Towards Scalable Simulation of Stochastic Bigraphs

Foundations for the Stochastic Bigraphical Abstract Machine (SBAM)

Espen Højsgaard
Jean Krivine

IT University Technical Report Series **TR-2011-148**

ISSN 1600–6100 **December 2011**

ISSN 1600–6100

ISBN 978-87-7949-247-9

Copies may be obtained by contacting:

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S
Denmark

Telephone:	+45 72 18 50 00
Telefax:	+45 72 18 50 01
Web	www.itu.dk

Abstract

We report on the progress of the development and implementation of an efficient and scalable simulation algorithm for stochastic bigraphical reactive systems (BRSs).

The starting point is the stochastic simulation algorithm for the κ-calculus by Danos et al. [12] (henceforth KaSim). Since the κ-calculus is a graphical formalism with a straightforward BRS representation, we are hopeful that their algorithm generalizes to BRSs. The KaSim algorithm relies on a number of concepts that have not previously been developed for BRSs: embeddings, localized matching, redex and agent modifications, and fine-grained conflict/causality analysis at the level of rules exploiting the notion of modification. In this report, we rigorously develop bigraph embeddings and redex/agent modifications, give an algorithm for localized matching, and outline a fine-grained conflict/causality analysis.

Our implementation strategy is to represent the bigraphical structures as directly as possible, as we believe that this eases implementation and increases trust in correctness. However, it is difficult to directly represent the structures of the usual presentation of the theory of BRSs: any non-trivial BRS contains an infinite number of ground reaction rules, since the set of rules must be closed under support equivalence and a parametric reaction rules generates an infinite set of ground reaction rules. In addition, the usual presentation of the dynamic theory of BRSs combines poorly with the stochastic semantics: the stochastic semantics rely on support, i.e., concrete bigraphs, while dynamics are defined up to support equivalence. We therefore develop, and prove equivalent, an alternative dynamic theory for BRSs without these problems: (i) the set of rules need not be closed under support equivalence, (ii) parametric reaction rules are first-class citizens, and (iii) integrates a (generalized) stochastic semantics. The development is based on the more general theory of reactive systems, and is thus applicable in more settings than just (stochastic) BRSs.

The completed parts of our work have been implemented in a prototype called the Stochastic Bigraphical Abstract Machine (SBAM), which currently allows stochastic simulation of BRSs where each redex consists of a single connected component and is *solid*, i.e., matches are determined by support translations of its nodes.

Contents

1 Introduction

The theory of bigraphs arose as a generalization of process calculi, and provides a unifying framework for modeling systems of mobile and communicating agents. The theory excels in that it provides a general method for deriving labeled transition systems (LTSs) from reaction semantics, with the nice property that, in the derived LTSs, bisimulation is a congruence.

But it is also a theory with a nice graphical representation, which enables models that are more intuitive than corresponding process calculi models. This has recently been exploited by Damgaard et al. [9, 10] and Bacci et al. [1] to give models of protein interaction and dynamic compartmentalization in cellular biology. In combination with Krivine et al.'s stochastic semantics for bigraphs [25], these works enable us to construct models of biological cells that may be simulated by a computer; the only thing missing is a simulator for stochastic bigraphs, which is what we have set out to build.

Our starting point is the efficient and scalable simulator for the κ-calculus [11] by Danos et al. and its underlying algorithm (which we call KaSim) [12]: Since the κ-calculus is a graphical formalism with a straightforward BRS representation, we are hopeful that KaSim generalizes to bigraphs. The KaSim algorithm relies on a number of concepts that have not previously been developed for BRSs: embeddings, localized matching, redex and agent modifications, and fine-grained causality analysis at the level of rules exploiting the notion of modification. In this report, we rigorously develop bigraph embeddings and redex/agent modifications, and outline algorithms for localized matching and fine-grained causality analysis:

bigraph embeddings:
We develop a general theory of bigraph embeddings that are isomorphic to decompositions of the form $H = C \circ ((G \otimes \mathsf{id}_X) \circ D \otimes \mathsf{id}_{\langle k,Y \rangle})$ where D is discrete (some detail omitted). In particular, embeddings of redexes into agents are isomorphic to matches. We also show that embeddings of so-called solid bigraphs are determined by support translations of their nodes.

edit scripts:
We propose a set of minimal modifications to a redex, called *edits*, and show how the modification of a redex can be transferred to an agent through an embedding, giving rise to an alternative, but equivalent, way to define reaction. We prove that a sequence of edits, an *edit script*, can realize any parametric reaction rule and vice versa.

rule activation and inhibition:
We outline an approach to characterizing causality and conflict at the level of rules, called respectively rule activation and rule inhibition, based on the idea of characterizing overlaps between bigraphs as a set of pullbacks in the category of embeddings.

anchored matching:
We give a localized matching algorithm, based on the idea of expanding a partial embedding of a redex to total embeddings.

However, before we can hope to generalize and implement the KaSim algorithm for bigraphs, we must develop a formulation of stochastic bigraphs that is more amenable to implementation than the usual formulations. For example, neither Milner's definition of bigraph dynamics [29] nor Krivine et al.'s stochastic semantics for bigraphs [25] lend themselves easily to implementation for the following reasons:

- The various definitions, e.g., the definitions of matches, reactions, and stochastic rates, rely on *support*, i.e., node and edge identities. But at the same time, the same definitions always close under support equivalence, whereby it becomes unclear how to handle support in practice.
- The, from a modeling perspective, essential concept of *parametric* reaction rules are treated as generators of infinite families of *ground* (non-parametric) reaction rules, which clearly cannot be represented directly in an implementation.

Furthermore, the formulation of stochastic bigraphs in [25] have two minor deficiencies: it only defines stochastic semantics for BRSs with linear rules and so-called *solid* redexes, and there is a gap between the definition of the reaction semantics and the stochastic semantics, as they rely on seemingly different definitions of matches. We develop a theory of *stochastic parametric reactive systems (SPRS)* which unifies and generalizes Milner's reactive systems and the stochastic semantics of Krivine et al., while avoiding the above issues. Similar to Milner's reactive systems, we define SPRSs at the more abstract level of s-categories, of which bigraphs are an instance.

1.1 Related work

Parametric Reactive Systems Our SPRSs are related to, and their formulation inspired by, the *parametric reactive systems* of Debois, where parametric reaction rules are also first-class citizens [13]. However, contrary to our formulation, Debois does not make explicit that context and parameter may be connected without the involvement of the redex. This has the consequence that bigraphical reaction rules become generators of infinite families of rules. Furthermore, we go further than Debois, by formally showing that our formulation is equivalent to the usual (non-parametric) reactive systems.

Bigraph Implementations A number of implementations of bigraphs are being developed at various institutions. Unfortunately, it is hard to find the implementations themselves or papers describing them, but here is a complete list of the implementations which we are aware of:

BigMC: A model checker for bigraphs which includes a command line interface and visualization [4].

bigraphspace: A Java library which provides a tuple-space-like API based on bigraphs [21].

Big Red: A graphical editor for bigraphs with easily extensible support for various file formats [17].

BigWB: A graphical workbench for bigraphs, aiming at providing a unifying GUI for the various bigraph tools (work in progress, no website or papers at the time of writing).

BPL Tool: A command line tool for experimenting with (abstract) binding bigraphs based on Damgaard et al.'s inductive characterization of matching in binding bigraphs [6] [7, 20, 23].

CLF based: Beauquier and Schürmann have given a model of bigraphs in the type theory CLF [3], and thus CLF implementations, such as Celf [33], may be used for bigraphs.

DBtk: A tool for directed bigraphs, which provides calculation of IPOs, matching, and visualization [2].

SAT based algorithm: Sevegnani et al. have presented a SAT based algorithm for matching in place graphs with sharing [34] and an implementation is in progress based on MiniSAT [14].

Bigraphs vs Graph Transformation Ehrig and Milner have explored the connection between traditional graph transformation and the bigraph approach [15, 28]. They have in particular focused on the fact that in the traditional approach, graphs are objects in a category whereas they are morphisms in the bigraphical approach. Following ideas by Cattani [8] and Sobocinsky [35] they use the cospan construction to turn objects into morphisms, and the coslice construction to turn morphisms into objects, thereby enabling transfer of results. As part of this work, Milner defines embeddings of ground link graphs and show that they are isomorphic to link graph contexts [28]; this definition serves as the basis of our bigraph embeddings.

Stochastic Simulation of Process Algebra A number of simulators for various stochastic process algebras have been developed in recent years, most notably

KaSim: A simulator for the κ-calculus based on the Gillespie-based algorithm presented in [12].

PEPA: The PEPA Eclipse Plug-in Project [30] provides an editor and various tools for PEPA [22], including a stochastic simulator.

PRISM: Though not quite a stochastic simulator – it is a *probabilistic model checker* – it supports (a subset of) PEPA models and is very efficient [26].

SPiM: The Stochastic Pi Machine is, as the name implies, a simulator for the stochastic π-calculus [31].

1.2 Outline of the Report

Though this report is self contained, it is *not* a gentle introduction to bigraphs: we assume that the reader has a keen intuition of bigraphs and bigraphical reactive systems and a reasonable grasp of its categorical underpinnings.

The remainder of the report is organized as follows:

Section 2: Background
We provide a terse recap of the theory of bigraphs, along with a few new related definitions and results that we shall need in the following sections.

Section 3: The Simulation Algorithm
We give an overview of the KaSim algorithm, recast to the setting of stochastic bigraphs.

Section 4: Stochastic Parametric Reactive Systems
We develop *stochastic parametric reactive systems* and prove that they generate the same abstract reactions as Milner's reactive systems.

Section 5: Bigraph Embeddings
We develop a general notion of *bigraph embeddings* which are isomorphic to certain decompositions of bigraphs. In particular, in the case of redexes, bigraph embeddings are isomorphic to matches.

Section 6: Bigraph Edit Scripts
We develop *bigraph edit scripts*, fine-grained reconfigurations of redexes that may be mediated by embeddings and generate the same reactions as bigraphical parametric reaction rules.

Section 7: Rule Activation and Inhibition
We outline a construction of fine-grained causality/conflict relations between parametric reaction rules based on pullbacks in the category of bigraph embeddings.

Section 8: Anchored Matching
We give a backtracking algorithm for extending partial bigraph embeddings to total embeddings, which, since embeddings of redexes into agents are matches, is equivalent to a localized matching algorithm.

Section 9: Conclusions and Future Work
We conclude and discuss future work.

2 Background

We provide the necessary background theory upon which this report builds: some basic mathematical preliminaries and a recap of the basic theory of bigraphs and bigraphical reactive systems (enriched with a few simple definitions and results).

2.1 Mathematical Preliminaries

We briefly review the basic mathematical notations and conventions used in this report.

Natural Numbers and Sets We shall frequently treat a natural number m as a finite ordinal, the set of all preceding ordinals: $m = \{0, 1, \dots, m-1\}$. We write sets as capital letters, e.g., S, T, or as a symbol with a tilde on top to denote a set of what that symbol denotes, e.g., $\tilde{m}$ is a set of natural numbers. For singletons $S = \{s\}$ we often use s and S interchangeably. We write $S - s$ and $S + s$ to mean $S \setminus \{s\}$ and $S \cup \{s\}$ respectively. We write $S \# T$ for disjoint sets, i.e., $S \cap T = \emptyset$. We write $S \uplus T$ for the union of sets known or assumed to be disjoint. We write $S + T$ for the disjoint union $\{(0, s) \mid s \in S\} \cup \{(1, t) \mid t \in T\}$, and we write $\pi_i(S_0 + S_1)$ for S_i. We use $\bar{\imath}$ to mean $1 - i$ for $i \in \{0, 1\}$.

Vectors We write vectors as a symbol with an arrow on top to denote a vector of what that symbol denotes, e.g., $\vec{m}$ is a vector of natural numbers. We write $|\vec{\cdot}|$ for the number of elements in a vector and $\vec{\cdot}_i$ $(i \in |\vec{\cdot}|)$ for the ith element of the vector (i.e., indices begin at 0). We write $\{\vec{\cdot}\}$ for the set $\{\vec{\cdot}_i \mid i \in |\vec{\cdot}|\}$.

Functions We write Id_S for the identity function on the set S. We write $\emptyset_S$ for the function whose domain and codomain are the empty set and S respectively; S is sometimes omitted when it is understood from the context. We write $\vec{f} : \vec{S} \to T$ for the vector consisting of functions $f_i : S_i \to T$ $(i \in |\vec{f}| = |\vec{S}|)$, and symmetrically $\vec{f} : S \to \vec{T}$ for the vector consisting of functions $f_i : S \to T_i$ $(i \in |\vec{f}| = |\vec{T}|)$. We write $\{s_0 \mapsto t_0, \dots, s_{n-1} \mapsto t_{n-1}\}$ for the function mapping s_i to t_i (assuming the s_i are distinct) and $\{S \mapsto t\}$ for the function that maps all elements of S to t. For a function $f : S \to T$ we write $f - s : (S - s) \to T$ for the function defined as $(f - s)(s') = f(s')$ for $s' \in S - s$, and for sets $S' = \{s_0, \dots, s_{n-1}\} \subseteq S$ we write $f - S$ to mean $(\cdots(f - s_0)\cdots - s_{n-1}$. Symmetrically, we write $f[s \mapsto t] : (S + s) \to (T + t)$ for the function defined as

$$f[s \mapsto t](s') = \begin{cases} t & \text{if } s = s' \\ f(s') & \text{otherwise} \end{cases}.$$

If $f : S \to T$ is a function and $S' \subseteq S, T' \subseteq T$, then $f \restriction_{S'}$ denotes the restriction of f to S' and $f \downharpoonleft^{T'}$ denotes the outward restriction of f to T', i.e., $f \downharpoonleft^{T'}(s) = t$ iff $f(s) \in T'$, and we write $f(S')$ to mean $\{f(s) \mid s \in S'\}$. For an injective function $f : S \rightarrowtail T$ we write $f^{-1} : \mathsf{rng}(f) \rightarrowtail S$ for its inverse which is total, injective and surjective. For a function $f : S \to T$ we write f^{-1} for its preimage function defined as $f^{-1}(t) = \{s \in S \mid f(s) = t\} : T \to \mathcal{P}(S)$, and we extend the preimage function to sets $T' \subseteq T$ as follows: $f^{-1}(T') = \{s \in S \mid f(s) \in T'\} : \mathcal{P}(T) \to \mathcal{P}(S)$. For a function $f : S \to \mathcal{P}(T)$ we shall write $f(S')$ to mean $\bigcup_{s \in S'} f(s)$ when $S' \subseteq S$, and, by extension, we shall sometimes write $\mathsf{rng}(f)$ to mean $f(S)$ when the context prevents ambiguity. We write $f \downharpoonleft^{T'}$ for the outward restriction of f to $T' \subseteq T$, i.e., $f \downharpoonleft^{T'}(s) = f(s) \cap T'$. We say that f is *fully injective* iff $\forall s, s' \in S : f(s) \cap f(s') \neq \emptyset \Rightarrow s = s'$. When f is fully injective we write $f^{-1}(t)$ $(t \in \mathsf{rng}(f))$ for the unique s for which $t \in f(s)$. Note that for a fully injective function f we have $f^{-1} : \mathsf{rng}(f) \rightarrowtail S$, $t \in f(f^{-1}(t))$, and $\{s\} = f^{-1}(f(s))$. For two functions f and g with disjoint domains S and T we write $f \uplus g$ for the function with domain $S \uplus T$ such that $(f \uplus g) \restriction_S = f$ and $(f \uplus g) \restriction_T = g$. We write $\rightarrowtail$ to indicate that a function is injective and $\rightharpoonup$ indicates partiality. We write $\hookrightarrow$ ($\hookrightarrow$) for (partial) graph embeddings.

Stochastics We use $rand(S, f)$ to denote a random variate of a stochastic variable with outcomes S and probability distribution f.

2.2 Bigraphs

This section is not meant as an introduction to bigraphs, but rather as a simplified and unified reference for the parts of the bigraphical theory that we shall need in this report. In other words, we assume the reader is already familiar with bigraphs; please refer to Milner's recent book [29] for an introduction to, and more complete treatment of, the theory of bigraphs.

Below we recall the definitions, notation, conventions, and terminology of bigraphs that we shall use in this report. We follow Milner's book closely [29], most of the time verbatim, but we have in a few places omitted details that are irrelevant for our purposes (most significantly the notions of width and sorting), slightly tweaked the notation to improve the readability of this report, and corrected some minor mistakes.

We also prove a few straightforward results that will shall need later and introduce notation for extracting subgraphs from bigraphs.

2.2.1 Concrete Bigraphs

We assume that names, node-identifiers, and edge-identifiers are drawn from three countably infinite, mutually disjoint, sets: $\mathcal{X}$, $\mathcal{V}$, and $\mathcal{E}$, respectively.

Nodes in bigraphs are assigned kinds, called *controls*, and the controls specify the number of ports nodes of that kind have:

Definition 2.1 (basic signature (after [29, Def. 1.1]))**.** A *basic signature* takes the form $(\mathcal{K}, ar)$. It has a set $\mathcal{K}$ whose elements are kinds of node called *controls*, and a map $ar : \mathcal{K} \to \mathbb{N}$ assigning an *arity*, a natural number, to each control. The signature is denoted by $\mathcal{K}$ when the arity is understood. A bigraph over $\mathcal{K}$ assigns to each node a control, whose arity indexes the *ports* of a node, where links may be connected. □

A bigraph is a pair of a place graph and a link graph, called its constituents. Bigraphs and their constituents are either *concrete* or *abstract*; in this section we are concerned with concrete bigraphs, and we shall defer the discussion of abstract bigraphs to Section 2.2.3. We denote concrete bigraphs and their constituents by upper case letters $A, \ldots, H$.

We define concrete place and link graphs separately, and then combine them into bigraphs:

Definition 2.2 (concrete place graph (after [29, Def. 2.1]))**.** A *concrete place graph*

$$F = (V_F, \mathit{ctrl}_F, \mathit{prnt}_F) : m \to n$$

is a triple having an inner face m and an outer face n, both finite ordinals. These index respectively the *sites* and *roots* of the place graph. F has a finite set $V_F \subset \mathcal{V}$ of *nodes*, a *control map* $\mathit{ctrl}_F : V_F \to \mathcal{K}$ and a *parent map*

$$\mathit{prnt}_F : m \uplus V_F \to V_F \uplus n$$

which is acyclic, i.e., if $\mathit{prnt}^i_F(v) = v$ for some $v \in V_F$ then $i = 0$. We shall call nodes, roots and sites the *places* of F.

We say that a root i is *idle* iff there is no $c \in V_F \uplus m$ with $\mathit{prnt}_F(c) = i$. A site i is *guarding* iff its parent is a node, i.e., $\mathit{prnt}_F(i) \in V_F$. Sites and nodes are *siblings* if they have the same parent. A place graph with inner face $m = 0$ is called *ground* or an *agent*. □

Definition 2.3 (concrete link graph (after [29, Def. 2.2]))**.** A *concrete link graph*

$$F = (V_F, E_F, \mathit{ctrl}_F, \mathit{link}_F) : X \to Y$$

is a quadruple having an inner face X and an outer face Y, both finite subsets of $\mathcal{X}$, called respectively the *inner* and *outer names* of the link graph. F has finite sets $V_F \subset \mathcal{V}$ of *nodes* and $E_F \subset \mathcal{E}$ of *edges*, a *control map* $\mathit{ctrl}_F : V_F \to \mathcal{K}$ and a *link map*

$$\mathit{link}_F : X \uplus P_F \to E_F \uplus Y$$

where $P_F \overset{\text{def}}{=} \{(v,i) \mid v \in V_F \wedge i \in \mathit{ar}(\mathit{ctrl}(v))\}$ is the set of *ports* of F. Thus (v,i) is the ith port of node v. We write P_{v,ctrl_F} for the ports of node v given control map ctrl_F, i.e., it denotes the set $\{(v,i) \mid i \in \mathit{ar}(\mathit{ctrl}_F(v))\}$; we extend the notation to sets of nodes, $P_{V,\mathit{ctrl}_F} \overset{\text{def}}{=} \bigcup_{v\in V} P_{v,\mathit{ctrl}_F}$, and we sometimes omit *ctrl* when it is evident from the context. We shall call $X \uplus P_F$ the *points* of F, and $E_F \uplus Y$ its *links*. We say that outer names are *open* links and edges are *closed* links.

We say that a link l is *idle* iff there is no $p \in X \uplus P_F$ with $\mathit{link}_F(p) = l$. An inner name x is *guarding* iff its link is connected to a node, i.e., $\exists (v,i) \in P_F : \mathit{link}_F(v,i) = \mathit{link}_F(x)$. Points and inner names are *siblings* if they are connected to the same link. A link graph with inner face $X = \emptyset$ is called *ground* or an *agent*. □

Definition 2.4 (concrete bigraph (after [29, Def. 2.3])). An *interface*, denoted by upper case letters I, J, K, for bigraphs is a pair $I = \langle m, X\rangle$ of a place graph interface and a link graph interface. We call m the width of I, and we say that I is *nullary*, *unary* or *multiary* according as m is 0, 1 or >1. A *concrete bigraph*

$$F = (V_F, E_F, \mathit{ctrl}_F, \mathit{prnt}_F, \mathit{link}_F) : \langle k, X\rangle \to \langle m, Y\rangle$$

consists of a concrete place graph $F^P = (V_F, \mathit{ctrl}_F, \mathit{prnt}_F) : k \to n$ and a concrete link graph $F^L = (V_F, E_F, \mathit{ctrl}_F, \mathit{link}_F) : X \to Y$. We write the concrete bigraph as $F = \langle F^P, F^L\rangle$. We shall call $V_F \uplus E_F \uplus k \uplus X \uplus m \uplus Y$ the *entities* of F.

A bigraph is called *ground* or an *agent* iff both of its constituents are ground. A bigraph is called *discrete* iff it has no edges and its link map is a bijection. A bigraph is called *prime* iff is has no inner names and a unary outer face. □

It should be clear from the above definitions, that the choice of node- and edge-identifiers have no impact on the structure of the graphs. We shall now make this precise:

Definition 2.5 (support for bigraphs (after [29, Def. 2.4])). To each place graph, link graph or bigraph F is assigned a finite set $|F|$, its *support*. For a place graph we define $|F| = V_F$, and for a link graph or bigraph we define $|F| = V_F \uplus E_F$.

For two bigraphs $F : I \to J$ and $G : I \to J$, a support translation $\rho : |F| \to |G|$ from F to G consists of a pair of bijections $\rho_V : V_F \to V_G$ and $\rho_E : E_F \to E_G$ that respect structure, in the following sense:

(i) ρ preserves controls, i.e., $\mathit{ctrl}_G \circ \rho_V = \mathit{ctrl}_F$. It follows that ρ induces a bijection $\rho_P : P_F \to P_G$ on ports, defined by $\rho_P((v,i)) \overset{\text{def}}{=} (\rho_V(v), i)$.

(ii) ρ commutes with the structural maps as follows:

$$\begin{aligned} \mathit{prnt}_G \circ (\mathsf{Id}_m \uplus \rho_V) &= (\mathsf{Id}_n \uplus \rho_V) \circ \mathit{prnt}_F \\ \mathit{link}_G \circ (\mathsf{Id}_X \uplus \rho_P) &= (\mathsf{Id}_Y \uplus \rho_E) \circ \mathit{link}_F \,. \end{aligned}$$

Given F and the bijection ρ, these conditions uniquely determine G. We therefore denote G by $\rho \bullet F$, and call it the *support translation of F by ρ*. We call F and G *support equivalent*, and we write $F \doteq G$, if such a support translation exists. Support translations $\rho : |F| \to |F|$ where $\rho \bullet F = F$ are called *support automorphisms* if they are not identities.

Support translation is defined similarly for place graphs and link graphs.

□

The interfaces of bigraphs and their constituents enable their composition: if the inner face of a bigraph is the same the outer face of another, they may be composed:

Definition 2.6 (composition and identities (after [29, Def. 2.5]))**.** We define composition for place graphs and link graphs separately, and then combine them for the composition of bigraphs.

- If $F : k \to m$ and $G : m \to n$ are two place graphs with $|F| \# |G|$, their composite

  $$G \circ F = (V, ctrl, prnt) : k \to n$$

  has nodes $V = V_F \uplus V_G$ and control map $ctrl = ctrl_F \uplus ctrl_G$. Its parent map $prnt$ is defined as follows: If $w \in k \uplus V_F \uplus V_G$ is a site or node of $G \circ F$ then

  $$prnt(w) \overset{\text{def}}{=} \begin{cases} prnt_F(w) & \text{if } w \in k \uplus V_F \text{ and } prnt_F(w) \in V_F \\ prnt_G(j) & \text{if } w \in k \uplus V_F \text{ and } prnt_F(w) = j \in m \\ prnt_G(w) & \text{if } w \in V_G. \end{cases}$$

  The identity place graph at m is $\mathsf{id}_m \overset{\text{def}}{=} (\emptyset, \emptyset, \mathsf{Id}_m) : m \to m$.

- If $F : X \to Y$ and $G : Y \to Z$ are two link graphs with $|F| \# |G|$, their composite

  $$G \circ F = (V, E, ctrl, link) : X \to Z$$

  has $V = V_F \uplus V_G$, $E = E_F \uplus E_G$, $ctrl = ctrl_F \uplus ctrl_G$, and its link map $link$ is defined as follows: If $q \in X \uplus P_F \uplus P_G$ is a point of $G \circ F$ then

  $$link(q) \overset{\text{def}}{=} \begin{cases} link_F(q) & \text{if } q \in X \uplus P_F \text{ and } link_F(q) \in E_F \\ link_G(y) & \text{if } q \in X \uplus P_F \text{ and } link_F(q) = y \in Y \\ link_G(q) & \text{if } q \in P_G. \end{cases}$$

  The identity link graph at X is $\mathsf{id}_X \overset{\text{def}}{=} (\emptyset, \emptyset, \emptyset, \mathsf{Id}_X) : X \to X$.

- If $F : I \to J$ and $G : J \to K$ are two bigraphs with $|F| \# |G|$, their composite is

  $$G \circ F \overset{\text{def}}{=} \langle G^P \circ F^P, G^L \circ F^L \rangle : I \to K$$

  and the identity bigraph at $I = \langle m, X \rangle$ is $\langle \mathsf{id}_m, \mathsf{id}_X \rangle$.

We shall often omit the composition operator and simply write GF for $G \circ F$. □

Bigraphs also have a notion of partial tensor product, called *juxtaposition*, which is defined for *disjoint graphs*:

Definition 2.7 (disjoint graphical structure (after [29, Def. 2.6]))**.** Two place graphs F_i $(i = 0, 1)$ are *disjoint* if $|F_0| \# |F_1|$. Two link graphs $F_i : X_i \to Y_i$ are *disjoint* if $X_0 \# X_1$, $Y_0 \# Y_1$ and $|F_0| \# |F_1|$. Two bigraphs F_i are *disjoint* if $F_0^P \# F_1^P$ and $F_0^L \# F_1^L$.
In each of the three cases we write $F_0 \# F_1$. □

Definition 2.8 (juxtaposition and units (after [29, Def. 2.7]))**.** We define juxtaposition for place graphs and link graphs separately, and then combine them in order to juxtapose bigraphs. In each case we indicate the obvious unit for juxtaposition.

- For place graphs, the juxtaposition of two interfaces m_i $(i = 0, 1)$ is $m_0 + m_1$ and the unit is 0. If $F_i = (V_i, ctrl_i, prnt_i) : m_i \to n_i$ are disjoint place graphs $(i = 0, 1)$, their juxtaposition $F_0 \otimes F_1 : m_0 + m_1 \to n_0 + n_1$ is given by

  $$F_0 \otimes F_1 \overset{\text{def}}{=} (V_0 \uplus V_1, ctrl_0 \uplus ctrl_1, prnt_0 \uplus prnt'_1),$$

  where $prnt'_1(m_0 + i) = n_0 + j$ whenever $prnt_1(i) = j$.

- For link graphs, the juxtaposition of two disjoint link graph interfaces is $X_0 \uplus X_1$ and the unit is $\emptyset$. If $F_i = (V_i, E_i, ctrl_i, link_i) : X_i \to Y_i$ are disjoint link graphs ($i = 0, 1$), their juxtaposition $F_0 \otimes F_1 : X_0 \uplus X_1 \to Y_0 \uplus Y_1$ is given by

$$F_0 \otimes F_1 \overset{\text{def}}{=} (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1).$$

- For bigraphs, the juxtaposition of two disjoint interfaces $I_i = \langle m_i, X_i \rangle$ ($i = 0, 1$) is $\langle m_0 + m_1, X_0 \uplus X_1 \rangle$ and the unit is $\epsilon = \langle 0, \emptyset \rangle$. If $F_i =: I_i \to J_i$ are disjoint bigraphs ($i = 0, 1$), their juxtaposition $F_0 \otimes F_1 : I_0 \otimes I_1 \to J_0 \otimes J_1$ is given by

$$F_0 \otimes F_1 \overset{\text{def}}{=} \langle F_0^P \otimes F_1^P, F_o^L \otimes F_1^L \rangle.$$

□

Notations and terminology An interface $\langle n, X \rangle$ is sometimes written as n if $X = \emptyset$ or as X if $n = 0$; hence ϵ, 0 and $\emptyset$ all denote the same interface.

We shall denote bigraphs known to be ground using small letters and we shall often omit their inner face ϵ, e.g., $g : I$.

We call bigraphs with zero width *linkings* (sometimes *wirings*) and we use λ and ω to denote them. We shall often write linkings simply as their link map, and, as instance of this convention, the empty bigraph can be denoted by $\emptyset$. We call linkings with no edges *substitutions*, denoted by σ and τ. Discrete substitutions are called *renamings*, denoted by α and β. Ground substitutions are called (name) *introductions* and we denote them by their outer face X, or just x if X is the singleton set $\{x\}$. A linking with empty outer and inner faces and a single edge e is denoted by $/_e$. Linkings with empty outer face, a single inner name x, and a single edge e are called *closures*, denoted $/_e x$; the tensor product of multiple closures $/_{e_1} x_1 \otimes \cdots \otimes /_{e_n} x_n$ is sometimes written as $/_{[e_1,\ldots,e_n]}\{x_1, \ldots, x_n\}$. We sometimes omit edge when their identity is irrelevant.

We often omit identities in compositions when there is no ambiguity, and e.g., write $\sigma \circ G$ for $(\sigma \otimes \mathsf{id}_m) \circ G$. Also, we sometimes want to apply a linking $\lambda : X \uplus Y \to Z$ to a graph $g : I \to \langle m, X \rangle$ with fewer outer names than are in the inner face of the linking; in this case we write $\lambda \circ G$ to mean $(\lambda \otimes \mathsf{id}_m) \circ (G \otimes Y)$.

Bigraphs with no nodes or links are called *placings*. We shall often write placings simply as their parent map. Placings where the parent map is a bijection are called *permutations*, denoted π.

Subtrees, Subforests, and their Contexts We shall sometimes need to extract subgraphs from bigraphs:

Definition 2.9 (subtree, subforest)**.** Given a bigraph $H : \langle n, X \rangle \to \langle m, Y \rangle$ and a node or site $c \in V_H \uplus n$. Then the *subtree* rooted at c is the set of nodes and sites defined by

$$H \downharpoonleft^{c} \overset{\text{def}}{=} \{c' \mid c' \in k_H \uplus V_H \ \wedge \ \exists i \geq 0 : prnt_H^i(c') = c\}.$$

For a set of nodes or sites $C \subseteq V_H \uplus n$ we define the *subforest* as the union of the subtrees:

$$H \downharpoonleft^{C} = \bigcup_{c \in C} H \downharpoonleft^{c} .$$

□

We shall also need the dual, i.e., the context of subgraphs:

Definition 2.10 (context graph)**.** Given a bigraph $H : \langle n, X \rangle \to \langle m, Y \rangle$ and a node or root $p \in V_H \uplus m$. Then the *context graph* at p is the set of nodes, roots, and sites defined by

$$H \upharpoonleft_{p} \overset{\text{def}}{=} (V_H \uplus n \uplus m) \setminus H \downharpoonleft^{prnt^{-1}(p)} .$$

For a set of nodes or roots $P \subseteq V_H \uplus m$ we define the *context graph* as the intersection of the individual context graphs:

$$H \restriction_P = \bigcap_{p \in P} H \restriction_p .$$

□

Subforests and context graphs possess a number of properties:

Proposition 2.11 (subforest and context graph)**.** *Given a bigraph $H : \langle n, X\rangle \to \langle m, Y\rangle$, a set of nodes or sites $c \in C \subseteq V_H \uplus n$, and a set of nodes or roots $p \in P \subseteq V_H \uplus m$. Then we have the following properties:*

1. $H \downharpoonleft^C \subseteq n \uplus V_H$,
2. $c \in H \downharpoonleft^c$,
3. $C \subseteq H \downharpoonleft^C$,
4. $\forall c' \in H \downharpoonleft^c : \exists i \geq 0 : prnt_H^i(c') = c$,
5. $H \restriction_P \subseteq n \uplus V_H \uplus m$,
6. $H \restriction_P = (V_H \uplus n \uplus m) \setminus \{c \mid c \in k \uplus V_H \ \wedge \ \exists i > 0 : prnt_H^i(c) \in P\}$
7. $p \in H \restriction_P$, *and*
8. $P \subseteq H \restriction_P$ *if* $\forall v, p \in P : \forall i > 0 : prnt_H^i(v) \neq p$.

Proof. The first 5 properties are immediate from the definitions. We prove the remaining three:

6: Expanding the definitions we get:

$$\begin{aligned}
H \restriction_p &= (V_H \uplus n \uplus m) \setminus H \downharpoonleft^{prnt^{-1}(p)} \\
&= (V_H \uplus n \uplus m) \setminus \bigcup_{c \in prnt^{-1}(p)} H \downharpoonleft^c \\
&= (V_H \uplus n \uplus m) \setminus \bigcup_{c \in prnt^{-1}(p)} \{c' \mid c' \in n \uplus V_H \ \wedge \ \exists i \geq 0 : prnt_H^i(c') = c\} \\
&= (V_H \uplus n \uplus m) \setminus \{c' \mid c' \in n \uplus V_H \ \wedge \ \exists i \geq 0 : prnt_H^i(c') \in prnt^{-1}(p)\} \\
&= (V_H \uplus n \uplus m) \setminus \{c' \mid c' \in n \uplus V_H \ \wedge \ \exists i > 0 : prnt_H^i(c') = p\}.
\end{aligned}$$

7: From (6) we can deduce $p \in H \restriction_p$ iff p is a root or if it is a node satisfying $\forall i > 0 : prnt_H^i(p) \neq p$, which is satisfied since $prnt_H$ is acyclic.

8: We must show $\forall p, p' \in P : p' \in H \restriction_p$, assuming $\forall v, p \in P : \forall i > 0 : prnt_H^i(v) \neq p$. From the proof of (7) we have $p' \in H \restriction_p$ iff p' is a root or if it is a node satisfying $\forall i > 0 : prnt_H^i(p') \neq p$ which follows from the assumption. □

Derived operations When composing and juxtapositioning bigraphs we shall often want to fuse links from the two graphs if they have the same name, and we therefore introduce two derived operations, *parallel product* $\parallel$ and *nesting* (sometimes *dotting*) $.$:

Definition 2.12 (parallel product (after [29, Def. 3.11]))**.** The *parallel product* $\parallel$ is given on interfaces by

$$\langle m, X\rangle \parallel \langle n, Y\rangle \stackrel{\text{def}}{=} \langle m+n, X \cup Y\rangle.$$

Now let $G_i : I_i \to J_i$ $(i = 0, 1)$ be two bigraphs with disjoint supports. Denote the link map of G_i by $link_i$ $(i = 0, 1)$, and assume further that $link_0 \cup link_1$ is a function. Then the parallel product

$$G_0 \parallel G_1 : I_0 \parallel I_1 \to J_0 \parallel J_1$$

is defined just as tensor product, except that its link map allows name-sharing. □

Proposition 2.13 (parallel product (after [24, Prop. 9.14]))**.** *Let $G_0 \parallel G_1$ be defined. Then*

$$G_0 \parallel G_1 = \sigma(G_0 \otimes \tau G_1),$$

where the substitutions σ and τ are defined as follows: If z_i $(i \in n)$ are the names shared between G_0 and G_1, and w_i are fresh names in bijection with the z_i, then $\tau(z_i) = w_i$ and $\sigma(w_i) = \sigma(z_i) = z_i$ $(i \in n)$.

Definition 2.14 (nesting (after [29, Def. 3.13]))**.** Let $F : I \to \langle m, X\rangle$ and $G : m \to \langle n, Y\rangle$ be bigraphs. Define the *nesting* $G.F : I \to \langle n, X \cup Y\rangle$ by:

$$G.F \stackrel{\text{def}}{=} (\mathsf{id}_X \parallel G) \circ F.$$

□

2.2.2 S-categories and spm-categories

Large parts of the theory of bigraphs is formulated at the more general level of spm categories and s-categories, the definitions of which we shall briefly recall here. We shall assume that the reader is familiar with the basic definitions of category theory. We shall use upper case bold letters to denote categories (e.g., **C**) and we presuppose an infinite repository of *support elements* $\mathcal{S}$.

Definition 2.15 (partial monoidal category (after [29, Def. 2.10]))**.** A category is said to be *partial monoidal* when it has a partial *tensor product* $\otimes$ both on objects and on arrows satisfying the following conditions.

On objects, $I \otimes J$ and $J \otimes I$ are either both defined or both undefined. The same holds for $I \otimes (J \otimes K)$ and $(I \otimes J) \otimes K$; moreover, they are equal when defined. There is a *unit* object ϵ, often called the *origin*, for which $\epsilon \otimes I = I \otimes \epsilon = I$ for all I.

On arrows, the tensor product of $f : I_0 \to I_1$ and $g : J_0 \to J_1$ is defined iff $I_0 \otimes J_0$ and $I_1 \otimes J_1$ are both defined. The following must hold when both sides are defined:

(M1) $f \otimes (g \otimes h) = (f \otimes g) \otimes h$

(M2) $\mathsf{id}_\epsilon \otimes f = f \otimes \mathsf{id}_\epsilon = f$

(M3) $(f_1 \otimes g_1) \circ (f_0 \otimes g_0) = (f_1 \circ f_0) \otimes (g_1 \circ g_0)$.

A functor of partial monoidal categories preserves unit and tensor product. □

Definition 2.16 (spm category (after [29, Def. 2.11]))**.** A partial monoidal category is *symmetric* (spm) if, whenever $I \otimes J$ is defined, there is an arrow $\gamma_{I,J} : I \otimes J \to J \otimes I$ called a *symmetry*, satisfying the following equations when the compositions and products are defined:

(S1) $\gamma_{I,\epsilon} = \mathsf{id}_I$

(S2) $\gamma_{J,I} \circ \gamma_{I,J} = \mathsf{id}_{I \otimes J}$

(S3) $\gamma_{I_1,J_1} \circ (f \otimes g) = (g \otimes f) \circ \gamma_{I_0,J_0}$ (for $f : I_0 \to I_1, g : J_0 \to J_1$)

(S4) $\gamma_{I \otimes J,K} = (\gamma_{I,K} \otimes \mathsf{id}_J) \circ (\mathsf{id}_I \otimes \gamma_{J,K})$.

A functor between spm categories preserves unit, product and symmetries. □

Definition 2.17 (precategory (after [29, Def. 2.12]))**.** A *precategory* $\grave{\mathbf{C}}$ is like a category except that composition of f and g may be undefined even when $\mathsf{cod}(f) = \mathsf{dom}(g)$. We use a tag, as in $\grave{\mathbf{C}}$, to distinguish precategories. Composition satisfies the following conditions (the first being weaker than for a category):

(C1) if $g \circ f$ is defined then $\mathsf{cod}(f) = \mathsf{dom}(g)$

(C2) $h \circ (g \circ f) = (h \circ g) \circ f$ when either is defined

(C3) $\mathsf{id} \circ f = f$ and $f = f \circ \mathsf{id}$.

We understand (C3) to imply that composition of an arrow f with the identities on its domain and codomain is always defined. □

Definition 2.18 (s-category (after [29, Defs. 2.13 & A.1]))**.** An s-category $\grave{\mathbf{C}}$ is a precategory in which each arrow f is assigned a finite *support* $|f| \subset \mathcal{S}$. Further, $\grave{\mathbf{C}}$ possesses a partial tensor product, unit and symmetries, as in an spm category. The identities id_I and symmetries $\gamma_{I,J}$ are assigned empty support. In addition:

(i) For $f : I \to J$ and $g : J' \to K$, the composition $g \circ f$ is defined iff $J = J'$ and $|f| \# |g|$; then $|g \circ f| = |f| \uplus |g|$.

(ii) For $f : I_0 \to I_1$ and $g : J_0 \to J_1$, the tensor product $f \otimes g$ is defined iff $I_i \otimes J_i$ is defined $(i = 0, 1)$ and $|f| \# |g|$; then $|f \otimes g| = |f| \uplus |g|$.

The equations (M1)–(M3) and (S1)–(S4) from Definitions 2.15 and 2.16 are required to hold when both sides are defined.

For any arrow $f : I \to J$ in an s-category $\grave{\mathbf{C}}$ and any partial injective map $\rho : \mathcal{S} \rightharpoonup \mathcal{S}$ whose domain includes $|f|$, there is an arrow $\rho \bullet f : I \to J$ called a *support translation* of f. Support translations satisfy the following equations when both sides are defined:

(T1) $\rho \bullet \mathsf{id}_I = \mathsf{id}_I$

(T2) $\rho \bullet (f \circ g) = \rho \bullet f \circ \rho \bullet g$

(T3) $\mathsf{Id}_{|f|} \bullet f = f$

(T4) $(\rho' \circ \rho) \bullet f = \rho' \bullet (\rho \bullet f)$

(T3) $\rho \bullet f = (\rho \restriction_{|f|}) \bullet f$

(T4) $|\rho \bullet f| = \rho(|f|)$

(T5) $\rho \bullet (f \otimes g) = \rho \bullet f \otimes \rho \bullet g$.

Two arrows f and g are *support-equivalent*, written $f \simeq g$, if $\rho \bullet f = g$ for some support translation ρ. The *support automorphisms* of an arrow f are the non-identity support translations $\rho : |f| \to |f|$ such that $\rho \bullet f = f$.

A functor between s-categories preserves tensor product, unit, symmetries and support equivalence. □

We shall later need a few results about s-categories:

Lemma 2.19. *Let $a \simeq c \circ r$, $c \simeq d$, and $r \simeq s$ with $|d| \# |s|$ in some s-category $\grave{\mathbf{C}}$. Then $a \simeq d \circ s$.*

Proof. By the definition of support equivalence, we have support bijections $\rho : |c \circ r| \to |a|$, $\rho' : |d| \to |c|$ and $\rho'' : |s| \to |r|$ such that $a = \rho \bullet (c \circ r)$, $c = \rho' \bullet d$ and $r = \rho'' \bullet s$. Since $c \circ r$ is defined we have and $|c| \# |r|$. Thus $\rho' \uplus \rho''$ is well-defined and we get the following equivalences:

$$\begin{aligned} a &= \rho \bullet (c \circ r) \\ &= \rho \bullet ((\rho' \bullet d) \circ (\rho'' \bullet s)) \\ &= \rho \bullet (((\rho' \uplus \rho'') \bullet d) \circ ((\rho' \uplus \rho'') \bullet s)) \\ &= (\rho \circ (\rho' \uplus \rho'')) \bullet (d \circ s) \end{aligned}$$

and thus $a \simeq d \circ s$ (using the support translation equalities of Def. 2.18). □

Lemma 2.20. *Let $a \simeq c \circ r$ and $r \simeq s$ in some s-category $\grave{\mathbf{C}}$. Then $a \simeq c' \circ s$ for some c' with $c \simeq c'$.*

Proof. Choose some bijection $\rho'' : |c| \to \mathcal{S} \setminus |s|$ (where $\mathcal{S}$ is the infinite support repository of $\grave{\mathbf{C}}$) and let $c' \stackrel{\text{def}}{=} \rho'' \bullet c$. By construction we have $c \simeq c'$ and $|c'| \# |s|$, so by Lemma 2.19 $a \simeq c' \circ s$ as required. □

Obviously, spm categories can be seen as s-categories where arrows have empty support. Conversely, we can obtain spm categories from s-categories as follows:

Definition 2.21 (support quotient (after [29, Def. 2.14])). For any s-category $\grave{\mathbf{C}}$, its *support quotient*

$$\mathbf{C} \stackrel{\text{def}}{=} \grave{\mathbf{C}} / \simeq$$

is the spm category whose objects are those of $\grave{\mathbf{C}}$, and whose arrows $[f] : I \to J$ are support-equivalence classes of the homset $\grave{\mathbf{C}}(I \to J)$. The composition of $[f] : I \to J$ with $[g] : J \to K$ is defined as $[g] \circ [f] \stackrel{\text{def}}{=} [g' \circ f']$, where $f' \in [f]$ and $g' \in [g]$ are chosen with disjoint supports.

The tensor product is defined analogously. The identities and symmetries of $\mathbf{C}$ are singleton equivalence classes since they have empty support. □

Notations and terminology In any category $\mathbf{C}$ a pair of arrows $\vec{f} : I \to \vec{J}$ is a *span*. Dually, a pair of arrows $\vec{f} : \vec{I} \to J$ is a *cospan*.

If $\vec{f} : I \to \vec{J}$ is a span and $\vec{g} : \vec{J} \to K$ is a cospan satisfying $g_0 \circ f_0 = g_1 \circ f_1$, then $\vec{g}$ is a *bound* for $\vec{f}$ and, dually, $\vec{f}$ is a *cobound* for $\vec{g}$.

2.2.3 Bigraphical Categories

Concrete bigraphs and their constituents form s-categories, and by quotienting these by (essentially) support equivalence we get spm categories of abstract bigraphs.

Definition 2.22 (graphical s-categories (after [29, Def. 2.17])). A basic signature $\mathcal{K}$ was defined in Def. 2.1. Concrete place graphs, link graphs and bigraphs over an arbitrary signature were defined in Defs. 2.2, 2.3 and 2.4. We now cast each of these kinds of graph as arrows in an s-category, denoted respectively by $\grave{\text{P}}\text{G}(\mathcal{K})$, $\grave{\text{L}}\text{G}(\mathcal{K})$ and $\grave{\text{B}}\text{G}(\mathcal{K})$.

The objects in these three s-categories are called *interfaces*, or *faces*. For place graphs they are natural numbers, for link graphs they are finite name-sets, and for bigraphs they are pairs of a natural number m and a finite name-set.

Support for the three kinds of graph was defined in Def. 2.5, with support elements $\mathcal{V} \uplus \mathcal{E}$. *Composition* and *identities* were set out in Def. 2.6, and *juxtaposition* and *units* in Def. 2.8, determining *tensor product*.

To complete our definition it remains to define *symmetries* $\gamma_{I,J}$ as follows:

$$\begin{array}{lllll} \text{in } \grave{\text{P}}\text{G}: & \gamma_{m,n} \stackrel{\text{def}}{=} (\emptyset, \emptyset, prnt), & \text{where} & prnt(i) = n + i & (i \in m) \\ & & \text{and} & prnt(m + j) = j & (j \in n) \\ \text{in } \grave{\text{L}}\text{G}: & \gamma_{X,Y} \stackrel{\text{def}}{=} \mathsf{id}_{X \uplus Y} & & & \\ \text{in } \grave{\text{B}}\text{G}: & \gamma_{\langle m,X \rangle, \langle n,Y \rangle} \stackrel{\text{def}}{=} \langle \gamma_{m,n}, \gamma_{X,Y} \rangle. & & & \end{array}$$

□

Definition 2.23 (lean, lean-support quotient, abstract bigraphs (after [29, Def. 2.19]))**.** A bigraph is *lean* if it has no idle edges. Two bigraphs F and G are *lean-support equivalent*, written $F \Bumpeq G$, if they are support-equivalent ignoring their idle edges. It is easily seen that both composition and tensor product preserve this equivalence.

For the bigraphical s-category $\grave{\ }\textsc{Bg}(\mathcal{K})$, its *lean-support quotient*

$$\textsc{Bg}(\mathcal{K}) \stackrel{\text{def}}{=} \grave{\ }\textsc{Bg}(\mathcal{K}) / \Bumpeq$$

is the spm category whose objects are those of $\grave{\ }\textsc{Bg}(\mathcal{K})$ and whose arrows $[\![G]\!] : I \to J$, called *abstract bigraphs*, are lean-support equivalence classes of the homset $(I \to J)$ in $\grave{\ }\textsc{Bg}(\mathcal{K})$. Composition, tensor product, identities and symmetries for the lean-support quotient are defined just as for support quotient in Def. 2.21.

The spm categories $\textsc{Pg}(\mathcal{K})$ of abstract place graphs and $\textsc{Lg}(\mathcal{K})$ of abstract link graphs are constructed similarly.

We shall sometimes use hat to denote that an arrow is abstract, e.g., $\hat{G}$. We shall say that G is a *concretion* of $[\![G]\!]$. □

We shall later need the following result regarding lean support equivalence:

Lemma 2.24. *Let F, G, C, D be concrete bigraphs with $G \Bumpeq F$ and $F = C \circ D$. Then there exists concrete bigraphs C', D' with $C' \Bumpeq C$ and $D' \Bumpeq D$ such that $G = C' \circ D'$.*

Proof. Let ρ be a witness of the support equivalence part of $G \Bumpeq F$. Then $\rho \bullet G$ and $F = C \circ D$ differ only in their idle edges. Construct C'' and D'' by removing the idle edges of C and D that are not in $\rho \bullet G$, and add the idle edges of $|\rho \bullet G| \setminus |F|$ to either one. Then clearly $C'' \Bumpeq C$, $D'' \Bumpeq D$ and $\rho \bullet G = C'' \circ D''$. Since support translations are bijections, we have $G = \rho^{-1} \bullet C'' \circ \rho^{-1} \bullet D''$, $C' \stackrel{\text{def}}{=} \rho^{-1} \bullet C'' \bumpeq C'' \Bumpeq C$, and $D' \stackrel{\text{def}}{=} \rho^{-1} \bullet D'' \bumpeq D'' \Bumpeq D$ as required. □

Corollary 2.25 (lean support equivalence vs support equivalence)**.** *Let F, G, C, D be concrete bigraphs with $G \Bumpeq F$ and $F \bumpeq C \circ D$. Then there exists concrete bigraphs C', D' with $C' \Bumpeq C$ and $D' \Bumpeq D$ such that $G \bumpeq C' \circ D'$.*

Furthermore, if C is lean, a lean C' exists, i.e., $C \bumpeq C'$. The same holds for D and D'. Both C' and D' can be lean iff G is lean.

Proof. Let ρ' be the witness of $F \bumpeq C \circ D$ and use the construction from the proof of Lemma 2.24 for G and $\rho' \bullet F = C \circ D$ (obviously $G \Bumpeq \rho \bullet F$), putting all the idle edges of $|\rho \bullet G| \setminus |\rho' \bullet F|$ in either C'' or D''. □

The notations, terminology and derived operations pertaining to concrete bigraphs carry over to abstract bigraphs.

2.2.4 Reactive Systems

Bigraph dynamics are defined as an instance of the more general *basic reactive systems*, where a set of ground reaction rules generates a reaction relation by closing the set under all contexts and support equivalence:[1]

[1] We avoid the added complexity of the so-called *passive* contexts, contexts that disallow reaction, from this presentation as our work can be straightforwardly extended to include such contexts. Omitting passive contexts allows us to ignore the refinement of reactive systems to *wide reactive systems* [29, Def. 7.2], which are only introduced to handle passive contexts when deriving labeled transition systems (LTSs).

Definition 2.26 (basic reactive system (BaRS) (after [29, Sec. 7.1]))**.** A *basic reactive system*, written $\grave{\mathbf{C}}(\grave{\mathcal{R}})$, consists of an s-category $\grave{\mathbf{C}}$ equipped with a set $\grave{\mathcal{R}}$ of *reaction rules.* An arrow $a : \epsilon \to I$ in $\grave{\mathbf{C}}$ with domain ϵ is a *ground arrow* or *agent*, often written $a : I$.

Each reaction rule R consists of a pair $(r : I, r' : I)$ of ground arrows, a *redex* and a *reactum.* The set $\grave{\mathcal{R}}$ must be closed under support translation, i.e., if (r, r') is a rule then so is (s, s') whenever $r \bumpeq s$ and $r' \bumpeq s'$.

The *reaction relation* $\rightarrowtriangle$ over agents is the smallest such that $a \rightarrowtriangle a'$ whenever $a \bumpeq c \circ r$ and $a' \bumpeq c \circ r'$ for some reaction rule (r, r') and context c for r and r'.

□

When the underlying s-category disregards support we say that the BaRS is abstract:

Definition 2.27 (abstract BaRS (after [29, Def. 7.3]))**.** A BaRS is *abstract* if its underlying s-category is an spm category. □

We saw in section 2.2.2 how we can construct spm categories from s-categories by quotienting by support equivalence. Similarly, we may construct abstract BaRSs from concrete ones by quotienting by support equivalence. However, as we saw in section 2.2.3, abstract bigraphs are constructed from concrete bigraphs by quotienting by a more coarse-grained equivalence: lean-support equivalence. We therefore generalize the constructions of spm categories and abstract BaRSs from concrete ones to more general equivalences, so-called *abstractions*:

Definition 2.28 (structural congruence (after [29, Def. 7.4]))**.** An equivalence relation $\equiv$ on each homset of an s-category $\grave{\mathbf{C}}$ is a *structural congruence* if it is preserved by composition and tensor product. It is called an *abstraction* if it includes support equivalence. We denote the $\equiv$-equivalence class of f by $[\![f]\!]$.

In a BaRS $\grave{\mathbf{C}}_{(\mathsf{r})}(\grave{\mathcal{R}})$ an abstraction is *dynamic* if in addition it respects reaction, i.e., if $f \rightarrowtriangle f'$ and $g \equiv f$ then $g \rightarrowtriangle g'$ for some $g' \equiv f'$. □

Clearly, support equivalence is a dynamic abstraction on any BaRS:

Proposition 2.29 (support equivalence is a dynamic abstraction)**.** *Support equivalence $\bumpeq$ is a dynamic abstraction on a BaRS $\grave{\mathbf{C}}(\grave{\mathcal{R}})$.*

We can now state the more general abstraction constructions and show that they indeed yield spm categories and abstract BaRSs:

Definition 2.30 (quotient s-category (after [29, Def. 7.5]))**.** Let $\grave{\mathbf{C}}$ be an s-category, and let $\equiv$ be an abstraction on $\grave{\mathbf{C}}$. Then

$$\mathbf{C} \stackrel{\text{def}}{=} \grave{\mathbf{C}}/\equiv$$

is the spm category whose objects are those of $\grave{\mathbf{C}}$, and whose arrows $[\![f]\!] : I \to J$ are $\equiv$-equivalence classes of the homset $I \to J$ in $\grave{\mathbf{C}}$. The composition of $[\![f]\!] : I \to J$ with $[\![g]\!] : J \to K$ is defined as $[\![g]\!] \circ [\![f]\!] \stackrel{\text{def}}{=} [\![f' \circ g']\!]$, where $f' \in [\![f]\!]$ and $g' \in [\![g]\!]$ are chosen with disjoint supports. The tensor product is defined analogously. The identities and symmetries in $\mathbf{C}$ are necessarily the equivalence classes of their $\grave{\mathbf{C}}$ counterparts. □

Lemma 2.31 (quotient s-category)**.** *Let $\grave{\mathbf{C}}$ be an s-category, and let $\equiv$ be an abstraction on $\grave{\mathbf{C}}$. Then the quotient $\mathbf{C} = \grave{\mathbf{C}}/\equiv$ is an spm category. Its construction defines a functor of s-categories*

$$[\![\cdot]\!] : \grave{\mathbf{C}} \to \mathbf{C}.$$

Proof. $\mathbf{C}$ is an spm category, since it inherits the tensor product, unit, and symmetries of the s-category $\grave{\mathbf{C}}$, and the construction eliminates the partiality of composition and tensor product. $[\![\cdot]\!]$ is a functor by construction and it is between s-categories because any spm category is an s-category with empty supports. □

Definition 2.32 (quotient BaRS (after [29, Def. 7.6]))**.** Let $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ be a BaRS, and $\equiv$ a dynamic abstraction on $\grave{\mathbf{C}}$. Then define $\mathbf{C}(\mathcal{R})$, the quotient of $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ by $\equiv$, as follows:

- $\mathbf{C} = \grave{\mathbf{C}}/\equiv$, and
- $\mathcal{R} = \{(\llbracket r \rrbracket, \llbracket r' \rrbracket) \mid (r, r') \in \grave{\mathcal{R}}\}$.

