Graphs, rewriting and causality in rule-based models

Vincent Danos[§], Jérôme Feret[†], Walter Fontana[¶], Russell Harmer[‡], Jonathan Hayman[†*],
Jean Krivine[‡], Chris Thompson-Walsh[*] and Glynn Winskel[*]
[§] LFCS, School of Informatics, University of Edinburgh, UK
[¶] Harvard Medical School, Boston, Massachusetts
[†] Laboratoire d'informatique de l'École normale suprieure (INRIA/ÉNS/CNRS), Paris, France
[‡] CNRS & Université Paris-Diderot, Paris, France
[*] Computer Laboratory, University of Cambridge, UK

***Abstract*—The Kappa calculus defines how a graph, representing a system of linked agents, can be modified by rules that specify which changes may occur at places that match specific local patterns. Though the calculus has a wide degree of applicability, it has emerged as a natural description of protein-protein interaction systems and pathways in molecular biology [1]. In this paper, we develop an intuitive graph-based semantics for Kappa that correctly handles subtle side-effects upon agent creation or deletion. This yields a single-pushout approach to rewriting based on spans of morphisms. The sequential application of rules gives rise to trajectories that may produce particular patterns of interest that one seeks to account for with a causal history (a pathway). We introduce several notions of trajectory compression, providing a foundation for techniques to reconstruct causal histories at increasing levels of conciseness.**

I. Introduction

The Kappa calculus [2] has emerged as a powerful tool in modelling biochemical systems, supporting sophisticated and efficient simulation [3] and static analysis [4] techniques. It is centered around the concept of *rules* that describe how links between sites on entities called agents are modified when local conditions on the link structure (or, more generally, the state of sites) are satisfied. In biological applications, agents typically represent proteins and links correspond to non-covalent associations between domains (sites) of proteins; rules then are intended to capture empirically sufficient conditions for modifications in the binding (or other) state of protein molecules. The use of rule-based approaches has the potential to make a profound impact in these fields [1], making it possible to analyze systems that would be intractable using traditional ODE-based techniques.

Kappa has an intuitive graphical interpretation. In this paper, formalise the structures involved and abstractly characterize the rewriting operation using a single-pushout technique [5] in which spans of morphisms play a central role. The aim is to develop a natural, stable foundation for Kappa, and the use of morphisms illuminates choices made in earlier work. The techniques provide the possibility of connecting to existing work on graph rewriting, in contrast to earlier definitions of Kappa that involve intricate syntax-based semantics. The single-pushout semantics is shown to be powerful enough to describe side effects in rules, for example correctly deleting links associated with any agent that is deleted even when the links themselves are not mentioned in the rule.

In the second half of the paper, we introduce forms of *trajectory compression*, non-standard causality analyses implemented in the Kappa simulator [6]. These are operations that, when a particular pattern is seen in the graph at some point during simulation (such as two of a particular kind of agent being linked together), determine which rule applications in the trajectory up to that point are relevant. These operations are vital to the provision for the modeller of concise accounts of how particular interesting forms of structure are generated. The forms of compression introduced progressively increase in their ability to remove actions that are irrelevant to the production of the pattern, for example being able to remove pairs of events that remove and subsequently re-introduce structure upon which the pattern depends.

II. The Kappa calculus

Abstractly, we consider systems comprising *agents*, typically representing proteins. Proteins can be viewed at various levels of abstraction, but for modelling pathways it is appropriate to view agents as entities having a number of *sites*. Sites can have *internal properties*, for example being phosphorylated or unphosphorylated or having charge. Sites can be related to each other by undirected *links*, representing for example covalent or non-covalent bonds. We call this collection of agents, sites and links a *mixture*. Beyond this, there is no extra structure, and in particular we do not attempt to model any spatial aspects such as the geometry of bonds or the location of agents within cells.

A rule is equipped with a *test pattern*; if the pattern can be found in the mixture, the rule can be applied to the agents matching the pattern. Application of a rule may involve the creation or destruction of agents or links along with changes in the internal properties of sites. Significantly, the test pattern of a rule does does not need to specify the full state of all agents involved, but only a partial aspect of some agents; this is neatly summed-up as: *"Don't care? Don't write."* However, the application of a rule induces a reaction, *i.e.* transition in which agents are fully specified, in the mixture.

Rules can be equipped with *rate constants*. These are values that influence how likely it is that an occurrence of a rule's test pattern in the mixture is modified as specified in the rule. The application of rules to a mixture is governed by a continuous-time Markov process, which implements standard stochastic

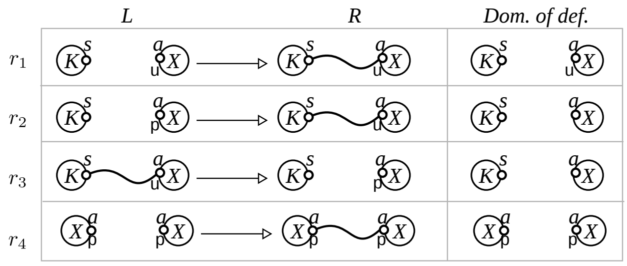


Fig. 1. Example set of rules

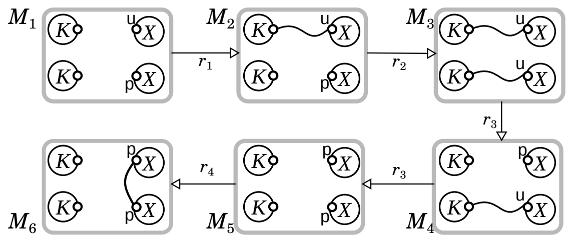
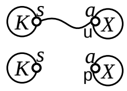


Fig. 2. Example trajectory (site identifiers omitted: a on X and s on K)

chemical kinetics. In this paper, however, we shall not consider any stochastic aspects of the calculus, so we consider a Kappa *system* to consist of an initial mixture and the set of rules that can be applied.

A. Graphical notation

We begin by introducing a simple graphical notation for describing Kappa systems. In a mixture, each agent exhibits all its sites and the properties that hold at them. For example:



Each agent has a *type*; the left agents have type K, in this case representing that they are proteins called kinases, *i.e.* proteins that transfer phosphate groups to specific targets. The small circles on agents represent sites. We label each site with an identifier and indicate, using a sans-serif font, the internal properties that hold at them. For example, the property p holds on the site a on the lower X agent to indicate that it is phosphorylated. We draw lines to connect linked sites.

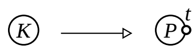
Example rules are presented in Figure 1. Rules have left- and right-hand sides and a 'domain of definition'. The left-hand side specifies the pattern that must be matched for the rule to be applied. Matchings must preserve the types of agents, the presence of sites and the internal properties on sites; they must also preserve both the presence *and absence* of links between given sites. The domain of definition defines the elements (agents, links and properties on sites) of the left-hand side that are not deleted by application of the rule. Anything in the right-hand side but not in the domain of definition is added by application of the rule.

The first rule expresses that a link can be created between a kinase at site s and an agent of type X at site a if a is unphosphorylated. The second rule allows a kinase to link to a phosphorylated site a on an X-agent. When it does so, a becomes viewed as unphosphorylated and so the p property is replaced by u; the label p therefore not drawn in the domain of definition. The third rule expresses that the link between a kinase and an X-agent can be broken, and when this happens the site a becomes phosphorylated. The final rule allows a link to be created between two distinct agents of type X at their a sites, providing the sites are phosphorylated.

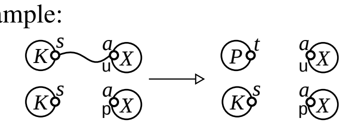
Only the second and third rules can initially be applied to the given mixture. The second rule can apply by matching its left-hand side with the lower kinase and X-agent in the mixture, resulting in creation of a link between s on the kinase and a on the X-agent and a becoming unphosphorylated. We cannot apply the second rule by matching the top agents in the mixture since, in the mixture, there is a link between the agents but, in the pattern, there is not; remember that matchings must preserve both the presence and absence of links. The third rule can apply by matching the top agents in the mixture, resulting in removal of the link between them and the p replacing the u tag on a on the X-agent.

An example sequence of rule applications is drawn in Figure 2, where the mixture considered earlier is encountered after one transition and then the second rule is applied.

The patterns and rules permitted by Kappa are rich, allowing efficient representations of biochemical systems. In particular, rules do not have to fully specify all involved agents. This has the consequence of introducing *side-effects*: changes to agents not referenced by the matching of the pattern into the mixture. For example, we might introduce the following rule that deletes a kinase K and introduces a phosphatase P:

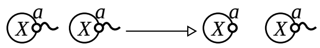


The 'domain of definition' of this rule is empty: nothing matched by left-hand side of the rule will be present following its application. An agent labelled P is present on the right-hand side and is not in the domain of definition, so the rule leads to the creation of an agent of type P. Note that we did not mention the site s on K in the left-hand side of the rule, so we do not care whether there is a link from s on the match of K. We can apply this rule, matching the top-left agent, to the earlier example:

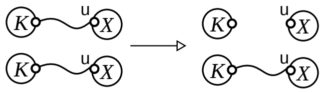


The link between the top agents is lost, so the rule had a side-effect: it affected the top X-agent without it being matched.

Again to allow efficient representation, Kappa allows rules to test for the existence of and delete links just by referencing one side of the link. As an example, consider the following:

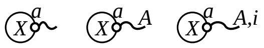


Take the domain of definition to be exactly the right-hand side of the rule. A matching for this rule involves two (distinct) X-type agents, both of which must have links from their site a. Applying the rule results in the link at a on the agent matching the left X being deleted. The other link, from a on the agent matching the right X, is present in the domain of definition and therefore preserved. This rule can be applied to the third mixture in the sequence above to yield the following reaction, matching the left X in the pattern with the top X in the mixture and the right X in the pattern with the bottom X:

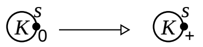


Note the side-effect of removing the link from the kinase. However, some care is necessary. We cannot apply the rule to the final mixture in the sequence in Fig. 2 since there would be an inconsistency: we would be torn between deleting and preserving the link connecting the X-agents in the mixture due to the different treatment of the two links in the rule. We shall exclude these problems by restricting matchings so that these kinds of one-ended links only go to agents outside the image of the matching of the pattern into the mixture, leading us to call them *external* links.

In addition to the links above, which will allow matchings where the link goes to any site on any type of agent, we can also specify that the link should go to any site on some agent of a particular type A, and that the link should go to a particular site i on some agent of a particular type A. In all situations, we require the link to lead outside the image of the matching. The three forms are drawn as follows:



Finally, patterns can test and modify internal properties on sites disregarding whether or not the site is linked. In this case, we draw the site filled to indicate that we are not concerned with link state. We could, for example, add charge information (such as 0 for neutral and + for positive) to the site s on K and introduce the following rule indicating the acquisition of positive charge. Had we drawn the circle for s unfilled, the rule would be applicable only if the site were initially unlinked.



III. Σ-GRAPHS AND MORPHISMS

Formalisation of the account above begins with *signatures*.

Definition 1. *A* signature *is a 4-tuple*

$$\Sigma = (\Sigma_{\mathsf{ag}}, \Sigma_{\mathsf{st}}, \Sigma_{\mathsf{ag-st}}, \Sigma_{\mathsf{prop}}).$$

- Σ_{ag} *is the set of* agent types*,*
- Σ_{st} *is the set of* site identifiers*,*
- $\Sigma_{\mathsf{ag-st}} : \Sigma_{\mathsf{ag}} \to \mathcal{P}_{\mathsf{fin}}(\Sigma_{\mathsf{st}})$ *is the* site map*, and*
- Σ_{prop} *is the set of* internal property identifiers

The set of agent types Σ_{ag} consists of labels to describe the nature of the agents of interest, for example the set of *kinds* of atoms, molecules or proteins to be considered. We use capital letters $A, B, A', \ldots$ to range over agent types. The set of site identifiers Σ_{st} represents the set of labels that can appear to identify sites on agents and is ranged over by $i, j, i', \ldots$. The function $\Sigma_{\mathsf{ag-st}}$ specifies the site identifiers that can be present on agents: any site on an agent of type A must be labelled with an identifier in $\Sigma_{\mathsf{ag-st}}(A)$. Finally, the set Σ_{prop} indicates the set of internal properties that sites might possess. For example, $\{\mathsf{u}, \mathsf{p}\}$ to represent 'unphosphorylated' and 'phosphorylated'.

Graphs with a given signature, *Σ-graphs*, shall play a central rôle in our semantics for the Kappa calculus: they shall include *mixtures*, represent types (*contact graphs*) and represent patterns (*site graphs*).

As discussed in the previous section, there are three forms of external link. The first form, written $-$, indicates just that the link connects to some other site. The second form, written A for some $A \in \Sigma_{\mathsf{ag}}$, indicates that the link connects to some site on some agent of type A. The final form, written (A, i) for some $A \in \Sigma_{\mathsf{ag}}$ and $i \in \Sigma_{\mathsf{ag-st}}(A)$, indicates that the link connects to site i on some agent of type A. The set of all possible external link labels is defined as:

$$\mathsf{Ext} = \{-\} \cup \Sigma_{\mathsf{ag}} \cup \{(A, i) \;:\; A \in \Sigma_{\mathsf{ag}} \;\&\; i \in \Sigma_{\mathsf{ag-st}}(A)\}$$

Definition 2. *A Σ-graph comprises:*

- *a set $\mathcal{A}$ of* agents *(ranged over by $m, n, \ldots$)*
- *an* agent type *assignment* $\mathsf{type} : \mathcal{A} \to \Sigma_{\mathsf{ag}}$
- *a set $\mathcal{S}$ of* sites*,*

$$\mathcal{S} \subseteq \{(n, i) \;:\; n \in \mathcal{A} \;\&\; i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}(n))\}$$

- *a symmetric* link *relation $\mathcal{L}$,*

$$\mathcal{L} \subseteq ((\mathcal{S} \cup \mathsf{Ext}) \times (\mathcal{S} \cup \mathsf{Ext})) \smallsetminus \mathsf{Ext}^2$$

- *a* property *set p_k for each $k \in \Sigma_{\mathsf{prop}}$,*

$$p_k \subseteq \{(n, i) \;:\; n \in \mathcal{A} \;\&\; i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}(n))\}$$

Two points are worth noting. Firstly, any agent has at most one site with any given identifier: an agent n cannot have two sites identified i. Secondly, the set p_k represents the set of sites that have internal property k. It is not in general the case that $p_k \subseteq \mathcal{S}$. The intuition is that we wish the set $\mathcal{S}$ to be the set of sites for which we represent knowledge of *linkage*. In a mixture, this will be all sites, but in a pattern it will be the sites that we require either to be or not to be linked. If we do not care about a site's link state in a pattern, it will not be in $\mathcal{S}$, but we may still wish to test or modify the properties that hold on the site. The sites in $p_k \smallsetminus \mathcal{S}$ are the filled sites referred to in the previous section, whereas sites in $\mathcal{S}$ are drawn hollow.

