considering the index values \emptyset to be 1-dimensional, the split-node generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ is a valid routed graph.

The proof of this statement is given in Appendix C.

The larger number of nodes leads to an associated skeletal supermap of a different type, featuring a larger number of slots. As however all slots for $\hat{\mathbf{V}}$ -nodes are fleshed out by routed isometries rather than routed supermaps, this need not impact the type of the supermap we ultimately obtain.

We continue by showing that indeed, both routed graphs generate the same class of processes by means of routed circuit decompositions. For each decomposition for $\mathcal{G}_{\text{QC-QC}(N)}$, we can specify a closely related equivalent decomposition for $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$.

Lemma 5.3. Consider the two routed graphs $\mathcal{G}_{\text{QC-QC}(N)}$ and $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$. Restricting to the case in which all \mathbf{V}_n and $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ are fleshed out by routed isometries, there is a canonical mapping between equivalent routed circuit decompositions (as defined in Definition 2.26) for the two routed graphs.

The proof of this statement is given in Appendix C, using an approach that readily generalises to a variety of routed graphs and nodes.

For the skeletal supermap associated with $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$, the Hilbert spaces associated with the arrows going in/out of these branch nodes then factorise as

$$\mathcal{H}_{\text{prac}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} = \bigoplus_{k=1}^N \mathcal{H}_k^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}} \oplus \mathcal{H}_{\emptyset}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} \quad (5.6)$$

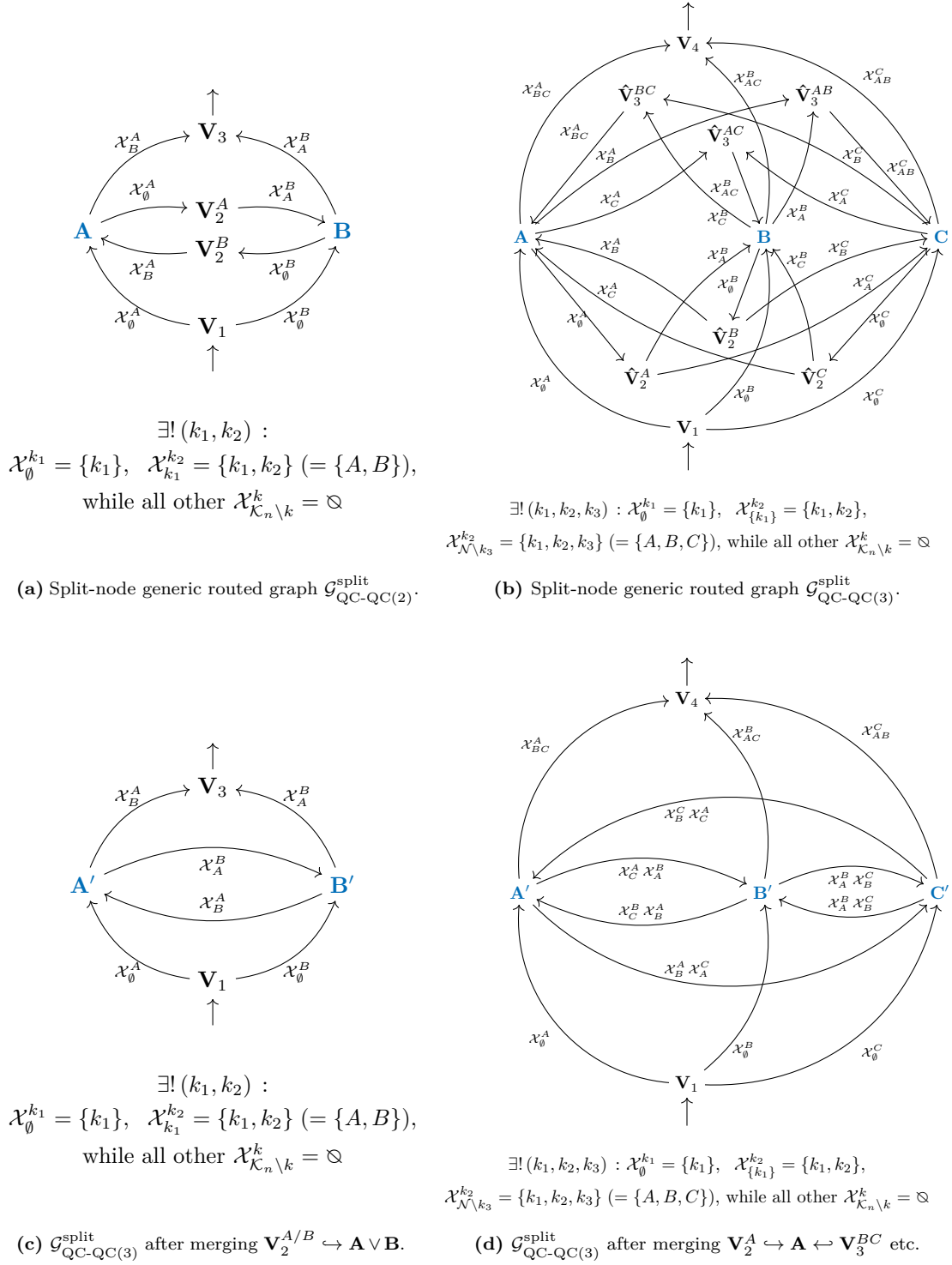


Figure 5.1: (a,b) The split-node generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ (for $N = 2, 3$) according to Definition 5.1, splitting the isometries \mathbf{V}_{n+1} featured in the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ shown in Fig. 4.2 into one node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ for each branch $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$. (c,d) The routed graphs obtained by merging internal isometries in $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ into agent nodes, denoted by \hookrightarrow , following the procedure established in Section 5.3. We obtain a reduced number of internal nodes, each of which is associated with applying not only their respective agent operations A_k , but also additional non-trivial “internal” pre- and post-processing encoding the action of the supermap. We thereby essentially recover the routed graphs shown in Figs. 6 and 21 of [39].

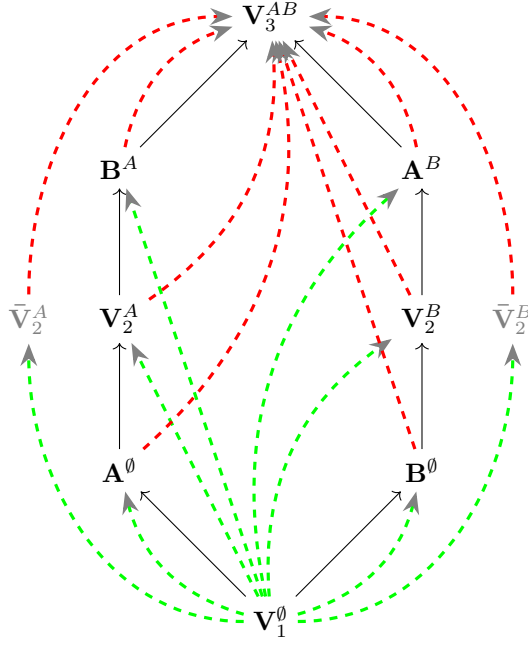


Figure 5.2: The branch graph for the split-node generic routed graph $\mathcal{G}_{\text{QC-QC}(2)}^{\text{split}}$ according to Definition 5.1, featuring additional branches $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ in addition to those present in the branch graph of $\mathcal{G}_{\text{QC-QC}(2)}$ shown in Fig. 4.5.

where $\mathcal{H}_k^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}} := \mathcal{H}_{\mathcal{K}_n \setminus k, k}^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}$ and $\mathcal{H}_{\mathbb{Q}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} := \bigotimes_{k=1}^N \mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}$, and similarly for $\mathcal{H}_{\text{prac}}^{\text{out}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})}$.

To obtain a given QC-QC from this alternative routed graph instead of $\mathcal{G}_{\text{QC-QC}(N)}$, we split the fleshing out for the \mathbf{V} -nodes along the same lines, obtaining $|J_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{in}} \rangle \rangle * |\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n} \rangle \rangle * |J_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{out}} \rangle \rangle$, where for $n = 1, \dots, N$:

$$J_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{in}} : \mathcal{H}_{\text{prac}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} = \bigoplus_{k=1}^N \mathcal{H}_k^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}} \oplus \mathcal{H}_{\mathbb{Q}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} \rightarrow \mathcal{H}^{\hat{\mathbf{A}}_n^O} \otimes \mathcal{H}^{\alpha_n} \otimes \mathcal{H}^{C'_n} \oplus \mathcal{H}_{\mathbb{Q}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})},$$

$$|\psi\rangle \in \mathcal{H}_k^{\mathbf{A}_k \rightarrow \hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}} \mapsto |\psi\rangle^{\hat{\mathbf{A}}_n^O \alpha_n} \otimes |\mathcal{K}_n \setminus k, k\rangle^{C'_n}, \quad (5.7)$$

$$|\Omega\rangle \in \mathcal{H}_{\mathbb{Q}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})} \mapsto |\Omega\rangle, \quad (5.8)$$

where $|\Omega\rangle$ is the single state in the 1-dimensional sector $\mathcal{H}_{\mathbb{Q}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})}$, and similarly define $|J_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{out}} \rangle \rangle$, while

$$|\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n} \rangle \rangle = |\tilde{\mathbf{V}}_{n+1}^{\mathcal{K}_n} \rangle \rangle \oplus |\bar{\mathbf{V}}_{n+1}^{\mathcal{K}_n} \rangle \rangle = \sum_{k \in \mathcal{K}_n, \ell \notin \mathcal{K}_n} |\tilde{\mathbf{V}}_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} \rangle \rangle^{\hat{\mathbf{A}}_n^O \alpha_n \bar{\mathbf{A}}_{n+1}^I \alpha_{n+1}} \oplus \mathbb{1}_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}) \rightarrow \text{out}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})}. \quad (5.9)$$

While in principle, we could apply an analogous technique for the routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\alpha}$, we would obtain a large number of arrows in the resulting graph in this case, connecting $\hat{\mathbf{V}}_n^{\mathcal{K}_n \setminus k}$ and $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ respectively.

Moreover, an analogous rewriting could be performed for the agent nodes \mathbf{A}_k , obtaining nodes $\mathbf{A}_k^{\mathcal{K}_n}$ for each $\mathcal{K}_n \subseteq \mathcal{N} \setminus k$. However, fleshing out the resulting slots for $\mathbf{A}_k^{\mathcal{K}_n}$ with monopartite supermaps (in the sense of a routed circuit decomposition according to Definition 2.26) would produce a supermap of a different type (than for $\mathcal{G}_{\text{QC-QC}(N)}$), increasing the number of slots within the routed supermap from N to 2^N . Furthermore, the resulting slot spaces are in direct sum form, including 1-dimensional Hilbert spaces $\mathcal{H}^{\text{in/out}(\hat{\mathbf{A}}_{\mathbb{Q}}^{\mathcal{K}_n})}$, preventing us from obtaining a unsectorised slot of non-trivial dimension by factorising. Therefore, no fleshing out for such $\hat{\mathbf{A}}$ -nodes could recover a meaningful unrouted supermap.

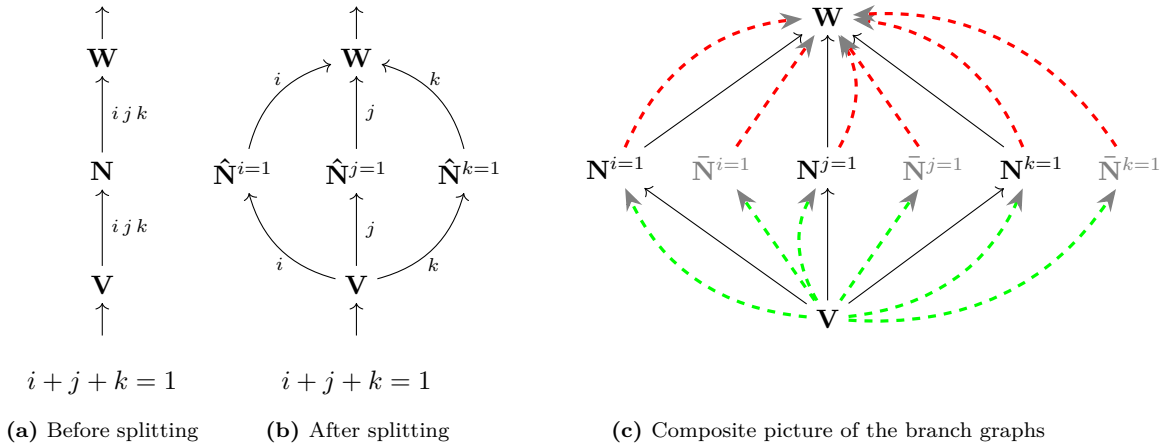


Figure 5.3: A general example for a node-splitting procedure, obtaining the routed graph shown in (b) from the routed graph shown in (a). While for the latter, we have $\text{Bran}(\lambda_{\mathbf{N}}) = \{\mathbf{N}^{i=1}, \mathbf{N}^{j=1}, \mathbf{N}^{k=1}\}$, each of these branches is represented by an individual node $\hat{\mathbf{N}}^\beta$ with $\text{Bran}(\lambda_{\hat{\mathbf{N}}^\beta}) = \{\hat{\mathbf{N}}^\beta, \mathbf{N}^\beta\}$ for the former. While in this example, \mathbf{N} has only a single parent / child, this is not generally required this to be the case.

In (c), a composite representation of the branch graph of both circuits are shown: They are identical, except that after the node splitting we obtain an additional branch $\bar{\mathbf{N}}^\beta$ complementing each original branch \mathbf{N}^β . It has the same weak parents as \mathbf{N}^β , and hence, features the same green (and, analogously, red) arrows. Accordingly, this branch does not add any non-trivial information to the branch graph and does not affect the presence of weak loops. Note that when considering the associated supermap, whether we flesh out \mathbf{N} or the nodes $\hat{\mathbf{N}}$ with monopartite supermaps amounts to different supermaps: For (b), we obtain a single agent performing a (potentially routed) operation, while for (a) we obtain three agents performing a (potentially routed) operation, dependent on the specific fleshing out. By contrast, if we flesh out these nodes with isometries (i.e. with no slots), the resulting maps are identical.

Furthermore, one may wonder whether the idea of splitting a node according to its branch structure can be generalised to arbitrary routed graphs. Indeed, this seems possible in many cases, first replacing a given node \mathbf{N} with a copy $\hat{\mathbf{N}}^\beta$ for each of its branches \mathbf{N}^β , endowed with a copy of the respective arrows, and then adjusting their indices such that for each $\hat{\mathbf{N}}^\beta$, we have $\lambda_{\hat{\mathbf{N}}^\beta} = \lambda_{\mathbf{N}^\beta} \oplus \mathbb{1}_{\bar{\mathbf{N}}^\beta}$. Here, the identity acts on a 1-dimensional subspace corresponding to the case that \mathbf{N}^β does not happen. A simple example for this is illustrated in Fig. 5.3. However, in cases where the branch structure is dictated by correlating multiple incoming indices⁴³ (and similarly for outgoing indices), there seems to be no straightforward way to adjust the index structure accordingly, as this hinders an association of individual arrow index values to a given branch. The precise conditions under which the index structure can be adjusted like this and a generic procedure to do so remain important open questions, especially for characterising a minimal set of routed circuit decompositions to obtain all (routed) processes obtainable from routed quantum circuits.

5.3 Merging nodes in the routed graph

If we compare our construction of $\mathcal{G}_{\text{QC-QC}(N)}$ (as well as $\mathcal{G}_{\text{QC-QC}(N)}^\alpha$ and $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$) with the routed graphs described in [39] for the examples of the quantum switch (shown there in Fig. 6) and the Grenoble process (in Fig. 16), one immediately notices that the latter have fewer nodes than those that we provide for the same scenarios. Indeed, while we can identify our nodes \mathbf{V}_1 and \mathbf{V}_{N+1} with their nodes P and F , respectively, the other nodes in the routed graphs of [39] correspond more naturally to the agent nodes \mathbf{A}_k , and there is no clear equivalent for the remaining “internal” \mathbf{V} -nodes of our representation.