□

Theorem 2.33 (abstract BaRS (after [29, Thm. 7.7]))**.** *The construction of Def. 2.30 and Def. 2.32, applied to a concrete BaRS* $\grave{\mathbf{C}}(\grave{\mathcal{R}})$*, yields an abstract BaRS* $\mathbf{C}(\mathcal{R})$*, whose underlying spm category* $\mathbf{C}$ *is the codomain of a functor of s-categories*

$$\llbracket\cdot\rrbracket : \grave{\mathbf{C}} \to \mathbf{C}.$$

Moreover the construction preserves the reaction relation, in the following sense:

1. *if* $f \longrightarrow\!\!\!\!\triangleright\, f'$ *in* $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ *then* $\llbracket f \rrbracket \longrightarrow\!\!\!\!\triangleright\, \llbracket f' \rrbracket$ *in* $\mathbf{C}(\mathcal{R})$
2. *if* $\llbracket f \rrbracket \longrightarrow\!\!\!\!\triangleright\, g'$ *in* $\mathbf{C}(\mathcal{R})$ *then* $f \longrightarrow\!\!\!\!\triangleright\, f'$ *in* $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ *for some* f' *with* $\llbracket f' \rrbracket = g'$*.*

Proof. The first part follows immediately from Lemma 2.31 and Def. 2.27.

1: We have $f \doteq c \circ r$ and $f' \doteq c \circ r'$ for some reaction rule (r, r') and context c for r and r'. We must show $\llbracket f \rrbracket \longrightarrow\!\!\!\!\triangleright\, \llbracket f' \rrbracket$, i.e., $\llbracket f \rrbracket \doteq \llbracket c' \rrbracket \circ \llbracket s \rrbracket$ and $\llbracket f' \rrbracket \doteq \llbracket c' \rrbracket \circ \llbracket s' \rrbracket$ for some reaction rule (s, s') and context $\llbracket c' \rrbracket$ for $\llbracket s \rrbracket$ and $\llbracket s' \rrbracket$.

Letting $c' \overset{\text{def}}{=} c$, $s \overset{\text{def}}{=} r$, $s' \overset{\text{def}}{=} r'$ and exploiting that $\equiv$ includes support equivalence and that $\llbracket\cdot\rrbracket$ is a functor, we get $(\llbracket s \rrbracket, \llbracket s' \rrbracket) = (\llbracket r \rrbracket, \llbracket r' \rrbracket) \in \mathcal{R}$, $\llbracket f \rrbracket = \llbracket c \circ r \rrbracket = \llbracket c' \circ s \rrbracket = \llbracket c' \rrbracket \circ \llbracket s \rrbracket$, and $\llbracket f' \rrbracket = \llbracket c \circ r' \rrbracket = \llbracket c' \circ s' \rrbracket = \llbracket c' \rrbracket \circ \llbracket s' \rrbracket$ as required.

2: We have $\llbracket f \rrbracket \doteq \llbracket c \rrbracket \circ \llbracket r \rrbracket$ and $g' \doteq \llbracket c \rrbracket \circ \llbracket r' \rrbracket$ for some reaction rule (r, r') and context $\llbracket c \rrbracket$ for $\llbracket r \rrbracket$ and $\llbracket r' \rrbracket$. We must show $f \longrightarrow\!\!\!\!\triangleright\, f'$ for some f' with $\llbracket f' \rrbracket = g'$, i.e., $f \doteq c' \circ s$ and $f' \doteq c' \circ s'$ for some reaction rule (s, s') and context c' for s and s'.

Let $s \overset{\text{def}}{=} r$, $s' \overset{\text{def}}{=} r'$ and choose some context $c' \in \llbracket c \rrbracket$ with $c' \# r$ and $c' \# r'$. By definition of composition in $\mathbf{C}$ and since abstraction includes support equivalence, we get $\llbracket f \rrbracket = \llbracket c \rrbracket \circ \llbracket r \rrbracket = \llbracket c' \circ r \rrbracket$ and $g' = \llbracket c \rrbracket \circ \llbracket r' \rrbracket = \llbracket c' \circ r' \rrbracket$. By Def. 2.26 we have $f \equiv c' \circ r \longrightarrow\!\!\!\!\triangleright\, c' \circ r'$ and since $\equiv$ is dynamic, there is some $f' \equiv c' \circ r' \in g'$ such that $f \longrightarrow\!\!\!\!\triangleright\, f'$ as required. □

2.2.5 Bigraphical Reactive Systems

While the rules of basic reactive systems are required to be ground, bigraphical reaction rules are allowed to take parameters. But by viewing such parametric reaction rules as generators of ground reaction rules, we can view bigraphical reactive systems as a sugared variant of basic reactive systems.

Before we define bigraphical parametric reaction rules and bigraphical reactive systems, let us first define what a parameter is and how a parametric reaction rule is allowed to manipulate it through *instantiation.*

First, note that ground bigraphs can be seen as the juxtaposition of a number of discrete primes bound together by some linking:

Corollary 2.34 (ground discrete normal form (DNF) (after [29, Corol. 3.10]))**.** *A ground bigraph* $g : \langle n, Z \rangle$ *can be expressed uniquely, up to renaming on* Y*, as* $g = (\mathsf{id}_n \otimes \lambda) \circ (d_0 \otimes \cdots \otimes d_{n-1})$*, where* $\lambda : Y \to Z$ *is a linking and the* d_i *are discrete primes.*

We shall regard the individual primes of a ground bigraph as parameters of reaction and shall allow each of them to be copied, discarded, or left unchanged using *instantiation*:

Definition 2.35 (instantiation (after [29, Def. 8.5]))**.** In a bigraphical s-category $`\mathbf{C} = `\text{BG}(\mathcal{K})$, let $\eta : n \to m$ be a map of finite ordinals. Define the *instance* function family $\bar{\eta}_{X,S} : `\mathbf{C}(\epsilon, \langle m, X\rangle) \to `\mathbf{C}(\epsilon, \langle n, X\rangle)$, indexed by name set X and support set S, on agents as follows: Given an agent $g : \langle m, X\rangle$, find its DNF $g = \lambda \circ (d_0 \otimes \cdots \otimes d_{m-1})$ (Corol. 2.34). Then

$$\bar{\eta}_{X,S}(g) \overset{\text{def}}{=} \lambda \circ (d'_0 \parallel \cdots \parallel d'_{n-1})$$

where $d'_j \simeq d_{\eta(j)}$ and $|d'_j| \# S$ for each $j \in n$. The function is defined up to $\simeq$.

We shall often omit X and/or S when they are evident from the context. □

We have reformulated Milner's definition to index $\bar{\eta}$ by X and S; X was already a somewhat implicit index whereas S is a technical measure that will allow us to ensure that instantiation chooses fresh support with respect to the context in which it will be used.

Note that $\bar{\eta}_{X,S}(g)$ has the same outer names as g and that linking commutes with instantiation:

Proposition 2.36 (linking an instance (after [29, Def. 8.4]))**.** *Linking commutes with instantiation; that is,* $\omega \circ \bar{\eta}_{X,S}(g) \simeq \bar{\eta}_{X,S}(\omega \circ g)$.

Let us now define parametric reaction rules for bigraphs and how they generate ground reaction rules:

Definition 2.37 (bigraphical parametric reaction rules (after [29, Def. 8.5]))**.** A *parametric reaction rule* R for bigraphs is a triple of the form

$$(R : m \to J, R' : m' \to J, \eta)$$

where R is the *parametric redex*, R' the *parametric reactum*, and $\eta : m' \to m$ a map of finite ordinals. The rule generates all ground reaction rules (r, r'), where

$$r \simeq R.d, \quad r' \simeq R'.\bar{\eta}(d)$$

and $d : \langle m, Y\rangle$ is discrete. □

With this definition in mind, it is clear that the following definition of bigraphical reactive systems is an instance of the basic reactive systems defined above:

Definition 2.38 (bigraphical reactive system (BRS) (after [29, Def. 8.6]))**.** A *(concrete) bigraphical reactive system (BRS)* over $\mathcal{K}$ consists of $`\text{BG}(\mathcal{K})$ equipped with a set $`\mathcal{R}$ of parametric reaction rules closed under support equivalence; that is, if $R \simeq S$ and $R' \simeq S'$ and $`\mathcal{R}$ contains $(R, R', \eta)$, then it also contains $(S, S', \eta)$. We denote the BRS by $`\text{BG}(\mathcal{K}, `\mathcal{R})$. □

Comparing the definition of BaRSs with the generation of ground bigraphical reaction rules from parametric ones, it is clear that reactions are generated by occurrences of redexes, what we call *matches*:

Definition 2.39 (match)**.** Given a parametric reaction rule $\mathsf{R} = (R, R', \eta)$, agent a, and bigraphs c,d, we say that (c, d) is a *match* of R in a iff $a \simeq c \circ R.d$. Two matches (c, d),(c', d') are regarded as the same if they differ only by a bijection on the outer faces of d and d'; otherwise they are *distinct*. We write $\mathit{match}(a, \mathsf{R})$ for the set of distinct matches of R in a. □

We may construct abstract BRSs by quotienting by lean-support equivalence $\mathrel{\hat{\simeq}}$ since it is a dynamic abstraction on BRSs:

Proposition 2.40 (lean-support equivalence is a dynamic abstraction)**.** *Lean-support equivalence* $\mathrel{\hat{\simeq}}$ *is a dynamic abstraction on a BRS* $`\text{BG}(\mathcal{K}, `\mathcal{R})$.

Proof. It is immediate from its definition that lean-support equivalence is an abstraction, so we just need to check that it is dynamic.

Assume bigraphs a, a', b with $a \longrightarrow a'$ and $b \mathrel{\hat{\simeq}} a$. Since $a \longrightarrow a'$ we must have $a \simeq c \circ r$ and $a' \simeq c \circ r'$ for some rule (r, r') and context c. By Corollary 2.25, noting that bigraph rules are lean, there are bigraphs d, s with $d \mathrel{\hat{\simeq}} c$ and $s \simeq r$ such that $b \simeq d \circ s$. Since rules are closed under support translation, there is a rule (s, s') with $s' \simeq r'$, and we thus have the reaction $b \simeq d \circ s \longrightarrow b'$ where $b' = d \circ s'$. Since lean-support equivalence is preserved by composition we have $b' = d \circ s' \mathrel{\hat{\simeq}} c \circ r' \simeq a'$ as required. □

3 The Simulation Algorithm

We give an overview of the simulation algorithm for the κ-calculus by Danos et al. [12] recast to stochastic bigraphs. We shall refer to the algorithm in loc. cit. as KaSim. This reformulation of KaSim to stochastic bigraphs is independent of the physical and stochastic underpinnings of the algorithm, so we shall not concern ourselves with the details of these matters; the interested reader may refer to [12, 18, 19].

3.1 Gillespie's algorithm

KaSim is a generalization of what is known as Gillespie's algorithm, an algorithm for stochastic simulation of coupled chemical reactions [18, 19]. It is based on the idea of assigning probabilities to reaction rules which are proportional to the number of instances of each rule in the current state of the system, and letting the frequency of reaction be proportional to the total number of rule instances.

Recast to bigraphs, the algorithm in overview works as follows: given a set of reaction rules $\mathcal{R}$, with each reaction rule R assigned a rate constant ϱ_{R}, an agent a, and a simulation time t_{stop}, perform the following steps:

0. Initialization:

Initialize the simulation state:

$t := 0$	current simulation time, initially 0
$M(a, \mathsf{R}) := \mathit{match}(a, \mathsf{R})$	set of matches of R's redex in a ($\forall \mathsf{R} \in \mathcal{R}$)
$\alpha_{\mathsf{R}} := \lvert M(a, \mathsf{R})\rvert \times \varrho_{\mathsf{R}}$	activity of R
$\alpha := \Sigma_{\mathsf{R}\in\mathcal{R}}\alpha_{\mathsf{R}}$	system activity

If $\alpha = 0$ then no reaction is possible and the simulation ends.

1. Monte Carlo step:

Sample the following random values:

$\mathsf{R} := \mathit{rand}(\mathcal{R}, \lambda\mathsf{R}.\frac{\lvert M(a,\mathsf{R})\rvert \times \varrho_{\mathsf{R}}}{\alpha})$	rule to be applied
$\phi := \mathit{rand}(M(a, \mathsf{R}), \lambda m.1/\lvert M(a, \mathsf{R})\rvert)$	match to be applied
$\delta t := \mathit{rand}(\mathbb{R}^+, \lambda t.\alpha e^{\alpha t})$	time advance

2. Update:

Update the simulation state:

$a := a'$, if $a \rightarrowtriangle_{\mathsf{R},\phi} a'$	perform reaction
$t := t + \delta t$	advance time
$M(a, \mathsf{R}) := \mathit{match}(a, \mathsf{R})$	update sets of matches ($\forall \mathsf{R} \in \mathcal{R}$)
$\alpha_{\mathsf{R}} := \lvert M(a, \mathsf{R})\rvert \times \varrho_{\mathsf{R}}$	update rule activities
$\alpha := \Sigma_{\mathsf{R}\in\mathcal{R}}\alpha_{\mathsf{R}}$	update system activity

3. Iterate:

If $t > t_{\text{stop}}$ or $\alpha = 0$, stop; otherwise repeat from step 1.

Figure 1 illustrates the simulation loop.

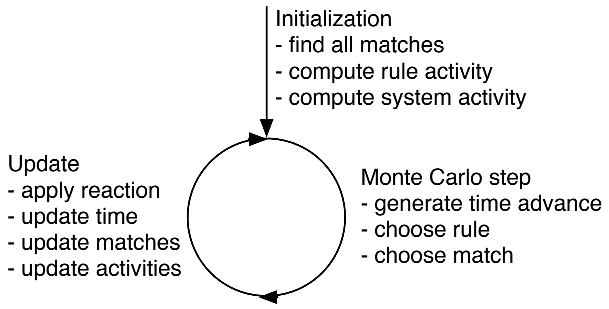

Figure 1: The basic simulation loop.

3.2 Incremental and Local Updates

It should be clear that the update step as expressed above does not scale: it requires recomputation of all matches at each simulation cycle. Instead, KaSim employs an incremental update phase where (i) matches are only removed from $M(a, \mathsf{R})$ if they are invalidated by the reaction and (ii) matches are only searched for in the parts of the agent that were affected by the reaction. Thus, the update phase actually consists of three steps[2]:

2a. Negative update:

Remove matches that will be invalidated by the chosen reaction and decrease activities accordingly.

2b. Rewrite:

Rewrite the agent using the chosen rule and match.

2c. Positive update:

Find new matches created by the reaction and increment activities accordingly.

These steps presume that we can determine conflict and causality in an efficient manner: we must be able to quickly identify (2a) the reactions that are in conflict with the chosen reaction and (2c) the reactions that it causes. This is achieved in KaSim by (i) assuming that rules are enriched with a notion of *modification* which characterizes how reaction modifies the redex, and (ii) assuming the existence of two relations, called the inhibition and activation maps, that characterize the interplay between rules:

inhibition: We say that rule R_0 *inhibits* rule R_1, written $\mathsf{R}_0 \# \mathsf{R}_1$, iff there is some agent a and embeddings $\phi_i : R_i \hookrightarrow a$ such that $\mathsf{cod}(\phi_0) \cap \mathsf{cod}(\phi_1)$ contains at least one entity modified by R_0.

activation: Rule R_0 *activates* rule R_1, written $\mathsf{R}_0 \prec \mathsf{R}_1$, iff there is some agent a and embeddings $\phi_0 : \Delta_0(R_0) \hookrightarrow a$, $\phi_1 : R_1 \hookrightarrow a$ such that $\mathsf{cod}(\phi_0) \cap \mathsf{cod}(\phi_1)$ contains at least one entity modified by R_0.

Note that these relations are not necessarily symmetric.

Assuming we can construct these relations during initialization, we may express the negative and positive update steps in more detail as follows (recall that R is the chosen rule, ϕ the chosen match, and that step 2b. sets $a := a'$):

2a. Negative update:

For each R' with $\mathsf{R} \# \mathsf{R}'$

[2]As a technicality, we have swapped steps 2a. and 2b. as it in the case of SBAM leads to a more direct implementation.

(i) remove the embeddings $\phi' : R' \hookrightarrow a$ from $M(a, \mathsf{R}')$ for which some elements of $\mathsf{rng}(\phi) \cap \mathsf{rng}(\phi')$ will be modified by the chosen reaction, and

(ii) decrease the system activity and the activity of the rule by the number of removed embeddings times $\varrho_{\mathsf{R}'}$.

2c. Positive update:

For each R' with $\mathsf{R} \prec \mathsf{R}'$

(i) add new embeddings $\phi' : R' \hookrightarrow a$ to $M(a, \mathsf{R}')$. At least one element of $\mathsf{rng}(\phi')$ must be modified by the reaction in order for ϕ' to be new, and

(ii) increase the system activity and the activity of the rule by the number of added embeddings times $\varrho_{\mathsf{R}'}$.

Thus, using this approach we avoid considering rules that are known to never generate reactions that are causally related to those of the chosen rule. But even in the worst case, $\prec = \# = \mathcal{R} \times \mathcal{R}$, this approach is an improvement since we have restricted the part of the agent that we need to consider.

Let us consider steps 2a(i) and 2c(i) in a bit more detail:

2a(i): Though we have restricted the set of embeddings we need to consider, it is still unclear how to efficiently identify the affected embeddings. The approach in KaSim is actually to not use the inhibition relation, but instead maintain a so-called *lift* map $l : |a| \to M(a, \cdot)$ from entities in the agent to the embeddings that have those entities in their co-domain. While less space-efficient, it enables us to quickly remove invalidated embeddings.

2c(i): The notion of modification that rules are enriched with, allows us to determine which entities in the agent have been modified, so we can perform localized matching as follows: for each modified entity $e \in |a|$ and entity $e' \in |R'|$ such that $[e' \mapsto e]$ is a partial embedding, attempt to extend it to a complete embedding of R'.

By extending we mean incrementally adding mappings $f \mapsto f'$ of entities $f \in |a| \setminus \mathsf{rng}(\phi')$, $f' \in |R'| \setminus \mathsf{dom}(\phi')$ which are adjacent to elements of $\mathsf{rng}(\phi')$ and $\mathsf{dom}(\phi')$ respectively. We call this *anchored matching*.

In κ, anchored matching is deterministic and there is at most one complete extension of a partial embedding. This is not generally the case for bigraphs.

Note that anchored matching only yields complete embeddings for redexes consisting of one connected component and the KaSim algorithm is actually slightly more complicated than what we have sketched above, as it handles redexes with more than one connected component. However, the KaSim approach to handling such redexes transfer unaltered to the bigraph version of the algorithm, so in this report we shall simply assume that redexes consist of exactly one connected component.

4 Stochastic Parametric Reactive Systems

Concrete bigraphs are a means to constructing a tractable behavioral theory for abstract bigraphs: Abstract bigraphs could be defined directly instead of being derived from concrete bigraphs[3]. However, abstract bigraphs have insufficient structure for constructing minimal transition labels which is a key construction in the behavioral theory of BRSs. By defining abstract bigraphs in terms of concrete bigraphs, where such minimal labels can be constructed, one gets the means for obtaining minimal labels for abstract bigraphs.

The dynamic theory of bigraphs have been designed with this construction in mind, which is reflected in the treatment of support : (a) sets of rules are required to be closed under support translation (cf. Def. 2.26 and Def. 2.38), (b) the construction of the reaction relation closes under support translation (cf. Def. 2.26), and (c) the construction of ground reaction rules from parametric ones closes under support translation (cf. Def. 2.37). In other words, these constructions are aimed at support equivalence classes of concrete bigraphs, i.e., abstract bigraphs.

While this approach is sufficient for most applications of abstract bigraphs, it is insufficient in the context of stochastic bigraphs, where support provides the means for counting matches: closing under support translation would lead to an infinite number of matches for non-trivial redexes. In their definition of stochastic bigraphs [25], Krivine et al. solve this issue by (a) replacing support equivalence by equality in the definition of a match (cf. Def. 2.39)[4], and (b) defining the number of matches in an abstract bigraph as the number of matches in one of its concretions. However, in loc. cit. the definition of a match is not (directly) related to the definition of the reaction relation, resulting in a conceptual gap between the usual reaction semantics and the stochastic reaction semantics. A unified presentation of these two aspects of stochastic BRSs would promote understanding.

Another issue is the infinite set of ground reaction rules that a parametric reaction rule generate. Clearly, we cannot represent these explicitly in an implementation, so we cannot implement BRSs directly as stated in Def. 2.26 and Def. 2.38. We believe that direct representations in implementations increase trust in correctness, and it would therefore be desirable if we could give a directly representable definition of BRSs with parametric reaction rules.

In this section we tackle both of these issues, by developing a variant of reactive systems, which we call *stochastic parametric reactive systems*, that have none of the above shortcomings while giving rise to the same abstract reaction relation. The idea is to prevent arbitrary support translation during reaction by restricting the use of support translation to the identification of matches of redexes in an agent.

Specifically, we incrementally develop the following kinds of reactive systems:

representative basic reactive systems (RBaRS):
Almost as BaRSs but different in two respects:

- the set of rules must not contain support equivalent rules and
- reaction cannot change the support of the context.

parametric reactive systems (PRS):
A generalization of RBaRSs where parametric reaction rules are first class citizens. Reaction is refined further by restricting the manipulation of support in parameters.

stochastic parametric reactive systems (SPRS):
PRSs equipped with a stochastic semantics, which for bigraphs generalizes the stochastic semantics of Krivine et al. [25]. The reaction relation is refined such that a match determines a reaction.

[3]Milner's algebra for abstract bigraphs could be one such definition [27].

[4]Krivine et al. use the term *occurrence* and use a slightly different definition, cf. [25, Def. 4.1 and Def. 4.2], but this is an insignificant technicality.

$$\begin{tikzcd}[row sep=large]
\text{BaRS} \arrow[d, bend right, "\text{choose rule representatives}"'] \arrow[d, phantom, "\equiv"] \\
\text{RBaRS} \arrow[u, bend right, "\text{close rule set under support translation}"'] \arrow[d, bend right, "\text{extend rules with the instantiation } \mathsf{id}_{\epsilon\to\epsilon}"'] \arrow[d, phantom, "\approx_{\rightarrowtriangle}/\equiv"] \\
\text{PRS} \arrow[u, bend right, "\text{generate ground reaction rules}"'] \arrow[d, bend right, "\text{extend rules with a rate constant}"'] \arrow[d, phantom, "\approx_{\rightarrowtriangle}/\equiv"] \\
\text{SPRS} \arrow[u, bend right, "\text{discard rate constants}"']
\end{tikzcd}$$

Figure 2: The four kinds of reactive systems and how to transform one into another. The topmost two are equivalent modulo dynamic abstraction $\equiv$ (e.g., $\bumpeq$) while the others have the same reaction relation modulo dynamic abstraction.

The concrete reaction relation becomes smaller for each increment, while the abstract reaction relation remains the same. To prove this, we for each of these systems show that its concrete reaction relation is closely related to that of its predecessor, indeed so closely that it becomes immediate that they have the same abstract reactions. Figure 2 gives an overview of the relations between the four kinds of reactive systems.

4.1 Representative Basic Reactive Systems

In this section we show how one may limit Milner's liberal use of support equivalence in the definition of basic reactive systems, while maintaining the same dynamic behavior. We do so in two steps: (1) first we show that we need not require the set of reaction rules to be closed under support translation, and then (2) we reduce the reaction relation to preclude support translation in the context.

Recall from Def. 2.26 that in a BaRS

- the set of reaction rules $\grave{\mathcal{R}}$ must be closed under support translation, i.e., if (r, r') is a rule then so is (s, s') whenever $r \bumpeq s$ and $r' \bumpeq s'$, and
- the reaction relation is also closed under support translation, since $a \rightarrowtriangle a'$ whenever $a \bumpeq c \circ r$ and $a' \bumpeq c \circ r'$ for some reaction rule $(r, r') \in \grave{\mathcal{R}}$.

Since the construction of the reaction relation closes under support translation, we need only consider one concretion, a *representative*, of an abstract rule in order to generate all the corresponding reactions:

Proposition 4.1 (reaction rule representatives are sufficient)**.** *Let $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ be a BaRS and let $a \rightarrowtriangle a'$ be a reaction generated by rule (r, r'). Then any other support equivalent rule (s, s'), i.e., $r \bumpeq s$ and $r' \bumpeq s'$, generates the same reaction.*

Proof. We have $a \bumpeq c \circ r$ and $a' \bumpeq c \circ r'$ for some context c for r and r', and must show that there is a context c' for s and s' such that $a \bumpeq c' \circ s$ and $a' \bumpeq c' \circ s'$. This follows from the proof of Lemma 2.20 if we strengthen the requirement on the choice of c' by also precluding the support of s', i.e., $\rho'' : |c| \to \mathcal{S} \setminus (|s| \cup |s'|)$. □

Thus we need not close the reaction rule set under support translation. But the reaction relation is still somewhat unmanageable, since it is closed under support translation, and one wonders why we must be able to change the support of the context of a reaction? Indeed, this does not strictly increase the number of reactions, in a sense that we shall now make precise.

First, note that the reaction relation is indeed closed under support equivalence:

Lemma 4.2. *Let* $`\mathbf{C}(`\mathcal{R})$ *be a BaRS and let* $a \rightarrowtriangle a'$. *Then* $\forall b, b' : b \bumpeq a \wedge b' \bumpeq a' \Rightarrow b \rightarrowtriangle b'$.

Proof. We have $a \bumpeq c \circ r$ and $a' \bumpeq c \circ r'$ for some rule (r, r') and context c. Since $b \bumpeq a \bumpeq c \circ r$ and $b' \bumpeq a' \bumpeq c \circ r'$ we also get $b \rightarrowtriangle b'$. □

Based on the above observations, it should be clear that a BaRS contains support equivalence classes of rules and reactions; this naturally leads to a notion of a *representative* BaRS:

Definition 4.3 (representative basic reactive system (RBaRS))**.** A *representative basic reactive system*, written $`\mathbf{C}_{\mathsf{r}}(`\mathcal{R})$, consists of an s-category $`\mathbf{C}$ equipped with a set $`\mathcal{R}$ of *reaction rules.*

The *reaction relation* $\rightarrowtriangle$ over agents is the smallest such that $a \rightarrowtriangle a'$ whenever $a = c \circ \rho \mathbin{\scriptstyle\blacksquare} r$ and $a' = c \circ \rho' \mathbin{\scriptstyle\blacksquare} r'$ for some reaction rule (r, r'), support translations ρ, ρ', and context c for $\rho \mathbin{\scriptstyle\blacksquare} r$ and $\rho' \mathbin{\scriptstyle\blacksquare} r'$. □

The intuition is that an RBaRS represents a BaRS by allowing us to single out representatives for each equivalence class of rules and reactions:

Definition 4.4 (RBaRS corresponding to BaRS)**.** Let $`\mathbf{C}(`\mathcal{R})$ be a BaRS. Then the RBaRS *corresponding* to $`\mathbf{C}(`\mathcal{R})$ is $`\mathbf{C}_{\mathsf{r}}(`\mathcal{R}_r)$, where $`\mathcal{R}_r$ contains a single chosen representative of each support equivalence class of rules, i.e.,

$$\begin{gathered}\forall (r, r') \in `\mathcal{R} : \exists (s, s') \in `\mathcal{R}_r : r \bumpeq s \wedge r' \bumpeq s' \\ \forall (r, r'), (s, s') \in `\mathcal{R}_r : r \bumpeq s \wedge r' \bumpeq s' \Rightarrow r = s \wedge r' = s'.\end{gathered}$$

□

Proposition 4.5 (RBaRS corresponding to BaRS)**.** *The RBaRS corresponding to a BaRS is indeed an RBaRS.*

Conversely, we can easily construct a BaRS from an RBaRS:

Definition 4.6 (BaRS corresponding to RBaRS)**.** Let $`\mathbf{C}_{\mathsf{r}}(`\mathcal{R})$ be an RBaRS. Then the BaRS *corresponding* to $`\mathbf{C}_{\mathsf{r}}(`\mathcal{R})$ is $`\mathbf{C}(`\mathcal{R}_*)$, where $`\mathcal{R}_*$ is the support equivalence closure of $`\mathcal{R}$, i.e., $`\mathcal{R}_* = \{(s, s') \mid (r, r') \in `\mathcal{R} \wedge r \bumpeq s \wedge r' \bumpeq s'\}$. □

Proposition 4.7 (BaRS corresponding to RBaRS)**.** *The BaRS corresponding to an RBaRS is indeed a BaRS.*

Note that this construction is inverse to the previous one:

Lemma 4.8. *For any BaRS* $`\mathbf{C}(`\mathcal{R})$, *the BaRS* $`\mathbf{C}((`\mathcal{R}_r)_*)$ *obtained through Def. 4.4 followed by Def. 4.6 is the same, i.e.,* $`\mathbf{C}(`\mathcal{R}) = `\mathbf{C}((`\mathcal{R}_r)_*)$.

Proof. Immediate from the definitions. □

It is immediate from the definitions, that reactions in an RBaRS are indeed reactions in the corresponding BaRS:

Proposition 4.9 (Representative Reactions are Reactions)**.** *Let* $\rightarrowtriangle_r$ *and* $\rightarrowtriangle_f$ *denote the reaction relations of a RBaRS and its corresponding BaRS respectively. Then*

$$a \rightarrowtriangle_r a' \quad \Rightarrow \quad a \rightarrowtriangle_f a'.$$

It is also clear that RBaRSs in general have a smaller reaction relations than their corresponding BaRS, since they do not allow reaction to change the support of the context. But this has a very limited impact: any series of reactions in a BaRS can be matched by a series of reactions followed by a single support translation in an RBaRS:

Proposition 4.10 (Representative Reactions are Sufficient)**.** *Let $\rightarrowtriangle_r$ and $\rightarrowtriangle_f$ denote the reaction relations of an RBaRS $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$ and its corresponding BaRS respectively. Then*

$$\forall n \in \mathbb{N}: \quad a \rightarrowtriangle_f^n a' \quad \Rightarrow \quad \exists a'' : a \rightarrowtriangle_r^n a'' \wedge a' \bumpeq a''.$$

Proof. By induction on n, the base case being trivial. In the induction case we have $a \rightarrowtriangle_f^{n-1} b \rightarrowtriangle_f a'$, $a \rightarrowtriangle_r^{n-1} c$, and $b \bumpeq c$, and must show $c \rightarrowtriangle_r a''$ and $a' \bumpeq a''$ for some a''.

From $b \rightarrowtriangle_f a'$ we get $b \bumpeq d \circ s$ and $a' \bumpeq d \circ s'$ for some rule $(s, s') \in \grave{\mathcal{R}}_*$, i.e., $b \bumpeq d \circ \rho \blacksquare r$ and $a' \bumpeq d \circ \rho' \blacksquare r'$ for some rule $(r, r') \in \grave{\mathcal{R}}$ and support translations ρ, ρ'. Since $c \bumpeq b \bumpeq d \circ \rho \blacksquare r$, we have $c = \rho'' \blacksquare (d \circ \rho \blacksquare r) = (\rho'' \blacksquare d) \circ ((\rho'' \circ \rho) \blacksquare r)$ for some support translation ρ''. We then choose any support translation ρ''' such that $a'' \stackrel{\text{def}}{=} (\rho'' \blacksquare d) \circ (\rho''' \blacksquare r')$ is defined and thus get $c \rightarrowtriangle_r a''$. By Lemma 2.19 a'' is support equivalent to a': $a' \bumpeq d \circ \rho' \blacksquare r' \bumpeq (\rho'' \blacksquare d) \circ (\rho''' \blacksquare r') = a''$. □

4.1.1 Abstract Representative Basic Reactive Systems

Let us now show that, once we abstract identities away, the reaction relations of RBaRS and BaRS are the same. The construction and properties of abstract BaRSs from Section 2.2.4 transfer unchanged to RBaRS, so we shall not repeat them here. We shall use $\mathbf{C}_{\mathsf{r}}(\mathcal{R})$ to denote the abstract RBaRS obtained as the quotient of an RBaRS $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$ by a dynamic abstraction.

An RBaRS has the same abstract reactions as its corresponding BaRS:

Theorem 4.11 (abstract RBaRSs are abstract BaRSs)**.** *Let $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$ be an RBaRS and let $\grave{\mathbf{C}}(\grave{\mathcal{R}}_*)$ be the corresponding BaRS. Then the quotient RBaRS $\mathbf{C}_{\mathsf{r}}(\mathcal{R})$ and quotient BaRS $\mathbf{C}(\mathcal{R}_*)$, both obtained using the construction of Def. 2.30 and Def. 2.32, are the same.*

Proof. Given that both $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$ and $\grave{\mathbf{C}}(\grave{\mathcal{R}}_*)$ have the same underlying s-category, the underlying spm categories of the quotients are also the same. Also, since abstraction includes support equivalence, we have $\mathcal{R} = \mathcal{R}_*$.

We now show that the reaction relations are also the same. Let $\rightarrowtriangle_f$, $\rightarrowtriangle_{[\![f]\!]}$, $\rightarrowtriangle_r$, and $\rightarrowtriangle_{[\![r]\!]}$ denote the reaction relations of $\grave{\mathbf{C}}(\mathcal{R})$, $\mathbf{C}(\mathcal{R})$, $\grave{\mathbf{C}}_{\mathsf{r}}(\mathcal{R})$, and $\mathbf{C}_{\mathsf{r}}(\mathcal{R})$ respectively.

$\rightarrowtriangle_{[\![f]\!]} \subseteq \rightarrowtriangle_{[\![r]\!]}$: Assume $[\![f]\!] \rightarrowtriangle_{[\![f]\!]} [\![f']\!]$. From Theorem 2.33 we have $f \rightarrowtriangle_f g'$ for some $g' \in [\![f']\!]$, and Prop. 4.10 then gives us $f \rightarrowtriangle_r h'$ for some $h' \bumpeq g'$. Since abstraction includes support equivalence we have $[\![h']\!] = [\![g']\!] = [\![f']\!]$, and Theorem 2.33 gives us $[\![f]\!] \rightarrowtriangle_{[\![r]\!]} [\![h']\!]$, we have $[\![f]\!] \rightarrowtriangle_{[\![r]\!]} [\![f']\!]$ as required.

$\rightarrowtriangle_{[\![r]\!]} \subseteq \rightarrowtriangle_{[\![f]\!]}$: Assume $[\![f]\!] \rightarrowtriangle_{[\![r]\!]} [\![f']\!]$. From Theorem 2.33 we have $f \rightarrowtriangle_r g'$ for some $g' \in [\![f']\!]$, and Prop. 4.9 then gives us $f \rightarrowtriangle_f g'$. Finally, Theorem 2.33 gives us $[\![f]\!] \rightarrowtriangle_{[\![f]\!]} [\![g']\!] = [\![f']\!]$ as required. □

Conversely, a BaRS has the same abstract reactions as its corresponding RBaRS:

Theorem 4.12 (abstract BaRSs are abstract RBaRSs)**.** *Let $\grave{\mathbf{C}}(\grave{\mathcal{R}})$ be a BaRS and let $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}}_r)$ be the corresponding RBaRS. Then the quotient BaRS $\mathbf{C}(\mathcal{R})$ and quotient RBaRS $\mathbf{C}_{\mathsf{r}}(\mathcal{R}_r)$, both obtained using the construction of Def. 2.30 and Def. 2.32, are the same.*

Proof. By Theorem 4.11, $\mathbf{C}_{\mathsf{r}}(\mathcal{R}_r)$ and $\mathbf{C}((\mathcal{R}_r)_*)$ are the same. The latter is the quotient of $\grave{\mathbf{C}}((\grave{\mathcal{R}}_r)_*)$, which, by Lemma 4.8, is the same as $\grave{\mathbf{C}}(\grave{\mathcal{R}})$, and thus $\mathbf{C}_{\mathsf{r}}(\mathcal{R}_r) = \mathbf{C}_{\mathsf{r}}((\mathcal{R}_r)_*) = \mathbf{C}(\mathcal{R})$. □

4.2 Parametric Reactive Systems

Having tamed the use of support equivalence in BaRSs, we now turn our attention to the rule set blow-up caused by treating parametric reaction rules as generators of ground reaction rules. To avoid this blow-up, we generalize RBaRSs to parametric reactive systems (PRSs) where parametric reaction rules are first-class citizens.

Definition 4.13 (parametric reactive systems (PRS))**.** A *parametric reactive system*, written $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$, consists of an s-category $\grave{\mathbf{C}}$ equipped with a set $\grave{\mathcal{R}}$ of parametric reaction rules, and two subcategories $\grave{\mathbf{I}}$ and $\grave{\mathbf{D}}$ of identities and parameters respectively. $\grave{\mathbf{C}}$ and $\grave{\mathbf{D}}$ must be closed under support translation.

A *parametric reaction rule* is a triple of the form

$$(R : I \to J, R' : I' \to J, \bar{\eta}_{J',S})$$

where R is the *parametric redex*, R' the *parametric reactum*, and the *instance function* is a function family $\bar{\eta}_{J' \in \grave{\mathbf{I}}, S \subset \mathcal{S}} : \grave{\mathbf{D}}(\epsilon, I \otimes J') \to \grave{\mathbf{D}}(\epsilon, I' \otimes J')$ defined for all finite S and whenever $I \otimes J'$ is defined.

Furthermore, instantiation maps must respect support equivalence and choose sufficiently fresh support, i.e.,

1. $d \bumpeq d' \Rightarrow \bar{\eta}_{J',S}(d) \bumpeq \bar{\eta}_{J',S'}(d')$ for any finite $S, S' \subset \mathcal{S}$, and
2. $|\bar{\eta}_{J',S}(d)| \# S$.

We shall often omit J' and/or S when they are evident from the context.

The *reaction relation* $\rightarrowtriangle$ over agents $a, a' \in \grave{\mathbf{C}}(\epsilon, \cdot)$ is the smallest such that $a \rightarrowtriangle a'$ whenever $a = c \circ (\rho \mathbin{\scriptscriptstyle\blacksquare} R \otimes \mathsf{id}_{J'}) \circ d$ and $a' = c \circ (\rho' \mathbin{\scriptscriptstyle\blacksquare} R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|c| \cup \mathsf{rng}(\rho')}(d)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \bar{\eta})$, support translations ρ, ρ' with $\mathsf{dom}(\rho) = |R|$ and $\mathsf{dom}(\rho') = |R'|$, context c for $\rho \mathbin{\scriptscriptstyle\blacksquare} R \otimes \mathsf{id}_{J'}$ and $\rho' \mathbin{\scriptscriptstyle\blacksquare} R' \otimes \mathsf{id}_{J'}$, parameter $d \in \grave{\mathbf{D}}(\epsilon, I \otimes J')$, and identity $\mathsf{id}_{J'} \in \grave{\mathbf{I}}$. □

To some extent, one could argue that we have simply moved the infinitude to the instance function. However, in the case of bigraphs the instance function is finitely representable, cf. Def. 2.37, so bigraphical PRSs with finite sets of reaction rules can be directly and finitely represented in an implementation.

Another important difference, when we compare this definition to Def. 4.3, and in particular the definition of the reaction relation, is that we have factored the parameter d out of the ground redex r. One would perhaps expect the equations to simply read $a = c \circ \rho \mathbin{\scriptscriptstyle\blacksquare} R \circ d$ and $a' = c \circ \rho' \mathbin{\scriptscriptstyle\blacksquare} R' \circ \bar{\eta}(d)$ – what is the purpose of the identities? The answer is that this enables parameter and context to be connected without the involvement of the redex. To illustrate this, let us examine how BRSs can be expressed as PRSs.

4.2.1 Bigraphical Parametric Reactive Systems

In bigraphs, context and parameter may share links (and nothing else) without the involvement of the redex. To see that this is the case, let us examine Milner's generation of ground rules from parametric ones, cf. Def. 2.37: It relies on the bigraph specific nesting operator '.' (Def. 2.14) which is derived from the parallel product '$\|$' (Def. 2.12) which again can be seen as derived from the tensor product '$\otimes$' . Unfolding the nestings and applying Prop. 2.13, we obtain

$$\begin{aligned} R.d &= (R \parallel \mathsf{id}_X) \circ d & &= \sigma(R \otimes \tau \circ \mathsf{id}_X) \circ d \\ R'.\bar{\eta}(d) &= (R' \parallel \mathsf{id}_X) \circ \bar{\eta}(d) & &= \sigma(R' \otimes \tau \circ \mathsf{id}_X) \circ \bar{\eta}(d) \end{aligned}$$

for a suitable bijection $\tau : X \to X'$ and a substitution σ that ensure definedness as well as the aliasing of the shared names between R and d.

Ignoring τ, this resembles an instance of our PRS reactions, namely in the case where the context is a substitution that aliases some of the links of the parameter and redex. But what about τ?

The purpose of τ is to rename the links of the parameter such that the tensor product is defined. But the names of the parameter are internal to the reaction, as they only serve as mediators in the decomposition of the agent into context, redex, and parameter. To see this, let us rewrite $R.d$ a bit more:

$$\begin{aligned} R.d &= (R \otimes \tau \circ \mathsf{id}_X) \circ d && \text{by def. of '.' and Prop. 2.13} \\ &= \sigma(R \circ \mathsf{id}_m \otimes \mathsf{id}_{X'} \circ \tau) \circ d && \text{by def. of identities} \\ &= \sigma(R \otimes \mathsf{id}_{X'}) \circ (\mathsf{id}_m \otimes \tau) \circ d && \text{since } \otimes \text{ is a functor} \end{aligned}$$

Since d is discrete so is $(\mathsf{id}_m \otimes \tau) \circ d$ and thus $R.d$ corresponds to the left hand side of a parametric reaction.

We shall now make this precise by defining a bigraphical PRS and showing that the ground bigraphical reaction rules generated from parametric ones are reactions in that PRS:

Definition 4.14 (bigraphical parametric reactive system (BPRS))**.** A *bigraphical parametric reactive system* over $\mathcal{K}$ with bigraphical parametric reaction rules $\grave{\mathcal{R}}$, written $\grave{\mathrm{B}}\mathrm{G}(\mathcal{K}, \grave{\mathcal{R}})$, is the parametric reactive system $\grave{\mathrm{B}}\mathrm{G}(\mathcal{K})(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ where $\grave{\mathbf{D}}$ consists of the discrete ground bigraphs of $\grave{\mathrm{B}}\mathrm{G}(\mathcal{K})$ and $\grave{\mathbf{I}}$ consists of the zero-width (i.e., link graph) identities. For each rule $(R : m \to J, R' : m' \to J, \eta) \in \grave{\mathcal{R}}$ we interpret η as the corresponding instance function family $\bar{\eta}_{X,S}$ as given in Def. 2.35. □

Proposition 4.15. *Let $\grave{\mathrm{B}}\mathrm{G}(\mathcal{K}, \grave{\mathcal{R}})$ be a BPRS. Then $R.d \longrightarrow R'.\bar{\eta}(d)$ is a reaction in the BPRS for any parametric rule $(R : m \to J, R' : m' \to J, \eta) \in \grave{\mathcal{R}}$ and discrete parameter $d : \langle m, Y\rangle$.*

Proof. By unfolding the derived operators in the left and right hand sides of the claimed reaction, and then rewriting them according to the categorical axioms, it becomes clear that it is indeed a reaction:

$$\begin{aligned} R.d &= \sigma(R \otimes \tau \circ \mathsf{id}_X) \circ d && \text{by def. of '.' and Prop. 2.13} \\ &= \sigma((\mathsf{Id}_{|R|} \mathbin{\blacksquare} R) \circ \mathsf{id}_m \otimes \mathsf{id}_{X'} \circ \tau) \circ d && \text{by def. of identities} \\ &= \sigma((\mathsf{Id}_{|R|} \mathbin{\blacksquare} R) \otimes \mathsf{id}_{X'}) \circ (\mathsf{id}_m \otimes \tau) \circ d && \text{since } \otimes \text{ is a functor} \end{aligned}$$

$$\begin{aligned} R'.\bar{\eta}(d) &= \sigma(R' \otimes \tau \circ \mathsf{id}_X) \circ \bar{\eta}(d) && \text{by def. of '.' and Prop. 2.13} \\ &= \sigma((\mathsf{Id}_{|R'|} \mathbin{\blacksquare} R') \circ \mathsf{id}_{m'} \otimes \mathsf{id}_{X'} \circ \tau) \circ \bar{\eta}(d) && \text{by def. of identities} \\ &= \sigma((\mathsf{Id}_{|R'|} \mathbin{\blacksquare} R') \otimes \mathsf{id}_{X'}) \circ (\mathsf{id}_{m'} \otimes \tau) \circ \bar{\eta}(d) && \text{since } \otimes \text{ is a functor} \\ &= \sigma((\mathsf{Id}_{|R'|} \mathbin{\blacksquare} R') \otimes \mathsf{id}_{X'}) \circ \bar{\eta}((\mathsf{id}_m \otimes \tau) \circ d) && \text{by Prop. 2.36} \end{aligned}$$

Note that σ and τ are validly chosen to be the same in both cases, since R and R' have the same outer names and ditto for d and $\bar{\eta}(d)$. □

Note that Prop. 4.15 only covers the ground reaction rule $(R.d, R'.\bar{\eta}(d))$, though $(R : m \to J, R' : m' \to J, \eta)$ generates all rules on the form $(\rho \mathbin{\blacksquare} (R.d), \rho' \mathbin{\blacksquare} (R'.\bar{\eta}(d)))$. This is because our definition of PRSs does include arbitrary support translation of parameters in the reaction relation, just as it was the case for RBaRSs. We could distinguish between PRSs and *representative* PRSs, analogously to the distinction between BaRSs and RBaRS, in which case all the generated ground reaction rules would be reactions in the PRS but not the representative PRS. However, we leave this as an exercise to the reader as we shall not need this distinction.

4.2.2 Relating Concrete PRSs and RBaRSs

Having demonstrated the crux of the correspondence between BRSs and BPRSs, let us now return to the general case of RBaRSs and PRSs. First, note that a RBaRS is a PRS in a very straightforward sense:

Definition 4.16 (PRS corresponding to a RBaRS)**.** Let $\grave{}\mathbf{C}_{\mathsf{r}}(\grave{}\mathcal{R})$ be a RBaRS. Then the *corresponding* PRS is $\grave{}\mathbf{C}(\grave{}\mathcal{R} \times \{\mathsf{id}_{\epsilon \to \epsilon}\}, \mathbf{1}, \mathbf{1})$ (where $\epsilon \in \grave{}\mathbf{C}$ is the singleton object of $\mathbf{1}$). □

Proposition 4.17 (PRS corresponding to a RBaRS)**.** *The PRS corresponding to an RBaRS is indeed a PRS.*

Dually, it is straightforward to derive a RBaRS from a PRS by simply generating ground reaction rules:

Definition 4.18 (RBaRS corresponding to a PRS)**.** Let $\grave{}\mathbf{C}(\grave{}\mathcal{R}, \grave{}\mathbf{D}, \grave{}\mathbf{I})$ be a PRS. Then the corresponding RBaRS is $\grave{}\mathbf{C}_{\mathsf{r}}(\grave{}\mathcal{R}')$ where $\grave{}\mathcal{R}'$ is generated from $\grave{}\mathcal{R}$ as follows: $(r, r') \in \grave{}\mathcal{R}'$ whenever $r = (R \otimes \mathsf{id}_{J'}) \circ d$ and $r' = (R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|R'|}(d)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \bar{\eta})$, parameter $d \in \grave{}\mathbf{D}(\epsilon, I \otimes J')$, and identity $\mathsf{id}_{J'} \in \grave{}\mathbf{I}$. □

Proposition 4.19 (RBaRS corresponding to a PRS)**.** *The RBaRS corresponding to a PRS is indeed an RBaRS.*

Note that the first of these constructions is the inverse of the second:

Lemma 4.20. *For any RBaRS $\grave{}\mathbf{C}_{\mathsf{r}}(\grave{}\mathcal{R})$, the RBaRS $\grave{}\mathbf{C}_{\mathsf{r}}(\grave{}\mathcal{R}')$ obtained through Def. 4.16 followed by Def. 4.18 is the same.*

Proof. Immediate from the definitions. □

From these definitions, it is no surprise that PRS reactions are also reactions in the corresponding RBaRS:

Proposition 4.21 (Parametric Reactions are Representative Reactions)**.** *Let $\longrightarrow_p$ and $\longrightarrow_r$ denote the reaction relations of a PRS $\grave{}\mathbf{C}(\grave{}\mathcal{R}, \grave{}\mathbf{D}, \grave{}\mathbf{I})$ and its corresponding RBaRS, respectively. Then*

$$a \longrightarrow_p a' \quad \Rightarrow \quad a \longrightarrow_r a'.$$

Proof. Assume $a \longrightarrow_p a'$, i.e., $a = c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d$ and $a' = c \circ (\rho' \bullet R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}(d)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \bar{\eta})$, support translations ρ, ρ', context c for $\rho \bullet R \otimes \mathsf{id}_{J'}$ and $\rho' \bullet R' \otimes \mathsf{id}_{J'}$, parameter $d \in \grave{}\mathbf{D}(\epsilon, I \otimes J')$, and identity $\mathsf{id}_{J'} \in \grave{}\mathbf{I}$.

By Def. 4.18 $((R \otimes \mathsf{id}_{J'}) \circ d, (R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|R'|}(d))$ is a rule in the corresponding RBaRS, and so, by Def. 4.3, $c \circ (\rho \uplus \mathsf{id}_{|d|}) \bullet ((R \otimes \mathsf{Id}_{J'}) \circ d) \longrightarrow_r c \circ (\rho' \uplus \rho'') \bullet ((R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|R'|}(d))$, where ρ'' is a witness of $\bar{\eta}_{|c| \cup \mathsf{rng}(\rho')}(d) \doteq \bar{\eta}_{|R'|}(d)$. Applying the definition of support translation it is easy to see that this is indeed $a \longrightarrow_r a'$. □

As we saw in the case of bigraphs, the reaction relation of a PRS will be smaller than that of its corresponding RBaRS, since RBaRSs allow support translation of parameters. But the PRS reaction relation characterizes that of the corresponding RBaRS, similar to how the reactions of an RBaRS characterizes the reactions of the corresponding BaRS, cf. Prop. 4.10: any series of RBaRS reactions can be matched by a series of PRS reactions followed by a single support translation:

Proposition 4.22 (Parametric Reactions are Sufficient)**.** *Let $\longrightarrow_p$ and $\longrightarrow_r$ denote the reaction relations of a PRS $\grave{}\mathbf{C}(\grave{}\mathcal{R}, \grave{}\mathbf{D}, \grave{}\mathbf{I})$ and its corresponding RBaRS, respectively. Then*

$$\forall n \in \mathbb{N} : \quad a \longrightarrow_r^n a' \quad \Rightarrow \quad \exists a'' : a \longrightarrow_p^n a'' \wedge a' \doteq a''.$$

Proof. By induction on n, the base case being trivial. In the induction case we have $a \longrightarrow_r^{n-1} b \longrightarrow_r a'$, $a \longrightarrow_p^{n-1} c$, and $b \doteq c$, and must show $c \longrightarrow_p a''$ and $a' \doteq a''$ for some a''. Let $\rho'' : |b| \to |c|$ be a witness of $b \doteq c$.

From $b \longrightarrow_r a'$ we get $b = e \circ \rho \bullet r$ and $a' = e \circ \rho' \bullet r'$ for some reaction rule (r, r'), support translations ρ, ρ', and context e for $\rho \bullet r$ and $\rho' \bullet r'$. By Def. 4.18 we must have $r = (R \otimes \mathsf{id}_{J'}) \circ d$ and

$r' = (R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|R'|}(d)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \bar{\eta})$, parameter $d \in {}^{`}\mathbf{D}(\epsilon, I \otimes J')$, and identity $\mathsf{id}_{J'} \in {}^{`}\mathbf{I}$.

We therefore get the following equalities for c:

$$\begin{aligned}
c &= \rho'' \blacksquare b && \rho'' \text{ is a witness of } b \bumpeq c\\
&= \rho'' \blacksquare (e \circ \rho \blacksquare r) && b \text{ is the LHS of a representative reaction}\\
&= \rho'' \blacksquare (e \circ \rho \blacksquare ((R \otimes \mathsf{id}_{J'}) \circ d)) && (r, r') \text{ generated from parametric rule}\\
&= \rho'' \blacksquare e \circ ((\rho'' \circ \rho) \blacksquare R \otimes \mathsf{id}_{J'}) \circ ((\rho'' \circ \rho) \blacksquare d) && \text{by def. of supp. trans.}
\end{aligned}$$

Letting $a'' = \rho'' \blacksquare e \circ (\rho' \blacksquare R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|\rho'' \blacksquare e| \cup |\rho' \blacksquare R'|}((\rho'' \circ \rho) \blacksquare d)$, Def. 4.13 gives us $c \rightarrowtriangle_p a''$, and $a' \bumpeq a''$ as witnessed by $\rho''' = \rho'' \restriction_{|e|} \uplus \mathsf{Id}_{|\rho' \blacksquare R'|} \uplus \rho'''' \circ (\rho' \restriction_{|\bar{\eta}(d)|})^{-1}$, where ρ'''' is a witness of $\bar{\eta}_{|R'|}(d) \bumpeq \bar{\eta}_{|\rho'' \blacksquare e| \cup |\rho' \blacksquare R'|}((\rho'' \circ \rho) \blacksquare d)$:

$$\begin{aligned}
\rho''' \blacksquare a' &= \rho''' \blacksquare (e \circ \rho' \blacksquare r') && a \text{ is the RHS of a representative reaction}\\
&= \rho''' \blacksquare (e \circ \rho' \blacksquare ((R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|R'|}(d))) && (r, r') \text{ generated from parametric rule}\\
&= \rho'' \blacksquare e \circ (\rho' \blacksquare R' \otimes \mathsf{id}_{J'}) \circ \rho'''' \blacksquare \bar{\eta}_{|R'|}(d) && \text{by def. of supp. trans.}\\
&= \rho'' \blacksquare e \circ (\rho' \blacksquare R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|\rho'' \blacksquare e| \cup |\rho' \blacksquare R'|}((\rho'' \circ \rho) \blacksquare d) && \text{since } \bar{\eta} \text{ respects supp. eq.}
\end{aligned}$$

□

4.2.3 Abstract Parametric Reactive Systems

As we have seen above, the difference between the reaction relations of a PRS and its corresponding RBaRS is that the latter includes support translation of parameters. So we expect that if we quotient these systems with an abstraction, as in Section 4.1.1, we get the same abstract reactive systems. This is indeed the case, as we shall show below.

But first, we must extend the abstraction constructions and results of Section 2.2.4 to PRSs:

Definition 4.23 (abstract PRS)**.** A PRS is *abstract* if its underlying s-category is an spm category. □

Since we shall need to abstract instantiation maps, we must require abstractions to be well-behaved with respect to these:

Definition 4.24 (dynamic PRS abstraction)**.** In a PRS ${}^{`}\mathbf{C}({}^{`}\mathcal{R}, {}^{`}\mathbf{D}, {}^{`}\mathbf{I})$, an abstraction $\equiv$, as defined in Def. 2.28, is *dynamic* if it respects reaction and instantiation, i.e.,

1. if $f \rightarrowtriangle f'$ and $g \equiv f$ then $g \rightarrowtriangle g'$ for some $g' \equiv f'$, and
2. if $(R : I \to J, R', \bar{\eta}) \in {}^{`}\mathcal{R}$, $d, d' \in {}^{`}\mathbf{D}(\epsilon, I \otimes J')$ and $d \equiv d'$ then $\bar{\eta}_S(d) \equiv \bar{\eta}_{S'}(d')$.

□

Clearly, support equivalence is a dynamic abstraction on any PRS:

Proposition 4.25 (support equivalence is a dynamic PRS abstraction)**.** *Support equivalence* $\bumpeq$ *is a dynamic abstraction on a PRS* ${}^{`}\mathbf{C}({}^{`}\mathcal{R}, {}^{`}\mathbf{D}, {}^{`}\mathbf{I})$.

More importantly, the lean-support equivalence of bigraphs is a dynamic abstraction on BPRSs:

Proposition 4.26 (lean-support equivalence is a dynamic BPRS abstraction)**.** *Lean-support equivalence* $\Bumpeq$ *is a dynamic abstraction on a bigraphical PRS* ${}^{`}\mathrm{B}\textsc{g}(\mathcal{K}, {}^{`}\mathcal{R})$.

Proof. The proof is similar to that of Prop. 2.40 only more tedious. □

We can now define how to obtain abstract PRSs by quotienting by dynamic abstractions:

Definition 4.27 (quotient PRS)**.** Let $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be a PRS, and $\equiv$ a dynamic abstraction on $\grave{\mathbf{C}}$. Then define $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$, the quotient of $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ by $\equiv$, as follows:

- $\mathbf{C} = \grave{\mathbf{C}}/\equiv$,
- $\mathbf{D} = \grave{\mathbf{D}}/\equiv$,
- $\mathbf{I} = \grave{\mathbf{I}}/\equiv$, and
- $\mathcal{R} = \{(\llbracket R \rrbracket, \llbracket R' \rrbracket, \overline{\eta}) \mid (R, R', \overline{\eta}) \in \grave{\mathcal{R}}\}$.