It is useful to introduce a notational convention that, for a Σ-graph G, we write $\mathcal{A}_G$ for its set of agents, type_G for its typing function, $\mathcal{S}_G$ for its set of link sites, $\mathcal{L}_G$ for its link relation and $p_{k,G}$ for the set of sites satisfying property k.

A. Homomorphisms

Homomorphisms between Σ-graphs are structure-preserving functions from the agents of one Σ-graph to the agents of another. They preserve structure in the sense of preserving the presence of sites, preserving links between sites and preserving properties held on sites. Central to their definition is the *link information order*. Given a typing function type, it is the least reflexive, transitive relation $\leq_{\mathsf{type}}$ s.t. for all $A \in \Sigma_{\mathsf{ag}}$ and $i \in \Sigma_{\mathsf{ag-st}}(A)$ and n s.t. $\mathsf{type}_G(n) = A$:

$$- \leq_{\mathsf{type}} A \leq_{\mathsf{type}} (A, i) \leq_{\mathsf{type}} (n, i)$$

Definition 3. *A* homomorphism *of* Σ*-graphs*

$$h : G \to H$$

is a (total) function on agents $h : \mathcal{A}_G \to \mathcal{A}_H$ *satisfying:*

- $\mathsf{type}_G(n) = \mathsf{type}_H(h(n))$ *for all* $n \in \mathcal{A}_G$
- *if* $(n,i) \in \mathcal{S}_G$ *then* $(h(n),i) \in \mathcal{S}_H$
- $\{(h(n),i) \ : \ (n,i) \in p_{G,k}\} \subseteq p_{H,k}$ *for all* $k \in \Sigma_{\mathsf{prop}}$
- h *respects link structure:* *if* $((n,i),x) \in \mathcal{L}_G$ *then there exists* y *s.t.* $\hat{h}(x) \leq_{\mathsf{type}_H} y$ *and* $((h(n),i),y) \in \mathcal{L}_H$*, where*

$$\hat{h}(x) = \begin{cases} (h(n),i) & \textit{if } x = (n,i) \\ x & \textit{otherwise, i.e. if } x \in \mathsf{Ext} \end{cases}$$

We write Σ-$\mathbf{Graph}$ for the category of Σ-graphs and homomorphisms.[1]

B. Site graphs and mixtures

Mixtures and *site graphs* are special kinds of Σ-graph. Mixtures represent the state to which rules are applied, and site graphs are patterns that mixtures might match via matchings. In particular, site graphs have no links to and from the same site on the same agent and have at most one link from any site. Mixtures additionally have all sites described by the signature on agents and have no external links.

Definition 4. *A (*Σ*-)site graph is a* Σ*-graph such that*

- *its symmetric link relation* $\mathcal{L}$ *is* irreflexive *and satisfies*

$$(x,y) \in \mathcal{L} \ \& \ (x,y') \in \mathcal{L} \implies y = y'.$$

A site graph is a mixture when:

- $\mathcal{S} = \{(n,i) \ : \ n \in \mathcal{A} \ \& \ i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}(A))\}$*, and*
- $\mathcal{L} \subseteq \mathcal{S} \times \mathcal{S}$

It should be noted that Kappa as presented here is slightly different from past presentations in that we allow a set of properties to hold at a site in a mixture instead of precisely one. This generalisation streamlines the coming account of rewriting, though the full version of this paper shall deal with both interpretations.

C. Pullbacks

The category Σ-$\mathbf{Graph}$ has pullbacks. Suppose homomorphisms $g : G \to J$ and $h : H \to J$. The canonical choice of their pullback is constructed by taking, as agents, pairs of agents of G and H that have the same image under g and h. Sites on the pairs must occur on both of the originating agents, and links are added containing as much information as possible whilst ensuring that the projections from the pullback into G and H preserve the link information order. The full construction is given in Appendix A.

[1] An artefact of taking homomorphisms to be simply functions on agents instead of on agents, sites, links and properties as in [7] is that any Σ-graph is isomorphic to that with its link relation expanded with elements lower in the link information order. For the present work, where we progress rapidly to site graphs, there is little difference between two definitions, so we retain the simpler definition.

Proposition 1. *The category* Σ-$\mathbf{Graph}$ *has pullbacks. The pullback of a morphism from a site graph against a morphism from a site graph is a site graph.*

IV. Morphisms for rewriting

We now turn to the use of homomorphisms in relating site graphs and mixtures before and after the application of rules. As they currently stand, homomorphisms lack the ability to represent the deletion of agents, so we must introduce *partial* morphisms.

Definition 5. *A partial morphism* $h : G \rightharpoonup H$ *between* Σ*-graphs* G *and* H *is a span* $G \hookleftarrow D \xrightarrow{h} H$ *where:*

- h *is a homomorphism and* D *is a* Σ*-graph*
- D *is a subgraph of* G*, i.e.* $\mathcal{A}_D \subseteq \mathcal{A}_G$ *and* $\mathcal{S}_D \subseteq \mathcal{S}_G$ *and* $\mathcal{L}_D \subseteq \mathcal{L}_G$ *and* $p_{D,k} \subseteq p_{G,k}$ *for all* $k \in \Sigma_{\mathsf{prop}}$.

Partial morphisms are composed by pullback in Σ-$\mathbf{Graph}$:

$$\begin{tikzcd}
 & & h^{-1}D' \arrow[dl, hook'] \arrow[dr] \arrow[dd, phantom, "\vee"] & & \\
 & D \arrow[dl, hook'] \arrow[dr, "h"] & & D' \arrow[dl, hook'] \arrow[dr, "k"] & \\
G & & H & & K
\end{tikzcd}$$

$h^{-1}D'$ is the inverse image of h applied to D', comprising elements of D that under h hit agents/sites/links of D'.

Write Σ-$\mathbf{Graph}_*$ for the category of Σ-graphs and partial morphisms and write Σ-$\mathbf{Site}_*$ for the category of Σ-site graphs and partial morphisms.

We will write $\mathrm{dom}(h)$ for the Σ-graph representing the domain of definition of a partial morphism h, and allow ourselves to write $n \in \mathrm{dom}(h)$ to mean that n is in its set of agents, $(n,i) \in \mathrm{dom}(h)$ to mean that (n,i) is in its set of link sites, and so on. We additionally write $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\alpha)$ if (n,i) is in its set representing the property k.

A. Matchings

Matchings will be used to express how patterns (site graphs) are found into mixtures. There are three key aspects to how patterns are applied. Firstly, distinct agents in patterns must match distinct agents in the mixture. Secondly, sites with no link in the pattern should have no link in the mixture, allowing patterns to express the absence of links on sites. Finally, as discussed in Section II-A, the image of an external link in the pattern cannot connect to any agent in the image of the pattern. We write $e : G \rightarrowtail H$ to signify that e is a matching.

Definition 6. *A* matching *of the site graph* G *in* H *is a homomorphism* $e : G \to H$ *such that*

- e *is injective,*
- *for all* $(n,i) \in \mathcal{S}_G$*, if there exists* y *s.t.* $((e(n),i),y) \in \mathcal{L}_H$ *then there exists* x *such that* $((n,i),x) \in \mathcal{L}_G$*, and*
- *for all* $(n,i) \in \mathcal{S}_G$*, if there exist* m *and* j *s.t.* $((e(n),i),(e(m),j)) \in \mathcal{L}_H$ *then* $((n,i),(m,j)) \in \mathcal{L}_G$.

B. Action maps

Every rule will be associated with a single *action map*, a special kind of partial morphism, to describe its effect. Viewing action maps as spans, the domain of definition will describe what is tested from the left-hand side of the rule. Anything in the left-hand side not in the domain of definition is to be deleted. The right leg of the span shall be injective, so rules cannot 'merge' agents together. Anything in the right-hand side not in the image of the domain of definition is created by the rule.

Action maps have two important aspects. Firstly, we wish it always to be the case that if the pattern of the left-hand side of the rule matches part of the mixture, following application of the rule to that part, it matches the right-hand side of the rule. Secondly, we wish to ensure that action maps preserve the property of being a mixture. Property (2) below ensures that sites are not added to or deleted from existing agents: we might otherwise introduce a site onto a preserved agent that failed to match due to the presence/absence of links, or fail to generate a mixture by deleting a site from a preserved agent. Property (3) ensures that any preserved external link is not promoted up the link information order; this is again to ensure that the right-hand side of the pattern will be consistent with the obtained mixture. Property (4) ensures that any link that is created is not an external link. Property (5) ensures that if an agent is created then so are all the sites consistent with the signature.

Definition 7. *An* action map $\alpha : G \rightharpoonup H$ *is a partial morphism of Σ-site graphs $\alpha : G \rightharpoonup H$ s.t.*

1) *α is partial injective, i.e. for all $m, n \in \mathrm{dom}(\alpha)$, if $\alpha(n) = \alpha(m)$ then $n = m$*
2) *if $n \in \mathrm{dom}(\alpha)$ then $(n,i) \in \mathcal{S}_G$ iff $(\alpha(n),i) \in \mathcal{S}_H$ iff $(n,i) \in \mathrm{dom}(\alpha)$.*
3) *if $((m,i),x) \in \mathcal{L}_G$ and $x \in \mathsf{Ext}$ and $((m,i),x) \in \mathrm{dom}(\alpha)$ then $((\alpha(m),i),x) \in \mathcal{L}_H$*
4) *if $((m,i),x) \in \mathcal{L}_H$ and $\nexists n,y$ s.t. $\alpha(n) = m$ and $((n,i),y) \in \mathrm{dom}(\alpha)$ then there exists $(p,j) \in \mathcal{S}_H$ s.t. $x = (p,j)$.*
5) *if $m \in \mathcal{A}_H$ and $m \notin \mathrm{image}(\alpha)$ then*

$$\forall i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}(m)) : (m,i) \in \mathcal{S}_H$$

To within isomorphism of H, we can always take α such that $\alpha(m) = n$ implies $m = n$, so preserving the identity of tested agents.

V. Rewriting

We begin this section by describing concretely the effect of applying a rule to a mixture, precisely spelling out the earlier account of actions. We then neatly characterize this abstractly as a (single) pushout.

Suppose an action map $\alpha : L \rightharpoonup R$ and a matching $e : L \rightarrowtail M$, for some mixture M. Firstly, the application of the rule can cause agents and links to be deleted and properties no longer to hold of sites. These deleted elements form the sets:

$$\begin{aligned}
\langle e,\alpha\rangle^-_{\mathsf{ag}} &= \{m \in \mathcal{A}_M \ : \ \exists n \in \mathcal{A}_L . e(n) = m \ \& \\
&\qquad\qquad n \notin \mathrm{dom}(\alpha)\} \\
\langle e,\alpha\rangle^-_{\mathsf{lnk}} &= \{((m,i),x),(x,(m,i)) \in \mathcal{L}_M \ : \ m \in \langle e,\alpha\rangle^-_{\mathsf{ag}} \text{ or} \\
&\qquad \exists n,y.e(n) = m \ \& \ ((n,i),y) \in \mathcal{L}_L \setminus \mathrm{dom}(\alpha)\} \\
\langle e,\alpha\rangle^-_k &= \{(m,i) \in p_{M,k} \ : \ m \in \langle e,\alpha\rangle^-_{\mathsf{ag}} \text{ or} \\
&\qquad \exists n.e(n) = m \ \& \ (n,i) \in p_{L,k} \ \& \\
&\qquad\qquad (n,i) \notin \mathrm{dom}_{\mathsf{prop},k}(\alpha)\}
\end{aligned}$$

Agents and links can also be created by rules, and rules can cause internal properties on sites to hold. A slight subtlety is that we cannot view an action map as specifying the exact identity of newly-created agents; they only talk about some number of agents with particular structure being created. For this reason, when specifying the set of agents created, we assume a bijection $\phi : \mathcal{A}_R \setminus \mathrm{image}(\alpha) \to X$ where X, the set of agents to be created, is any set disjoint from $\mathcal{A}_M \setminus \langle e,\alpha\rangle^-_{\mathsf{ag}}$. Given this bijection, define

$$u(n) = \begin{cases} e(m) & \text{if } \alpha(m) = n \\ \phi(n) & \text{if } n \notin \mathrm{image}(\alpha) \end{cases}$$

This is well-defined since α is partially injective. We now define the elements to be added to M by the rule:

$$\begin{aligned}
\langle \alpha,\phi\rangle^+_{\mathsf{ag}} &= X \\
\langle \alpha,\phi\rangle^+_{\mathsf{lnk}} &= \{((u(n),i),\hat{u}(x)) \ : \\
&\qquad ((n,i),x) \in \mathcal{L}_R \ \& \ \nexists y,m : \\
&\qquad \alpha(m) = n \ \& \ ((m,i),y) \in \mathrm{dom}(\alpha)\} \\
\langle \alpha,\phi\rangle^+_k &= \{(\phi(m),i) \ : \ (m,i) \in p_{k,R} \ \& \ \nexists n : \alpha(n) = m \\
&\qquad\qquad \& \ (n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\alpha)\}
\end{aligned}$$

We first construct a domain D' to represent the part of M that will be unchanged by the application of the rule. This is just M after the removal of the specified agents and links.

$$\begin{aligned}
&\mathcal{A}_{D'} = \mathcal{A}_M \setminus \langle e,\alpha\rangle^-_{\mathsf{ag}} \qquad \mathsf{type}_{D'}(m) = \mathsf{type}_M(m) \\
&\mathcal{S}_{D'} = \{(m,i) \ : \ m \in \mathcal{A}_{D'} \ \& \ i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}_{D'}(m))\} \\
&\mathcal{L}_{D'} = \mathcal{L}_M \setminus \langle e,\alpha\rangle^-_{\mathsf{lnk}} \qquad p_{D',k} = p_{M,k} \setminus \langle e,\alpha\rangle^-_k
\end{aligned}$$

It is straightforward to check that D' is a mixture. We now construct the result N by adding to D' the specified agents and links.

$$\begin{aligned}
\mathcal{A}_N &= \mathcal{A}_{D'} \cup \langle \alpha,\phi\rangle^+_{\mathsf{ag}} \\
\mathsf{type}_{D'}(m) &= \begin{cases} \mathsf{type}_{D'}(m) & \text{if } m \in \mathcal{A}_{D'} \\ \mathsf{type}_R(\phi^{-1}(m)) & \text{if } m \in \langle \alpha,\phi\rangle^+_{\mathsf{ag}} \end{cases} \\
\mathcal{S}_N &= \{(m,i) \ : \ m \in \mathcal{A}_N \ \& \ i \in \Sigma_{\mathsf{ag-st}}(\mathsf{type}_{D'}(m))\} \\
\mathcal{L}_N &= \mathcal{L}_{D'} \cup \langle \alpha,\phi\rangle^+_{\mathsf{lnk}} \\
p_{N,k} &= p_{D',k} \cup \langle \alpha,\phi\rangle^+_k
\end{aligned}$$

We now give the first key result, characterizing the mixture as a pushout. The result follows by a detailed analysis and is dependent on the earlier careful definitions of matchings and action maps. Its proof is presented in Appendix C.

