Stochastic mechanics of graph rewriting

Citation for published version:

Digital Object Identifier (DOI):
10.1145/2933575.2934537

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
LICS '16 Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Stochastic mechanics of graph rewriting

Nicolas Behr
University of Edinburgh
LFCS
nicolas.behr@ed.ac.uk

Vincent Danos
École Normale Supérieure de Paris
Département d’informatique
vincent.danos@ens.fr

Ilias Garnier
University of Edinburgh
LFCS
ilias.garnier@ed.ac.uk

Abstract

We propose an algebraic approach to stochastic graph-rewriting which extends the classical construction of the Heisenberg-Weyl algebra and its canonical representation on the Fock space. Rules are seen as particular elements of an algebra of “diagrams”: the diagram algebra \( \mathcal{D} \). Diagrams can be thought of as formal computational traces represented in partial time. They can be evaluated to normal diagrams (each corresponding to a rule) and generate an associative unital non-commutative algebra of rules: the rule algebra \( \mathcal{R} \). Evaluation becomes a morphism of unital associative algebras which maps general diagrams in \( \mathcal{D} \) to normal ones in \( \mathcal{R} \). In this algebraic reformulation, usual distinctions between graph observables (real-valued maps on the set of graphs defined by counting subgraphs) and rules disappear. Instead, natural algebraic structures of \( \mathcal{R} \) arise: formal observables are seen as rules with equal left and right hand sides and form a commutative subalgebra, the ones counting subgraphs forming a sub-subalgebra of identity rules. Actual graph-rewriting is recovered as a canonical representation of the rule algebra as linear operators over the vector space generated by (isomorphism classes of) finite graphs. The construction of the representation is in close analogy with and subsumes the classical (multi-type bosonic) Fock space representation of the Heisenberg-Weyl algebra.

This shift of point of view, away from its canonical representation to the rule algebra itself, has unexpected consequences. We find that natural variants of the evaluation morphism map give rise to concepts of graph transformations hitherto not considered. These will be described in a separate paper \([2]\). In this extended abstraction we limit ourselves to the simplest concept of double-pushout rewriting (DPO). We establish “jump-closure”, i.e. that the subalgebra \( \mathcal{R} \) arises from rules with equal left and right hand sides and form a commutative subalgebra, the ones counting subgraphs forming a sub-subalgebra of identity rules. Actual graph-rewriting is recovered as a canonical representation of the rule algebra as linear operators over the vector space generated by (isomorphism classes of) finite graphs. The construction of the representation is in close analogy with and subsumes the classical (multi-type bosonic) Fock space representation of the Heisenberg-Weyl algebra.

1. Introduction

Graphs and graph-like structures are basic components in the modern toolkit of modeling. They appear in varied situations such as the study of epidemics, social dynamics of opinions, ad hoc networks, spin glasses \([10]\), and also combinatorial chemical reaction networks \([6]\). Oftentimes, one has competing rewiring operations or rules which locally remodel the graph and, thus, naturally define a Markovian process on the discrete set of graphs. This is the situation we are interested in this paper. Our specific goal is to establish a new route to the study of these models. Traditionally one uses graph transformation systems and the notion of rule and rule application (see e.g. Ref. \([11]\) for a review). Here we posit as our primary object a notion of rule diagrams. Such diagrams can be seen as formal compositions of rules in “true concurrency” style. Operationally, diagrams can also be understood as neighbourhods of realizations of processes of interest. We put together a formalism to represent such diagrams and their evaluations. With this algebraisation of rule composition, the world of rules becomes autonomous – rules can be formally composed using the diagram algebra and then evaluated to linear combinations of rules by means of a specific evaluation mechanism. Four different variants of evaluation are conceivable \([2]\). We restrict here to the simplest form, DPO-rewriting, where no implicit edge-deletion is allowed when deleting a node. We find that rules form a unital associative algebra \( \mathcal{R} \), while (formal) graph observables are just special rules which form a commutative subalgebra of \( \mathcal{R} \). The vector space of finite graphs comes back into the picture as the carrier of a natural representation of the rule algebra. Actual DPO-type graph rewriting is now seen as the action induced by the representation.