We define $\overline{\eta}_S(\llbracket d \rrbracket) \overset{\text{def}}{=} \llbracket \overline{\eta}_{S'}(d) \rrbracket$ whenever $\overline{\eta}_{S'}(d)$ is defined; this is unambiguous since $\equiv$ is dynamic. $\square$

Theorem 4.28 (abstract PRS)**.** *The construction of Def. 2.30 and Def. 4.27, applied to a concrete PRS $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$, yields an abstract PRS $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$, whose underlying spm category $\mathbf{C}$ is the codomain of a functor of s-categories*

$$\llbracket \cdot \rrbracket : \grave{\mathbf{C}} \to \mathbf{C}.$$

Moreover the construction preserves the reaction relation, in the following sense:

1. *if $f \rightarrowtriangle f'$ in $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ then $\llbracket f \rrbracket \rightarrowtriangle \llbracket f' \rrbracket$ in $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$*
2. *if $\llbracket f \rrbracket \rightarrowtriangle g'$ in $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$ then $f \rightarrowtriangle f'$ in $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ for some f' with $\llbracket f' \rrbracket = g'$.*

Proof. The first part follows immediately from Lemma 2.31 and Def. 4.23.

1: We have $f = c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d$ and $f' = c \circ (\rho' \bullet R' \otimes \mathsf{id}_{J'}) \circ \overline{\eta}_{|c| \cup \mathsf{rng}(\rho')}(d)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \overline{\eta})$, support translations ρ, ρ' with $\mathsf{dom}(\rho) = |R|$ and $\mathsf{dom}(\rho') = |R'|$, context c for $\rho \bullet R \otimes \mathsf{id}_{J'}$ and $\rho' \bullet R' \otimes \mathsf{id}_{J'}$, parameter $d \in \grave{\mathbf{D}}(\epsilon, I \otimes J')$, and identity $\mathsf{id}_{J'} \in \grave{\mathbf{I}}$.

Quotienting f and f' we get $\llbracket f \rrbracket = \llbracket c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d \rrbracket = \llbracket c \rrbracket \circ (\llbracket R \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \llbracket d \rrbracket$ and $\llbracket f' \rrbracket = \llbracket c \circ (\rho' \bullet R' \otimes \mathsf{id}_{J'}) \circ \overline{\eta}_{|c| \cup \mathsf{rng}(\rho')}(d) \rrbracket = \llbracket c \rrbracket \circ (\llbracket R' \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \overline{\eta}_{\emptyset}(\llbracket d \rrbracket)$, since $\llbracket \cdot \rrbracket$ is a functor, abstraction includes support equivalence, and $\overline{\eta}_{\emptyset}(\llbracket d \rrbracket) = \llbracket \overline{\eta}_{|c| \cup \mathsf{rng}(\rho')}(d) \rrbracket$. Thus, $\llbracket f \rrbracket \rightarrowtriangle \llbracket f' \rrbracket$ as required.

2: We have $\llbracket f \rrbracket = \llbracket c \rrbracket \circ (\llbracket R \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \llbracket d \rrbracket$ and $g' = \llbracket c \rrbracket \circ (\llbracket R' \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \overline{\eta}_{\emptyset}(\llbracket d \rrbracket)$ for some parametric reaction rule $(R : I \to J, R' : I' \to J, \overline{\eta})$, context $\llbracket c \rrbracket$ for $\llbracket R \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket$ and $\llbracket R' \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket$, parameter $\llbracket d \rrbracket \in \mathbf{D}(\epsilon, I \otimes J')$, and identity $\llbracket \mathsf{id}_{J'} \rrbracket \in \mathbf{I}$ (we disregard the support translations ρ, ρ' since spm categories are s-categories with empty support).

Choose some context $c' \in \llbracket c \rrbracket$ and parameter $d' \in \llbracket d \rrbracket$ with supports that are disjoint from each other and from $|R|, |R'|$. By definition of composition in $\mathbf{C}$ and since $\overline{\eta}_{\emptyset}(\llbracket d \rrbracket) = \llbracket \overline{\eta}_{|c'| \cup |R'|}(d') \rrbracket$, we get $\llbracket f \rrbracket = \llbracket c \rrbracket \circ (\llbracket R \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \llbracket d \rrbracket = \llbracket c' \circ (R \otimes \mathsf{id}_{J'}) \circ d' \rrbracket$ and $g' = \llbracket c \rrbracket \circ (\llbracket R' \rrbracket \otimes \llbracket \mathsf{id}_{J'} \rrbracket) \circ \overline{\eta}_{\emptyset}(\llbracket d \rrbracket) = \llbracket c' \circ (R' \otimes \mathsf{id}_{J'}) \circ \overline{\eta}_{|c'| \cup |R'|}(d') \rrbracket$. By Def. 4.13 we have $f \equiv c' \circ (R \otimes \mathsf{id}_{J'}) \circ d' \rightarrowtriangle c' \circ (R' \otimes \mathsf{id}_{J'}) \circ \overline{\eta}_{|c'| \cup |R'|}(d')$ and since $\equiv$ is dynamic, there is some $f' \equiv c' \circ (R' \otimes \mathsf{id}_{J'}) \circ \overline{\eta}_{|c'| \cup |R'|}(d') \in g'$ such that $f \rightarrowtriangle f'$ as required. $\square$

Now we are ready for the main results of this section, which states that a PRS has the same abstract behavior as its corresponding RBaRS and vice versa:

Theorem 4.29 (abstract PRSs are abstract RBaRSs)**.** *Let $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be a PRS and let $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}}')$ be the corresponding RBaRS. Then the quotient PRS $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$ and quotient RBaRS $\mathbf{C}_{\mathsf{r}}(\mathcal{R}')$ have the same reaction relations.*

Proof. Let $\rightarrowtriangle_r$, $\rightarrowtriangle_{\llbracket r \rrbracket}$, $\rightarrowtriangle_p$, and $\rightarrowtriangle_{\llbracket p \rrbracket}$ denote the reaction relations of $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}}')$, $\mathbf{C}_{\mathsf{r}}(\mathcal{R}')$, $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$, and $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$ respectively.

$\rightarrowtriangle_{\llbracket r \rrbracket} \subseteq \rightarrowtriangle_{\llbracket p \rrbracket}$: Assume $\llbracket f \rrbracket \rightarrowtriangle_{\llbracket r \rrbracket} \llbracket f' \rrbracket$. From Theorem 2.33 we have $f \rightarrowtriangle_r g'$ for some $g' \in \llbracket f' \rrbracket$, and Prop. 4.22 then gives us $f \rightarrowtriangle_p h'$ for some $h' \doteq g'$. Since abstraction includes support equivalence we have $\llbracket h' \rrbracket = \llbracket g' \rrbracket = \llbracket f' \rrbracket$, and Theorem 4.28 gives us $\llbracket f \rrbracket \rightarrowtriangle_{\llbracket p \rrbracket} \llbracket h' \rrbracket$, we have $\llbracket f \rrbracket \rightarrowtriangle_{\llbracket p \rrbracket} \llbracket f' \rrbracket$ as required.

$\rightarrowtriangle_{\llbracket p \rrbracket} \subseteq \rightarrowtriangle_{\llbracket r \rrbracket}$: Assume $\llbracket f \rrbracket \rightarrowtriangle_{\llbracket p \rrbracket} \llbracket f' \rrbracket$. From Theorem 4.28 we have $f \rightarrowtriangle_p g'$ for some $g' \in \llbracket f' \rrbracket$, and Prop. 4.21 then gives us $f \rightarrowtriangle_r g'$. Finally, Theorem 2.33 gives us $\llbracket f \rrbracket \rightarrowtriangle_{\llbracket r \rrbracket} \llbracket g' \rrbracket = \llbracket f' \rrbracket$ as required. $\square$

Conversely, an RBaRS has the same abstract reactions as its corresponding PRS:

Theorem 4.30 (abstract RBaRSs are abstract PRSs)**.** *Let* $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$ *be an RBaRS and let* $\grave{\mathbf{C}}(\grave{\mathcal{R}} \times \{\mathsf{id}_{\epsilon\to\epsilon}\}, \mathbf{1}, \mathbf{1})$ *be the corresponding PRS. Then the quotient RBaRS* $\mathbf{C}_{\mathsf{r}}(\mathcal{R})$ *and quotient PRS* $\mathbf{C}(\mathcal{R} \times \{\mathsf{id}_{\epsilon\to\epsilon}\}, \mathbf{1}, \mathbf{1})$ *are the same.*

Proof. By Theorem 4.29, $\mathbf{C}(\mathcal{R} \times \{\mathsf{id}_{\epsilon\to\epsilon}\}, \mathbf{1}, \mathbf{1})$ is equal to the quotient $\mathbf{C}_{\mathsf{r}}(\mathcal{R}')$ of its corresponding RBaRS $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}}')$. By Lemma 4.20, the latter is the same as $\grave{\mathbf{C}}_{\mathsf{r}}(\grave{\mathcal{R}})$, and thus $\mathbf{C}(\mathcal{R} \times \{\mathsf{id}_{\epsilon\to\epsilon}\}, \mathbf{1}, \mathbf{1}) = \mathbf{C}_{\mathsf{r}}(\mathcal{R}') = \mathbf{C}_{\mathsf{r}}(\mathcal{R})$. □

4.3 Stochastic Parametric Reactive Systems

We now proceed to give a stochastic semantics to PRSs, generalizing and recasting the work on stochastic reduction semantics for a subset of BRSs in [25].

Intuitively, this is done by interpreting reaction rules as follows: a redex models a physical configuration that may lead to reaction which, over stochastic time, results in the physical configuration described by the corresponding reactum. In other words, we assign reaction rules a stochastic speed which will allow us to assign stochastic behavior to reactions.

In more detail, we wish to associate reactions with a rate, which "is the parameter of an exponential distribution that characterizes the stochastic behavior of that reaction" [25]. The rate of a reaction is then derived from the reaction rules that generate that reaction as follows:

(a) We associate a *rate constant* ϱ with every reaction rule (its speed).

(b) The sum of the rate constants of all the rule instances that generate a reaction is its rate.

It is desirable for the rate of a reaction to be determined from the matches in its left-hand-side, since the stochastic behavior of an agent may then be determined without considering the right-hand-sides of reaction. In other words, we want each match to determine a single reaction. This is not the case for the definition of PRSs in the previous section: since we are free to choose any support translation of the reactum a match leads to infinitely many reactions for non-trivial reactums. We shall therefore refine the definition of the PRS reaction relation such that the support translation of the reactum is deterministic.

First, let us make precise what a match is in a PRS.

4.3.1 Matches

The usual definition of a bigraph match (cf. Def. 2.39) is too coarse-grained for PRSs, as it is defined up to support equivalence. Instead, we shall use the following definition:

Definition 4.31 (match)**.** In a PRS $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$, a *match* o of a parametric rule $\mathsf{R} = (R : I \to J, R' : I' \to J, \bar{\eta}_{J',S})$ in an agent a is a quadruple

$$(\rho, \mathsf{id}_{J'}, c, d)$$

where $\rho : |R| \rightarrowtail |a|$ is a support translation, $\mathsf{id}_{J'} \in \grave{\mathbf{I}}$ an identity, c a context, and $d \in \grave{\mathbf{D}}(\epsilon, I \otimes J')$ a parameter such that $a = c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d$.

Two matches $(\rho, \mathsf{id}_I, c, d), (\rho', \mathsf{id}_{I'}, c', d')$ are regarded as the same if they differ only by an iso between I and I'; otherwise they are *distinct*. We say that a match *results* in a' if $c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d \twoheadrightarrow a'$. We write $\mu_{\mathsf{R}}[a]$ for the number of distinct matches of R in a, and $\mu_{\mathsf{R}}[a, a']$ for the number of distinct matches of R in a resulting in a'. □

Note that non-trivial support automorphisms, i.e., a non-identity support translation $\rho : |G| \to |G|$ such that $\rho \bullet G = G$, give rise to distinct matches:

Lemma 4.32. *Given a* match $o = (\rho, \mathsf{id}_{J'}, c, d)$ *of a parametric rule* $\mathsf{R} = (R : I \to J, R' : I' \to J, \bar{\eta}_{J',S})$ *in an agent* a*, all in a PRS* $`\mathbf{C}(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$*. Then any support automorphism* $\rho' : |R| \to |R|$ *for* R *gives rise to a match* $o' = (\rho \circ \rho', \mathsf{id}_{J'}, c, d)$*. Furthermore, if* ρ' *is not the identity then* o *and* o' *are distinct.*

Proof. Since

$$\begin{aligned} a &= c \circ (\rho \bullet R \otimes \mathsf{id}_{J'}) \circ d \\ &= c \circ (\rho \bullet (\rho' \bullet R) \otimes \mathsf{id}_{J'}) \circ d \\ &= c \circ ((\rho \circ \rho') \bullet R \otimes \mathsf{id}_{J'}) \circ d \end{aligned}$$

o' is an match. Assuming ρ' is not an identity we have $\rho \neq \rho \circ \rho'$ since both ρ and ρ' are bijections, and thus o' is distinct from o. □

This is a point where our stochastic semantics differ from that of Krivine et al. [25]: they consider matches the same if they differ by a support automorphism. However, the difference is just a matter of convention and boils down to a scaling of rate constants by the number of support automorphisms of the corresponding redexes – we leave the details as an exercise to the reader.

4.3.2 Deterministic Support Translation of Reactums

Let us now turn to the matter of ensuring that a match determines a single reaction. The solution is rather simple: we shall simply assume the existence of a family of canonical support translations, $\bar{\rho}_{S\subset\mathcal{S},T\subset\mathcal{S}} : S \to \mathcal{S} \setminus T$, defined for finite S and T. For a reactum R' to be inserted in a context with support T, $\bar{\rho}_{|R'|,T}$ is the canonical support translation of R' such that its support becomes disjoint from the context. We shall often omit S and/or T when they are evident from the context.

As was the case for the instantiation families $\bar{\eta}_{J',S}$, one could think that we have introduced an intractable infinite structure. However, $\bar{\rho}_{S,T}$ is simply a technical measure that need not be specified or represented in practice: when we abstract away support, the choice of the support translation family becomes irrelevant. In other words, as long as we are only interested in the abstract behavior of SPRSs, an implementation is free to generate suitable support for reactums as it pleases.

We can now define stochastic PRSs:

Definition 4.33 (stochastic parametric reactive systems (SPRS))**.** A *stochastic parametric reactive system*, written $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$, is a PRS apart from the addition of rates and that reactum support is chosen canonically in the reaction relation:

A *stochastic parametric reaction rule* is a quadruple of the form

$$(R : I \to J, R' : I' \to J, \bar{\eta}_{J',S}, \varrho)$$

where the first three elements are as before and $\varrho \in \mathbb{R}_+$ is its *rate constant*.

The *reaction relation* $\twoheadrightarrow$ over agents $a, a' \in `\mathbf{C}(\epsilon, \cdot)$ is the smallest such that $a \twoheadrightarrow a'$ whenever $(\rho, \mathsf{id}_{J'}, c, d)$ is a match of some parametric reaction rule $\mathsf{R} = (R, R', \bar{\eta}, \varrho)$ in a and $a' = c \circ (\bar{\rho}_{|R'|,|c|} \bullet R' \otimes \mathsf{id}_{J'}) \circ \bar{\eta}_{|c| \cup \mathsf{rng}(\bar{\rho}_{|R'|,|c|})}(d)$.

We define the rate $\mathit{rate}[a, a']$ of a reaction $a \twoheadrightarrow a'$ to be

$$\mathit{rate}[a, a'] \stackrel{\text{def}}{=} \sum_{\mathsf{R}=(R,R',\bar{\eta}_{J',S},\varrho) \in `\mathcal{R}} \varrho \cdot \mu_{\mathsf{R}}[a, a'].$$

□

For now, let us disregard stochastics and focus on the relation between SPRSs and PRSs (cf. Def. 4.13). The difference from PRSs is that we have refined the reaction relation such that a match determines a single reaction instead of an infinite family of support equivalent reactions (for non-trivial reactums).

Let us make the relation between SPRSs and PRSs precise in the same manner used in the previous sections; the proofs are trivial so we omit them.

Definition 4.34 (SPRS corresponding to PRS)**.** Let $`\mathbf{C}(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ be a PRS. Then the SPRS *corresponding* to $`\mathbf{C}(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ is $`\mathbf{C}_s(`\mathcal{R}', `\mathbf{D}, `\mathbf{I})$ where

$$\begin{aligned} `\mathcal{R}' = \{&(R : I \to J, R' : I' \to J, \bar{\eta}_{J',S}, 1) \\ &\mid (R : I \to J, R' : I' \to J, \bar{\eta}_{J',S}) \in `\mathcal{R}\}. \end{aligned}$$

□

Proposition 4.35 (SPRS corresponding to PRS)**.** *The SPRS corresponding to a PRS is indeed an SPRS.*

Definition 4.36 (PRS corresponding to SPRS)**.** Let $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ be an SPRS. Then the PRS *corresponding* to $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ is $`\mathbf{C}(`\mathcal{R}', `\mathbf{D}, `\mathbf{I})$ where

$$\begin{aligned} `\mathcal{R}' = \{&(R : I \to J, R' : I' \to J, \bar{\eta}_{J',S}) \\ &\mid (R : I \to J, R' : I' \to J, \bar{\eta}_{J',S}, \varrho) \in `\mathcal{R}\}. \end{aligned}$$

□

Proposition 4.37 (PRS corresponding to SPRS)**.** *The PRS corresponding to an SPRS is indeed a PRS.*

Lemma 4.38. *For any PRS* $`\mathbf{C}(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$*, the PRS* $`\mathbf{C}(`\mathcal{R}', `\mathbf{D}, `\mathbf{I})$ *obtained through Def. 4.34 followed by Def. 4.36 is the same.*

Proof. Immediate from the definitions. □

Proposition 4.39 (Stochastic Parametric Reactions are Parametric Reactions)**.** *Let* $\longrightarrow_s$ *and* $\longrightarrow_p$ *denote the reaction relations of an SPRS* $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ *and its corresponding PRS, respectively. Then*

$$a \longrightarrow_s a' \quad \Rightarrow \quad a \longrightarrow_p a'.$$

Proposition 4.40 (Stochastic Parametric Reactions are Sufficient)**.** *Let* $\longrightarrow_s$ *and* $\longrightarrow_p$ *denote the reaction relations of an SPRS* $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ *and its corresponding PRS, respectively. Then*

$$\forall n \in \mathbb{N} : \quad a \longrightarrow_p^n a' \quad \Rightarrow \quad \exists a'' : a \longrightarrow_s^n a'' \wedge a' \doteqdot a''.$$

4.3.3 Abstract Stochastic Parametric Reactive Systems

Given the relations between PRSs and SPRSs we saw above it is clear that the abstractions of their reaction relations are the same. But before we can make this formal, we must first define how to construct abstract SPRSs. The abstraction constructions and results for PRSs (cf. Def. 4.2.3) transfer directly to SPRSs, except that the quotient construction must be extended to handle rates:

Definition 4.41 (quotient SPRS)**.** Let $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ be an SPRS, and $\equiv$ a dynamic abstraction on $`\mathbf{C}$. Then define $\mathbf{C}_s(\mathcal{R}, \mathbf{D}, \mathbf{I})$, the quotient of $`\mathbf{C}_s(`\mathcal{R}, `\mathbf{D}, `\mathbf{I})$ by $\equiv$, as in Def. 4.27.

The rate of reaction in $\mathbf{C}_s(\mathcal{R}, \mathbf{D}, \mathbf{I})$ is defined as:

$$\mathit{rate}[\hat{a}, \hat{a}'] = \sum_{a' \in \hat{a}'} \mathit{rate}[a, a'] \quad \text{for any } a \in \hat{a}.$$

□

Thus we define the rate of an abstract reaction by choosing a representative of its left-hand-side and then summing the rates of all reactions into the equivalence class of the right-hand-side.

This is well-defined since $\equiv$ is a *dynamic* abstraction, and thus reactions, and thereby matches, and rates are independent of the choice of representative:

Proposition 4.42. *If $a \equiv b$ for a dynamic abstraction $\equiv$, then*

$$\sum_{a' \in \hat{a}'} \mathit{rate}[a, a'] = \sum_{a' \in \hat{a}'} \mathit{rate}[b, a'].$$

Proof. Follows straightforwardly from the fact that $\equiv$ is dynamic (cf. Def. 4.24). □

We can now show that, indeed, SPRSs have the same abstract reactions as their corresponding PRSs:

Theorem 4.43 (abstract SPRSs are abstract PRSs)**.** *Let $\grave{\mathbf{C}}_s(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be an SPRS and let $\grave{\mathbf{C}}(\grave{\mathcal{R}}', \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be the corresponding PRS. Then the quotient SPRS $\mathbf{C}_s(\mathcal{R}, \mathbf{D}, \mathbf{I})$ and quotient PRS $\mathbf{C}(\mathcal{R}', \mathbf{D}, \mathbf{I})$ have the same reaction relations.*

Proof. Follows easily from Prop. 4.39 and Prop. 4.40. □

Conversely, a PRS has the same abstract reactions as its corresponding SPRS:

Theorem 4.44 (abstract PRSs are abstract SPRSs)**.** *Let $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be a PRS and let $\grave{\mathbf{C}}_s(\grave{\mathcal{R}}', \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be the corresponding SPRS. Then the quotient PRS $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$ and quotient SPRS $\mathbf{C}_s(\mathcal{R}', \mathbf{D}, \mathbf{I})$ have the same reaction relations.*

Proof. By Theorem 4.43, $\mathbf{C}_s(\mathcal{R}', \mathbf{D}, \mathbf{I})$ has the same reaction relation as $\mathbf{C}(\mathcal{R}'', \mathbf{D}, \mathbf{I})$, the quotient of its corresponding PRS $\grave{\mathbf{C}}(\grave{\mathcal{R}}'', \grave{\mathbf{D}}, \grave{\mathbf{I}})$. By Lemma 4.38, the latter is the same as $\grave{\mathbf{C}}(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$, and thus $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I}) = \mathbf{C}(\mathcal{R}'', \mathbf{D}, \mathbf{I})$ so $\mathbf{C}_s(\mathcal{R}', \mathbf{D}, \mathbf{I})$ has the same reaction relation as $\mathbf{C}(\mathcal{R}, \mathbf{D}, \mathbf{I})$. □

Finally, as a sanity check, we verify that rates are consistent with the reaction relation:

Proposition 4.45 (consistency)**.** *Let $\grave{\mathbf{C}}_s(\grave{\mathcal{R}}, \grave{\mathbf{D}}, \grave{\mathbf{I}})$ be an SPRS and $\mathbf{C}_s(\mathcal{R}, \mathbf{D}, \mathbf{I})$ its quotient by a dynamic abstraction $\equiv$ on $\grave{\mathbf{C}}$. Then*

$$\mathit{rate}[a, a'] > 0 \text{ iff } a \rightarrowtriangle a', \quad \text{and} \quad \mathit{rate}[\hat{a}, \hat{a}'] > 0 \text{ iff } \hat{a} \rightarrowtriangle \hat{a}'$$

Proof. $\mathit{rate}[a, a'] > 0 \Rightarrow a \rightarrowtriangle a'$: From the definition of $\mathit{rate}[a, a']$ we see that there must be some rule $\mathsf{R} = (R, R', \bar{\eta}_{J',S}, \varrho)$ with $\mu_{\mathsf{R}}[a, a'] > 0$ and thus, by definition of $\mu_{\mathsf{R}}[a, a']$, $a \rightarrowtriangle a'$.

$a \rightarrowtriangle a' \Rightarrow \mathit{rate}[a, a'] > 0$: By the definition of the reaction relation, we have a match of a rule in a resulting in a' and thus $\mu_{\mathsf{R}}[a, a'] > 0$ which implies $\mathit{rate}[a, a'] > 0$.

$\mathit{rate}[\hat{a}, \hat{a}'] > 0 \Rightarrow \hat{a} \rightarrowtriangle \hat{a}'$: From the definition of $\mathit{rate}[\hat{a}, \hat{a}']$ we see that there must be some $a \in \hat{a}, a' \in \hat{a}'$ such that $\mathit{rate}[a, a'] > 0$ and thus, as shown above, $a \rightarrowtriangle a'$. Lastly, Theorem 4.28 gives us $\hat{a} \rightarrowtriangle \hat{a}'$.

$\hat{a} \rightarrowtriangle \hat{a}' \Rightarrow \mathit{rate}[\hat{a}, \hat{a}'] > 0$: Theorem 4.28 tells us that there must be some $a \in \hat{a}, a' \in \hat{a}'$ such that $a \rightarrowtriangle a'$ and thus, as shown above, $\mathit{rate}[a, a'] > 0$. Now it is obvious from its definition that $\mathit{rate}[\hat{a}, \hat{a}'] > 0$. □

5 Bigraph Embeddings

In the previous section we defined stochastic parametric reactive systems. A key component in that development was to make precise the algebraic notion of a *match* of a parametric redex in an agent. The KaSim algorithm relies on a representation of matches as embeddings (cf. Section 3), so in this section we shall develop a general theory of bigraph embeddings, where embeddings of redexes are isomorphic to matches.

We shall exploit the orthogonality of the link and place graphs, by developing link and place graph embeddings independently and then combine them to obtain bigraph embeddings.

In overview, the development proceeds as follows:

embedding maps:
We define an embedding of a graph as the union of maps of identities (i.e., a support translation) and maps of the inner and outer faces. The maps must satisfy certain conditions that ensure structure preservation and correspondence with certain algebraic decompositions.

embedding/context isomorphism:
For place graphs and bigraphs, we show that embeddings are isomorphic to certain decompositions, giving constructions in both directions. For redexes this implies that embeddings and matches are isomorphic.

Link graphs seem to lack the necessary structure for embeddings to have this property.

We shall take special interest in the embeddings of a particular class of bigraphs: those that are *solid*. Simply put, a bigraph is solid if all elements of the outer and inner interfaces are connected to a node and not connected to each other. Solid bigraphs are interesting for two reasons: many bigraphical models in the literature have solid redexes[5] and an embedding of a solid bigraph is determined by a support translation of its nodes, making matches compactly representable.

5.1 Link Graph Embeddings

Embeddings of link graphs are mostly what one would expect of a graph embedding: a pair of injections of the nodes and edges which preserve the structure of the embedded graph (i.e., a support translation). In addition, we need to specify how the names of the interfaces should be mapped; in bigraphs, a context is allowed to alias names, so any map from the outer face names to the links of the host graph will do. Dually, we map the names of the inner face to sets of points in the host graph.

The definition is based on Milner's definition of link graph inclusion [28], extended to cover link graphs in general and with a minor correction[6].

Definition 5.1 (link graph embedding)**.** Let $G : X_G \to Y_G, H : X_H \to Y_H$ be two concrete link graphs. Then a *link graph embedding*, written $\phi : G \hookrightarrow H$, is a map $\phi : |G| \uplus X_G \uplus Y_G \to |H| \uplus \mathcal{P}(X_H \uplus P_H) \uplus Y_H$, where $\phi = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{e}} \uplus \phi^{\mathsf{i}} \uplus \phi^{\mathsf{o}}$ satisfies the following conditions:

maps:

(LGE-1) $\phi^{\mathsf{v}} : V_G \rightarrowtail V_H$ is an injective map

(LGE-2) $\phi^{\mathsf{e}} : E_G \rightarrowtail E_H$ is an injective map

(LGE-3) $\phi^{\mathsf{i}} : X_G \rightarrowtail \mathcal{P}(X_H \uplus P_H)$ is a fully injective map

(LGE-4) $\phi^{\mathsf{o}} : Y_G \to E_H \uplus Y_H$ is an arbitrary map

injectivity:

(LGE-5) $\mathsf{rng}(\phi^{\mathsf{e}}) \# \mathsf{rng}(\phi^{\mathsf{o}})$

[5]For example, all BRSs in [24, 25, 29] have solid redexes.

[6]In [28] Milner missed that embeddings must be surjective on the points of an edge, cf. Example 1.

(LGE-6) $\mathsf{rng}(\phi^{\mathsf{i}}) \mathrel{\#} \mathsf{rng}(\phi^{\mathsf{port}})$

surjective on edge points:

(LGE-7) $\phi^{\mathsf{p}} \circ \mathit{link}_G^{-1} \restriction_{E_G} = \mathit{link}_H^{-1} \circ \phi^{\mathsf{e}}$

structure preservation:

(LGE-8) $\mathit{ctrl}_G = \mathit{ctrl}_H \circ \phi^{\mathsf{v}}$

(LGE-9) $\forall p \in X_G \uplus P_G : \forall p' \in \phi^{\mathsf{p}}(p) : (\phi^{\mathsf{l}} \circ \mathit{link}_G)(p) = \mathit{link}_H(p')$

where

$$\begin{aligned} \phi^{\mathsf{l}} &= \phi^{\mathsf{e}} \uplus \phi^{\mathsf{o}} && && \text{(map of links)} \\ \phi^{\mathsf{port}}(v,i) &= (\phi^{\mathsf{v}}(v), i) && ((v,i) \in P_G) && \text{(map of ports)} \\ \phi^{\mathsf{p}} &= \phi^{\mathsf{i}} \uplus \phi^{\mathsf{port}} && && \text{(map of points).} \end{aligned}$$

We do not take the codomain of ϕ as part of its definition, and thus it may be an embedding into several link graphs. We write $\phi \mathbin{\scriptstyle\blacksquare} G$ when applying the underlying support translation to G. If any of the maps are partial, ϕ is *partial*, written $\phi : G \hookrightharpoonup H$. Partial embeddings need only satisfy the conditions where they are defined; in particular the surjectivity condition only applies to an edge e iff ϕ^{e} is defined for e and ϕ^{p} is defined for $\mathit{link}_G^{-1}(e)$. A partial embedding is said to be *non-trivial* iff its range is non-empty. □

Condition (LGE-7) deserves an explanation:

Example 1. Consider the following ground link graphs

$$\begin{aligned} G &= (\{v\}, \{e\}, \{v \mapsto K\}, \{(v,0) \mapsto e\}) : \emptyset \to \emptyset \\ H &= (\{v, v'\}, \{e\}, \{v \mapsto K, v' \mapsto K\}, \{(v,0) \mapsto e, (v',0) \mapsto e\}) : \emptyset \to \emptyset \\ ar(K) &= 1 \end{aligned}$$

Then the following would be a link graph embedding if we did not include condition (LGE-7):

$$\phi = \mathsf{Id}_{\{v,e\}} : G \hookrightarrow H$$

But there is no link graph C such that $H = C \circ \phi \mathbin{\scriptstyle\blacksquare} G$! □

The problem is that the context cannot add more points to an edge, so the points of an edge in G must cover all the points the corresponding edge in H.

In [28] Milner showed that, in the case of ground link graphs, contexts and embeddings are isomorphic. Unfortunately, there is no such correspondence in the general case, as the following example shows:

Example 2. Consider the following link graphs

$$\begin{aligned} G &= (\{v\}, \emptyset, \{v \mapsto K\}, \emptyset) : \emptyset \to \emptyset \\ H &= (\{v, v'\}, \emptyset, \{v \mapsto K, v' \mapsto K\}, \emptyset) : \emptyset \to \emptyset \\ ar(K) &= 0 \end{aligned}$$

Then $\phi = \mathsf{Id}_{\{v\}} : G \hookrightarrow H$ is a link graph embedding and there are two different decompositions of H that include $\phi \mathbin{\scriptstyle\blacksquare} G$: $H = C \circ \phi \mathbin{\scriptstyle\blacksquare} G \circ D = D \circ \phi \mathbin{\scriptstyle\blacksquare} G \circ C$ where

$$\begin{aligned} C &= (\emptyset, \emptyset, \emptyset, \emptyset) : \emptyset \to \emptyset \\ D &= (\{v'\}, \emptyset, \{v' \mapsto K\}, \emptyset) : \emptyset \to \emptyset \end{aligned}$$

□

Though one could perhaps recover the correspondence by restricting to some canonical contexts, we shall not pursue this here, as we shall recover the correspondence once we combine link and place graph embeddings.

5.1.1 Solid Link Graphs

For an important class of link graphs, those that are *solid*, embeddings are determined by the injections of nodes:

Definition 5.2 (solid link graph (after [25, Def. 2.1]))**.** A link graph is *solid* iff these conditions hold:

1. no links are idle
2. no inner names are siblings
3. every inner name is guarding
4. no outer name is linked to an inner name.

□

The notion of solidness comes from stochastic bigraphs [25] where redexes are required to be solid, which essentially ensures that a match is determined by the support translation. We have strengthened the condition in two respects, which enables us to obtain a stronger and more general result without diminishing the set of solid redexes: (a) we preclude idle edges and (b) we require inner names to be guarding. To see that these condition do not rule out any redexes, remember that (a) we may simple choose concretions of the abstract redexes with no idle edges, and (b) that redexes have no inner names and thus 3. is vacuously satisfied.

The conditions ensure that an embedding and its context and parameters are determined by just the support translation of the nodes:

Proposition 5.3 (solid link graph embeddings)**.** *Given a solid link graph $G : X_G \to Y_G$ and an embedding $\phi : G \hookrightarrow H$ into a link graph $H : X_H \to Y_H$. Then ϕ^{e}, ϕ^{i}, and ϕ^{o} are uniquely determined from ϕ^{v}.*

Proof. Here we give only the constructions of ϕ^{e}, ϕ^{i} and ϕ^{o}. Proofs that they are unique and satisfy the embedding conditions may be found in Appendix A.1.1.

ϕ^{e}: Construct the map of each edge $e \in E_G$ as follows: choose a port $p = (v, i) \in \mathit{link}_G^{-1}(e)$, which is always possible since no edge is idle and every inner name is guarding, and let

$$\phi^{\mathsf{e}}(e) = \mathit{link}_H(\phi^{\mathsf{v}}(v), i).$$

ϕ^{i}: Construct the map of each inner name $x \in X_G$ as follows:

$$\begin{aligned}
\phi^{\mathsf{i}}(x) &= \mathit{points}_{H,x} \setminus \phi^{\mathsf{p}}(P_{G,x}) \\
\mathit{points}_{H,x} &= (\mathit{link}_H^{-1} \circ \phi^{\mathsf{e}})(\mathit{link}_G(x)) \\
P_{G,x} &= (\mathit{link}_G^{-1} \circ \mathit{link}_G)(x) \setminus \{x\} \\
\phi^{\mathsf{p}}(v, i) &= (\phi^{\mathsf{v}}(v), i).
\end{aligned}$$

ϕ^{o}: Construct the map of each outer name $y \in Y_G$ as follows: choose a port $p = (v, i) \in \mathit{link}_G^{-1}(y)$, which is always possible since no outer name is idle or connected to an inner name, and let

$$\phi^{\mathsf{o}}(y) = \mathit{link}_H(\phi^{\mathsf{v}}(v), i).$$

□

5.2 Place Graph Embeddings

As for link graph embeddings, place graph embeddings are simply support translations along with maps of the interfaces:

Definition 5.4 (place graph embedding)**.** Let $G : k_G \to m_G, H : k_H \to m_H$ be two concrete place graphs. Then a *place graph embedding*, written $\phi : G \hookrightarrow H$, is a map $\phi : |G| \uplus k_G \uplus m_G \to |H| \uplus \mathcal{P}(k_H \uplus V_H) \uplus m_H$, where $\phi = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}}$ satisfies the following conditions:

maps:

(PGE-1) $\phi^{\mathsf{v}} : V_G \rightarrowtail V_H$ is an injective map

(PGE-2) $\phi^{\mathsf{s}} : k_G \to \mathcal{P}(k_H \uplus V_H)$ is a fully injective map

(PGE-3) $\phi^{\mathsf{r}} : m_G \to V_H \uplus m_H$ is an arbitrary map

injectivity:

(PGE-4) $\mathsf{rng}(\phi^{\mathsf{v}}) \# \mathsf{rng}(\phi^{\mathsf{r}})$

(PGE-5) $\mathsf{rng}(\phi^{\mathsf{s}}) \# \mathsf{rng}(\phi^{\mathsf{v}})$

(PGE-6) $H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \# \mathsf{rng}(\phi^{\mathsf{r}})$

surjective on node children:

(PGE-7) $\phi^{\mathsf{c}} \circ prnt_G^{-1} \restriction_{V_G} = prnt_H^{-1} \circ \phi^{\mathsf{v}}$

structure preservation:

(PGE-8) $ctrl_G = ctrl_H \circ \phi^{\mathsf{v}}$

(PGE-9) $\forall c \in k_G \uplus V_G : \forall c' \in \phi^{\mathsf{c}}(c) : (\phi^{\mathsf{f}} \circ prnt_G)(c) = prnt_H(c')$

where

$$\phi^{\mathsf{f}} = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{r}} \qquad \text{(map of parents)}$$
$$\phi^{\mathsf{c}} = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \qquad \text{(map of children).}$$

We do not take the codomain of ϕ as part of its definition, and thus it may be an embedding into several place graphs. We write $\phi \centerdot G$ when applying the underlying support translation to G. If any of the maps are partial, ϕ is *partial*, written $\phi : G \hookrightarrow H$. Partial embeddings need only satisfy the conditions where they are defined; in particular the surjectivity condition only applies to a node v iff ϕ^{v} is defined for v and ϕ^{c} is defined for $prnt_G^{-1}(v)$. A partial embedding is said to be *non-trivial* iff its range is non-empty. □

The conditions are analogous to those for link graph embeddings, except condition (PGE-6) which deserves an explanation. Let us first motivate it by an example:

Example 3. Consider the following place graphs

$$G = (\emptyset, \emptyset, \{0 \mapsto 0, 1 \mapsto 1\}) : 2 \to 2$$
$$H = (\{v, v'\}, \{v \mapsto K, v' \mapsto K\}, \{v \mapsto 0, v' \mapsto v\}) : 1$$

Then the following would be a place graph embedding if we did not include condition (PGE-6):

$$\phi = \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} : G \hookrightarrow H$$
$$\phi^{\mathsf{r}} = \{0 \mapsto 0, 1 \mapsto v'\}$$
$$\phi^{\mathsf{s}} = \{0 \mapsto \{v\}, 1 \mapsto \emptyset\}$$

But there are no place graphs C and D such that $H = C \circ \phi \centerdot G \circ D$! The problem is that root 1 of G is mapped to node v' which is part of the tree that site 0 is mapped to. □

The issue is that there are no place graph operations that can make one root of a place graph a descendant of one of its other roots. In other words, roots do not just model possibly disjoint locations, but subtrees that are disjoint. This is a design choice in the bigraphical model, and it is out of scope for this report to investigate the consequences of relaxing this restriction. Thus, for embeddings to correspond to decompositions, we need to rule out embeddings where one root is mapped to a descendant of another, hence condition (PGE-6).

The decompositions that we can express with an embedding are the following:

Definition 5.5 (embedding corresponding to decomposition)**.** Given a place graph decomposition

$$H = C \circ (G \circ D \otimes \mathsf{id}_k) \circ \pi.$$

Then the *corresponding* embedding $\phi = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} : G \hookrightarrow H$ is defined by

$$\phi^{\mathsf{v}} = \mathsf{Id}_{V_G} \qquad \phi^{\mathsf{r}} = \mathit{prnt}_C \restriction_{m_G} \qquad \phi^{\mathsf{s}} = (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ \mathit{prnt}_D^{-1} \restriction_{k_G} .$$

□

Proposition 5.6 (embedding corresponding to decomposition)**.** *The embedding $\phi : G \hookrightarrow H$ of Def. 5.5 is indeed an embedding.*

Proof. Cf. Appendix A.1.2. □

Note that in the case where $k = 0$ and $\pi = \mathsf{id}$, the decomposition becomes $H = C \circ G \circ D$, demonstrating that embeddings are indeed just decompositions into context, redex, and parameter. The identity id_k allows some of the sites of H to be in the context C. The permutation π is a technical measure to handle the fact that sites are not names but consecutive numbers: it expresses that the sites of H may belong to either the context or the parameter, and in a decomposition we have to partition and renumber them accordingly.

Let us make this precise, by defining when we consider decompositions equivalent:

Definition 5.7 (decomposition equivalence)**.** Say that two decompositions

$$\begin{aligned} H &= C \circ (G \circ D \otimes \mathsf{id}_k) \circ \pi \\ &= C' \circ (G \circ D' \otimes \mathsf{id}_k) \circ \pi' \end{aligned}$$

are the same iff they differ only on their internal numbering of sites, i.e.,

$$\begin{aligned} V_D &= V_{D'} \\ V_C &= V_{C'} \\ \mathit{prnt}_D \restriction_{V_D} &= \mathit{prnt}_{D'} \restriction_{V_D} \\ \mathit{prnt}_C \restriction_{V_C \uplus m_G} &= \mathit{prnt}_{C'} \restriction_{V_C \uplus m_G} \\ \mathit{prnt}_D \circ \pi \downharpoonleft^{k_D} &= \mathit{prnt}_{D'} \circ \pi' \downharpoonleft^{k_D} \\ \mathit{prnt}_C(\pi(i) - k_D + m_G) &= \mathit{prnt}_{C'}(\pi'(i) - k_D + m_G) \qquad (i \in \pi^{-1}(k_H \setminus k_D)). \end{aligned}$$

□

Let us now turn to showing that embeddings can only express such decompositions:

Definition 5.8 (decomposition corresponding to embedding)**.** Given a place graph $G : k_G \to m_G$ and an embedding $\phi : G \hookrightarrow H$ into a place graph $H : k_H \to m_H$. Then the *corresponding* decomposition into *parameter* $\mathit{prmt}(\phi)$ and *context* $\mathit{ctxt}(\phi)$ place graphs are as defined in Figure 3. □

Proposition 5.9 (embeddings are decompositions)**.** *Given a place graph $G : k_G \to m_G$ and an embedding $\phi : G \hookrightarrow H$ into a place graph $H : k_H \to m_H$. Then construction Def. 5.8 defines a decomposition up to decomposition equivalence.*

$$
\begin{aligned}
prmt(\phi) &\overset{\text{def}}{=} (V_D, ctrl_H \upharpoonright_{V_D}, prnt_D) : k_D \to k_G \qquad \text{where} \\
V_D &= V_H \cap H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
\tilde{k}_D &= k_H \cap H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
k_D &= |\tilde{k}_D| \\
f_D &: k_D \rightarrowtail \tilde{k}_D \qquad \text{a bijection} \\
prnt_D &= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \upharpoonright_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D})
\end{aligned}
$$

$$
\begin{aligned}
ctxt(\phi) &\overset{\text{def}}{=} (V_C, ctrl_H \upharpoonright_{V_C}, prnt_C) : k_C \to m_H \qquad \text{where} \\
V_C &= (V_H \setminus \phi^{\mathsf{v}}(V_G)) \setminus V_D \\
\tilde{k}_C &= k_H \setminus \tilde{k}_D \\
k_C &= m_G + |\tilde{k}_C| \\
f'_C &: |\tilde{k}_C| \rightarrowtail \tilde{k}_C \qquad \text{a bijection} \\
f_C(i + m_G) &= f'_C(i) \quad \text{for } i \in |\tilde{k}_C| \\
prnt_C &= \phi^{\mathsf{r}} \uplus prnt_H \upharpoonright_{V_C} \uplus prnt_H \circ f_C \\
f'(i + k_D) &= f'_C(i) \quad \text{for } i \in |\tilde{k}_C| \\
\pi &= f_D^{-1} \uplus f'^{-1} : k_H \to k_H
\end{aligned}
$$

$$
H = ctxt(\phi) \circ (\phi \centerdot G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi
$$

Figure 3: Decomposition of place graph $H : k_H \to m_H$ into parameter $prmt(\phi)$ and context $ctxt(\phi)$ corresponding to an embedding $\phi : G \hookrightarrow H$ of a place graph $G : k_G \to m_G$.

Proof. Cf. Appendix A.1.3. □

The bijections f_D and f'_C are the realizations of the internal partitioning and renumbering of sites that we discussed above and they express the variation within decomposition equivalence classes. Note that the support translation of G slightly muddles the correspondence with the decomposition of Def. 5.5 above. However, letting $F \stackrel{\text{def}}{=} \phi \bullet G$ we see that an embedding indeed specifies such a decomposition.

Together, the above constructions form an isomorphism between embeddings and decompositions:

Theorem 5.10 (embeddings and decompositions are isomorphic)**.** *The constructions of Def. 5.5 and Def. 5.8 are mutually inverse.*

Proof. Cf. Appendix A.1.4. □

When we get to edit scripts, we shall need a number of disjointness results in addition to the injectivity conditions:

Lemma 5.11. *Given a place graph $G : k_G \to m_G$ and an embedding $\phi : G \hookrightarrow H$ into a place graph $H : k_H \to m_H$. Then*

1. $\mathsf{rng}(\phi^{\mathsf{f}}) \# H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}$,
2. $\mathsf{rng}(\phi^{\mathsf{c}}) \# H \upharpoonleft_{\mathsf{rng}(\phi^{\mathsf{r}})}$,
3. $H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \# H \upharpoonleft_{\mathsf{rng}(\phi^{\mathsf{r}})}$, *and*
4. $\forall i \in k_G : \mathsf{rng}(\phi^{\mathsf{c}}) \# (H \downharpoonleft^{\phi^{\mathsf{s}}(i)} \setminus \phi^{\mathsf{s}}(i))$.

Proof. Cf. Appendix A.1.5. □

5.2.1 Solid Place Graphs

As was the case with link graph embeddings, place graph embeddings are determined by the injection of nodes *iff* the place graph is solid:

Definition 5.12 (solid place graph (after [25, Def. 2.1]))**.** A place graph is *solid* iff these conditions hold:

1. no roots are idle
2. no sites are siblings
3. every site is guarding .

□

Proposition 5.13. *Given a solid place graph $G : k_G \to m_G$ and an embedding $\phi : G \hookrightarrow H$ into a place graph $H : k_H \to m_H$. Then ϕ^{s} and ϕ^{r} are uniquely determined from ϕ^{v}.*

Proof. Here we give only the constructions of ϕ^{s} and ϕ^{r}. Proofs that they are unique and satisfy the embedding conditions may be found in Appendix A.1.6.

ϕ^{s}: Construct the map of each site $i \in k_G$ as follows:

$$\begin{aligned}
\phi^{\mathsf{s}}(i) &= \mathit{children}_{H,i} \setminus \phi^{\mathsf{v}}(\mathit{siblings}_{G,i}) \\
\mathit{children}_{H,i} &= (\mathit{prnt}_H^{-1} \circ \phi^{\mathsf{v}})(\mathit{prnt}_G(i)) \\
\mathit{siblings}_{G,i} &= (\mathit{prnt}_G^{-1} \circ \mathit{prnt}_G)(i) \setminus \{i\}.
\end{aligned}$$

ϕ^{r}: Construct the map of each root $j \in m_G$ as follows: choose a node $v \in prnt_G^{-1}(j)$, which is always possible since no root is idle or has a site as a child, and let

$$\phi^{\mathsf{r}}(j) = (prnt_H \circ \phi^{\mathsf{v}})(v).$$

□

5.3 Bigraph Embeddings

Having defined embeddings for each of the two constituent graphs, we can now define embeddings of bigraphs as the combination of the two, adding only a single condition:

Definition 5.14 (bigraph embedding)**.** Let $G : \langle k_G, X_G\rangle \to \langle m_G, Y_G\rangle, H : \langle k_H, X_H\rangle \to \langle m_H, Y_H\rangle$ be two concrete bigraphs. Then a *bigraph embedding*, written $\phi : G \hookrightarrow H$, is a map $\phi : |G| \uplus k_G \uplus m_G \uplus X_G \uplus Y_G \to |H| \uplus \mathcal{P}(k_H \uplus V_H) \uplus m_H \uplus \mathcal{P}(X_H \uplus P_H) \uplus Y_H$, where $\phi^P = \phi \restriction_{V_G \uplus k_G \uplus m_G} : G^P \hookrightarrow H^P$ is place graph embedding and $\phi^L = \phi \restriction_{|G| \uplus X_G \uplus Y_G} : G^L \hookrightarrow H^L$ is a link graph embedding. Furthermore, the map must satisfy the following condition:

consistency:

(BGE-1) $\mathsf{rng}(\phi^{\mathsf{i}}) \subseteq X_H \uplus P_{H\downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \cap V_H}$.

We do not take the codomain of ϕ as part of its definition, and thus it may be an embedding into several bigraphs. We write $\phi \bullet G$ when applying the underlying support translation to G. If ϕ^{P} or ϕ^{L} are partial, ϕ is *partial*, written $\phi : G \hookrightarrow H$; if either is *non-trivial*, so is ϕ. □

The consistency condition ensures that the link graph embedding only maps inner names to ports on nodes that are in the place graph parameter. If we did not have this condition, the link and place graph embeddings might disagree on whether a node belongs to the context or the parameter.

We saw in the previous section that the place graph structure gives us a unique way to separate the nodes that are not in the image of the embedding into a context and parameter. Link graphs have less structure and for a given embedding there may be several ways to decompose the link graph (e.g., different ways to partition the edges between context and parameter). Depending on the definition of reaction, this may affect the reaction relation. For pure bigraphs, Milner resolves this issue by disallowing inner names in redexes and requiring parameters to be discrete [29, Def. 8.5]. For binding bigraphs, inner names are allowed in redexes as long as they are local and the definition of discreteness is conservatively extended to exempt bound links [24, Sec. 11]. We shall adapt (a variant of) the latter approach in order to avoid restricting redexes and to make our work extensible to binding bigraphs: we shall require the parameter to be discrete *except* that (1) we shall discard the bijection constraint for the links that connect to the redex, and (2) inner names can only connect to the redex. We call this *semi-discreteness*:

Definition 5.15 (semi-discrete bigraph)**.** A bigraph $D : \langle X_D, k_D\rangle \to \langle X_G \uplus X_I, m_D\rangle$ is *semi-discrete on* X_G iff it has no edges, no outer name is idle, $link_D \downharpoonright^{X_I}$ is a bijection, and $link_D(X_D) \subseteq X_G$. □

Thus the decompositions that our embeddings correspond to are the following:

Definition 5.16 (embedding corresponding to decomposition)**.** Given a bigraph

$$H = C \circ ((G \otimes \mathsf{id}_{X_I}) \circ D \otimes \mathsf{id}_k \otimes \alpha) \circ (\pi \otimes \mathsf{id}_{X_H})$$

where D is semi-discrete on X_G. Then the *corresponding* embedding $\phi = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{e}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} \uplus \phi^{\mathsf{i}} \uplus \phi^{\mathsf{o}} : G \hookrightarrow H$ is defined by

$$\phi^{\mathsf{v}} = \mathsf{Id}_{V_G} \qquad \phi^{\mathsf{r}} = prnt_C \restriction_{m_G} \qquad \phi^{\mathsf{s}} = (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1} \restriction_{k_G}$$

$$\phi^{\mathsf{e}} = \mathsf{Id}_{E_G} \qquad \phi^{\mathsf{o}} = link_C \restriction_{Y_G} \qquad \phi^{\mathsf{i}} = link_D^{-1} \restriction_{X_G} .$$

□

Proposition 5.17 (embedding corresponding to decomposition)**.** *The embedding $\phi : G \hookrightarrow H$ of Def. 5.16 is indeed an embedding.*

Proof. Cf. Appendix A.1.7. □

As we discussed for place graph embeddings, id_k allows sites of H to be in the context C and π is a technical artifact reflecting that sites are consecutive numbers and not names. Similarly, the renaming α allows inner names of H to be in the context C, though suitably renamed to handle the case where inner names of H collide with the outer names of G. If $\mathsf{id}_k = \alpha = \mathsf{id}_\epsilon$ and π is an identity, the decomposition becomes $H = C \circ (G \otimes \mathsf{id}_{X_I}) \circ D$, again demonstrating that embeddings are just decompositions into context, redex, and parameter. The identity id_{X_I} expresses the fact that we allow the parameter and context to share links without the involvement of the redex; in this sense the exact choice of X_I is internal to the decomposition.

Let us extend our definition of decomposition equivalence to disregard the internal names:

Definition 5.18 (decomposition equivalence)**.** Say that two decompositions

$$\begin{aligned} H &= C \circ ((G \otimes \mathsf{id}_{X_I}) \circ D \otimes \mathsf{id}_k \otimes \alpha) \circ (\pi \otimes \mathsf{id}_{X_H}) \\ &= C' \circ ((G \otimes \mathsf{id}_{X_{I'}}) \circ D' \otimes \mathsf{id}_k \otimes \alpha') \circ (\pi' \otimes \mathsf{id}_{X_H}) \end{aligned}$$

with D,D' discrete, are the same iff the place graph decompositions are the same and the link graph decompositions differ only in their internal names, i.e.,

$$\begin{aligned} E_D &= E_{D'} & \mathit{link}_D \downharpoonleft^{X_G} &= \mathit{link}_{D'} \downharpoonleft^{X_G} \\ E_C &= E_{C'} & \mathit{link}_C \upharpoonright_{P_C \uplus Y_G} &= \mathit{link}_{C'} \upharpoonright_{P_C \uplus Y_G} \\ && \mathit{link}_C \circ \alpha &= \mathit{link}_{C'} \circ \alpha' \\ && \mathit{link}_C \circ \mathit{link}_D \downharpoonleft^{X_I} &= \mathit{link}_{C'} \circ \mathit{link}_{D'} \downharpoonleft^{X_{I'}} \ . \end{aligned}$$

□

In the case of bigraph matches, i.e., Def. 4.31 instantiated to BPRSs, this definition captures exactly what it means for matches to be the same:

Proposition 5.19 (matches are decompositions)**.** *Two matches in an agent are the same iff they are equivalent decompositions.*

Proof. Cf. Appendix A.1.8. □

As we did for place graphs, we shall now prove that embeddings can only express such decompositions, by showing how to construct them from an embedding:

Definition 5.20 (decomposition corresponding to embedding)**.** Given a bigraph $G : \langle k_G, X_G\rangle \to \langle m_G, Y_G\rangle$ and an embedding $\phi : G \hookrightarrow H$ into a bigraph $H : \langle k_H, X_H\rangle \to \langle m_H, Y_H\rangle$. Then the *corresponding* decomposition into *parameter* $\mathit{prmt}(\phi)$ and *context* $\mathit{ctxt}(\phi)$ bigraphs are as defined in Figure 4. □

Proposition 5.21 (embeddings are decompositions)**.** *Given a bigraph $G : \langle k_G, X_G\rangle \to \langle m_G, Y_G\rangle$ and an embedding $\phi : G \hookrightarrow H$ into a bigraph $H : \langle k_H, X_H\rangle \to \langle m_H, Y_H\rangle$. Then constructions Def. 5.8 and Def. 5.20 define a decomposition up to decomposition equivalence.*

Proof. Cf. Appendix A.1.9. □

We discussed the place graph aspects of the construction in Section 5.2. The bijections link'_D and α_C are the realizations of the choice of suitable internal names as discussed above and they express the variation within decomposition equivalence classes.

Let us now prove that these constructions form an isomorphism between embeddings and decompositions:

$$
\begin{aligned}
\mathit{prmt}(\phi) &\overset{\text{def}}{=} (V_D, \emptyset, \mathit{ctrl}_D, \mathit{prnt}_D, \mathit{link}_D) : \langle k_D, X_D \rangle \to \langle k_G, X_G \uplus X_I \rangle \quad \text{where} \\
P'_D &= P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}) \\
X_D &= \mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H \\
X_I &: \text{ a set of names satisfying} \\
&\qquad |X_I| = |P'_D|,\ X_I \# X_G, \text{ and } X_I \# Y_G \\
\mathit{link}'_D &: P'_D \rightarrowtail X_I \text{ a bijection} \\
\mathit{link}_D &= (\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_D
\end{aligned}
$$

$$
\begin{aligned}
\mathit{ctxt}(\phi) &\overset{\text{def}}{=} (V_C, E_C, \mathit{ctrl}_C, \mathit{prnt}_C, \mathit{link}_C) : \langle k_C, Y_G \uplus X_I \uplus X_C \rangle \to \langle m_H, Y_H \rangle \quad \text{where} \\
E_C &= E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}}) \\
X'_C &= X_H \setminus X_D \\
X_C &: \text{ a set of names satisfying} \\
&\qquad |X_C| = |X'_C|,\ X_C \# Y_G, \text{ and } X_C \# X_I \\
\alpha_C &: X_C \rightarrowtail X'_C \text{ a bijection} \\
\mathit{link}_C &= \phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C)
\end{aligned}
$$

$$
H = \mathit{ctxt}(\phi) \circ ((\phi \centerdot G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|} \otimes \alpha_C^{-1}) \circ (\pi \otimes \mathsf{id}_{X_H})
$$

Figure 4: Decomposition of bigraph $H : \langle k_H, X_H \rangle \to \langle m_H, Y_H \rangle$ into parameter $\mathit{prmt}(\phi)$ and context $\mathit{ctxt}(\phi)$ corresponding to an embedding $\phi : G \hookrightarrow H$ of a place graph $G : \langle k_G, X_G \rangle \to \langle m_G, Y_G \rangle$. The decomposition of the place graph is given in Figure 3.