Theorem 1. *Given an action $\alpha : L \rightharpoonup R$ and a matching $e : L \rightarrowtail M$, define $\beta_0(m) = m$ for any $m \in \mathcal{A}_{D'}$. The constructions above yield a pushout in the category Σ-$\mathbf{Site}_*$*

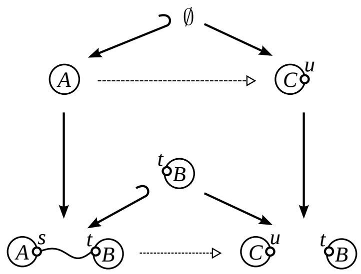
$$\begin{tikzcd} L \arrow[r, "\alpha"] \arrow[d, "e"'] & R \arrow[d, "u"] \\ M \arrow[r, "\beta"'] & N \end{tikzcd} \quad \text{where} \quad \begin{tikzcd} & D' \arrow[dl, hook'] \arrow[dr, "\beta_0"] & \\ M \arrow[rr, "\beta"'] & & N \end{tikzcd}$$

Furthermore, N is a mixture, u is a matching and β is an action map.

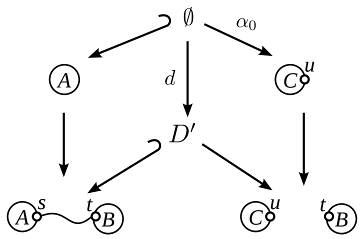
A. Examples

We saw a number of example rule applications in Section II, which can straightforwardly be verified to match the definition of the pushout above.

One of the key aspects of the single-pushout based approach to rule application is that it allows rules to have side-effects. For example it allows the following pushout where an agent of type A is deleted, having the side-effect of dropping the link connecting A and B in the mixture, and an agent of type C is created.



This example distinguishes our framework from double-pushout rewriting, as was studied for a restricted form of Kappa in [7]. Double-pushout rewriting would apply a rule $\alpha : L \rightharpoonup R$ with domain D and right leg $\alpha_0 : D \to R$ to a mixture M to yield M' if there were a morphism $d : D \to D'$ such that M is the pushout of $D \hookrightarrow L$ against d and M' is the pushout of $\alpha_0 : D \to R$ against d. Clearly, there is no hope of finding D' that would cause the following two squares to be pushouts, demonstrating that double-pushout rewriting lacks the generality needed here.



VI. Trajectory compression

The pushout theorem described in the previous section underpins the use of Kappa to simulate biochemical systems. Given an initial mixture and a set of rules described by action maps, the simulation rewrites mixtures to obtain a *trajectory*: a finite sequence of mixtures connected by action maps:

$$M_1 \xrightarrow{\beta_1} M_2 \cdots\cdots M_n \xrightarrow{\beta_n} M_{n+1}$$

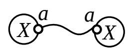
Each step in the sequence is justified by a pushout

$$\begin{tikzcd} M_i \arrow[r, "\beta_i"] & M_{i+1} \\ L_i \arrow[u, "e_i"] \arrow[r, "\alpha_i"'] & R_i \arrow[u, "u_i"'] \end{tikzcd}$$

in which each action map $\alpha_i : L_i \rightharpoonup R_i$ is associated with some rule r_i.

One of the key practical uses of a trajectory is to investigate how particular *observable* entities come to exist. An observable is simply a site graph (a pattern), for example a particular form of protein complex. The simulation is run until the observable is seen to exist, and then the user is presented with an account of what actions in the trajectory were necessary for the observable to occur. This account of how the observable entity is created is called its *story*. Of course, in general the trajectory obtained by simulation will contain actions that are unnecessary for the production of the observable, so it is necessary to obtain as concise an account as possible. This process is called *compression*.

For example, when studying the trajectory presented in Figure 2, we might be interested in the production of links between X agents. As the observable, we would choose the site graph:



We might wish to compress the given trajectory with respect to this observable by removing the actions that create and then remove the link between the lower kinase and X-agent since their combined effect is nil.

The ultimate goal of compression is to obtain a trajectory that is as short as possible showing the relevant parts of the given trajectory in the production of the observable. As we shall see, there is a choice to be made in deciding upon which actions an observable depends, leading to various levels of fidelity in the obtained account: *Mazurkiewicz compression* giving the most detail, then *weak compression* and finally *strong compression*. Weak and strong compression are practically-useful, and have been implemented using constraint programming in the Kappa simulator [6].

When studying how an observable comes to arise, it is convenient to view the observable as an *action*, namely the action that only tests whether the site graph is matched. The question then becomes: what sequence of actions are necessary for the occurrence of the observable action? This leads to a first attempt to capture the idea of compression by connecting to a notion familiar from traditional independence models: the ability to permute *independent* events.

A. Mazurkiewicz compression

Definition 8. *A Mazurkiewicz trace language is a tuple (S, E, I) where E is a set of* events, *S is a prefix-closed set of (finite)* sequences *of events and $I \subseteq E \times E$ is a symmetric, binary relation of* independence. *The set S is required to be closed under permutation of consecutive independent events: if $\pi e_1 e_2 \pi' \in S$ and $e_1 I e_2$ then $\pi e_2 e_1 \pi' \in S$.*

In this section on Mazurkiewicz compression only, for simplicity we restrict attention to Kappa without the creation or deletion of agents and without external links in rules; all that can take place is the creation and deletion of links and internal properties on sites. As shall be explained, Mazurkiewicz compression of a trajectory for an observable shall be based on performing permutations of events in a given trace to make a particular event occur as early as possible.

We now describe the Mazurkiewicz trace language formed when a set of Kappa rules is applied to an initial mixture M.

Intuitively, an event will record which rule is applied and the *graph* of the embedding of its left-hand side into the mixture to which it is applied. Formally, they are pairs $v = (r, \epsilon)$ where r is a rule and ϵ is a bijection $\epsilon : \mathcal{A}_L \cong \mathcal{A}_v$ between the agents $\mathcal{A}_L$ on the left-hand side of the rule and some set $\mathcal{A}_v$. For any function f, let $|f|$ denote its graph. The set S has sequences of the form $((r_1, |e_1|), (r_2, |e_2|), \ldots, (r_n, |e_n|))$ for any trajectory as drawn at the beginning of Section VI.

Two events in a trace are independent if they overlap only on their constant parts. Formally, for a set of links L, define

$$\epsilon(L) = \{((\epsilon(n), i), ((\epsilon(m), j)) \ : \ ((m,i),(n,j)) \in L\}$$

and for a set of agent-site identifier pairs p define $\epsilon(p) = \{(\epsilon(n), i) \ : \ (n,i) \in p\}$. Let the rules r_1 and r_2 have action maps $\alpha_1 : L_1 \rightharpoonup R_1$ and $\alpha_2 : L_2 \rightharpoonup R_2$. Define $(r_1, \epsilon_1) I (r_2, \epsilon_2)$ iff

- $\epsilon_1(\mathcal{L}_{L_1}) \cap \epsilon_2(\mathcal{L}_{L_2}) \subseteq \epsilon_1(\mathcal{L}_{D_1}) \cap \epsilon_2(\mathcal{L}_{D_2})$, and
- $\epsilon_1(p_{L_1,k}) \cap \epsilon_2(p_{L_2,k}) \subseteq \epsilon_1(p_{D_1,k}) \cap \epsilon_2(p_{D_2,k})$ for all $k \in \Sigma_{\mathsf{prop}}$.

It is straightforward to verify that a Mazurkiewicz trace language is formed:

Lemma 1. *For the given Mazurkiewicz trace language generated from a set of rules and an initial mixture, if $v_1 I v_2$ and $\pi v_1 v_2 \pi' \in S$ then $\pi v_2 v_1 \pi' \in S$.*

Note the importance of recording only the graph of the embedding rather than the embedding itself. This 'diamond property' gives rise to an equivalence on elements of S, viewing them up-to the permutation of consecutive independent events. Formally, we take the relation $\sim \subseteq S \times S$ to be the reflexive, transitive closure of the relation $\sim_0 \subseteq S \times S$ which relates two sequences of events iff they are equal apart from the permutation of two consecutive independent events:

$$\begin{aligned} \pi \sim_0 \pi' \quad \iff \quad & \exists \pi_1, \pi_2, v_1, v_2 \text{ s.t. } v_1 I v_2 \\ & \& \ \pi = \pi_1 v_1 v_2 \pi_2 \ \& \ \pi' = \pi_1 v_2 v_1 \pi_2 \end{aligned}$$

Modulo this equivalence, we define trace inclusion as:

$$\pi \lesssim \pi' \iff \exists \pi_0 \ : \ \pi\pi_0 \sim \pi'$$

Definition 9 (Mazurkiewicz maximally compressed)**.** *The event v is* maximally compressed *in a trace $\pi_1 v \pi_2$ if, for any π_1', π_2' such that $\pi_1 v \pi_2 \sim \pi_1' v \pi_2'$, we have $\pi_1 \lesssim \pi_1'$.*

It is a standard property of partial-order models for concurrency [8] that there is a unique minimal history giving rise to any given event in a trace:

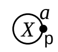
Proposition 2. *For any event v in a trace $\pi_1 v \pi_2$, there exist π_1', π_2' such that $\pi_1 v \pi_2 \sim \pi_1' v \pi_2'$ and v is maximally compressed in $\pi_1' v \pi_2'$. Furthermore, for any $\pi_1'' v \pi_2'' \sim \pi_1 v \pi_2$ in which v is maximally compressed, we have $\pi_1' \sim \pi_1''$ and $\pi_2' \sim \pi_2''$.*

This trace π_1' is the *maximal (Mazurkiewicz) compression* of v. Given the original trajectory, we form it by moving as many events as possible to occur after v by permuting independent actions; the event v causally depends on all the events in π_1'.

B. Examples

As an example, recall the trajectory in Figure 2. Let its associated sequence be $(v_1 v_2 v_3 v_4 v_5)$, with, for example, $v_1 = (r_1, \epsilon_1)$ where ϵ_1 is the graph of the matching that takes the X-agent in the rule to the top X-agent in the mixture and the kinase in the rule to the top kinase in the mixture. We have $v_1 I v_2$, $v_1 I v_4$, $v_2 I v_3$ and $v_3 I v_4$.

Suppose that we are interested in an observable with the following pattern:

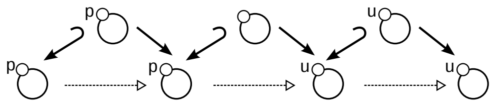


We can consider this pattern to be a rule r_{obs} with the identity on the pattern as its action map. This rule can be applied if there is an X-type agent with a phosphorylated site a which may or may not be linked to some other agent, so we have a trajectory $(v_1 v_2 v_3 v_4 w)$ where w is an event representing an occurrence of the observation: $w = (r_{obs}, \{x \mapsto x_1\})$ where x is the agent in the pattern and x_1 is the top X-agent in M_5. We wish to compress this trajectory to remove any events unnecessary for the observable w. Firstly, since $v_4 I w$, we can apply Lemma 1 to infer that $(v_1 v_2 v_3 w v_4)$ is a trace. Furthermore, as summarized in the following diagram, since $v_2 I v_3$ there is a trace $(v_1 v_3 v_2 w v_4)$, and since $v_2 I w$ there is a trace $(v_1 v_3 w v_2 v_4)$:

$$\begin{tikzcd}
{} \arrow[r, "v_1"] & {} \arrow[r, "v_3"] \arrow[d, "v_2"'] \arrow[dr, phantom, "I"] & {} \arrow[r, "w"] \arrow[d, "v_2"] \arrow[dr, phantom, "I"] & {} \arrow[d, "v_2"] \\
& {} \arrow[r, "v_3"'] & {} \arrow[r, "w"] \arrow[d, "v_4"'] \arrow[dr, phantom, "I"] & {} \arrow[d, "v_4"] \\
& & {} \arrow[r, "w"] & {}
\end{tikzcd}$$

The lemma cannot be applied to force the events v_1 or v_3 to occur after w, so the trace $(v_1 v_3 w)$ is maximally-compressed with respect to w. This reflects the fact that the phosphorylation of x_1 only depends on the events where the kinase links to x_1 and then subsequently phosphorylates the site a.

There are two key areas where Mazurkiewicz compression fails to compress trajectories in a practically-useful way. One class of examples arises from trajectories where the presence of structure is tested, then deleted and then an observable depending on the deletion is seen; Mazurkiewicz compression cannot remove the unnecessary testing actions. Consider a system which has three rules, one testing whether a site is phosphorylated, another changing it from phosphorylated to unphosphorylated and a third rule (representing an observable) that tests whether it is unphosphorylated. We have the following trajectory (omitting as many labels as possible):



Considering the corresponding sequence of events, since the first two events are not independent, the first event that needlessly tests for the phosphorylation of the site will be in the maximal compression of the final event. A solution to this might be to introduce an *asymmetric* independence relation.

A more significant weakness is that Mazurkiewicz compression only allows the pairwise comparison of events. Suppose, for example, that we are interested in an observable expressing that there is no link between a pair of sites. In a trajectory where such a link is created, deleted, and then finally the observable is seen, we would hope to compress the story down to just the observable initially holding. However, the causal dependence of the observable on the deletion, which in turn depends on the creation, precludes any form of compression.

C. Simple compression

We now introduce *simple* compression, returning to the full Kappa calculus with external links and the creation and deletion of agents. Simple compression bypasses both of the issues with Mazurkiewicz compression by determining directly whether a given rule can be applied at an earlier point in the trajectory. It requires that the action map representing the *composition* of the intermediate actions demonstrates that the part of the mixture that is tested by the rule is preserved by the intermediate actions.