In this section, we will demonstrate that starting from the split-node generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$, we indeed recover their routed graphs and routed circuit decompositions as alternative

⁴³Consider e.g. a routed subgraph $\mathbf{A} \xrightarrow{i} \mathbf{X} \xleftarrow{j} \mathbf{B}$ with $i, j \in \{1, 2\}$ and no 1-dimensional index values, where $\text{Bran}(\lambda_{\mathbf{X}}) = \{\mathbf{X}^\gamma, \mathbf{X}^\delta\}$, $\text{Ind}_{\mathbf{X}^\gamma}^{\text{in}} = \{(1, 1); (2, 2)\}$ and $\text{Ind}_{\mathbf{X}^\delta}^{\text{in}} = \{(1, 2); (2, 1)\}$.

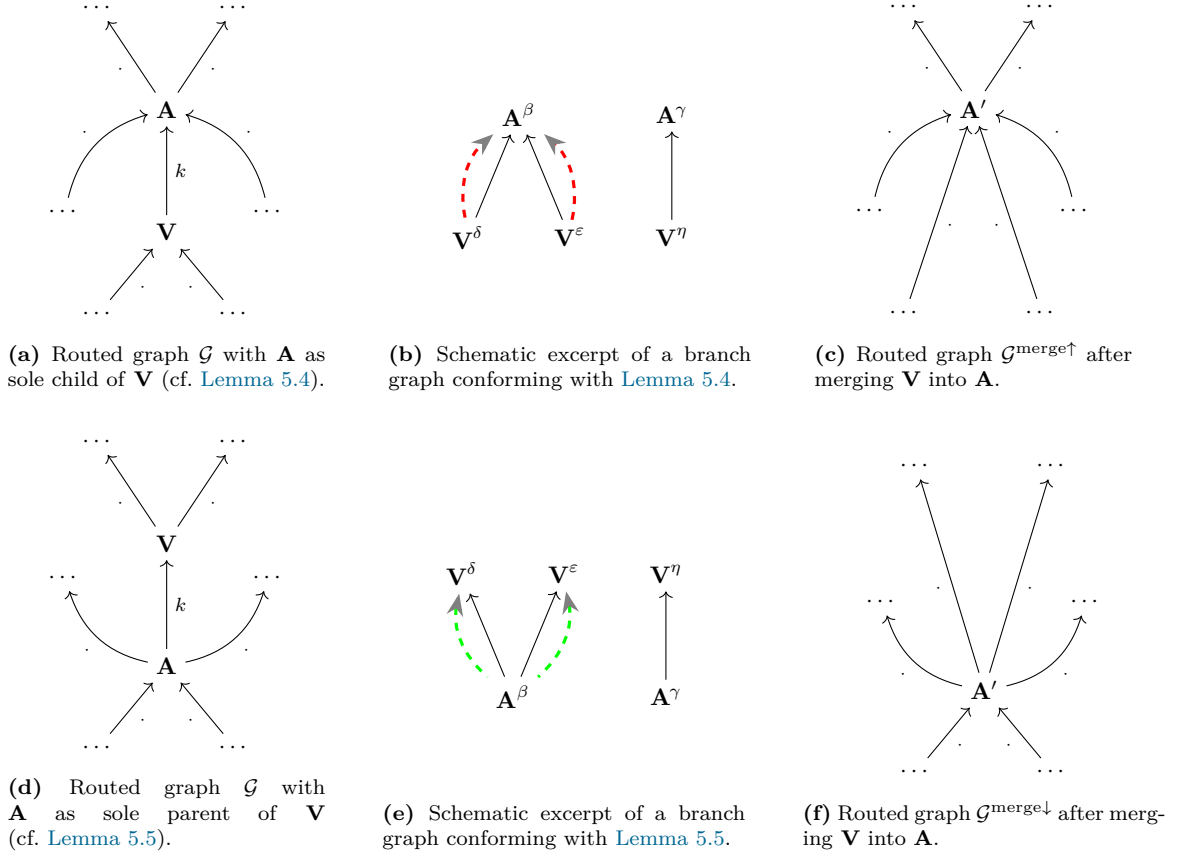


Figure 5.4: Routed and branch graph architectures which allow for node merging according to Proposition 5.6. For the branch graph excerpts, we only show the most relevant arrows for merging, which will no longer be featured in the branch graph obtained afterwards. In the associated branch graph, we observe that each branch of \mathbf{V} is associated with exactly one branch of \mathbf{A} . For details regarding the derivation of the branch graph, see the proof of Lemma 5.4 in Appendix C.

representations – albeit at the cost of more complex fleshing outs for the remaining nodes. Indeed, by merging internal nodes into agent nodes, one can effectively recast the isometries as pre- and post-processing for the respective nodes. However, we will find that a decomposition with no internal nodes other than P and F is generally not possible for generic QC-QCs when $N > 3$.

In order to recover the routed circuits of [39] from our approach, we will first introduce a slightly more general procedure, demonstrating that the idea of node merging is applicable to a wider variety of routed graphs, before applying it to the generic routed graph which forms the focus of our study.

Lemma 5.4. Let $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N} \in \text{Nodes}_{\Gamma}})$ be a valid routed graph without any weak loops, and $\mathbf{A}, \mathbf{V} \in \text{Nodes}_{\Gamma}$ such that \mathbf{A} is the sole child of \mathbf{V} (cf. Fig. 5.4a), and the associated branch structure is such that

$$\forall \mathbf{A}^{\beta} \in \text{Bran}(\lambda_{\mathbf{A}}) \quad \exists! \mathbf{V}^{\gamma} \in \text{Bran}(\lambda_{\mathbf{V}}) : \text{LinkVal}(\mathbf{V}^{\gamma}, \mathbf{A}^{\beta}) \neq \emptyset \quad (5.10)$$

Consider another routed graph $\mathcal{G}^{\text{merge}\uparrow}$ obtained by merging \mathbf{V} into \mathbf{A} , as follows:

- the two nodes being replaced by a single node \mathbf{A}' , with incoming arrows $\text{in}(\mathbf{A}') = \text{in}(\mathbf{V}) \cup (\text{in}(\mathbf{A}) \setminus \text{out}(\mathbf{V}))$ and outgoing arrows $\text{out}(\mathbf{A}') = \text{out}(\mathbf{A})$, as shown in Fig. 5.4c,
- the resulting node \mathbf{A}' being endowed with the route $\lambda_{\mathbf{A}'} = \lambda_{\mathbf{A}} * \lambda_{\mathbf{V}}$.

Then $\mathcal{G}^{\text{merge}\uparrow}$ is a valid routed graph.

The statement of this and the following lemma, as well as schematic excerpts of the associated branch graphs, are illustrated in Fig. 5.4.

Lemma 5.5. Let $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N} \in \text{Nodes}_{\Gamma}})$ be a valid routed graph without any weak loops, and $\mathbf{A}, \mathbf{V} \in \text{Nodes}_{\Gamma}$ such that \mathbf{A} is the sole parent of \mathbf{V} (cf. Fig. 5.4d) and the associated branch structure is such that

$$\forall \mathbf{A}^{\beta} \in \text{Bran}(\lambda_{\mathbf{A}}) \quad \exists! \mathbf{V}^{\gamma} \in \text{Bran}(\lambda_{\mathbf{V}}) : \quad \text{LinkVal}(\mathbf{A}^{\beta}, \mathbf{V}^{\gamma}) \neq \emptyset \quad (5.11)$$

Consider another routed graph $\mathcal{G}^{\text{merge}\downarrow}$ obtained by merging \mathbf{V} into \mathbf{A} , similarly to Lemma 5.4. Then $\mathcal{G}^{\text{merge}\downarrow}$ is a valid routed graph.

The proofs of these lemmas are given in Appendix C. While the naming of the nodes \mathbf{A} and \mathbf{V} are highly suggestive with regard to the generic routed graphs and its alternative versions, they are not intended as restrictions with regard to the scope of these two lemmas.

Note that the merged routed graphs $\mathcal{G}^{\text{merge}\downarrow}$ and $\mathcal{G}^{\text{merge}\uparrow}$ are essentially created by merging the branches \mathbf{A}^{β} and \mathbf{V}^{γ} . In the branch graph, the respective branches are merged along their respective strong parent relations. The branches of the merged node feature a subset of the union of the weak parent/child relations of the original nodes, with weak parent relations “within” merged branches vanishing. Additionally, due to the potentially reduced number of branches and the coarse-graining of the image of the choice function, generally some green arrows *to* \mathbf{V}, \mathbf{A} as well as some red arrows *from* these nodes may vanish from the routed graph under this coarse-graining.⁴⁴

The resulting routed graphs are not only valid, but indeed generate, in a precise sense, the same class of processes by means of routed circuit representations.

Proposition 5.6. Any process with a routed circuit decomposition (as defined in Definition 2.26) according to the routed graph \mathcal{G} , where \mathbf{V} is fleshed out with a routed isometry (an “internal” node), can also be expressed with a routed circuit decomposition according to $\mathcal{G}^{\text{merge}\uparrow} / \mathcal{G}^{\text{merge}\downarrow}$, related to \mathcal{G} via Lemma 5.4 / Lemma 5.5, and vice versa.

The proof of this lemma is given in Appendix C. Essentially, the merge composes the channels and supermaps fleshing out the skeletal supermap associated with \mathcal{G} using the link product.

For $\mathcal{G}^{\text{merge}\downarrow}$, we would like to point out that the dimension of $\mathcal{H}^{\mathbf{A} \rightarrow \mathbf{V}}$ could generally restrict the information flow from $\mathcal{H}^{\text{in}(\mathbf{A})}$ to $\mathcal{H}^{\text{out}(\mathbf{V})}$. Therefore, the equivalence of both representations does not necessarily hold on the level of the individual skeletal supermaps associated with \mathcal{G} and $\mathcal{G}^{\text{merge}\downarrow}$, respectively. However, as the same restriction may alternatively be introduced on the level of the fleshing out, this fact does not impair the fact that both routed graphs indeed generate the same set of routed circuit decompositions.

In practice, when merging an internal node \mathbf{V} into an agent node \mathbf{A} , we will denote this as $\mathbf{V} \hookrightarrow \mathbf{A}$, merging the internal node “into” the agent node, which will then be denoted as \mathbf{A}' .

We now continue by applying these results to our generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$. For $N = 2$ and $N = 3$, we see that a rewriting in this fashion allows to get rid of all internal nodes in $\mathcal{G}_{\text{QC-QC}(N)}$, except for \mathbf{V}_1 and \mathbf{V}_{N+1} .

Considering $\mathcal{G}_{\text{QC-QC}(3)}^{\text{split}}$ as an example, we observe that both $\hat{\mathbf{V}}_2^{k_1}$ and $\hat{\mathbf{V}}_3^{k_2}$ have agent nodes as unique parent/child nodes. Therefore, we may absorb $\hat{\mathbf{V}}_2^{k_1} \hookrightarrow \mathbf{A}_{k_1}$ and $\hat{\mathbf{V}}_3^{k_2} \hookrightarrow \mathbf{A}_{\mathcal{N} \setminus \mathcal{K}_2}$. The resulting routed graph is illustrated in Fig. 5.1d (denoting again the agents $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ as $(\mathbf{A}, \mathbf{B}, \mathbf{C})$).⁴⁵ For $N = 2$, analogous arguments apply.

Indeed, this allows us to ultimately recover the routed graph examples from [39].⁴⁶

⁴⁴However, for the generic routed graph as our main case of interest, this will not be the case.

⁴⁵To see explicitly how the composition of branches plays out, consider the case of $\hat{\mathbf{V}}_2^B \hookrightarrow \mathbf{B}$ in detail. Here, $\text{Bran}(\lambda_{\hat{\mathbf{V}}_2^B}) = \{\mathbf{V}_2^B, \hat{\mathbf{V}}_2^B\}$ and $\text{Bran}(\lambda_{\mathbf{B}}) = \{\mathbf{B}^{\mathcal{K}_n} \mid \mathcal{K}_n \subset \mathcal{N} \setminus \mathbf{B}\}$. Then, \mathbf{V}_2^B and \mathbf{B}^{\emptyset} are composed, while the 1-dimensional branch is trivially composed with all other branches of \mathbf{B} . This leaves their respective operations unchanged, as the operation for $\hat{\mathbf{V}}_2^B$, specified by Eq. (5.8), acts identically (and on one-dimensional tensor factors, cf. Eq. (4.21)).

⁴⁶By contrast, introducing additional arrows to represent ancillary systems, as described in Section 5.1, introduces

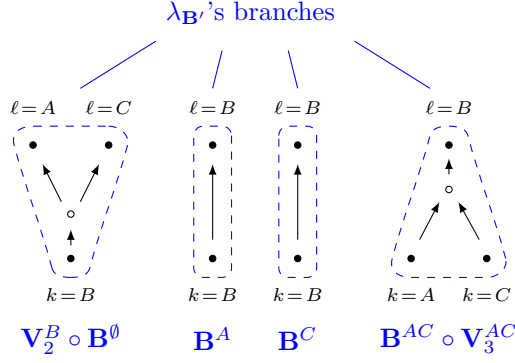


Figure 5.5: The branch structure for \mathbf{B}' in Fig. 5.1b, given by $\lambda_{\mathbf{B}'} = \lambda_{\hat{\mathbf{V}}_3^{AC}} * \lambda_{\mathbf{B}} * \lambda_{\hat{\mathbf{V}}_2^B}$ with a fine-graining into the branches of the original routed graph being shown in Fig. 4.5. The branch structures for \mathbf{A}' and \mathbf{C}' are given analogously by permutation. Due to the node merging, non-trivial index relations and bifurcation choices are carried over to the agent node \mathbf{B}' , reproducing the branch structure given for the nodes of the Grenoble process in Fig. 23 of [39].

Example 5.7 (Quantum switch). To represent the quantum switch without any internal node to represent V_2 explicitly, we take Fig. 5.1a as a starting point. Due to the simplicity of the routed graph, we have some freedom for the rewrite: Both for $\hat{\mathbf{V}}_2^A$ and $\hat{\mathbf{V}}_2^B$, we may choose to absorb them either into \mathbf{A} or \mathbf{B} , i.e., both Lemma 5.4 and Lemma 5.5 could be applied we to recover the quantum switch as presented in Fig. 6 of [39]. For Fig. 5.1c and this example, we choose $\mathbf{V}_2^A \hookrightarrow \mathbf{A}$ and $\mathbf{V}_2^B \hookrightarrow \mathbf{B}$.⁴⁷

Regarding the fleshing out, we consider the resulting node \mathbf{B}' as an example. Here, we take the composition of the fleshing outs $|\mathbf{B}\rangle\rangle$ and $|\hat{\mathbf{V}}_2^B\rangle\rangle$ from $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ (according to the link product) along the node merging as the new fleshing out:

$$\begin{aligned} |\mathbf{B}'\rangle\rangle &= |\mathbf{B}\rangle\rangle * |\hat{\mathbf{V}}_2^B\rangle\rangle \\ &= |J_{\mathbf{B}}^{\text{in}}\rangle\rangle * |J_{\mathbf{B}}^{\text{out}}\rangle\rangle * |J_{\mathbf{V}_2^B}^{\text{in}}\rangle\rangle * |\hat{V}_2^B\rangle\rangle * |J_{\mathbf{V}_2^B}^{\text{out}}\rangle\rangle. \end{aligned} \quad (5.12)$$

and similarly for $|\mathbf{A}\rangle\rangle$. This reconstructs the routed graph of Fig. 6 of [39] when relabelling their indices accordingly.⁴⁸

Example 5.8 (Grenoble process). The Grenoble process is a tripartite process with dynamical causal order, and was originally introduced in [34]. We use the transformation techniques introduced in this section to reconstruct the routed quantum circuit of the Grenoble process given in [39]. To reconstruct their routed graph, we take Fig. 5.1b as a starting point. We merge $\hat{\mathbf{V}}_2^A \hookrightarrow \mathbf{A} \leftarrow \hat{\mathbf{V}}_3^{BC}$, $\hat{\mathbf{V}}_2^B \hookrightarrow \mathbf{B} \leftarrow \hat{\mathbf{V}}_3^{AC}$, $\hat{\mathbf{V}}_2^C \hookrightarrow \mathbf{C} \leftarrow \hat{\mathbf{V}}_3^{AB}$, obtaining Fig. 5.1d. In this graph, the arrows connecting the agents feature multiple indices, as several node mergings induce an arrow between those.