Ideas presented here are anticipated by Löwe \([12]\) with the concept of rule composition for single-pushout rewriting (SPO). Diagrams themselves are implicit in recent constructions on graph-rewriting traces \([7]\). But it is only by decoupling the algebra of rules from its representations that we can operationalise these ideas and develop an efficient and versatile combinatorial framework for quantitative graph rewriting. Indeed, our construction embodies a combinatorial engine for accurate handling of the many counting situations which arise in the manipulation of graph rewriting systems. In particular, a special case of this construction is that of discrete typed graphs (no edges): \( \mathcal{R} \) then boils down to the Heisenberg-Weyl algebra, and \( \mathcal{R} \)’s representation to the traditional interpretation of this algebra as acting on the multi-type Fock space. Our combinatorial engine thus subsumes analytic combinatorics on the Fock space \([4]\). Another type of combinatorial scenario we can put our engine to work on is the derivation of the formal backward equation for graph observables. Such equations are widely used in the study of stochastic graph models and often obtained via ad hoc counting arguments. A recent example is Ref. \([11]\) p21]. The derivation relies on a re-derivation of the jump-closure theorem. Not only do we find a much cleaner derivation, but it also generalises in a
straightforward manner to obtain a compact formula for the case of correlators of observables (a.k.a. multivariate moments). Besides, and this is a more subtle difference, we derive jump-closure for DPO-rewriting which is more intricate than the SPO-version obtained earlier [8].

Here is an outline of the following sections:

Acknowledgments. The authors acknowledge support by the Advanced ERC grant RULE/320823. The reviewers are warmly thanked for their insightful comments.

2. Relations and Graphs

We work with relations over finite sets. The set of relations between the sets $A$ and $B$ will be denoted by $\text{Rel}(A, B)$; the subset of one-to-one relations between $A$ and $B$ (equivalently partial injective maps from $A$ to $B$) will be denoted by $\text{Rel}_1(A, B)$. The domain and codomain of a relation $r$ are defined as $\text{dom}(r) = \{a | \exists b. (a, b) \in r\}$ and $\text{cod}(r) = \{b | \exists a. (a, b) \in r\}$.

Given $r$ in $\text{Rel}(A, B)$, and $U \subseteq A$, we write $r[U] := \{b | \exists u \in U. (u, b) \in r\} \subseteq B$. The identity relation in $\text{Rel}(A, A)$ will be written $\text{id}_A$. The sequential composition of $r$ in $\text{Rel}(A, B)$ and $s$ in $\text{Rel}(B, C)$ will be written $rs$ in $\text{Rel}(A, C)$. The Kleene closure of $r$ in $\text{Rel}(A, A)$ will be written $r^\ast$ in $\text{Rel}(A, A)$, and we write $r^+ \ast$ for $r^r \ast r^\ast$. We also define $\text{cl}_\ast(r)$ to be the equivalence relation generated by $r$ in $\text{Rel}(A, A)$, and $r^{-1}$ in $\text{Rel}(B, A)$ to be the relation inverse to $r$ in $\text{Rel}(A, B)$.

We also use directed graphs defined as tuples $G = (V, E, s, t)$ where $V$ and $E$ are finite sets of vertices and edges, and $s, t : E \rightarrow V$ map edges to their source and target.

The connected component relation $\text{cc}(G) \subseteq V \times V$ is the equivalence relation $\text{cl}_\ast(\{(s(e), t(e)) | e \in E\})$. A partial injective morphism of graphs from $(V, E, s, t)$ to $(V', E', s', t')$ is a pair of one-to-one relations $f_V \in \text{Rel}_1(V, V')$, $f_E \in \text{Rel}_1(E, E')$ such that $f_V \circ s = s' \circ f_V$ and $f_E \circ t = t' \circ f_V$. When $f_V$ and $f_E$ are bijections, one says that $(f_V, f_E)$ is an isomorphism. The set of isomorphism classes of finite graphs will be denoted $G_{\equiv}$.

3. The rule diagram and rule algebra

We introduce rule diagrams, a syntax for formal and "truly concurrent" traces of graph rewriting systems. These diagrams admit a notion of composition, which encompasses the usual notion of matching, and a notion of normalization, which implements rewriting. These diagrams and their normal forms span algebras which in the next sections will be the basis for an interpretation of stochastic graph rewriting systems as representations.

3.1 Polarized discrete diagrams

Rule diagrams and their reduction semantics are defined in terms of simpler polarized discrete diagrams (pdds) that correspond to traces of set rewriting ([11], Sec. 2) processes. We will denote the set of discrete diagrams, defined below, by $D_0$.