Suppose that we have an action map $\beta_1 : M_1 \rightarrow M_2$ and there is a rule r_2 with action map $\alpha_2 : L_2 \rightarrow R_2$ with a matching $e_2 : L_2 \rightarrowtail M_2$. The application of r_2 to M_2 can be *simply compressed* to occur prior to the action β_1 if there is a matching $e_2' : L_2 \rightarrowtail M_1$ such that $\beta_1 \circ e_2' = e_2$. This implies that everything required for the application of r_2 to the same agents (tracking their identity through β_1), links and properties was present in M_1 and that β_1 did not modify them.

Lemma 2. *Let $\beta_1 : M_1 \rightarrow M_2$ and $\alpha_2 : L_2 \rightarrow R_2$ and there be a matching $e_2 : L_2 \rightarrowtail M_1$. Let M_3 be a pushout of α_2 against e_2 with pushout morphisms $\beta_2 : M_2 \rightarrow M_3$ and $u_2 : R_2 \rightarrowtail M_3$. If there is a matching $e_2' : L_2 \rightarrowtail M_1$ such that $\beta_1 \circ e_2' = e_2$ then, letting M_2' be the pushout of α_2 against e_2' with pushout morphisms $\beta_2' : M_1 \rightarrow M_2'$ and $u_2' : R_2 \rightarrowtail M_2'$, there is a unique morphism $\gamma_2 : M_2' \rightarrow M_3$ such that $\gamma_2 \circ \beta_2' = \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' = u_2$. Furthermore, γ_2 is an action map.*

The situation can be summarized as follows, drawing the pushouts M_2' and M_3 described in Theorem 1. Note the presence of an action map $\gamma_2 : M_2' \rightarrow M_3$ making the upper square and surrounding triangle commute; this map exists since M_2' is a pushout and is easily shown to be an action.

$$
\begin{tikzcd}
M_2' \arrow[rr, dotted, "\gamma_2"] & & M_3 \\
M_1 \arrow[u, "\beta_2'"] \arrow[rr, "\beta_1"] \arrow[urr, phantom, "="] & & M_2 \arrow[u, "\beta_2"'] \\
 & L_2 \arrow[ul, "e_2'"] \arrow[ur, "e_2"'] \arrow[d, "\alpha_2"] \arrow[u, phantom, "="] & \\
 & R_2 \arrow[uuul, bend left, "u_2'"] \arrow[uuur, bend right, "u_2"'] &
\end{tikzcd}
$$

The map γ_2 might not be derivable as a trajectory of rules in the given Kappa system. It describes the *residual* of the compression: an action summarizing the effect of the intermediate actions on the mixture formed by moving the application of r_2 forward, excluding the modified part.

The result above can be applied to move a single rule application forward in a trajectory. Given a trajectory $M_1 \xrightarrow{\beta_1} \cdots \xrightarrow{\beta_n} M_{n+1}$ derived from a sequence of rule applications as drawn at the beginning of Section VI, for $i < j$, the action β_j can be moved forward to occur in marking M_i if there is a matching $e_j' : L_j \rightarrowtail M_i$ such that $\beta_{j-1} \circ \cdots \circ \beta_i \circ e_j' = e_j$, in which case we obtain a residual γ_j as follows:

$$
\begin{tikzcd}
 & & M_j' \arrow[rrr, "\gamma_j"] & & & M_{j+1} \arrow[r, "\beta_{j+1}"] & \cdots M_n \\
M_1 \arrow[r, "\beta_1"] & \cdots & M_i \arrow[u, "\beta_j'"] \arrow[r, "\beta_i"] & \cdots & \arrow[r, "\beta_{j-1}"] & M_j \arrow[u, "\beta_j"'] &
\end{tikzcd}
$$

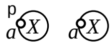
Using the residual, it may be necessary to compress a number of prior rule applications to obtain a maximally compressed trajectory for a given observable: we push towards the beginning of the trajectory, in sequence, the actions upon which the rule application that we are compressing against depends.

Suppose that we wish to compress with respect to the final transition $M_n \xrightarrow{\beta_n} M_{n+1}$. A set of indices must be selected $\{i, j, k, \ldots, m\}$ (written in ascending order) for which there is a sequence of compression steps:

$$
\begin{tikzcd}
M_n' \arrow[rrrr, "\gamma_n"] & & & & M_{n+1} \\
M_m' \arrow[u, "\beta_n'"] \arrow[rrr, "\gamma_m"] & & & M_{m+1} \arrow[r, "\beta n - 1 \circ \ldots \circ \beta_{m+1}"] & M_n \arrow[u, "\beta_n"'] \\
M_j' \arrow[u, dotted, no head] \arrow[rr, "\gamma_j"] & & M_{j+1} \arrow[r, "\beta_{k-1} \circ \ldots \circ \beta_{j+1}"] & M_k \arrow[u, dotted, no head] & \\
M_i' \arrow[u, "\beta_j'"] \arrow[r, "\gamma_i"] & M_{i+1} \arrow[r, "\beta_{j-1} \circ \ldots \circ \beta_{i+1}"] & M_j \arrow[u, "\beta_j"'] & & \\
M_1 \arrow[u, "\beta_i'"] \arrow[r, "\beta_{i-1} \circ \ldots \circ \beta_1"] & M_i \arrow[u, "\beta_i"'] & & &
\end{tikzcd}
$$

If the set of indices $\{i, j, k, \ldots, m\}$ is minimal w.r.t. set inclusion, the trajectory $M_1 \xrightarrow{\gamma_i} M_i' \cdots \xrightarrow{\gamma_m} M_m' \xrightarrow{\gamma_n} M_n'$ is a *maximal simple compression* with respect to the rule application $M_n \overset{\beta_n}{\rightharpoondown} M_{n+1}$. The trajectory records a number of rule applications selected in order from the original trajectory that lead to the observable, each rule application acting on a sub-Σ-graph related to that acted-on by the original rule applications through the residuals.

Example: Returning to the Kappa rules described in Figure 1 and the trajectory in Figure 2, consider the mixture M_5 and the following site graph G_0:



This represents an observable matched when there are two X-type agents with no links from their sites and at least one of them has a phosphorylated site a.

Let the matching $e_0 : G_0 \rightarrowtail M_5$ take the left X in G_0 to the top X in M_5 and the right X in G_0 to the bottom X in M_5. We have a trajectory $M_1 \overset{\beta_1}{\rightharpoondown} M_2 \overset{\beta_2}{\rightharpoondown} M_3 \overset{\beta_3}{\rightharpoondown} M_4 \overset{\beta_4}{\rightharpoondown} M_5 \overset{\beta_0}{\rightharpoondown} M_5$ where β_0 is the action testing the observable G_0. Simple compression can be applied as shown in Figure 3. The residual maps γ_3 and γ_0 both act as the identity on agents. γ_3 has the site on the top X phosphorylated and no property on the bottom X; it acts like β_2 to remove the bottom link, justifying the absence of the phosphorylation tag. γ_0 similarly has no p property on the lower X agent since it is temporarily removed in the composition $\beta_4 \circ \gamma_3$. The result is a maximally compressed trajectory with the sequence of rule applications β_1', β_3' and β_0', showing that the pattern can be matched in the same way as it was for the original trajectory simply following application of the rules r_1 and r_2. We have 'compressed' the original trajectory by removing the actions that unnecessarily affected the bottom X-agent. Mazurkiewicz compression could not have been applied to this trajectory due to the second weakness discussed in Section VI-B.

We omit details, but the example describing the first weakness of Mazurkiewicz compression is described using simple compression with two steps, the first moving the dephosphorylation action to occur in the initial mixture and the second moving the observable to occur immediately afterwards.

D. Weak compression

The framework described above for simple compression tests, for a given action, whether the corresponding rule could have been applied at an earlier point in the trajectory and, between that point and the original application, nothing that the rule requires to exist is affected. There is, however, an asymmetry: whilst, as we have seen, simple compression allows an event to be pushed back along a trajectory that temporarily adds structure (links or properties) that prevents application of the rule, it does not allow the event to be pushed back if structure (links or properties) is temporarily *removed*. This is addressed by *weak* compression.

The difference between weak compression and simple is that we now only require that e_2 be equal to $\beta_1 \circ e_2'$ *on agents*. This lifts the requirement that any links or properties required by e_2 are preserved through β_1.

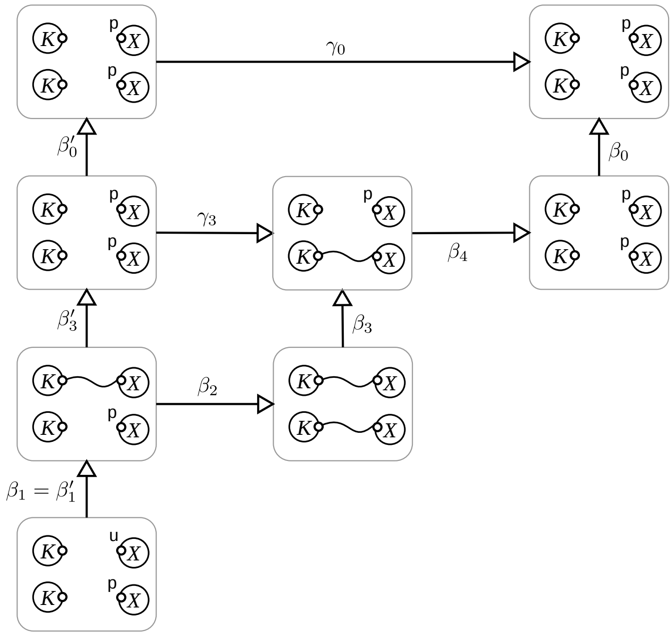


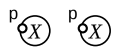
Fig. 3. An example of simple compression

We define the relation $\sim_{\mathsf{ag}}$ to be the least relation relating two partial morphisms $f, g : G \rightharpoonup H$ if $\forall n \in \mathcal{A}_G : n \in \mathrm{dom}(f) \iff n \in \mathrm{dom}(g)$ and $\forall n \in \mathrm{dom}(f) : f(n) = g(n)$.

Lemma 3. *Let $\beta_1 : M_1 \rightharpoondown M_2$ and $\alpha_2 : L_2 \rightharpoondown R_2$ and there be a matching $e_2 : L_2 \rightarrowtail M_1$. Let M_3 be a pushout of α_2 against e_2 with pushout morphisms $\beta_2 : M_2 \rightharpoondown M_3$ and $u_2 : R_2 \rightarrowtail M_3$. If there is a matching $e_2' : L_2 \rightarrowtail M_1$ such that $\beta_1 \circ e_2' \sim_{\mathsf{ag}} e_2$ then, letting M_2' be the pushout of α_2 against e_2' with pushout morphisms $\beta_2' : M_1 \rightharpoondown M_2'$ and $u_2' : R_2 \rightarrowtail M_2'$, there is a morphism $\gamma_2 : M_2' \to M_3$, unique up to $\sim_{\mathsf{ag}}$, such that $\gamma_2 \circ \beta_2' \sim_{\mathsf{ag}} \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' \sim_{\mathsf{ag}} u_2$. Furthermore, γ_2 is an action map.*

The proof of the lemma is given in Appendix D. The weak compression lemma can be applied in sequence to a number of rule applications in a given trajectory in exactly the same way as simple compression was, yielding a maximal weak compression.

Example: Returning to the example in Section VI-C, suppose now that we have the observable



matching in the same way as before via $e_0 : G_0 \rightarrowtail M_5$ taking the left X in G_0 to the top X in M_5 and the right X to the bottom X in M_5. With simple compression, the domain of definition of the map γ_3 did not include the phosphorylated property on the site on x_2. Simple compression cannot therefore be applied with respect to this new observable: there is no matching e_0' such that $\beta_4 \circ \gamma_3 \circ e_0' = e_0$. However, there is a matching e_0' such that $\beta_4 \circ \gamma_3 \circ e_0' \sim_{\mathsf{ag}} e_0$, namely the matching above. Weak compression can therefore be applied to generate the same compression with respect to this new observable.

E. Strong compression

Weak compression allows rule applications to be pushed back along trajectories to points where the same rule can be applied to act on the same agents, their identity being tracked by the action maps β_i. Strong compression relaxes this requirement: a strong compression can push a rule application backwards in a trajectory to any point where there is *any* matching of the pattern representing the left hand side of the corresponding rule; the rule does not have to be applied to the same agents.

A sequence of rules $s_1 \ldots s_m$ is said to be *realisable* from mixture N_1 if there is a trajectory $N_1 \overset{\sigma_1}{\rightharpoonup} N_2 \overset{\sigma_2}{\rightharpoonup} \ldots \overset{\sigma_m}{\rightharpoonup} N_{m+1}$ such that the rule applied to generate σ_i is s_i.

Consider a trajectory $M_1 \overset{\beta_1}{\rightharpoonup} M_2 \overset{\beta_2}{\rightharpoonup} \ldots \overset{\beta_n}{\rightharpoonup} M_{n+1}$ where the rule applied to generate β_i is r_i. A *strong compression* with respect to the rule application $\beta_n : M_n \rightharpoonup M_{n+1}$ is a sequence $s_1 \ldots s_m$ that is realisable from M_1 for which there is a monotone injection $f : \{i \ : \ 0 < i \leq m\} \to \{j \ : \ 0 < j \leq n\}$ such that $s_i = r_{f(i)}$ and $f(m) = n$. The compression is *maximal* if no rule can be removed from the sequence to yield a shorter strong compression.

Labelling the action maps with their rule applications, we obtain the following diagram:

$$\begin{tikzcd}
 & N_1 \arrow[r, "\sigma_1", "s_1"'] \arrow[dl, dotted, "="'] \arrow[d, dotted] & N_2 \arrow[d, dotted] \arrow[r, dotted, no head] & N_i \arrow[r, "\sigma_i", "s_i"'] \arrow[d, dotted] & N_{i+1} \arrow[d, dotted] \arrow[r, dotted, no head] & N_m \arrow[r, "\sigma_m", "s_m"'] \arrow[d, dotted] & N_{m+1} \arrow[d, dotted] \\
M_1 \arrow[r, dotted, no head] & M_{f(1)} \arrow[r, "\beta_{f(1)}", "s_1"'] & M_{f(1)+1} \arrow[r, dotted, no head] & M_{f(i)} \arrow[r, "\beta_{f(i)}", "s_i"'] & M_{f(i)+1} \arrow[r, dotted, no head] & M_n \arrow[r, "\beta_n", "s_n"'] & M_{n+1}
\end{tikzcd}$$

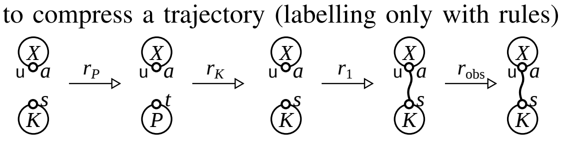
Any simple or weak compression is a strong compression, and the earlier constructions yield action maps for the dotted arrows. In the case of a simple compression, the dotted areas will commute up to equality; in the case of a weak compression, the areas will commute up to $\sim_{\mathsf{ag}}$.