Renaming \mathbf{V}_1 to P and \mathbf{V}_4 to F , we obtain the same skeletal supermap as in [39] (up to a renaming of the index values and Hilbert spaces). For example, for the merged node \mathbf{B}' we obtain the branch structure shown in Fig. 5.5, reflecting Fig. 23 of [39].

Regarding the fleshing out, we again consider the node \mathbf{B}' as an example. We take the composition of the fleshing outs $|\mathbf{B}\rangle\rangle$, $|\hat{\mathbf{V}}_2^B\rangle\rangle$ and $|\hat{\mathbf{V}}_3^{AC}\rangle\rangle$ along the node merging as the new fleshing

additional parents for the \mathbf{V} -nodes, which prevents us from applying the same technique for simplification: each node $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ features additional parents $\mathbf{V}_n^{\mathcal{K}_n \setminus k}$.

⁴⁷This results in $\mathbf{A}' \xrightarrow{X_A^B} \mathbf{B}'$, with the alternative resulting in $\mathbf{A}' \xrightarrow{X_0^A} \mathbf{B}'$, and equally when exchanging A / B .

⁴⁸Specifically, one translates their index values as follows:

$$i, l = 1 \rightarrow \mathcal{X}_1^A = \{A\} \quad j, k = 1 \rightarrow \mathcal{X}_1^B = \{B\} \quad m = 1 \rightarrow \mathcal{X}_2^A = \{A, B\} \quad n = 1 \rightarrow \mathcal{X}_2^B = \{A, B\}. \quad (5.13)$$

For all of their indices, if they take the value 0, we set the associated index $\mathcal{X}_n^k = \emptyset$.

out:

$$\begin{aligned}
|\mathbf{B}'\rangle\rangle &= |\hat{\mathbf{V}}_3^{AC}\rangle\rangle * |\mathbf{B}\rangle\rangle * |\hat{\mathbf{V}}_2^B\rangle\rangle \\
&= |J_{\mathbf{V}_3^{AC}}^{\text{in}}\rangle\rangle * |\hat{\mathbf{V}}_3^{AC}\rangle\rangle * |J_{\mathbf{V}_3^{AC}}^{\text{out}}\rangle\rangle * |J_{\mathbf{B}}^{\text{in}}\rangle\rangle * |J_{\mathbf{B}}^{\text{out}}\rangle\rangle * |J_{\mathbf{V}_2^B}^{\text{in}}\rangle\rangle * |\hat{\mathbf{V}}_2^B\rangle\rangle * |J_{\mathbf{V}_2^B}^{\text{out}}\rangle\rangle. \tag{5.14}
\end{aligned}$$

How this fleshing out relates to the fleshing out specified in Fig. 25 of [39] is described in detail in Appendix D.

For larger N , the possibility of merging nodes into agent nodes in $\mathcal{G}_{\text{QC-QC}(N)}$ applies analogously for the internal nodes $\mathbf{V}_2^{\mathcal{K}_1}$ and $\mathbf{V}_N^{\mathcal{K}_{N-1}}$. For the remaining internal nodes, however, the technique is no longer applicable, as the remaining \mathbf{V} -nodes in $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ will feature multiple parents and children respectively. However, a specific fleshing out of this routed graph (according to a specific QC-QC implementation) may potentially be compatible with a more fine-grained branch structure. Imposing such a branch structure can effectively restrict the set of processes supported by the corresponding routed graph. In Appendix E, we investigate this fact in more detail and reproduce an example of a 4-partite process [41, 52] which cannot be entirely fine-grained in such a fashion.

Note that we do not reconstruct here the representation of the quantum 3-switch given in Section 4.1 of [39]. Doing so would require further adjustments to the index constraints on the routed graph, restricting the set of supported processes even further. We leave this for future work.

5.4 Removing arrows in the routed graph

In addition to considering alternatives to the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$ to obtain arbitrary QC-QCs, we may consider ways to simplify the routed graph when the QC-QC has additional structure yielding simplified connectivity.

Consider a given fleshing out of a skeletal supermap for the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$. If for a given arrow $\mathbf{M} \rightarrow \mathbf{N}$, the only sector that can be populated (i.e. whose index value can be selected) is $\mathcal{H}_{\mathbb{Q}}^{\mathbf{M} \rightarrow \mathbf{N}}$ (cf. Section 4.3.1), we may remove the respective arrow from the routed graph – if for the adjoint of the graph, the respective space is also never populated. This follows as in this case, the respective 1-dimensional tensor factor in the formal spaces $\mathcal{H}^{\text{out}(\mathbf{M})}$ and $\mathcal{H}^{\text{in}(\mathbf{N})}$ may be omitted (up to an isomorphism), with the index associated to the arrow being constantly \mathbb{Q} for all remaining bifurcation choices $\vec{\ell}$ (see Section 4.1.4) – both for the original and the adjoint graph. The branch graph and the skeletal supermap simplify accordingly: in comparison to the original branch graph, the strong parent relations associated with the respective arrows (i.e., branches \mathbf{M}^γ and \mathbf{N}^β) disappear, as they require an arrow from $\mathbf{M} \rightarrow \mathbf{N}$ in the first place. Furthermore, due to the bifurcation choice for the branch happening being removed, it never happens⁴⁹ and all green arrows may be removed as well. To finally remove the respective red arrows as well, the same argument is applied to the adjoint graph. We can then remove all branches no longer associated with any arrows.

For the remaining weak arrows in the graph, one needs to check whether the corresponding functional dependence in the choice function remains. For any routed graphs (even beyond the generic routed graph), each node will no longer have weak parents if only a single branch of it remains, or if in the resulting graph a given branch \mathbf{M}^γ is a weak parent of just one branch of a given node \mathbf{N} : a non-trivial choice always requires multiple possibilities (for the branch which happens).

Specifically for a given implementation of a QC-QC (and therewith, a choice of $\{V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell}\}$ yielding the respective process according to Eq. (3.7)), arrows may be removed in particular if for some ℓ, n , the condition $V_{\mathcal{K}_n \setminus k, k}^{\rightarrow \ell} = 0$ is satisfied for all \mathcal{K}_n, k . Then in the image of the isometry given by $|\mathbf{V}_{n+1}\rangle\rangle$, only the sector $\mathcal{H}_{\mathbb{Q}}^{\mathbf{V}_{n+1} \rightarrow \mathbf{A}_\ell}$ will be populated, and $\mathcal{X}_{n+1}^\ell = \mathbb{Q}$ for any choice of operations A_k and past state $|\psi\rangle^P$. By extension, the sector $\mathcal{H}_{\mathbb{Q}}^{\mathbf{A}_\ell \rightarrow \mathbf{V}_{n+2}}$ is fixed as well. Hence, both the arrows $\mathbf{V}_{n+1} \xrightarrow{\mathcal{X}_{n+1}^\ell} \mathbf{A}_\ell$ and $\mathbf{A}_\ell \xrightarrow{\mathcal{X}_{n+1}^\ell} \mathbf{V}_{n+2}$ may be omitted from the routed graph, and

⁴⁹And therefore, all potential further bifurcation choices made within this branch never matter.

we may modify \tilde{V}_{n+2} such that for all $m \notin \mathcal{K}_n \cup \ell$ and all $\mathcal{K}_n \not\supseteq \ell$, $V_{\mathcal{K}_n, \ell}^{\rightarrow m} = 0$ holds – which then ensures the analogous property for the adjoint of the graph. For the associated branch graph, incoming green dashed arrows will generally always vanish if the respective branch is now certain either to happen or not to happen, and analogously for the red dashed arrows. In such a QC-QC, there is always an implementation in which A_ℓ never acts in the $(n + 1)$ th position, which allows to remove the corresponding arrows.

Example 5.9. Consider a bipartite process compatible with a fixed order $A \prec B$. Then clearly, there exist implementations of this process where $V_{\emptyset, \emptyset}^{\rightarrow B} = V_{\emptyset, B}^{\rightarrow A} = 0$. Hence, for any choice of agent operations, $\mathcal{X}_1^B = \mathcal{X}_2^A = \emptyset$, and there exists an implementation in which $V_{B, A}^{\rightarrow F} = 0$ as well. Accordingly, we may remove the arrows

$$\mathbf{V}_1 \xrightarrow{\mathcal{X}_0^B} \mathbf{B}, \quad \mathbf{B} \xrightarrow{\mathcal{X}_0^B} \mathbf{V}_2, \quad \mathbf{V}_2 \xrightarrow{\mathcal{X}_B^A} \mathbf{A} \quad \text{and} \quad \mathbf{A} \xrightarrow{\mathcal{X}_B^A} \mathbf{V}_3, \quad (5.15)$$

from the graph in Fig. 4.2a, leaving us with the acyclic routed graph

$$\longrightarrow \mathbf{V}_1 \xrightarrow{\mathcal{X}_0^A} \mathbf{A} \xrightarrow{\mathcal{X}_0^A} \mathbf{V}_2 \xrightarrow{\mathcal{X}_A^B} \mathbf{B} \xrightarrow{\mathcal{X}_A^B} \mathbf{V}_3 \longrightarrow . \quad (5.16)$$

Here, the indices always take the values $\mathcal{X}_1^A = \{A\}$, $\mathcal{X}_2^B = \{A, B\}$, and could be omitted from the graph. The resulting branch graph is then almost identical to the routed graph, as for each node, there is only a single branch, which always happens.

$$\mathbf{V}_1^{\emptyset} \xrightarrow{\mathcal{X}_0^A} \mathbf{A}^{\emptyset} \xrightarrow{\mathcal{X}_0^A} \mathbf{V}_2^A \xrightarrow{\mathcal{X}_A^B} \mathbf{B}^A \xrightarrow{\mathcal{X}_A^B} \mathbf{V}_3^{AB}. \quad (5.17)$$

6 Discussion

In providing a representation of arbitrary QC-QCs in terms of routed quantum circuits, we have established an application of the latter framework well beyond the particular examples previously considered, extending it to a wide class of processes. Our construction notably provides a more fine-grained picture of the logical structure of QC-QCs, with the control of the causal order being represented by indices \mathcal{X}_n^k in the routed graph.

The fact that we achieved this by specifying just a single template for the routed graph and a generic fleshing out is testament to the ability of the framework to condense entire classes of processes in terms of their underlying information-processing structure, which is concisely encapsulated in the routed graph. In addition, the more fine-grained branch graph it brings forth may be closely related to the underlying causal structure of the respective process, as has been tentatively demonstrated in [51].

6.1 Contrasting routed circuit decompositions

In Section 4 of [39], explicit constructions were given for two instances of QC-QCs: the quantum switch [21] and the Grenoble process [34]. These constructions differ from our generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, with fewer nodes: internal nodes corresponding to the agents A_k , as well as nodes for the past and future P and F , which play roles analogous to our internal nodes \mathbf{V}_1 and \mathbf{V}_{N+1} . By featuring additional internal nodes \mathbf{V}_n , our construction appears to yield a significantly more verbose routed graph representation. In Section 5.3, we discussed how one can, under certain conditions, nonetheless recover the structure of their routed graph from ours by merging nodes. However, for $N > 3$, a representation without further internal (i.e., non-agent) nodes can no longer generally be found, as we exemplify in Appendix E. The generalisation of the routed graphs given in [39], without additional internal nodes, is thus unable to capture the full expressivity of QC-QCs.

Even in the $N = 2$ and $N = 3$ cases, the more compact routed graph representation of [39] comes at the cost of a more complex fleshing out for the agent nodes \mathbf{A}_k , which must not only act as an adapter for allowing the local operations of the agents to be plugged in, but also includes the, in general non-trivial, (internal) routing of the process.

However, with our generic routed graph (modified to represent the ancillary systems explicitly in $\mathcal{G}_{\text{QC-QC}(N)}^\alpha$), we can mostly forgo the necessity of a fleshing out to describe the class of QC-QCs.

Indeed, for the \mathbf{V} -nodes the routes actually restrict the associated slots precisely to valid QC-QC isometries, and any routed isometries following the routes constitutes a valid fleshing out. For the \mathbf{A} -nodes, as observed in [Section 5.1](#), allowing the agents to perform any routed operation⁵⁰ following the routes essentially gives them just the power to perform routed operations dependent on the causal order – which in fact mirrors many physical implementations suggested so far more closely, while still conveying a form of ICO [[51](#), [53](#)]. One can moreover show that each such routed process may still be simulated as a conventional (non-routed) QC-QC, potentially decreasing the relevance of this distinction.

Indeed, consider a process where a coherent copy of the control system $\bar{C}^{(k)}$ is passed to each agent A_k . Then, they may perform their operation controlled on the set of agents $\mathcal{K}_n \setminus k$ who acted previously, as in a routed process, or perform a measurement on the causal order, destroying the coherence of the quantum control in the QC-QC. Any N -slot QC-QC process may be amended to pass a copy of the control to the agents in such a fashion, increasing the dimensions of $\mathcal{H}_k^{A/O}$ accordingly. After their operation, the coherent copy may then be undone – presuming, the agents did not modify the copy – and be traced out without harming the coherence of the control. If the agents modify their copy of the control however, they are transcending the set of operations which would be allowed in a routed switch, and may collapse the action of the process to a QC-CC (with *classical* control of causal order).

Finally, we briefly comment on the index structure we utilised. While in [[39](#)], all indices used in practice only take values $\{0, 1\}$, we use the index value to encode subsets $\mathcal{K}_n \subseteq \mathcal{N}$ (or \otimes) – with the notable exception of [Eq. \(5.5\)](#). Thereby, we demonstrate how a more complex set of possible index values may be helpful to obtain an adequate routed graph.

6.2 Characterising classes of processes

Several previous works have proposed subclasses of QC-QCs with specific properties of interest. These include a restriction to classical control (QC-CC) [[34](#)], to superpositions of fixed orders (Sup [[54](#), [55](#)], or the more general QC-supFO class [[56](#)]), and to non-influenceable causal orders (QC-NIQC), capturing a weak kind of dynamicality of causal order [[56](#)].

Typically, the restrictions on these subclasses can be characterised as restrictions on the internal operations of a QC-QC. Our routed quantum circuit construction provides a new perspective from which to study these classes, as these restrictions may correspond to additional constraints on the routed graph. For instance, modifying the index structure to capture the full order (k_1, \dots, k_N) from \mathbf{V}_1 all the way to \mathbf{V}_{N+1} should yield a superposition of fixed orders. Generally, constraining the routed graph will involve introducing additional structure into the indices, effectively fine-graining the sectorisation of the systems to store additional information on the causal order, and imposing restrictions on the branch structure of the internal nodes \mathbf{V}_{n+1} . The resulting graphs can then potentially be simplified following techniques analogous to those presented in [Section 5](#).

Conversely, imposing constraints directly on the routed graph may be an interesting avenue to study new types of QC-QCs, a potential example being the subclass of processes which can be represented without involving internal transformations beyond local pre- and post-processing in-between the agents (cf. [Appendix E](#)). When assuming the operations associated with each branch of the respective nodes to be localised in spacetime, this property in particular affects the embeddability of the associated circuit into spacetime [[11](#), [57](#)].

In addition to capturing these classes in terms of routed quantum circuits, the properties of these representations may also serve as starting points to generalise the concepts they capture to non-QC-QC processes. For instance, in the language of routed quantum circuits, green dashed arrows in the branch graph may indicate whether a node has an influence on the order of later nodes: beyond QC-QCs, this has been showcased for both the Lugano process [[39](#)], and more recently, for a process with “unilateral determination of causal order” [[48](#)]. Studying this relation in detail would potentially allow the notion of dynamicality, which has so far been formalised for QC-QCs only [[56](#)], to be studied in processes beyond QC-QCs.