Theorem 5.22 (embeddings and decompositions are isomorphic)**.** *The constructions of Def. 5.16 and Def. 5.20 are mutually inverse.*

Proof. Cf. Appendix A.1.10. □

As an instance of this result we get that redex embeddings into agents are isomorphic to matches:

Corollary 5.23 (matches isomorphic to redex embeddings into agents)**.** *In a BPRS* $\grave{}\mathrm{BG}(\grave{}\mathcal{R})$*, a match* $o = (\rho, \mathsf{id}_{X_I}, c, d)$ *of a parametric rule* $\mathsf{R} = (R : m \to \langle n, Y \rangle, R', \eta)$ *in an agent a is isomorphic to the embedding*

$$\phi = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{e}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} \uplus \phi^{\mathsf{i}} \uplus \phi^{\mathsf{o}} : R \hookrightarrow a$$

$$\phi^{\mathsf{v}} = \rho \restriction_{V_R} \qquad \phi^{\mathsf{r}} = \mathit{prnt}_c \restriction_n \qquad \phi^{\mathsf{s}} = \mathit{prnt}_d^{-1} \restriction_m$$

$$\phi^{\mathsf{e}} = \rho \restriction_{E_R} \qquad \phi^{\mathsf{o}} = \mathit{link}_c \restriction_Y \qquad \phi^{\mathsf{i}} = \emptyset.$$

Proof. Follows immediately from Prop. 5.19 and Theorem 5.22. □

The disjointness results for place graph embeddings extend to bigraph embeddings; we shall need them in Section 6.

Corollary 5.24. *Given a bigraph* $G : \langle k_G, X_G \rangle \to \langle m_G, Y_G \rangle$ *and an embedding* $\phi : G \hookrightarrow H$ *into a bigraph* $H : \langle k_H, X_H \rangle \to \langle m_H, Y_H \rangle$*. Then*

1. $\mathsf{rng}(\phi^{\mathsf{port}}) \# P_{H\downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})}}$ *and*
2. $\mathsf{rng}(\phi^{\mathsf{port}}) \# P_{H\upharpoonright_{\mathsf{rng}(\phi^{\mathsf{r}})}}$.

Proof. 1: From Lemma 5.11 we have $\mathsf{rng}(\phi^{\mathsf{v}}) \# H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})}$ so clearly $\mathsf{rng}(\phi^{\mathsf{port}}) \# P_{H\downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})}}$.

2: From Lemma 5.11 we have $\mathsf{rng}(\phi^{\mathsf{v}}) \# H \upharpoonright_{\mathsf{rng}(\phi^{\mathsf{r}})}$ so clearly $\mathsf{rng}(\phi^{\mathsf{port}}) \# P_{H\upharpoonright_{\mathsf{rng}(\phi^{\mathsf{r}})}}$. □

5.3.1 Solid Bigraphs

The results regarding solid link and place graphs of course also hold for bigraphs, i.e., embeddings of *solid* bigraphs are determined by the injection of nodes:

Definition 5.25 (solid bigraph (after [25, Def. 2.1][7]))**.** A bigraph is *solid* iff these conditions hold:

1. no roots or links are idle
2. no sites or inner names are siblings
3. every site and inner name is guarding
4. no outer name is linked to an inner name .

□

Corollary 5.26. *Given a solid bigraph* $G : \langle k_G, X_G \rangle \to \langle m_G, Y_G \rangle$ *and an embedding* $\phi : G \hookrightarrow H$ *into a bigraph* $H : \langle k_H, X_H \rangle \to \langle m_H, Y_H \rangle$*. Then* ϕ^{s}*,* ϕ^{r}*,* ϕ^{e}*,* ϕ^{i}*, and* ϕ^{o} *are uniquely determined from* ϕ^{v}*.*

Proof. Follows from Prop. 5.3 and Prop. 5.13. □

[7]These conditions are slightly stronger than those in loc. cit. cf. Sec. 5.1.1.

6 Bigraph Edit Scripts

As we have seen in the previous sections, bigraphical reactions are usually defined in terms of replacement: rewriting is performed by replacing a redex with a reactum. While this yields a simple and elegant presentation of reaction semantics, it does not capture the relation between entities in the redex and reactum, which is needed in the KaSim algorithm: we require a description of what is *modified* by a reaction rule, which implies a relation between entities before and after reaction.

In this section, we shall develop an alternative formulation of bigraphical reaction based on reconfiguration instead of replacement. The key ideas are:

reconfiguration rules:
Reconfiguration rules are fine-grained descriptions of how a reaction modifies the redex. They consist of a redex and an *edit script*: a series of minimal modifications, *edits*, to the redex which turn it into the reactum.

reaction as reconfiguration:
Exploiting that matches and embeddings are isomorphic, we define reaction as the mediation of edits to agents through embeddings.

This formulation is equivalent to the usual formulation in that it generates the same abstract reactions, but in addition it provides the notion of modification that is needed for the KaSim algorithm: edit scripts allow us to characterize causation and conflict in a fine-grained and concise way, as we already saw in the overview of KaSim (cf. Section 3.2).

In overview, the development proceeds as follows:

Section 6.1: Patterns
To simplify the development, we first introduce an alternative formulation of concrete bigraphs, where roots and sites are named instead of being consecutive numbers. We shall call these *patterns* to avoid confusion with the usual concrete bigraphs.

We also recast BPRSs and bigraph embeddings to this setting where redexes and reactums are patterns.

Section 6.2: Edits
We introduce a set of minimal *edits* and define how they reconfigure *compatible* redexes, i.e., redexes that have a suitable structure for the edit to be meaningful.

Next, we show how we can extract an instantiation map from an edit, which is necessary in order to relate edits to reaction rules.

Finally, we transfer edits to agents, by defining how an embedding of a redex can *mediate* an edit to the agent, and show that such mediated edits correspond to abstract reactions in certain BPRSs.

Section 6.3: Edit Scripts
Reaction rules cannot in general be expressed as a single edit, but require a sequence of edits, i.e., an *edit script*. We show how the concepts and results for edits transfer to such edit scripts.

Section 6.4: Reconfiguration Systems
Putting the above developments together, we define *reconfiguration rules* and *reconfiguration systems (RCSs)*. We give constructions between RCSs and BPRSs that preserve and reflect abstract reactions.

6.1 Patterns

A pattern is an alternative representation of a concrete bigraph where roots and sites are named. To avoid confusion with the names of the link graph, we shall call these identifiers *variables*. We shall assume a countably infinite set $\mathcal{U}$ of variables, ranged over by $q, r \in Q, R$ and disjoint from $\mathcal{X}$, $\mathcal{V}$, and $\mathcal{E}$.

Definition 6.1 (pattern)**.** A *pattern*

$$\tilde{P} = (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}) : \langle Q, X\rangle \to \langle R, Y\rangle$$

is an alternative representation of the concrete bigraph

$$[\![\tilde{P}]\!] \stackrel{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{[\![\tilde{P}]\!]}, link_{\tilde{P}}) : \langle |Q|, X\rangle \to \langle |R|, Y\rangle$$

where

$$\begin{aligned}
prnt_{\tilde{P}} &: Q \uplus V_{\tilde{P}} \to V_{\tilde{P}} \uplus R,\\
Q &= \{q_0, \ldots, q_{k-1}\} \qquad \text{where } \forall i \in [0; k-2] : q_i < q_{i+1},\\
R &= \{r_0, \ldots, r_{m-1}\} \qquad \text{where } \forall i \in [0; m-2] : r_i < r_{i+1},\\
prnt_{[\![\tilde{P}]\!]} &= (\mathsf{Id}_{V_{\tilde{P}}} \uplus \{r_0 \mapsto 0, \ldots, r_{m-1} \mapsto m-1\})\\
&\quad \circ prnt_{\tilde{P}}\\
&\quad \circ (\mathsf{Id}_{V_{\tilde{P}}} \uplus \{0 \mapsto q_0, \ldots, k-1 \mapsto q_{k-1}\}).
\end{aligned}$$

□

As we shall see when we define edits, the virtue of patterns is that we can add and remove sites without having to renumber those that remain. Note that there are infinitely many patterns that correspond to a concrete bigraph:

Proposition 6.2. *Two patterns $\tilde{P}$,$\tilde{P}'$ differ only by order preserving bijections on their outer and inner variables respectively iff $[\![\tilde{P}]\!] = [\![\tilde{P}']\!]$.*

Proof. Immediate from the definition. □

Parametric reaction rules, BPRSs, and bigraph embeddings are easily adapted to patterns:

Definition 6.3 (pattern rule)**.** A *pattern rule*

$$\mathsf{R} = (\tilde{P} : Q \to \langle R, Y\rangle, \tilde{P}' : Q' \to \langle R, Y\rangle, \eta : Q' \to Q)$$

where η is a function called the *variable instance map*, is an alternative representation of the parametric bigraphical reaction rule

$$[\![\mathsf{R}]\!] \stackrel{\text{def}}{=} ([\![\tilde{P}]\!] : |Q| \to \langle |R|, Y\rangle, [\![\tilde{P}']\!] : |Q'| \to \langle |R|, Y\rangle, [\![\eta]\!] : |Q'| \to |Q|)$$

where

$$\begin{aligned}
Q &= \{q_0, \ldots, q_{k-1}\} && \text{where } \forall i \in [0; k-2] : q_i < q_{i+1},\\
Q' &= \{q'_0, \ldots, q'_{m-1}\} && \text{where } \forall i \in [0; m-2] : q'_i < q'_{i+1},\\
[\![\eta]\!](i) &= j && \text{if } \eta(q'_i) = q_j.
\end{aligned}$$

□

Definition 6.4 (pattern-based BPRS)**.** A *pattern-based BPRS* over $\mathcal{K}$ with pattern rules $\grave{\mathcal{R}}$, written $\grave{\text{B}}\text{G}(\mathcal{K}, \grave{\mathcal{R}})$, is the BPRS $\grave{\text{B}}\text{G}(\mathcal{K}, \grave{\mathcal{R}}')$ where $\grave{\mathcal{R}}' = \{[\![\mathsf{R}]\!] \mid \mathsf{R} \in \grave{\mathcal{R}}\}$. □

		Affected Entity Type			
		PORT	NODE	EDGE	SITE
Effect	REBIND	$\odot_{(v,i)\mapsto l}$			
	CHANGE CONTROL		$\circledcirc_{v:K}$		
	ADD		$\oplus_{v:K_{\vec{y}}@p}$	$\oplus_{e}$	
	DELETE		$\ominus_{v}$	$\ominus_{e}$	$\ominus_{q}$
	MOVE		$\oslash_{v@p}$		$\oslash_{q@p}$
	COPY				$\otimes_{q\to r@p}$

Table 1: The set of edits organized by their effect (rows) and the type of entity they affect (columns) ($v \in \mathcal{V}$, $K \in \mathcal{K}$, $\{\vec{y}\} \subseteq \mathcal{X}$, $p \in \mathcal{V} \uplus \mathcal{U}$, $e \in \mathcal{E}$, $q, r \in \mathcal{U}$,$i \in \mathbb{N}$, and $l \in \mathcal{E} \uplus \mathcal{X}$).

Definition 6.5 (pattern embedding)**.** Let $\tilde{P} : \langle Q_{\tilde{P}}, X_{\tilde{P}}\rangle \to \langle R_{\tilde{P}}, Y_{\tilde{P}}\rangle$ and $\tilde{H} : \langle Q_{\tilde{H}}, X_{\tilde{H}}\rangle \to \langle R_{\tilde{H}}, Y_{\tilde{H}}\rangle$ be patterns. Then a *pattern embedding*, written $\phi : \tilde{P} \hookrightarrow \tilde{H}$, is a map $\phi : |\tilde{P}| \uplus Q_{\tilde{P}} \uplus R_{\tilde{P}} \uplus X_{\tilde{P}} \uplus Y_{\tilde{P}} \to |\tilde{H}| \uplus \mathcal{P}(Q_{\tilde{H}} \uplus V_{\tilde{H}}) \uplus R_{\tilde{H}} \uplus \mathcal{P}(X_{\tilde{H}} \uplus P_{\tilde{H}}) \uplus Y_{\tilde{H}}$ which is an alternative representation of the bigraph embedding

$$\llbracket\phi\rrbracket : \llbracket\tilde{P}\rrbracket \hookrightarrow \llbracket\tilde{H}\rrbracket$$

where the constituent maps are defined as:

$$\begin{aligned}
\llbracket\phi\rrbracket^{*} &\overset{\text{def}}{=} \phi^{*} && \text{where } * \in \{\mathsf{v}, \mathsf{e}, \mathsf{i}, \mathsf{o}\}\\
\llbracket\phi\rrbracket^{\mathsf{s}} &\overset{\text{def}}{=} (\mathsf{Id}_{V_{\tilde{H}}} \uplus \{q_{\tilde{H},0} \mapsto 0, \ldots, q_{\tilde{P},k_{\tilde{H}}-1} \mapsto k_{\tilde{H}} - 1\})\\
&\qquad \circ\, \phi^{\mathsf{s}} \circ \{0 \mapsto q_{\tilde{P},0}, \ldots, k_{\tilde{P}} - 1 \mapsto q_{\tilde{P},k_{\tilde{P}}-1}\}\\
\llbracket\phi\rrbracket^{\mathsf{r}} &\overset{\text{def}}{=} (\mathsf{Id}_{V_{\tilde{H}}} \uplus \{r_{\tilde{H},0} \mapsto 0, \ldots, r_{\tilde{P},m_{\tilde{H}}-1} \mapsto m_{\tilde{H}} - 1\})\\
&\qquad \phi^{\mathsf{r}} \circ \{0 \mapsto r_{\tilde{P},0}, \ldots, m_{\tilde{P}} - 1 \mapsto r_{\tilde{P},m_{\tilde{P}}-1}\}\\
Q_{\tilde{P}} &= \{q_{\tilde{P},0}, \ldots, q_{\tilde{P},k_{\tilde{P}}-1}\} && \text{where } \forall i \in [0; k_{\tilde{P}} - 2] : q_{\tilde{P},i} < q_{\tilde{P},i+1}\\
R_{\tilde{P}} &= \{r_{\tilde{P},0}, \ldots, r_{\tilde{P},m_{\tilde{P}}-1}\} && \text{where } \forall i \in [0; m_{\tilde{P}} - 2] : r_{\tilde{P},i} < r_{\tilde{P},i+1}\\
Q_{\tilde{H}} &= \{q_{\tilde{H},0}, \ldots, q_{\tilde{H},k_{\tilde{H}}-1}\} && \text{where } \forall i \in [0; k_{\tilde{H}} - 2] : q_{\tilde{H},i} < q_{\tilde{H},i+1}\\
R_{\tilde{H}} &= \{r_{\tilde{H},0}, \ldots, r_{\tilde{H},m_{\tilde{H}}-1}\} && \text{where } \forall i \in [0; m_{\tilde{H}} - 2] : r_{\tilde{H},i} < r_{\tilde{H},i+1}.
\end{aligned}$$

Clearly, it is easy to define embeddings between patterns and concrete bigraphs in a similar manner, and we shall freely use such embeddings. □

6.2 Edits

An *edit* is a minimal reconfiguration of a pattern. We are concerned with edits that correspond to reactions and shall therefore only consider edits of redexes, i.e., patterns with no inner names, which preserve the outer face.

Definition 6.6 (edits)**.** An *edit* δ over a signature $\mathcal{K}$ is any of the operators in Table 1. The application of an edit δ to a pattern $\tilde{P}$, written $\delta(\tilde{P})$, is defined in Table 2. We say that an edit δ is *compatible* with a pattern $\tilde{P}$ iff $\delta(\tilde{P})$ is defined. □

It is straightforward to verify that edits yield patterns:

Rebind a port:

$$\odot_{(v,i)\mapsto l}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}, \mathit{link}_{\tilde{P}}[(v,i)\mapsto l]) \quad : \quad Q \to \langle R, Y\rangle$$
if $v \in V_{\tilde{P}}$ and $i \in ar(\mathit{ctrl}_{\tilde{P}}(v))$ and $l \in E_{\tilde{P}} \uplus Y$

Change a control:

$$\circledcirc_{v:K}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}[v\mapsto K], \mathit{prnt}_{\tilde{P}}, \mathit{link}_{\tilde{P}}) \quad : \quad Q \to \langle R, Y\rangle$$
if $v \in V_{\tilde{P}}$ and $ar(K) = ar(\mathit{ctrl}_{\tilde{P}}(v))$

Add node or edge:

$$\oplus_{v:K_{\vec{y}}@p}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}} + v, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}[v\mapsto K], \mathit{prnt}_{\tilde{P}}[v\mapsto p], \mathit{link}_{\tilde{P}}[(v,0)\mapsto \vec{y}_0, \ldots, (v,n-1)\mapsto \vec{y}_{n-1}]) \quad : \quad Q \to \langle R, Y\rangle$$
if $v \notin V_{\tilde{P}}$ and $p \in V_{\tilde{P}} \uplus R$ and $\{\vec{y}\} \subseteq E_{\tilde{P}} \uplus Y$ and $ar(K) = n$

$$\oplus_{e}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}} + e, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}, \mathit{link}_{\tilde{P}}) \quad : \quad Q \to \langle R, Y\rangle$$
if $e \notin E_{\tilde{P}}$

Delete node, edge, or site:

$$\ominus_{v}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}} - v, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}} - v, \mathit{prnt}_{\tilde{P}} - v, \mathit{link}_{\tilde{P}} - P_v) : \quad Q \to \langle R, Y\rangle$$
if $v \in V_{\tilde{P}}$ and $\mathit{prnt}_{\tilde{P}}^{-1}(v) = \emptyset$

$$\ominus_{e}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}} - e, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}, \mathit{link}_{\tilde{P}}) \quad : \quad Q \to \langle R, Y\rangle$$
if $e \in E_{\tilde{P}}$ and $\mathit{link}_{\tilde{P}}^{-1}(e) = \emptyset$

$$\ominus_{q}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}} - q, \mathit{link}_{\tilde{P}}) \quad : (Q - q) \to \langle R, Y\rangle$$
if $q \in Q$

Move node or site:

$$\oslash_{v@p}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[v\mapsto p], \mathit{link}_{\tilde{P}}) \quad : \quad Q \to \langle R, Y\rangle$$
if $v \in V_{\tilde{P}}$ and $p \in V_{\tilde{P}} \uplus R$

$$\oslash_{q@p}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[q\mapsto p], \mathit{link}_{\tilde{P}}) \quad : \quad Q \to \langle R, Y\rangle$$
if $q \in Q$ and $p \in V_{\tilde{P}} \uplus R$

Copy site:

$$\otimes_{q\to r@p}(\tilde{P}) \overset{\text{def}}{=} (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[r\mapsto p], \mathit{link}_{\tilde{P}}) \quad : (Q + r) \to \langle R, Y\rangle$$
if $q \in Q$ and $r \notin Q$ and $p \in V_{\tilde{P}} \uplus R$

Table 2: Application of edit δ to pattern $\tilde{P} : Q \to \langle R, Y\rangle$, denoted $\delta(\tilde{P})$.

Proposition 6.7 (edits)**.** *Given a pattern $\tilde{P} : Q \to \langle R, Y \rangle$ and a compatible edit δ. Then $\delta(\tilde{P}) : Q' \to \langle R, Y \rangle$ is a pattern and*

$$Q' = \begin{cases} Q - q & \textit{if } \delta = \ominus_q \\ Q + r & \textit{if } \delta = \otimes_{q \to r@p} \\ Q & \textit{otherwise.} \end{cases}$$

Note that other choices of edits are possible. For instance, one could imagine allowing deletion of nodes that have children – but what should happen to the children? Should they also be deleted or should they become children of the deleted node's parent? We have chosen the above set of edits as they express minimal modifications to each of the elements of a concrete bigraph quintuple which result in another concrete bigraph. As we shall see below, the chosen set of edits is sufficient for completeness, so we leave it to future work to explore other sets of edits.

6.2.1 Deriving Named Instance Maps

To relate edits to reaction rules and reaction, we must define how edits relate to instantiations. There are two important things to take into account when we define the instantiations for edits: (a) edits are not tied to a specific pattern and thus the corresponding instance map should be the identity on all sites except those that are affected by the edit; and (b) edits will be chained together into edit scripts and thus the corresponding instance maps should be easily composable.

We meet these criteria by defining instance maps for edits in two steps:

forward instance map: A map which is defined for all compatible patterns and describes how sites are modified by an edit.

derived instance map: From the forward map we derive the variable instance maps for specific compatible patterns.

As we shall see when we get to edit scripts, the forward instance maps compose easily and we can reuse the derivation of the pattern specific instance maps.

Definition 6.8 (forward instance map)**.** A *forward instance map* $\digamma : \mathcal{U} \to \mathcal{P}(\mathcal{U})$ is a fully injective map on named sites.

The forward instance map $\mathit{finst}(\delta)$ corresponding to an edit δ is

$$\begin{aligned} \mathit{finst}(\ominus_q) &\stackrel{\text{def}}{=} \mathsf{Id}_{\mathcal{U}}[q \mapsto \emptyset] \\ \mathit{finst}(\otimes_{q \to r@a}) &\stackrel{\text{def}}{=} \mathsf{Id}_{\mathcal{U}-r}[q \mapsto \{q, r\}] \\ \mathit{finst}(\delta) &\stackrel{\text{def}}{=} \mathsf{Id}_{\mathcal{U}} \qquad \text{in all other cases} \end{aligned}$$

□

It is clear from this definition and the definition of edits that a forward instance map corresponding to an edit maps the sites of compatible patterns to the sites of the resulting patterns:

Proposition 6.9 (forward instance map)**.** *Given a pattern $\tilde{P} : Q \to J$ and a compatible edit δ. Then $Q \subseteq \mathsf{dom}(\mathit{finst}(\delta))$ and $\mathit{finst}(\delta)(Q) = Q'$ where $\delta(\tilde{P}) : Q' \to J$.*

To derive an instance map for a particular compatible pattern, we simply restrict the domain of the forward instance map and then invert it:

Definition 6.10 (derived variable instance map)**.** The *derived variable instance map* $\mathit{inst}_Q(\digamma)$ corresponding to a forward instance map $\digamma$ for a set of named sites $Q \subseteq \mathsf{dom}(\digamma)$ is

$$\mathit{inst}_Q(\digamma) \stackrel{\text{def}}{=} (\digamma \restriction_Q)^{-1}.$$

Note that the inverse of the forward instance map $\digamma^{-1}$, and thus $\mathit{inst}_Q(\digamma)$, is a function since $\digamma$ is fully injective. □

Corollary 6.11 (derived variable instance map)**.** *Given a pattern $\tilde{P} : Q \to J$ and a compatible edit δ. Then $inst_Q(finst(\delta)) : Q' \to Q$ where $\delta(\tilde{P}) : Q' \to J$.*

Proof. Follows from Def. 6.10 and Prop. 6.9. □

6.2.2 Mediating edits

In order to realize reactions by using edits, we must define how an embedding of a pattern can mediate the edit of an agent. Since we wish to combine sequences of edits, the result of a mediated edit should be both a new agent and a new embedding that can mediate the next edit into the new agent. Furthermore, we shall define mediation of edits through embeddings into arbitrary bigraphs as we shall need this to characterize conflict and causality in Section 7.

Definition 6.12 (mediated edits)**.** For a pattern $\tilde{P}$ and compatible edit δ, the *mediated edit*, written $\delta(a, \phi)$, of a pattern $\tilde{H}$ through an embedding $\phi : \tilde{P} \hookrightarrow \tilde{H}$ is defined in Table 3. When $(\tilde{H}', \phi') = \delta(\tilde{H}, \phi)$ we shall often abuse the notation and write $\delta(\tilde{H}, \phi)$ for $\tilde{H}'$. □

Proposition 6.13 (mediated edits)**.** *Given a pattern $\tilde{P} : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle$, a compatible edit δ, and an embedding $\phi : \tilde{P} \hookrightarrow \tilde{H}$ into a pattern $\tilde{H} : \langle Q_{\tilde{H}}, X_{\tilde{H}} \rangle \to I$. Then $\tilde{H}' : \langle Q'_{\tilde{H}}, X_{\tilde{H}} \rangle \to I$ is a pattern and $\phi' : \tilde{P}' \hookrightarrow \tilde{H}'$ is an embedding, where $(\tilde{H}' : \langle Q'_{\tilde{H}}, X_{\tilde{H}} \rangle \to I, \phi') = \delta(\tilde{H}, \phi)$ and $\tilde{P}' = \delta(\tilde{P})$ for some set of variables $Q'_{\tilde{H}}$.*

Proof. Cf. Appendix A.2.1. □

Mediated edits are well-behaved in the sense that the reconfigurations they cause in agents can also be expressed as reactions:

Lemma 6.14 (mediated edits are reactions)**.** *Given a pattern $\tilde{P} : Q \to \langle R, Y \rangle$, a compatible edit δ, and an embedding $\phi : \tilde{P} \hookrightarrow a$ into a concrete agent $a : \langle m_a, Y_a \rangle$. Then $a \twoheadrightarrow a'$, where $(a', \phi') = \delta(a, \phi)$, is a reaction in any pattern-based BPRS containing the rule $(\tilde{P}, \delta(\tilde{P}), inst_Q(finst(\delta)))$.*

Proof. Cf. Appendix A.2.2. □

The converse does not hold for two reasons:

- A single edit of course cannot express any reaction. We will solve this by introducing edit scripts in the next section.
- Edits are much more restrictive in their handling of support: we cannot change support arbitrarily, but are only free to choose support for added/copied nodes and edges.

We can, however, show that any abstract reaction can be realized by edits. In general, this requires edit scripts, but let us first show that abstract reactions generated by rules derived from edits can also be obtained through mediated edits:

Lemma 6.15. *Given a pattern-based BPRS and a reaction $a \twoheadrightarrow a'$ generated by some pattern rule $\mathsf{R} = (\tilde{P} : Q \to \langle R, Y \rangle, \delta(\tilde{P}) : Q' \to \langle R, Y \rangle, inst_Q(finst(\delta)))$. Then there is an agent a'' and embeddings $\phi : \tilde{P} \hookrightarrow a$, $\phi' : \delta(\tilde{P}) \hookrightarrow a''$ such that $a' \simeq a''$ and $(a'', \phi') = \delta(a, \phi)$.*

Proof. Cf. Appendix A.2.3. □

Rebind a port:

$$\odot_{(v,i)\mapsto l}(\tilde{H},\phi) \stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}, \mathit{prnt}_{\tilde{H}}, \mathit{link}_{\tilde{H}}[(\phi(v),i)\mapsto\phi(l)]),\phi)$$

Change a control:

$$\circledcirc_{v:K}(\tilde{H},\phi) \stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}[\phi(v)\mapsto K], \mathit{prnt}_{\tilde{H}}, \mathit{link}_{\tilde{H}}),\phi)$$

Add node or edge:

$$\begin{aligned}
\oplus_{v:K_{\vec{y}}@p}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}+v', E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}[v'\mapsto K], \mathit{prnt}_{\tilde{H}}[v'\mapsto\phi(p)],\\
&\qquad \mathit{link}_{\tilde{H}}[(v',0)\mapsto\phi(\vec{y}_0),\ldots,(v',n-1)\mapsto\phi(\vec{y}_{n-1})]),\\
&\qquad \phi[v\mapsto v'])\\
&\quad \text{if } v'\notin V_{\tilde{H}}\\
\oplus_{e}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}+e', \mathit{ctrl}_{\tilde{H}}, \mathit{prnt}_{\tilde{H}}, \mathit{link}_{\tilde{H}}),\phi[e\mapsto e'])\\
&\quad \text{if } e'\notin E_{\tilde{H}}
\end{aligned}$$

Delete node, edge, or parameter:

$$\begin{aligned}
\ominus_{v}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}-\phi(v), E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}-\phi(v), \mathit{prnt}_{\tilde{H}}-\phi(v), \mathit{link}_{\tilde{H}}-P_{\phi(v)}),\phi-v)\\
\ominus_{e}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}-\phi(e), \mathit{ctrl}, \mathit{prnt}_{\tilde{H}}, \mathit{link}_{\tilde{H}}),\phi-e)\\
\ominus_{q}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}\setminus\tilde{H}\downharpoonleft^{\phi(q)}, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}-\tilde{H}\downharpoonleft^{\phi(q)}, \mathit{prnt}_{\tilde{H}}-\tilde{H}\downharpoonleft^{\phi(q)}, \mathit{link}_{\tilde{H}}-P_{\tilde{H}\downharpoonleft^{\phi(q)}})\\
&\qquad :\langle Q\setminus\tilde{H}\downharpoonleft^{\phi(q)}, X\rangle\to I,\\
&\qquad \phi-q)
\end{aligned}$$

Move node or parameter:

$$\begin{aligned}
\oslash_{v@p}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}, \mathit{prnt}_{\tilde{H}}[\phi(v)\mapsto\phi(p)], \mathit{link}_{\tilde{H}}),\phi)\\
\oslash_{q@p}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}, \mathit{prnt}_{\tilde{H}}[\phi(q)\mapsto\phi(p)], \mathit{link}_{\tilde{H}}),\phi)
\end{aligned}$$

Copy parameter:

$$\begin{aligned}
\otimes_{q\to r@p}(\tilde{H},\phi) &\stackrel{\text{def}}{=} ((V_{\tilde{H}}\uplus V_r, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}\uplus\mathit{ctrl}_r, \mathit{prnt}_{\tilde{H}}\uplus\mathit{prnt}_r, \mathit{link}_{\tilde{H}}\uplus\mathit{link}_r),\\
&\qquad :\langle Q\uplus Q_r, X\rangle\to I,\\
&\qquad \phi[r\mapsto f^{-1}(\phi(q))])\\
&\qquad \text{where} \quad
\begin{aligned}
V_q &= \tilde{H}\downharpoonleft^{\phi(q)}\cap V_{\tilde{H}} & Q_q &= \tilde{H}\downharpoonleft^{\phi(q)}\cap Q\\
|V_r| &= |V_q| & |Q_r| &= |Q_q|\\
V_r &\# V_{\tilde{H}} & Q_r &\# Q\\
f_v &: V_r\rightarrowtail V_q & f_s &: Q_r\rightarrowtail Q_q\\
f &= f_v\uplus f_s & &\\
\mathit{ctrl}_r &= \mathit{ctrl}_{\tilde{H}}\circ f_v & &\\
\mathit{prnt}_r &= \{f^{-1}(\phi(q))\mapsto\phi(p)\}\uplus f_v^{-1}\circ\mathit{prnt}_{\tilde{H}}\circ(f-f^{-1}(\phi(q))) & &\\
\mathit{link}_r(v,i) &= \mathit{link}_{\tilde{H}}(f_v(v),i)\quad(v\in V_r) & &
\end{aligned}
\end{aligned}$$

Table 3: Mediating compatible edit δ of $\tilde{P}$ to $\tilde{H}:\langle Q,X\rangle\to I$ through embedding $\phi:\tilde{P}\hookrightarrow\tilde{H}$. Interfaces are omitted for clarity in all but the two cases where they change.

6.3 Edit Scripts

In the previous section we took care to define the constructions such that they could easily be transferred to sequences of edits. In this section we reap the benefits: we transfer the concepts and results from edits to edit scripts without further comment.

Definition 6.16 (edit script)**.** An *edit script* is a finite sequence of edits $\Delta = \delta_1 \cdots \delta_n$. An edit script is *compatible* with a pattern $\tilde{P}$ iff δ_1 is compatible with $\tilde{P}$ *and* $n = 1$ or $\delta_2 \cdots \delta_n$ is compatible with $\delta(\tilde{P})$. The *application* of an edit script Δ to a compatible pattern $\tilde{P}$ is defined to be

$$\Delta(\tilde{P}) \overset{\text{def}}{=} \delta_n(\cdots \delta_1(\tilde{P}) \cdots).$$

Similarly, the *mediated application* of an edit script Δ to a pattern $\tilde{H}$ through an embedding $\phi : \tilde{P} \hookrightarrow \tilde{H}$ of a compatible pattern $\tilde{P}$ is defined to be

$$\Delta(\tilde{H}, \phi) \overset{\text{def}}{=} \delta_n(\cdots \delta_1(\tilde{H}, \phi) \cdots).$$

When $(\tilde{H}', \phi') = \Delta(\tilde{H}, \phi)$ we shall often abuse the notation and write $\Delta(\tilde{H}, \phi)$ for $\tilde{H}'$. □

Corollary 6.17 (edit scripts)**.** *Given a pattern $\tilde{P} : Q \to \langle R, Y \rangle$ and a compatible edit script Δ. Then $\Delta(\tilde{P}) : Q' \to \langle R, Y \rangle$ is a pattern.*

Proof. Follows from Prop. 6.7 by straightforward induction on the length of Δ. □

Corollary 6.18 (mediated edit scripts)**.** *Given a pattern $\tilde{P} : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle$, a compatible edit script Δ, and an embedding $\phi : \tilde{P} \hookrightarrow \tilde{H}$ into a pattern $\tilde{H} : \langle Q_{\tilde{H}}, X_{\tilde{H}} \rangle \to I$. Then $\tilde{H}' : \langle Q'_{\tilde{H}}, X_{\tilde{H}} \rangle \to I$ is an agent and $\phi' : \tilde{P}' \hookrightarrow \tilde{H}'$ is an embedding, where $(\tilde{H}' : \langle Q'_{\tilde{H}}, X_{\tilde{H}} \rangle \to I, \phi') = \Delta(\tilde{H}, \phi)$ and $\tilde{P}' = \Delta(\tilde{P})$ for some set of variables $Q'_{\tilde{H}}$.*

Proof. Follows from Prop. 6.13 by straightforward induction on the length of Δ. □

Definition 6.19 (edit script forward instance map)**.** The *forward instance map* $\mathit{finst}(\Delta)$ corresponding to an edit script $\Delta = \delta_1 \cdots \delta_n$ is

$$\mathit{finst}(\Delta) \overset{\text{def}}{=} \mathit{finst}(\delta_n) \circ (\cdots \circ (\mathit{finst}(\delta_1) \downharpoonleft^{\mathsf{dom}(\mathit{finst}(\delta_2))}) \cdots \downharpoonleft^{\mathsf{dom}(\mathit{finst}(\delta_n))}).$$

□

Corollary 6.20 (edit script forward instance map)**.** *Given a pattern $\tilde{P} : Q \to J$ and a compatible edit script Δ. Then $Q \subseteq \mathsf{dom}(\mathit{finst}(\Delta))$ and $\mathit{finst}(\Delta)(Q) = Q'$ where $\Delta(\tilde{P}) : Q' \to J$.*

Proof. Follows from Prop. 6.9 by straightforward induction on the length of Δ. □

Corollary 6.21 (derived variable instance map)**.** *Given a pattern $\tilde{P} : Q \to J$ and a compatible edit script Δ. Then $\mathit{inst}_Q(\mathit{finst}(\Delta)) : Q' \to Q$ is a variable instance map where $\Delta(\tilde{P}) : Q' \to J$.*

Proof. Follows from Def. 6.10 and Corol. 6.20. □

Corollary 6.22 (mediated edit scripts are reactions)**.** *Given a pattern $\tilde{P} : Q \to \langle R, Y \rangle$, a compatible edit script Δ, and an embedding $\phi : \tilde{P} \hookrightarrow a$ into a concrete agent $a : \langle m_a, Y_a \rangle$. Then $a \twoheadrightarrow a'$, where $(a', \phi') = \Delta(a, \phi)$, is a reaction in any pattern-based BPRS containing the rule $(\tilde{P}, \Delta(\tilde{P}), \mathit{inst}_Q(\mathit{finst}(\Delta)))$.*

Proof. Follows from Lemma 6.14 by straightforward induction on the length of Δ. □

Corollary 6.23. *Given a pattern-based BPRS and a reaction $a \twoheadrightarrow a'$ generated by some pattern rule $\mathsf{R} = (\tilde{P} : Q \to \langle R, Y \rangle, \Delta(\tilde{P}) : Q' \to \langle R, Y \rangle, \mathit{inst}_Q(\mathit{finst}(\Delta)))$. Then there is an agent a'' and embeddings $\phi : \tilde{P} \hookrightarrow a$, $\phi' : \Delta(\tilde{P}) \hookrightarrow a''$ such that $a' \simeq a''$ and $(a'', \phi') = \Delta(a, \phi)$.*

Proof. Follows from Lemma 6.15 by straightforward induction on the length of Δ. □

6.4 Reconfiguration Systems

In the previous sections, we have seen (a) that edit scripts generate reactions and (b) that if we can express a pattern rule as an edit script, that edit script generates the same abstract reactions as the pattern rule. Thus, if any pattern rule can be expressed as an edit script, we can generate all abstract reactions through edit scripts. In this section we shall give a construction of an edit script for any pattern rule, thus providing the final piece of the puzzle.

To show the correspondence between mediated edits and reactions, we shall define *reconfiguration rules* and *reconfiguration systems (RCSs)* and show that they are equivalent to pattern rules and pattern-based BPRS.

In overview, the development proceeds as follows:

reconfiguration rules:
We first define *reconfiguration rules* as pairs of patterns and edit scripts. We give constructions of pattern rules from reconfiguration rules and vice versa.

reconfiguration systems:
Next, we define RCSs, give constructions of pattern-based BPRS from RCSs and vice versa, and show that the constructions preserve and reflect abstract reactions.

6.4.1 Reconfiguration Rules

A reconfiguration rule is simply a pattern paired with a compatible edit script:

Definition 6.24 (reconfiguration rule)**.** A *reconfiguration rule* $\tilde{R} = (\tilde{P}, \Delta)$ consists of a pattern $\tilde{P} : Q \to J$ and a compatible edit script Δ. □

From a reconfiguration rule, we can easily construct a pattern rule:

Definition 6.25 (pattern rule corresponding to reconfiguration rule)**.** The pattern rule corresponding to a reconfiguration rule $\tilde{R}$ is defined as

$$[\![\tilde{R}]\!] \overset{\text{def}}{=} (\tilde{P}, \Delta(\tilde{P}), \mathit{inst}_Q(\mathit{finst}(\Delta))).$$

□

Proposition 6.26 (pattern rule corresponding to reconfiguration rule)**.** *The pattern rule corresponding to a reconfiguration rule is indeed a pattern rule.*

Proof. Let

$$\tilde{R} = (\tilde{P} : Q \to \langle R, Y \rangle, \Delta)$$
$$[\![\tilde{R}]\!] = (\tilde{P}, \Delta(\tilde{P}), \mathit{inst}_Q(\mathit{finst}(\Delta))).$$

By Corol. 6.17 and Corol. 6.21, $\Delta(\tilde{P}) : Q' \to \langle R, Y \rangle$ is a pattern and $\mathit{inst}_Q(\mathit{finst}(\Delta)) : Q' \to Q$ is a variable instance map. □

The reverse direction is more tricky: in general, there will be infinitely many edit scripts that express a reaction, since we can always extend an edit script by two edits that cancel out, e.g., by adding and removing an edge. Here we shall give a naive construction, which first removes all the nodes, edges, and redundant sites from the redex and then builds up the reactum. In overview, the edit script will consist of the following steps:

(1) Copy the sites that will be in the reactum to a root (using fresh variables to avoid clashes.

(2) Delete the original sites.

$$es(\mathsf{R}) =$$

(1)	$\otimes_{\eta(q'_0)\to f(q'_1)@r} \cdots \otimes_{\eta(q'_{n'})\to f(q'_{n'})@r}$	copy sites as prescribed by η but using temporary names and placed at root r
(2)	$\ominus_{q_1} \cdots \ominus_{q_n}$	delete redex sites
(3)	$\ominus_{v_1} \cdots \ominus_{v_k}$	delete redex nodes
(4)	$\ominus_{e_1} \cdots \ominus_{e_m}$	delete redex edges
(5)	$\oplus_{e'_1} \cdots \oplus_{e_{m'}}$	add reactum edges
(6)	$\oplus_{v'_1 : \mathit{ctrl}_{\tilde{P}'}(v'_1)_{[\ldots,\mathit{link}_{\tilde{P}'}(v'_1,i),\ldots]} @\, \mathit{prnt}_{\tilde{P}'}(v'_1)} \cdots \oplus_{v'_{k'} : \mathit{ctrl}_{\tilde{P}'}(v'_{k'})_{[\ldots,\mathit{link}_{\tilde{P}'}(v'_{k'},i),\ldots]} @\, \mathit{prnt}_{\tilde{P}'}(v'_{k'})}$	add reactum nodes
(7)	$\otimes_{f(q'_1)\to q'_1 @\, \mathit{prnt}_{\tilde{P}'}(q'_1)} \cdots \otimes_{f(q'_{n'})\to q'_{n'} @\, \mathit{prnt}_{\tilde{P}'}(q'_{n'})}$	copy sites to their proper places with proper names
(8)	$\ominus_{f(q'_1)} \cdots \ominus_{f(q'_{n'})}$	delete temporary sites

for some $r \in R$ and set of variables $Q'' \,\#\, Q \cup Q'$ in bijection to Q', i.e., $f : Q' \rightarrowtail Q''$, and assuming the following sequencing of the entities of $\tilde{P}$ and $\tilde{P}'$:

$$\begin{aligned}
Q &= \{q_1, \ldots, q_n\} & Q' &= \{q'_1, \ldots, q'_{n'}\} \\
E_{\tilde{P}} &= \{e_1, \ldots, e_m\} & E_{\tilde{P}'} &= \{e'_1, \ldots, e'_{m'}\} \\
V_{\tilde{P}} &= \{v_1, \ldots, v_k\} & &\text{where } \forall i, j \in [1;k] : \mathit{prnt}_{\tilde{P}}(v_i) = v_j \Rightarrow i < j \\
V_{\tilde{P}'} &= \{v'_1, \ldots, v'_{k'}\} & &\text{where } \forall i, j \in [1;k'] : \mathit{prnt}_{\tilde{P}'}(v'_i) = v'_j \Rightarrow i > j.
\end{aligned}$$

Figure 5: Naive construction of edit script from pattern rule $\mathsf{R} = (\tilde{P} : Q \to \langle R, Y\rangle, \tilde{P}' : Q' \to \langle R, Y\rangle, \eta : Q' \to Q)$, denoted by $es(\mathsf{R})$.

(3) Delete all nodes (deleting children before their parents, i.e., bottom-up).

(4) Delete all edges.

(5) Add the edges of the reactum.

(6) Add the nodes of the reactum (adding parents before their children).

(7) Copy the sites to their proper place in the reactum (assigning the proper variable to the copy).

(8) Delete the sites created in step (1).

The formal construction is given by the following definition:

Definition 6.27 (naive pattern rule edit script)**.** For a pattern rule R the *corresponding naive edit script*, written $es(\mathsf{R})$, is defined in Figure 5. □

Proposition 6.28 (naive pattern rule edit script)**.** *Given a pattern rule* $\mathsf{R} = (\tilde{P} : Q \to J, \tilde{P}' : Q' \to J, \eta)$*. Then* $es(\mathsf{R})$ *is compatible with* $\tilde{P}$*,* $es(\mathsf{R})(\tilde{P}) = \tilde{P}'$*, and* $\mathit{inst}_Q(\mathit{finst}(es(\mathsf{R}))) = \eta$*.*

Proof. Cf. Appendix A.2.4. □

Thus, for any parametric reaction rule we can construct a corresponding reconfiguration rule:

Definition 6.29 (reconfiguration rule corresponding to pattern rule)**.** The reconfiguration rule corresponding to a pattern rule R is defined as

$$(\tilde{P}, es(\mathsf{R})).$$

□

Proposition 6.30 (reconfiguration rule corresponding to pattern rule)**.** *The reconfiguration rule corresponding to a pattern rule is indeed a reconfiguration rule.*

Proof. Follows immediately from Def. 6.24 and Prop. 6.28. □

6.4.2 Reconfiguration Systems

Let us now give an alternative formulation of BPRSs based on reconfiguration rules:

Definition 6.31 (reconfiguration systems (RCS))**.** A *reconfiguration system (RCS)* over $\mathcal{K}$, written $\grave{}\mathrm{B}\textsc{g}(\mathcal{K}, \tilde{\mathcal{R}})$, consists of the s-category $\grave{}\mathrm{B}\textsc{g}(\mathcal{K})$ equipped with a set $\tilde{\mathcal{R}}$ of reconfiguration rules.

The *reaction relation* $\longrightarrow\!\!\triangleright$ over agents a, a' is the smallest such that $a \longrightarrow\!\!\triangleright\, a'$ whenever $\phi : \tilde{P} \hookrightarrow a$ is a match of some reconfiguration rule $\tilde{R} = (\tilde{P}, \Delta) \in \tilde{\mathcal{R}}$ in a and $(a', \phi') = \Delta(a, \phi)$ for some embedding ϕ'. □

From the results for edit scripts, it should be clear that RCSs have the same abstract reaction relations as pattern-based BPRSs. We shall now show this formally.

First, we can construct a pattern-based BPRS with the same abstract reactions as an RCS:

Definition 6.32 (BPRS corresponding to RCS)**.** The pattern-based BPRS *corresponding* to an RCS $\grave{}\mathrm{B}\textsc{g}(\mathcal{K}, \tilde{\mathcal{R}})$ is

$$\grave{}\mathrm{B}\textsc{g}(\mathcal{K}, \{[\![\tilde{R}]\!] \mid \tilde{R} \in \tilde{\mathcal{R}}\}).$$

□

Proposition 6.33 (BPRS corresponding to RCS)**.** *The pattern-based BPRS corresponding to an RCS is indeed a pattern-based BPRS.*

Proof. Follows immediately from Def. 6.4 and Prop. 6.26. □

Theorem 6.34 (abstract reaction equivalence of BPRS corresponding to RCS)**.** *Given a reaction $a \longrightarrow\!\!\triangleright_r a'$ in an RCS, then the corresponding pattern-based BPRS has the same reaction. Conversely, for any reaction $a \longrightarrow\!\!\triangleright_p a'$ in the corresponding BPRS, there is an agent a'' such that $a' \simeq a''$ and $a \longrightarrow\!\!\triangleright_r a''$ in the RCS.*

Proof. $\Rightarrow$: Assume a reaction $a \longrightarrow\!\!\triangleright_r a'$ in an RCS, i.e., there is some match $\phi : \tilde{P} \hookrightarrow a$ of a reconfiguration rule $\tilde{R} = (\tilde{P}, \Delta) \in \tilde{\mathcal{R}}$ in a and $(a', \phi') = \Delta(a, \phi)$ for some embedding ϕ'. By Def. 6.32 and Def. 6.25 the corresponding pattern-based BPRS contains the rule $(\tilde{P}, \Delta(\tilde{P}), \mathit{inst}_Q(\mathit{finst}(\Delta)))$, so, by Corol. 6.22, $a \longrightarrow\!\!\triangleright_p a'$.

$\Leftarrow$: Assume a reaction $a \longrightarrow\!\!\triangleright_p a'$ generated by some pattern rule $(\tilde{P}, \Delta(\tilde{P}), \mathit{inst}_Q(\mathit{finst}(\Delta)))$ in the pattern-based BPRS corresponding to an RCS. By Corol. 6.23, there is an agent a'' and embeddings $\phi : \tilde{P} \hookrightarrow a$, $\phi' : \Delta(\tilde{P}) \hookrightarrow a''$ such that $a' \simeq a''$ and $(a'', \phi') = \Delta(a, \phi)$. Thus, $a \longrightarrow\!\!\triangleright_r a''$. □

Conversely, for any pattern-based BPRS we can construct an RCS which has the same abstract reactions:

Definition 6.35 (RCS corresponding to BPRS)**.** The RCS *corresponding* to a pattern-based BPRS $\grave{}\mathrm{B}\textsc{g}(\mathcal{K}, \grave{}\mathcal{R})$ is

$$\grave{}\mathrm{B}\textsc{g}(\mathcal{K}, \{(\tilde{P}, es(\mathsf{R})) \mid \mathsf{R} \in \grave{}\mathcal{R}\}).$$

□

Proposition 6.36 (RCS corresponding to BPRS)**.** *The RCS corresponding to a pattern-based BPRS is indeed an RCS.*

Proof. Follows immediately from Def. 6.31 and Prop. 6.30. $\square$

Theorem 6.37 (abstract reaction equivalence of RCS corresponding to BPRS)**.** *Given a reaction $a \rightarrowtriangle_p a'$ in a pattern-based BPRS, then there is an agent a'' such that $a' \simeq a''$ and $a \rightarrowtriangle_r a''$ in the corresponding RCS. Conversely, any reaction $a \rightarrowtriangle_r a'$ in the corresponding RCS is a reaction in the pattern-based BPRS.*

Proof. $\Rightarrow$: Assume a reaction $a \rightarrowtriangle_p a'$ in the pattern-based BPRS generated by some pattern rule $(\tilde{P}, \tilde{P}', \eta)$, i.e., there is a match $\phi : \tilde{P} \hookrightarrow a$. By Def. 6.35 the corresponding RCS has the reconfiguration rule $(\tilde{P}, es(\mathsf{R}))$ and, by Prop. 6.28, $es(\mathsf{R})(\tilde{P}) = \tilde{P}'$, and $inst_Q(\mathit{finst}(es(\mathsf{R}))) = \eta$. Thus, by Corol. 6.23, there is an agent a'' and embeddings $\phi : \tilde{P} \hookrightarrow a$, $\phi' : \Delta(\tilde{P}) \hookrightarrow a''$ such that $a' \simeq a''$ and $(a'', \phi') = \Delta(a, \phi)$. Thus, $a \rightarrowtriangle_r a''$.

$\Leftarrow$: Assume a reaction $a \rightarrowtriangle_r a'$ in generated by some reconfiguration rule $(\tilde{P}, es(\mathsf{R}))$ in the RCS corresponding to a pattern-based BPRS with $\mathsf{R} = (\tilde{P}, \tilde{P}', \eta)$. By Prop. 6.28, $\mathsf{R} = (\tilde{P}, es(\mathsf{R})(\tilde{P}), inst_Q(\mathit{finst}(es(\mathsf{R}))))$, so, by Corol. 6.22, $a \rightarrowtriangle_p a'$. $\square$

7 Rule Activation and Inhibition

The KaSim algorithm presumes that we can characterize causality and conflict at the level of reaction rules: it requires two relations over reaction rules, activation $\mathsf{R}_0 \prec \mathsf{R}_1$ and inhibition $\mathsf{R}_0 \mathbin{\#} \mathsf{R}_1$, capturing whether a reaction using R_0 can cause or prevent, respectively, reactions using R_1. These relations reduce the number of rules that must be considered in the positive and negative update phases of the algorithm.

In this section, we outline how we hope to construct these relations through a characterization of causality and conflict in terms of pullbacks and pushouts in the category of bigraph embeddings: intuitively, a pullback characterizes one way two bigraphs can overlap in a context, while the pushout of a pullback is the minimal example of such an overlap. For simplicity, we shall assume that rules are linear, i.e., they do not copy or delete parameters.

This section proceeds as follows:

1. First, we give definitions of the usual notions of causality and conflict in the contexts of reconfiguration systems and show how they can be expressed in terms of embeddings and edit scripts.

2. We then define and discuss the category of bigraph embeddings and state a number of conjectures which our approach relies on. In particular, we argue that the embedding category has pullbacks and pushout of pullbacks if we relax the embedding conditions slightly.

3. Finally, we discuss how our conjectures about the category of bigraph embeddings should allow us to characterize causality and conflict at the level of rules and thus construct the activation and inhibition relations.

7.1 Causality and Conflict

The notions of causality and conflict between events are well-studied in the literature also for graph rewriting, though sometimes through their duals, sequential and parallel independence [16, 32]: events are causally related if one must precede the other, and in conflict if one prevents the other. In this section we shall, essentially, take events to be reactions in an RCS where the reaction relation $a \mathrel{-\!\triangleright}_{\mathsf{R},\phi} a'$ is extended with labels that record the rule $\mathsf{R} = (\tilde{P}, \Delta)$ and embedding $\phi : \tilde{P} \hookrightarrow a$ that generated the reaction.

More precisely, we define causality and conflict as follows:

Definition 7.1 (causality)**.** In an RCS, say that reaction $a \mathrel{-\!\triangleright}_{\mathsf{R}_0,\phi_0} b$ causes reaction $b \mathrel{-\!\triangleright}_{\mathsf{R}_1,\phi_1} c$ iff there is no b' such that $a \mathrel{-\!\triangleright}_{\mathsf{R}_1,\phi_1} b'$. □

Definition 7.2 (conflict)**.** In an RCS, say that reaction $a \mathrel{-\!\triangleright}_{\mathsf{R}_0,\phi_0} b$ conflicts with reaction $a \mathrel{-\!\triangleright}_{\mathsf{R}_1,\phi_1} b'$ iff there is no c such that $b \mathrel{-\!\triangleright}_{\mathsf{R}_1,\phi_1} c$. □

Inspecting the definition of RCSs (Def. 6.31) it is clear that causality and conflict can be restated in terms of embeddings and edit scripts as follows:

Proposition 7.3 (causality)**.** *In an RCS, the reaction $a \mathrel{-\!\triangleright}_{\mathsf{R}_0,\phi_0} \Delta_0(a, \phi_0)$ causes the reaction $\Delta_0(a, \phi_0) \mathrel{-\!\triangleright}_{\mathsf{R}_1,\phi_1} \Delta_1(\Delta_0(a, \phi_0), \phi_1)$ iff $\phi_1 : \tilde{P}_1 \not\hookrightarrow a$, where $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ $(i = 0{,}1)$. The following diagram illustrates the situation:*

$$\begin{tikzcd}
\tilde{P}_0 \arrow[d, hook, "\phi_0"'] & \tilde{P}_1 \arrow[dl, hook, "\phi_1"', "\times" marking] \arrow[d, hook, "\phi_1"] & \\
a \arrow[r, -{Triangle[open]}] & \Delta_0(a, \phi_0) \arrow[r, -{Triangle[open]}] & \Delta_1(\Delta_0(a, \phi_0), \phi_1).
\end{tikzcd}$$

Proposition 7.4 (conflict)**.** *In an RCS, the reaction* $a \rightarrowtriangle_{\mathsf{R}_0,\phi_0} \Delta_0(a,\phi_0)$ *conflicts with the reaction* $a \rightarrowtriangle_{\mathsf{R}_1,\phi_1} \Delta_1(a,\phi_1)$ *iff* $\phi_1 : \tilde{P}_1 \not\hookrightarrow \Delta_0(a,\phi_0)$*, where* $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ *(i = 0,1). The following diagram illustrates the situation:*

$$\begin{tikzcd}
 & \tilde{P}_0 \arrow[d, hook, "\phi_0"'] & \tilde{P}_1 \arrow[dl, hook', "\phi_1"'] \arrow[d, hook, "/" marking, "\phi_1"] \\
\Delta_1(a,\phi_1) & a \arrow[l, -{Triangle[open]}] \arrow[r, -{Triangle[open]}] & \Delta_0(a,\phi_0)
\end{tikzcd} .$$

In both cases, reaction using R_0 must modify something in the range of ϕ_1 since it becomes, or stops being, an embedding. Furthermore, the reaction generated by ϕ_0 can only modify the parts of the agent that are in its range. Thus, there must be some overlap between the embeddings which is modified by reaction. It turns out that we can express this modification of overlaps concisely using category theory, so let us now discuss the category of bigraph embeddings.

7.2 Category of Bigraph Embeddings

The embeddings of Section 5 form categories where the objects are graphs and the arrows are embeddings:

Definition 7.5 (category of bigraph embeddings)**.** The bigraphical embedding categories LGEMB($\mathcal{K}$), PGEMB($\mathcal{K}$), BGEMB($\mathcal{K}$) over a basic signature $\mathcal{K}$ respectively have link graphs, place graphs, and bigraphs over $\mathcal{K}$ as their objects and the arrows are embeddings.

Composition is function composition and identities $\mathsf{id}_G : G \hookrightarrow G$ are identity functions on the support and interfaces of G. □

These categories should not be confused with the usual bigraphical categories (cf. Section 2.2.3) where the arrows are bigraphs. For the remainder of this section we shall use lower case letters $f, g, \ldots$ for arrows in addition to ϕ.

We are interested in these categories as the categorical notions of *pullback* and *pushout*, if the embedding categories have them, will allow us to characterize overlaps and minimal contexts exhibiting those overlaps, respectively. Let us discuss these two notions and their interpretations in some depth:

7.2.1 Pullbacks of Embeddings

An overlap between two embeddings $\vec{\phi} : \vec{G} \hookrightarrow H$ can be thought of as a maximal shared subgraph I and a pair of embeddings $\vec{p} : I \hookrightarrow \vec{G}$ such that the following diagram is a pullback diagram in the category of bigraph embeddings:

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"'] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & G_0 \arrow[d, hook, "\phi_0"] \\
G_1 \arrow[r, hook, "\phi_1"'] & H
\end{tikzcd} .$$

The subgraph should be maximal to ensure that $\vec{p} : I \hookrightarrow \vec{G}$ describes all of the overlap and the pullback property captures this maximality requirement. More precisely, if there is another subgraph I' and embeddings $\vec{p'} : I' \hookrightarrow \vec{G}$ that make the above diagram commute, then I' is a subgraph of I, i.e., there is a (unique) embedding $u : I' \hookrightarrow I$ such that the following diagram commutes:

$$\begin{tikzcd}
I' \arrow[dr, dotted, hook, "u"] \arrow[drr, hook, "p'_0"] \arrow[ddr, hook, "p'_1"'] & & \\
& I \arrow[r, hook, "p_0"'] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & G_0 \arrow[d, hook, "\phi_0"] \\
& G_1 \arrow[r, hook, "\phi_1"'] & H
\end{tikzcd} \quad .$$

As an interesting special case, note that if the two embeddings do not overlap, the pullback is the empty bigraph and two empty maps.