F. Examples

As a first example, returning to the example for simple compression, the pattern where one X-agent has a phosphorylated unlinked site and the other has an unlinked site that may or may not be phosphorylated is matched in the initial mixture M_1, albeit with a different matching to that in the original trajectory. The maximal strong compression with respect to this observable is simply the action representing the match in the initial state.

A key area of application of strong compression is where an agent is deleted and subsequently an analogous agent is created. For example, we might have a pair of rules r_P and r_K with action maps

both with empty domain of definition (the first was seen in Section II-A). Alongside the rule r_1 from Fig. 1, we may wish to compress a trajectory (labelling only with rules)



where the last action is an observation of a link between a K-agent and an X-agent. No weak compression would be possible, intuitively since attempting to push the application of r_1 forward in the trajectory would result in the rule being applied to different agent (since K is not in the domain of definition of the composition of the maps for r_P and r_K). However, strong compression is possible, yielding the maximal compression where the observable follows r_1.

VII. Related work and conclusion

We have seen in this paper how a graphical rewriting semantics, based on single-pushouts, can be given to Kappa. This complements the work developed in [7], where a double-pushout semantics was given to a side-effect free version of the calculus with the intention of using contact graphs to provide types.

In future work, we intend to show how contact graphs can be used to provide types for the system presented here. The analogue of the adjunction representing change of contact graph presented there, with the presence of external links introduced in this paper, can be used as a foundation for ideas on views that are important in static analysis [4]. We also intend to study how this adjunction can be used to describe techniques for providing a flexible degree of context-sensitivity in the abstraction of information flow that is used for reducing quantitative semantics [9]. It would also be interesting to determine whether an approach between single- and double-pushout rewriting, such as sesqui-pushouts [10], can fruitfully be applied. Other interesting areas for future work are how to add further structure to mixtures, such as regions [11] to capture spatial aspects, and to describe how partial order structure can be obtained for compressed trajectories.

Acknowledgments

JH and GW gratefully acknowledge the support of the ERC through the Advanced Grant ECSYM, and JF and JH the support of the ANR AbstractCell Chair of Excellence

References

[1] J. Bachman and P. Sorger, “New approaches to modeling complex biochemistry,” *Nature Methods*, vol. 8, no. 2, pp. 130–131, 2011.
[2] V. Danos and C. Laneve, “Formal molecular biology,” *TCS*, vol. 325, 2004.
[3] V. Danos, J. Feret, W. Fontana, and J. Krivine, “Scalable simulation of cellular signaling networks,” in *Proc. APLAS*, 2007.
[4] ——, “Abstract interpretation of cellular signalling networks,” in *Proc. VMCAI*, 2008.
[5] M. Löwe, “Algebraic approach to single-pushout graph transformation,” *TCS*, vol. 109, 1993.
[6] kappalanguage.org, “The kappa simulator.”
[7] V. Danos, R. Harmer, and G. Winskel, “Constraining rule-based dynamics with types,” *MSCS*, 2012.
[8] G. Winskel and M. Nielsen, “Models for concurrency,” in *Handbook of Logic and the Foundations of Computer Science*. OUP, 1995.
[9] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine, “Abstracting the differential semantics of rule-based models: exact and automated model reduction,” in *Proc. LICS*, 2010.
[10] A. Corradini, T. Heindel, F. Hermann, and B. König, “Sesqui-pushout rewriting,” in *Proc. ICGT*, 2006.
[11] J. Hayman, C. Thompson-Walsh, and G. Winskel, “Simple containment structures in rule-based modelling of biochemical systems,” in *Proc. SASB*, 2011.

Appendix A
Pullbacks

Given morphisms $g : G \to J$ and $h : H \to J$, we shall give a pullback P with morphisms $p : P \to G$ and $q : P \to H$:

$$\begin{tikzcd} P \arrow[r, "q"] \arrow[d, "p"'] \arrow[dr, phantom, "\lrcorner", very near start] & H \arrow[d, "h"] \\ G \arrow[r, "g"'] & J \end{tikzcd}$$

Proposition 3. *The Σ-graph P and morphisms p, q defined as*

$$\begin{array}{rcl}
\mathcal{A}_P & = & \{(m,n) \;:\; m \in \mathcal{A}_G \;\&\; n \in \mathcal{A}_H \\
& & \qquad\qquad \& \; g(m) = h(n)\} \\
p(m,n) & = & m \\
q(m,n) & = & n \\
\mathsf{type}_P(m,n) & = & \mathsf{type}_G(m) = \mathsf{type}_H(n) = \mathsf{type}_J(g(m)) \\
\mathcal{S}_P & = & \{((m,n),i) \;:\; (m,i) \in \mathcal{S}_G \;\&\; (n,i) \in \mathcal{S}_H\} \\
\mathcal{L}_P & = & \{(((m,n),i),z),(z,((m,n),i)) \;: \\
& & \quad z \text{ is } \le_{\mathsf{type}_P}\text{-maximal s.t.} \\
& & \quad \exists x.((m,i),x) \in \mathcal{L}_G \;\&\; \hat{p}(z) \le_{\mathsf{type}_G} x \\
& & \quad \& \; \exists y.((n,i),y) \in \mathcal{L}_G \;\&\; \hat{q}(z) \le_{\mathsf{type}_H} y\} \\
p_{k,P} & = & \{((m,n),i) \;:\; (m,i) \in p_{k,G} \;\&\; (n,i) \in p_{k,H}\}
\end{array}$$

form a pullback in the category Σ-**Graph**.

Proof: Let Q be a Σ-graph and $r : Q \to G$ and $s : Q \to H$ be homomorphisms satisfying $gr = hs$. It is easy to see that the function t from the agents of Q defined as $t(n) = (r(n), s(n))$ yields an agent of P and that this is the unique function from the agents of Q to the agents of P such that $pt = r$ and $qt = s$.

It is easy to check the first three requirements for t to be a homomorphism, $t : Q \to P$. To see that t respects link structure, suppose that $((n,i),w) \in \mathcal{L}_Q$. Since r and s are homomorphisms, there exist x and y such that $((r(n),i),x) \in \mathcal{L}_G$ and $((s(n),i),y) \in \mathcal{L}_H$ and $\hat{r}(w) \le x$ and $\hat{s}(w) \le y$. We consider cases for w:

First take $w \in \mathsf{Ext}$. We have $\hat{p}(w) = w$ and $\hat{r}(w) = w$, so x satisfies $\hat{p}(w) \le_{\mathsf{type}_G} x$. Similarly, y satisfies $\hat{q}(w) \le_{\mathsf{type}_H} y$. Hence from the definition of $\mathcal{L}_P$, there exists z such that $w = \hat{t}(w) \le_{\mathsf{type}_P} z$ and $(((r(n),s(n)),i),z) \in \mathcal{L}_P$, as required, recalling that $t(n) = (r(n),s(n))$.

Now take $w \notin \mathsf{Ext}$. There exist m and j such that $w = (m,j)$. Since r and s are homomorphisms, we have $((r(n),i),(r(m),j)) \in \mathcal{L}_G$ and $((s(n),i),(s(m),j)) \in \mathcal{L}_H$. Due to the $\le_{\mathsf{type}_P}$-maximality of $((r(m),s(m)),j)$ and that $\hat{p}((r(m),s(m)),j) = (r(m),j)$ and $\hat{q}((r(m),s(m)),j) = (s(m),j)$, it is easy to verify that $(((r(n),s(n)),i),((r(m),s(m)),j)) \in \mathcal{L}_P$, as required. □

Appendix B
Partial morphisms: Concrete view

A partial morphism from G to H can alternatively be described more directly as a partial function from the agents of G to the agents of H alongside sets of sites, links and properties on which the partial morphism is defined.

Proposition 4. *A partial morphism $h : G \rightharpoonup H$ between Σ-site graphs G and H can equivalently be described as a partial function $h : \mathcal{A}_G \rightharpoonup \mathcal{A}_H$, a set $\mathcal{S}_0 \subseteq \mathcal{S}_G$, a symmetric relation $\mathcal{L}_0 \subseteq \mathcal{L}_G$ and for every $k \in \Sigma_{\mathsf{prop}}$ a set $p_{0,k} \subseteq p_{k,G}$ satisfying, for all n, i, k and x:*

- *if $(n,i) \in \mathcal{S}_0$ then $h(n)$ defined and $(h(n),i) \in \mathcal{S}_H$*
- *if $(n,i) \in p_{0,k}$ then $h(n)$ defined and $(h(n),i) \in p_{H,k}$*
- *if $((n,i),x) \in \mathcal{L}_0$ then $(n,i) \in \mathcal{S}_0$ and $\hat{h}(x)$ defined and there exists y such that $((h(n),i),y) \in \mathcal{L}_H$ and $\hat{h}(x) \le_{\mathsf{type}_H} y$, where the operation $\hat{\cdot}$ is extended to partial functions as*

$$\hat{h}(x) = \begin{cases} \textit{undefined} & \textit{if } x = (n,i) \textit{ and } h(n) \textit{ undefined} \\ (h(n),i) & \textit{if } x = (n,i) \textit{ and } h(n) \textit{ defined} \\ x & \textit{if } x \in \mathsf{Ext} \end{cases}$$

Proof: Clear from the definition: the apex of the span is formed with agents n such that $h(n)$ is defined, sites $\mathcal{S}_0$, links $\mathcal{L}_0$ and properties $p_{0,k}$ for every k. □

Appendix C
Rewriting: Proofs

Lemma 4. *D' is a mixture.*

Proof: The key property in showing that D' is a Σ-graph is that if $n \in \langle e,\alpha\rangle^-_{\mathsf{ag}}$ then $((n,i),x) \in \langle e,\alpha\rangle^-_{\mathsf{lnk}}$ for all i and x such that $((n,i),x) \in \mathcal{L}_M$, which is simply part of the definition. It is clear that D' is, furthermore, a mixture since by definition it has all possible sites, and any self-link or site with multiple links would have to occur as such in M. □

The following lemma expresses the key property that any deletion of links does not cause side-effects inside the image of the pattern L. This encapsulates the fact that matchings must take external links to links connecting agents outside their image, the final requirement on matchings.

Lemma 5. *If $((m,i),x) \in \mathrm{dom}(\alpha)$ and $((e(m),i),y) \in \mathcal{L}_M$ then $((e(m),i),y) \in \mathcal{L}_{D'}$.*

Proof: For contradiction, suppose that $((e(m),i),y) \in \langle e,\alpha\rangle^-_{\mathsf{lnk}}$. It is clear that $e(m) \notin \langle e,\alpha\rangle^-_{\mathsf{ag}}$ since $m \in \mathrm{dom}(\alpha)$ as a consequence of Proposition 4. Furthermore, as a consequence of the assumption that $((m,i),x) \in \mathrm{dom}(\alpha)$, since L is a site graph and e is injective, there are no n and z such that $e(n) = e(m)$ and $((n,i),z) \in \mathcal{L}_L \setminus \mathrm{dom}(\alpha)$. The only remaining possibility is that $y = (e(m'),i')$ for some m' and i' such that either $m' \in \langle e,\alpha\rangle^-_{\mathsf{ag}}$ or there exists z such that $((m',i'),z) \in \mathcal{L}_L \setminus \mathrm{dom}(\alpha)$.

From the final constraint on matchings, we see that $((m,i),(m',i')) \in \mathcal{L}_L$ and hence $x = (m',i')$ since L is a site graph. However, then we have $((m,i),(m',i)) \in \mathrm{dom}(\alpha)$, from which we immediately derive a contradiction using Proposition 4 since we may infer that $m' \in \mathrm{dom}(\alpha)$ and there can be no z such that $((m',i'),z) \in \mathcal{L}_L \setminus \mathrm{dom}(\alpha)$. □

Lemma 6. *The newly-created elements are disjoint from those that are preserved:*

$$\begin{aligned}\mathcal{A}_{D'} \cap \langle \alpha, \phi \rangle^{+}_{\mathsf{ag}} &= \varnothing \\ \mathcal{L}_{D'} \cap \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}} &= \varnothing \\ p_{D',k} \cap \langle \alpha, \phi \rangle^{+}_{k} &= \varnothing\end{aligned}$$

Proof: The case for agents is an immediate consequence of the definitions, and the case for properties is similar to that for links, on which we now focus.

Suppose, for contradiction, that there exists $((n,i),x) \in \mathcal{L}_{D'} \cap \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$. Since $((n,i),x) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$, there exist m and y such that $((m,i),y) \in \mathcal{L}_R$ and $u(m) = n$ and $\hat{u}(y) = x$, but $\nexists z, p$ such that both $\alpha(p) = m$ and $((p,i),z) \in \mathrm{dom}(\alpha)$.

Since $((n,i),x) \in \mathcal{L}_{D'}$ and D' is a Σ-graph by Lemma 4, we have $n \in \mathcal{A}_{D'}$ and hence $n = u(m) \notin X$. It follows from the definition of u that there exists $p \in \mathrm{dom}(\alpha)$ such that $n = e(p)$ and $\alpha(p) = m$.

There exists w such that $((p,i),w) \in \mathcal{L}_L$ since u is an embedding and therefore sends empty sites to empty sites and by assumption $((n,i),x) \in \mathcal{L}_{D'}$ so $((n,i),x) \in \mathcal{L}_M$. Furthermore, from the definition of $\mathcal{L}_{D'}$ we have $((n,i),x) \notin \langle e, \alpha \rangle^{-}_{\mathsf{lnk}}$, so $((p,i),w) \in \mathrm{dom}(\alpha)$. This immediately yields a contradiction to the non-existence of z and p above. □

Lemma 7. *u is an embedding.*

Proof: Homomorphism: It is straightforward to establish that $u(m) \in \mathcal{A}_N$ for all $m \in \mathcal{A}_R$.

Suppose that $((m,i),x) \in \mathcal{L}_R$; we shall show that there exists y such that $((u(m),i),y) \in \mathcal{L}_N$ and $\hat{u}(x) \leq_{\mathsf{type}_N} y$.