⁵⁰With the skeletal supermap associated with the routed graph from [Section 5.1](#), that is, to prevent the agents from accessing the ancillary systems α_n .

6.3 Consequences for the routed quantum circuit framework

Last but not least, we consider the impact of our results on the general status of the routed quantum circuit framework. By extending the number of processes with an established routed quantum circuit decomposition from a small handful to the entire class of QC-QCs, we confirm the applicability of the framework to a wide class of physically implementable processes. Moreover, recent results connecting QC-QCs with the causal box framework [33, 41, 58] indicate that QC-QCs appear to be the most general class of processes compatible (and implementable) within a fixed (acyclic) spacetime. Combined with our results, this would imply that skeletal supermaps associated with routed graphs (without weak loops) – or even the subset of routed graphs from our construction – are capable of expressing any quantum process implementable in a fixed spacetime.⁵¹ This would provide some additional support the first main conjecture of [39], concerning the hypothesis that any unitary process may be obtained from fleshing out some skeletal supermap associated with a routed graph. The results of [52] further indicate that these statements may also be generalisable from fixed acyclic spacetimes to any settings where all agents (independently) record a monotonically increasing time, e.g. capturing gravitational quantum switch scenarios. Further progress in this direction, extending the result beyond processes implementable within a fixed spacetime, would most likely require significant progress in the study of causal decompositions of unitary channels [59, 60].

Considering the fact that QC-QCs do not violate causal inequalities [34], our results are in agreement with the second major conjecture of [39] – that a skeletal supermap violates causal inequalities if and only if its branch graph features weak loops. However, as our procedure only yields a small subset of valid routed graphs without weak loops, we are unable to provide any stronger statement in that regard.

The joint perspective obtained by studying QC-QCs and routed quantum circuits may also be valuable to provide further insight on the composability of processes. Indeed, although the validity conditions for routed graphs provide a condition to affirm the validity of compositions, a systematic study of the resulting family of routed graphs has been lacking so far. Restricting to compositions of routed graphs which represent faithful reconstructions of QC-QCs may help both to render the scope of this study more manageable and to restrict oneself to physically admissible implementations [40, 61–64]. Conversely, the clear graphical representation of connectivity may also allow one to restrict the class of processes attainable by composing a restricted set of “elementary” processes.

7 Acknowledgements

We thank Hlér Kristjánsson, Ved Kunte, Raphaël Mothe, Pierre Pocreau, Matthias Salzger, V. Vilasini and Emilien de Bank for helpful discussions.

This research was funded in part by l’Agence Nationale de la Recherche (ANR) projects ANR-15-IDEX-02 and ANR-22-CE47-0012, the PEPR integrated project EPiQ ANR-22-PETQ-0007 as part of Plan France 2030, and the STEP2 grant ANR-22-EXES-0013. For the purpose of open access, the authors have applied a CC-BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

References

- [1] Hans Reichenbach. “The direction of time”. [University of California Press](#). Berkeley (1956).
- [2] Peter Spirtes, Clark Glymour, and Richard Scheines. “Causation, prediction, and search”. [Springer New York](#). NY (1993).
- [3] Judea Pearl. “Causality: Models, reasoning and inference”. [Cambridge University Press](#). (2009). 2nd edition.

⁵¹However, this alone does not imply the ability to interpret each (valid) skeletal supermap as a blueprint for a non-routed process!

- [4] Christopher J. Wood and Robert W. Spekkens. “The lesson of causal discovery algorithms for quantum correlations: Causal explanations of Bell-inequality violations require fine-tuning”. *New J. Phys.* **17**, 033002 (2015).
- [5] John-Mark A. Allen, Jonathan Barrett, Dominic C. Horsman, Ciarán M. Lee, and Robert W. Spekkens. “Quantum common causes and quantum causal models”. *Phys. Rev. X* **7**, 031021 (2017).
- [6] Jonathan Barrett, Robin Lorenz, and Ognjan Oreshkov. “Quantum causal models” (2019). [arXiv:1906.10726](https://arxiv.org/abs/1906.10726).
- [7] Lucien Hardy. “Quantum gravity computers: On the theory of computation with indefinite causal structure”. Pages 379–401. Springer Netherlands. (2009). [arXiv:quant-ph/0701019](https://arxiv.org/abs/quant-ph/0701019).
- [8] Časlav Brukner. “Bounding quantum correlations with indefinite causal order”. *New J. Phys.* **17**, 083034 (2015).
- [9] Fabio Costa and Sally Shrapnel. “Quantum causal modelling”. *New J. Phys.* **18**, 063032 (2016).
- [10] Jonathan Barrett, Robin Lorenz, and Ognjan Oreshkov. “Cyclic quantum causal models”. *Nature Commun.* **12** (2021).
- [11] V. Vilasini and Renato Renner. “Embedding cyclic information-theoretic structures in acyclic space-times: No-go results for indefinite causality”. *Phys. Rev. A* **110**, 022227 (2024). [arXiv:2203.11245](https://arxiv.org/abs/2203.11245).
- [12] Emily Adlam. “Is there causation in fundamental physics? New insights from process matrices and quantum causal modelling”. *Synthese* **201**, 152 (2023).
- [13] V. Vilasini and Renato Renner. “Fundamental limits for realizing quantum processes in space-time”. *Phys. Rev. Lett.* **133**, 080201 (2024). [arXiv:2408.13387](https://arxiv.org/abs/2408.13387).
- [14] Giulio Chiribella, Giacomo Mauro D’Ariano, and Paulo Perinotti. “Transforming quantum operations: Quantum supermaps”. *Europhys. Lett.* **83**, 30004 (2008). [arXiv:0804.0180](https://arxiv.org/abs/0804.0180).
- [15] Ognjan Oreshkov, Fabio Costa, and Časlav Brukner. “Quantum correlations with no causal order”. *Nature Commun.* **3**, 1092 (2012). [arXiv:1105.4464](https://arxiv.org/abs/1105.4464).
- [16] Magdalena Zych, Fabio Costa, Igor Pikovski, and Časlav Brukner. “Bell’s theorem for temporal order”. *Nature Commun.* **10**, 3772 (2019).
- [17] Natália S. Móller, Bruna Sahdo, and Nelson Yokomizo. “Quantum switch in the gravity of earth”. *Phys. Rev. A* **104**, 042414 (2021). [arXiv:2012.03989](https://arxiv.org/abs/2012.03989).
- [18] Veronika Baumann, Marius Krumm, Philippe Allard Guérin, and Časlav Brukner. “Noncausal Page-Wootters circuits”. *Phys. Rev. Res.* **4**, 013180 (2022).
- [19] Anne-Catherine de la Hamette, Viktoria Kabel, Marios Christodoulou, and Časlav Brukner. “Quantum diffeomorphisms cannot make indefinite causal order definite” (2024). [arXiv:2211.15685](https://arxiv.org/abs/2211.15685).
- [20] Bruna Sahdo. “Indefinite order in the interface of quantum mechanics and gravity”. Master’s thesis. Federal University of Minas Gerais. (2023). [arXiv:2310.02290](https://arxiv.org/abs/2310.02290).
- [21] Giulio Chiribella, Giacomo Mauro D’Ariano, Paolo Perinotti, and Benoit Valiron. “Quantum computations without definite causal structure”. *Phys. Rev. A* **88**, 022318 (2013). [arXiv:0912.0195](https://arxiv.org/abs/0912.0195).
- [22] Mateus Araújo, Adrien Feix, Miguel Navascués, and Časlav Brukner. “A purification postulate for quantum mechanics with indefinite causal order”. *Quantum* **1**, 10 (2017).
- [23] Āmin Baumeler, Adrien Feix, and Stefan Wolf. “Maximal incompatibility of locally classical behavior and global causal order in multiparty scenarios”. *Phys. Rev. A* **90**, 042106 (2014). [arXiv:1403.7333](https://arxiv.org/abs/1403.7333).
- [24] Āmin Baumeler and Stefan Wolf. “The space of logically consistent classical processes without causal order”. *New J. Phys.* **18**, 013036 (2016).
- [25] Giulio Chiribella. “Perfect discrimination of no-signalling channels via quantum superposition of causal structures”. *Phys. Rev. A* **86**, 040301 (2012). [arXiv:1109.5154](https://arxiv.org/abs/1109.5154).
- [26] Adrien Feix, Mateus Araújo, and Časlav Brukner. “Quantum superposition of the order of parties as a communication resource”. *Phys. Rev. A* **92**, 052326 (2015). [arXiv:1508.07840](https://arxiv.org/abs/1508.07840).
- [27] Philippe Allard Guérin, Adrien Feix, Mateus Araújo, and Časlav Brukner. “Exponential communication complexity advantage from quantum superposition of the direction of communication”. *Phys. Rev. Lett.* **117**, 100502 (2016). [arXiv:1605.07372](https://arxiv.org/abs/1605.07372).

- [28] Timoteo Colnaghi, Giacomo Mauro D’Ariano, Stefano Facchini, and Paolo Perinotti. “Quantum computation with programmable connections between gates”. *Phys. Lett. A* **376**, 2940–2943 (2012). [arXiv:1109.5987](#).
- [29] Stefano Facchini and Simon Perdrix. “Quantum circuits for the unitary permutation problem”. In *Theory and Applications of Models of Computation*. Pages 324–331. Springer International Publishing (2015). [arXiv:1405.5205](#).
- [30] Mateus Araújo, Fabio Costa, and Časlav Brukner. “Computational advantage from quantum-controlled ordering of gates”. *Phys. Rev. Lett.* **113**, 250402 (2014). [arXiv:1401.8127](#).
- [31] Márcio M. Taddei, Jaime Cariñe, Daniel Martínez, Tania García, Nayda Guerrero, Alastair A. Abbott, Mateus Araújo, Cyril Branciard, Esteban S. Gómez, Stephen P. Walborn, Leandro Aolita, and Gustavo Lima. “Computational advantage from the quantum superposition of multiple temporal orders of photonic gates”. *PRX Quantum* **2**, 010320 (2021).
- [32] Alastair A. Abbott, Mehdi Mhalla, and Pierre Pocreau. “Classical and quantum query complexity of Boolean functions under indefinite causal order”. In *Proc. 22nd Int. Conf. Quantum Phys. Log. (QPL 2025)*. to appear in *EPTCS* (2025). [arXiv:2506.05187](#).
- [33] Christopher Portmann, Christian Matt, Ueli Maurer, Renato Renner, and Björn Tackmann. “Causal boxes: Quantum information-processing systems closed under composition”. *IEEE Trans. Inf. Theory* **63**, 3277–3305 (2017). [arXiv:1512.02240](#).
- [34] Julian Wechs, Hippolyte Dourdent, Alastair A. Abbott, and Cyril Branciard. “Quantum circuits with classical versus quantum control of causal order”. *PRX Quantum* **2**, 030335 (2021).
- [35] Alexandre Clément and Simon Perdrix. “PBS-calculus: A graphical language for coherent control of quantum computations”. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). [arXiv:2002.09387](#).
- [36] Cyril Branciard, Alexandre Clément, Mehdi Mhalla, and Simon Perdrix. “Coherent control and distinguishability of quantum channels via PBS-diagrams”. In *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*. *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). [arXiv:2103.02073](#).
- [37] Tom Purves and Anthony J. Short. “Quantum theory cannot violate a causal inequality”. *Phys. Rev. Lett.* **127**, 110402 (2021). [arXiv:2101.09107](#).
- [38] Pablo Arrighi, Christopher Cedzich, Marin Costes, Ulysse Rémond, and Benoît Valiron. “Addressable quantum gates”. *ACM Trans. Quantum Comput.* (2023). [arXiv:2109.08050](#).
- [39] Augustin Vanrietvelde, Nick Ormrod, Hlér Kristjánsson, and Jonathan Barrett. “Consistent circuits for indefinite causal order” (2023). [arXiv:2206.10042](#).
- [40] Kostia Chardonnet, Marc de Visme, Benoît Valiron, and Renaud Vilmart. “The Many-Worlds Calculus”. *Log. Methods Comput. Sci* **21**, 2 (2025).
- [41] Matthias Salzger. “Connecting indefinite causal order processes to composable quantum protocols in a spacetime”. *Master’s thesis*. ETH Zurich. (2022). [arXiv:2304.06735](#).
- [42] Matthias Salzger and V Vilasini. “Mapping indefinite causal order processes to composable quantum protocols in a spacetime”. *New J. Phys.* **27**, 023002 (2025).
- [43] Augustin Vanrietvelde, Hlér Kristjánsson, and Jonathan Barrett. “Routed quantum circuits”. *Quantum* **5**, 503 (2021).
- [44] Giulio Chiribella and Hlér Kristjánsson. “Quantum Shannon theory with superpositions of trajectories”. *Proc. Roy. Soc. A* **475**, 20180903 (2019).
- [45] G. Chiribella, G. M. D’Ariano, and P. Perinotti. “Quantum circuit architecture”. *Phys. Rev. Lett.* **101**, 060401 (2008). [arXiv:0712.1325](#).
- [46] Giulio Chiribella, Giacomo Mauro D’Ariano, and Paolo Perinotti. “Theoretical framework for quantum networks”. *Phys. Rev. A* **80**, 022339 (2009). [arXiv:0904.4483](#).
- [47] Bob Coecke and Aleks Kissinger. “Picturing quantum processes: A first course in quantum theory and diagrammatic reasoning”. *Cambridge University Press*. (2017).
- [48] Ilyass Mejdoub and Augustin Vanrietvelde. “Unilateral determination of causal order in a cyclic process” (2025). [arXiv:2506.18540](#).
- [49] Man-Duen Choi. “Completely positive linear maps on complex matrices”. *Linear Algebra Appl.* **10**, 285–290 (1975).

- [50] Fabio Costa and Lee Rozema. “Scalable quantum control of causal order with single use of operations” (2025). In preparation; presented at Relativistic Quantum Information - North 2024.
- [51] Nick Ormrod, Augustin Vanrietvelde, and Jonathan Barrett. “Causal structure in the presence of sectorial constraints, with application to the quantum switch”. *Quantum* **7**, 1028 (2023).
- [52] Lukas Schmitt. “Operational causality in the Page-Wootters framework”. *Master’s thesis*. ETH Zurich. (2023).
- [53] Anne-Catherine de la Hamette, Viktoria Kabel, and Āaslav Brukner. “What an event is not: unravelling the identity of events in quantum theory and gravity” (2024). [arXiv:2404.00159](#).
- [54] Qiushi Liu, Zihao Hu, Haidong Yuan, and Yuxiang Yang. “Optimal strategies of quantum metrology with a strict hierarchy”. *Phys. Rev. Lett.* **130**, 070803 (2023). [arXiv:2203.09758](#).
- [55] Raphaël Mothe, Cyril Branciard, and Alastair A. Abbott. “Reassessing the advantage of indefinite causal orders for quantum metrology”. *Phys. Rev. A* **109**, 062435 (2024). [arXiv:2312.12172](#).
- [56] Raphaël Mothe, Alastair A. Abbott, and Cyril Branciard. “Correlations and quantum circuits with dynamical causal order” (2025). [arXiv:2507.07992](#).
- [57] Matthias Salzger and John H. Selby. “A decompositional framework for process theories in spacetime” (2025). [arXiv:2411.08266](#).
- [58] Emilien de Bank, V. Vilasini, and Cyril Branciard. “A fine-grained perspective on indefinite causal order: Modelling agents in spacetime” (2025). In preparation.
- [59] Robin Lorenz and Jonathan Barrett. “Causal and compositional structure of unitary transformations”. *Quantum* **5**, 511 (2021).
- [60] Augustin Vanrietvelde, Octave Mestoudjian, and Pablo Arrighi. “Causal decompositions of 1D quantum cellular automata” (2025). [arXiv:2506.22219](#).
- [61] Will Simmons and Aleks Kissinger. “A complete logic for causal consistency” (2024). [arXiv:2403.09297](#).
- [62] Aleks Kissinger and Sander Uijlen. “A categorical semantics for causal structure”. *Log. Methods Comput. Sci.* **15**, 3 (2019).
- [63] Eleftherios-Ermis Tselentis and Āmin Baumeler. “Admissible causal structures and correlations”. *PRX Quantum* **4**, 040307 (2023).
- [64] Luca Apadula, Alessandro Bisio, and Paolo Perinotti. “No-signalling constrains quantum computation with indefinite causal structure”. *Quantum* **8**, 1241 (2024).