We believe that the embedding formulation in Section 5 will have to be relaxed slightly in order for the embedding categories to have pullbacks. Let us first illustrate the issue with the embeddings of Section 5 through an example:

Example 4. Assume that we have the following overlapping link graph embeddings:

$$\begin{aligned}
G_0 &= [x \mapsto v, y \mapsto w] : \{x, y\} \to \{v, w\} \\
G_1 &= [\{x, y, z\} \mapsto u] : \{x, y, z\} \to \{u\} \\
H &= [\{x, y\} \mapsto u] : \{x, y\} \to \{u\} \\
\phi_0 &= \mathsf{Id}_{\{x,y\}}[\{v, w\} \mapsto u] : G_0 \hookrightarrow H \\
\phi_1 &= \mathsf{Id}_{\{x,y,u\}}[z \mapsto \emptyset] : G_1 \hookrightarrow H.
\end{aligned}$$

What should the pullback be? Since G_0 embeds into G_1 via ϕ_0 it seems reasonable that G_0 could be the maximal overlap, i.e.,:

$$I' = G_0 \qquad\qquad g_0 = \mathsf{Id}_{G_0} : I' \hookrightarrow G_0 \qquad\qquad g_1 = \phi_0 : I' \hookrightarrow G_1.$$

Alas, this is not a pullback! Consider the link graph

$$I'' = G_0 \otimes [z \mapsto a] : \{x, y, z\} \to \{v, w, a\}.$$

It has the following embeddings into G_0,G_1

$$\begin{aligned}
f_0 &= \mathsf{Id}_{G_0} \uplus [a \mapsto v, z \mapsto \emptyset] : I'' \hookrightarrow G_0 \\
f'_0 &= \mathsf{Id}_{G_0} \uplus [a \mapsto w, z \mapsto \emptyset] : I'' \hookrightarrow G_0 \\
f_1 = f'_1 &= \phi_0 \uplus [a \mapsto u, z \mapsto z] : I'' \hookrightarrow G_1
\end{aligned}$$

which satisfy $\phi_0 \circ f_0 = \phi_0 \circ f'_0 = \phi_1 \circ f_1$. If $\vec{g} : I' \hookrightarrow \vec{G}$ was a pullback there should be unique embeddings $u : I'' \hookrightarrow I'$, $u' : I'' \hookrightarrow I'$ such that $f_0 = g_0 \circ u$, $f'_0 = g_0 \circ u'$, and $f_1 = g_1 \circ u = g_1 \circ u'$. But this is impossible since $\mathsf{rng}(f_1) \ni z \notin \mathsf{rng}(g_1)$.

Furthermore, neither $\vec{f} : I'' \hookrightarrow \vec{G}$ nor $\vec{f'} : I'' \hookrightarrow \vec{G}$ are pullbacks. For example, if $\vec{f} : I'' \hookrightarrow \vec{G}$ was a pullback, there should be a unique $u : I'' \hookrightarrow I''$ such that $f'_0 = f_0 \circ u$ and $f'_1 = f_1 \circ u = f'_1 \circ u$. Since $f'_1(z) = z$ and $f'^{-1}_1(z) = z$ we must have $u(z) = z$, and since $f'_0(a) = f'_0(w) = w$ and $f_0^{-1}(w) = w$ we must have $u(a) = u(w) = w$. But this violates structure preservation which dictates $(u \circ \mathit{prnt}_{I''})(z) = \mathit{prnt}_{I''}(z) \Rightarrow u(a) = a$.

Intuitively, the problems arise because ϕ_1 maps z to the empty set. As we saw, though $\vec{g} : I' \hookrightarrow \vec{G}$ is an overlap, other subgraphs may include z which is not in I' and thus I' is not maximal. But if we add z to the overlap, we have to choose a single link of G_0 which it belongs to, cf. f_0 and f'_0. Any such choice is equally good but not isomorphic to the others and thus not a pullback.

It is easy to transfer this example to place graph embeddings (and this is a good exercise for the reader!) and the issues thus apply in that setting as well. □

We believe that this problem can be solved by allowing outer names/roots to embed into multiple outer names/roots, i.e., by changing conditions (LGE-4) and (PGE-3) to

(LGE-4') $\phi^{\mathsf{o}} : Y_G \to E_H \uplus \mathcal{P}(Y_H) \setminus \emptyset$ is an arbitrary map

(PGE-3') $\phi^{\mathsf{r}} : m_G \to V_H \uplus \mathcal{P}(m_H) \setminus \emptyset$ is an arbitrary map

The structure preservation conditions will ensure that outer names/roots, which contain at least one point/child that is not mapped to the empty set, will only be mapped to a single link/place. In other words, only outer names/roots, where all their points/children map to the empty set, are allowed to map to multiple outer names/roots. Intuitively, this models the situation where it is irrelevant which of the outer names/roots the outer name/root maps to.

Note that the results about decompositions and bigraph embeddings in Section 5 probably do not hold for the more general embeddings allowed by the relaxed conditions. However, this is unimportant, as we only need those result for the subset of embeddings that satisfy the original conditions, and they are still embeddings under the new conditions.

With the relaxed conditions on embeddings, we can construct the pullback in the above example:

Example 5. The pullback of the embeddings $\vec{\phi} : \vec{G} \hookrightarrow H$ from Example 4 is

$$\begin{aligned} I = I'' &= G_0 \otimes [z \mapsto a] : \{x, y, z\} \to \{v, w, a\} \\ p_0 &= \mathsf{Id}_{G_0} \uplus [a \mapsto \{v, w\}, z \mapsto \emptyset] : I \hookrightarrow G_0 \\ p_1 &= \phi_0 \uplus [a \mapsto u, z \mapsto z] : I \hookrightarrow G_1 . \end{aligned}$$

The unique embeddings $u_g : I' \hookrightarrow I$,$u_f : I'' \hookrightarrow I$, and $u_{f'} : I'' \hookrightarrow I$ for the spans $\vec{g} : I' \hookrightarrow \vec{G}$, $\vec{f} : I'' \hookrightarrow \vec{G}$, and $\vec{f'} : I'' \hookrightarrow \vec{G}$, respectively, are

$$u_g = \mathsf{Id}_{\{x,y,v,w\}} \qquad u_f = \mathsf{Id}_I \qquad u_{f'} = \mathsf{Id}_I .$$

□

We have a construction which we believe gives pullbacks for link graph embeddings, but have yet to prove it correct. So far, however, we have no indications that it should not be correct, and we believe that the construction transfers to place graphs and bigraphs. We therefore venture a conjecture:

Conjecture 7.6 (pullbacks in the category of bigraph embeddings)**.** *The bigraphical embedding categories* LgEmb($\mathcal{K}$), PgEmb($\mathcal{K}$), BgEmb($\mathcal{K}$)*, where conditions (LGE-4) and (PGE-3) are replaced by conditions (LGE-4') and (PGE-3'), have pullbacks.*

Furthermore, since bigraphs are finite, we believe that there are only a finite number of ways that two bigraphs can overlap in any context, when we disregard the choice of support in those contexts. In other words, we conjecture that there is a finite set of pullbacks, up to isomorphism, for any two bigraphs:

Conjecture 7.7. *For any two objects $\vec{G}$ in one of the bigraphical embedding categories* LgEmb($\mathcal{K}$), PgEmb($\mathcal{K}$), BgEmb($\mathcal{K}$)*, where conditions (LGE-4) and (PGE-3) are replaced by conditions (LGE-4') and (PGE-3'), there are finitely many spans $\vec{p} : I \hookrightarrow \vec{G}$ (up to iso on I) which are pullbacks of a cospan $\phi : \vec{G} \hookrightarrow H$.*

In other words, we expect to be able to construct a finite representation of all possible overlaps between two bigraphs. We shall defer discussion of this construction until we have discussed pushouts on which the construction relies.

7.2.2 Pushouts of Embeddings

Where a pullback is a maximal subgraph that characterizes the overlap of two embeddings, the dual notion of a *pushout*, if it exists, is a minimal context where two embeddings exhibit a given overlap. Pushouts do not in general exist in the bigraphical embedding categories as the following example illustrates:

Example 6. Consider the following span of place graph embeddings (we leave out controls for brevity):

$$\begin{aligned}
I &= (\{v\}, [v \mapsto 0]) : 1 \\
G_0 &= (\{v, u\}, [v \mapsto u, u \mapsto 0]) : 1 \\
G_1 &= (\{v, w\}, [v \mapsto 0, w \mapsto 0]) : 1 \\
g_0 &= [v \mapsto v, 0 \mapsto u] : I \hookrightarrow G_0 \\
g_1 &= [v \mapsto v, 0 \mapsto 0] : I \hookrightarrow G_1.
\end{aligned}$$

This span has no pushout because G_1 insists that v has a sibling whereas G_0 insists that v has no siblings and clearly no context can satisfy both of these requirements. □

Intuitively, this just means that we cannot single out a subgraph in two bigraphs and then construct a context where they overlap at that subgraph – which is unsurprising, since embeddings are structure preserving.

However, for pullbacks we know, by definition, that there are contexts where the overlap can be found. The remaining question is then: can we construct a minimal such context? We believe the answer is yes, but, as was the case for pullbacks, we shall have to relax the embedding conditions, as the following example illustrates:

Example 7. Consider the following pullback of place graph embeddings (we again leave out controls for brevity and use named sites and roots q, r, s for clarity):

$$\begin{aligned}
I &= (\emptyset, \emptyset) : \emptyset \\
G_0 &= (\{v\}, [s \mapsto v, v \mapsto r]) : \{s\} \to \{r\} \\
G_1 &= (\{w\}, [w \mapsto q]) : \{q\} \\
H &= (\{v, u, x, w\}, [w \mapsto x, x \mapsto u, u \mapsto v, v \mapsto r]) : \{r\} \\
p_0 &= \emptyset : I \hookrightarrow G_0 \\
p_1 &= \emptyset : I \hookrightarrow G_1 \\
\phi_0 &= [v \mapsto v, s \mapsto u, r \mapsto r] : G_0 \hookrightarrow H \\
\phi_1 &= [w \mapsto w, q \mapsto x] : G_1 \hookrightarrow H.
\end{aligned}$$

What should the pushout of $\vec{p} : I \hookrightarrow \vec{G}$ be? The two embeddings do not overlap, so what is the canonical context where two place graphs do not overlap? Perhaps the pushout should be the tensor product of the two place graphs:

$$K = G_0 \otimes G_1 \qquad o_0 = \mathsf{Id}_{G_0} : G_0 \hookrightarrow K \qquad o_1 = \mathsf{Id}_{G_1} : G_1 \hookrightarrow K.$$

Alas, there is no embedding $u : K \hookrightarrow H$ such that $u \circ o_0 = \phi_0$ and $u \circ o_1 = \phi_1$ since condition (PGE-6) prevents us from mapping the root q to a descendant of the site s (in fact, this was the motivation for adding that condition, as discussed in Example 3).

If we disregard condition (PGE-6) then $\vec{o} : \vec{G_0} \hookrightarrow \vec{G_1}$ is a pushout. In particular, the embedding $u = \phi_0 \uplus \phi_1$ is the only one that satisfies $u \circ o_0 = \phi_0$ and $u \circ o_1 = \phi_1$. □

Thus, it seems that in order to have pushouts, we must discard condition (PGE-6). In fact, we believe that this all that is required in order to have pushouts of pullbacks in the bigraphical embedding categories. As for pullbacks, we have a construction of pushouts for pullbacks of link graph embeddings, which we think transfers to place graphs and bigraphs. But we have yet to prove it correct, though so far nothing indicates that it is incorrect. So, again, we venture a conjecture:

Conjecture 7.8. *For any pullback $\vec{p} : I \hookrightarrow \vec{G}$ of some cospan in one of the bigraphical embedding categories* LgEmb($\mathcal{K}$), PgEmb($\mathcal{K}$), BgEmb($\mathcal{K}$), *where condition (PGE-6) is discarded and conditions (LGE-4) and (PGE-3) are replaced by conditions (LGE-4') and (PGE-3'), there is a pushout $\vec{o} : \vec{G} \hookrightarrow K$.*

7.2.3 Characterizing Overlaps

As we discussed in Section 7.2.1, we think that there are only finitely many pullbacks for any two bigraphs (up to iso). While we hope to eventually find a direct method for enumerating these pullbacks, we will initially be content with any method. In particular, we believe that the following brute-force method will work (assuming we wish to characterize the overlaps of $\vec{G}$):

1. Generate overlap candidates, i.e., spans $\vec{g} : I' \hookrightarrow \vec{G}$, by equating subsets of entities of G_0 and G_1. It is still unclear to us exactly how this should be done, but for let us assume that we can generate such candidates and that there are finitely many.

2. For each overlap candidate $\vec{g} : I' \hookrightarrow \vec{G}$, apply the pushout construction. If it results in a bound $\vec{o} : \vec{G} \hookrightarrow K$ for $\vec{g}$, i.e., K is a bigraph and $o_0 \circ g_0 = o_1 \circ g_1$, then we can construct a pullback.

Let us now assume that the conjectures of the previous two sections hold and that we have a method for obtaining the finite set of pullback spans for any pair $\vec{G}$ of bigraphs. In other words, for each pair $\vec{G}$ of bigraphs assume that we have a finite set of pullback-pushout (PP) diagrams

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & G_0 \arrow[d, hook, "o_0"] \\
G_1 \arrow[r, hook, "o_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd} .$$

such that for any pair of embeddings $\vec{\phi} : \vec{G} \hookrightarrow H'$ its pullback is in one of the PP diagrams, i.e.,

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & G_0 \arrow[d, hook, "o_0"'] \arrow[ddr, hook, bend left, "\phi_0"] & \\
G_1 \arrow[r, hook, "o_1"] \arrow[drr, hook, bend right, "\phi_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, hook, dotted, "u"] & \\
& & H'
\end{tikzcd} .$$

Thus, if our conjectures hold, we can give a finite characterization of all overlaps between two bigraphs in any context.

7.3 PP Diagrams, Activation and Inhibition

Let us now return to the question of characterizing causality at the level of rules, i.e., the activation and inhibition relations. The characterization of overlaps based on PP diagrams, as outlined in the previous section, in itself provides the means to approximate the activation and inhibition relations $\prec$ and $\#$. For example, if two redexes can never overlap in a context, i.e., the only PP diagram has $I = \emptyset$, then they can never be in conflict. However, as discussed above in Section 7.1, we can do better than that by exploiting that edit scripts provide a notion of modification, and in this section we outline how.

We shall need a development for edit scripts which we, due to time constraints, only state as an assumption: we assume that we can construct an inverse Δ^{-1} of any linear edit script Δ which satisfies

$$\begin{aligned} & \phi : \Delta(\tilde{P}) \hookrightarrow H \\ \Rightarrow \quad & \exists H', \phi' : \tilde{P} \hookrightarrow H' \; . \; (H, \phi) = \Delta(H', \phi') \wedge (H', \phi') = \Delta^{-1}(H, \phi). \end{aligned}$$

7.3.1 Inhibition

Recall from Section 3.2 that rule R_0 *inhibits* rule R_1, written $\mathsf{R}_0 \# \mathsf{R}_1$ iff R_0 generates at least one reaction which conflicts with a reaction generated by R_1.

We believe that we can construct the inhibition relation through the PP diagrams of the previous section. More precisely, we make the following conjecture:

Conjecture 7.9. *Assume an agent a, two linear reconfiguration rules $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ $(i = 0{,}1)$, and embeddings $\vec{\phi} : \vec{\tilde{P}} \hookrightarrow a$. By definition, the rules and embeddings generate the reactions $a \twoheadrightarrow_{\mathsf{R}_0, \phi_0} \Delta_0(a, \phi_0)$ and $a \twoheadrightarrow_{\mathsf{R}_1, \phi_1} \Delta_1(a, \phi_1)$, as illustrated by the following diagram:*

$$\begin{tikzcd} \tilde{P}_0 \arrow[dr, hook, "\phi_0"] & & \tilde{P}_1 \arrow[dl, hook', "\phi_1"'] \\ \Delta_0(a, \phi_0) & a \arrow[l, -open triangle] \arrow[r, -open triangle] & \Delta_1(a, \phi_1) \end{tikzcd} .$$

Also, assume that the cospan $\vec{\phi} : \vec{\tilde{P}} \hookrightarrow a$ has the PP diagram

$$\begin{tikzcd} I \arrow[r, hook, "p_0"'] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & \tilde{P}_0 \arrow[d, hook, "o_0"'] \arrow[ddr, hook, "\phi_0"] & \\ \tilde{P}_1 \arrow[r, hook, "o_1"] \arrow[drr, hook, "\phi_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, hook, dotted, "u"] & \\ & & a \end{tikzcd} .$$

We then conjecture

$$\begin{aligned} \phi_0 : \tilde{P}_0 \not\hookrightarrow \Delta_1(a, \phi_1) \quad & \Leftrightarrow \quad o_0 : \tilde{P}_0 \not\hookrightarrow \Delta_1(H, o_1) \\ \phi_1 : \tilde{P}_1 \not\hookrightarrow \Delta_0(a, \phi_0) \quad & \Leftrightarrow \quad o_1 : \tilde{P}_1 \not\hookrightarrow \Delta_0(H, o_0). \end{aligned}$$

Assuming this conjecture holds, we get that PP diagrams characterize inhibition:

Theorem 7.10. *Given two linear reconfiguration rules $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ $(i = 0{,}1)$ then $\mathsf{R}_0 \# \mathsf{R}_1$ iff there is a PP diagram*

$$\begin{tikzcd} I \arrow[r, hook, "p_0"'] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & \tilde{P}_0 \arrow[d, hook, "o_0"'] \\ \tilde{P}_1 \arrow[r, hook, "o_1"] & H \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

such that $o_1 : \tilde{P}_1 \not\leftrightarrow \Delta_0(H, o_0)$.

Proof. Follows immediately from Prop. 7.4 and Conjecture 7.9. □

7.3.2 Activation

Recall from Section 3.2 that rule R_0 *activates* rule R_1, written $\mathsf{R}_0 \prec \mathsf{R}_1$ iff R_0 generates at least one reaction which enables a reaction generated by R_1.

We believe that we can construct the activation relation through the PP diagrams of the previous section. More precisely, we make the following conjecture:

Conjecture 7.11. *Assume an agent a, two linear reconfiguration rules* $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ *(i = 0,1), and embeddings* $\phi_0 : \tilde{P}_0 \hookrightarrow a$, $\phi'_0 : \Delta_0(\tilde{P}) \hookrightarrow a'$, $\phi_1 : \tilde{P}_1 \hookrightarrow a'$ *where* $(a', \phi'_0) = \Delta_0(a, \phi_0)$, *as illustrated by the following diagram:*

$$\begin{tikzcd}
\tilde{P}_0 \arrow[d, hook, "\phi_0"'] & \Delta_0(\tilde{P}_0) \arrow[d, hook, "\phi'_0"] & \tilde{P}_1 \arrow[dl, hook, "\phi_1"] \\
a \arrow[r, -{Triangle[open]}] & \Delta_0(a, \phi_0) & .
\end{tikzcd}$$

Also, assume that the cospan $\phi'_0 : \Delta_0(\tilde{P}_0) \hookrightarrow a'$, $\phi_1 : \tilde{P}_1 \hookrightarrow a'$ *has the PP diagram*

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & \Delta_0(\tilde{P}_0) \arrow[d, hook, "o_0"'] \arrow[ddr, hook, "\phi'_0"] & \\
\tilde{P}_1 \arrow[r, hook, "o_1"] \arrow[drr, hook, "\phi_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, hook, dotted, "u"] & \\
& & \Delta_0(a, \phi_0) \quad .
\end{tikzcd}$$

We then conjecture

$$\phi_1 : \tilde{P}_1 \not\leftrightarrow a \qquad \Leftrightarrow \qquad o_1 : \tilde{P}_1 \not\leftrightarrow \Delta_0^{-1}(H, o_0).$$

Assuming this conjecture holds, we get that PP diagrams characterize activation:

Theorem 7.12. *Given two linear reconfiguration rules* $\mathsf{R}_i = (\tilde{P}_i, \Delta_i)$ *(i = 0,1) then* $\mathsf{R}_0 \prec \mathsf{R}_1$ *iff there is a PP diagram*

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"] \arrow[dr, phantom, "\lrcorner", very near start] & \Delta_0(\tilde{P}_0) \arrow[d, hook, "o_0"'] \\
\tilde{P}_1 \arrow[r, hook, "o_1"] & H \arrow[ul, phantom, "\ulcorner", very near start] \quad .
\end{tikzcd}$$

such that $o_1 : \tilde{P}_1 \not\leftrightarrow \Delta_0^{-1}(H, o_0)$.

Proof. Follows immediately from Prop. 7.3 and Conjecture 7.11. □

8 Anchored Matching

A pillar in the scalability of the KaSim algorithm is that, after reaction, we only search for new matches in the parts of the agent that have been modified. In other words, KaSim requires a localized matching algorithm that only searches a subset of the agent. Such an algorithm has not yet been presented in the bigraph literature. Previously published matching algorithms find matches anywhere in the agent [20, 34]; these algorithms are useful for the initialization phase of KaSim, where all matches must be found, but it is unclear how to specialize them to local matching.

In this section we shall present a localized matching algorithm, based on the idea of expanding partial embeddings to total embeddings, i.e., matches. Not only is this a localized matching algorithm, but it combines well with our edit scripts and causality analysis:

1. Mediation of an edit script results in a reaction $a \twoheadrightarrow a'$ and an embedding $\phi_0 : R_0' \hookrightarrow a'$ of the reactum of the applied reaction rule $\mathsf{R}_0 = (R_0, R_0', \eta_0)$ into the resulting agent a'.

2. For any rule R_1 that may be activated by $\mathsf{R}_1 = (R_1, R_1', \eta_1)$, i.e., $\mathsf{R}_0 \prec \mathsf{R}_1$, the causality analysis gives us a set of PP diagrams of the form

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & R_0' \arrow[d, hook, "o_0"] \\
R_1 \arrow[r, hook, "o_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

   where $I \neq \epsilon$, which characterizes all matches of R_0' and R_1 that overlap exactly as prescribed by $\vec{p} : I \hookrightarrow \vec{R}$.

3. Iff $\phi_0 \circ o_0^{-1} : H \rightharpoonup a'$ is a partial embedding

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & R_0' \arrow[d, hook, "o_0"] \arrow[ddr, hook, "\phi_0"] & \\
R_1 \arrow[r, hook, "o_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, hook, harpoon, "\phi_0 \circ o_0^{-1}"] & \\
 & & a'
\end{tikzcd}$$

   then any and all extensions to a total embedding of H, i.e., $\phi_H : H \hookrightarrow a'$ where $\phi_H \restriction_{\mathsf{rng}(o_0)} = \phi_0 \circ o_0^{-1}$, is a match of R_1, as witnessed by the composition $\phi_H \circ o_1 : R_1 \hookrightarrow H$

$$\begin{tikzcd}
I \arrow[r, hook, "p_0"] \arrow[d, hook, "p_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & R_0' \arrow[d, hook, "o_0"] \arrow[ddr, hook, "\phi_0"] & \\
R_1 \arrow[r, hook, "o_1"'] \arrow[drr, hook, "\phi_H \circ o_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, hook, "\phi_H"] & \\
 & & a'.
\end{tikzcd}$$

We call the initial partial embedding of H the *anchor* and, derived from this, we call the algorithm *anchored matching.* Note that we assume that the redex is one connected component. However, as discussed in the original presentation of the KaSim algorithm [12], this is not a problem, since the separate connected components of a redex can be matched separately and then combined ad hoc during simulation; see loc. cit. for details.

8.1 Algorithm

Due to time constraints, we shall only present a very naive algorithm, and only discuss possible optimizations. While this algorithm is too naive to be practical in an implementation, it is sufficient for communicating our approach and it can probably serve as a nice and simple starting point for soundness and completeness proofs.

The algorithm builds on the following two ideas:

fringe: For a bigraph $G : \langle k, X\rangle \to \langle m, Y\rangle$ and a subset of its entities $S \subset V_G \uplus E_G \uplus k \uplus X \uplus m \uplus Y$, the *fringe* of S in G are the entities of G that are adjacent to entities of S but not in S, i.e.,

$$\begin{aligned} \mathit{fringe}(G,S) = {} & \{c \in (k \uplus V_G) \setminus S && \mid \exists p \in S : \mathit{prnt}(c) = p\} \\ & \uplus \{p \in (V_G \uplus m) \setminus S && \mid \exists c \in S : \mathit{prnt}(c) = p\} \\ & \uplus \{l \in E_G \uplus Y \setminus S && \mid \exists v \in S, i \in \mathbb{N} : \mathit{link}(p) = l \\ & && \quad \lor \exists x \in S : \mathit{link}(x) = l\} \\ & \uplus \{v \in V_G \setminus S && \mid \exists l \in S, i \in \mathbb{N} : \mathit{link}(v,i) = l\} \\ & \uplus \{x \in X \setminus S && \mid \exists l \in S : \mathit{link}(x) = l\}. \end{aligned}$$

If G consists of a single connected component and S is non-empty, then if we keep expanding S by entities in its fringe, eventually S will cover G.

valid extension: For a partial embedding $\phi : G \hookrightarrow a$ into an agent $a : \langle m, Y\rangle$, an entity s in the fringe of ϕ in G, and subset T of the entities of a, i.e.,

$$\begin{gathered} s \in \mathit{fringe}(G, \mathsf{dom}(\phi)) \\ T \subseteq V_a \uplus E_a \uplus m \uplus Y \end{gathered}$$

the define the predicate $\mathit{validExt}(\phi, s \mapsto T)$ to be true when $\phi[s \mapsto T] : G \hookrightarrow a$ is also a partial embedding. Note that T is a subset of the entities of a, since embeddings of sites and inner names map to subsets; for the other entities T should be a singleton.

Note that the injectivity and structure preservation conditions on embeddings imply that T must be on the fringe of ϕ in a.

The algorithm is listed in Algorithm 1. In brief, it works as follows:

line 2: If the fringe of ϕ in G is empty, ϕ is total (since G is a connected component) and thus we have found a match.

line 6: Otherwise, choose an entity s on the fringe of ϕ in G which shall be matched next.

line 7: For any possible mapping of s to entities T on the fringe of ϕ in a:

line 8: If $s \mapsto T$ is a valid extension of ϕ

line 9: find all total extensions of $\phi[s \mapsto T]$.

The obvious places to optimize this algorithm are the choices of s and T:

choosing T: It should be possible to only choose T's such that $\mathit{validExt}(\phi, s \mapsto T)$. In particular, the structure preservation conditions on embeddings should guide the choice of T. For instance, if s is a node, T should be a singleton $\{t\}$ where t is a node with $\mathit{ctrl}_G(s) = \mathit{ctrl}_a(t)$, and if s is on the fringe of ϕ because $\mathit{prnt}_G(s) \in \mathsf{dom}(\phi)$, then $t \in \mathit{prnt}_a^{-1}(\phi(\mathit{prnt}_G(s)))$.

Algorithm 1 The Anchored Matching algorithm

```
Require: $\phi : G \hookrightarrow a$ non-trivial, $G$ has one connected component
 1: procedure ANCHORED-MATCHING($\phi : G \hookrightarrow a$)
 2:     if $\mathit{fringe}(G, \mathsf{dom}(\phi)) = \emptyset$ then
 3:         return $\{\phi\}$
 4:     else
 5:         $M \leftarrow \emptyset$
 6:         choose $s \in \mathit{fringe}(G, \mathsf{dom}(\phi))$
 7:         for all $T \subseteq \mathit{fringe}(a, \mathsf{rng}(\phi))$ do
 8:             if $\mathit{validExt}(\phi, s \mapsto T)$ then
 9:                 $M \leftarrow M \cup$ ANCHORED-MATCHING($\phi[s \mapsto T]$)
10:         return $M$
```

choosing s**:** The heuristic for choosing s is critical for narrowing down the number of T's we will have to explore for each s. We believe that a good strategy could be to choose an s that is estimated to have a small number of possible embeddings.

For instance, note that if s is on the fringe of ϕ because of one of its children c, i.e., $c \in \mathit{prnt}_G^{-1}(s) \cap \mathsf{dom}(\phi)$, then structure preservation dictates that the only choice is $T = \{\mathit{prnt}_a(\phi(c))\}$. Similarly, embeddings of nodes and inner names determine the embeddings of the connected links.

We therefore envision a representation of the fringe of ϕ in G as a prioritized queue, where the priority of each entity is an estimation number of possible embeddings based on its adjacency relation to $\mathsf{dom}(\phi)$. Whenever the embedding is extended with an entity, the estimate of adjacent entities in the fringe should be updated.

Furthermore, note that an inner will only be on the fringe of ϕ in G if the link is connected to is already mapped by ϕ. Together with the embedding conditions, this means that extending an embedding by an inner name will only affect the choice of sibling inner names. In fact, there are very few restrictions on how such sibling inner names should be mapped, and it therefore seems reasonable to postpone matching of inner names to the end. Similarly, sites could also be postponed.

9 Conclusions and Future Work

In this report we have laid a firm, formal foundation for an implementation of stochastic bigraphs:

1. We have defined *stochastic parametric reactive systems*, an alternative foundation for the dynamic semantics of bigraphs which is amenable to implementation: support is handled explicitly, parametric reaction rules are first-class citizens, and the stochastic rates of an agent is determined by its matches. Furthermore, we have shown that stochastic parametric reactive systems have the same abstract reactions as Milner's reactive systems.

2. We have defined *bigraph embeddings* and shown that they are isomorphic to certain decompositions of bigraphs; in particular, embeddings of redexes into agents are isomorphic to matches. Furthermore, we have shown that embeddings of a solid bigraph are determined by support translations of its nodes.

3. We have proposed, and proven sound and complete, a set of minimal *edits* of parametric redexes which, when put in sequence to form *edit scripts*, are equivalent to parametric reaction rules and generate the same abstract reactions.

4. We have outlined a characterization of causality and conflict for linear parametric reaction rules, based on pullbacks in the category of bigraph embeddings.

5. We have given a localized matching algorithm: starting from a partial match, i.e., a partial embedding, of a connected component, it finds all completions.

The presented work is part of an effort to build an efficient and scalable simulator for stochastic bigraphs: the Stochastic Bigraphical Abstract Machine. So far a prototype based on SPRSs, embeddings, edit scripts, and anchored matching have been implemented. It allows stochastic simulation of certain BRSs: all controls must be active, reaction rules must be linear, and redexes must be solid and consist of a single connected component.

9.1 Future Work

Localized matching and causality analysis of rules should be investigated in more detail. In particular, we must prove our conjectures about the embedding categories and soundness and completeness of anchored matching. Furthermore, it it unclear whether the pullback approach to characterizing causality and conflict will (a) result in a practical algorithm, and (b) generalize to non-linear reaction rules.

Our presentation of bigraph embeddings, and the related proofs, could probably be simplified by using a formulation of concrete bigraphs where roots and sites are named (as we did in our development of edit scripts). In fact, we believe that, for many purposes, the theory of bigraphs would be simpler to work with if roots and sites were named.

From an implementation perspective, there is a need for representing sets of embeddings efficiently: in biological models there will often be a large number of embeddings of each redex. This needs further investigation, but as a first step we believe the following conjecture may prove useful: embeddings of solid bigraphs are determined by the support translation of the leaves of the place graph.

For the biological simulation scenarios we have in mind, we expect the user to provide edit scripts as they provide a natural way to express protein-protein interaction as well as dynamic compartmentalization. However, there might be applications where the user would prefer to provide reaction rules and have the system infer suitable edit scripts. While we have given a simple construction of edit scripts for any parametric reaction rule, it is very naive and assumes that there is no relation between nodes and edges of redex and reactum, resulting in inefficient simulation. It should therefore be investigated how one can derive better edit scripts. This seems related to the tree edit distance problem, where one wishes to find a minimal edit script that transforms one tree into another [5].

References

[1] G. Bacci, D. Grohmann, and M. Miculan. Bigraphical models for protein and membrane interactions. In *Proceedings of the Third International Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2009)*, pages 3–18. EPTCS 11, 2009.

[2] G. Bacci, D. Grohmann, and M. Miculan. Dbtk: A toolkit for directed bigraphs. In *CALCO*, pages 413–422, 2009.

[3] M. Beauquier and C. Schürmann. A bigraph reactive systems realtion model. Technical Report TR-2010-126, IT University of Copenhagen, June 2010.

[4] BigMC. BigMC – Bigraphical Model Checker. `http://bigraph.org/bigmc/`.

[5] P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337: 217–239, June 2005.

[6] L. Birkedal, T. C. Damgaard, A. J. Glenstrup, and R. Milner. Matching of bigraphs. *Electronic Notes in Theoretical Computer Science*, 175(4):3–19, 2007.

[7] BPLTool. BPL Tool. `http://www.itu.dk/research/pls/wiki/index.php/BPL_Tool`.

[8] G. L. Cattani, J. J. Leifer, and R. Milner. Contexts and embeddings for closed shallow action graphs. Technical Report UCAM-CL-TR-496, University of Cambridge, Computer Laboratory, July 2000.

[9] T. C. Damgaard and J. Krivine. A generic language for biological systems based on bigraphs. Technical Report TR-2008-115, IT University of Copenhagen, December 2008.

[10] T. C. Damgaard, V. Danos, and J. Krivine. A language for the cell. Technical Report TR-2008-116, IT University of Copenhagen, December 2008.

[11] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325, 2004.

[12] V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable simulation of cellular signaling networks. In *Proceedings of the 5th Asian conference on Programming languages and systems*, APLAS'07, pages 139–157. Springer-Verlag, 2007.

[13] S. Debois. Computation in the informatic jungle. Technical Report TR-2011-147, IT University of Copenhagen, 2011. (forthcoming).

[14] N. Eén and N. Sörensson. MiniSAT. `http://minisat.se`.

[15] H. Ehrig. Bigraphs meet double pushouts. *Bulletin of the EATCS*, 78:72–85, 2002.

[16] H. Ehrig, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors. *Handbook of graph grammars and computing by graph transformation, Volume 3: Concurrency, Parallelism, and Distribution.* World Scientific Publishing Co., Inc., 1999. ISBN 9-810240-21-X.

[17] A. Faithfull. Big Red. `http://www.itu.dk/research/pls/wiki/index.php/Big_Red`, 2010.

[18] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[19] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[20] A. J. Glenstrup, T. C. Damgaard, L. Birkedal, and E. Højsgaard. An implementation of bigraph matching. Technical Report TR-2010-135, IT University of Copenhagen, December 2010.

[21] C. Greenhalgh. bigraphspace. `http://bigraphspace.svn.sourceforge.net/`, 2009.

[22] J. Hillston. *A compositional approach to performance modelling.* Cambridge University Press, 1996. ISBN 0-521-57189-8.

[23] E. Højsgaard and A. J. Glenstrup. The BPL Tool: A tool for experimenting with bigraphical reactive systems. Technical Report TR-2011-145, IT University of Copenhagen, October 2011.

[24] O. H. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge – Computer Laboratory, February 2004.

[25] J. Krivine, R. Milner, and A. Troina. Stochastic bigraphs. *Electronic Notes in Theoretical Computer Science*, 218:73 – 96, 2008. ISSN 1571-0661. Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIV).

[26] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, pages 585–591, 2011.

[27] R. Milner. Axioms for bigraphical structure. *Journal of Mathematical Structures in Computer Science*, 15(6):1005–1032, 2005.

[28] R. Milner. Embeddings and contexts for link graphs. In H.-J. Kreowski, U. Montanari, F. Orejas, G. Rozenberg, and G. Taentzer, editors, *Formal Methods in Software and Systems Modeling*, volume 3393 of *Lecture Notes in Computer Science*, pages 343–351. Springer Berlin / Heidelberg, 2005.

[29] R. Milner. *The Space and Motion of Communicating Agents.* Cambridge University Press, 2009.

[30] PEPAPlugIn. The PEPA Plug-in Project. `http://www.dcs.ed.ac.uk/pepa/tools/plugin/index.html`.

[31] C. Priami. Stochastic π-calculus. *Computer Journal*, 38(7):578–589, 1995.

[32] G. Rozenberg, editor. *Handbook of graph grammars and computing by graph transformation, Volume 1: Foundations.* World Scientific Publishing Co., Inc., 1997. ISBN 98-102288-48.

[33] A. Schack-Nielsen and C. Schürmann. Celf - a logical framework for deductive and concurrent systems (system description). In *IJCAR*, pages 320–326, 2008.

[34] M. Sevegnani, C. Unsworth, and M. Calder. A SAT based algorithm for the matching problem in bigraphs with sharing. Technical Report TR-2010-311, University of Glasgow, Department of Computing Science, 2010.

[35] P. Sobocinsky. Relative pushouts in graphical reactive systems. February 2002.

A Proofs

A.1 Bigraph Embeddings

A.1.1 Proof of Prop. 5.3

ϕ^{e}: Construct the map of each edge $e \in E_G$ as follows: choose a port $p = (v, i) \in \mathit{link}_G^{-1}(e)$, which is always possible since no edge is idle and every inner name is guarding, and let

$$\phi^{\mathsf{e}}(e) = \mathit{link}_H(\phi^{\mathsf{v}}(v), i).$$

By construction it satisfies condition (LGE-9); it must satisfy the other conditions since ϕ is an embedding and ϕ^{e} is the only embedding of edges that will satisfy condition (LGE-9): To see that ϕ^{e} is unique, assume that there is a different ϕ'^{e}, i.e., $\phi^{\mathsf{e}}(e) \neq \phi'^{\mathsf{e}}(e)$ for some $e \in E_G$. Since they both satisfy condition (LGE-9), the following must hold for the port p we chose when defining $\phi^{\mathsf{e}}(e)$:

$$\begin{aligned}\phi^{\mathsf{e}}(e) &= (\phi^{\mathsf{e}} \circ \mathit{link}_G)(p)\\ &= (\mathit{link}_H \circ \phi^{\mathsf{p}})(p)\\ &= (\phi'^{\mathsf{e}} \circ \mathit{link}_G)(p) = \phi'^{\mathsf{e}}(e)\end{aligned}$$

which contradicts our assumption that ϕ^{e} and ϕ'^{e} are different.

ϕ^{i}: Construct the map of each inner name $x \in X_G$ as follows:

$$\begin{aligned}\phi^{\mathsf{i}}(x) &= \mathit{points}_{H,x} \setminus \phi^{\mathsf{p}}(P_{G,x})\\ \mathit{points}_{H,x} &= (\mathit{link}_H^{-1} \circ \phi^{\mathsf{e}})(\mathit{link}_G(x))\\ P_{G,x} &= (\mathit{link}_G^{-1} \circ \mathit{link}_G)(x) \setminus \{x\}\\ \phi^{\mathsf{p}}(v, i) &= (\phi^{\mathsf{v}}(v), i).\end{aligned}$$

This is well-defined since no outer name of G is linked to an inner name, thus $\mathit{link}_G(x) \in E_G$, and no inner names are siblings. By construction it satisfies condition (LGE-7); it must satisfy the other conditions since ϕ is an embedding and ϕ^{i} is the only embedding of inner names that will satisfy condition (LGE-7): To see that ϕ^{i} is unique, assume that there is a different ϕ'^{i} that satisfies the conditions of Def. 5.1, i.e., there must be some $x \in X_G$ and $p \in X_H \uplus P_H$ with $p \in \phi^{\mathsf{i}}(x), p \notin \phi'^{\mathsf{i}}(x)$ (or vice versa). Since they both satisfy condition (LGE-7) and G no outer name is linked to an inner name we have:

$$\begin{aligned}\mathit{link}_G(x) &\in E_G\\ (\phi^{\mathsf{p}} \circ \mathit{link}_G^{-1} \restriction_{E_G})(\mathit{link}_G(x)) &= (\mathit{link}_H^{-1} \circ \phi^{\mathsf{e}})(\mathit{link}_G(x))\\ &= (\phi'^{\mathsf{p}} \circ \mathit{link}_G^{-1} \restriction_{E_G})(\mathit{link}_G(x))\end{aligned}$$

where

$$\phi^{\mathsf{p}}(p') = \begin{cases}(\phi^{\mathsf{v}}(v), i) & \text{if } p' = (v, i) \in P_G\\ \phi^{\mathsf{i}}(p') & \text{if } p' \in X_G\end{cases}$$

$$\phi'^{\mathsf{p}}(p') = \begin{cases}(\phi^{\mathsf{v}}(v), i) & \text{if } p' = (v, i) \in P_G\\ \phi'^{\mathsf{i}}(p') & \text{if } p' \in X_G\end{cases}$$

And since $p \in \phi^{\mathsf{i}}(x)$, and $\phi^{\mathsf{p}}(p')$ and $\phi'^{\mathsf{p}}(p')$ agree on ports, we must have $p \in \phi'^{\mathsf{i}}(x')$ for some $x' \in \mathit{link}_G^{-1} \restriction_{E_G} (\mathit{link}_G(x))$. But no inner names are siblings, so $x' = x$ and thus $p \in \phi'^{\mathsf{i}}(x')$ which contradicts our assumption that $p \notin \phi'^{\mathsf{i}}(x)$.

ϕ^{o}: Construct the map of each outer name $y \in Y_G$ as follows: choose a port $p = (v, i) \in link_G^{-1}(y)$, which is always possible since no outer name is idle or connected to an inner name, and let

$$\phi^{\mathsf{o}}(y) = link_H(\phi^{\mathsf{v}}(v), i).$$

By construction it satisfies condition (LGE-9); it must satisfy the other conditions since ϕ is an embedding and ϕ^{o} is the only embedding of outer names that will satisfy condition (LGE-9): To see that ϕ^{o} is unique, assume that there is a different ϕ'^{o}, i.e., $\phi^{\mathsf{o}}(y) \neq \phi'^{\mathsf{o}}(y)$ for some $y \in Y_G$. Since they both satisfy condition (LGE-9), the following must hold for the port p we chose when defining $\phi^{\mathsf{o}}(y)$:

$$\begin{aligned} \phi^{\mathsf{o}}(y) &= (\phi^{\mathsf{o}} \circ link_G)(p) \\ &= (link_H \circ \phi^{\mathsf{p}})(p) \\ &= (\phi'^{\mathsf{o}} \circ link_G)(p) = \phi'^{\mathsf{o}}(y) \end{aligned}$$

which contradicts our assumption that ϕ^{o} and ϕ'^{o} are different. □

A.1.2 Proof of Prop. 5.6

From the definitions of support translation, composition, and tensor product we have:

$$\begin{aligned} V_H &= V_C \uplus V_G \uplus V_D & C &: m_G + k \to m_H \\ ctrl_H &= ctrl_C \uplus ctrl_G \uplus ctrl_D & D &: k_D \to k_G \\ k_D &\subseteq k_H \end{aligned}$$

To show that ϕ is an embedding we need to express the parent map of H in terms of it decomposition $C \circ (G \circ D \otimes \mathsf{id}_k) \circ \pi$. We construct the map incrementally according to the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8)

$G \circ D$: We write $prnt_1$ for the parent map of the resulting place graph:

$$prnt_1(w) = \begin{cases} prnt_D(w) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) \in V_D \\ prnt_G(j) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) = j \in k_G \\ prnt_G(w) & \text{if } w \in V_G \end{cases}$$

$G \circ D \otimes \mathsf{id}_k$: We write $prnt_2$ for the parent map of the resulting place graph:

$$\begin{aligned} prnt'_{\mathsf{id}_k}(k_D + i) &= m_G + i \qquad (i \in k) \\ prnt_2(w) &= (prnt_1 \uplus prnt'_{\mathsf{id}_k})(w) \\ &= \begin{cases} prnt_D(w) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) \in V_D \\ prnt_G(j) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) = j \in k_G \\ prnt_G(w) & \text{if } w \in V_G \\ m_G - k_D + w & \text{if } w \in (k_D + k) \setminus k_D \end{cases} \end{aligned}$$

$(G \circ D \otimes \mathsf{id}_k) \circ \pi$: We write $prnt_3$ for the parent map of the resulting place graph:

$$\begin{aligned} prnt_3(w) &= \begin{cases} prnt_2(\pi(w)) & \text{if } w \in k_H \\ prnt_2(w) & \text{if } w \in V_G \uplus V_D \end{cases} \\ &= \begin{cases} prnt_D(\pi(w)) & \text{if } w \in k_H \text{ and } \pi(w) \in k_D \text{ and } prnt_D(\pi(w)) \in V_D \\ prnt_G(j) & \text{if } w \in k_H \text{ and } \pi(w) \in k_D \text{ and } prnt_D(\pi(w)) = j \in k_G \\ m_G - k_D + \pi(w) & \text{if } w \in k_H \text{ and } \pi(w) \in (k_D + k) \setminus k_D \\ prnt_D(w) & \text{if } w \in V_D \text{ and } prnt_D(w) \in V_D \\ prnt_G(j) & \text{if } w \in V_D \text{ and } prnt_D(w) = j \in k_G \\ prnt_G(w) & \text{if } w \in V_G \end{cases} \end{aligned}$$

$H = C \circ (G \circ D \otimes \mathsf{id}_k) \circ \pi$:

$$prnt_H(w) = \begin{cases} prnt_3(w) & \text{if } w \in k_H \uplus V_G \uplus V_D \text{ and } prnt_3(w) \in V_G \uplus V_D \\ prnt_C(j) & \text{if } w \in k_H \uplus V_G \uplus V_D \text{ and } prnt_3(w) = j \in k_C \\ prnt_C(w) & \text{if } w \in V_C \end{cases}$$

$$= \begin{cases} prnt_D(\pi(w)) & \text{if } w \in k_H \text{ and } \pi(w) \in k_D \\ & \quad \text{and } prnt_D(\pi(w)) \in V_D \\ prnt_G(j) & \text{if } w \in k_H \text{ and } \pi(w) \in k_D \\ & \quad \text{and } prnt_D(\pi(w)) = j \in k_G \\ & \quad \text{and } prnt_G(j) \in V_G \\ prnt_D(w) & \text{if } w \in V_D \text{ and } prnt_D(w) \in V_D \\ prnt_G(j) & \text{if } w \in V_D \text{ and } prnt_D(w) = j \in k_G \\ & \quad \text{and } prnt_G(j) \in V_G \\ prnt_G(w) & \text{if } w \in V_G \text{ and } prnt_G(w) \in V_G \\ prnt_C(j) & \text{if } w \in k_H \text{ and } \pi(w) \in k_D \\ & \quad \text{and } prnt_D(\pi(w)) = i \in k_G \\ & \quad \text{and } prnt_G(i) = j \in m_G \\ prnt_C(m_G - k_D + \pi(w)) & \text{if } w \in k_H \text{ and } \pi(w) \in (k_D + k) \setminus k_D \\ prnt_C(j) & \text{if } w \in V_D \text{ and } prnt_D(w) = i \in k_G \\ & \quad \text{and } prnt_G(i) = j \in m_G \\ prnt_C(j) & \text{if } w \in V_G \text{ and } prnt_G(w) = j \in m_G \\ prnt_C(w) & \text{if } w \in V_C \end{cases}$$

We can now verify that ϕ is a place graph embedding, i.e., that it satisfies the conditions of Def. 5.4:

(PGE-1) Satisfied since Id_{V_G} is an identity map.

(PGE-2) Since $prnt_D : k_D \uplus V_D \to V_D \uplus k_G$, $k_D \subseteq k_H$, $V_D \subseteq V_H$, and $\pi : k_H \to k_H$ we have $\phi^{\mathsf{s}} = (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1} \restriction_{k_G} : k_G \to \mathcal{P}(k_H \uplus V_H)$; it is fully injective since $prnt_D$, Id_{V_D}, and π are functions.

(PGE-3) Since $prnt_C : (m_G + k) \uplus V_C \to V_C \uplus m_H$ and $V_C \subseteq V_H$ we have $\phi^{\mathsf{r}} = prnt_C \restriction_{m_G} : m_G \to V_H \uplus m_H$.

(PGE-4) Satisfied since $\mathsf{rng}(\phi^{\mathsf{v}}) = V_G$, $\mathsf{rng}(\phi^{\mathsf{r}}) = \mathsf{rng}(prnt_C \restriction_{m_G}) \subseteq V_C \uplus m_H$, and $V_G \,\#(V_C \uplus m_H)$.

(PGE-5) Satisfied since $\mathsf{rng}(\phi^{\mathsf{s}}) = \mathsf{rng}((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1} \restriction_{k_G}) \subseteq k_H \uplus V_D$, $\mathsf{rng}(\phi^{\mathsf{v}}) = V_G$, and $V_G \,\#(k_H \uplus V_D)$.

(PGE-6) We have $H \downharpoonleft^{\phi^{\mathsf{s}}(k_G)} \subseteq k_H \uplus V_D$ which can be seen as follows (noting that $prnt_H(w) \in V_D \Rightarrow w \in k_H \uplus V_D$):

$$\begin{aligned} H \downharpoonleft^{\phi^{\mathsf{s}}(k_G)} &= \{c' \mid c' \in k_H \uplus V_H \ \wedge\ \exists i \geq 0 : prnt_H^i(c') \in \phi^{\mathsf{s}}(k_G)\} \\ &= \{c' \mid c' \in k_H \uplus V_H \ \wedge\ \exists i \geq 0 : prnt_H^i(c') \in ((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1})(k_G)\} \\ &= \{c' \mid c' \in k_H \uplus V_H \ \wedge\ \exists i \geq 0 : prnt_H^i(c') \in k_H \uplus V_D\} \\ &= \{c' \mid c' \in k_H \uplus V_D \ \wedge\ \exists i \geq 0 : prnt_H^i(c') \in V_D\} \\ &\subseteq k_H \uplus V_D \end{aligned}$$

Since $\mathsf{rng}(\phi^{\mathsf{r}}) = \mathsf{rng}(prnt_C \restriction_{m_G}) \subseteq V_C \uplus m_H$ and $(k_H \uplus V_D) \,\#(V_C \uplus m_H)$ the condition is satisfied.

(PGE-7) Satisfied since we have the following equalities:

$$\begin{aligned}(\phi^{\mathsf{c}} \circ prnt_G^{-1}\restriction_{V_G})(w) &= \phi^{\mathsf{c}}(prnt_G^{-1}(w)) \\ &= \phi^{\mathsf{s}}(k_G \cap prnt_G^{-1}(w)) \\ &\quad \cup \phi^{\mathsf{v}}(V_G \cap prnt_G^{-1}(w)) \\ &= ((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1}\restriction_{k_G})(k_G \cap prnt_G^{-1}(w)) \\ &\quad \cup (V_G \cap prnt_G^{-1}(w)) \\ &= \{i \in \pi^{-1}(k_D) \mid prnt_D(\pi(i)) = j \in k_G \text{ and } prnt_G(j) = w\} \\ &\quad \cup \{v \in V_D \mid prnt_D(v) = j \in k_G \text{ and } prnt_G(j) = w\} \\ &\quad \cup \{v \in V_G \mid prnt_G(v) = w\} \\ &= prnt_H^{-1}(w) \\ &= (prnt_H^{-1} \circ \mathsf{Id}_{V_G})(w) \\ &= (prnt_H^{-1} \circ \phi^{\mathsf{v}})(w).\end{aligned}$$

(PGE-8) Satisfied since $\phi^{\mathsf{v}} = \mathsf{id}_{V_G}$ and $ctrl_G \subseteq ctrl_H$.

(PGE-9) We check the condition separately for the nodes and sites:

$v \in V_G$: We check the condition separately for the cases where the parent is a node or a root:

$prnt_G(v) \in V_G$:

$$\begin{aligned}(\phi^{\mathsf{f}} \circ prnt_G)(v) &= (\phi^{\mathsf{v}} \circ prnt_H)(v) \\ &= (\mathsf{Id}_{V_G} \circ prnt_H)(v) \\ &= (prnt_H \circ \mathsf{Id}_{V_G})(v) \\ &= (prnt_H \circ \phi^{\mathsf{c}})(v).\end{aligned}$$

$prnt_G(v) \in m_G$:

$$\begin{aligned}(\phi^{\mathsf{f}} \circ prnt_G)(v) &= (\phi^{\mathsf{r}} \circ prnt_G)(v) \\ &= (prnt_C \circ prnt_G \circ \mathsf{Id}_{V_G})(v) \\ &= (prnt_H \circ \phi^{\mathsf{c}})(v).\end{aligned}$$

$i \in k_G$: We check the condition separately for the cases where the parent is a node or a root:

$prnt_G(i) \in V_G$:

$$\begin{aligned}(\phi^{\mathsf{f}} \circ prnt_G)(i) &= (\phi^{\mathsf{v}} \circ prnt_G)(i) \\ &= (\mathsf{Id}_{V_G} \circ prnt_G)(i) \\ &= prnt_G(i) \\ &= prnt_H(((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1}\restriction_{k_G})(i)) \\ &= (prnt_H \circ \phi^{\mathsf{s}})(i) \\ &= (prnt_H \circ \phi^{\mathsf{c}})(i).\end{aligned}$$

$prnt_G(i) \in m_G$:

$$\begin{aligned}(\phi^{\mathsf{f}} \circ prnt_G)(i) &= (\phi^{\mathsf{r}} \circ prnt_G)(i) \\ &= (prnt_C \circ prnt_G)(i) \\ &= prnt_H(((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1}\restriction_{k_G})(i)) \\ &= (prnt_H \circ \phi^{\mathsf{s}})(i) \\ &= (prnt_H \circ \phi^{\mathsf{c}})(i).\end{aligned}$$

□

A.1.3 Proof of Prop. 5.9

We first show that $\mathit{prmt}(\phi)$ and $\mathit{ctxt}(\phi)$ are indeed place graphs:

$\mathit{prmt}(\phi)$: Clearly, $\mathit{ctrl}_H \restriction_{V_D}$ is a control map defined for V_D. We must check that the parent map $\mathit{prnt}_D : k_D \uplus V_D \to V_D \uplus k_G$ is (1) well-defined and (2) acyclic:

1. Since $\phi^{\mathsf{s}} : k_G \to \mathcal{P}(k_H \uplus V_H)$ is fully injective, $(\phi^{\mathsf{s}})^{-1}$ is a function and thus prnt_D is clearly well-defined.

2. It is immediate that prnt_D is acyclic iff $\forall c \in k_D \uplus V_D : \exists i > 0 : \mathit{prnt}^i_D(c) \in k_G$. This is clearly the case for the elements of $(f_D \uplus \mathsf{Id}_{V_D})^{-1}(\mathsf{dom}((\phi^{\mathsf{s}})^{-1}))$ and for the remaining elements it follows from the definition of the subtree operator, cf. Def. 2.9, and the fact that $\mathsf{rng}(\mathit{prnt}_H) \# k_D$.

$\mathit{ctxt}(\phi)$: Clearly, $\mathit{ctrl}_H \restriction_{V_C}$ is a control map defined for V_C. We must check that the parent map $\mathit{prnt}_C : k_C \uplus V_C \to V_C \uplus m_H$ is (1) well-defined and (2) acyclic:

1. The constituent functions have the following domains and codomains:

$$\begin{aligned} \phi^{\mathsf{r}} &: m_G \to V_H \uplus m_H \\ \mathit{prnt}_H \circ f_C &: \{i + m_G \mid i \in |\tilde{k}_C|\} \to V_C \uplus m_H \\ \mathit{prnt}_H \restriction_{V_C} &: V_C \to V_C \uplus m_H \end{aligned}$$

   Since $k_C = m_G + |\tilde{k}_C|$ it is clear that prnt_C is well-defined, but we must show $\mathsf{rng}(\phi^{\mathsf{r}}) \subseteq V_C \uplus m_H$ to know $\mathsf{cod}(\mathit{prnt}_C) = V_C \uplus m_H$. Since $V_C = (V_H \setminus \phi^{\mathsf{v}}(V_G)) \setminus V_D$ and $V_D = V_H \cap H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}$ this amounts to showing $\mathsf{rng}(\phi^{\mathsf{r}}) \# \mathsf{rng}(\phi^{\mathsf{v}})$ and $\mathsf{rng}(\phi^{\mathsf{r}}) \# H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}$, which follows from the fact that ϕ is an embedding and thus satisfies conditions (PGE-4) and (PGE-6).