If there exist no z and m such that $\alpha(m) = n$ and $((m,i),z) \in \mathrm{dom}(\alpha)$ then we have $((u(m),i),\hat{u}(x)) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$, which is a subset of $\mathcal{L}_N$, as required.

Now suppose that there exist z and m such that $\alpha(m) = n$ and $((m,i),z) \in \mathrm{dom}(\alpha)$. We have $((e(m),i),y) \in \mathcal{L}_M$ for some y such that $\hat{e}(z) \leq_{\mathsf{type}_M} y$ since e is an embedding and therefore total. Furthermore, $((e(m),i),y) \in \mathcal{L}_{D'}$ by Lemma 5 and hence $((e(m),i),y) \in \mathcal{L}_N$. Since $\alpha(m) = n$, by definition we have $u(n) = m$.

Consider cases for z. If $z \in \mathsf{Ext}$ then $z = x$ since α is an action map and cannot promote an external link. Hence $\hat{e}(z) = \hat{u}(x) = z$, so, recalling that $\hat{e}(z) \leq_{\mathsf{type}_M} y$, we have $\hat{u}(x) \leq_{\mathsf{type}_N} y$. If $z \notin \mathsf{Ext}$ then there exist m' and i' such that $z = (m', i')$. We have $y = (e(m'), i')$ since e is an embedding and $x = (\alpha(m'), i')$ since R is a site graph and $((m,i),z) \in \mathrm{dom}(\alpha)$. We have $\hat{u}(\alpha(m'), i') = (u(\alpha(m')), i') = (e(m'), i') = y$, and therefore trivially $\hat{u}(x) \leq_{\mathsf{type}_N} y$.

Injective: Suppose that $u(m) = n = u(m')$. We shall show that $m = m'$.

If $m, m' \notin \mathrm{image}(\alpha)$ then $u(m) = \phi(m) = \phi(m') = u(m')$, and hence $m = m'$ since ϕ is a bijection. If $m, m' \in \mathrm{image}(\alpha)$ then clearly we must have $m = m'$ since α and e are injective. The only remaining case is when precisely one of m, m' is in $\mathrm{image}(\alpha)$, so without loss of generality take $m \in \mathrm{image}(\alpha)$ and $m' \notin \mathrm{image}(\alpha)$. We shall derive a contradiction since we cannot possibly have $m = m'$ in this case.

There exists q such that $\alpha(q) = m$ and hence $u(m) = e(q)$. By definition, $e(q) \notin \langle e, \alpha \rangle^{-}_{\mathsf{ag}}$, so $n = e(q) \in D'$. However, we have $u(m') = \phi(m')$, and therefore $n = u(m') \in X$. However, $X \cap D' = \varnothing$, and so $n \notin D'$: a contradiction.

Emptiness preservation: We now show that u preserves emptiness. Suppose that $(n,i) \in \mathcal{S}_R$ and $((u(n),i),y) \in \mathcal{L}_N$. We wish to show that there exists x such that $((n,i),x) \in \mathcal{L}_R$. Recalling the definition of $\mathcal{L}_N$, there are two cases to consider. Firstly, if $((u(n),i),y) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$ then there exists n' such that $u(n') = u(n)$ and $((n',i),x) \in \mathcal{L}_R$. We have already seen that u is injective, so $n = n'$, and hence we arrive at the required x. Secondly, if $((u(n),i),y) \in \mathcal{L}_{D'}$ then $u(n) \in \mathcal{A}_{D'}$. By Lemma 6, we have $u(n) \notin \langle \alpha, \phi \rangle^{+}_{\mathsf{ag}}$, so there exists $m \in \mathrm{dom}(\alpha)$ such that $\alpha(m) = n$ and $u(n) = e(m)$. Since α is an action, it does not delete or create sites without deleting or creating their associated agents, so $(m,i) \in \mathrm{dom}(\alpha)$ since $(n,i) \in \mathcal{S}_R$. By assumption, e is an embedding and therefore preserves emptiness, so there exists z such that $((m,i),z) \in \mathcal{L}_L$. We have already established that $((u(n),i),y) = ((e(m),i),y) \in \mathcal{L}_{D'}$ so in particular $((e(m),i),y) \notin \langle e, \alpha \rangle^{-}_{\mathsf{lnk}}$. From the definition of $\langle e, \alpha \rangle^{-}_{\mathsf{lnk}}$, we see that $((m,i),z) \in \mathrm{dom}(\alpha)$, and so from Proposition 4 we immediately infer that there exists x such that $((n,i),x) \in \mathcal{L}_R$.

Externality: We conclude by showing that the target agent of the image of any external link is outside the image of the embedding. Suppose, for contradiction, that there exist x, n, p, i and j such that $x \in \mathsf{Ext}$ and $((n,i),x) \in \mathcal{L}_R$ and $((u(n),i),(u(p),j)) \in \mathcal{L}_N$. Since R is a site graph and u is injective, we cannot possibly have $((u(n),i),(u(p),j)) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$, so $((u(n),i),(u(p),j)) \in \mathcal{L}_{D'}$. Immediately, since D' is a Σ-graph, we have $u(p) \in \mathcal{A}_{D'}$ and hence $u(p) \notin \langle \alpha, \phi \rangle^{+}_{\mathsf{ag}}$ by Lemma 6. From the definition of $\langle \alpha, \phi \rangle^{+}_{\mathsf{ag}}$, there exists q such that $\alpha(q) = p$. All new links created by actions are concrete, so there exist m and y such that $\alpha(m) = n$ and $((m,i),y) \in \mathrm{dom}(\alpha)$. Furthermore, actions cannot promote external links, so $x = y$. Note that $e(m) = u(n)$ and $e(q) = u(p)$, and hence $((e(m),i),(e(q),j)) \in \mathcal{L}_{D'}$. We now arrive at the required contradiction: e is assumed to be an embedding, but $((m,i),x) \in \mathcal{L}_L$ and $((e(m),i),(e(q),j)) \in \mathcal{L}_M$. □

Lemma 8. *N is a mixture.*

Proof: Σ-graph: We begin by showing that N is a Σ-graph, firstly by showing that the link relation is symmetric. It is clear that $\mathcal{L}_{D'}$ is symmetric since D' is a Σ-graph, so symmetry of $\mathcal{L}_N$ shall follow from the symmetry of $\langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$.

Suppose that $((m,i),x) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$. There exist n and y such that $u(n) = m$ and $((n,i),y) \in \mathcal{L}_R$ and $\hat{u}(y) = x$ but there exist no p and z such that both $\alpha(p) = n$ and $((p,i),z) \in \mathrm{dom}(\alpha)$. Action morphisms can only create concrete links, so $y = (n', i')$ for some n', i', and therefore $x = (u(n'), i')$.

By assumption, $\mathcal{L}_R$ is symmetric, so $((n',i'),(n,i)) \in \mathcal{L}_R$. All that remains before we can apply the definition to conclude that $(x,(m,i)) = ((u(n'),i),\hat{u}(n,i)) \in \langle \alpha, \phi \rangle^{+}_{\mathsf{lnk}}$ is to show that there are no p' and z' such that both $\alpha(p') = n'$ and $((p',i'),z') \in \mathrm{dom}(\alpha)$. Suppose, for contradiction, that such p' and z' exist. We cannot have $z \in \mathsf{Ext}$ since actions cannot promote external links, so $z = (p, j)$ for some p and j. It follows that $((p', i'), (\alpha(p), j)) \in \mathcal{L}_R$. However, R is assumed to be a site graph, so $i = j$ and $\alpha(p) = n$. It follows immediately that $((n, i), y) \notin \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$, giving the required contradiction.

We now show that if $\ell = ((n, i), x) \in \mathcal{L}_N$ then $n \in \mathcal{A}_N$. There are two cases to consider: $\ell \in \mathcal{L}_{D'}$ and $\ell \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$.

Suppose that $\ell \in \mathcal{L}_{D'}$. Since D' is a Σ-graph, we have $n \in \mathcal{A}_{D'}$ and therefore $n \in \mathcal{A}_N$.

Now suppose that $\ell \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$. There exist m and y such that $n = u(m)$ and $x = \hat{u}(y)$ and $((n, i), x) \in \mathcal{L}_R$. If $m \notin \mathrm{image}(\alpha)$ then $u(m) = \phi(m) \in X = \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$, so $u(m) \in \mathcal{A}_N$. If there exists $p \in \mathrm{dom}(\alpha)$ such that $\alpha(p) = m$ then $n = u(m) = e(p)$. From the definition, we have $n \in \mathcal{A}_{D'}$ and hence $n \in \mathcal{A}_N$.

<u>Site graph</u>: We now turn to the first requirement for N to be a site graph, that there are no n and i such that $((n, i), (n, i)) \in \mathcal{L}_N$.

Suppose, for contradiction, that such n and i exist. We cannot have $((n, i), (n, i)) \in \mathcal{L}_{D'}$ since we have already seen that D' is a mixture, so we must have $((n, i), (n, i)) \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$. Hence there exist m and x such that $((m, i), x) \in \mathcal{L}_R$ and $u(m) = n$ and $\hat{u}(x) = (n, i)$. From the definition of $\hat{u}$, we have $x = (p, i)$ for some p such that $u(p) = n$. By injectivity of u, we have $p = m$ so $((m, i), (m, i)) \in \mathcal{L}_R$, contradicting R being a site graph.

We now turn to the second requirement for N to be a site graph, that if $((n, i), x), ((n, i), x') \in \mathcal{L}_N$ then $x = x'$. Let $\ell = ((n, i), x)$ and $\ell' = ((n, i), x')$. Clearly, $x = x'$ if $\ell, \ell' \in D'$ since D' is a mixture, and it is easy to check that if $\ell, \ell' \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$ then $\ell = \ell'$ due to the injectivity of u on agents. We now show that the other remaining case, where $\ell \in D'$ and $\ell' \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$ (or without loss of generality, vice versa), is absurd.

Since $\ell' \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$, there exist m, y such that $((m, i), y) \in \mathcal{L}_R$ and $u(m) = n$ and $\hat{u}(y) = x'$ and there are no p and z such that both $\alpha(p) = m$ and $((p, i), z) \in \mathrm{dom}(\alpha)$.

We have $n \in \mathcal{A}_{D'}$ since $((n, i), x) \in \mathcal{L}_{D'}$ and D' is a mixture, so $n \notin \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$ by Lemma 6. It follows that there exists $p \in \mathrm{dom}(\alpha)$ such that $\alpha(p) = m$. Note that $u(m) = n = e(p)$ by definition, and since α is an action we have $(p, i) \in \mathcal{S}_L$. Since e is an embedding, it preserves emptiness, so there exists z such that $((p, i), z) \in \mathcal{L}_L$ because $((e(p), i), x) \in \mathcal{L}_{D'} \subseteq \mathcal{L}_M$. We saw earlier that we cannot have $((p, i), z) \in \mathrm{dom}(\alpha)$, so we must have $\ell \in \langle e, \alpha \rangle^-_{\mathsf{lnk}}$, contradicting $\ell \in D'$.

<u>Mixture</u>: It is clear that N has all possible sites, so all we need to show is that it has no external links. Suppose that $((m, i), y) \in \mathcal{L}_N$. Either $((m, i), y) \in \mathcal{L}_{D'}$, in which case $((m, i), y) \in \mathcal{L}_M$ and hence there exist m', i' such that $y = (m', i')$ since M is a mixture, or $((m, i), y) \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$. In this case, there exist n and x such that $m = u(n)$ and $((n, i), x) \in \mathcal{L}_R$ and $y = \hat{u}(x)$ and $\nexists z, p$ such that both $((p, i), z) \in \mathrm{dom}(\alpha)$ and $\alpha(p) = m$. Since α is an action map, it only creates concrete links; there exist q and j such that $x = (q, j)$, so $y = (u(q), j)$. The link $((m, i), y)$ is therefore not an external link, as required. $\square$

Theorem 2 (Theorem 1)**.** *Given an action $\alpha : L \rightharpoonup R$ and a matching $e : L \rightarrowtail M$, define $\beta_0(m) = m$ for any $m \in \mathcal{A}_{D'}$. The constructions above yield a pushout in the category Σ-$\mathbf{Site}_*$*

$$\begin{tikzcd} L \arrow[r, "\alpha"] \arrow[d, "e"'] & R \arrow[d, "u"] \arrow[dl, phantom, "\ulcorner", very near end] \\ M \arrow[r, "\beta"'] & N \end{tikzcd} \quad \textit{where} \quad \begin{tikzcd} & D' \arrow[dl] \arrow[dr, "\beta_0"] & \\ M \arrow[rr, "\beta"'] & & N \end{tikzcd}$$

Furthermore, N is a mixture, u is a matching and β is an action map.

Proof: <u>β is an action map</u>: Trivial since M, D' and N are mixtures and β acts, where defined, as the identity.
<u>$\beta \circ e = u \circ \alpha$</u>: This follows straightforwardly from the definitions, apart from links where Lemma 5 excludes the possibility of a side-effect preventing commutation. In full detail:

Agents: Let $m \in \mathcal{A}_L$. We first prove that

$$m \in \mathrm{dom}(\alpha) \implies e(m) \in D' \;\; \& \;\; u\alpha(m) = \beta(e(m)).$$

Suppose that $m \in \mathrm{dom}(\alpha)$. By injectivity of e, we have $e(m) \notin \langle e, \alpha \rangle^-_{\mathsf{ag}}$, so $e(m) \in D'$. From the definition of u, we have $u(\alpha(m)) = e(m)$, and applying the definition of β we immediately obtain $u(\alpha(m)) = \beta(e(m))$.

If, on the other hand, $m \notin \mathrm{dom}(\alpha)$ then $e(m) \in \langle e, \alpha \rangle^-_{\mathsf{ag}}$, so $e(m) \notin D'$. It follows immediately that $\beta \circ e(m) = u \circ \alpha(m)$.

Sites: We wish to show that $(m, i) \in \mathrm{dom}(\alpha)$ iff $(e(m), i) \in D'$. There are two cases to consider. Firstly, if $m \notin \mathrm{dom}(\alpha)$ then clearly $(m, i) \notin \mathrm{dom}(\alpha)$. By commutation on agents, we have $e(m) \notin \mathcal{A}_{D'}$, so $(e(m), i) \notin \mathcal{S}_{D'}$. The second case has $m \in \mathrm{dom}(\alpha)$. In this case, we have $(m, i) \in \mathrm{dom}(\alpha)$ since α is an action and therefore not delete sites without deleting their agents. By commutation on agents, we have $e(m) \in \mathcal{A}_{D'}$, so from the definition of D' we have $(e(m), i) \in \mathcal{S}_{D'}$.