Appendix

A Derivation of the choice function for $\mathcal{G}_{\text{QC-QC}(N)}$

Here we will evaluate explicitly the choice relation $\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}}$, given by Eq. (4.13):

$$\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}} = \lambda_{\mathbf{A}_1}^{\text{aug}} * \dots * \lambda_{\mathbf{A}_N}^{\text{aug}} * \lambda_{\mathbf{V}_1}^{\text{aug}} * \dots * \lambda_{\mathbf{V}_{N+1}}^{\text{aug}}. \quad (\text{A.1})$$

For this, let us start by explicitly writing the Boolean matrix elements of the different augmented relations in the graph:

- For $\lambda_{\mathbf{A}_k}^{\text{aug}}$, according to Eqs. (4.2) and (4.9), one can write (using the label $\mathcal{K}_{n_k-1}^{[\mathbf{A}_k]}$ rather than $\mathcal{K}_n \setminus k$ for the branch that happens at node \mathbf{A}_k , to distinguish it further below from the labels at the other nodes):

$$\left(\lambda_{\mathbf{A}_k}^{\text{aug}}\right)_{(\mathcal{X}_n^k)_{n=1}^N}^{(\mathcal{X}_n^k)_{n=1}^N, \mathcal{K}_{n_k-1}^{[\mathbf{A}_k]}} = \delta_{\mathcal{X}_n^k, \mathcal{K}_{n_k-1}^{[\mathbf{A}_k]} \cup k} \left(\prod_{n \neq n_k} \delta_{\mathcal{X}_n^k, \mathbb{Q}} \right). \quad (\text{A.2})$$

- For $\lambda_{\mathbf{V}_{n+1}}^{\text{aug}}$ ($n = 1, \dots, N-1$), according to Eqs. (4.4) and (4.10), one can write (here for ease of notations we keep the label \mathcal{K}_n for the branch that happens at node \mathbf{V}_{n+1} , rather than introducing $\mathcal{K}_n^{[\mathbf{V}_{n+1}]}$):

$$\begin{aligned} & \left(\Lambda_{\mathbf{V}_{n+1}}^{\text{aug}}\right)_{(\mathcal{X}_n^k)_{k=1}^N, \vec{\ell}_n = (\ell(\mathcal{K}'_n))_{\mathcal{K}'_n}}^{(\mathcal{X}_{n+1}^{\ell})_{\ell=1}^N, \mathcal{K}_n} \\ &= \left(\sum_{k \in \mathcal{K}_n} \left[\delta_{\mathcal{X}_n^k, \mathcal{K}_n} \left(\prod_{k' \in \mathcal{N} \setminus k} \delta_{\mathcal{X}_n^{k'}, \mathbb{Q}} \right) \right] \right) \left(\delta_{\mathcal{X}_{n+1}^{\ell(\mathcal{K}_n)}, \mathcal{K}_n \cup \ell(\mathcal{K}_n)} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\mathcal{K}_n)} \delta_{\mathcal{X}_{n+1}^{\ell'}, \mathbb{Q}} \right) \right), \end{aligned} \quad (\text{A.3})$$

while for $\lambda_{\mathbf{V}_1}^{\text{aug}}$ and $\lambda_{\mathbf{V}_{N+1}}^{\text{aug}}$, according to Eqs. (4.3), (4.11) and (4.12), one can write (ignoring the trivial inputs and outputs, see Fig. 4.4):

$$\left(\Lambda_{\mathbf{V}_1}^{\text{aug}}\right)_{\ell(\emptyset)}^{(\mathcal{X}_1^{\ell})_{\ell=1}^N} = \delta_{\mathcal{X}_1^{\ell(\emptyset)}, \ell(\emptyset)} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\emptyset)} \delta_{\mathcal{X}_1^{\ell'}, \mathbb{Q}} \right) \quad (\text{A.4})$$

$$\text{and } \left(\Lambda_{\mathbf{V}_{N+1}}^{\text{aug}}\right)_{(\mathcal{X}_N^k)_{k=1}^N} = \sum_{k \in \mathcal{N}} \left[\delta_{\mathcal{X}_N^k, \mathcal{N}} \left(\prod_{k' \in \mathcal{N} \setminus k} \delta_{\mathcal{X}_N^{k'}, \mathbb{Q}} \right) \right]. \quad (\text{A.5})$$

Recalling from Section 2.1.1 how the “link product” $*$ in Eq. (A.1) evaluates, we can write the Boolean matrix elements of the choice relation as

$$\begin{aligned} & \left(\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}}\right)_{\vec{\ell} = (\ell(\mathcal{K}'_n))_{n=0, \dots, N-1}; \mathcal{K}'_n}^{(\mathcal{K}_{n_k-1}^{[\mathbf{A}_k]})_{k \in \mathcal{N}}, (\mathcal{K}_n)_{n=1}^{N-1}} \\ &= \sum_{(\mathcal{X}_n^k)_{n;k}} \left(\prod_{k \in \mathcal{N}} \left(\lambda_{\mathbf{A}_k}^{\text{aug}}\right)_{(\mathcal{X}_n^k)_{n=1}^N}^{(\mathcal{X}_n^k)_{n=1}^N, \mathcal{K}_{n_k-1}^{[\mathbf{A}_k]}} \right) \\ & \quad \left(\Lambda_{\mathbf{V}_1}^{\text{aug}}\right)_{\ell(\emptyset)}^{(\mathcal{X}_1^{\ell})_{\ell=1}^N} \left(\prod_{n=1}^{N-1} \left(\Lambda_{\mathbf{V}_{n+1}}^{\text{aug}}\right)_{(\mathcal{X}_n^k)_{k=1}^N, \vec{\ell}_n = (\ell(\mathcal{K}'_n))_{\mathcal{K}'_n}}^{(\mathcal{X}_{n+1}^{\ell})_{\ell=1}^N, \mathcal{K}_n} \right) \left(\Lambda_{\mathbf{V}_{N+1}}^{\text{aug}}\right)_{(\mathcal{X}_N^k)_{k=1}^N}. \end{aligned} \quad (\text{A.6})$$

To evaluate this expression, using Eqs. (A.2) to (A.5) above, one may start by noticing that for $m = 1, \dots, N$,

$$\begin{aligned} & \left(\delta_{\mathcal{X}_m^{\ell(\mathcal{K}_{m-1})}, \mathcal{K}_{m-1} \cup \ell(\mathcal{K}_{m-1})} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\mathcal{K}_{m-1})} \delta_{\mathcal{X}_m^{\ell'}, \mathbb{Q}} \right) \right) \left(\sum_{k \in \mathcal{K}_m} \left[\delta_{\mathcal{X}_m^k, \mathcal{K}_m} \left(\prod_{k' \in \mathcal{N} \setminus k} \delta_{\mathcal{X}_m^{k'}, \mathbb{Q}} \right) \right] \right) \\ &= \delta_{\mathcal{K}_m, \mathcal{K}_{m-1} \cup \ell(\mathcal{K}_{m-1})} \delta_{\mathcal{X}_m^{\ell(\mathcal{K}_{m-1})}, \mathcal{K}_{m-1} \cup \ell(\mathcal{K}_{m-1})} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\mathcal{K}_{m-1})} \delta_{\mathcal{X}_m^{\ell'}, \mathbb{Q}} \right). \end{aligned} \quad (\text{A.7})$$

From this one easily obtains, recursively, that

$$\begin{aligned} & (\Lambda_{\mathbf{V}_1}^{\text{aug}})_{\ell^{(0)}}^{(\mathcal{X}_1^\ell)_{\ell=1}^N} \left(\prod_{n=1}^m (\Lambda_{\mathbf{V}_{n+1}}^{\text{aug}})_{(\mathcal{X}_n^k)_{k=1}^N, \vec{\ell}_n = (\ell(\mathcal{K}'_n))_{\mathcal{K}'_n}} (\mathcal{X}_{n+1}^\ell)_{\ell=1}^N, \mathcal{K}_n \right) \\ &= \delta_{\mathcal{K}_1, \ell^{(0)}} \delta_{\mathcal{K}_2, \mathcal{K}_1 \cup \ell(\mathcal{K}_1)} \cdots \delta_{\mathcal{K}_m, \mathcal{K}_{m-1} \cup \ell(\mathcal{K}_{m-1})} \prod_{n=0}^m \left(\delta_{\mathcal{X}_{n+1}^{\ell(\mathcal{K}_n)}, \mathcal{K}_n \cup \ell(\mathcal{K}_n)} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\mathcal{K}_n)} \delta_{\mathcal{X}_{n+1}^{\ell'}, \mathbb{Q}} \right) \right), \end{aligned} \quad (\text{A.8})$$

and that the third line in Eq. (A.6) takes the same form as above (for $m = N - 1$), so that (recalling also Eq. (A.2))

$$\begin{aligned} & (\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}})_{\vec{\ell} = (\ell(\mathcal{K}'_n))_{n=0, \dots, N-1; \mathcal{K}'_n}}^{(\mathcal{K}_{n_k}^{[A_k]})_{k \in \mathcal{N}}, (\mathcal{K}_n)_{n=1}^{N-1}} \\ &= \delta_{\mathcal{K}_1, \ell^{(0)}} \delta_{\mathcal{K}_2, \mathcal{K}_1 \cup \ell(\mathcal{K}_1)} \cdots \delta_{\mathcal{K}_{N-1}, \mathcal{K}_{N-2} \cup \ell(\mathcal{K}_{N-2})} \\ & \quad \sum_{(\mathcal{X}_n^k)_{n,k}} \left[\prod_{k \in \mathcal{N}} \left(\delta_{\mathcal{X}_n^k, \mathcal{K}_{n_k}^{[A_k]} \cup k} \left(\prod_{n \neq n_k} \delta_{\mathcal{X}_n^k, \mathbb{Q}} \right) \right) \right] \left[\prod_{n=0}^{N-1} \left(\delta_{\mathcal{X}_{n+1}^{\ell(\mathcal{K}_n)}, \mathcal{K}_n \cup \ell(\mathcal{K}_n)} \left(\prod_{\ell' \in \mathcal{N} \setminus \ell(\mathcal{K}_n)} \delta_{\mathcal{X}_{n+1}^{\ell'}, \mathbb{Q}} \right) \right) \right]. \end{aligned} \quad (\text{A.9})$$

Recalling the recursive definition of $\mathcal{K}_1^{\vec{\ell}} := \ell^{(0)}$, $\mathcal{K}_{n+1}^{\vec{\ell}} := \mathcal{K}_n^{\vec{\ell}} \cup \ell(\mathcal{K}_n^{\vec{\ell}})$ we introduced in Eq. (4.14), one can see that the above expression can only be nonzero if the \mathcal{K}_n 's satisfy the same recursive constraints, i.e. if $\mathcal{K}_n = \mathcal{K}_n^{\vec{\ell}}$ for all $n = 1, \dots, N - 1$. The last product term in square brackets above then restricts the Boolean sum to only index values \mathcal{X}_n^k such that, for any $n = 1, \dots, N$, $\mathcal{X}_n^k = \mathcal{K}_{n-1} \cup \ell(\mathcal{K}_{n-1}) = \mathcal{K}_n^{\vec{\ell}}$ if $k = \ell(\mathcal{K}_{n-1}) = \ell(\mathcal{K}_{n-1}^{\vec{\ell}})$ and $\mathcal{X}_n^k = \mathbb{Q}$ otherwise. For the first product term in square bracket to be nonzero, one then needs these to satisfy $\mathcal{X}_{n_k}^k \neq \mathbb{Q}$: this requires that for each $k \in \mathcal{N}$, n_k be such that $k = \ell(\mathcal{K}_{n_k-1}^{\vec{\ell}})$, i.e., n_k must be equal to $n_k^{\vec{\ell}}$ as we defined in Eq. (4.14). In that case $\mathcal{X}_{n_k}^k = \mathcal{K}_{n_k-1}^{\vec{\ell}}$, and the first Kronecker delta in the last line of Eq. (A.9) requires $\mathcal{K}_{n_k}^{\vec{\ell}} = \mathcal{K}_{n_k-1}^{[A_k]} \cup k$ – or equivalently, $\mathcal{K}_{n_k-1}^{[A_k]} = \mathcal{K}_{n_k-1}^{\vec{\ell}}$ – to be nonzero. Reciprocally, satisfying these various conditions indeed gives an overall value of 1 for the expression in Eq. (A.9) above.

All in all, we find that we can write

$$(\Lambda_{\mathcal{G}_{\text{QC-QC}(N)}})_{\vec{\ell} = (\ell(\mathcal{K}'_n))_{n=0, \dots, N-1; \mathcal{K}'_n}}^{(\mathcal{K}_{n_k}^{[A_k]})_{k \in \mathcal{N}}, (\mathcal{K}_n)_{n=1}^{N-1}} = \left(\prod_{k \in \mathcal{N}} \delta_{\mathcal{K}_{n_k-1}^{[A_k]}, \mathcal{K}_{n_k-1}^{\vec{\ell}}} \right) \left(\prod_{n=1}^{N-1} \delta_{\mathcal{K}_n, \mathcal{K}_n^{\vec{\ell}}} \right). \quad (\text{A.10})$$

As one sees, for any fixed input value $\vec{\ell} = (\ell(\mathcal{K}'_n))_{n=0, \dots, N-1; \mathcal{K}'_n}$ of the choice relation, there exists a unique output value $((\mathcal{K}_{n_k}^{[A_k]})_{k \in \mathcal{N}}, (\mathcal{K}_n)_{n=1}^{N-1})$ for which all Kronecker deltas above give the value 1, so that the Boolean matrix element is 1. The choice relation thus defines a proper function, as described in Eq. (4.15) of the main text.

B Obtaining a QC-QC by fleshing out a skeletal supermap

We provide here the detailed calculations that allowed us to obtain

$$\begin{aligned} & \left(\bigotimes_{n=0}^N |\mathbf{V}_{n+1}\rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) \\ &= \sum_{(k_1, k_2, \dots, k_N)} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2}\rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * |V_{\{k_1\}, k_2}^{\rightarrow k_3}\rangle^{A_{k_2}^O \alpha_2, A_{k_3}^I \alpha_3} * \dots \\ & \quad \dots * |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N}\rangle^{A_{k_{N-1}}^O \alpha_{N-1}, A_{k_N}^I \alpha_N} * |V_{\{k_1, \dots, k_{N-1}\}, k_N}^{\rightarrow F}\rangle^{A_{k_N}^O \alpha_N, F} = |w_{\text{QC-QC}}\rangle, \end{aligned} \quad (\text{B.1})$$

as in Eq. (4.34).