2. Since prnt_H is acyclic and $\mathsf{rng}(\mathit{prnt}_H) \# k_C$, prnt_C is acyclic.

To see that any valid choices of f_D and f'_C yield equivalent decompositions, cf. Def. 5.7, assume that we have two other bijections

$$g_D : k_D \rightarrowtail \tilde{k}_D \qquad\qquad g'_C : |\tilde{k}_C| \rightarrowtail \tilde{k}_C$$

and construct the corresponding parent maps $\mathit{prnt}_{D'}$ and $\mathit{prnt}_{C'}$ and permutation π'

$$\begin{aligned} \mathit{prnt}_{D'} &= ((\phi^{\mathsf{s}})^{-1} \uplus \mathit{prnt}_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (g_D \uplus \mathsf{Id}_{V_D}) \\ g_C(i + m_G) &= g'_C(i) \quad \text{for } i \in |\tilde{k}_C| \\ \mathit{prnt}_{C'} &= \phi^{\mathsf{r}} \uplus \mathit{prnt}_H \restriction_{V_C} \uplus \mathit{prnt}_H \circ g_C \\ g'(i + k_D) &= g'_C(i) \quad \text{for } i \in |\tilde{k}_C| \\ \pi' &= g_D^{-1} \uplus g'^{-1} : k_H \to k_H. \end{aligned}$$

Finally, we check the equivalence conditions

$$\begin{aligned}
prnt_D \restriction_{V_D} &= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}) \restriction_{V_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \restriction_{V_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (g_D \uplus \mathsf{Id}_{V_D}) \restriction_{V_D} \\
&= prnt_{D'} \restriction_{V_D} \\
prnt_C \restriction_{V_C \uplus m_G} &= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_C) \restriction_{V_C \uplus m_G} \\
&= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C}) \restriction_{V_C \uplus m_G} \\
&= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ g_C) \restriction_{V_C \uplus m_G} \\
&= prnt_{C'} \restriction_{V_C \uplus m_G} \\
prnt_D \circ \pi \downharpoonleft^{k_D} &= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}) \circ \pi \downharpoonleft^{k_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}) \circ (f_D^{-1} \uplus f'^{-1}) \downharpoonleft^{k_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \restriction_{\tilde{k}_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (g_D \uplus \mathsf{Id}_{V_D}) \circ (g_D^{-1} \uplus g'^{-1}) \downharpoonleft^{k_D} \\
&= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (g_D \uplus \mathsf{Id}_{V_D}) \circ \pi' \downharpoonleft^{k_D} \\
&= prnt_{D'} \circ \pi' \downharpoonleft^{k_D} \\
prnt_C(\pi(i) - k_D + m_G) &= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_C)(\pi(i) - k_D + m_G) \\
&= prnt_H(f_C((f_D^{-1} \uplus f'^{-1})(i) - k_D + m_G)) \\
&= prnt_H(f'_C(f'^{-1}(i) - k_D)) \\
&= prnt_H(f'_C({f'_C}^{-1}(i))) \\
&= prnt_H(i) \\
&= prnt_H(g'_C({g'_C}^{-1}(i))) \\
&= prnt_H(g'_C(g'^{-1}(i) - k_D)) \\
&= prnt_H(g_C((g_D^{-1} \uplus g'^{-1})(i) - k_D + m_G)) \\
&= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ g_C)(\pi'(i) - k_D + m_G) \\
&= prnt_{C'}(\pi'(i) - k_D + m_G).
\end{aligned}$$

We now show that $prmt(\phi)$ and $ctxt(\phi)$ are indeed parameter and context for the embedding of G:
Let

$$\begin{aligned}
&D : k_D \to k_G = prmt(\phi), \\
&C : k_C \to m_H = ctxt(\phi), and \\
&(V, ctrl, prnt) = ctxt(\phi) \circ (\phi \blacksquare G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi.
\end{aligned}$$

By the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8) we have the following equalities:

$$\begin{aligned}
V &= V_C \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D && \text{Defs. 2.6 and 2.8} \\
&= ((V_H \setminus \phi^{\mathsf{v}}(V_G)) \setminus V_D) \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D && \text{Def. 5.8} \\
&= V_H
\end{aligned}$$

$$
\begin{aligned}
ctrl &= ctrl_H \restriction_{V_C} \uplus ctrl_G \circ (\phi^{\mathsf{v}})^{-1} \uplus ctrl_H \restriction_{V_D} && \text{Defs. 2.6 and 2.8} \\
&= ctrl_H \restriction_{V_C} \uplus ctrl_H \circ \phi^{\mathsf{v}} \circ (\phi^{\mathsf{v}})^{-1} \uplus ctrl_H \restriction_{V_D} && \text{Condition (PGE-8)} \\
&= ctrl_H \restriction_{V_C} \uplus ctrl_H \restriction_{\phi^{\mathsf{v}}(V_G)} \uplus ctrl_H \restriction_{V_D} && \phi^{\mathsf{v}} \text{ is injective on } V_G \\
&= ctrl_H && V_H = V_C \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D
\end{aligned}
$$

We construct the parent map *prnt* incrementally according to the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8), and then verify $prnt(w) = prnt_H(w)$:

$\phi \blacksquare G$: We write $prnt_{\phi \blacksquare G}$ for the parent map of the resulting place graph:

$$prnt_{\phi \blacksquare G} = (\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}).$$

$\phi \blacksquare G \circ prmt(\phi)$: We write $prnt_1$ for the parent map of the resulting place graph:

$$prnt_1(w) = \begin{cases} prnt_D(w) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) \in V_D \\ prnt_{\phi \blacksquare G}(j) & \text{if } w \in k_D \uplus V_D \text{ and } prnt_D(w) = j \in k_G \\ prnt_{\phi \blacksquare G}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_G) \end{cases}.$$

$\phi \blacksquare G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}$: We write $prnt_2$ for the parent map of the resulting place graph:

$$prnt_2 = prnt_1 \uplus prnt'_{\mathsf{id}_{|\tilde{k}_C|}}$$

where

$$prnt'_{\mathsf{id}_{|\tilde{k}_C|}}(k_D + i) = m_G + i \quad \text{for } i \in |\tilde{k}_C|.$$

$(\phi \blacksquare G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi$: We write $prnt_3$ for the parent map of the resulting place graph:

$$
\begin{aligned}
prnt_3(w) &= \begin{cases} \pi(w) & \text{if } w \in k_H \uplus \emptyset \text{ and } \pi(w) \in \emptyset \\ prnt_2(j) & \text{if } w \in k_H \uplus \emptyset \text{ and } \pi(w) = j \in k_H \\ prnt_2(w) & \text{if } w \in \phi^{\mathsf{v}}(V_G) \uplus V_D \end{cases} \\
&= \begin{cases} prnt_2(j) & \text{if } w \in k_H \text{ and } \pi(w) = j \in k_H \\ prnt_1(w) & \text{if } w \in \phi^{\mathsf{v}}(V_G) \uplus V_D \end{cases} \\
&= \begin{cases} prnt_1(j) & \text{if } w \in k_H \text{ and } \pi(w) = j \in k_D \\ prnt'_{\mathsf{id}_{|\tilde{k}_C|}}(j) & \text{if } w \in k_H \text{ and } \pi(w) = j \in k_H \setminus k_D \\ prnt_1(w) & \text{if } w \in \phi^{\mathsf{v}}(V_G) \uplus V_D \end{cases} \\
&= \begin{cases} prnt_1(j) & \text{if } w \in k_H \text{ and } \pi(w) = j \in k_D \\ j + m_G - k_D & \text{if } w \in k_H \text{ and } \pi(w) = j \in k_H \setminus k_D \\ prnt_1(w) & \text{if } w \in \phi^{\mathsf{v}}(V_G) \uplus V_D \end{cases}
\end{aligned}
$$

$ctxt(\phi) \circ (\phi \blacksquare G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi$: Finally, we have

$$prnt(w) = \begin{cases} prnt_3(w) & \text{if } w \in k_H \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D \text{ and } prnt_3(w) \in \phi^{\mathsf{v}}(V_G) \uplus V_D \\ prnt_C(j) & \text{if } w \in k_H \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D \text{ and } prnt_3(w) = j \in k_C \\ prnt_C(w) & \text{if } w \in V_C \end{cases}.$$

Let us now verify $\mathit{prnt} = \mathit{prnt}_H$. prnt is defined for $w \in k_H \uplus V_C \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D$ and so is prnt_H since $V_C \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D = V_H$, so let us consider $\mathit{prnt}(w)$ in each case (noting that $k_H = \tilde{k}_C \uplus \tilde{k}_D$):

$w \in \tilde{k}_C$**:** This implies

$$\begin{aligned}
& & \pi(w) &= f'^{-1}(w) \\
&\Leftrightarrow & k_D \leq \pi(w) &< k_D + |\tilde{k}_C| \\
&\Rightarrow & \mathit{prnt}_3(w) &= \pi(w) + m_G - k_D \text{ and } m_G \leq \mathit{prnt}_3(w) < m_G + |\tilde{k}_C| \\
&\Rightarrow & \mathit{prnt}(w) &= \mathit{prnt}_C(\pi(w) + m_G - k_D) \\
& & &= (\mathit{prnt}_H \circ f_C)(\pi(w) + m_G - k_D) \\
& & &= (\mathit{prnt}_H \circ f_C)(f'^{-1}(w) + m_G - k_D) \\
& & &= (\mathit{prnt}_H \circ f_C)((f'_C)^{-1}(w) + m_G) \\
& & &= (\mathit{prnt}_H \circ f_C)((f_C)^{-1}(w)) \\
& & &= \mathit{prnt}_H(w)
\end{aligned}$$

$w \in \tilde{k}_D$**:** This implies

$$\begin{aligned}
& & \pi(w) &= (f_D)^{-1}(w) \in k_D \\
&\Rightarrow & \mathit{prnt}_3(w) &= \mathit{prnt}_1(\pi(w))
\end{aligned}$$

which further divides into two cases:

$\mathit{prnt}_D(\pi(w)) \in V_D$**:** This implies

$$\begin{aligned}
& & \mathit{prnt}_1(\pi(w)) &= \mathit{prnt}_D(\pi(w)) \in V_D \\
&\Rightarrow & \mathit{prnt}(w) &= \mathit{prnt}_3(w) = \mathit{prnt}_D(\pi(w)) \\
& & &= \mathit{prnt}_D((f_D)^{-1}(w)) \\
& & &= (((\phi^{\mathsf{s}})^{-1} \uplus \mathit{prnt}_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \\
& & &\qquad \circ (f_D \uplus \mathsf{Id}_{V_D}))((f_D)^{-1}(w)) \\
& & &= \mathit{prnt}_H(w)
\end{aligned}$$

$\mathit{prnt}_D(\pi(w)) \in k_G$**:** This implies

$$\begin{aligned}
\mathit{prnt}_1(\pi(w)) &= \mathit{prnt}_{\phi \blacksquare G}(\mathit{prnt}_D(\pi(w))) \\
&= \mathit{prnt}_{\phi \blacksquare G}(\mathit{prnt}_D((f_D)^{-1}(w))) \\
&= \mathit{prnt}_{\phi \blacksquare G}((((\phi^{\mathsf{s}})^{-1} \uplus \mathit{prnt}_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \\
&\qquad \circ (f_D \uplus \mathsf{Id}_{V_D}))((f_D)^{-1}(w))) \\
&= \mathit{prnt}_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w))
\end{aligned}$$

which again divides into two cases:

$prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \in \phi^{\mathsf{v}}(V_G)$**:** This implies

$$\begin{aligned}
prnt(w) &= prnt_3(w) = prnt_1(\pi(w)) = prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \\
&= ((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}))((\phi^{\mathsf{s}})^{-1}(w)) \\
&= (\phi^{\mathsf{v}} \circ prnt_G \downharpoonright^{V_G})((\phi^{\mathsf{s}})^{-1}(w)) \\
&= (prnt_H \circ \phi^{\mathsf{s}})((\phi^{\mathsf{s}})^{-1}(w)) \\
&= prnt_H(w)
\end{aligned}$$

$prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \in k_C$**:** This implies

$$\begin{aligned}
prnt(w) &= prnt_C(prnt_3(w)) = prnt_C(prnt_1(\pi(w))) \\
&= prnt_C(prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w))) \\
&= prnt_C(((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}))((\phi^{\mathsf{s}})^{-1}(w))) \\
&= prnt_C(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\
&= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_C)(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\
&= \phi^{\mathsf{r}}(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\
&= (prnt_H \circ \phi^{\mathsf{s}})((\phi^{\mathsf{s}})^{-1}(w)) \\
&= prnt_H(w)
\end{aligned}$$

$w \in V_D$**:** This implies

$$prnt_3(w) = prnt_1(w)$$

which further divides into two cases:

$prnt_D(w) \in V_D$**:** This implies

$$\begin{aligned}
& & prnt_1(w) &= prnt_D(w) \in V_D \\
&\Rightarrow & prnt(w) &= prnt_3(w) = prnt_D(w) \\
& & &= (((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}))(w) \\
& & &= (prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})})(w) \\
& & &= prnt_H(w)
\end{aligned}$$

$prnt_D(w) \in k_G$**:** This implies

$$\begin{aligned}
prnt_1(w) &= prnt_{\phi \blacksquare G}(prnt_D(w)) \\
&= prnt_{\phi \blacksquare G}((((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}))(w)) \\
&= prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w))
\end{aligned}$$

which again divides into two cases:

$prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \in \phi^{\mathsf{v}}(V_G)$**:** This implies

$$\begin{aligned}
prnt(w) &= prnt_3(w) = prnt_1(w) = prnt_{\phi \blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \\
&= ((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}))((\phi^{\mathsf{s}})^{-1}(w)) \\
&= (\phi^{\mathsf{v}} \circ prnt_G \downharpoonright^{V_G})((\phi^{\mathsf{s}})^{-1}(w)) \\
&= (prnt_H \circ \phi^{\mathsf{s}})((\phi^{\mathsf{s}})^{-1}(w)) \\
&= prnt_H(w)
\end{aligned}$$

$prnt_{\phi\blacksquare G}((\phi^{\mathsf{s}})^{-1}(w)) \in k_C$**:** This implies

$$\begin{aligned} prnt(w) &= prnt_C(prnt_3(w)) = prnt_C(prnt_1(w)) \\ &= prnt_C(prnt_{\phi\blacksquare G}((\phi^{\mathsf{s}})^{-1}(w))) \\ &= prnt_C(((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}))((\phi^{\mathsf{s}})^{-1}(w))) \\ &= prnt_C(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\ &= (\phi^{\mathsf{r}} \uplus prnt_H\restriction_{V_C} \uplus prnt_H \circ f_C)(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\ &= \phi^{\mathsf{r}}(prnt_G((\phi^{\mathsf{s}})^{-1}(w))) \\ &= (prnt_H \circ \phi^{\mathsf{s}})((\phi^{\mathsf{s}})^{-1}(w)) \\ &= prnt_H(w) \end{aligned}$$

$w \in V_C$**:**

$$\begin{aligned} prnt(w) &= prnt_C(w) && \text{cf. def. of } prnt \\ &= prnt_H(w) && \text{cf. def. of } prnt_C \end{aligned}$$

$w \in \phi^{\mathsf{v}}(V_G)$**:**

$$\begin{aligned} prnt_3(w) &= prnt_1(w) = prnt_{\phi\blacksquare G}(w) \\ &= ((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ ((\phi^{\mathsf{v}})^{-1} \uplus \mathsf{Id}_{k_G}))(w) \\ &= ((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w) \end{aligned}$$

There are two cases for $prnt_3(w)$:

$prnt_3(w) \in \phi^{\mathsf{v}}(V_G) \uplus V_D$**:** This implies

$$\begin{aligned} prnt(w) &= prnt_3(w) = prnt_1(w) = prnt_{\phi\blacksquare G}(w) \\ &= ((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w) \\ &= (\phi^{\mathsf{v}} \circ prnt_G\downharpoonleft^{V_G} \circ (\phi^{\mathsf{v}})^{-1})(w) \\ &= (prnt_H \circ \phi^{\mathsf{v}} \circ (\phi^{\mathsf{v}})^{-1})(w) \\ &= prnt_H(w) \end{aligned}$$

$prnt_3(w) \in k_C$**:** This implies

$$\begin{aligned} prnt(w) &= prnt_C(prnt_3(w)) = prnt_C(prnt_1(w)) = prnt_C(prnt_{\phi\blacksquare G}(w)) \\ &= prnt_C(((\phi^{\mathsf{v}} \uplus \mathsf{Id}_{m_G}) \circ prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w)) \\ &= prnt_C((prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w)) \\ &= (\phi^{\mathsf{r}} \uplus prnt_H\restriction_{V_C} \uplus prnt_H \circ f_C)((prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w)) \\ &= \phi^{\mathsf{r}}((prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w)) \\ &= (\phi^{\mathsf{r}} \circ prnt_G \circ (\phi^{\mathsf{v}})^{-1})(w) \\ &= (prnt_H \circ \phi^{\mathsf{v}} \circ (\phi^{\mathsf{v}})^{-1})(w) \\ &= prnt_H(w) \end{aligned}$$

□

A.1.4 Proof of Theorem 5.10

Def. 5.5 $\circ$ Def. 5.8 $= \mathsf{Id}$: Assume

$$\begin{aligned}
H &= C \circ (G \circ D \otimes \mathsf{id}_k) \circ \pi \\
\phi &= \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} : G \hookrightarrow H \\
\phi^{\mathsf{v}} &= \mathsf{Id}_{V_G} \\
\phi^{\mathsf{r}} &= \mathit{prnt}_C \restriction_{m_G} \\
\phi^{\mathsf{s}} &= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ \mathit{prnt}_D^{-1} \restriction_{k_G}
\end{aligned}$$

and the results from the proof of Prop. 5.6.

Now, using construction Def. 5.8 we obtain:

$$\begin{aligned}
\mathit{prmt}(\phi) &= (V_{D'}, \mathit{ctrl}_H \restriction_{V_{D'}}, \mathit{prnt}_{D'}) : k_{D'} \to k_G \qquad \text{where} \\
V_{D'} &= V_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
\tilde{k}_{D'} &= k_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
k_{D'} &= |\tilde{k}_{D'}| \\
f_{D'} &: k_{D'} \rightarrowtail \tilde{k}_{D'} \qquad \text{a bijection} \\
\mathit{prnt}_{D'} &= ((\phi^{\mathsf{s}})^{-1} \uplus \mathit{prnt}_H \restriction_{(V_{D'} \uplus \tilde{k}_{D'}) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_{D'} \uplus \mathsf{Id}_{V_{D'}})
\end{aligned}$$

$$\begin{aligned}
\mathit{ctxt}(\phi) &= (V_{C'}, \mathit{ctrl}_H \restriction_{V_{C'}}, \mathit{prnt}_{C'}) : k_{C'} \to m_H \qquad \text{where} \\
V_{C'} &= (V_H \setminus \phi^{\mathsf{v}}(V_G)) \setminus V_{D'} \\
\tilde{k}_{C'} &= k_H \setminus \tilde{k}_{D'} \\
k_{C'} &= m_G + |\tilde{k}_{C'}| \\
f'_{C'} &: |\tilde{k}_{C'}| \rightarrowtail \tilde{k}_{C'} \qquad \text{a bijection} \\
f_{C'}(i + m_G) &= f'_{C'}(i) \quad \text{for } i \in |\tilde{k}_{C'}| \\
\mathit{prnt}_{C'} &= \phi^{\mathsf{r}} \uplus \mathit{prnt}_H \restriction_{V_{C'}} \uplus \mathit{prnt}_H \circ f_{C'}
\end{aligned}$$

$$\begin{aligned}
H &= \mathit{ctxt}(\phi) \circ (\phi \blacksquare G \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_{C'}|}) \circ \pi' \\
&= \mathit{ctxt}(\phi) \circ (G \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_{C'}|}) \circ \pi' \\
\pi' &= f_{D'}^{-1} \uplus f'^{-1} : k_H \to k_H \\
f'(i + k_{D'}) &= f'_{C'}(i) \quad \text{for } i \in |\tilde{k}_{C'}|.
\end{aligned}$$

We must show $D = \mathit{prmt}(\phi)$, $C = \mathit{ctxt}(\phi)$, and $\pi = \pi'$. First, let us unfold some of the definitions:

$$
\begin{aligned}
\mathsf{rng}(\phi^{\mathsf{s}}) &= \mathsf{rng}((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ \mathit{prnt}_D^{-1}\restriction_{k_G}) \\
&= (\mathsf{Id}_{V_D} \uplus \pi^{-1})(\{c \mid c \in k_D \uplus V_D \wedge \mathit{prnt}_D(c) \in k_G\}) \\
&= \{c \mid (c \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D(\pi(c)) \in k_G) \vee (c \in V_D \wedge \mathit{prnt}_D(c) \in k_G)\} \\
H\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} &= \{c' \mid c' \in k_H \uplus V_H \ \wedge\ \exists i \geq 0 : \mathit{prnt}_H^i(c') \in \mathsf{rng}(\phi^{\mathsf{s}})\} \\
&= \{c' \mid c' \in k_H \uplus V_H \\
&\qquad \wedge \exists i \geq 0 : (\mathit{prnt}_H^i(c') \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D(\pi(\mathit{prnt}_H^i(c'))) \in k_G) \\
&\qquad\quad \vee (\mathit{prnt}_H^i(c') \in V_D \wedge \mathit{prnt}_D(\mathit{prnt}_H^i(c')) \in k_G)\} \\
&= \{c' \mid \exists i > 0 : (c' \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D(\pi(c')) \in k_G) \\
&\qquad\quad \vee (c' \in V_D \wedge \mathit{prnt}_D(c') \in k_G) \\
&\qquad\quad \vee (c' \in \pi^{-1}(k_D) \wedge \mathit{prnt}_H^{i-1}(\mathit{prnt}_D(\pi(c'))) \in V_D \wedge \mathit{prnt}_D(\mathit{prnt}_H^{i-1}(\mathit{prnt}_D(\pi(c')))) \in k_G) \\
&\qquad\quad \vee (c' \in V_D \wedge \mathit{prnt}_H^{i-1}(\mathit{prnt}_D(c')) \in V_D \wedge \mathit{prnt}_D(\mathit{prnt}_H^{i-1}(\mathit{prnt}_D(c'))) \in k_G)\} \\
&= \{c' \mid \exists i > 0 : (c' \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D(\pi(c')) \in k_G) \\
&\qquad\quad \vee (c' \in V_D \wedge \mathit{prnt}_D(c') \in k_G) \\
&\qquad\quad \vee (c' \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D^{i-1}(\mathit{prnt}_D(\pi(c'))) \in V_D \wedge \mathit{prnt}_D(\mathit{prnt}_D^{i-1}(\mathit{prnt}_D(\pi(c')))) \in k_G) \\
&\qquad\quad \vee (c' \in V_D \wedge \mathit{prnt}_D^{i-1}(\mathit{prnt}_D(c')) \in V_D \wedge \mathit{prnt}_D(\mathit{prnt}_D^{i-1}(\mathit{prnt}_D(c'))) \in k_G)\} \\
&= \{c' \mid \exists i > 0 : (c' \in \pi^{-1}(k_D) \wedge \mathit{prnt}_D^i(\pi(c')) \in k_G) \\
&\qquad\quad \vee (c' \in V_D \wedge \mathit{prnt}_D^i(c') \in k_G)\} \\
&= \pi^{-1}(k_D) \uplus V_D \\
\tilde{k}_{D'} &= k_H \cap H\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} = k_H \cap (\pi^{-1}(k_D) \uplus V_D) = \pi^{-1}(k_D) \\
k_{D'} &= |\tilde{k}_{D'}| = |\pi^{-1}(k_D)| = k_D \\
f_{D'} &: k_D = k_{D'} \quad \rightarrowtail \quad \tilde{k}_{D'} = \pi^{-1}(k_D) \\
\tilde{k}_{C'} &= k_H \setminus \tilde{k}_{D'} = k_H \setminus \pi^{-1}(k_D) = \pi^{-1}(k_H \setminus k_D) \\
k_{C'} &= m_G + |\tilde{k}_{C'}| = m_G + |\pi^{-1}(k_H \setminus k_D)| = m_G + k_H - k_D \\
f'_{C'} &: (k_H - k_D) = |\tilde{k}_{C'}| \quad \rightarrowtail \quad \tilde{k}_{C'} = \pi^{-1}(k_H \setminus k_D) \\
f_{C'}(i + m_G) &= f'_{C'}(i) \quad \text{for } i \in |\tilde{k}_{C'}| = (k_H - k_D) \\
f'(i + k_{D'}) &= f'_{C'}(i) \quad \text{for } i \in |\tilde{k}_{C'}| = (k_H - k_D)
\end{aligned}
$$

With these in mind, we proceed to prove $D = \mathit{prmt}(\phi)$, $C = \mathit{ctxt}(\phi)$, and $\pi = \pi'$:

$\pi = \pi'$: Remember that in Def. 5.8 we are free to choose the two bijections $f_{D'}$,$f'_{C'}$ as they are internal to the decomposition. We choose them to be suitable restrictions of the inverse of π:

$$
\begin{aligned}
f_{D'} &= \pi^{-1}\restriction_{k_D} : k_D \rightarrowtail \pi^{-1}(k_D) \\
f'_{C'}(i) &= \pi^{-1}(i + k_D) : (k_H - k_D) \rightarrowtail \pi^{-1}(k_H \setminus k_D)
\end{aligned}
$$

Expanding these in the derived functions we get:

$$\begin{aligned}
f_{C'}(i+m_G) &= f'_{C'}(i) = \pi^{-1}(i+k_D) \\
f'(i) &= f'_{C'}(i-k_D) = \pi^{-1}(i-k_D+k_D) = \pi^{-1}(i) \quad \text{for } (i-k_D) \in |\tilde{k}_{C'}| = k_H - k_D \\
\pi' &= f_{D'}^{-1} \uplus f'^{-1} \\
\pi'(i) &= \begin{cases} f_{D'}^{-1}(i) & \text{if } i \in \pi^{-1}(k_D) \\ f'^{-1}(i) & \text{if } i \in \pi^{-1}(k_H \setminus k_D) \end{cases} \\
&= \begin{cases} \pi(i) & \text{if } i \in \pi^{-1}(k_D) \\ \pi(i) & \text{if } i \in \pi^{-1}(k_H \setminus k_D) \end{cases} \\
&= \pi(i).
\end{aligned}$$

$D = prmt(\phi)$**:** It suffices to show $V_{D'} = V_D$ and $prnt_{D'} = prnt_D$, which is easily seen by unfolding the definitions:

$$\begin{aligned}
V_{D'} &= V_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} = V_H \cap (\pi^{-1}(k_D) \uplus V_D) = V_D \\
prnt_{D'} &= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_{D'} \uplus \tilde{k}_{D'}) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_{D'} \uplus \mathsf{Id}_{V_{D'}}) \\
&= (((\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1} \restriction_{k_G})^{-1} \\
&\qquad \uplus prnt_H \restriction_{(V_D \uplus \pi^{-1}(k_D)) \setminus (\{c \mid (c \in \pi^{-1}(k_D) \wedge prnt_D(\pi(c)) \in k_G) \vee (c \in V_D \wedge prnt_D(c) \in k_G)\})}) \\
&\quad \circ (f_{D'} \uplus \mathsf{Id}_{V_D}) \\
&= ((prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi)) \restriction_{\{c \mid (c \in \pi^{-1}(k_D) \wedge prnt_D(\pi(c)) \in k_G) \vee (c \in V_D \wedge prnt_D(c) \in k_G)\}} \\
&\qquad \uplus (prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi)) \restriction_{\{c \mid (c \in \pi^{-1}(k_D) \wedge prnt_D(\pi(c)) \notin k_G) \vee (c \in V_D \wedge prnt_D(c) \notin k_G)\})}) \\
&\quad \circ (f_{D'} \uplus \mathsf{Id}_{V_D}) \\
&= ((prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi)) \restriction_{\{c \mid (c \in \pi^{-1}(k_D) \wedge prnt_D(\pi(c)) \in k_G) \vee (c \in V_D \wedge prnt_D(c) \in k_G)\}} \\
&\qquad \uplus (prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi)) \restriction_{\{c \mid (c \in \pi^{-1}(k_D) \wedge prnt_D(\pi(c)) \notin k_G) \vee (c \in V_D \wedge prnt_D(c) \notin k_G)\})}) \\
&\quad \circ (f_{D'} \uplus \mathsf{Id}_{V_D}) \\
&= prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi) \circ (f_{D'} \uplus \mathsf{Id}_{V_D}) \\
&= prnt_D \circ (\mathsf{Id}_{V_D} \uplus \pi' \circ f_{D'}) \\
&= prnt_D \circ (\mathsf{Id}_{V_D} \uplus (f_{D'}^{-1} \uplus f'^{-1}) \circ f_{D'}) \\
&= prnt_D\,.
\end{aligned}$$

$C = ctxt(\phi)$**:** It suffices to show $V_{C'} = V_C$ and $prnt_{C'} = prnt_C$, which is seen by unfolding the definitions:

$$\begin{aligned}
V_{C'} &= (V_H \setminus \phi^{\mathsf{v}}(V_G)) \setminus V_{D'} = (V_H \setminus V_G) \setminus V_D = V_C \\
(prnt_H \circ f_{C'})(i) &= prnt_C(m_G + \pi(f_{C'}(i))) \quad \text{for } i \in k_H - k_D \\
&= prnt_C(m_G + \pi(\pi^{-1}(i - m_G + k_D))) \\
&= prnt_C(m_G + i - m_G + k_D) \\
&= prnt_C(i + k_D) \\
prnt_C(i) &= prnt_H(\pi^{-1}(i - m_G + k_D)) \quad \text{for } i \in (m_G + k_H - k_D) \setminus m_G \\
&= (prnt_H \circ f_{C'})(i) \\
prnt_{C'} &= \phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_{C'}} \uplus prnt_H \circ f_{C'} \\
&= prnt_C \restriction_{m_G} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_{C'} \\
&= prnt_C \restriction_{m_G} \uplus prnt_C \restriction_{V_C} \uplus prnt_C \restriction_{(m_G + k_H - k_D) \setminus m_G} \\
&= prnt_C\,.
\end{aligned}$$

Def. 5.5 $\circ$ Def. 5.8 $=$ Id: Assume a place graph $G : k_G \to m_G$, an embedding $\phi : G \hookrightarrow H$ into a place graph $H : k_H \to m_H$ (for simplicity, assume $\phi \mathbin{\scriptstyle\blacksquare} G = G$),

$$\begin{aligned}
prmt(\phi) &= (V_D, ctrl_H \restriction_{V_D}, prnt_D) : k_D \to k_G \quad \text{where} \\
V_D &= V_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
\tilde{k}_D &= k_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \\
k_D &= |\tilde{k}_D| \\
f_D &: k_D \rightarrowtail \tilde{k}_D \quad \text{a bijection} \\
prnt_D &= ((\phi^{\mathsf{s}})^{-1} \uplus prnt_H \restriction_{(V_D \uplus \tilde{k}_D) \setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D})
\end{aligned}$$

$$\begin{aligned}
ctxt(\phi) &= (V_C, ctrl_H \restriction_{V_C}, prnt_C) : k_C \to m_H \quad \text{where} \\
V_C &= (V_H \setminus V_G) \setminus V_D \\
\tilde{k}_C &= k_H \setminus \tilde{k}_D \\
k_C &= m_G + |\tilde{k}_C| \\
f'_C &: |\tilde{k}_C| \rightarrowtail \tilde{k}_C \quad \text{a bijection} \\
f_C(i + m_G) &= f'_C(i) \quad \text{for } i \in |\tilde{k}_C| \\
prnt_C &= \phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_C \\
H &= ctxt(\phi) \circ (G \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi \\
\pi &= f_D^{-1} \uplus f'^{-1} : k_H \to k_H \\
f'(i + k_D) &= f'_C(i) \quad \text{for } i \in |\tilde{k}_C|
\end{aligned}$$

where we assume that the bijections are chosen as in the previous proof case, i.e.,

$$\begin{aligned}
f_D &= \pi^{-1} \restriction_{k_D} : k_D \rightarrowtail \pi^{-1}(k_D) \\
f'_C(i) &= \pi^{-1}(i + k_D) : (k_H - k_D) \rightarrowtail \pi^{-1}(k_H \setminus k_D).
\end{aligned}$$

Now, using construction Def. 5.5 we obtain:

$$\begin{aligned}
\phi' &= \phi'^{\mathsf{v}} \uplus \phi'^{\mathsf{s}} \uplus \phi'^{\mathsf{r}} : G \hookrightarrow H \\
\phi'^{\mathsf{v}} &= \mathsf{Id}_{V_G} \\
\phi'^{\mathsf{r}} &= prnt_C \restriction_{m_G} \\
\phi'^{\mathsf{s}} &= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ prnt_D^{-1} \restriction_{k_G} .
\end{aligned}$$

We must prove $\phi = \phi'$, and it suffices to show $\phi^{\mathsf{v}} = \phi'^{\mathsf{v}}$, $\phi^{\mathsf{r}} = \phi'^{\mathsf{r}}$, and $\phi^{\mathsf{s}} = \phi'^{\mathsf{s}}$.

$\phi^{\mathsf{v}} = \phi'^{\mathsf{v}}$: Satisfied by assumption.

$\phi^{\mathsf{r}} = \phi'^{\mathsf{r}}$: Easily seen by unfolding the definitions:

$$\begin{aligned}
\phi'^{\mathsf{r}} &= prnt_C \restriction_{m_G} \\
&= (\phi^{\mathsf{r}} \uplus prnt_H \restriction_{V_C} \uplus prnt_H \circ f_C) \restriction_{m_G} \\
&= \phi^{\mathsf{r}}.
\end{aligned}$$

$\phi^{\mathsf{s}} = \phi'^{\mathsf{s}}$: Easily seen by unfolding the definitions:

$$\begin{aligned}
\phi'^{\mathsf{s}} &= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ \mathit{prnt}_D^{-1} \restriction_{k_G} \\
&= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ (((\phi^{\mathsf{s}})^{-1} \uplus \mathit{prnt}_H \restriction_{(V_D \uplus \tilde{k}_D)\setminus \mathsf{rng}(\phi^{\mathsf{s}})}) \circ (f_D \uplus \mathsf{Id}_{V_D}))^{-1} \restriction_{k_G} \\
&= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ (f_D^{-1} \uplus \mathsf{Id}_{V_D}) \circ (\phi^{\mathsf{s}} \uplus (\mathit{prnt}_H \restriction_{(V_D \uplus \tilde{k}_D)\setminus \mathsf{rng}(\phi^{\mathsf{s}})})^{-1}) \restriction_{k_G} \\
&= (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ (\pi \uplus \mathsf{Id}_{V_D}) \circ \phi^{\mathsf{s}} \\
&= \phi^{\mathsf{s}}.
\end{aligned}$$

A.1.5 Proof of Lemma 5.11

1: We show that $v \in H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} = \{v \mid v \in V_H \;\wedge\; \exists i \geq 0 : \mathit{prnt}_H^i(v) \in \phi^{\mathsf{s}}(k_G)\}$ implies $v \notin \mathsf{rng}(\phi^{\mathsf{f}})$ by induction on i:

$i = 0$**:** We have $v \in \mathsf{rng}(\phi^{\mathsf{s}})$ and conditions (PGE-5) and (PGE-6) then give us $v \notin \mathsf{rng}(\phi^{\mathsf{f}})$.

$i > 0$**:** We have $\mathit{prnt}_H^i(v) \in \mathsf{rng}(\phi^{\mathsf{s}})$, i.e., $\mathit{prnt}_H^{i-1}(\mathit{prnt}_H(v)) \in \mathsf{rng}(\phi^{\mathsf{s}})$ which by the induction hypothesis means $\mathit{prnt}_H(v) \notin \mathsf{rng}(\phi^{\mathsf{f}})$. Thus condition (PGE-9) cannot be satisfied if $v \in \mathsf{rng}(\phi^{\mathsf{v}})$ and condition (PGE-6) prevents $v \in \mathsf{rng}(\phi^{\mathsf{r}})$, so $v \notin \mathsf{rng}(\phi^{\mathsf{f}})$.

2: First, we show that $\mathit{prnt}_G^i(c) = p$ $(i > 0)$ implies $\mathit{prnt}_H^i(\phi^{\mathsf{c}}(c)) = \phi^{\mathsf{f}}(p)$. We show this by induction on i:

$i = 1$**:** Assuming $\mathit{prnt}_G(c) = p$, we have $\phi^{\mathsf{f}}(\mathit{prnt}_G(c)) = \phi^{\mathsf{f}}(p)$ and thus, by condition (PGE-9), $\mathit{prnt}_H(\phi^{\mathsf{c}}(c)) = \phi^{\mathsf{f}}(p)$ as required.

$i > 1$**:** Assuming $\mathit{prnt}_G^i(c) = p$, we have $\mathit{prnt}_G^{i-1}(\mathit{prnt}_G(c)) = p$ and thus, by the induction hypothesis, $\mathit{prnt}_H^{i-1}(\phi^{\mathsf{c}}(\mathit{prnt}_G(c))) = \phi^{\mathsf{f}}(p)$. Since $\mathsf{dom}(\phi^{\mathsf{c}}) \cap \mathsf{cod}(\mathit{prnt}_G) = V_G$, $\phi^{\mathsf{f}} = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{r}}$ and $\phi^{\mathsf{c}} = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}}$, we have

$$\begin{aligned}
&\mathit{prnt}_H^{i-1}(\phi^{\mathsf{c}}(\mathit{prnt}_G(c))) \\
=&\mathit{prnt}_H^{i-1}(\phi^{\mathsf{v}}(\mathit{prnt}_G(c))) \\
=&\mathit{prnt}_H^{i-1}(\phi^{\mathsf{f}}(\mathit{prnt}_G(c))) \\
=&\phi^{\mathsf{f}}(p)
\end{aligned}$$

and thus, by condition (PGE-9), $\mathit{prnt}_H^{i-1}(\mathit{prnt}_H(\phi^{\mathsf{c}}(c))) = \mathit{prnt}_H^i(\phi^{\mathsf{c}}(c)) = \phi^{\mathsf{f}}(p)$ as required.

Now, since prnt_G is acyclic we have:

$$c \in V_G \uplus k_G \Rightarrow \exists i > 0 : \mathit{prnt}_G^i(c) \in m_G$$

and thus, using the above result, we have

$$c \in V_G \uplus k_G \Rightarrow \exists i > 0 : \mathit{prnt}_H^i(\phi^{\mathsf{c}}(c)) \in \phi^{\mathsf{r}}(m_G) = \mathsf{rng}(\phi^{\mathsf{r}})$$

i.e., for any $c \in \mathsf{rng}(\phi^{\mathsf{c}})$ we have

$$\exists i > 0 : \mathit{prnt}_H^i(c) \in \mathsf{rng}(\phi^{\mathsf{r}}).$$

But this cannot be satisfied by any $c \in H \restriction_{\mathsf{rng}(\phi^{\mathsf{r}})}$, since Prop. 2.11 gives us

$$\begin{aligned}
H \restriction_{\mathsf{rng}(\phi^{\mathsf{r}})} = (&V_H \uplus k_H \uplus m_H) \\
&\setminus \{c' \mid c' \in k_H \uplus V_H \;\wedge\; \exists i > 0 : \mathit{prnt}_H^i(c') \in \mathsf{rng}(\phi^{\mathsf{r}})\}
\end{aligned}$$

i.e., $c \in H \restriction_{\mathsf{rng}(\phi^{\mathsf{r}})}$ iff $\forall i > 0 : \mathit{prnt}_H^i(c) \notin \mathsf{rng}(\phi^{\mathsf{r}})$.

3: From the previous proof case, we have that any $c \in \mathsf{rng}(\phi^{\mathsf{c}})$ satisfies

$$\exists i > 0 : \mathit{prnt}^i_H(c) \in \mathsf{rng}(\phi^{\mathsf{r}}).$$

Now, from Prop. 2.11 we have that $c \in H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}$ implies $c \in k_H \uplus V_H$ and $\exists i \geq 0 : \mathit{prnt}^i_H(c) \in \phi^{\mathsf{s}}(k_G)$, which combined the above result and the fact $\phi^{\mathsf{s}}(k_G) \subseteq \mathsf{rng}(\phi^{\mathsf{c}})$ we get

$$\exists i > 0 : \mathit{prnt}^i_H(c) \in \mathsf{rng}(\phi^{\mathsf{r}})$$

which by Prop. 2.11 means $c \notin H \upharpoonright_{\mathsf{rng}(\phi^{\mathsf{r}})}$.

4: Let $i \in k_G$ be a site and $c \in H \downharpoonleft^{\phi^{\mathsf{s}}(i)} \setminus \phi^{\mathsf{s}}(i)$. Then by Def. 2.9

$$\begin{aligned} & H \downharpoonleft^{\phi^{\mathsf{s}}(i)} \setminus \phi^{\mathsf{s}}(i) \\ &= \{c \mid c \in k_H \uplus V_H \ \wedge \ \exists i \geq 0 : \mathit{prnt}^i_H(c) \in \phi^{\mathsf{s}}(i)\} \setminus \phi^{\mathsf{s}}(i) \\ &= \{c \mid c \in k_H \uplus V_H \ \wedge \ \exists i > 0 : \mathit{prnt}^i_H(c) \in \phi^{\mathsf{s}}(i)\}. \end{aligned}$$

We show that $c \in H \downharpoonleft^{\phi^{\mathsf{s}}(i)} \setminus \phi^{\mathsf{s}}(i)$ implies $c \notin \mathsf{rng}(\phi^{\mathsf{c}}) \uplus \mathsf{rng}(\phi^{\mathsf{r}})$ by induction on i:

$i = 1$: We have $\mathit{prnt}_H(c) \in \phi^{\mathsf{s}}(i)$. We obtain a contradiction if $c \in \mathsf{rng}(\phi^{\mathsf{c}}) \uplus \mathsf{rng}(\phi^{\mathsf{r}})$:

- **$c \in \mathsf{rng}(\phi^{\mathsf{c}})$:** $(\phi^{\mathsf{c}})^{-1}(c)$ is defined and so, by condition (PGE-9), $\mathit{prnt}_H(c) = \phi^{\mathsf{f}}(\mathit{prnt}_G((\phi^{\mathsf{c}})^{-1}(c))) \in \phi^{\mathsf{s}}(i)$ which violates conditions (PGE-5) and (PGE-6).
- **$c \in \mathsf{rng}(\phi^{\mathsf{r}})$:** This violates condition (PGE-6).

$i > 1$: We have $\mathit{prnt}^i_H(c) = \mathit{prnt}^{i-1}_H(\mathit{prnt}_H(c)) \in \phi^{\mathsf{s}}(i)$, so by the induction hypothesis $\mathit{prnt}_H(c) \notin \mathsf{rng}(\phi^{\mathsf{c}}) \uplus \mathsf{rng}(\phi^{\mathsf{r}})$. We obtain a contradiction if $c \in \mathsf{rng}(\phi^{\mathsf{c}})$, because then $(\phi^{\mathsf{c}})^{-1}(c)$ is defined and so, by condition (PGE-9), $\mathit{prnt}_H(c) = \phi^{\mathsf{f}}(\mathit{prnt}_G((\phi^{\mathsf{c}})^{-1}(c)))$ which contradicts $\mathit{prnt}_H(c) \notin \mathsf{rng}(\phi^{\mathsf{c}}) \uplus \mathsf{rng}(\phi^{\mathsf{r}})$.

□

A.1.6 Proof of Prop. 5.13

ϕ^{s}: Construct the map of each site $i \in k_G$ as follows:

$$\begin{aligned} \phi^{\mathsf{s}}(i) &= \mathit{children}_{H,i} \setminus \phi^{\mathsf{v}}(\mathit{siblings}_{G,i}) \\ \mathit{children}_{H,i} &= (\mathit{prnt}^{-1}_H \circ \phi^{\mathsf{v}})(\mathit{prnt}_G(i)) \\ \mathit{siblings}_{G,i} &= (\mathit{prnt}^{-1}_G \circ \mathit{prnt}_G)(i) \setminus \{i\}. \end{aligned}$$

This is well-defined since every site of G is guarding, thus $\mathit{prnt}_G(i) \in V_G$, and no sites are siblings. By construction it satisfies condition (PGE-7); it must satisfy the other conditions since ϕ is an embedding and ϕ^{s} is the only embedding of inner names that will satisfy condition (PGE-7): To see that ϕ^{s} is unique, assume that there is a different ϕ'^{s} that satisfies the conditions of Def. 5.4. For the two to be different, there must be some $i \in k_G$ and $c \in k_H \uplus V_H$ with $c \in \phi^{\mathsf{s}}(i), c \notin \phi'^{\mathsf{s}}(i)$ (or vice versa). Since they both satisfy condition (PGE-7) and no root has a site as a child we have:

$$\begin{aligned} \mathit{prnt}_G(i) &\in V_G \\ (\phi^{\mathsf{c}} \circ \mathit{prnt}^{-1}_G \upharpoonright_{V_G})(\mathit{prnt}_G(i)) &= (\mathit{prnt}^{-1}_H \circ \phi^{\mathsf{v}})(\mathit{prnt}_G(i)) \\ &= (\phi'^{\mathsf{c}} \circ \mathit{prnt}^{-1}_G \upharpoonright_{V_G})(\mathit{prnt}_G(i)) \end{aligned}$$

where

$$\begin{aligned} \phi^{\mathsf{c}} &= \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \\ \phi'^{\mathsf{c}} &= \phi^{\mathsf{v}} \uplus \phi'^{\mathsf{s}}. \end{aligned}$$

And since $c \in \phi^{\mathsf{s}}(i)$, and ϕ^{c} and ϕ'^{c} agree on nodes, we must have $c \in \phi'^{\mathsf{s}}(i')$ for some $i' \in prnt_G^{-1} \restriction_{V_G} (prnt_G(i))$. But no sites are siblings, so $i' = i$ and thus $c \in \phi'^{\mathsf{s}}(i)$ which contradicts our assumption that $c \notin \phi'^{\mathsf{s}}(i)$.

ϕ^{r}: Construct the map of each root $j \in m_G$ as follows: choose a node $v \in prnt_G^{-1}(j)$, which is always possible since no root is idle or has a site as child, and let

$$\phi^{\mathsf{r}}(j) = (prnt_H \circ \phi^{\mathsf{v}})(v).$$

By construction it satisfies condition (PGE-9); it must satisfy the other conditions since ϕ is an embedding and ϕ^{r} is the only embedding of outer names that will satisfy condition (PGE-9): To see that ϕ^{r} is unique, assume that there is a different ϕ'^{r} that satisfies the conditions of Def. 5.4. For the two to be different, we must have $\phi^{\mathsf{r}}(j) \neq \phi'^{\mathsf{r}}(j)$ for some $j \in m_G$. But since they both satisfy condition (PGE-9), the following must hold for the node v we chose when defining $\phi^{\mathsf{r}}(j)$:

$$\begin{aligned} \phi^{\mathsf{r}}(j) &= (\phi^{\mathsf{r}} \circ prnt_G)(v) \\ &= (prnt_H \circ \phi^{\mathsf{c}})(v) \\ &= (\phi'^{\mathsf{r}} \circ prnt_G)(v) = \phi'^{\mathsf{r}}(j) \end{aligned}$$

which contradicts our assumption that ϕ^{r} and ϕ'^{r} are different. □

A.1.7 Proof of Prop. 5.17

It is clear that the place graph may be expressed as

$$H^P = C^P \circ (G^P \circ D^P \otimes \mathsf{id}_k) \circ \pi$$

and thus Prop. 5.6 applies, whereby we have that $\phi^P = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} : G^P \hookrightarrow H^P$ is a place graph embedding and that $V_D = V_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})}$. What remains to show is that $\phi^L = \phi^{\mathsf{v}} \uplus \phi^{\mathsf{e}} \uplus \phi^{\mathsf{i}} \uplus \phi^{\mathsf{o}} : G^L \hookrightarrow H^L$ is a link graph embedding and that the two embeddings are consistent.

From the definitions of support translation, composition, tensor product, and $V_D = V_H \cap H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})}$ we have:

$$\begin{aligned} V_H &= V_C \uplus V_G \uplus V_D & ctrl_H &= ctrl_C \uplus ctrl_G \uplus ctrl_D \\ P_H &= P_C \uplus P_G \uplus P_D & C &: \langle m_G + k, Y_G \uplus X_I \uplus X_C \rangle \to \langle m_H, Y_H \rangle \\ E_H &= E_C \uplus E_G & D &: \langle k_D, X_D \rangle \to \langle k_G, X_G \uplus X_I \rangle \\ X_H &= X_D \uplus X'_C & \alpha &: X'_C \to X_C \\ P_D &= P_{H \downharpoonright^{\mathsf{rng}(\phi^{\mathsf{s}})} \cap V_H} \end{aligned}$$

To show that ϕ^L is a link graph embedding, we need to express the link map of H in terms of it decomposition. It is clear that the link graph may be expressed as

$$H^L = C^L \circ ((G^L \otimes \mathsf{id}_{X_I}) \circ D^L \otimes \alpha).$$

We construct the link map incrementally according to the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8)

$G^L \otimes \mathsf{id}_{X_I}$: We write $link_1$ for the link map of the resulting link graph:

$$link_1 = link_G \uplus \mathsf{Id}_{X_I}.$$

$(G^L \otimes \mathsf{id}_{X_I}) \circ D^L$: We write link_2 for the link map of the resulting link graph:

$$\mathit{link}_2(p) = \begin{cases} \mathit{link}_D(p) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) \in \emptyset \\ \mathit{link}_1(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = y \in X_G \uplus X_I \\ \mathit{link}_1(p) & \text{if } p \in P_G \end{cases}$$

$$= \begin{cases} \mathit{link}_G(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = y \in X_G \\ y & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = y \in X_I \\ \mathit{link}_G(p) & \text{if } p \in P_G. \end{cases}$$

$(G^L \otimes \mathsf{id}_{X_I}) \circ D^L \otimes \alpha$: We write link_3 for the link map of the resulting link graph:

$$\mathit{link}_3 = \mathit{link}_2 \uplus \alpha.$$

$H^L = C^L \circ ((G^L \otimes \mathsf{id}_{X_I}) \circ D^L \otimes \alpha)$:

$$\mathit{link}_H = \begin{cases} \mathit{link}_3(p) & \text{if } p \in X_D \uplus X'_C \uplus P_D \uplus P_G \text{ and } \mathit{link}_3(p) \in E_G \\ \mathit{link}_C(y) & \text{if } p \in X_D \uplus X'_C \uplus P_D \uplus P_G \text{ and } \mathit{link}_3(p) = y \in Y_G \uplus X_I \uplus X_C \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases}$$

$$= \begin{cases} \mathit{link}_2(p) & \text{if } p \in X_D \uplus P_D \uplus P_G \text{ and } \mathit{link}_2(p) \in E_G \\ \mathit{link}_C(y) & \text{if } p \in X_D \uplus P_D \uplus P_G \text{ and } \mathit{link}_2(p) = y \in Y_G \uplus X_I \\ \mathit{link}_C(\alpha(p)) & \text{if } p \in X'_C \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases}$$

$$= \begin{cases} \mathit{link}_G(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = y \in X_G \\ & \quad \text{and } \mathit{link}_G(y) \in E_G \\ \mathit{link}_G(p) & \text{if } p \in P_G \text{ and } \mathit{link}_G(p) \in E_G \\ \mathit{link}_C(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = z \in X_G \\ & \quad \text{and } \mathit{link}_G(z) = y \in Y_G \\ \mathit{link}_C(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = y \in X_I \\ \mathit{link}_C(y) & \text{if } p \in P_G \text{ and } \mathit{link}_G(p) = y \in Y_G \\ \mathit{link}_C(\alpha(p)) & \text{if } p \in X'_C \\ \mathit{link}_C(p) & \text{if } p \in P_C. \end{cases}$$

We can now verify ϕ^L is a link graph embedding and that ϕ is a bigraph embedding, i.e., they satisfy the conditions of Def. 5.1 and Def. 5.14:

ϕ^L:

- **(LGE-1)** Satisfied since Id_{V_G} is an identity map.
- **(LGE-2)** Satisfied since Id_{E_G} is an identity map.
- **(LGE-3)** Since $\mathit{link}_D : X_D \uplus P_D \to X_G \uplus X_I$, $X_D \subseteq X_H$, and $P_D \subseteq P_H$ we have $\phi^{\mathsf{i}} = \mathit{link}_D^{-1} \upharpoonright_{X_G} : X_G \to \mathcal{P}(X_H \uplus X_H)$; it is fully injective since link_D is a function.
- **(LGE-4)** Satisfied since $E_C \subseteq E_H$.
- **(LGE-5)** Satisfied since $\mathsf{rng}(\phi^{\mathsf{e}}) = \mathsf{rng}(\mathsf{Id}_{E_G}) = E_G$, $\mathsf{rng}(\phi^{\mathsf{o}}) = \mathsf{rng}(\mathit{link}_C \upharpoonright_{Y_G}) \subseteq E_C \uplus Y_H$, and $(E_c \uplus Y_H) \# E_G$.
- **(LGE-6)** Satisfied since $\mathsf{rng}(\phi^{\mathsf{i}}) = \mathsf{rng}(\mathit{link}_D^{-1} \upharpoonright_{X_G}) \subseteq X_D \uplus P_D$, $\mathsf{rng}(\phi^{\mathsf{port}}) \subseteq P_G$, and $X_D \uplus P_D \# P_G$.

(LGE-7) Satisfied since we have the following equalities:

$$\begin{aligned}
link_H^{-1} \circ \phi^{\mathsf{e}} &= ((\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ link_G \circ (\phi^{\mathsf{p}})^{-1})^{-1} \circ \phi^{\mathsf{e}} \\
&= \phi^{\mathsf{p}} \circ link_G^{-1} \circ (\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G})^{-1} \circ \phi^{\mathsf{e}} \\
&= \phi^{\mathsf{p}} \circ link_G^{-1} \restriction_{E_G} .
\end{aligned}$$

(LGE-8) Satisfied since $\phi^{\mathsf{v}} = \mathsf{id}_{V_G}$ and $ctrl_G \subseteq ctrl_H$.

(LGE-9) We check the condition separately for the ports and inner names:

$p \in P_G$: We check the condition for edges and outer names separately (noting $\phi^{\mathsf{p}}(p) = p$):

$link_G(p) \in E_G$:

$$\begin{aligned}
(\phi^{\mathsf{l}} \circ link_G)(p) &= (\phi^{\mathsf{e}} \circ link_H)(p) \\
&= (\mathsf{Id}_{E_G} \circ link_H)(p) \\
&= (link_H \circ \phi^{\mathsf{p}})(p).
\end{aligned}$$

$link_G(p) \in Y_G$:

$$\begin{aligned}
(\phi^{\mathsf{l}} \circ link_G)(p) &= (\phi^{\mathsf{o}} \circ link_G)(p) \\
&= (link_C \restriction_{Y_G} \circ link_G)(p) \\
&= (link_C \circ link_G \circ \phi^{\mathsf{p}})(p) \\
&= (link_H \circ \phi^{\mathsf{p}})(p).
\end{aligned}$$

$x \in X_G$: We check the condition for edges and outer names separately:

$link_G(x) \in E_G$:

$$\begin{aligned}
(\phi^{\mathsf{l}} \circ link_G)(x) &= (\phi^{\mathsf{e}} \circ link_G)(x) \\
&= (\mathsf{Id}_{E_G} \circ link_G)(x) \\
&= link_G(x) \\
&= link_H(link_D^{-1}(x)) \\
&= (link_H \circ \phi^{\mathsf{p}})(x).
\end{aligned}$$

$link_G(x) \in Y_G$:

$$\begin{aligned}
(\phi^{\mathsf{l}} \circ link_G)(x) &= (\phi^{\mathsf{o}} \circ link_G)(x) \\
&= (link_C \circ link_G)(x) \\
&= link_H(link_D^{-1}(x) \\
&= (link_H \circ \phi^{\mathsf{p}})(x).
\end{aligned}$$

ϕ: We have verified that ϕ is both a place and link graph embedding, so it only remains to show that these are consistent:

(BGE-1) Satisfied since $\mathsf{rng}(\phi^{\mathsf{i}}) = \mathsf{rng}(link_D^{-1} \restriction_{X_G}) \subseteq X_D \uplus P_D$, $X_D \subseteq X_H$, and $P_D = P_{H \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \cap V_H}$.