Links: Let $((m, i), x) \in \mathcal{L}_L$. Since e is a homomorphism into a mixture, there exists unique y such that $((e(m), i), y) \in \mathcal{L}_M$. We wish to show that $((m, i), x) \in \mathrm{dom}(\alpha)$ iff $((e(m), i), y) \in \mathcal{L}_{D'}$. We consider three cases.

Firstly, if $m \notin \mathrm{dom}(\alpha)$ then since α is a partial morphism, by Proposition 4 we have $((n, i), x) \notin \mathrm{dom}(\alpha)$. We have $e(m) \in \langle e, \alpha \rangle^-_{\mathsf{ag}}$, and hence $((e(m), i), y) \in \langle e, \alpha \rangle^-_{\mathsf{lnk}}$. It follows immediately from the definition of D' that $((e(m), i), y) \notin \mathcal{L}_{D'}$.

Now suppose that $m \in \mathrm{dom}(\alpha)$ but $((m, i), x) \notin \mathrm{dom}(\alpha)$. From the definition, $((e(m), i), y) \in \langle e, \alpha \rangle^-_{\mathsf{lnk}}$, so $((e(m), i), y) \notin \mathcal{L}_{D'}$.

The final case, where $m \in \mathrm{dom}(\alpha)$ and $((m, i), x) \in \mathrm{dom}(\alpha)$, is covered in Lemma 5.

Properties: Suppose that $(m, i) \in p_{L,k}$. We wish to show that $(m, i) \in \mathrm{dom}_{\mathsf{prop},k}(\alpha)$ iff $(e(m), i) \in p_{D',k}$. We have already seen that $m \in \mathrm{dom}(\alpha)$ iff $e(m) \in \mathcal{A}_{D'}$, so if $m \notin \mathrm{dom}(\alpha)$ then clearly both $(m, i) \notin \mathrm{dom}_{\mathsf{prop},k}(\alpha)$ and $(e(m), i) \notin p_{D',k}$. Assume, then, that $m \in \mathrm{dom}(\alpha)$, so $e(m) \notin \langle e, \alpha \rangle^-_{\mathsf{ag}}$ since e is a matching and therefore injective. From the definition of $\langle e, \alpha \rangle^-_{\mathsf{prop}}$, in this case we have $(e(m), i) \in \langle e, \alpha \rangle^-_{\mathsf{prop}}$ iff $(m, i) \notin \mathrm{dom}_{\mathsf{prop},k}\alpha$, so from the definition of D' we obtain $(m, i) \in \mathrm{dom}_{\mathsf{prop},k}(\alpha)$ iff $(e(m), i) \in p_{D',k}$.

Universal property: Let P be a Σ-graph and $f : M \to P$ and $g : R \to P$ be morphisms in Σ-$\mathbf{Site}_*$ such that $f \circ e = g \circ \alpha$. We define a morphism h that is the unique morphism $h : N \to P$ such that the triangles in the following diagram commute:

$$\begin{tikzcd}
L \arrow[r, "\alpha"] \arrow[d, "e"'] & R \arrow[d, "u"] \arrow[ddr, bend left, "g"] & \\
M \arrow[r, "\beta"'] \arrow[drr, bend right, "f"'] & N \arrow[dr, dotted, "h"] & \\
& & P
\end{tikzcd}$$

Define $h : N \to P$ as:

- for $n \in \mathcal{A}_{D'}$: $h(n) = f(n)$
  for $n \in \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$: $h(n) = g(\phi^{-1}(n))$
- for $(n, i) \in \mathcal{S}_N$, if $n \in \mathcal{A}_{D'}$:
  $$(n, i) \in \mathrm{dom}(h) \iff (n, i) \in \mathrm{dom}(f)$$
  if $n \in \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$:
  $$(n, i) \in \mathrm{dom}(h) \iff (\phi^{-1}(n), i) \in \mathrm{dom}(g)$$
- for $((n, i), x) \in \mathcal{L}_{D'}$:
  $$((n, i), x) \in \mathrm{dom}(h) \iff ((n, i), x) \in \mathrm{dom}(f)$$
  for $((u(n), i), \hat{u}(x)) \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$:
  $$((u(n), i), \hat{u}(x)) \in \mathrm{dom}(h) \iff ((n, i), x) \in \mathrm{dom}(g)$$
- for $(n, i) \in p_{D',k}$:
  $$(n, i) \in \mathrm{dom}_{\mathsf{prop},k}(h) \iff (n, i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$$
  for $(n, i) \in \langle \alpha, \phi \rangle^+_{\mathsf{prop},k}$:
  $$(n, i) \in \mathrm{dom}_{\mathsf{prop},k}(h) \iff (n, i) \in \mathrm{dom}_{\mathsf{prop},k}(g)$$

$g = h \circ u$: **Agents:** Let $n \in \mathcal{A}_R$. There are two cases to consider. Firstly, if $u(n) \in \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$ then $u(n) = \phi(n)$, so $h(u(n)) = g(n)$ by definition. Now suppose that $u(n) \in D'$. From the definition of h, we have $h(u(n)) = f(u(n))$. There exists $p \in \mathrm{dom}(\alpha)$ such that $\alpha(p) = n$; otherwise, we would have $u(n) = \phi(n)$ and so $u(n) \in \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$, contradicting $u \in D'$. Moreover, p is unique by partial injectivity of α. Hence, from the definition of u, we have $u(n) = e(p)$. Since the outer square commutes,

$$h(u(n)) = f(u(n)) = f \circ e(p) = g \circ \alpha(p) = g(n),$$

as required.

Sites: Let $(n, i) \in \mathcal{S}_R$. We must show that $(u(n), i) \in \mathrm{dom}(h)$ iff $(n, i) \in \mathrm{dom}(g)$. There are two cases to consider.

Firstly, if $n \in \mathrm{image}(\alpha)$ then there exists m such that $\alpha(m) = n$ and, since α is an action map, we have $(m, i) \in \mathcal{S}_L$ and $(m, i) \in \mathrm{dom}(\alpha)$. Recalling that $\beta \circ e = u \circ \alpha$, and hence $e(m) = u(\alpha(m))$, from the definition of h we have $(u(\alpha(m)), i) \in \mathrm{dom}(h)$ iff $(u(n), i) \in \mathrm{dom}(f)$ iff $(e(m), i) \in \mathrm{dom}(f)$. The outer square commutes, so $(e(m), i) \in \mathrm{dom}(f)$ iff $(\alpha(m), i) \in \mathrm{dom}(g)$. Thus $(u(n), i) \in \mathrm{dom}(h)$ iff $(n, i) \in \mathrm{dom}(g)$, as required.

Secondly, if $n \notin \mathrm{image}(\alpha)$, we have $u(n) = \phi(n) \in \langle \alpha, \phi \rangle^+_{\mathsf{ag}}$. It follows immediately from the definition of h that $(u(n), i) \in \mathrm{dom}(h)$ iff $(n, i) \in \mathrm{dom}(g)$.

Links: Let $((n, i), x) \in \mathcal{L}_R$. Since u is a matching, there exists y such that $((u(n), i), y) \in \mathcal{L}_N$ and $\hat{u}(x) \leq_{\mathsf{type}_N} y$. We must show that $((u(n), i), y) \in \mathrm{dom}(h)$ iff $((n, i), x) \in \mathrm{dom}(g)$.

First suppose that $((u(n), i), y) \in \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$. Since e is a matching and R is a site graph, from the definition of $\langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$ we have $y = \hat{u}(x)$. From the definition of h, we immediately obtain $((n, i), x) \in \mathrm{dom}(g)$ iff $((u(n), i), y) \in \mathrm{dom}(h)$.

Now suppose that $((u(n), i), y) \notin \langle \alpha, \phi \rangle^+_{\mathsf{lnk}}$. From the definition of $\mathcal{L}_N$, we have $((u(n), i), y) \in \mathcal{L}_{D'}$, and therefore $((u(n), i), y) \in \mathrm{dom}(h)$ iff $((u(n), i), y) \in \mathrm{dom}(f)$. We have $u(n) \in D'$, so there exists $m \in \mathrm{dom}(\alpha)$ such that $\alpha(m) = n$; otherwise, we would have $u(n) = \phi(n)$, so $u(n) \in X$ which is assumed to be disjoint from D', giving a contradiction. Since α is an action map, $(m, i) \in \mathrm{dom}(\alpha)$. Recalling that $\beta \circ e = e \circ \alpha$, we have $e(m) = u(n)$. Matchings take empty sites to empty sites and $((u(n), i), y) \in \mathcal{L}_M$ since $((u(n), i), y) \in \mathcal{L}_{D'}$, so there exists z such that $((m, i), z) \in \mathcal{L}_L$ and $\hat{e}(z) \leq_{\mathsf{type}_M} y$. Moreover, $((m, i), z) \in \mathrm{dom}(\alpha)$ since otherwise we would have $((u(n), i), y) = ((e(m), i), y) \in \langle e, \alpha \rangle^-_{\mathsf{lnk}}$, contradicting $((u(n), i), y) \in \mathcal{L}_{D'}$. Since L is a site graph and we have $f \circ e = \alpha \circ g$, it must be the case that $((e(m), i), y) \in \mathrm{dom}(f)$ iff $((n, i), x) \in \mathrm{dom}(g)$. We earlier saw that $((u(n), i), y) \in \mathrm{dom}(h)$ iff $((u(n), i), y) \in \mathrm{dom}(f)$, so the case is complete.

Properties: Let $(n, i) \in p_{R,k}$. Since u is a matching, $(u(n), i) \in p_{N,k}$. We must show that $(u(n), i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(n, i) \in \mathrm{dom}_{\mathsf{prop},k}(g)$. If $(u(n), i) \in \langle \alpha, \phi \rangle^+_k$, the property follows immediately from the definition of h. Otherwise, from the definition of N, we have $(u(n), i) \in p_{D',k}$ and therefore $(u(n), i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(n, i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$. We have $u(n) \in D'$, so, as seen in the previous case, there exists $m \in \mathrm{dom}(\alpha)$ such that $\alpha(m) = n$ and $e(m) = u(n)$. By assumption, $(u(n), i) \notin \langle \alpha, \phi \rangle^+_k$, so $(m, i) \in \mathrm{dom}_{\mathsf{prop},k}(\alpha)$. Since e is a matching and therefore total, $(e(m), i) \in p_{M,k}$. Since $f \circ e = \alpha \circ g$, it must be the case that $(e(m), i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$ iff $(n, i) \in \mathrm{dom}_{\mathsf{prop},k}(g)$. We earlier saw that $(u(n), i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(u(n), i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$, so the case is complete.

h is a morphism: It follows straightforwardly from f and g being morphisms and the disjointness of $\mathcal{A}_{D'}$ and $\langle \alpha, \phi \rangle^+_{\mathsf{ag}}$ that h is a partial function that preserves types, sites and properties. We now consider link specification.

Suppose that $\ell = ((m, i), x) \in \mathcal{L}_N$ and $\ell \in \mathrm{dom}(h)$. We shall show that there exists y such that $((h(m), i), y) \in \mathcal{L}_P$ and $\hat{h}(x) \leq_{\mathsf{type}_P} y$. Consider the two distinct cases for $\ell \in \mathcal{L}_N$.

- If $\ell \in \mathcal{L}_{D'}$ then we must have $\ell \in \mathrm{dom}(f)$ since $\ell \in \mathrm{dom}(h)$. Since f is a morphism, it follows from Proposition 4 that $(m, i) \in \mathrm{dom}(f)$ and $\hat{f}(x)$ defined and $((f(m), i), y) \in \mathcal{L}_P$ for some y such that $\hat{f}(x) \leq_{\mathsf{type}_P} y$. We must have $m \in \mathcal{A}_{D'}$ and $(m, i) \in \mathcal{S}_{D'}$, so we have $h(m) = f(m)$ and $(m, i) \in \mathrm{dom}(h)$. It is easy to check that $\hat{h}(x)$ is defined as a consequence of $\hat{f}(x)$ being defined: the only non-trivial case is where $x = (n, j)$ for some n and j, and since we have $\ell \in D'$ we have $n \in D'$, and therefore $h(n) = f(n)$, so $f(n)$ must be defined and thus $h(n)$ must be defined.

- If $\ell \in \langle\alpha,\phi\rangle^+_{\mathsf{lnk}}$ then there exist n and z such that $((n,i),z) \in \mathcal{L}_R$ and $m = u(n)$ and $x = \hat{u}(z)$. Recalling that $g = h \circ u$ and that $\ell \in \mathrm{dom}(h)$, we must have $((n,i),z) \in \mathrm{dom}(g)$. Hence $(n,i) \in \mathrm{dom}(g)$ and $\hat{g}(z)$ is defined and there exists y such that $((g(n),i),y) \in \mathcal{L}_P$ and $\hat{g}(z) \leq_{\mathsf{type}_P} y$, by Proposition 4.
From the earlier-established fact that $g = h \circ e$, we must have $(m,i) \in \mathrm{dom}(h)$. We now show that $\hat{h}(x)$ is defined. If $z \in \mathsf{Ext}$, we have $x = \hat{u}(z) = z$, so $\hat{h}(x)$ is certainly defined. If $z \notin \mathsf{Ext}$, it must be the case that $z = (n',i')$ and since $\hat{u}(z) = x$ we have $x = (u(n'),i')$. We have $g(n')$ defined since $\hat{g}(n',i')$ is defined, and again since $g = h \circ u$ we therefore have $h(u(n'))$ defined, from which we may conclude that $\hat{h}(u(n'),i') = \hat{h}(x)$ is defined.
We conclude the case by showing that $\hat{h}(x) \leq_{\mathsf{type}_P} y$. First, if $z \in \mathsf{Ext}$ then $x = \hat{u}(z) = z$ and $\hat{g}(z) = z$ and $\hat{h}(x) = x$. From the earlier assumption that $\hat{g}(z) \leq_{\mathsf{type}_P} y$, we immediately obtain $\hat{h}(x) \leq y$. Alternatively, if $z \notin \mathsf{Ext}$, then there exist n',i' such that $z = (n',i')$ and hence $x = (\hat{u}(n'),i')$. Since $\hat{g}(n',i') \leq_{\mathsf{type}_P} y$, we must have $y = (g(n'),i')$. Note that $h(u(n')) = g(n')$ since $g = h \circ u$, and hence $\hat{h}(x) = \hat{h}(u(n'),i') = \hat{g}(n',i') = y$, as required.