To prove this, we first show recursively that, for any $m = 0, \dots, N - 1$,

$$\begin{aligned}
& \left(\bigotimes_{n=0}^m |\mathbf{V}_{n+1}\rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) \\
&= \sum_{(k_1, k_2, \dots, k_m)} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2}\rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * \dots * |V_{\{k_1, \dots, k_{m-2}\}, k_{m-1}}^{\rightarrow k_m}\rangle^{A_{k_{m-1}}^O \alpha_{m-1}, A_{k_m}^I \alpha_m} \\
&\quad * \bigoplus_{k_{m+1}} |V_{\{k_1, \dots, k_{m-1}\}, k_m}^{\rightarrow k_{m+1}}\rangle^{A_{k_m}^O \alpha_m, A_{k_{m+1}}^I \alpha_{m+1}} * |\mathbb{1}\rangle^{A_{k_{m+1}}^O \alpha_{m+1}, \mathcal{H}_{\{k_1, \dots, k_m\}, k_{m+1}}^{\mathbf{A}_{k_{m+1}} \rightarrow \mathbf{V}_{m+2}}} \\
&\quad * \left(\bigotimes_{k \in \mathcal{N} \setminus \{k_1, \dots, k_{m+1}\}} |\mathbf{A}_k\rangle \right). \quad (\text{B.2})
\end{aligned}$$

Indeed, starting from $m = 0$, one has (changing the dummy index ℓ in Eq. (4.26) to k_1)

$$\begin{aligned}
|\mathbf{V}_1\rangle * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) &= \bigoplus_{k_1 \in \mathcal{N}} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, \mathcal{H}_{\emptyset, k_1}^{\mathbf{V}_1 \rightarrow \mathbf{A}_{k_1}}} * \left(|\mathbf{A}_{k_1}\rangle \otimes \bigotimes_{k \in \mathcal{N} \setminus k_1} |\mathbf{A}_k\rangle \right) \\
&= \bigoplus_{k_1} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, \mathcal{H}_{\emptyset, k_1}^{\mathbf{V}_1 \rightarrow \mathbf{A}_{k_1}}} * \left(\bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k_1} |\mathbb{1}\rangle^{\mathcal{H}_{\mathcal{K}_n \setminus \{k_1\}, k_1}^{\mathbf{V}_n \rightarrow \mathbf{A}_{k_1}}, A_{k_1}^I \alpha_n} * |\mathbb{1}\rangle^{A_{k_1}^O \alpha_n, \mathcal{H}_{\mathcal{K}_n \setminus \{k_1, k_1\}}^{\mathbf{A}_{k_1} \rightarrow \mathbf{V}_{n+1}}} \right) * \left(\bigotimes_{k \in \mathcal{N} \setminus k_1} |\mathbf{A}_k\rangle \right) \\
&= \bigoplus_{k_1} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1} * |\mathbb{1}\rangle^{A_{k_1}^O \alpha_1, \mathcal{H}_{\emptyset, k_1}^{\mathbf{A}_{k_1} \rightarrow \mathbf{V}_2}} * \left(\bigotimes_{k \in \mathcal{N} \setminus k_1} |\mathbf{A}_k\rangle \right), \quad (\text{B.3})
\end{aligned}$$

as in Eq. (B.2) – with the understanding that for $m = 0$, only the direct sum over k_1 was left, and that the link product of all $|V_{\cdot, \cdot}\rangle$'s reduced to just $|V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1}$.

Supposing that Eq. (B.2) holds for some value $m = 0, \dots, N - 2$, one can then write (similarly changing the dummy index ℓ in Eq. (4.25) to k_{m+2})

$$\begin{aligned}
& \left(\bigotimes_{n=0}^{m+1} |\mathbf{V}_{n+1}\rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) = \left(\bigotimes_{n=0}^m |\mathbf{V}_{n+1}\rangle \right) * |\mathbf{V}_{m+2}\rangle * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \right) \\
&= \sum_{(k_1, k_2, \dots, k_m)} |V_{\emptyset, \emptyset}^{\rightarrow k_1}\rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2}\rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * \dots * |V_{\{k_1, \dots, k_{m-2}\}, k_{m-1}}^{\rightarrow k_m}\rangle^{A_{k_{m-1}}^O \alpha_{m-1}, A_{k_m}^I \alpha_m} \\
&\quad * \bigoplus_{k_{m+1}} |V_{\{k_1, \dots, k_{m-1}\}, k_m}^{\rightarrow k_{m+1}}\rangle^{A_{k_m}^O \alpha_m, A_{k_{m+1}}^I \alpha_{m+1}} * |\mathbb{1}\rangle^{A_{k_{m+1}}^O \alpha_{m+1}, \mathcal{H}_{\{k_1, \dots, k_m\}, k_{m+1}}^{\mathbf{A}_{k_{m+1}} \rightarrow \mathbf{V}_{m+2}}} \\
&\quad * |\mathbf{V}_{m+2}\rangle * \left(\bigotimes_{k \in \mathcal{N} \setminus \{k_1, \dots, k_{m+1}\}} |\mathbf{A}_k\rangle \right) \\
&= \sum_{(k_1, k_2, \dots, k_m)} \dots * \bigoplus_{k_{m+1}} |V_{\{k_1, \dots, k_{m-1}\}, k_m}^{\rightarrow k_{m+1}}\rangle^{A_{k_m}^O \alpha_m, A_{k_{m+1}}^I \alpha_{m+1}} * |\mathbb{1}\rangle^{A_{k_{m+1}}^O \alpha_{m+1}, \mathcal{H}_{\{k_1, \dots, k_m\}, k_{m+1}}^{\mathbf{A}_{k_{m+1}} \rightarrow \mathbf{V}_{m+2}}} \\
&\quad * \bigoplus_{\substack{\mathcal{K}_{m+1}, k \in \mathcal{K}_{m+1}, \\ k_{m+2} \notin \mathcal{K}_{m+1}}} |V_{\mathcal{K}_{m+1} \setminus \{k, k\}}^{\rightarrow k_{m+2}}\rangle^{\mathcal{H}_{\mathcal{K}_{m+1} \setminus \{k, k\}}^{\mathbf{A}_k \rightarrow \mathbf{V}_{m+2}}, \mathcal{H}_{\mathcal{K}_{m+1}, k_{m+2}}^{\mathbf{V}_{m+2} \rightarrow \mathbf{A}_{k_{m+2}}}}} * \left(|\mathbf{A}_{k_{m+2}}\rangle \otimes \bigotimes_{k \in \mathcal{N} \setminus \{k_1, \dots, \dots, k_{m+2}\}} |\mathbf{A}_k\rangle \right) \\
&= \sum_{(k_1, k_2, \dots, k_m)} \dots * \bigoplus_{k_{m+1}} |V_{\{k_1, \dots, k_{m-1}\}, k_m}^{\rightarrow k_{m+1}}\rangle^{A_{k_m}^O \alpha_m, A_{k_{m+1}}^I \alpha_{m+1}} * |\mathbb{1}\rangle^{A_{k_{m+1}}^O \alpha_{m+1}, \mathcal{H}_{\{k_1, \dots, k_m\}, k_{m+1}}^{\mathbf{A}_{k_{m+1}} \rightarrow \mathbf{V}_{m+2}}} \\
&\quad * \bigoplus_{k_{m+2}} |V_{\{k_1, \dots, k_m\}, k_{m+1}}^{\rightarrow k_{m+2}}\rangle^{\mathcal{H}_{\{k_1, \dots, k_m\}, k_{m+1}}^{\mathbf{A}_{k_{m+1}} \rightarrow \mathbf{V}_{m+2}}, \mathcal{H}_{\{k_1, \dots, k_{m+1}\}, k_{m+2}}^{\mathbf{V}_{m+2} \rightarrow \mathbf{A}_{k_{m+2}}}}} \\
&\quad * \left(\bigoplus_{n=1}^N \bigoplus_{\mathcal{K}_n \ni k_{m+2}} |\mathbb{1}\rangle^{\mathcal{H}_{\mathcal{K}_n \setminus \{k_{m+2}, k_{m+2}\}}^{\mathbf{V}_n \rightarrow \mathbf{A}_{k_{m+2}}}, A_{k_{m+2}}^I \alpha_n} * |\mathbb{1}\rangle^{A_{k_{m+2}}^O \alpha_n, \mathcal{H}_{\mathcal{K}_n \setminus \{k_{m+2}, k_{m+2}\}}^{\mathbf{A}_{k_{m+2}} \rightarrow \mathbf{V}_{n+1}}} \right) * \left(\bigotimes_{k \in \mathcal{N} \setminus \{k_1, \dots, \dots, k_{m+2}\}} |\mathbf{A}_k\rangle \right)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{(k_1, k_2, \dots, k_m)} \cdots * \sum_{k_{m+1}} |V_{\{k_1, \dots, k_{m-1}\}, k_m}^{\rightarrow k_{m+1}} \rangle \rangle^{A_{k_m}^O \alpha_m, A_{k_{m+1}}^I \alpha_{m+1}} \\
&\quad * \bigoplus_{k_{m+2}} |V_{\{k_1, \dots, k_m\}, k_{m+1}}^{\rightarrow k_{m+2}} \rangle \rangle^{A_{k_{m+1}}^O \alpha_{m+1}, A_{k_{m+2}}^I \alpha_{m+2}} * |\mathbb{1}\rangle \rangle^{A_{k_{m+2}}^O \alpha_{m+2}, \mathcal{H}_{\{k_1, \dots, k_{m+1}\}, k_{m+2}}^{\mathbf{A}_{k_{m+2}} \rightarrow \mathbf{V}_{m+3}}} \\
&\quad * \left(\bigotimes_{k \in \mathcal{N} \setminus \{k_1, \dots, k_{m+2}\}} |\mathbf{A}_k\rangle \rangle \right), \quad (\text{B.4})
\end{aligned}$$

so that we obtain the same form as Eq. (B.2), for the value of $m + 1$ instead of m – which indeed proves recursively that Eq. (B.2) holds for all $m = 0, \dots, N - 1$. For $m = N - 1$ more specifically, we obtain:

$$\begin{aligned}
&\left(\bigotimes_{n=0}^{N-1} |\mathbf{V}_{n+1}\rangle \rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \rangle \right) \\
&= \sum_{(k_1, \dots, k_{N-1})} |V_{\emptyset, \emptyset}^{\rightarrow k_1} \rangle \rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2} \rangle \rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * \cdots * |V_{\{k_1, \dots, k_{N-3}\}, k_{N-2}}^{\rightarrow k_{N-1}} \rangle \rangle^{A_{k_{N-2}}^O \alpha_{N-2}, A_{k_{N-1}}^I \alpha_{N-1}} \\
&\quad * \bigoplus_{k_N} |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N} \rangle \rangle^{A_{k_{N-1}}^O \alpha_{N-1}, A_{k_N}^I \alpha_N} * |\mathbb{1}\rangle \rangle^{A_{k_N}^O \alpha_N, \mathcal{H}_{\{k_1, \dots, k_{N-1}\}, k_N}^{\mathbf{A}_{k_N} \rightarrow \mathbf{V}_{N+1}}}. \quad (\text{B.5})
\end{aligned}$$

With this in place, we can then write

$$\begin{aligned}
&\left(\bigotimes_{n=0}^N |\mathbf{V}_{n+1}\rangle \rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \rangle \right) = \left(\bigotimes_{n=0}^{N-1} |\mathbf{V}_{n+1}\rangle \rangle \right) * \left(\bigotimes_{k \in \mathcal{N}} |\mathbf{A}_k\rangle \rangle \right) * |\mathbf{V}_{N+1}\rangle \rangle \\
&= \sum_{(k_1, \dots, k_{N-1})} |V_{\emptyset, \emptyset}^{\rightarrow k_1} \rangle \rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2} \rangle \rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * \cdots * |V_{\{k_1, \dots, k_{N-3}\}, k_{N-2}}^{\rightarrow k_{N-1}} \rangle \rangle^{A_{k_{N-2}}^O \alpha_{N-2}, A_{k_{N-1}}^I \alpha_{N-1}} \\
&\quad * \bigoplus_{k_N} |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N} \rangle \rangle^{A_{k_{N-1}}^O \alpha_{N-1}, A_{k_N}^I \alpha_N} * |\mathbb{1}\rangle \rangle^{A_{k_N}^O \alpha_N, \mathcal{H}_{\{k_1, \dots, k_{N-1}\}, k_N}^{\mathbf{A}_{k_N} \rightarrow \mathbf{V}_{N+1}}} \\
&\quad * \bigoplus_{k \in \mathcal{N}} |V_{\mathcal{N} \setminus k, k}^{\rightarrow F} \rangle \rangle^{\mathcal{H}_{\mathcal{N} \setminus k, k}^{\mathbf{A}_k \rightarrow \mathbf{V}_{N+1}, F}}, \\
&= \sum_{(k_1, \dots, k_{N-1})} |V_{\emptyset, \emptyset}^{\rightarrow k_1} \rangle \rangle^{P, A_{k_1}^I \alpha_1} * |V_{\emptyset, k_1}^{\rightarrow k_2} \rangle \rangle^{A_{k_1}^O \alpha_1, A_{k_2}^I \alpha_2} * \cdots * |V_{\{k_1, \dots, k_{N-3}\}, k_{N-2}}^{\rightarrow k_{N-1}} \rangle \rangle^{A_{k_{N-2}}^O \alpha_{N-2}, A_{k_{N-1}}^I \alpha_{N-1}} \\
&\quad * \sum_{k_N} |V_{\{k_1, \dots, k_{N-2}\}, k_{N-1}}^{\rightarrow k_N} \rangle \rangle^{A_{k_{N-1}}^O \alpha_{N-1}, A_{k_N}^I \alpha_N} * |V_{\{k_1, \dots, k_{N-1}\}, k_N}^{\rightarrow F} \rangle \rangle^{A_{k_N}^O \alpha_N, F}, \quad (\text{B.6})
\end{aligned}$$

which gives precisely Eq. (4.34).

C Proofs for Section 5

Lemma 5.2. Considering the index values \mathfrak{Q} to be 1-dimensional, the split-node generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ is a valid routed graph.

Proof. The split-node generic routed graph is a modified version of the generic routed graph, featuring multiple nodes $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ in place of a single original node \mathbf{V}_{n+1} , being endowed with a subset of its arrows. However, all existing branches are preserved under this change.

The modification preserves the property of univocality: For the additional branches $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ with their singleton input/output index sets, no non-trivial bifurcation choice $\text{Ind}_{\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}}^{\text{out}}$ is made. Therefore, the choice function for $\mathcal{G}_{\text{QC-QC}(N)}$ (cf. Eq. (2.7)) remains practically unchanged: Whenever $\Lambda_{(\Gamma, (\lambda_N)_N)}$ evaluates to branch $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ for node \mathbf{V}_{n+1} , the new choice relation $\Lambda_{(\Gamma', (\lambda'_N)_N)}$ will evaluate to branch $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ for node $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, and to branch $\bar{\mathbf{V}}_{n+1}^{\mathcal{K}'_n}$ for any node $\bar{\mathbf{V}}_{n+1}^{\mathcal{K}'_n}$ with $\mathcal{K}'_n \neq \mathcal{K}_n$. An analogous argument applies for the adjoint for the graph, confirming bi-univocality of the resulting graph.

For the branch graph, the existing nodes $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ and the arrows connected therewith remain unchanged, while we receive new arrows $\bar{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ endowed with the same incoming green arrows and outgoing red arrows as $\mathbf{V}_{n+1}^{\mathcal{K}_n}$. By contrast, the $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ do not participate in any strong parent/child relationship, as they feature only a single, one-dimensional index value. Hence, the partial order of arrows, independent of colour, in the branch graph is preserved from the generic routed graph, and the rewriting does not introduce any loops into the new branch graph. \square

Lemma 5.3. Consider the two routed graphs $\mathcal{G}_{\text{QC-QC}(N)}$ and $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$. Restricting to the case in which all \mathbf{V}_n and $\mathbf{V}_{n+1}^{\mathcal{K}_n}$ are fleshed out by routed isometries, there is a canonical mapping between equivalent routed circuit decompositions (as defined in [Definition 2.26](#)) for the two routed graphs.