□

A.1.8 Proof of Prop. 5.19

$\Rightarrow$: Assume two matches $(\rho, \mathsf{id}_I, c, d), (\rho', \mathsf{id}_{I'}, c', d')$ in an agent a that are regarded the same, i.e.,

$$a = c \circ (\rho \centerdot R \otimes \mathsf{id}_{X_I}) \circ d = c' \circ (\rho \centerdot R \otimes \mathsf{id}_{X_{I'}}) \circ d'$$

$$\begin{array}{cccc} V_c = V_{c'} & E_c = E_{c'} & ctrl_c = ctrl_{c'} & prnt_c = prnt_{c'} \\ V_d = V_{d'} & E_d = E_{d'} = \emptyset & ctrl_d = ctrl_{d'} & prnt_d = prnt_{d'} \end{array}$$
$$link_c \restriction_{P_c \uplus Y_G} = link_{c'} \restriction_{P_c \uplus Y_G}$$

and there is a bijection $\alpha : X_{I'} \rightarrowtail X_I$ such that

$$link_c \circ \alpha = link_{c'} \restriction_{X_{I'}} \qquad link_d = \alpha \circ link_{d'} \, .$$

The matches are clearly decompositions

$$\begin{aligned} a &= c \circ ((\rho \centerdot R \otimes \mathsf{id}_{X_I}) \circ d \otimes \mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \circ (\mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \\ &= c' \circ ((\rho \centerdot R \otimes \mathsf{id}_{X_{I'}}) \circ d' \otimes \mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \circ (\mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \end{aligned}$$

and all but the following decomposition equivalence condition are trivially satisfied:

$$\begin{aligned} link_c \circ link_d \downharpoonright^{X_I} &= link_c \circ (\alpha \circ link_{d'}) \downharpoonright^{X_I} \\ &= link_c \circ \alpha \circ link_{d'} \downharpoonright^{X_{I'}} \\ &= link_{d'} \circ link_{d'} \downharpoonright^{X_{I'}} \, . \end{aligned}$$

$\Leftarrow$: Assume two decompositions of an agent a

$$\begin{aligned} a &= c \circ ((\rho \centerdot R \otimes \mathsf{id}_{X_I}) \circ d \otimes \mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \circ (\mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \\ &= c' \circ ((\rho \centerdot R \otimes \mathsf{id}_{X_{I'}}) \circ d' \otimes \mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \circ (\mathsf{id}_0 \otimes \mathsf{id}_\emptyset) \end{aligned}$$

where R is a redex and $d,'d$ are discrete and ground, and assume that they are equivalent, i.e.,

$$\begin{array}{cccc} V_c = V_{c'} & E_c = E_{c'} & ctrl_c = ctrl_{c'} & prnt_c = prnt_{c'} \\ V_d = V_{d'} & E_d = E_{d'} = \emptyset & ctrl_d = ctrl_{d'} & prnt_d = prnt_{d'} \\ link_c \restriction_{P_c \uplus Y_G} = link_{c'} \restriction_{P_c \uplus Y_G} & & link_C \circ link_D \downharpoonright^{X_I} = link_{C'} \circ link_{D'} \downharpoonright^{X_{I'}} \, . & \end{array}$$

Clearly, the decompositions are matches

$$a = c \circ (\rho \centerdot R \otimes \mathsf{id}_{X_I}) \circ d = c' \circ (\rho \centerdot R \otimes \mathsf{id}_{X_{I'}}) \circ d'$$

differing only by a bijection $\alpha : X_{I'} \rightarrowtail X_I$ defined by

$$\alpha = link_d \circ link_{d'}^{-1} \, .$$

$\square$

A.1.9 Proof of Prop. 5.21

We first show that $prmt(\phi)$ and $ctxt(\phi)$ are indeed bigraphs:

From Prop. 5.9 we have that $prmt(\phi)^P$ and $ctxt(\phi)^P$ are place graphs, so we just need to show that the following are link graphs:

$$\begin{aligned} prmt(\phi)^L &= (V_D, \emptyset, ctrl_D, link_D) : X_D \to X_G \uplus X_I \\ ctxt(\phi)^L &= (V_C, E_C, ctrl_C, link_C) : Y_G \uplus X_I \uplus X_C \to Y_H. \end{aligned}$$

In both cases the node sets and control maps are shared with the corresponding place graphs and are thus well-defined, and thus we just need to show that $link_D$ and $link_C$ are well-defined:

$link_D$: Since ϕ^{i} is fully injective $(\phi^{\mathsf{i}})^{-1}$ is a function and thus so is $link_D$.

What remains to show is $\mathsf{dom}(link_D) = X_D \uplus P_D$ and $\mathsf{cod}(link_D) = X_G \uplus X_I$. Since, by definition, we have $\mathsf{dom}(link'_D) = P'_D \subseteq P_D$ and $\mathsf{cod}(link'_D) = X_I$, this amounts to showing $\mathsf{dom}((\phi^{\mathsf{i}})^{-1}) = X_D \uplus (P_D \setminus P'_D)$ and $\mathsf{rng}((\phi^{\mathsf{i}})^{-1}) \subseteq X_G \uplus X_I$. The latter is immediate since $\mathsf{dom}(\phi^{\mathsf{i}}) = X_G$. To see $\mathsf{dom}((\phi^{\mathsf{i}})^{-1}) = \mathsf{rng}(\phi^{\mathsf{i}}) = X_D \uplus (P_D \setminus P'_D)$, we expand the definitions of X_D and P'_D:

$$\begin{aligned} X_D \uplus (P_D \setminus P'_D) &= (\mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H) \uplus (P_D \setminus (P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}))) \\ &= (\mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H) \uplus (P_D \cap \mathsf{rng}(\phi^{\mathsf{i}})) \\ &= \mathsf{rng}(\phi^{\mathsf{i}}) \cap (X_H \uplus P_D). \end{aligned}$$

Thus $\mathsf{dom}((\phi^{\mathsf{i}})^{-1}) = X_D \uplus (P_D \setminus P'_D)$ if $\mathsf{rng}(\phi^{\mathsf{i}}) \subseteq X_H \uplus P_D$. But this is exactly the consistency condition (BGE-1) and must thus be satisfied since ϕ is an embedding.

$link_C$: The constituent functions have the following domains and codomains:

$$\begin{aligned} \phi^{\mathsf{o}} &: Y_G \to E_H \uplus Y_H \\ link_H &: X_H \uplus P_H \to E_H \uplus Y_H \\ \mathsf{Id}_{P_C} &: P_C \to P_C \\ link'^{-1}_D &: X_I \to P'_D \\ \alpha_C &: X_C \to X'_C \end{aligned}$$

Since $link'_D$ is a bijection, $link'^{-1}_D : X_I \to P'_D$ is a function. By construction we have $X_I \# X_C$ and clearly $P_C \# X_I \uplus X_C$, so $\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C : P_C \uplus X_I \uplus X_C \to P_C \uplus P'_D \uplus X'_C$ is a function. By construction $X'_C \subseteq X_H$ and as shown in the proof of Theorem 5.9 we have $V_C, V_D \subseteq V_H$ and by definition $ctrl_C = ctrl_H \restriction_{V_C}$ and $ctrl_D = ctrl_H \restriction_{V_D}$, so $P_C \subseteq P_H$ and $P'_D \subseteq P_D \subseteq P_H$, and thus $link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)$ is a function. Also by construction we have $Y_G \# X_I$, $Y_G \# X_C$ and clearly $P_C \# Y_G$, so $link_C$ is a function.

What remains to show is $\mathsf{dom}(link_C) = Y_G \uplus X_I \uplus X_C \uplus P_C$ and $\mathsf{cod}(link_C) = E_C \uplus Y_H$. The first is immediate from the domains of the constituent functions. The latter amounts to showing $\mathsf{cod}(\phi^{\mathsf{o}}) = \mathsf{cod}(link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)) = E_C \uplus Y_H$. Noting $\mathsf{cod}(\phi^{\mathsf{o}}) : E_H \uplus Y_H$, $\mathsf{cod}(link_H) : E_H \uplus Y_H$, $\mathsf{cod}(link_H)$, and $E_C \uplus Y_H = (E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}})) \uplus Y_H$, we just have to show $\mathsf{rng}(\phi^{\mathsf{o}}) \# \mathsf{rng}(\phi^{\mathsf{e}})$ and $\mathsf{rng}(link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)) \# \mathsf{rng}(\phi^{\mathsf{e}})$:

$\mathsf{rng}(\phi^{\mathsf{o}}) \# \mathsf{rng}(\phi^{\mathsf{e}})$**:** This is the first injectivity condition for link graph embeddings, condition (LGE-5), and is thus assumed to be satisfied.

$\mathsf{rng}(link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)) \# \mathsf{rng}(\phi^{\mathsf{e}})$**:** By the surjectivity condition (LGE-7), a point $p \in link_H^{-1}(e)$ of an edge $e \in \mathsf{rng}(\phi^{\mathsf{e}})$ in the image of edges must be in the image of points, i.e.,

$$\begin{aligned} p &\in \mathsf{rng}(\phi^{\mathsf{p}}) \\ &= \mathsf{rng}(\phi^{\mathsf{port}}) \uplus \mathsf{rng}(\phi^{\mathsf{i}}) \\ &\subseteq \mathsf{rng}(\phi^{\mathsf{port}}) \uplus X_H \uplus P_D. \end{aligned}$$

where the last inclusion follows from the consistency condition (BGE-1). Note that $\mathsf{rng}(\phi^{\mathsf{port}}) \# P_D$ and $\mathsf{rng}(\phi^{\mathsf{port}}) \# X_H$.

But the images of the three functions Id_{P_C}, link'^{-1}_D, α_C are not in the image of points, which can be seen as follows:

P_C: Immediate from the construction of P_C:

$$P_C = P_H \setminus \mathsf{rng}(\phi^{\mathsf{port}}) \setminus P_D.$$

P'_D: Unfolding the construction of P'_D it becomes immediate:

$$P'_D = P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}).$$

X'_C: Unfolding the construction of X'_C it becomes immediate:

$$\begin{aligned} X'_C &= X_H \setminus X_D \\ &= X_H \setminus (\mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H) \\ &= X_H \setminus \mathsf{rng}(\phi^{\mathsf{i}}). \end{aligned}$$

We now turn to the matter of showing that the construction is defined up to decomposition equivalence. Since the place graph decomposition is defined up to decomposition equivalence, it suffices to show that any valid choices of X_I, link'_D, X_C, and α_C yield equivalent decompositions cf. Def. 5.18.

Assume that we have alternative choices:

$$\begin{aligned} X_{I'} &: \text{ a set of names satisfying} \\ &\qquad |X_{I'}| = |P'_D|,\ X_{I'} \# X_G, \text{ and } X_{I'} \# Y_G \\ \mathit{link}'_{D'} &: P'_D \rightarrowtail X_{I'} \text{ a bijection} \\ X_{C'} &: \text{ a set of names satisfying} \\ &\qquad |X_{C'}| = |X'_C|,\ X_{C'} \# Y_G, \text{ and } X_{C'} \# X_{I'} \\ \alpha_{C'} &: X_{C'} \rightarrowtail X'_C \text{ a bijection} \end{aligned}$$

and construct the corresponding link maps

$$\begin{aligned} \mathit{link}_{D'} &= (\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_{D'} \\ \mathit{link}_{C'} &= \phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_{D'} \uplus \alpha_{C'}). \end{aligned}$$

Finally, we check the equivalence conditions:

$$\begin{aligned}
\mathit{link}_D \restriction^{X_G} &= ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_D) \restriction^{X_G} \\
&= (\phi^{\mathsf{i}})^{-1} \restriction^{X_G} \\
&= ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_{D'}) \restriction^{X_G} \\
&= \mathit{link}_{D'} \restriction^{X_G} \\
\mathit{link}_C \restriction_{P_C \uplus Y_G} &= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C)) \restriction_{P_C \uplus Y_G} \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ \mathsf{Id}_{P_C}) \restriction_{P_C \uplus Y_G} \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_{C'})) \restriction_{P_C \uplus Y_G} \\
&= \mathit{link}_{C'} \restriction_{P_C \uplus Y_G} \\
\mathit{link}_C \circ \alpha_C^{-1} &= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C)) \circ \alpha_C^{-1} \\
&= \mathit{link}_H \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_{C'})) \circ \alpha_{C'}^{-1} \\
&= \mathit{link}_{C'} \circ \alpha_{C'}^{-1} \\
\mathit{link}_C \circ \mathit{link}_D \restriction^{X_I} &= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C)) \circ ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_D) \restriction^{X_I} \\
&= \mathit{link}_H \restriction_{P'_D} \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_{D'} \uplus \alpha_{C'})) \circ ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_{D'}) \restriction^{X_I} \\
&= \mathit{link}_{C'} \circ \mathit{link}_{D'} \restriction^{X_{I'}} .
\end{aligned}$$

We now show that $\mathit{prmt}(\phi)$ and $\mathit{ctxt}(\phi)$ are indeed parameter and context for the embedding of G:
Let

$$\begin{aligned}
D : \langle k_D, X_D \rangle \to \langle k_G, X_G \uplus X_I \rangle &= \mathit{prmt}(\phi), \\
C : \langle k_C, Y_G \uplus X_I \uplus X_C \rangle \to \langle m_H, Y_H \rangle &= \mathit{ctxt}(\phi), \mathit{and} \\
(V, E, \mathit{ctrl}, \mathit{prnt}, \mathit{link}) &= \mathit{ctxt}(\phi) \\
&\quad \circ ((\phi \bullet G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{\langle |\tilde{k}_C|, X_C \rangle}) \\
&\quad \circ (\pi \otimes \mathsf{id}_{X_H}).
\end{aligned}$$

As composition and tensor product of bigraphs are defined pointwise on the constituent place and link graphs (cf. Def. 2.6 and Def. 2.8), it is straightforward to see that the proof of Theorem 5.9 is also a proof of $V_H = V, \mathit{ctrl}_H = \mathit{ctrl}, \mathit{prnt}_H = \mathit{prnt}$, since this theorem only adds link graph structure.

By the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8) we have the following equalities:

$$\begin{aligned}
E &= E_C \uplus \phi^{\mathsf{e}}(E_G) \uplus E_D && \text{Defs. 2.6 and 2.8} \\
&= (E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}})) \uplus \phi^{\mathsf{e}}(E_G) \uplus \emptyset && \text{Def. 5.20} \\
&= E_H .
\end{aligned}$$

To prove $\mathit{link}_H = \mathit{link}$, we construct the link map link incrementally according to the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8), and then verify $\mathit{link}_H(p) = \mathit{link}(p)$:

$\phi \bullet G$: We write $\mathit{link}_{\phi \bullet G}$ for the link map of the resulting bigraph:

$$\mathit{link}_{\phi \bullet G} = (\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ \mathit{link}_G \circ ((\phi^{\mathsf{p}})^{-1} \uplus \mathsf{Id}_{X_G}).$$

where

$$\phi^{\mathsf{p}}(v,i) = (\phi^{\mathsf{v}}(v), i).$$

Also, we write $P_{\phi \bullet G}$ for the ports of that bigraph: $P_{\phi \bullet G} = \phi^{\mathsf{p}}(P_G)$.

$\phi \bullet G \otimes \mathsf{id}_{X_I}$: We write link_1 for the link map of the resulting bigraph:

$$\mathit{link}_1 = \mathit{link}_{\phi \bullet G} \uplus \mathsf{Id}_{X_I}.$$

$(\phi \bullet G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi)$: We write link_2 for the link map of the resulting bigraph:

$$\begin{aligned}\mathit{link}_2(p) &= \begin{cases}\mathit{link}_D(p) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) \in \emptyset \\ \mathit{link}_1(x) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = x \in X_G \uplus X_I \\ \mathit{link}_1(p) & \text{if } p \in P_{\phi \bullet G}\end{cases} \\ &= \begin{cases}\mathit{link}_1(x) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = x \in X_G \uplus X_I \\ \mathit{link}_1(p) & \text{if } p \in P_{\phi \bullet G}\end{cases}.\end{aligned}$$

$(\phi \bullet G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|} \otimes \alpha_C^{-1}$: We write link_3 for the link map of the resulting bigraph:

$$\mathit{link}_3 = \mathit{link}_2 \uplus \alpha_C^{-1}.$$

$\pi \otimes \mathsf{id}_{X_H}$: We write link_4 for the link map of the resulting bigraph:

$$\mathit{link}_4 = \mathsf{Id}_{\emptyset} \uplus \mathsf{Id}_{X_H} = \mathsf{Id}_{X_H}.$$

$((\phi \bullet G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|} \otimes \alpha_C^{-1}) \circ (\pi \otimes \mathsf{id}_{X_H})$: We write link_5 for the link map of the resulting bigraph:

$$\begin{aligned}\mathit{link}_5(p) &= \begin{cases}\mathit{link}_4(p) & \text{if } p \in X_H \text{ and } \mathit{link}_4(p) \in \emptyset \\ \mathit{link}_3(x) & \text{if } p \in X_H \text{ and } \mathit{link}_4(p) = x \in X_H \\ \mathit{link}_3(p) & \text{if } p \in P_{\phi \bullet G} \uplus P_D\end{cases} \\ &= \begin{cases}\mathit{link}_3(x) & \text{if } p \in X_H \text{ and } \mathsf{Id}_{X_H}(p) = x \in X_H \\ \mathit{link}_3(p) & \text{if } p \in P_{\phi \bullet G} \uplus P_D\end{cases} \\ &= \begin{cases}\mathit{link}_3(p) & \text{if } p \in X_H \\ \mathit{link}_3(p) & \text{if } p \in P_{\phi \bullet G} \uplus P_D\end{cases} \\ &= \mathit{link}_3(p).\end{aligned}$$

$\mathit{ctxt}(\phi) \circ ((\phi \bullet G \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|} \otimes \alpha_C^{-1}) \circ (\pi \otimes \mathsf{id}_{X_H})$: Finally, we have

$$\mathit{link}(p) = \begin{cases}\mathit{link}_3(p) & \text{if } p \in X_H \uplus P_{\phi \bullet G} \uplus P_D \text{ and } \mathit{link}_3(p) \in \phi^{\mathsf{e}}(E_G) \\ \mathit{link}_C(x) & \text{if } p \in X_H \uplus P_{\phi \bullet G} \uplus P_D \text{ and } \mathit{link}_3(p) = x \in Y_G \uplus X_I \uplus X_C \\ \mathit{link}_C(p) & \text{if } p \in P_C\end{cases}$$

Unfolding the definitions of the involved link maps, we get the following equalities:

$$
\begin{aligned}
\mathit{link}(p) &= \begin{cases}
\alpha_C^{-1}(p) & \text{if } p \in X'_C \text{ and } \alpha_C^{-1}(p) \in \phi^{\mathsf{e}}(E_G)\\
\mathit{link}_2(p) & \text{if } p \in X_D \uplus P_{\phi\blacksquare G} \uplus P_D \text{ and } \mathit{link}_2(p) \in \phi^{\mathsf{e}}(E_G)\\
\mathit{link}_C(y) & \text{if } p \in X_D \uplus P_{\phi\blacksquare G} \uplus P_D \text{ and } \mathit{link}_2(p) = y \in Y_G \uplus X_I\\
\mathit{link}_C(y) & \text{if } p \in X'_C \text{ and } \alpha_C^{-1}(p) = y \in X_C\\
\mathit{link}_C(p) & \text{if } p \in P_C
\end{cases}\\
&= \begin{cases}
\mathit{link}_1(x) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_1(x) \in \phi^{\mathsf{e}}(E_G) \text{ and } \mathit{link}_D(p) = x \in X_G \uplus X_I\\
\mathit{link}_1(p) & \text{if } p \in P_{\phi\blacksquare G} \text{ and } \mathit{link}_1(p) \in \phi^{\mathsf{e}}(E_G)\\
\mathit{link}_C(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_1(x) = y \in Y_G \uplus X_I \text{ and } \mathit{link}_D(p) = x \in X_G \uplus X_I\\
\mathit{link}_C(y) & \text{if } p \in P_{\phi\blacksquare G} \text{ and } \mathit{link}_1(p) = y \in Y_G\\
\mathit{link}_C(y) & \text{if } p \in X'_C \text{ and } \alpha_C^{-1}(p) = y \in X_C\\
\mathit{link}_C(p) & \text{if } p \in P_C
\end{cases}\\
&= \begin{cases}
\mathit{link}_{\phi\blacksquare G}(x) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_{\phi\blacksquare G}(x) \in \phi^{\mathsf{e}}(E_G) \text{ and } \mathit{link}_D(p) = x \in X_G\\
\mathit{link}_{\phi\blacksquare G}(p) & \text{if } p \in P_{\phi\blacksquare G} \text{ and } \mathit{link}_{\phi\blacksquare G}(p) \in \phi^{\mathsf{e}}(E_G)\\
\mathit{link}_C(x) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_D(p) = x \in X_I\\
\mathit{link}_C(y) & \text{if } p \in X_D \uplus P_D \text{ and } \mathit{link}_{\phi\blacksquare G}(x) = y \in Y_G \text{ and } \mathit{link}_D(p) = x \in X_G\\
\mathit{link}_C(y) & \text{if } p \in P_{\phi\blacksquare G} \text{ and } \mathit{link}_{\phi\blacksquare G}(p) = y \in Y_G\\
\mathit{link}_C(y) & \text{if } p \in X'_C \text{ and } \alpha_C^{-1}(p) = y \in X_C\\
\mathit{link}_C(p) & \text{if } p \in P_C
\end{cases}.
\end{aligned}
$$

Let us now verify $\mathit{link} = \mathit{link}_H$. link is defined for $X_D \uplus X'_C \uplus P_C \uplus P_{\phi\blacksquare G} \uplus P_D$ and so is link_H since $X_D \uplus X'_C = X_H$ and $V_C \uplus \phi^{\mathsf{v}}(V_G) \uplus V_D = V_H$, $\mathit{ctrl}_H \restriction_{V_C} \uplus\, \mathit{ctrl}_G \circ (\phi^{\mathsf{v}})^{-1} \uplus \mathit{ctrl}_H \restriction_{V_D} = \mathit{ctrl}_H$ which implies $P_C \uplus P_{\phi\blacksquare G} \uplus P_D = P_H$. Let us examine each case of $\mathit{link}(p)$ separately:

$p \in X_D \uplus P_D$ **and** $\mathit{link}_{\phi\blacksquare G}(x) \in \phi^{\mathsf{e}}(E_G)$ **and** $\mathit{link}_D(p) = x \in X_G$**:**

We have

$$
\begin{aligned}
\mathit{link}(p) &= \mathit{link}_{\phi\blacksquare G}(\mathit{link}_D(p))\\
&= ((\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ \mathit{link}_G \circ ((\phi^{\mathsf{p}})^{-1} \uplus \mathsf{Id}_{X_G}))(\mathit{link}_D(p))\\
&= (\phi^{\mathsf{e}} \circ \mathit{link}_G)(\mathit{link}_D(p))\\
&= (\phi^{\mathsf{e}} \circ \mathit{link}_G)(((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_D)(p))\\
&= (\phi^{\mathsf{e}} \circ \mathit{link}_G)((\phi^{\mathsf{i}})^{-1}(p))\\
&= (\mathit{link}_H \circ \phi^{\mathsf{p}'})((\phi^{\mathsf{i}})^{-1}(p))\\
&= \mathit{link}_H(p)
\end{aligned}
$$

where

$$
\phi^{\mathsf{p}'}(p) = \begin{cases}
(\phi^{\mathsf{v}}(v), i) & \text{if } p = (v, i) \in P_G\\
\phi^{\mathsf{i}}(p) & \text{if } p \in X_G
\end{cases}.
$$

$p \in P_{\phi\blacksquare G}$ **and** $\mathit{link}_{\phi\blacksquare G}(p) \in \phi^{\mathsf{e}}(E_G)$**:**

We have

$$\begin{aligned} link(p) &= link_{\phi \blacksquare G}(p) \\ &= ((\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ link_G \circ ((\phi^{\mathsf{p}})^{-1} \uplus \mathsf{Id}_{X_G}))(p) \\ &= (\phi^{\mathsf{e}} \circ link_G \circ (\phi^{\mathsf{p}})^{-1})(p) \\ &= (link_H \circ \phi^{\mathsf{p}'} \circ (\phi^{\mathsf{p}})^{-1})(p) \\ &= link_H(p) \end{aligned}$$

where

$$\phi^{\mathsf{p}'}(p) = \begin{cases} (\phi^{\mathsf{v}}(v), i) & \text{if } p = (v,i) \in P_G \\ \phi^{\mathsf{i}}(p) & \text{if } p \in X_G \end{cases}.$$

$p \in X_D \uplus P_D$ **and** $link_D(p) = x \in X_I$**:** We have

$$\begin{aligned} link(p) &= link_C(link_D(p)) \\ &= link_C(((\phi^{\mathsf{i}})^{-1} \uplus link'_D)(p)) \\ &= link_C(link'_D(p)) \\ &= (\phi^{\mathsf{o}} \uplus link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C))(link'_D(p)) \\ &= (link_H \circ link'^{-1}_D)(link'_D(p)) \\ &= link_H(p). \end{aligned}$$

$p \in X_D \uplus P_D$ **and** $link_{\phi \blacksquare G}(x) = y \in Y_G$ **and** $link_D(p) = x \in X_G$**:**

We have

$$\begin{aligned} link(p) &= link_C(link_{\phi \blacksquare G}(link_D(p))) \\ &= link_C(link_{\phi \blacksquare G}(((\phi^{\mathsf{i}})^{-1} \uplus link'_D)(p))) \\ &= link_C(link_{\phi \blacksquare G}((\phi^{\mathsf{i}})^{-1}(p))) \\ &= link_C(((\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ link_G \circ ((\phi^{\mathsf{p}})^{-1} \uplus \mathsf{Id}_{X_G}))((\phi^{\mathsf{i}})^{-1}(p))) \\ &= link_C(link_G((\phi^{\mathsf{i}})^{-1}(p))) \\ &= (\phi^{\mathsf{o}} \uplus link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C))(link_G((\phi^{\mathsf{i}})^{-1}(p))) \\ &= \phi^{\mathsf{o}}(link_G((\phi^{\mathsf{i}})^{-1}(p))) \\ &= (\phi^{\mathsf{o}} \circ link_G)((\phi^{\mathsf{i}})^{-1}(p)) \\ &= (link_H \circ \phi^{\mathsf{p}'})((\phi^{\mathsf{i}})^{-1}(p)) \\ &= link_H(p) \end{aligned}$$

where

$$\phi^{\mathsf{p}'}(p) = \begin{cases} (\phi^{\mathsf{v}}(v), i) & \text{if } p = (v,i) \in P_G \\ \phi^{\mathsf{i}}(p) & \text{if } p \in X_G \end{cases}.$$

$p \in P_{\phi \blacksquare G}$ **and** $link_{\phi \blacksquare G}(p) = x \in Y_G$**:**

We have

$$\begin{aligned}
\mathit{link}(p) &= \mathit{link}_C(\mathit{link}_{\phi\blacksquare G}(p)) \\
&= \mathit{link}_C(((\phi^{\mathsf{e}} \uplus \mathsf{Id}_{Y_G}) \circ \mathit{link}_G \circ ((\phi^{\mathsf{p}})^{-1} \uplus \mathsf{Id}_{X_G}))(p)) \\
&= \mathit{link}_C((\mathit{link}_G \circ (\phi^{\mathsf{p}})^{-1})(p)) \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C))((\mathit{link}_G \circ (\phi^{\mathsf{p}})^{-1})(p)) \\
&= \phi^{\mathsf{o}}((\mathit{link}_G \circ (\phi^{\mathsf{p}})^{-1})(p)) \\
&= (\phi^{\mathsf{o}} \circ \mathit{link}_G \circ (\phi^{\mathsf{p}})^{-1})(p) \\
&= (\mathit{link}_H \circ \phi^{\mathsf{p}'} \circ (\phi^{\mathsf{p}})^{-1})(p) \\
&= \mathit{link}_H(p)
\end{aligned}$$

where

$$\phi^{\mathsf{p}'}(p) = \begin{cases} (\phi^{\mathsf{v}}(v), i) & \text{if } p = (v,i) \in P_G \\ \phi^{\mathsf{i}}(p) & \text{if } p \in X_G \end{cases}.$$

$p \in X'_C$ and $\alpha_C^{-1}(p) = y \in X_C$:

We have

$$\begin{aligned}
\mathit{link}(p) &= \mathit{link}_C(\alpha_C^{-1}(p)) \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C))(\alpha_C^{-1}(p)) \\
&= (\mathit{link}_H \circ \alpha_C)(\alpha_C^{-1}(p)) \\
&= \mathit{link}_H(p).
\end{aligned}$$

$p \in P_C$:

We have

$$\begin{aligned}
\mathit{link}(p) &= \mathit{link}_C(p) \\
&= (\phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_D \uplus \alpha_C))(p) \\
&= \mathit{link}_H(p).
\end{aligned}$$

□

A.1.10 Proof of Theorem 5.22

Def. 5.20 ∘ Def. 5.16 = Id: Assume

$$H = C \circ ((G \otimes \mathsf{id}_{X_I}) \circ D \otimes \mathsf{id}_k \otimes \alpha) \circ (\pi \otimes \mathsf{id}_{X_H})$$

$$\begin{array}{lll}
\phi^{\mathsf{v}} = \mathsf{Id}_{V_G} & \phi^{\mathsf{r}} = \mathit{prnt}_C \restriction_{m_G} & \phi^{\mathsf{s}} = (\mathsf{Id}_{V_D} \uplus \pi^{-1}) \circ \mathit{prnt}_D^{-1} \restriction_{k_G} \\
\phi^{\mathsf{e}} = \mathsf{Id}_{E_G} & \phi^{\mathsf{o}} = \mathit{link}_C \restriction_{Y_G} & \phi^{\mathsf{i}} = \mathit{link}_D^{-1} \restriction_{X_G}
\end{array}$$

where D is semi-discrete on X_G. Also, assume the results from the proof of Prop. 5.17.

It is clear that the place graph may be expressed as

$$H^P = C^P \circ (G^P \circ D^P \otimes \mathsf{id}_k) \circ \pi$$

and thus Theorem 5.10 applies, so using construction Def. 5.20 we obtain

$$\begin{aligned} prmt(\phi) &\overset{\text{def}}{=} (V_D, \emptyset, ctrl_D, prnt_D, link_{D'}) : \langle k_D, X_{D'}\rangle \to \langle k_G, X_G \uplus X_{I'}\rangle \\ ctxt(\phi) &\overset{\text{def}}{=} (V_C, E_{C'}, ctrl_C, prnt_C, link_{C'}) : \langle k_C, Y_G \uplus X_{I'} \uplus X_{C'}\rangle \to \langle m_H, Y_H\rangle \end{aligned}$$

$$\begin{aligned} P'_{D'} &= P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}) \\ X_{D'} &= \mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H \\ X_{I'} &: \text{ a set of names satisfying} \\ &\quad |X_{I'}| = |P'_{D'}|,\ X_{I'} \mathbin{\#} X_G, \text{ and } X_{I'} \mathbin{\#} Y_G \\ link'_{D'} &: P'_{D'} \rightarrowtail X_{I'} \text{ a bijection} \\ link_{D'} &= (\phi^{\mathsf{i}})^{-1} \uplus link'_{D'} \end{aligned}$$

$$\begin{aligned} E_{C'} &= E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}}) \\ X'_{C'} &= X_H \setminus X_{D'} \\ X_{C'} &: \text{ a set of names satisfying} \\ &\quad |X_{C'}| = |X'_{C'}|,\ X_{C'} \mathbin{\#} Y_G, \text{ and } X_{C'} \mathbin{\#} X_{I'} \\ \alpha_{C'} &: X_{C'} \rightarrowtail X'_{C'} \text{ a bijection} \\ link_{C'} &= \phi^{\mathsf{o}} \uplus link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_{D'} \uplus \alpha_{C'}) \end{aligned}$$

$$H = ctxt(\phi) \circ ((\phi \centerdot G \otimes \mathsf{id}_{X_{I'}}) \circ prmt(\phi) \otimes \mathsf{id}_k \otimes \alpha^{-1}_{C'}) \circ (\pi \otimes \mathsf{id}_{X_H}).$$

Thus it suffices to show $D^L = prmt(\phi)^L$, $C^L = ctxt(\phi)^L$, $X_I = X_{I'}$, and $\alpha = \alpha^{-1}_{C'}$.

Let us first unfold some of the definitions (noting that $X_D \subseteq X_H$ and that D is semi-discrete on X_G):

$$\begin{aligned} P'_D &= P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}) \\ &= P_D \setminus \mathsf{rng}(link_D^{-1} \restriction_{X_G}) \\ &= P_D \setminus link_D^{-1}(X_G) \\ X_{D'} &= \mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H \\ &= \mathsf{rng}(link_D^{-1} \restriction_{X_G}) \cap X_H \\ &= \mathsf{rng}(link_D^{-1} \restriction_{X_G}) \cap X_H \\ &= X_D \\ E_{C'} &= E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}}) \\ &= E_H \setminus E_G \\ &= E_C \\ X'_{C'} &= X_H \setminus X_{D'} \\ &= X_H \setminus X_D \\ &= X'_C. \end{aligned}$$

With these in mind, we proceed to prove $D^L = prmt(\phi)^L$, $C^L = ctxt(\phi)^L$, $X_I = X_{I'}$, $X_C = X_{C'}$, and $\alpha = \alpha^{-1}_{C'}$:

$X_I = X_{I'}$: Remember that we are free to choose $X_{I'}$ as it is internal to the decomposition, as long as it satisfies $|X_{I'}| = |P_D|$, $X_{I'} \mathbin{\#} X_G$, and $X_{I'} \mathbin{\#} Y_G$. From the initial decomposition we know that X_I satisfies these conditions, and thus we simply choose $X_{I'} = X_I$.

Similarly, we are free to choose a suitable bijection $\mathit{link}'_{D'} : P'_D \rightarrowtail X_I$, so we simply choose $\mathit{link}'_{D'} = \mathit{link}_D \restriction_{P'_D}$.

$X_C = X_{C'}$: Again, we are free to choose $X_{C'}$ as it is internal to the decomposition, as long as it satisfies $|X_{C'}| = |X'_{C'}|$, $X_{C'} \# Y_G$, and $X_{C'} \# X_{I'}$. From the initial decomposition we know that X_C satisfies these conditions, and thus we simply choose $X_{C'} = X_C$.

$\alpha = \alpha_{C'}^{-1}$: Again, we are free to choose $\alpha_{C'}$ as it is internal to the decomposition. So we simply choose $\alpha_{C'} = \alpha^{-1}$.

$D^L = \mathit{prmt}(\phi)^L$: Since $D^P = \mathit{prmt}(\phi)^P$ it suffices to show $E_D = \emptyset$, and $\mathit{link}_D = \mathit{link}_{D'}$:

$E_D = \emptyset$: Satisfied since D is semi-discrete.

$\mathit{link}_D = \mathit{link}_{D'}$: Easily seen by expanding the definitions:

$$\begin{aligned}
\mathit{link}_{D'} &= (\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_{D'} \\
&= (\mathit{link}_D^{-1} \restriction_{X_G})^{-1} \uplus \mathit{link}_D \restriction_{P'_D} \\
&= \mathit{link}_D \downharpoonleft^{X_G} \uplus \mathit{link}_D \restriction_{P_D \setminus \mathit{link}_D^{-1}(X_G)} \\
&= \mathit{link}_D\,.
\end{aligned}$$

$C^L = \mathit{ctxt}(\phi)^L$: Since $C^P = \mathit{ctxt}(\phi)^P$ and $E_{C'} = E_C$ it suffices to show $\mathit{link}_C = \mathit{link}_{C'}$ which is easily seen by expanding the definitions:

$$\begin{aligned}
\mathit{link}_{C'} &= \phi^{\mathsf{o}} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus \mathit{link}'^{-1}_{D'} \uplus \alpha_{C'}) \\
&= \mathit{link}_C \restriction_{Y_G} \uplus \mathit{link}_H \circ (\mathsf{Id}_{P_C} \uplus (\mathit{link}_D \restriction_{P'_D})^{-1} \uplus \alpha^{-1}) \\
&= \mathit{link}_C \restriction_{Y_G} \uplus \mathit{link}_H \restriction_{P_C} \uplus \mathit{link}_H \circ (\mathit{link}_D \restriction_{P_D \setminus \mathit{link}_D^{-1}(X_G)})^{-1} \uplus \mathit{link}_H \circ \alpha^{-1}) \\
&= \mathit{link}_C \restriction_{Y_G} \uplus \mathit{link}_C \restriction_{P_C} \uplus \mathit{link}_H \circ (\mathit{link}_D \downharpoonleft^{X_I})^{-1} \uplus \mathit{link}_C \restriction_{X_C} \\
&= \mathit{link}_C \restriction_{Y_G} \uplus \mathit{link}_C \restriction_{P_C} \uplus \mathit{link}_C \restriction_{X_I} \uplus \mathit{link}_C \restriction_{X_C} \\
&= \mathit{link}_C\,.
\end{aligned}$$

Def. 5.16 $\circ$ Def. 5.20 $=$ Id: Assume a bigraph $G : \langle k_G, X_G \rangle \to \langle m_G, Y_G \rangle$ and an embedding $\phi : G \hookrightarrow H$

into a bigraph $H : \langle k_H, X_H \rangle \to \langle m_H, Y_H \rangle$ (for simplicity, assume $\phi \mathbin{\blacksquare} G = G$),

$$\begin{aligned}
prmt(\phi) &\overset{\text{def}}{=} (V_D, \emptyset, ctrl_D, prnt_D, link_D) : \langle k_D, X_D \rangle \to \langle k_G, X_G \uplus X_I \rangle \\
ctxt(\phi) &\overset{\text{def}}{=} (V_C, E_C, ctrl_C, prnt_C, link_C) : \langle k_C, Y_G \uplus X_I \uplus X_C \rangle \to \langle m_H, Y_H \rangle \\
(V_D, ctrl_D, prnt_D) : k_D \to k_G &= prmt(\phi^P) \\
P'_D &= P_D \setminus \mathsf{rng}(\phi^{\mathsf{i}}) \\
X_D &= \mathsf{rng}(\phi^{\mathsf{i}}) \cap X_H \\
X_I &: \text{ a set of names satisfying} \\
&\qquad |X_I| = |P'_D|,\ X_I \# X_G, \text{ and } X_I \# Y_G \\
link'_D &: P'_D \rightarrowtail X_I \text{ a bijection} \\
link_D &= (\phi^{\mathsf{i}})^{-1} \uplus link'_D
\end{aligned}$$

$$\begin{aligned}
(V_C, ctrl_C, prnt_C) : k_C \to m_H &= ctxt(\phi^P) \\
E_C &= E_H \setminus \mathsf{rng}(\phi^{\mathsf{e}}) \\
X'_C &= X_H \setminus X_D \\
X_C &: \text{ a set of names satisfying} \\
&\qquad |X_C| = |X'_C|,\ X_C \# Y_G, \text{ and } X_C \# X_I \\
\alpha_C &: X_C \rightarrowtail X'_C \text{ a bijection} \\
link_C &= \phi^{\mathsf{o}} \uplus link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)
\end{aligned}$$

$$\begin{aligned}
H &= ctxt(\phi) \circ ((G \otimes \mathsf{id}_{X_I}) \circ prmt(\phi) \otimes \mathsf{id}_{|\tilde{k}_C|} \otimes \alpha_C^{-1}) \circ (\pi \otimes \mathsf{id}_{X_H}) \\
\pi &= f_D^{-1} \uplus f'^{-1} : k_H \to k_H \\
f'(i + k_D) &= f'_C(i) \quad \text{for } i \in |\tilde{k}_C|.
\end{aligned}$$

where we assume that the name sets and bijections are chosen as in the previous proof case, i.e.,

$$X_{I'} = X_I \qquad X_{C'} = X_C \qquad link'_{D'} = link_D \restriction_{P'_D} \qquad \alpha_{C'} = \alpha^{-1}.$$

It is clear that the place graph may be expressed as

$$H^P = ctxt(\phi)^P \circ (G^P \circ prmt(\phi)^P \otimes \mathsf{id}_{|\tilde{k}_C|}) \circ \pi$$

and thus Theorem 5.10 applies, so using construction Def. 5.16 we obtain

$$\phi' = \phi^{\mathsf{v}} \uplus \phi'^{\mathsf{e}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}} \uplus \phi'^{\mathsf{i}} \uplus \phi'^{\mathsf{o}} : G \hookrightarrow H$$

$$\phi'^{\mathsf{e}} = \mathsf{Id}_{E_G} \qquad \phi'^{\mathsf{o}} = link_C \restriction_{Y_G} \qquad \phi'^{\mathsf{i}} = link_D^{-1} \restriction_{X_G}.$$

We must prove $\phi = \phi'$, and it suffices to show $\phi^{\mathsf{e}} = \phi'^{\mathsf{e}}$, $\phi^{\mathsf{o}} = \phi'^{\mathsf{o}}$, and $\phi^{\mathsf{i}} = \phi'^{\mathsf{i}}$.

$\phi^{\mathsf{e}} = \phi'^{\mathsf{e}}$: Satisfied by assumption.

$\phi^{\mathsf{o}} = \phi'^{\mathsf{o}}$: Easily seen by unfolding the definitions:

$$\begin{aligned}
\phi'^{\mathsf{o}} &= link_C \restriction_{Y_G} \\
&= (\phi^{\mathsf{o}} \uplus link_H \circ (\mathsf{Id}_{P_C} \uplus link'^{-1}_D \uplus \alpha_C)) \restriction_{Y_G} \\
&= \phi^{\mathsf{o}}.
\end{aligned}$$

$\phi^{\mathsf{i}} = \phi'^{\mathsf{i}}$**:** Easily seen by unfolding the definitions:

$$\begin{aligned}\phi'^{\mathsf{i}} &= \mathit{link}_D^{-1}\restriction_{X_G}\\ &= ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}'_D)^{-1}\restriction_{X_G}\\ &= ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}_D\restriction_{P'_D})^{-1}\restriction_{X_G}\\ &= ((\phi^{\mathsf{i}})^{-1} \uplus \mathit{link}_D\restriction_{P_D\setminus \mathit{link}_D^{-1}(X_G)})^{-1}\restriction_{X_G}\\ &= \phi^{\mathsf{i}}.\end{aligned}$$

□

A.2 Bigraph Edit Scripts

A.2.1 Proof of Prop. 6.13

It is straightforward to check that $\tilde{H}'$ is a pattern, since δ is compatible with $\tilde{P}$ and ϕ is an embedding and thus satisfies the embedding conditions of Def. 5.1 Def. 5.4, and Def. 5.14. Also, $\tilde{H}'$ clearly has the same outer face and inner names as $\tilde{H}$ since mediated edits can only affect the set of inner variables of a patterns interfaces.

What remains is to check that for each mediated edit, $\phi' : \tilde{P}' \hookrightarrow \tilde{H}'$ is an embedding, i.e., that it satisfies the embedding conditions. In most cases this follows easily from the fact that ϕ satisfies the conditions and we shall omit these, but a few cases are more interesting. Note that, by Prop. 6.7, $\tilde{P}'$ is a pattern.

$\oplus_{v:K_{\vec{y}}@p}$**:** We have $v \notin V_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R_{\tilde{P}}$, $v' \notin V_{\tilde{H}}$, and

$$\begin{aligned}\tilde{P}' &= (V_{\tilde{P}} + v, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}[v \mapsto K], \mathit{prnt}_{\tilde{P}}[v \mapsto p],\\ &\qquad \mathit{link}_{\tilde{P}}[(v,0) \mapsto \vec{y}_0, \dots, (v, n-1) \mapsto \vec{y}_{n-1}]) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}}\rangle\\ \tilde{H}' &= (V_{\tilde{H}} + v', E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}}[v' \mapsto K], \mathit{prnt}_{\tilde{H}}[v' \mapsto \phi(p)],\\ &\qquad \mathit{link}_{\tilde{H}}[(v',0) \mapsto \phi(\vec{y}_0), \dots, (v', n-1) \mapsto \phi(\vec{y}_{n-1})])\\ \phi' &= \phi[v \mapsto v'].\end{aligned}$$

The interesting cases are:

(LGE-7) Assuming $e \in E_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}&(\phi'^{\mathsf{p}} \circ (\mathit{link}_{\tilde{P}}[(v,0) \mapsto \vec{y}_0, \dots, (v,n-1) \mapsto \vec{y}_{n-1}])^{-1})(e)\\ =&\phi'^{\mathsf{p}}(\mathit{link}_{\tilde{P}}^{-1}(e) \cup [(v,0) \mapsto \vec{y}_0, \dots, (v,n-1) \mapsto \vec{y}_{n-1}]^{-1}(e))\\ =&(\phi'^{\mathsf{p}} \circ \mathit{link}_{\tilde{P}}^{-1})(e) \cup (\phi'^{\mathsf{p}} \circ [(v,0) \mapsto \vec{y}_0, \dots, (v,n-1) \mapsto \vec{y}_{n-1}]^{-1})(e)\\ =&(\phi^{\mathsf{p}} \circ \mathit{link}_{\tilde{P}}^{-1})(e) \cup [(v',0) \mapsto \vec{y}_0, \dots, (v',n-1) \mapsto \vec{y}_{n-1}]^{-1}(e)\\ =&(\mathit{link}_{\tilde{H}}^{-1} \circ \phi^{\mathsf{e}})(e) \cup ([(v',0) \mapsto \phi(\vec{y}_0), \dots, (v',n-1) \mapsto \phi(\vec{y}_{n-1})]^{-1} \circ \phi'^{\mathsf{e}})(e)\\ =&(\mathit{link}_{\tilde{H}}^{-1} \circ \phi'^{\mathsf{e}})(e) \cup ([(v',0) \mapsto \phi(\vec{y}_0), \dots, (v',n-1) \mapsto \phi(\vec{y}_{n-1})]^{-1} \circ \phi'^{\mathsf{e}})(e)\\ =&((\mathit{link}_{\tilde{H}}[(v',0) \mapsto \phi(\vec{y}_0), \dots, (v',n-1) \mapsto \phi(\vec{y}_{n-1})])^{-1} \circ \phi'^{\mathsf{e}})(e).\end{aligned}$$

(PGE-7) We check the two cases for $w \in V_{\tilde{P}} + v$:

$w \in V_{\tilde{P}}$**:** We have the following equalities:

$$\begin{aligned}
&(\phi'^{\mathsf{c}} \circ (prnt_{\tilde{P}}[v \mapsto p])^{-1})(w)\\
=&\phi'^{\mathsf{c}}(prnt_{\tilde{P}}^{-1}(w) \cup [v \mapsto p]^{-1}(w))\\
=&(\phi'^{\mathsf{c}} \circ prnt_{\tilde{P}}^{-1})(w) \cup (\phi'^{\mathsf{c}} \circ [v \mapsto p]^{-1})(w)\\
=&(\phi^{\mathsf{c}} \circ prnt_{\tilde{P}}^{-1})(w) \cup [v' \mapsto p]^{-1}(w)\\
=&(prnt_{\tilde{H}}^{-1} \circ \phi^{\mathsf{v}})(w) \cup ([v' \mapsto \phi(p)]^{-1} \circ \phi'^{\mathsf{v}})(w)\\
=&(prnt_{\tilde{H}}^{-1} \circ \phi'^{\mathsf{v}})(w) \cup ([v' \mapsto \phi(p)]^{-1} \circ \phi'^{\mathsf{v}})(w)\\
=&((prnt_{\tilde{H}}[v' \mapsto \phi(p)])^{-1} \circ \phi'^{\mathsf{v}})(w).
\end{aligned}$$

$w = v$**:** Since $v \notin V_{\tilde{P}}$ and $v' \notin V_{\tilde{H}}$ we have $v \neq p$, $v \notin \mathsf{rng}(prnt_{\tilde{P}})$, $v' \neq \phi(p)$, and $v' \notin \mathsf{rng}(prnt_{\tilde{H}})$. Thus $(prnt_{\tilde{P}}[v \mapsto p])^{-1}(v) = \emptyset = (prnt_{\tilde{H}}[v' \mapsto \phi(p)])^{-1}(v')$.

$\oplus_e$: We have $e \notin E_{\tilde{P}}$, $e' \notin E_{\tilde{H}}$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}} + e, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle\\
\tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}} + e', ctrl_{\tilde{H}}, prnt_{\tilde{H}}, link_{\tilde{H}})\\
\phi' &= \phi[e \mapsto e'].
\end{aligned}$$

All the conditions are obviously satisfied.

$\ominus_v$: We have $v \in V_{\tilde{P}}$, $prnt_{\tilde{P}}^{-1}(v) = \emptyset$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}} - v, E_{\tilde{P}}, ctrl_{\tilde{P}} - v, prnt_{\tilde{P}} - v, link_{\tilde{P}} - P_v) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle\\
\tilde{H}' &= (V_{\tilde{H}} - \phi(v), E_{\tilde{H}}, ctrl_{\tilde{H}} - \phi(v), prnt_{\tilde{H}} - \phi(v), link_{\tilde{H}} - P_{\phi(v)})\\
\phi' &= \phi - v.
\end{aligned}$$

The interesting cases are:

(LGE-7) Assuming $e \in E_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}
&(\phi'^{\mathsf{p}} \circ (link_{\tilde{P}} - P_v)^{-1})(e)\\
=&\phi^{\mathsf{p}}(link_{\tilde{P}}^{-1}(e) \setminus P_v)\\
=&(\phi^{\mathsf{p}} \circ link_{\tilde{P}}^{-1})(e) \setminus \phi(P_v)\\
=&(link_{\tilde{H}}^{-1} \circ \phi^{\mathsf{e}})(e) \setminus P_{\phi(v)}\\
=&((link_{\tilde{H}} - P_{\phi(v)})^{-1} \circ \phi'^{\mathsf{e}})(e).
\end{aligned}$$

(PGE-7) Assuming $w \in V_{\tilde{P}} - v$ we have the following equalities:

$$\begin{aligned}
&(\phi'^{\mathsf{c}} \circ (prnt_{\tilde{P}} - v)^{-1})(w)\\
=&\phi^{\mathsf{c}}(prnt_{\tilde{P}}^{-1}(w) - v)\\
=&(\phi^{\mathsf{c}} \circ prnt_{\tilde{P}}^{-1})(w) - \phi(v)\\
=&(prnt_{\tilde{H}}^{-1} \circ \phi^{\mathsf{v}})(w) - \phi(v)\\
=&((prnt_{\tilde{H}} - \phi(v))^{-1} \circ \phi'^{\mathsf{v}})(w).
\end{aligned}$$

$\ominus_e$: We have $e \in E_{\tilde{P}}$, $link_{\tilde{P}}^{-1}(e) = \emptyset$, and

$$\begin{aligned}\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}} - e, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\ \tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}} - \phi(e), ctrl_{\tilde{H}}, prnt_{\tilde{H}}, link_{\tilde{H}}) \\ \phi' &= \phi - e.\end{aligned}$$

All the conditions are obviously satisfied.

$\ominus_q$**:** We have $q \in Q_{\tilde{P}}$, and

$$\begin{aligned}\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}} - q, link_{\tilde{P}}) : (Q_{\tilde{P}} - q) \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\ \tilde{H}' &= (V_{\tilde{H}} \setminus \tilde{H}\downharpoonleft^{\phi(q)}, E_{\tilde{H}}, ctrl_{\tilde{H}} - \tilde{H}\downharpoonleft^{\phi(q)}, prnt_{\tilde{H}} - \tilde{H}\downharpoonleft^{\phi(q)}, link_{\tilde{H}} - P_{\tilde{H}\downharpoonleft^{\phi(q)}}) \\ &\quad : \langle Q_{\tilde{H}} \setminus \tilde{H}\downharpoonleft^{\phi(q)}, X_{\tilde{H}} \rangle \to I \\ \phi' &= \phi - q.\end{aligned}$$

The interesting cases are:

(LGE-7) Assuming $e \in E_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}&(\phi'^{\mathsf{p}} \circ link_{\tilde{P}}^{-1})(e) \\ =&(\phi^{\mathsf{p}} \circ link_{\tilde{P}}^{-1})(e) \setminus P_{\tilde{H}\downharpoonleft^{\phi(q)}} \\ =&(link_{\tilde{H}}^{-1} \circ \phi^{\mathsf{e}})(e) \setminus P_{\tilde{H}\downharpoonleft^{\phi(q)}} \\ =&((link_{\tilde{H}} - P_{\tilde{H}\downharpoonleft^{\phi(q)}})^{-1} \circ \phi'^{\mathsf{e}})(e)\end{aligned}$$

since $\mathsf{rng}(\phi^{\mathsf{p}}) = \mathsf{rng}(\phi^{\mathsf{port}}) \# P_{\tilde{H}\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}} \supseteq P_{\tilde{H}\downharpoonleft^{\phi(q)}}$, cf. Corollary 5.24.

(PGE-7) Assuming $w \in V_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}&(\phi'^{\mathsf{c}} \circ (prnt_{\tilde{P}} - q)^{-1})(w) \\ =&\phi^{\mathsf{c}}(prnt_{\tilde{P}}^{-1}(w) - q) \setminus (\tilde{H}\downharpoonleft^{\phi^{\mathsf{s}}(q)} \setminus \phi^{\mathsf{s}}(q)) \\ =&((\phi^{\mathsf{c}} \circ prnt_{\tilde{P}}^{-1})(w) \setminus \phi(q)) \setminus (\tilde{H}\downharpoonleft^{\phi^{\mathsf{s}}(q)} \setminus \phi^{\mathsf{s}}(q)) \\ =&(prnt_{\tilde{H}}^{-1} \circ \phi^{\mathsf{v}})(w) \setminus \tilde{H}\downharpoonleft^{\phi^{\mathsf{s}}(q)} \\ =&((prnt_{\tilde{H}} - \tilde{H}\downharpoonleft^{\phi^{\mathsf{s}}(q)})^{-1} \circ \phi'^{\mathsf{v}})(w)\end{aligned}$$

since $\mathsf{rng}(\phi^{\mathsf{c}}) \# (\tilde{H}\downharpoonleft^{\phi^{\mathsf{s}}(q)} \setminus \phi^{\mathsf{s}}(q))$, cf. Lemma 5.11.

$\oslash_{v@p}$**:** We have $v \in V_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R_{\tilde{P}}$, and

$$\begin{aligned}\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}[v \mapsto p], link_{\tilde{P}}) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\ \tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}}, ctrl_{\tilde{H}}, prnt_{\tilde{H}}[\phi(v) \mapsto \phi(p)], link_{\tilde{H}}) \\ \phi' &= \phi.\end{aligned}$$

All the conditions are obviously satisfied.

$\oslash_{q@p}$**:** We have $q \in Q_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R_{\tilde{P}}$, and

$$\begin{aligned}\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}[q \mapsto p], link_{\tilde{P}}) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\ \tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}}, ctrl_{\tilde{H}}, prnt_{\tilde{H}}[\phi(q) \mapsto \phi(p)], link_{\tilde{H}}) \\ \phi' &= \phi.\end{aligned}$$

All the conditions are obviously satisfied.

$\otimes_{q\to r@p}$: We have $q \in Q_{\tilde{P}}$, $r \notin Q_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R_{\tilde{P}}$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[r \mapsto p], \mathit{link}_{\tilde{P}}) : (Q_{\tilde{P}} + r) \to \langle R_{\tilde{P}}, Y_{\tilde{P}}\rangle \\
\tilde{H}' &= (V_{\tilde{H}} \uplus V_r, E_{\tilde{H}}, \mathit{ctrl}_{\tilde{H}} \uplus \mathit{ctrl}_r, \mathit{prnt}_{\tilde{H}} \uplus \mathit{prnt}_r, \mathit{link}_{\tilde{H}} \uplus \mathit{link}_r) \\
&\quad : \langle Q_{\tilde{H}} \uplus Q_r, X\rangle \to I \\
\phi' &= \phi[r \mapsto f^{-1}(\phi(q))]
\end{aligned}$$

where

$$\begin{aligned}
V_q &= \tilde{H}\downharpoonleft^{\phi(q)} \cap V_{\tilde{H}} & Q_q &= \tilde{H}\downharpoonleft^{\phi(q)} \cap Q \\
|V_r| &= |V_q| & |Q_r| &= |Q_q| \\
V_r &\,\#\, V_{\tilde{H}} & Q_r &\,\#\, Q \\
f_v &: V_r \rightarrowtail V_q & f_s &: Q_r \rightarrowtail Q_q \\
f &= f_v \uplus f_s \\
\mathit{ctrl}_r &= \mathit{ctrl}_{\tilde{H}} \circ f_v \\
\mathit{prnt}_r &= \{f^{-1}(\phi(q)) \mapsto \phi(p)\} \uplus f_v^{-1} \circ \mathit{prnt}_{\tilde{H}} \circ (f - f^{-1}(\phi(q))) \\
\mathit{link}_r(v,i) &= \mathit{link}_{\tilde{H}}(f_v(v), i) \quad (v \in V_r)
\end{aligned}$$

The interesting cases are:

(LGE-7) Assuming $e \in E_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}
&(\phi'^{\mathsf{p}} \circ \mathit{link}_{\tilde{P}}^{-1})(e) \\
=&(\phi^{\mathsf{p}} \circ \mathit{link}_{\tilde{P}}^{-1})(e) \\
=&(\mathit{link}_{\tilde{H}}^{-1} \circ \phi^{\mathsf{e}})(e) \\
=&(\mathit{link}_{\tilde{H}}^{-1} \circ \phi'^{\mathsf{e}})(e) \cup (\mathit{link}_r^{-1} \circ \phi'^{\mathsf{e}})(e) \\
=&((\mathit{link}_{\tilde{H}} \uplus \mathit{link}_r)^{-1} \circ \phi'^{\mathsf{e}})(e)
\end{aligned}$$

since $\mathit{link}_{\tilde{H}}^{-1}(\mathsf{rng}(\phi^{\mathsf{e}})) \subseteq \mathsf{rng}(\phi^{\mathsf{p}}) = \mathsf{rng}(\phi^{\mathsf{port}}) \,\#\, P_{\tilde{H}\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}} \supseteq P_{\tilde{H}\downharpoonleft^{\phi(q)}}$, cf. condition (LGE-7) and Corollary 5.24, and thus $\mathit{link}_r^{-1}(\mathsf{rng}(\phi'^{\mathsf{e}})) = \phi'^{\mathsf{p}}((\mathit{link}_{\tilde{H}} \restriction_{P_{V_q}})^{-1}(\mathsf{rng}(\phi'^{\mathsf{e}}))) = \phi'^{\mathsf{p}}((\mathit{link}_{\tilde{H}} \restriction_{P_{\tilde{H}\downharpoonleft^{\phi(q)}}}$ $)^{-1}(\mathsf{rng}(\phi^{\mathsf{e}}))) = \phi'^{\mathsf{p}}(\emptyset) = \emptyset$.