Definition 1 (Polarized discrete diagram). A pdd is a tuple $d = (i, o, r, m)$, where $i$ and $o$ are finite, disjoint input and output sets, and where $r$ in $\text{Rel}(i, o)$, $m$ in $\text{Rel}(o, i)$ will be called the rule and the match relations, respectively. We require the pdd to be acyclic. Formally, this corresponds to requiring that $\text{id}_i \cap (r; m)^+ = \emptyset$ and symmetrically, $\text{id}_o \cap (m; r)^+ = \emptyset$. A pdd is normal whenever $m = \emptyset$.

The input and output sets should be thought of as vertices on which some finite set of rules operate. These rules (grouped in the rule relation) are themselves strung together along the match relation. Pdds admit a simple graphical syntax that we now illustrate on small examples. In the pictures that follow, inputs will be depicted as $\circ$, outputs as $\bullet$, the rule relation as dotted arrows and the match relation as full arrows. The acyclicity of pdds induces a partial order on elements that we will interpret as the global arrow of time – the diagrams will be displayed vertically, with time going upwards. Besides the empty pdd $d_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$, the simplest examples correspond to the creation, annihilation and preservation of a vertex, corresponding to normal pdds $d_c$, $d_a$, $d_p$, respectively:

\[ d_c \quad d_a \quad d_p \]

More concretely, $d_c := (\emptyset, \bullet, \emptyset, \emptyset)$, $d_a := (\{\emptyset\}, \emptyset, \emptyset, \emptyset)$ and $d_p := (\{\emptyset\}, \bullet, (\{\emptyset\}), \emptyset)$. A rule that matches a vertex and creates another one can be presented as the pdd $d_1 = (\{\emptyset\}, \{\bullet_1, \bullet_2\}, (\{\emptyset_1, \emptyset_2\}), \emptyset)$, displayed here:

\[ \bullet_1 \rightarrow \bullet_2 \]

As an illustration of a non-normal pdd, we can compose (as will be made precise in Prop. [6]) two instances of the previous pdd, for example matching the top instance’s input to $\bullet_2$:

\[ \bullet_4 \rightarrow \bullet_5 \]

The information of this pdd is encoded concretely as follows:

\[ d_2 = (\{\emptyset_0, \emptyset_3\}, \{\bullet_1, \bullet_2, \bullet_4, \bullet_5\}, (\{\emptyset_0, \bullet_2\}, (\emptyset_3, \emptyset_5), (\{\bullet_2, \emptyset_3\})) \]

These examples hint at the fact that pdds are composed of a union of alternating sequences of elements of $i$ and $o$ (as the sequence $(\emptyset_0, \bullet_1, \emptyset_2, \bullet_3, \emptyset_4, \bullet_5, \emptyset_6)$ in [5]) corresponding to the history (that we call worldline) of some element during the rewriting process. This trivial consequence of the choice of one-to-one relations for the rule and match relations together with acyclicity of pdds is pivotal in the definition of rule diagrams.

Definition 2 (Worldlines). Let $d = (i, o, r, m)$ be a pdd. We define the worldline relation $\omega(d)$ in $\text{Rel}(i \cup o, i \cup o)$ as $\text{cl}_\ast(r \cup m)$.

Informally, $\omega(d)[\{x\}]$ is the connected component of $x$ in $d$ seen as a bipartite graph. Equivalently, by the one-to-oneness and acyclicity conditions, we can describe $\omega(d)$ as follows:
Lemma 3. Let \( d = (i, o, r, m) \) be a pdd and \( x, y \in i \cup o \) such that \((x, y) \in \omega(d)\). Then:

\[
\begin{align*}
\text{x, y} \in i & \Rightarrow (x, y) \in (r; m)^* \cup (r; m)^*-1 \\
\text{x, y} \in o & \Rightarrow (x, y) \in (m; r)^* \cup (m; r)^*-1 \\
\text{x} \in i, \text{y} \in o & \Rightarrow (x, y) \in r; (m; r)^* \cup (m; r)^*-1 \\
\text{x} \in o, \text{y} \in i & \Rightarrow (x, y) \in m; (r; m)^* \cup (r; m)^*-1
\end{align*}
\]

A pdd also has a natural notion of interface, namely the unmatched elements of \( i \) and \( o \).

Definition 4 (Interface of a pdd). Let \( d = (i, o, r, m) \) be a pdd. We define its input interface by \( I(d) := \text{dom}(m) \) and its output interface by \( O(d) := \text{dom}(m) \).