$f = h \circ \beta$: **Agents:** Let $m \in \mathcal{A}_M$. Either $m \notin \langle e,\alpha\rangle^-_{\mathsf{ag}}$ or $m \in \langle e,\alpha\rangle^-_{\mathsf{ag}}$. In the first case, where $m \notin \langle e,\alpha\rangle^-_{\mathsf{ag}}$, we have $m \in \mathcal{A}_{D'}$ and $\beta(m) = m$, and from the definition of h we obtain $h(n) = f(n)$. For the second case, suppose that $m \in \langle e,\alpha\rangle^-_{\mathsf{ag}}$. From the definition of D', we have $m \notin \mathcal{A}_{D'}$, *i.e.* $m \notin \mathrm{dom}(\beta)$. We must therefore show that $m \notin \mathrm{dom}(f)$. Applying the definition of $\langle e,\alpha\rangle^-_{\mathsf{ag}}$, there exists (necessarily unique, by injectivity) $n \in \mathcal{A}_L$ such that $e(n) = m$ and $n \notin \mathrm{dom}(\alpha)$. We therefore cannot have $m \in \mathrm{dom}(f)$ since $g \circ \alpha = f \circ e$.

Sites: Suppose that $(m,i) \in \mathcal{S}_M$. We wish to show that $(m,i) \in \mathrm{dom}(f)$ iff $(m,i) \in \mathcal{S}_{D'}$ and $(\beta(m),i) \in \mathrm{dom}(h)$. There are two cases to consider. If $m \in \langle e,\alpha\rangle^-_{\mathsf{ag}}$ then $m \notin \mathcal{A}_{D'}$ so $(m,i) \notin \mathcal{S}_{D'}$. As in the case for agents, we cannot have $m \in \mathrm{dom}(f)$, so $(m,i) \notin \mathrm{dom}(f)$. If, on the other hand, $n \notin \langle e,\alpha\rangle^-_{\mathsf{ag}}$ then $(m,i) \in D'$. In Lemma 4, we saw that D' is a mixture, so $(m,i) \in \mathcal{S}_{D'}$. We have $\alpha(m) = m$ and, from the definition of h, we have $(m,i) \in \mathrm{dom}(h)$ iff $(m,i) \in \mathrm{dom}(f)$, as required.

Links: Suppose that $\ell = ((m,i),(n,j)) \in \mathcal{L}_M$. We wish to show that $\ell \in \mathrm{dom}(f)$ iff $\ell \in \mathcal{L}_{D'}$ and $\ell \in \mathrm{dom}(h)$. There are two cases to consider.

Firstly, if $\ell \in \langle e,\alpha\rangle^-_{\mathsf{lnk}}$ we have $\ell \notin \mathcal{L}_{D'}$, *i.e.* $\ell \notin \mathrm{dom}(\beta)$. There exists p such that either $p \notin \mathrm{dom}(\alpha)$ or there exists z such that $((p,i),z) \in \mathcal{L}_L \smallsetminus \mathrm{dom}(\alpha)$, and furthermore either $e(p) = m$ or $e(p) = n$. Suppose that $e(p) = m$; the other case is symmetric. If $p \notin \mathrm{dom}(\alpha)$, we have $m \notin \mathrm{dom}(f)$ and so $\ell \notin \mathrm{dom}(f)$. Otherwise, applying Proposition 4 and recalling that M is a mixture and therefore only has one link at site (m,i), we immediately see again that $\ell \notin \mathrm{dom}(f)$.

Alternatively, if $\ell \notin \langle e,\alpha\rangle^-_{\mathsf{lnk}}$ then $\ell \in \mathcal{L}_{D'}$ and $\beta(m) = m$ and $\beta(n) = n$. Immediately from the definition of h, we have $\ell \in \mathrm{dom}(f)$ iff $\ell \in \mathrm{dom}(h)$, as required.

Properties: Suppose that $(m,i) \in p_{M,k}$. We wish to show that $(m,i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$ iff $(m,i) \in p_{D',k}$ and $(m,i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$. There are two cases to consider.

Firstly, if $(m,i) \in \langle e,\alpha\rangle^-_k$ then $(m,i) \notin p_{D',k}$. There exists n such that $e(n) = m$ and either $n \notin \mathrm{dom}(\alpha)$ or $(n,i) \in p_{L,k} \smallsetminus \mathrm{dom}_{\mathsf{prop},k}(\alpha)$. If $n \notin \mathrm{dom}(\alpha)$ then, since $f \circ e = g \circ \alpha$, we have $m \notin \mathrm{dom}(f)$ and so $(m,i) \notin \mathrm{dom}_{\mathsf{prop},k}(f)$. If $(n,i) \in p_{L,k} \smallsetminus \mathrm{dom}_{\mathsf{prop},k}(\alpha)$ then, since $f \circ e = g \circ \alpha$, we must again have $(m,i) \notin \mathrm{dom}_{\mathsf{prop},k}(f)$.

Secondly, if $(m,i) \notin \langle e,\alpha\rangle^-_k$ then $(m,i) \in p_{D',k}$ and $\beta(m) = m$. Immediately from the definition of h, we have $(m,i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$ iff $\ell \in \mathrm{dom}_{\mathsf{prop},k}(h)$, as required.

h is unique: Let h' be any (partial) morphism such that $h' \circ \beta = f$ and $h' \circ u = g$.

Agents: For $n \in \mathcal{A}_{D'}$, we have $\beta(n) = n$, so

$$h'\beta(n) = h'(n) = f(n) = h(n).$$

For $n \in \langle\alpha,\phi\rangle^+_{\mathsf{ag}}$, noting that $n = u(\phi^{-1}(n))$, we have

$$h'(n) = h'u\phi^{-1}(n) = g(\phi^{-1}(n)) = h(n).$$

Sites: Let $(n,i) \in \mathcal{S}_N$. The case is trivial if $n \in \mathcal{A}_{D'}$. Suppose that $n \in \langle\alpha,\phi\rangle^+_{\mathsf{ag}}$. There is no $m \in \mathrm{dom}(\alpha)$ such that $\alpha(m) = \phi^{-1}(n)$. Hence, since α is an action and therefore creates agents with all sites, we have $(\phi^{-1}(n),i) \in \mathcal{S}_R$. By commutation,

$$(\phi^{-1}(n),i) \in \mathrm{dom}(g) \iff (n,i) \in \mathrm{dom}(h'),$$

so $(n,i) \in \mathrm{dom}(h')$ iff $(n,i) \in \mathrm{dom}(h)$.

Links: Suppose that $\ell = ((m,i),(n,j)) \in \mathcal{L}_N$. We must show that $\ell \in \mathrm{dom}(h)$ iff $\ell \in \mathrm{dom}(h')$. Either $\ell \in \mathcal{L}_{D'}$ or $\ell \in \langle\alpha,\phi\rangle^+_{\mathsf{lnk}}$

First suppose that $\ell \in \mathcal{L}_{D'}$. We have $\ell \in \mathrm{dom}(h')$ iff $\ell \in \mathrm{dom}(f)$ since $f = h' \circ \beta$. From the definition of h, we immediately obtain $\ell \in \mathrm{dom}(h')$ iff $\ell \in \mathrm{dom}(h)$.

Now suppose that $\ell \in \langle\alpha,\phi\rangle^+_{\mathsf{lnk}}$. There exist p,q such that $u(p) = m$ and $u(q) = n$ and $((p,i),(q,j)) \in \mathcal{L}_R$. Since u is a matching, it is injective, and so p and q are unique such that $u(p) = m$ and $u(q) = n$. Furthermore, u is total and so trivially $((p,i),(q,j)) \in \mathrm{dom}(u)$. By $g = h' \circ u$ we therefore have $((p,i),(q,j)) \in \mathrm{dom}(g) \iff \ell \in \mathrm{dom}(h')$. From the definition of h, we immediately obtain $\ell \in \mathrm{dom}(h)$ iff $\ell \in \mathrm{dom}(h')$, as required.

Properties: Suppose that $(n,i) \in p_{N,k}$. We must show that $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h')$. Either $(n,i) \in p_{D',k}$ or $(n,i) \in \langle\alpha,\phi\rangle^+_k$.

First suppose that $(n,i) \in p_{D',k}$. We have $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h')$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(f)$ since $f = h' \circ \beta$. From the definition of h, we immediately obtain $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h')$.

Now suppose that $(n,i) \in \langle\alpha,\phi\rangle^+_k$. There exists m such that $u(m) = n$ and $(m,i) \in p_{R,k}$. Since u is a matching, it is injective and so m is unique such that $u(m) = n$. Furthermore, u is total and so trivially $(m,i) \in \mathrm{dom}_{\mathsf{prop},k}(u)$. By $g = h' \circ$

u, we have $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(g)$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h')$. From the definition of h, we again immediately obtain $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h)$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(h')$, as required. □

APPENDIX D
COMPRESSION: PROOFS

The proofs on Mazurkiewicz compression are standard — see *e.g.* [8].

Lemma 9 (Lemma 2)**.** *Let $\beta_1 : M_1 \rightharpoonup M_2$ and $\alpha_2 : L_2 \rightharpoonup R_2$ and there be a matching $e_2 : L_2 \rightarrowtail M_1$. Let M_3 be a pushout of α_2 against e_2 with pushout morphisms $\beta_2 : M_2 \rightharpoonup M_3$ and $u_2 : R_2 \rightarrowtail M_3$. If there is a matching $e_2' : L_2 \rightarrowtail M_1$ such that $\beta_1 \circ e_2' = e_2$ then, letting M_2' be the pushout of α_2 against e_2' with pushout morphisms $\beta_2' : M_1 \rightharpoonup M_2'$ and $u_2' : R_2 \rightarrowtail M_2'$, there is a unique morphism $\gamma_2 : M_2' \to M_3$ such that $\gamma_2 \circ \beta_2' = \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' = u_2$. Furthermore, γ_2 is an action map.*

Proof: It is simple to see that the outer square in the following diagram commutes, so, from the characterization of M_2' as a pushout in Theorem 1, we immediately obtain a unique morphism γ_2 such that $\gamma_2 \circ \beta_2' = \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' = u_2$.

$$\begin{tikzcd}
& L_2 \arrow[dl, "\alpha_2"'] \arrow[dr, "e_2'"] & \\
R_2 \arrow[dr, "u_2'"] \arrow[ddr, bend right, "u_2"'] & & M_1 \arrow[dl, "\beta_2'"'] \arrow[ddl, bend left, "\beta_2\circ\beta_1"] \\
& M_2' \arrow[d, dotted, "\gamma_2"] & \\
& M_3 &
\end{tikzcd}$$

The proof that the map is an action is a straightforward, concrete analysis of the map constructed in the proof of Theorem 1. □

Lemma 10 (Lemma 3)**.** *Let $\beta_1 : M_1 \rightharpoonup M_2$ and $\alpha_2 : L_2 \rightharpoonup R_2$ and there be a matching $e_2 : L_2 \rightarrowtail M_1$. Let M_3 be a pushout of α_2 against e_2 with pushout morphisms $\beta_2 : M_2 \rightharpoonup M_3$ and $u_2 : R_2 \rightarrowtail M_3$. If there is a matching $e_2' : L_2 \rightarrowtail M_1$ such that $\beta_1 \circ e_2' \sim_{\mathsf{ag}} e_2$ then, letting M_2' be the pushout of α_2 against e_2' with pushout morphisms $\beta_2' : M_1 \rightharpoonup M_2'$ and $u_2' : R_2 \rightarrowtail M_2'$, there is a morphism $\gamma_2 : M_2' \to M_3$, unique up to $\sim_{\mathsf{ag}}$, such that $\gamma_2 \circ \beta_2' \sim_{\mathsf{ag}} \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' \sim_{\mathsf{ag}} u_2$. Furthermore, γ_2 is an action map.*

Proof: Without loss of generality, it is sufficient to take the pushouts M_2' and M_3 to be as presented in Section V.

Define the partial morphism $\gamma_2 : M_2' \to M_3$ as follows.

- *Agents:* For $n \in \mathrm{dom}(\beta_2')$, define $\gamma_2(n) = \beta_2 \circ \beta_1(n)$. For $n \notin \mathrm{dom}(\beta_2')$, define $\gamma_2(n) = u_2(m)$ for the unique m such that $e_2'(m) = n$.
- *Sites:* For $(n,i) \in \mathcal{S}_{M_2'}$, define $(n,i) \in \mathrm{dom}(\gamma_2)$ iff $n \in \mathrm{dom}(\gamma_2)$.
- *Links:* For $((n,i),x) \in \mathrm{dom}(\beta_2')$, define $((n,i),x) \in \mathrm{dom}(\gamma_2)$ iff $((n,i),x) \in \mathrm{dom}(\beta_2 \circ \beta_1)$. For $((n,i),x) \notin \mathrm{dom}(\beta_2')$, define $((n,i),x) \in \mathrm{dom}(\gamma_2)$ always.
- *Properties:* For $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\beta_2')$, define $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\gamma_2)$ iff $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\beta_2 \circ \beta_1)$. For $(n,i) \notin \mathrm{dom}_{\mathsf{prop},k}(\beta_2')$, define $(n,i) \in \mathrm{dom}_{\mathsf{prop},k}(\gamma_2)$ always.

It is straightforward to show that this is well-defined and a partial morphism, as it is to show that it is to show uniqueness and that $\gamma_2 \circ \beta_2' \sim_{\mathsf{ag}} \beta_2 \circ \beta_1$ and $\gamma_2 \circ u_2' \sim_{\mathsf{ag}} u_2$.

Since M_2' and M_3 are mixtures, the only non-trivial requirement in showing that γ_2 is an action is partial injectivity. The proof proceeds by contradiction, and the only interesting case is when we suppose that there exist $m \in \mathrm{dom}(\beta_2')$ and $m' \in \langle \alpha_2, \phi_2' \rangle^+_{\mathsf{ag}}$ such that $\gamma_2(m) = \gamma_2(m')$ (both defined), where ϕ_2' is the bijection giving the identities of the newly-created elements in the pushout M_2'. By definition, $\gamma_2(m) = \beta_2(\beta_1(m))$, so $\gamma_2(m) \in \mathrm{dom}(\beta_2)$. However, since $m' \in \langle \alpha_2, \phi_2' \rangle^+_{\mathsf{ag}}$, there exists n such that $u_2'(n) = m$ and there is no $p \in \mathrm{dom}(\alpha_2)$ s.t. $\alpha_2(p) = n$. We arrive at a contradiction since $e_2(p) = \gamma_2(m')$ by definition, but e_2 must take send such elements outside the image of β_2. □