Proof. Any routed isometry V fleshing out the slot for a node \mathbf{V}_{n+1} needs to follow the branched route $\lambda_{\mathbf{V}_{n+1}}$ and therefore be of the form $V = \bigoplus_{\mathcal{K}_n} V^{\mathcal{K}_n}$ with $V^{\mathcal{K}_n} : \mathcal{H}_{\mathcal{K}_n}^{\text{in}(\mathbf{V}_{n+1})} \rightarrow \mathcal{H}_{\mathcal{K}_n}^{\text{out}(\mathbf{V}_{n+1})}$. Therefore, we can take these $V^{\mathcal{K}_n}$ individually to flesh out the slots associated with the branch nodes $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$, with $\hat{V}^{\mathcal{K}_n} := V^{\mathcal{K}_n} \oplus \mathbb{1}_{\mathbf{V}_{n+1}^{\mathcal{K}_n}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}) \rightarrow \text{out}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})}$, following the respective routes. Therefore, any routed circuit decomposition based on \mathcal{G} can equivalently be obtained using $\mathcal{G}^{\text{split}}$, provided that the slot for \mathbf{V}_{n+1} is fleshed out with a routed isometry rather than an routed superisometry.

Conversely, consider arbitrary routed circuit compositions for $\mathcal{G}^{\text{split}}$. While the most general fleshing out for the slot associated with $\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}$ would be

$$\hat{V}^{\mathcal{K}_n} = V^{\mathcal{K}_n} \oplus e^{i\varphi_{\mathcal{K}_n}} \cdot \mathbb{1}_{\mathbf{V}_{n+1}^{\mathcal{K}_n}}^{\text{in}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n}) \rightarrow \text{out}(\hat{\mathbf{V}}_{n+1}^{\mathcal{K}_n})}, \quad (\text{C.1})$$

we can restrict to $\varphi_{\mathcal{K}_n} = 0$, as a given phase $\varphi_{\mathcal{K}_n}$ could always be absorbed into the remaining $V^{\mathcal{K}_n}$. Therefore, any routed circuit decomposition based on $\mathcal{G}_{\text{QC-QC}(N)}^{\text{split}}$ can equivalently be obtained using $\mathcal{G}_{\text{QC-QC}(N)}$ as well (again, given that the slot for \mathbf{V}_{n+1} is fleshed out with a routed isometry rather than an routed superisometry). \square

Lemma 5.4. Let $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N} \in \text{Nodes}_{\Gamma}})$ be a valid routed graph without any weak loops, and $\mathbf{A}, \mathbf{V} \in \text{Nodes}_{\Gamma}$ such that \mathbf{A} is the sole child of \mathbf{V} (cf. [Fig. 5.4a](#)), and the associated branch structure is such that

$$\forall \mathbf{A}^{\beta} \in \text{Bran}(\lambda_{\mathbf{A}}) \quad \exists! \mathbf{V}^{\gamma} \in \text{Bran}(\lambda_{\mathbf{V}}) : \text{LinkVal}(\mathbf{V}^{\gamma}, \mathbf{A}^{\beta}) \neq \emptyset \quad (5.10)$$

Consider another routed graph $\mathcal{G}^{\text{merge}\uparrow}$ obtained by merging \mathbf{V} into \mathbf{A} , as follows:

- the two nodes being replaced by a single node \mathbf{A}' , with incoming arrows $\text{in}(\mathbf{A}') = \text{in}(\mathbf{V}) \cup (\text{in}(\mathbf{A}) \setminus \text{out}(\mathbf{V}))$ and outgoing arrows $\text{out}(\mathbf{A}') = \text{out}(\mathbf{A})$, as shown in [Fig. 5.4c](#),
- the resulting node \mathbf{A}' being endowed with the route $\lambda_{\mathbf{A}'} = \lambda_{\mathbf{A}} * \lambda_{\mathbf{V}}$.

Then $\mathcal{G}^{\text{merge}\uparrow}$ is a valid routed graph.

Proof. By construction, every $k \in \text{Ind}_{\mathbf{V} \rightarrow \mathbf{A}}^{\text{prac}} = \bigsqcup_{\mathbf{V}^{\gamma}} \text{Ind}_{\mathbf{V}^{\gamma}}^{\text{out}}$ can be associated with a branch \mathbf{V}^{γ} . [Equation \(5.10\)](#) additionally ensures that each k is associated with exactly one branch \mathbf{A}^{β} and $\bigsqcup_{\mathbf{A}^{\beta} : \text{LinkVal}(\mathbf{V}^{\gamma}, \mathbf{A}^{\beta}) \neq \emptyset} \text{Ind}_{\mathbf{A}^{\beta}}^{\text{in}} = \text{Ind}_{\mathbf{V}^{\gamma}}^{\text{out}}$.

Concatenating the branches \mathbf{A}^{β} and \mathbf{V}^{γ} (or rather, their respective part of the relation) to a single branch \mathbf{A}'^{γ} within a joint node \mathbf{A}' yields a ‘‘completely connected’’ (sub-)relation $\text{Ind}_{\mathbf{V}^{\gamma}}^{\text{in}} \rightarrow \bigsqcup_{\mathbf{A}^{\beta}} \text{Ind}_{\mathbf{A}^{\beta}}^{\text{out}}$, which may correspond to a branch (if the entire relation turns out to be branched). Concatenating all branches in this fashion, altogether we indeed confirm the resulting relation $\lambda_{\mathbf{A}'} = \lambda_{\mathbf{V}} * \lambda_{\mathbf{A}}$ to be branched.

We now consider how this changes the choice function $\Lambda_{\mathcal{G}}$: For the arguments, the bifurcation choices $\text{Ind}_{\mathbf{V}^{\gamma}}^{\text{in}}$ and $\text{Ind}_{\mathbf{A}^{\beta}}^{\text{out}}$ are affected, while in its image, multiple branches of \mathbf{A}^{β} may be coarse-grained to a single value of the choice function. However, due to the structure of the graph, any choice made at \mathbf{V}^{γ} is completely washed out at \mathbf{A}^{β} anyway, keeping the arguments of the

choice function practically unchanged for determining the remaining indices. Therefore, the choice function remains a function, and the graph is univocal.

As the scenario is inherently asymmetric, we also consider univocality for the adjoint graph \mathcal{G}^\top explicitly. In this graph, we have $\bigsqcup_{\mathbf{A}^\beta: \text{LinkVal}(\mathbf{A}^\beta, \mathbf{V}^\gamma) \neq \emptyset} \text{Ind}_{\mathbf{A}^\beta}^{\text{out}} = \text{Ind}_{\mathbf{V}^\gamma}^{\text{in}}$, and a branched route $\lambda_{\mathbf{A}^\beta}^\top$. However, the arguments of the choice function change in a different manner, with any bifurcation choice made within the branch \mathbf{V}^γ being preserved through \mathbf{A}^β , and therefore leaving the choice function practically invariant once more.

Finally, if the original graph did not feature any loops (which means that the branch graph is acyclic even ignoring any colours), contracting across an arrow can only introduce loops into the graph if another directed path exists between the respective nodes in the branch graph. As \mathbf{A} is the sole child of \mathbf{V} , this can not be the case here. Therefore, the routed graph is valid. \square

Lemma 5.5. Let $\mathcal{G} = (\Gamma, (\lambda_{\mathbf{N}})_{\mathbf{N} \in \text{Nodes}_\Gamma})$ be a valid routed graph without any weak loops, and $\mathbf{A}, \mathbf{V} \in \text{Nodes}_\Gamma$ such that \mathbf{A} is the sole parent of \mathbf{V} (cf. Fig. 5.4d) and the associated branch structure is such that

$$\forall \mathbf{A}^\beta \in \text{Bran}(\lambda_{\mathbf{A}}) \quad \exists! \mathbf{V}^\gamma \in \text{Bran}(\lambda_{\mathbf{V}}) : \quad \text{LinkVal}(\mathbf{A}^\beta, \mathbf{V}^\gamma) \neq \emptyset \quad (5.11)$$

Consider another routed graph $\mathcal{G}^{\text{merge}\downarrow}$ obtained by merging \mathbf{V} into \mathbf{A} , similarly to Lemma 5.4. Then $\mathcal{G}^{\text{merge}\downarrow}$ is a valid routed graph.

Proof. This is the adjoint scenario for Lemma 5.4, so the proof follows by analogy. \square

Proposition 5.6. Any process with a routed circuit decomposition (as defined in Definition 2.26) according to the routed graph \mathcal{G} , where \mathbf{V} is fleshed out with a routed isometry (an ‘‘internal’’ node), can also be expressed with a routed circuit decomposition according to $\mathcal{G}^{\text{merge}\uparrow} / \mathcal{G}^{\text{merge}\downarrow}$, related to \mathcal{G} via Lemma 5.4 / Lemma 5.5, and vice versa.

Proof. The backward direction of the proof is obvious: For any fleshing out of a mergeable node pair \mathbf{A}, \mathbf{V} in \mathcal{G} , we obtain an associated fleshing out for $\mathcal{G}^{\text{merge}\uparrow/\downarrow}$ by fleshing out the resulting node with $|\mathbf{A}\rangle\rangle * |\mathbf{V}\rangle\rangle$, composing the respective routed (super)maps along the composition of the respective routes.

For the forward direction, it remains to show that splitting the merged node into two, ‘‘undoing’’ the merge, does not restrict the set of possible operations due to a insufficient dimension for (some) sectors of $\mathcal{H}^{\mathbf{V} \rightarrow \mathbf{A}} / \mathcal{H}^{\mathbf{A} \rightarrow \mathbf{V}}$, which could lead to a more restricted set of processes. To do so, we remind of the fact that the practical (and therefore operationally relevant) Hilbert space $\mathcal{H}_{\text{prac}}^{\text{in}/\text{out}}(\mathbf{N})$ of a node \mathbf{V} is given by the direct sum of all individual branch spaces $\mathcal{H}_{\mathbf{V}^\gamma}^{\text{in}/\text{out}}(\mathbf{N})$.

I. Proof for $\mathcal{G}^{\text{merge}\uparrow}$: For any fleshing out of \mathbf{V} (with a routed isometry), we have $\forall \mathbf{V}^\gamma \in \text{Bran}(\lambda_{\mathbf{V}}) : \dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{out}}(\mathbf{V})) \geq \dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{in}}(\mathbf{V}))$. Therefore, no valid dimension assignment for $\mathcal{H}^{\mathbf{V} \rightarrow \mathbf{A}}$ (with $\dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{out}}(\mathbf{V})) = \dim(\bigoplus_{k \in \text{Ind}_{\mathbf{V}^\gamma}^{\text{out}}} \mathcal{H}_k^{\mathbf{V} \rightarrow \mathbf{A}})$) can restrict the set of possible operations of form $\mathcal{H}^{\text{in}}(\mathbf{V}) \xrightarrow{\lambda_{\mathbf{A} \rightarrow \mathbf{V}}} \mathcal{H}^{\text{out}}(\mathbf{A})$.

II. Proof for $\mathcal{G}^{\text{merge}\downarrow}$:

First, consider the case where the skeletal supermap specifies that \mathbf{V} must be fleshed out by a routed unitary, which entails that the dimension assignment for the skeletal supermap is such that $\forall \mathbf{V}^\gamma \in \text{Bran}(\lambda_{\mathbf{V}}) : \dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{out}}(\mathbf{V})) = \dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{in}}(\mathbf{V}))$. In this case, the dimension for $\dim(\mathcal{H}_{\mathbf{V}^\gamma}^{\text{in}}(\mathbf{V})) = \dim(\bigoplus_{k \in \text{Ind}_{\mathbf{V}^\gamma}^{\text{in}}} \mathcal{H}_k^{\mathbf{V} \rightarrow \mathbf{A}})$ is clearly sufficient to not restrict the set of possible operations $\mathcal{H}^{\text{in}}(\mathbf{A}) \xrightarrow{\lambda_{\mathbf{A} \rightarrow \mathbf{V}}} \mathcal{H}^{\text{out}}(\mathbf{V})$. However, for the routed circuit decomposition as a whole, the restriction to a routed unitary is actually without loss of generality: Specifying a monopartite routed superisometry to flesh out \mathbf{A} , we may choose it so that it prepares some independent state and sends it to \mathbf{V} , allowing to recover any (routed) isometry by composition of this state with the (routed) unitary. The specifics of this argument are analogous to the proof of Corollary 2.25. \square

D Reconstruction of the routed circuit decomposition of the Grenoble process in [39]

In Figure 25 of [39], a routed quantum circuit decomposition of the Grenoble process is displayed. While we outlined the general procedure to obtain this representation from the $\mathcal{G}_{\text{QC-QC}(N)}$ in Example 5.8, we did not specify a detailed mapping between the two representations.

Here, we provide the detailed translations required to relate both representations. For the index values, we have:

$$\begin{aligned}
l = 1 &\rightarrow \mathcal{X}_0^A = \{A\}, & m = 1 &\rightarrow \mathcal{X}_0^B = \{B\}, & n = 1 &\rightarrow \mathcal{X}_0^C = \{C\}, \\
l_1 = 1 &\rightarrow \mathcal{X}_A^B = \{A, B\}, & m_1 = 1 &\rightarrow \mathcal{X}_B^C = \{B, C\}, & n_1 = 1 &\rightarrow \mathcal{X}_C^A = \{C, A\}, \\
l_2 = 1 &\rightarrow \mathcal{X}_A^C = \{A, C\}, & m_2 = 1 &\rightarrow \mathcal{X}_B^A = \{A, B\}, & n_2 = 1 &\rightarrow \mathcal{X}_C^B = \{B, C\}, \\
f = 1 &\rightarrow \mathcal{X}_{BC}^A = \{A, B, C\}, & g = 1 &\rightarrow \mathcal{X}_{AC}^B = \{A, B, C\}, & h = 1 &\rightarrow \mathcal{C}_{AB}^C = \{A, B, C\}.
\end{aligned} \tag{D.1}$$

For all of their indices, if they take the value 0, we set the associated index $\mathcal{X}_{\mathcal{K}_n}^k = \emptyset$.

Slightly modifying the circuit by routing E^g trivially through W_B , for the Hilbert spaces we have:

$$\mathcal{H}^{P_C} \otimes \mathcal{H}^{P_T} \sim \mathcal{H}^P, \tag{D.2}$$

$$\mathcal{H}^{F_T} \sim \mathcal{H}^F, \tag{D.3}$$

$$\mathcal{H}^{F_A} \otimes \mathcal{H}^{F_C} \sim \mathcal{H}^{\alpha_F}, \tag{D.4}$$

$$\mathcal{H}^{E^g} \otimes \mathcal{H}^{Q^{g m_1 n_2}} \sim \mathcal{H}^{\bar{\alpha}} \otimes \mathcal{H}^{\bar{C}^{(k)}}, \tag{D.5}$$

$$\mathcal{H}^{E^g} \otimes \mathcal{H}^{S^g} \sim \mathcal{H}^{C \rightarrow V_4}, \tag{D.6}$$

$$\mathcal{H}^{E^{g=1}} \otimes \mathcal{H}^{S^{g=1}} \sim \mathcal{H}^{\alpha_3} \otimes \mathcal{H}^{C_3^{(B)}}, \tag{D.7}$$

$$\mathcal{H}^{X^{m_1}} \sim \mathcal{H}^{B \rightarrow C}, \tag{D.8}$$

$$\mathcal{H}^{Y^{m_2 n_2}} \sim \mathcal{H}^{B \rightarrow A} \tag{D.9}$$

etc. Similarly, we identify

$$U_P \sim |\mathbf{V}_1\rangle, \quad \mathcal{A}_F \sim |\mathbf{V}_4\rangle, \tag{D.10}$$

$$V_B \sim |J_{\mathbf{V}_3^A}^{\text{in}}\rangle * |\tilde{V}_3^{AC}\rangle * |J_{\mathbf{V}_3^A}^{\text{out}}\rangle * |J_{\mathbf{B}}^{\text{in}}\rangle, \tag{D.11}$$

$$W_B \sim |J_{\mathbf{B}}^{\text{out}}\rangle * |J_{\mathbf{V}_2^B}^{\text{in}}\rangle * |\tilde{V}_2^B\rangle * |J_{\mathbf{V}_2^B}^{\text{out}}\rangle \tag{D.12}$$

etc. Indeed we see that system E^g being withheld from the agent B ensures that its output dimension does not change, even though $2 = \dim(\mathcal{H}^{\alpha_3}) > \dim(\mathcal{H}^{\alpha_2}) = 1$, while the agent's system dimensions are constant. While our notations may differ significantly, we reproduce essentially the same representation for the process.