(PGE-7) Assuming $w \in V_{\tilde{P}}$ we have the following equalities:

$$\begin{aligned}
&(\phi'^{\mathsf{c}} \circ (\mathit{prnt}_{\tilde{P}}[r \mapsto p])^{-1})(w) \\
=&\phi'^{\mathsf{c}}(\mathit{prnt}_{\tilde{P}}^{-1}(w) \cup [r \mapsto p]^{-1}(w)) \\
=&(\phi'^{\mathsf{c}} \circ \mathit{prnt}_{\tilde{P}}^{-1})(w) \cup (\phi'^{\mathsf{c}} \circ [r \mapsto p]^{-1})(w) \\
=&(\phi^{\mathsf{c}} \circ \mathit{prnt}_{\tilde{P}}^{-1})(w) \cup [f^{-1}(\phi(q)) \mapsto p]^{-1}(w) \\
=&(\mathit{prnt}_{\tilde{H}}^{-1} \circ \phi^{\mathsf{v}})(w) \cup ([f^{-1}(\phi(q)) \mapsto \phi'^{\mathsf{v}}(p)]^{-1} \circ \phi'^{\mathsf{v}})(w) \\
&\cup ((f_v^{-1} \circ \mathit{prnt}_{\tilde{H}} \circ (f - f^{-1}(\phi(q))))^{-1} \circ \phi'^{\mathsf{v}})(w) \\
=&(\mathit{prnt}_{\tilde{H}}^{-1} \circ \phi'^{\mathsf{v}})(w) \\
&\cup (([f^{-1}(\phi(q)) \mapsto \phi'^{\mathsf{v}}(p)] \uplus f_v^{-1} \circ \mathit{prnt}_{\tilde{H}} \circ (f - f^{-1}(\phi(q))))^{-1} \circ \phi'^{\mathsf{v}})(w) \\
=&(\mathit{prnt}_{\tilde{H}}^{-1} \circ \phi'^{\mathsf{v}})(w) \cup (\mathit{prnt}_r^{-1} \circ \phi'^{\mathsf{v}})(w) \\
=&((\mathit{prnt}_{\tilde{H}} \uplus \mathit{prnt}_r)^{-1} \circ \phi'^{\mathsf{v}})(w)
\end{aligned}$$

since $\mathsf{rng}(f_v^{-1}) = V_r \,\#\, V_{\tilde{H}} \supseteq \mathsf{rng}(\phi'^{\mathsf{v}})$ and thus $((f_v^{-1} \circ \mathit{prnt}_{\tilde{H}} \circ (f - f^{-1}(\phi(q))))^{-1} \circ \phi'^{\mathsf{v}})(w) = \emptyset$.

$\odot_{(v,i)\mapsto l}$: We have $v \in V_{\tilde{P}}$, $i \in ar(ctrl_{\tilde{P}}(v))$, $l \in E_{\tilde{P}} \uplus Y_{\tilde{P}}$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}[(v,i) \mapsto l]) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\
\tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}}, ctrl_{\tilde{H}}, prnt_{\tilde{H}}, link_{\tilde{H}}[(\phi(v),i) \mapsto \phi(l)]) \\
\phi' &= \phi.
\end{aligned}$$

All the conditions are obviously satisfied.

$\circledcirc_{v:K}$: We have $v \in V_{\tilde{P}}$, $ar(K) = ar(ctrl_{\tilde{P}}(v))$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}[v \mapsto K], prnt_{\tilde{P}}, link_{\tilde{P}}) : Q_{\tilde{P}} \to \langle R_{\tilde{P}}, Y_{\tilde{P}} \rangle \\
\tilde{H}' &= (V_{\tilde{H}}, E_{\tilde{H}}, ctrl_{\tilde{H}}[\phi(v) \mapsto K], prnt_{\tilde{H}}, link_{\tilde{H}}) \\
\phi' &= \phi.
\end{aligned}$$

All the conditions are obviously satisfied.

□

A.2.2 Proof of Lemma 6.14

By Corol. 5.23 we have a match

$$a = ctxt([\![\phi]\!]) \circ ([\![\phi]\!] \cdot [\![\tilde{P}]\!] \otimes \mathsf{id}_{X_I}) \circ prmt([\![\phi]\!])$$

for some set of names X_I, so we just have to show

$$a' = ctxt([\![\phi]\!]) \circ ([\![\phi']\!] \cdot [\![\delta(\tilde{P})]\!] \otimes \mathsf{id}_{X_I}) \circ \overline{[\![inst_Q(finst(\delta))]\!]}(prmt([\![\phi]\!]))$$

and $a \twoheadrightarrow a'$ then follows from cf. Def. 4.14.

Let

$$\begin{aligned}
D : \langle |Q|, X_I \rangle &= prmt([\![\phi]\!]) \\
C : \langle |R|, Y \uplus X_I \rangle \to J &= ctxt([\![\phi]\!]) \\
\tilde{P}' : Q' \to \langle R, Y \rangle &= \delta(\tilde{P}) \\
Q &= \{q_0, \ldots, q_{k-1}\} && \text{where } \forall i \in [0; k-2] : q_i < q_{i+1} \\
Q' &= \{q'_0, \ldots, q'_{k'-1}\} && \text{where } \forall i \in [0; k'-2] : q'_i < q'_{i+1} \\
R &= \{r_0, \ldots, r_{m-1}\} && \text{where } \forall i \in [0; m-2] : r_i < r_{i+1}.
\end{aligned}$$

By the definitions of parameter (cf. Defs. 5.8 and 5.20) and patterns (cf. Def. 6.1) we have the following equalities:

$$\begin{aligned}
D : \langle |Q|, X_I \rangle = (&a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}, \emptyset, ctrl_a \upharpoonleft_{a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}}, \\
&[\phi(q_0) \mapsto 0, \ldots, \phi(q_{k-1}) \mapsto k-1] \\
&\quad \uplus prnt_a \upharpoonleft_{a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \setminus \mathsf{rng}(\phi^{\mathsf{s}})}, \\
&link'_D) && \text{Defs. 5.8 and 5.20} \\
= {}& d_0 \otimes \cdots \otimes d_{|Q|-1} && \text{Def. 2.34 and } link'_D \text{ a bijection} \\
V_{d_i} = {}& a \downharpoonleft^{\phi^{\mathsf{s}}(q_i)} && \text{Defs. 6.1 and 2.9}
\end{aligned}$$

where

$$link'_D : P_{a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}} \to X_I \text{ a bijection.}$$

By the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8) and the constructions from Defs. 5.8, 5.20, and 6.1 we have the following equalities:

$$\begin{aligned}
V_a &= V_C \uplus \phi^{\mathsf{v}}(V_{\tilde{P}}) \uplus V_D \\
E_a &= E_C \uplus \phi^{\mathsf{e}}(E_{\tilde{P}}) \\
ctrl_a &= ctrl_a \restriction_{V_C} \uplus ctrl_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]} \uplus ctrl_a \restriction_{V_D} \\
prnt_a(w) &= \begin{cases}
prnt_D(w) & \text{if } w \in V_D \text{ and } prnt_D(w) \in V_D \\
prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) & \text{if } w \in V_D \text{ and } prnt_D(w) = i \in |Q| \\
& \quad \text{and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
prnt_C(j) & \text{if } w \in V_D \text{ and } prnt_D(w) = i \in |Q| \\
& \quad \text{and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) = j \in |R| \\
prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
prnt_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) = i \in |R| \\
prnt_C(w) & \text{if } w \in V_C
\end{cases} \\
link_a(p) &= \begin{cases}
link_C(x) & \text{if } p \in P_D \text{ and } link_D(p) = x \in X_I \\
link_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } link_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) \in \phi^{\mathsf{e}}(E_{\tilde{P}}) \\
link_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } link_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) = y \in Y \\
link_C(p) & \text{if } p \in P_C
\end{cases}
\end{aligned}$$

where

$$\begin{aligned}
ctrl_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]} &= ctrl_{\tilde{P}} \circ (\phi^{\mathsf{v}})^{-1} \\
link_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]} &= \phi^{\mathsf{l}} \circ link_{\tilde{P}} \circ (\phi^{\mathsf{p}})^{-1}, \\
prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]} &= (\phi^{\mathsf{v}} \uplus \{r_0 \mapsto 0, \ldots, r_{m-1} \mapsto m-1\}) \\
&\quad \circ prnt_{\tilde{P}} \\
&\quad \circ ((\phi^{\mathsf{v}})^{-1} \uplus \{0 \mapsto q_0, \ldots, k-1 \mapsto q_{k-1}\}).
\end{aligned}$$

Similarly, unfolding the definitions of composition and tensor product (cf. Def. 2.6 and Def. 2.8) and the constructions from Defs. 5.8, 5.20, and 6.1, we see that we have to show the following equalities in order for a' to be on the prescribed form:

$$\begin{aligned}
V_{a'} &= V_C \uplus \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \uplus V_{D'}\\
E_{a'} &= E_C \uplus \phi'^{\mathsf{e}}(E_{\tilde{P}'})\\
\mathit{ctrl}_{a'} &= \mathit{ctrl}_a \restriction_{V_C} \uplus \mathit{ctrl}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]} \uplus \mathit{ctrl}_a \restriction_{V_{D'}}\\
\mathit{prnt}_{a'}(w) &= \begin{cases}
\mathit{prnt}_{D'}(w) & \text{if } w \in V_{D'}\\
 & \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'}\\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'}\\
 & \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'|\\
 & \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'})\\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'}\\
 & \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'|\\
 & \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) = j \in |R|\\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'})\\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) = i \in |R|\\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases}\\
\mathit{link}_{a'}(p) &= \begin{cases}
\mathit{link}_C(x) & \text{if } p \in P_{D'} \text{ and } \mathit{link}_{D'}(p) = x \in X_I\\
\mathit{link}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(p) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'})\\
\mathit{link}_C(y) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(p) = y \in Y\\
\mathit{link}_C(p) & \text{if } p \in P_C
\end{cases}
\end{aligned}$$

where

$$\begin{aligned}
D' : \langle |Q'|, X_I\rangle &= \overline{[\![\mathit{inst}_Q(\mathit{finst}(\delta))]\!]}(\mathit{prmt}([\![\phi]\!])),\\
\mathit{ctrl}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]} &= \mathit{ctrl}_{\tilde{P}'} \circ (\phi'^{\mathsf{v}})^{-1},\\
\mathit{link}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]} &= \phi'^{\mathsf{l}} \circ \mathit{link}_{\tilde{P}'} \circ (\phi'^{\mathsf{p}})^{-1},\\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]} &= (\phi'^{\mathsf{v}} \uplus \{r_0 \mapsto 0, \dots, r_{m-1} \mapsto m-1\})\\
&\quad \circ \mathit{prnt}_{\tilde{P}'}\\
&\quad \circ ((\phi'^{\mathsf{v}})^{-1} \uplus \{0 \mapsto q'_0, \dots, k-1 \mapsto q'_{k-1}\}),
\end{aligned}$$

We show that this is the case for each edit:

$\oplus_{v:K_{\vec{y}}@p}$**:** We have $v \notin V_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R$, $v' \notin V_a$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}} + v, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}[v \mapsto K], \mathit{prnt}_{\tilde{P}}[v \mapsto p],\\
&\qquad \mathit{link}_{\tilde{P}}[(v,0) \mapsto \vec{y}_0, \dots, (v, n-1) \mapsto \vec{y}_{n-1}]) : Q \to \langle R, Y\rangle\\
a' &= (V_a + v', E_a, \mathit{ctrl}_a[v' \mapsto K], \mathit{prnt}_a[v' \mapsto \phi(p)],\\
&\qquad \mathit{link}_a[(v',0) \mapsto \phi(\vec{y}_0), \dots, (v', n-1) \mapsto \phi(\vec{y}_{n-1})])\\
\phi' &= \phi[v \mapsto v']\\
\mathit{inst}_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}}\\
D' : \langle |Q'|, X_I\rangle &= D : \langle |Q|, X_I\rangle.
\end{aligned}$$

The interesting cases are the parent and link maps:

$$
\begin{aligned}
\mathit{prnt}_{a'}(w) &= \mathit{prnt}_a[v' \mapsto \phi(p)](w) \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) \in V_D \\
\mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(i) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } \mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } \mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C \\
\phi(p) & \text{if } w = v'
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_{D'}(w) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'} \\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) + v' \\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) + v' \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) + v' \\
\mathit{prnt}_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) + v' \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_{D'}(w) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'} \\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad\text{and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases}
\end{aligned}
$$

since $v' \notin V_a \uplus m_a = \mathsf{cod}(\phi^{\mathsf{f}}) \supseteq \mathsf{rng}(\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]})$ and $\mathit{prnt}_{[\![\phi']\!]\bullet[\![\tilde{P}']\!]} = \mathit{prnt}_{[\![\phi]\!]\bullet[\![\tilde{P}]\!]}[v' \mapsto \phi(p)]$.

$$\begin{aligned}
\mathit{link}_{a'}(p) &= \mathit{link}_a[(v',0) \mapsto \phi(\vec{y}_0), \ldots, (v', n-1) \mapsto \phi(\vec{y}_{n-1})](p) \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_D \text{ and } \mathit{link}_D(p) = x \in X_I \\ \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) \in \phi^{\mathsf{e}}(E_{\tilde{P}}) \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \\ [\ldots, (v', i) \mapsto \phi(\vec{y}_i), \ldots](p) & \text{if } p \in P_{v'} \end{cases} \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_{D'} \text{ and } \mathit{link}_{D'}(p) = x \in X_I \\ \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \uplus P_{v'} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'}) \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \uplus P_{v'} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases} \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_{D'} \text{ and } \mathit{link}_{D'}(p) = x \in X_I \\ \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'}) \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases}
\end{aligned}$$

since $\mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]} = \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}[\ldots, (v', i) \mapsto \phi(\vec{y}_i), \ldots]$.

$\oplus_e$: We have $e \notin E_{\tilde{P}}$, $e' \notin E_a$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}} + e, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}, \mathit{link}_{\tilde{P}}) : Q \to \langle R, Y \rangle \\
a' &= (V_a, E_a + e', \mathit{ctrl}_a, \mathit{prnt}_a, \mathit{link}_a) \\
\phi' &= \phi[e \mapsto e'] \\
\mathit{inst}_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle.
\end{aligned}$$

The interesting case is the link map:

$$\begin{aligned}
\mathit{link}_{a'}(p) &= \mathit{link}_a(p) \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_D \text{ and } \mathit{link}_D(p) = x \in X_I \\ \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) \in \phi^{\mathsf{e}}(E_{\tilde{P}}) \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases} \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_D \text{ and } \mathit{link}_D(p) = x \in X_I \\ \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) \in \phi^{\mathsf{e}}(E_{\tilde{P}}) + e' \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \text{ and } \mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases} \\
&= \begin{cases} \mathit{link}_C(x) & \text{if } p \in P_{D'} \text{ and } \mathit{link}_{D'}(p) = x \in X_I \\ \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'}) \\ \mathit{link}_C(y) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } \mathit{link}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(p) = y \in Y \\ \mathit{link}_C(p) & \text{if } p \in P_C \end{cases}
\end{aligned}$$

since $e' \notin E_a \uplus Y \supseteq \mathsf{rng}(\mathit{link}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]})$.

$\ominus_v$: We have $v \in V_{\tilde{P}}$, $prnt_{\tilde{P}}^{-1}(v) = \emptyset$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}} - v, E_{\tilde{P}}, ctrl_{\tilde{P}} - v, prnt_{\tilde{P}} - v, link_{\tilde{P}} - P_v) : Q \to \langle R, Y \rangle \\
a' &= (V_a - \phi(v), E_a, ctrl_a - \phi(v), prnt_a - \phi(v), link_a - P_{\phi(v)}) \\
\phi' &= \phi - v \\
inst_Q(finst(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle .
\end{aligned}$$

The interesting case is the link map:

$$\begin{aligned}
link_{a'}(p) &= (link_a - P_{\phi(v)})(p) \\
&= \begin{cases}
link_C(x) & \text{if } p \in P_D \setminus P_{\phi(v)} \text{ and } link_D(p) = x \in X_I \\
link_{\llbracket\phi\rrbracket \blacksquare \llbracket\tilde{P}\rrbracket}(p) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \setminus P_{\phi(v)} \text{ and } link_{\llbracket\phi\rrbracket \blacksquare \llbracket\tilde{P}\rrbracket}(p) \in \phi^{\mathsf{e}}(E_{\tilde{P}}) \\
link_C(y) & \text{if } p \in P_{\phi^{\mathsf{v}}(V_{\tilde{P}})} \setminus P_{\phi(v)} \text{ and } link_{\llbracket\phi\rrbracket \blacksquare \llbracket\tilde{P}\rrbracket}(p) = y \in Y \\
link_C(p) & \text{if } p \in P_C \setminus P_{\phi(v)}
\end{cases} \\
&= \begin{cases}
link_C(x) & \text{if } p \in P_{D'} \text{ and } link_{D'}(p) = x \in X_I \\
link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}} - v)} \text{ and } link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'}) \\
link_C(y) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}} - v)} \text{ and } link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) = y \in Y \\
link_C(p) & \text{if } p \in P_C
\end{cases} \\
&= \begin{cases}
link_C(x) & \text{if } p \in P_{D'} \text{ and } link_{D'}(p) = x \in X_I \\
link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) \in \phi'^{\mathsf{e}}(E_{\tilde{P}'}) \\
link_C(y) & \text{if } p \in P_{\phi'^{\mathsf{v}}(V_{\tilde{P}'})} \text{ and } link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket}(p) = y \in Y \\
link_C(p) & \text{if } p \in P_C
\end{cases}
\end{aligned}$$

since $\phi(v) \in \phi^{\mathsf{v}}(V_{\tilde{P}})$, $\phi^{\mathsf{v}}(V_{\tilde{P}}) \# V_D$, $\phi^{\mathsf{v}}(V_{\tilde{P}}) \# V_C$, and $link_{\llbracket\phi'\rrbracket \blacksquare \llbracket\tilde{P}'\rrbracket} = link_{\llbracket\phi\rrbracket \blacksquare \llbracket\tilde{P}\rrbracket} - P_{\phi(v)}$.

$\ominus_e$: We have $e \in E_{\tilde{P}}$, $link_{\tilde{P}}^{-1}(e) = \emptyset$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}} - e, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}) : Q \to \langle R, Y \rangle \\
a' &= (V_a, E_a - \phi(e), ctrl_a, prnt_a, link_a) \\
\phi' &= \phi - e \\
inst_Q(finst(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle .
\end{aligned}$$

All the equalities obviously hold.

$\ominus_q$**:** We have $q \in Q$, and (noting that a has no sites)

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}} - q, link_{\tilde{P}}) : (Q - q) \to \langle R, Y \rangle \\
a' &= (V_a \setminus a\downharpoonleft^{\phi(q)}, E_a, ctrl_a - a\downharpoonleft^{\phi(q)}, prnt_a - a\downharpoonleft^{\phi(q)}, link_a - P_{a\downharpoonleft^{\phi(q)}})) \\
\phi' &= \phi - q \\
i_q &= i \qquad \text{if } q_i = q \\
inst_Q(finst(\delta)) &= (\mathsf{Id}_{\mathcal{U}}[q \mapsto \emptyset]\upharpoonright_Q)^{-1} \\
&= \mathsf{Id}_{Q-q} \\
\llbracket inst_Q(finst(\delta)) \rrbracket &= \llbracket \mathsf{Id}_{Q-q} \rrbracket \\
&= \mathsf{Id}_{i_q} \uplus [i_q \mapsto i_q + 1, \ldots, |Q| - 2 \mapsto |Q| - 1] \\
D' : \langle |Q'|, X_I \rangle &= \overline{\llbracket \mathsf{Id}_{Q-q} \rrbracket}(D)
\end{aligned}$$

By the definition of instantiation (Def. 2.35), we obtain:

$$\begin{aligned}
D' : \langle |Q'|, X_I\rangle &= \overline{[\![\mathsf{Id}_{Q-q}]\!]}(D) \\
&= d_{[\![\mathsf{Id}_{Q-q}]\!](0)} \parallel \cdots \parallel d_{[\![\mathsf{Id}_{Q-q}]\!](|Q|-2)} \\
&= d_0 \otimes \cdots \otimes d_{i_q-1} \otimes d_{i_q+1} \otimes \cdots \otimes d_{|Q|-1} \\
&= (a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \setminus a \downharpoonleft^{\phi^{\mathsf{s}}(q)}, \emptyset, \mathit{ctrl}_a \restriction_{a\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \setminus a\downharpoonleft^{\phi^{\mathsf{s}}(q)}}, \\
&\qquad ([\![\mathsf{Id}_{Q-q}]\!]^{-1} \uplus \mathsf{Id}_{a\downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})} \setminus a\downharpoonleft^{\phi^{\mathsf{s}}(q)}}) \\
&\qquad\quad \circ (\mathit{prnt}_D - a \downharpoonleft^{\phi^{\mathsf{s}}(q)}), \\
&\qquad \mathit{link}'_D - P_{a\downharpoonleft^{\phi^{\mathsf{s}}(q)}})
\end{aligned}$$

The interesting case is the parent map:

$$\begin{aligned}
\mathit{prnt}_{a'}(w) &= (\mathit{prnt}_a - a \downharpoonleft^{\phi(q)})(w) \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_D \setminus a \downharpoonleft^{\phi(q)} \text{ and } \mathit{prnt}_D(w) \in V_D \\
\mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(i) & \text{if } w \in V_D \setminus a \downharpoonleft^{\phi(q)} \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_D \setminus a \downharpoonleft^{\phi(q)} \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \setminus a \downharpoonleft^{\phi(q)} \text{ and } \mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \setminus a \downharpoonleft^{\phi(q)} \text{ and } \mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C \setminus a \downharpoonleft^{\phi(q)}
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'} \\
\mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases}
\end{aligned}$$

since

$$\begin{aligned}
\phi^{\mathsf{v}}(V_{\tilde{P}}) \,\#\, a \downharpoonleft^{\phi(q)} & & \text{Lemma 5.11} \\
V_C \,\#\, a \downharpoonleft^{\phi(q)} & & \text{Def. 5.8} \\
\mathit{prnt}_{D'} &= ([\![\mathsf{Id}_{Q-q}]\!]^{-1} \uplus \mathsf{Id}_{V_{D'}}) \\
&\qquad \circ (\mathit{prnt}_D - a \downharpoonleft^{\phi^{\mathsf{s}}(q)}) \\
\mathsf{rng}(\mathit{prnt}_{D'}) &= V_{D'} \uplus |Q'| \\
\mathit{prnt}_{[\![\phi']\!]\blacksquare[\![\tilde{P}']\!]} &= \mathit{prnt}_{[\![\phi]\!]\blacksquare[\![\tilde{P}]\!]} \circ ([\![\mathsf{Id}_{Q-q}]\!] \uplus \mathsf{Id}_{\mathsf{rng}(\phi'^{\mathsf{v}})}).
\end{aligned}$$

$\oslash_{v@p}$: We have $v \in V_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[v \mapsto p], \mathit{link}_{\tilde{P}}) : Q \to \langle R, Y\rangle \\
a' &= (V_a, E_a, \mathit{ctrl}_a, \mathit{prnt}_a[\phi(v) \mapsto \phi(p)], \mathit{link}_a) \\
\phi' &= \phi \\
\mathit{inst}_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I\rangle &= D : \langle |Q|, X_I\rangle.
\end{aligned}$$

The interesting case is the parent map:

$$
\begin{aligned}
\mathit{prnt}_{a'}(w) &= \mathit{prnt}_a[\phi(v) \mapsto \phi(p)](w) \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_D - \phi(v) \text{ and } \mathit{prnt}_D(w) \in V_D \\
\mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) & \text{if } w \in V_D - \phi(v) \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_D - \phi(v) \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) - \phi(v) \text{ and } \mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) - \phi(v) \text{ and } \mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C - \phi(v) \\
\phi(p) & \text{if } w = \phi(v)
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_{D'}(w) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'} \\
\mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q| \\
& \quad \text{and } \mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) = j \in |R| \\
\mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases}
\end{aligned}
$$

since $\phi(v) \in \phi^{\mathsf{v}}(V_{\tilde{P}})$, $\phi^{\mathsf{v}}(V_{\tilde{P}}) \# V_D$, $\phi^{\mathsf{v}}(V_{\tilde{P}}) \# V_C$, and $\mathit{prnt}_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]} = \mathit{prnt}_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}[\phi(v) \mapsto \phi(p)]$.

$\oslash_{q@p}$**:** We have $q \in Q$, $p \in V_{\tilde{P}} \uplus R$, and

$$
\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}}, \mathit{prnt}_{\tilde{P}}[q \mapsto p], \mathit{link}_{\tilde{P}}) : Q \to \langle R, Y \rangle \\
a' &= (V_a, E_a, \mathit{ctrl}_a, \mathit{prnt}_a[\phi(q) \mapsto \phi(p)], \mathit{link}_a) \\
\phi' &= \phi \\
\mathit{inst}_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle.
\end{aligned}
$$

The interesting case is the parent map:

$$\begin{aligned}
prnt_{a'}(w) &= prnt_a[\phi(q) \mapsto \phi(p)](w) \\
&= \begin{cases}
prnt_D(w) & \text{if } w \in V_D \setminus \phi(q) \text{ and } prnt_D(w) \in V_D \\
prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) & \text{if } w \in V_D \setminus \phi(q) \text{ and } prnt_D(w) = i \in |Q| \\
 & \quad \text{and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
prnt_C(j) & \text{if } w \in V_D \setminus \phi(q) \text{ and } prnt_D(w) = i \in |Q| \\
 & \quad \text{and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(i) = j \in |R| \\
prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \setminus \phi(q) \text{ and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
prnt_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \setminus \phi(q) \text{ and } prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}(w) = i \in |R| \\
prnt_C(w) & \text{if } w \in V_C \setminus \phi(q) \\
\phi(p) & \text{if } w \in \phi(q)
\end{cases} \\
&= \begin{cases}
prnt_{D'}(w) & \text{if } w \in V_{D'} \text{ and } prnt_{D'}(w) \in V_{D'} \\
prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) & \text{if } w \in V_{D'} \text{ and } prnt_{D'}(w) = i \in |Q| \\
 & \quad \text{and } prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
prnt_C(j) & \text{if } w \in V_{D'} \text{ and } prnt_{D'}(w) = i \in |Q| \\
 & \quad \text{and } prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(i) = j \in |R| \\
prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
prnt_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]}(w) = i \in |R| \\
prnt_C(w) & \text{if } w \in V_C
\end{cases}
\end{aligned}$$

since $prnt_D(\phi(q)) \in |Q|$, $\phi(q) \subseteq V_D$, cf. Def. 5.8, $V_D \,\#\, \phi^{\mathsf{v}}(V_{\tilde{P}})$, $V_D \,\#\, V_C$, and $prnt_{[\![\phi']\!] \bullet [\![\tilde{P}']\!]} = prnt_{[\![\phi]\!] \bullet [\![\tilde{P}]\!]}[i \mapsto p']$ if $q = q_i$ and

$$p' = \begin{cases} j & \text{if } p = r_j \\ \phi^{\mathsf{v}}(p) & \text{if } p \in V_{\tilde{P}} \end{cases} .$$

$\otimes_{q \to r @ p}$**:** We have $q \in Q$, $r \notin Q$, $p \in V_{\tilde{P}} \uplus R$, and (noting that a has no sites)

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}[r \mapsto p], link_{\tilde{P}}) : (Q + r) \to \langle R, Y \rangle \\
a' &= (V_a \uplus V_r, E_a, ctrl_a \uplus ctrl_r, prnt_a \uplus prnt_r, link_a \uplus link_r) \\
\phi' &= \phi[r \mapsto f_v^{-1}(\phi(q))] \\
Q' &= \{q_0, \dots, q_{i_r - 1}, r, q_{i_r}, \dots, q_{k-1}\} \qquad \text{where } q_{i_r - 1} < r < q_{i_r} \\
i_q &= i \qquad \text{if } q_i = q \\
inst_Q(\mathit{finst}(\delta)) &= (\mathsf{Id}_{\mathcal{U} - r}[q \mapsto \{q, r\}] \restriction_Q)^{-1} \\
&= \mathsf{Id}_Q[r \mapsto q] \\
[\![inst_Q(\mathit{finst}(\delta))]\!] &= [\![\mathsf{Id}_Q[r \mapsto q]]\!] \\
&= \mathsf{Id}_{i_r} \uplus [i_r \mapsto i_q] \uplus [i_r + 1 \mapsto i_r, \dots, |Q| \mapsto |Q| - 1] \\
D' : \langle |Q'|, X_I \rangle &= \overline{[\![\mathsf{Id}_Q[r \mapsto q]]\!]}(D)
\end{aligned}$$

where

$$\begin{aligned}
V_q &= a \downharpoonleft^{\phi(q)} \\
|V_r| &= |V_q| \\
V_r &\mathbin{\#} V_a \\
f_v &: V_r \rightarrowtail V_q \\
ctrl_r &= ctrl_a \circ f_v \\
prnt_r &= \{f_v^{-1}(\phi(q)) \mapsto \phi(p)\} \uplus f_v^{-1} \circ prnt_a \circ (f_v - f_v^{-1}(\phi(q))) \\
link_r(v,i) &= link_a(f_v(v), i) \quad (v \in V_r)
\end{aligned}$$

By the definition of instantiation (Def. 2.35), we obtain:

$$\begin{aligned}
D' : \langle |Q'|, X_I \rangle &= \overline{[\![\mathsf{Id}_Q[r \mapsto q]]\!]}(D) \\
&= d_{[\![\mathsf{Id}_Q[r \mapsto q]]\!](0)} \parallel \cdots \parallel d_{[\![\mathsf{Id}_Q[r \mapsto q]]\!](|Q|)} \\
&= d_0 \otimes \cdots \parallel d_{i_r - 1} \parallel d_r \parallel d_{i_r} \parallel \cdots \otimes d_{|Q|-1} \\
&= (V_D \uplus V_r, \emptyset, ctrl_a \restriction_{a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}} \uplus ctrl_r, \\
&\qquad ((\mathsf{Id}_{i_r} \uplus [i_r + 1 \mapsto i_r, \ldots, |Q| \mapsto |Q| - 1])^{-1} \uplus \mathsf{Id}_{a \downharpoonleft^{\mathsf{rng}(\phi^{\mathsf{s}})}}) \circ prnt_D \\
&\qquad \uplus ([i_r \mapsto i_q]^{-1} \uplus f_v^{-1}) \circ prnt_D \circ f_v, \\
&\qquad link_D \uplus link_r) \\
d_r &= f_v^{-1} \mathbin{\blacksquare} d_{i_q}.
\end{aligned}$$

The interesting case is the parent map:

$$
\begin{aligned}
\mathit{prnt}_{a'}(w) &= (\mathit{prnt}_a \uplus \mathit{prnt}_r)(w) \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) \in V_D \\
\mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) = j \in |R| \\
\mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi^{\mathsf{v}}(V_{\tilde{P}}) \text{ and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C \\
\mathit{prnt}_r(w) & \text{if } w \in V_r
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_D(w) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) \in V_D \\
\mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_D \text{ and } \mathit{prnt}_D(w) = i \in |Q| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(i) = j \in |R| \\
\mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{\llbracket \phi \rrbracket \blacksquare \llbracket \tilde{P} \rrbracket}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C \\
\mathit{prnt}_{D'}(w) & \text{if } w \in V_r \text{ and } \mathit{prnt}_{D'}(w) \in V_r \\
\phi(p) & \text{if } w \in V_r \text{ and } \mathit{prnt}_{D'}(w) = i_r
\end{cases} \\
&= \begin{cases}
\mathit{prnt}_{D'}(w) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) \in V_{D'} \\
\mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(i) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(i) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(j) & \text{if } w \in V_{D'} \text{ and } \mathit{prnt}_{D'}(w) = i \in |Q'| \\
 & \quad \text{and } \mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(i) = j \in |R| \\
\mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(w) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(w) \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \\
\mathit{prnt}_C(i) & \text{if } w \in \phi'^{\mathsf{v}}(V_{\tilde{P}'}) \text{ and } \mathit{prnt}_{\llbracket \phi' \rrbracket \blacksquare \llbracket \tilde{P}' \rrbracket}(w) = i \in |R| \\
\mathit{prnt}_C(w) & \text{if } w \in V_C
\end{cases}
\end{aligned}
$$

since

$$\begin{aligned}
prnt_{D'} &= ((\mathsf{Id}_{i_r} \uplus [i_r+1 \mapsto i_r, \ldots, |Q| \mapsto |Q|-1])^{-1} \uplus \mathsf{Id}_{a\downarrow^{\mathsf{rng}(\phi^{\mathsf{s}})}}) \circ prnt_D \\
&\quad \uplus ([i_r \mapsto i_q]^{-1} \uplus f_v^{-1}) \circ prnt_D \circ f_v
\end{aligned}$$

$$\begin{aligned}
prnt_r &= [f_v^{-1}(\phi(q)) \mapsto \phi(p)] \\
&\quad \uplus f_v^{-1} \circ prnt_a \circ (f_v - f_v^{-1}(\phi(q))) \\
&= [i_q \mapsto \phi(p)] \circ prnt_D \circ [f_v^{-1}(\phi(q)) \mapsto \phi(q)] \\
&\quad \uplus f_v^{-1} \circ prnt_a \circ (f_v - f_v^{-1}(\phi(q))) \\
&= [\phi(p) \mapsto i_q]^{-1} \circ prnt_D \circ [f_v^{-1}(\phi(q)) \mapsto \phi(q)] \\
&\quad \uplus f_v^{-1} \circ prnt_a \circ (f_v - f_v^{-1}(\phi(q))) \\
&= [\phi(p) \mapsto i_q]^{-1} \circ prnt_D \circ f_v \restriction_{f_v^{-1}(\phi(q))} \\
&\quad \uplus f_v^{-1} \circ prnt_D \circ (f_v - f_v^{-1}(\phi(q))) \\
&= ([\phi(p) \mapsto i_q]^{-1} \uplus f_v^{-1}) \circ prnt_D \circ f_v \\
&= ([i_r \mapsto \phi(p)] \uplus \mathsf{Id}_{V_r}) \circ ([i_r \mapsto i_q]^{-1} \uplus f_v^{-1}) \circ prnt_D \circ f_v \\
&= ([i_r \mapsto \phi(p)] \uplus \mathsf{Id}_{V_r}) \circ prnt_{D'} \restriction_{V_r}
\end{aligned}$$

$$prnt_{[\![\phi']\!] \blacksquare [\![\tilde{P}']\!]} = prnt_{[\![\phi]\!] \blacksquare [\![\tilde{P}]\!]} \circ (\mathsf{Id}_{i_r} \uplus [i_r \mapsto i_q] \uplus [i_r+1 \mapsto i_r, \ldots, |Q| \mapsto |Q|-1] \uplus \mathsf{Id}_{\mathsf{rng}(\phi')}).$$

$\odot_{(v,i)\mapsto l}$: We have $v \in V_{\tilde{P}}$, $i \in ar(ctrl_{\tilde{P}}(v))$, $l \in E_{\tilde{P}} \uplus Y$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}}, link_{\tilde{P}}[(v,i) \mapsto l]) : Q \to \langle R, Y \rangle \\
a' &= (V_a, E_a, ctrl_a, prnt_a, link_a[(\phi(v), i) \mapsto \phi(l)]) \\
\phi' &= \phi \\
inst_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle.
\end{aligned}$$

All the equalities obviously hold.

$\circledcirc_{v:K}$: We have $v \in V_{\tilde{P}}$, $ar(K) = ar(ctrl_{\tilde{P}}(v))$, and

$$\begin{aligned}
\tilde{P}' &= (V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}[v \mapsto K], prnt_{\tilde{P}}, link_{\tilde{P}}) : Q \to \langle R, Y \rangle \\
a' &= (V_a, E_a, ctrl_a[\phi(v) \mapsto K], prnt_a, link_a) \\
\phi' &= \phi \\
inst_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_{\mathcal{U}} \\
D' : \langle |Q'|, X_I \rangle &= D : \langle |Q|, X_I \rangle.
\end{aligned}$$

All the equalities obviously hold.

□

A.2.3 Proof of Lemma 6.15

By Def. 6.4 we have a match of R in a and thus by Corol. 5.23 we have an embedding $\phi : \tilde{P} \hookrightarrow a$ such that

$$\begin{aligned}
a &= ctxt([\![\phi]\!]) \circ ([\![\phi]\!] \blacksquare [\![\tilde{P}]\!] \otimes \mathsf{id}_{X_I}) \circ prmt([\![\phi]\!]) \\
a' &= ctxt([\![\phi]\!]) \circ (\rho' \blacksquare [\![\delta(\tilde{P})]\!] \otimes \mathsf{id}_{X_I}) \circ \overline{[\![inst_Q(\mathit{finst}(\delta))]\!]}(prmt([\![\phi]\!])).
\end{aligned}$$

What remains is to show $(a'', \phi') = \delta(a, \phi)$ and $a' \doteq a''$ for each edit. The difficulty lies in the instantiation and the interesting cases are those that delete or copy a parameter; the others are very similar to each other, and we only show the first one.

$\oplus_{v:K_{\vec{y}}@p}$: We have $v \notin V_{\tilde{P}}$, $p \in V_{\tilde{P}} \uplus R$, $\{\vec{y}\} \subseteq E_{\tilde{P}} \uplus Y$, $ar(K) = n$, and

$$\begin{aligned} a'' &= (V_a + v', E_a, ctrl_a[v' \mapsto K], prnt_a[v' \mapsto \phi(p)], \\ &\qquad link_a[(v',0) \mapsto \phi(\vec{y}_0), \ldots, (v', n-1) \mapsto \phi(\vec{y}_{n-1})]) \\ \phi' &= \phi[v \mapsto v'] \\ inst_Q(finst(\delta)) &= \mathsf{Id}_Q \end{aligned}$$

for some $v' \notin V_a$.

By Corol. 5.23 we have

$$a'' = ctxt([\![\phi']\!]) \circ ([\![\phi']\!] \bullet [\![\delta(\tilde{P})]\!] \otimes \mathsf{id}_{X_I}) \circ prmt([\![\phi']\!]).$$

From Def. 5.20 and $\phi' = \phi[v \mapsto v']$ it is clear that $ctxt([\![\phi']\!]) = ctxt([\![\phi]\!])$ and $prmt([\![\phi']\!]) = prmt([\![\phi]\!])$, so it is sufficient to show

$$\begin{aligned} prmt([\![\phi]\!]) &\doteq \overline{[\![inst_Q(finst(\delta))]\!]}(prmt([\![\phi]\!])) \\ [\![\phi']\!] \bullet [\![\delta(\tilde{P})]\!] &\doteq \rho' \bullet [\![\delta(\tilde{P})]\!]. \end{aligned}$$

where the latter is immediate from the definition of support equivalence. By definition of instantiation (cf. Def. 2.35) and $inst_Q(finst(\delta)) = \mathsf{Id}_Q$ we get

$$\begin{aligned} \overline{[\![inst_Q(finst(\delta))]\!]}(prmt([\![\phi]\!])) &= \overline{[\![\mathsf{Id}_Q]\!]}(prmt([\![\phi]\!])) \\ &= \overline{\mathsf{Id}_{|Q|}}(prmt([\![\phi]\!])) \\ &\doteq prmt([\![\phi]\!]) \end{aligned}$$

as required.

$\oplus_e$: Similar to the first case.

$\ominus_v$: Similar to the first case.

$\ominus_e$: Similar to the first case.

$\ominus_q$: We have $q \in Q$ and (noting that a has no sites)

$$\begin{aligned} a'' &= (V_a \setminus a\downharpoonleft^{\phi(q)}, E_a, ctrl_a - a\downharpoonleft^{\phi(q)}, \\ &\qquad prnt_a - a\downharpoonleft^{\phi(q)}, link_a - P_{a\downharpoonleft^{\phi(q)}}) \\ \phi' &= \phi - q \\ inst_Q(finst(\delta)) &= \mathsf{Id}_{Q-q}. \end{aligned}$$

For simplicity, assume $\forall q' \in Q - q : q > q'$.

By Corol. 5.23 we have

$$a'' = ctxt([\![\phi']\!]) \circ ([\![\phi']\!] \bullet [\![\delta(\tilde{P})]\!] \otimes \mathsf{id}_{X_I}) \circ prmt([\![\phi']\!]).$$

From Def. 5.20 and $\phi' = \phi - q$ it is easy to see that

$$\begin{aligned} ctxt([\![\phi']\!]) &= ctxt([\![\phi]\!]) \\ prmt([\![\phi]\!]) &= d_0 \otimes \cdots \otimes d_{|Q|-1} \\ prmt([\![\phi']\!]) &= d_0 \otimes \cdots \otimes d_{|Q|-2} \end{aligned}$$

so it is sufficient to show

$$\begin{aligned}\mathit{prmt}(\phi') &\doteq \overline{[\![\mathit{inst}_Q(\mathit{finst}(\delta))]\!]}(\mathit{prmt}([\![\phi]\!]))\\ [\![\phi']\!] \centerdot [\![\delta(\tilde{P})]\!] &\doteq \rho' \centerdot [\![\delta(\tilde{P})]\!].\end{aligned}$$

where the latter is immediate from the definition of support equivalence. By definition of instantiation (cf. Def. 2.35) and $\mathit{inst}_Q(\mathit{finst}(\delta)) = \mathsf{Id}_{Q-q}$ we get

$$\begin{aligned}\overline{[\![\mathit{inst}_Q(\mathit{finst}(\delta))]\!]}(\mathit{prmt}([\![\phi]\!])) &= \overline{[\![\mathsf{Id}_{Q-q}]\!]}(\mathit{prmt}([\![\phi]\!]))\\ &= \overline{\mathsf{Id}_{|Q|-2}}(\mathit{prmt}([\![\phi]\!]))\\ &\doteq d_0 \otimes \cdots \otimes d_{|Q|-2}\\ &\doteq \mathit{prmt}([\![\phi']\!])\end{aligned}$$

as required.

$\oslash_{v@p}$: Similar to the first case.

$\oslash_{q@p}$: Similar to the first case.

$\otimes_{q\to r@p}$: We have $q \in Q$, $r \notin Q$, $p \in V_{\tilde{P}} \uplus R$, and (noting that a has no sites)

$$\begin{aligned}a'' &= (V_a \uplus V_r, E_a, \mathit{ctrl}_a \uplus \mathit{ctrl}_r, \mathit{prnt}_a \uplus \mathit{prnt}_r, \mathit{link}_a \uplus \mathit{link}_r)\\ \phi' &= \phi[r \mapsto f_v^{-1}(\phi(q))]\\ \mathit{inst}_Q(\mathit{finst}(\delta)) &= \mathsf{Id}_Q[r \mapsto q]\end{aligned}$$

where

$$\begin{aligned}V_q &= a\downharpoonleft^{\phi(q)}\\ |V_r| &= |V_q|\\ V_r &\# V_a\\ f_v &: V_r \rightarrowtail V_q\\ \mathit{ctrl}_r &= \mathit{ctrl}_a \circ f_v\\ \mathit{prnt}_r &= \{f_v^{-1}(\phi(q)) \mapsto \phi(p)\} \uplus f_v^{-1} \circ \mathit{prnt}_a \circ (f_v - f_v^{-1}(\phi(q)))\\ \mathit{link}_r(v,i) &= \mathit{link}_a(f_v(v),i).\end{aligned}$$

For simplicity, assume $r > q$ and $\forall q' \in Q - q : q > q'$.

By Corol. 5.23 we have

$$a'' = \mathit{ctxt}([\![\phi']\!]) \circ ([\![\phi']\!] \centerdot [\![\delta(\tilde{P})]\!] \otimes \mathsf{id}_{X_I}) \circ \mathit{prmt}([\![\phi']\!]).$$

From Def. 5.20 and $\phi' = \phi[r \mapsto f_v^{-1}(\phi(q))]$ it is easy to see that

$$\begin{aligned}\mathit{ctxt}([\![\phi']\!]) &= \mathit{ctxt}([\![\phi]\!])\\ \mathit{prmt}([\![\phi]\!]) &= d_0 \otimes \cdots \otimes d_{|Q|-1} = (V_D, \emptyset, \mathit{ctrl}_D, \mathit{prnt}_D, \mathit{link}_D)\\ R &= (V_r, \emptyset, \mathit{ctrl}_r, \mathit{prnt}_R, \mathit{link}_R)\\ \mathit{prnt}_R &= \{f_v^{-1}(\phi(q)) \mapsto 0)\} \uplus f_v^{-1} \circ \mathit{prnt}_a \circ (f_v - f_v^{-1}(\phi(q)))\\ \mathit{link}_R(v,i) &= \mathit{link}_D(f_v(v),i)\\ \mathit{prmt}([\![\phi']\!]) &= \mathit{prmt}([\![\phi]\!]) \parallel R\\ d_{|Q|-1} &= (V_q, \emptyset, \mathit{ctrl}_D\restriction_{V_q}, \mathit{prnt}_D\restriction_{V_q}, \mathit{link}_D\restriction_{P_{V_q}})\\ &\doteq R\end{aligned}$$

so it is sufficient to show

$$prmt(\phi') \doteq \overline{[\![inst_Q(\mathit{finst}(\delta))]\!]}(prmt([\![\phi]\!]))$$
$$[\![\phi']\!] \bullet [\![\delta(\tilde{P})]\!] \doteq \rho' \bullet [\![\delta(\tilde{P})]\!].$$

where the latter is immediate from the definition of support equivalence. By definition of instantiation (cf. Def. 2.35), $inst_Q(\mathit{finst}(\delta)) = \mathsf{Id}_Q[r \mapsto q]$, and $d_{|Q|-1} \doteq R$ we get

$$\begin{aligned}\overline{[\![inst_Q(\mathit{finst}(\delta))]\!]}(prmt([\![\phi]\!])) &= \overline{[\![\mathsf{Id}_Q[r \mapsto q]]\!]}(prmt([\![\phi]\!]))\\ &= \overline{\mathsf{Id}_{|Q|-1}[|Q| \mapsto |Q|-1]}(prmt([\![\phi]\!]))\\ &\doteq prmt([\![\phi]\!]) \parallel R\\ &\doteq prmt([\![\phi']\!])\end{aligned}$$

as required.

$\odot_{(v,i)\mapsto l}$**:** Similar to the first case.

$\circledcirc_{v:K}$**:** Similar to the first case.

□

A.2.4 Proof of Prop. 6.28

For each section of the script, we show (1) that it is compatible with the pattern at that point, (2) what the resulting pattern is, and (3) what the resulting forward instance map is. Finally, we show that the named instance map we derive from the forward instance map is η.

$\otimes_{\eta(q'_i)\to f(q'_i)@r}$**:** Compatible since $\eta(q'_i) \in \mathsf{cod}(\eta) = Q$, $f(q'_i) \in Q'' \# Q$, and the redex must have a root if it has sites, since the parent map is acyclic.

The resulting pattern is

$$(V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, prnt_{\tilde{P}} \uplus [f(Q') \mapsto r], link_{\tilde{P}}) : Q \uplus Q'' \to \langle R, Y\rangle.$$

The resulting forward instance map is

$$\begin{aligned}&\mathsf{Id}_{\mathcal{U}-f(q'_{n'})}[\eta(q'_{n'}) \mapsto \{\eta(q'_{n'}), f(q'_{n'})\}]\\ &\quad\circ (\cdots \circ (\mathsf{Id}_{\mathcal{U}-f(q'_1)}[\eta(q'_1) \mapsto \{\eta(q'_1), f(q'_1)\}] \downharpoonleft^{\mathcal{U}-f(q'_2)}) \cdots \downharpoonleft^{\mathcal{U}-f(q'_{n'})})\\ &= \mathsf{Id}_{\mathcal{U}-f(Q')}[\eta(q'_1) \mapsto \{\eta(q'_1), f(q'_1)\}, \ldots, \eta(q'_{n'}) \mapsto \{\eta(q'_{n'}), f(q'_{n'})\}].\end{aligned}$$

$\ominus_{q_i}$**:** Compatible since $q_i \in Q$.

The resulting pattern is

$$(V_{\tilde{P}}, E_{\tilde{P}}, ctrl_{\tilde{P}}, (prnt_{\tilde{P}} \uplus [f(Q') \mapsto r]) - Q, link_{\tilde{P}}) : Q'' \to \langle R, Y\rangle.$$

The resulting forward instance map is

$$\begin{aligned}&\mathsf{Id}_{\mathcal{U}}[q_n \mapsto \emptyset]\\ &\quad\circ (\cdots \circ (\mathsf{Id}_{\mathcal{U}}[q_1 \mapsto \emptyset]\\ &\quad\circ (\mathsf{Id}_{\mathcal{U}-f(Q')}[\eta(q'_1) \mapsto \{\eta(q'_1), f(q'_1)\}, \ldots, \eta(q'_{n'}) \mapsto \{\eta(q'_{n'}), f(q'_{n'})\}] \downharpoonleft^{\mathcal{U}})\\ &\quad\downharpoonleft^{\mathcal{U}}) \cdots \downharpoonleft^{\mathcal{U}})\\ &= \mathsf{Id}_{\mathcal{U}-f(Q')}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset]\\ &\qquad [\eta(q'_1) \mapsto \{f(q'_1)\}, \ldots, \eta(q'_{n'}) \mapsto \{f(q'_{n'})\}].\end{aligned}$$

$\ominus_{v_i}$: Compatible since $v_i \| n V_{\tilde{P}}$ and the nodes has no remaining children since sites are only at root r and for any node v$v_j \in \mathit{prnt}_{\tilde{P}}^{-1}(v_i)$ we have $j < i$ and thus v_j has already been deleted.

The resulting pattern is

$$\begin{aligned}&(\emptyset, E_{\tilde{P}}, \mathit{ctrl}_{\tilde{P}} - V_{\tilde{P}}, ((\mathit{prnt}_{\tilde{P}} \uplus [f(Q') \mapsto r]) - Q) - V_{\tilde{P}}, \mathit{link}_{\tilde{P}} - P_{\tilde{P}}) : Q'' \to \langle R, Y \rangle \\ = {} & (\emptyset, E_{\tilde{P}}, \emptyset, [f(Q') \mapsto r], \emptyset) : Q'' \to \langle R, Y \rangle.\end{aligned}$$

The forward instance map is unchanged.

$\ominus_{e_i}$: Compatible since $e_i \| n E_{\tilde{P}}$ and there are no points left.

The resulting pattern is

$$(\emptyset, \emptyset, \emptyset, [f(Q') \mapsto r], \emptyset) : Q'' \to \langle R, Y \rangle.$$

The forward instance map is unchanged.

$\oplus_{e'_i}$: Compatible since there are no edges left.

The resulting pattern is

$$(\emptyset, E_{\tilde{P}'}, \emptyset, [f(Q') \mapsto r], \emptyset) : Q'' \to \langle R, Y \rangle.$$

The forward instance map is unchanged.

$\oplus_{v'_i : \mathit{ctrl}_{\tilde{P}'}(v'_i)_{[\ldots, \mathit{link}_{\tilde{P}'}(v'_i, i), \ldots]} @ \, \mathit{prnt}_{\tilde{P}'}(v'_i)}$: Compatible since there are no nodes left, the links and roots are all present and if the parent is a node $v'_j = \mathit{prnt}_{\tilde{P}'}(v'_i)$ then we have $j < i$ and thus it has been added before its children.

The resulting pattern is

$$(V_{\tilde{P}'}, E_{\tilde{P}'}, \mathit{ctrl}_{\tilde{P}'}, (\mathit{prnt}_{\tilde{P}'} - Q') \uplus [f(Q') \mapsto r], \mathit{link}_{\tilde{P}'}) : Q'' \to \langle R, Y \rangle.$$

The forward instance map is unchanged.

$\otimes_{f(q'_i) \to q'_i @ \, \mathit{prnt}_{\tilde{P}'}(q'_i)}$: Compatible since $f(q'_i) \in Q''$, $q'_i \in Q' \# Q''$, and $\mathit{prnt}_{\tilde{P}'}(q'_i) \in V_{\tilde{P}'} \uplus R$.

The resulting pattern is

$$(V_{\tilde{P}'}, E_{\tilde{P}'}, \mathit{ctrl}_{\tilde{P}'}, \mathit{prnt}_{\tilde{P}'} \uplus [f(Q') \mapsto r], \mathit{link}_{\tilde{P}'}) : Q' \uplus Q'' \to \langle R, Y \rangle.$$

The resulting forward instance map is

$$\begin{aligned}&\mathsf{Id}_{\mathcal{U} - q'_{n'}}[f(q'_{n'}) \mapsto \{f(q'_{n'}), q'_{n'}\}] \\ &\circ (\cdots \circ (\mathsf{Id}_{\mathcal{U} - q'_1}[f(q'_1) \mapsto \{f(q'_1), q'_1\}] \\ &\circ (\mathsf{Id}_{\mathcal{U} - f(Q')}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\ &\qquad [\eta(q'_1) \mapsto \{f(q'_1)\}, \ldots, \eta(q'_{n'}) \mapsto \{f(q'_{n'})\} \lfloor^{\mathcal{U} - q'_1}) \\ &\lfloor^{\mathcal{U} - q'_2}) \cdots \lfloor^{\mathcal{U} - q'_{n'}}) \\ = {} & \mathsf{Id}_{(\mathcal{U} - f(Q')) - Q'}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\ &\qquad [\eta(q'_1) \mapsto \{f(q'_1), q'_1\}, \ldots, \eta(q'_{n'}) \mapsto \{f(q'_{n'}), q'_{n'}\}].\end{aligned}$$

$\ominus_{f(q'_i)}$**:** Compatible since $f(q'_i) \in Q''$.

The resulting pattern is

$$(V_{\tilde{P}'}, E_{\tilde{P}'}, \mathit{ctrl}_{\tilde{P}'}, \mathit{prnt}_{\tilde{P}'}, \mathit{link}_{\tilde{P}'}) : Q' \to \langle R, Y \rangle = \tilde{P}'.$$

The resulting forward instance map is

$$\begin{aligned}
&\mathit{finst}(\mathit{es}(\mathsf{R})) \\
={}& \mathsf{Id}_{\mathcal{U}}[f(q'_{n'}) \mapsto \emptyset] \\
&\circ (\cdots \circ (\mathsf{Id}_{\mathcal{U}}[f(q'_1) \mapsto \emptyset] \\
&\circ (\mathsf{Id}_{(\mathcal{U}-f(Q'))-Q'}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\
&\qquad [\eta(q'_1) \mapsto \{f(q'_1), q'_1\}, \ldots, \eta(q'_{n'}) \mapsto \{f(q'_{n'}), q'_{n'}\}] \downharpoonleft^{\mathcal{U}}) \\
&\downharpoonleft^{\mathcal{U}}) \cdots \downharpoonleft^{\mathcal{U}}) \\
={}& \mathsf{Id}_{(\mathcal{U}-f(Q'))-Q'}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\
&\qquad [\eta(q'_1) \mapsto \{q'_1\}, \ldots, \eta(q'_{n'}) \mapsto \{q'_{n'}\}]
\end{aligned}$$

The derived named instance map for the entire script is thus:

$$\begin{aligned}
\mathit{inst}_Q(\mathit{finst}(\mathit{es}(\mathsf{R}))) &= (\mathsf{Id}_{(\mathcal{U}-f(Q'))-Q'}[q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\
&\qquad [\eta(q'_1) \mapsto \{q'_1\}, \ldots, \eta(q'_{n'}) \mapsto \{q'_{n'}\}] \upharpoonleft_Q)^{-1} \\
&= ([q_1 \mapsto \emptyset, \ldots, q_n \mapsto \emptyset] \\
&\qquad [\eta(q'_1) \mapsto \{q'_1\}, \ldots, \eta(q'_{n'}) \mapsto \{q'_{n'}\}])^{-1} \\
&= [q'_1 \mapsto \eta(q'_1), \ldots, q'_{n'} \mapsto \eta(q'_{n'})] \\
&= \eta.
\end{aligned}$$

□