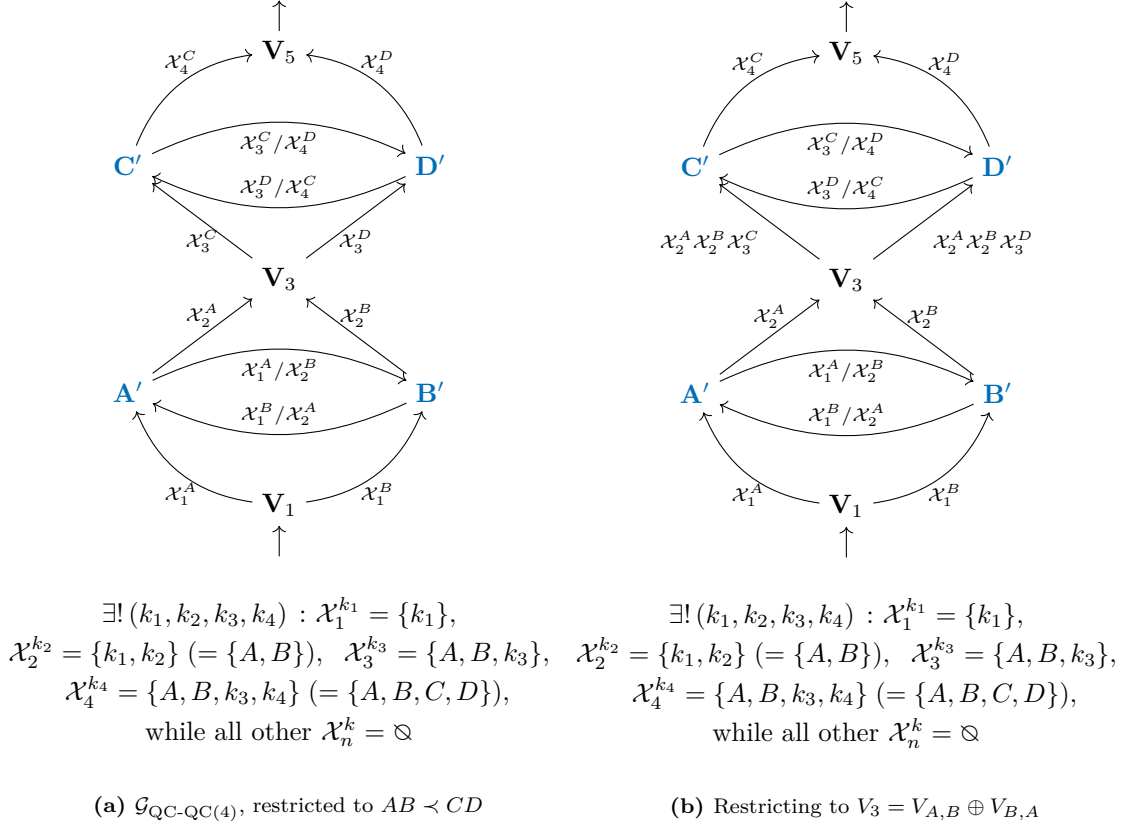


Figure E.1: Two routed graphs for 4-partite processes with $AB \prec CD$. While (a) can represent any QC-QC with this order, representation (b) is only applicable if \mathbf{V}_3 admits a more fine-grained branch structure into two branches $\mathbf{V}_{A,B}$ and $\mathbf{V}_{B,A}$. [Example E.1](#) can not be represented with the latter graph.

E Restricting to agent-localised processing

In [Section 5.3](#), we have demonstrated how to obtain an alternative routed circuit decomposition with fewer internal nodes by modifying our generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$, eliminating \mathbf{V}_2 and \mathbf{V}_N . Based on this, we consider an example which demonstrates how a removal of further internal nodes (between \mathbf{V}_2 and \mathbf{V}_N) generally fails, and under which conditions we may recover the possibility to get rid of further internal nodes.

Example E.1 (Zurich process [[41](#), [52](#)]). The *Zurich process*, introduced in [[41](#)] as double quantum switch (and later independently proposed in Appendix B of [[52](#)]), is given as follows:

$$V_{\emptyset, \emptyset}^{\rightarrow k_1} = \frac{1}{\sqrt{2}} |\psi\rangle^{A_{k_1}^I} \quad (\text{E.1})$$

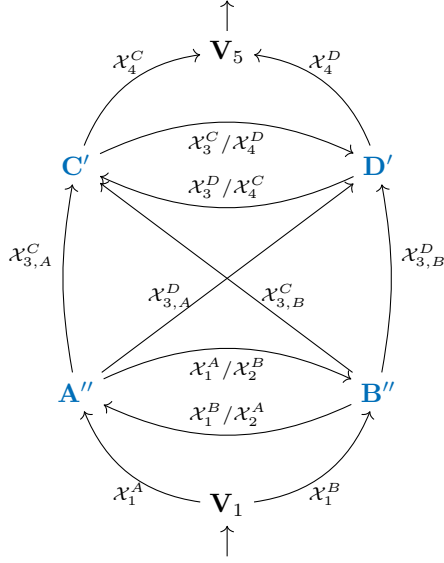
$$V_{\emptyset, k_1}^{\rightarrow k_2} = \mathbb{1}^{A_{k_1}^O \rightarrow A_{k_1}^I} \quad (\text{E.2})$$

$$V_{\{k_1\}, k_2}^{\rightarrow k_3} = \begin{cases} -\frac{1}{\sqrt{2}} \mathbb{1}^{A_{k_2}^O \rightarrow A_{k_3}^I}, & k_2 = B \text{ and } k_3 = C \\ \frac{1}{\sqrt{2}} \mathbb{1}^{A_{k_2}^O \rightarrow A_{k_3}^I}, & \text{else} \end{cases} \quad (\text{E.3})$$

$$V_{\{k_1, k_2\}, k_3}^{\rightarrow k_4} = \mathbb{1}^{A_{k_3}^O \rightarrow A_{k_4}^I} \quad (\text{E.4})$$

$$V_{\{k_1, k_2, k_3\}, k_4}^{\rightarrow F} = \mathbb{1}^{A_{k_4}^O \rightarrow F} \otimes |k_4 \bmod 2\rangle^{\alpha_F} \quad (\text{E.5})$$

where it is assumed that $k_1, k_2 \in \{A, B\}$ and $k_3, k_4 \in \{C, D\}$, while all other internal operations are assumed to vanish. Hence, it can be represented using the routed graph depicted in [Fig. E.1a](#). Intuitively, we can see the process as two subsequent quantum switches, with the operation performed



$$\begin{aligned}
& \exists! (k_1, k_2, k_3, k_4) : \mathcal{X}_1^{k_1} = \{k_1\}, \\
& \mathcal{X}_2^{k_2} = \{k_1, k_2\} (= \{A, B\}), \quad \mathcal{X}_{3,k_2}^{k_3} = \{A, B, k_3\}, \\
& \mathcal{X}_4^{k_4} = \{A, B, k_3, k_4\} (= \{A, B, C, D\}), \\
& \text{while all other } \mathcal{X}_n^k, \mathcal{X}_{3,k_2}^{k_3} = \emptyset
\end{aligned}$$

Figure E.2: A routed graph which may represent all 4-partite processes which feature order $AB \prec CD$ and support only agent-localised transformations. Any process supported by this routed graph can be understood as a wiring of independent pre- and post-processing (with memory) localised to the individual agents. This representation is obtained starting from Fig. E.1b, by additionally assuming a more fine-grained branch structure imposing $V_3 = V_{A,B} \oplus V_{B,A}$ for the internal node \mathbf{V}_3 . We split the respective node into two nodes $\hat{\mathbf{V}}_{B,A}$ and $\hat{\mathbf{V}}_{A,B}$ according to Section 5.2, and merge the resulting nodes into their respective parents \mathbf{A}' and \mathbf{B}' afterwards to obtain \mathbf{A}'' and \mathbf{B}'' . One would obtain the same final result for the routed graph by assuming a branch structure imposing $V_3 = V^{\rightarrow C} \oplus V^{\rightarrow D}$ instead, albeit with \mathbf{C} and \mathbf{D} featuring the $''$ instead.

at \tilde{V}_3 acting as a basis change of the control system from the computational basis to $\{|+\rangle, |-\rangle\}$.⁵²

Having only a single branch, \tilde{V}_3 does not admit a direct sum decomposition $\tilde{V}_3 = V_{A,B} \oplus V_{B,A}$ or $\tilde{V}_3 = V^{\rightarrow C} \oplus V^{\rightarrow D}$ into independent isometries. Indeed, the respective maps $V_{k_1, k_2}^{\rightarrow k_3}$ do not constitute isometries on the respective subspaces when considered in isolation.

This raises the question whether we can capture relevant subclasses of processes, allowing for such a decomposition, by adjusting the index constraints to produce a more fine-granular branch structure. For instance, the structure $V_{A,B} \oplus V_{B,A}$ suggests that at the third agent C or D , the second agent is encoded in $\mathcal{H}^{\text{in}(C/D)}$ ⁵³. This is not the case for the Zurich process, but would allow for many simpler 4-partite processes, including a concatenation of two quantum switches controlled by the same basis or by two separate, uncorrelated qubits.

Encoding this branch structure using additional index values and constraints upon them, we obtain Fig. E.1b from Fig. E.1a. The changed branch structure then allows to perform further node splitting (according to Lemma 5.3) and node merging (according to Proposition 5.6), getting rid of \mathbf{V}_3 in the routed graph similarly as for \mathbf{V}_2 . Remarkably, we would have received the same end result for the routed graph if we had chosen a branch structure imposing $V^{\rightarrow C} \oplus V^{\rightarrow D}$ instead, ultimately performing a merge of $\hat{\mathbf{V}}^{\rightarrow C} \hookrightarrow \mathbf{C}'$ and $\hat{\mathbf{V}}^{\rightarrow D} \hookrightarrow \mathbf{D}'$. This reflects the fact that after merging, the routed graph recovers the symmetry of the original routed graph Fig. E.1a under flipping all arrows.

This suggests a more general class of processes with an arbitrary number of agents, which can be represented without any need of internal nodes other than \mathbf{V}_1 and \mathbf{V}_{N+1} . The action of the internal isometries is then restricted to local pre- and post-processing⁵⁴ at the respective agents: At each node, the next agent is independently chosen, without interference from any other agents. Altogether, this amounts of a process which decomposes into a set of quantum 1-combs (and two routed isometries) wired together in a routed circuit. We specify the generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{local}}$ capturing implementations of this form as follows:

Definition E.2 ($\mathcal{G}_{\text{QC-QC}(N)}^{\text{local}}$). The routed graph $\mathcal{G}_{\text{QC-QC}(N)}^{\text{local}}$, representing *agent-localised processes* with minimal internal nodes, is defined as follows:

- There are 2 types of nodes: “**A**-nodes” \mathbf{A}_k , for $k = 1, \dots, N$, and “**V**-nodes” \mathbf{V}_1 and \mathbf{V}_{N+1} .
- The indexed graph involves the following indexed arrows:

- open-ended arrows $\longrightarrow \mathbf{V}_1$ and $\mathbf{V}_{N+1} \longrightarrow$,
- $\forall k = 1, \dots, N$,

$$\mathbf{V}_1 \xrightarrow{\mathcal{X}_1^k} \mathbf{A}_k \quad \text{and} \quad \mathbf{A}_k \xrightarrow{\mathcal{X}_N^k} \mathbf{V}_{N+1}, \quad (\text{E.6})$$

with the indices \mathcal{X}_n^k taking one of two values each: either $\mathcal{X}_1^k = k$ or $\mathcal{X}_N^k = \mathcal{N} \setminus k$ as its value, or $\mathcal{X}_n^k = \emptyset$. The latter value will later be considered 1-dimensional.

- $\forall n = 2, \dots, N-1, \quad \forall k = 1, \dots, N, \quad \forall \ell = 1, \dots, N$,

$$\mathbf{A}_k \xrightarrow{\mathcal{X}_{n,k}^\ell} \mathbf{A}_\ell, \quad (\text{E.7})$$

with a given index $\mathcal{X}_{n,k}^\ell$ taking either $\mathcal{X}_{n,k}^\ell = \mathcal{K}_n$ with $k, \ell \in \mathcal{K}_n$ as its value, or $\mathcal{X}_n^k = \emptyset$. The latter value will later be considered 1-dimensional.

- The routes are specified such that

⁵²In particular, this implies that even dependent on isometries A, B, C, D chosen by the agents, no POVM $\mathcal{M}(A, B, C, D)$ on $\mathcal{H}^F \otimes \mathcal{H}^{\alpha_F}$ can return the entire causal order in the process, while this is possible for the quantum switch or the Grenoble process.

⁵³In absence of an ancillary system α_3 , it will be encoded in \mathcal{H}^{C^I/D^I} . If the respective information is passed through an ancillary, it will be passed further along through the process instead.

⁵⁴with a side-channel $\bar{\alpha}$

- for each node \mathbf{A}_k , the lists of all input and output index values $(\mathcal{X}_{n,j}^k)_{n,j} \in \text{Ind}_{\mathbf{A}_k}^{\text{in}}$ and $(\mathcal{X}_{n,k}^\ell)_{n,\ell} \in \text{Ind}_{\mathbf{A}_k}^{\text{out}}$ must satisfy that there is exactly one $1 \leq n \leq N-1$ such that

$$\exists! j, \mathcal{X}_{n,j}^k \neq \emptyset, \exists! \ell, \mathcal{X}_{n+1,k}^\ell \neq \emptyset; \quad \text{and} \quad \mathcal{X}_{n+1,k}^\ell = \mathcal{X}_{n,j}^k \cup \ell. \quad (\text{E.8})$$

For all other values of n , all indices $\mathcal{X}_{n,j}^k$ and $\mathcal{X}_{n,k}^\ell$ are \emptyset .

- for \mathbf{V}_1 , $\mathcal{X}_\emptyset^\ell \in \text{Ind}_{\mathbf{V}_1}^{\text{out}}$ must satisfy $\exists! \ell, \mathcal{X}_\emptyset^\ell \neq \emptyset$.
- for \mathbf{V}_N , $\mathcal{X}_N^k \in \text{Ind}_{\mathbf{V}_N}^{\text{in}}$ must satisfy $\exists! k, \mathcal{X}_{N \setminus k}^k \neq \emptyset$.

They condense to the *global index constraints* that $\exists! (k_1, k_2, \dots, k_N)$:

$$\mathcal{X}_1^{k_1} = \{k_1\}, \mathcal{X}_{2,k_1}^{k_2} = \{k_1, k_2\}, \dots, \mathcal{X}_{N-1, k_{N-2}}^{k_{N-1}} = \{k_1, k_2, \dots, k_{N-1}\}, \mathcal{X}_N^k = \mathcal{N}$$

while all other $\mathcal{X}_{n,k}^\ell = \emptyset$. This captures the understanding that, along each global path, all operations A_k are applied once and only once, in a given order.

For $N = 4$, this graph is shown in [Fig. E.2](#).

Actually, one may simplify $\mathcal{G}_{\text{QC-QC}(N)}^{\text{local}}$ by removing the label n for each index $\mathcal{X}_{n,k}^\ell$, as generally $|\mathcal{X}_{n,k}^\ell| = n$, further estranging $\mathcal{G}_{\text{QC-QC}(N)}^{\text{local}}$ from the original generic routed graph $\mathcal{G}_{\text{QC-QC}(N)}$. As expected, we confirm that up to relabelling, $\mathcal{G}_{\text{QC-QC}(2)}^{\text{local}}$ and $\mathcal{G}_{\text{QC-QC}(3)}^{\text{local}}$ reproduce the generic routed graphs we received after node merging in [Section 5.3](#).

We conjecture that there is an implementation-agnostic SDP characterisation of the subclass of QC-QCs which may be implemented in this fashion, and leave proving this conjecture for future work.