For example, the diagram \( d_2 \) given in (3) has \( I(d_2) = \{\emptyset_0\} \), and \( O(d_2) = \{A, B, \emptyset_2, \emptyset_3\} \).

Using interfaces, we can define matches between pdds, and compose pdds along a given match on their interfaces:

Definition 5 (Diagram matches). Let \( d = (i, o, r, m) \), \( d' = (i', o', r', m') \) be disjoint pdds; a diagram match \( n \) from \( d \) to \( d' \) is an element of \( \text{Rel}_1(O(d), I(d')) \). We write \( M_0(d, d') \) for the set of such diagram matches. For \( n \) in \( M_0(d, d') \), we define a composed pdd \( d \rhd_n d' \) as

\[
d \rhd_n d' := (i \cup i', o \cup o', r \cup r', m \cup m' \cup n)
\]

Continuing the previous example, diagram \( d_2 \) in (3) corresponds to the composition of \( d_1 \) with \( d'_1 = \{\emptyset_3, \{A, B, \emptyset_2, \emptyset_3\}, \emptyset\} \) along the match relation \( m = \{\emptyset_2, \emptyset_3\} \).

Proposition 6. Let \( d, d', d'' \) be pdds. Then there exists a bijection \( \alpha_{d,d',d''} : A \rightarrow B \) between the two sets:

\[
A = \{(n, n') | n \in M_0(d, d'), n' \in M_0(d \rhd_n d', d'')\}
\]
\[
B = \{(w, w') | w \in M_0(d', d''), w' \in M_0(d, d' \rhd_w d'')\}
\]

such that, for all \((w, w') = \alpha_{d,d',d''}(n, n')\),

\[
d \rhd_n d' \rhd_w d'' = d \rhd_w (d' \rhd_w d'')
\]

Proof. Let \( d \rhd_n d' \rhd_w d'' \) be a composite. By definition, \( O(d \rhd_n d') \supseteq O(d') \supseteq O(d_0) \supseteq \emptyset \), therefore \( n' \) uniquely decomposes as \( n' = n_0 \cup n_1 \) where \( n_0 \in \text{Rel}_1(O(d_0), \text{dom}(m), I(d')) \) and \( n_1 \in \text{Rel}_1(O(d'), I(d')) \). One then obtains the sought identity by setting \( w = n \cup n_0 \) and \( w' = n_1 \). The converse computability yields the claimed bijection.

Observe that acyclicity and one-to-oneness of the relations \( r \) and \( m \) imply trivially that the intersection of the input interface with a worldline equivalence class is at most a singleton, and similarly for the output interface. This motivates the definition of the boundary relations, which relate elements of a pdd with the interfaces through the worldline relation:

Definition 7 (Boundary relations). Let \( d = (i, o, r, m) \) be given. We define the input boundary relation \( \mathcal{L}_i(d) \) in \( \text{Rel}_1(i \cup o, I(d)) \), and the output boundary relation \( \mathcal{L}_o(d) \) in \( \text{Rel}_1(i \cup o, O(d)) \):

\[
\begin{align*}
\mathcal{L}_i(d) & := \omega(d) \cup \text{id}(d) \\
\mathcal{L}_o(d) & := \omega(d) \cup \text{id}(d)
\end{align*}
\]

The notion of normalization that we apply to pdds corresponds to taking the trace of the worldline relation against the interface of a diagram:

Definition 8 (Normalization). We define

\[
\mathcal{C}(d) := (I(d), \mathcal{O}(d), \mathcal{L}_i(d), \mathcal{L}_o(d), \emptyset).
\]

Lemma 9. Let \( d \) and \( \mathcal{C}(d) \) be as in Def. \( \mathcal{C} \) One has:

(i) \( \mathcal{C}(d) \) is a pdd;
(ii) \( \mathcal{C}(d) = d \) iff \( d \) is normal;
(iii) \( I(d) = \mathcal{I}(\mathcal{C}(d)) \), \( O(d) = \mathcal{O}(\mathcal{C}(d)) \);
(iv) \( \mathcal{L}_i(d)^{-1}; \mathcal{L}_o(d)^{-1}; \mathcal{O}(d) = \mathcal{L}_i(\mathcal{C}(d))^{-1}; \mathcal{L}_o(\mathcal{C}(d))^{-1}; \mathcal{O}(\mathcal{C}(d))^{-1} \).

Proof. (i), (ii) are clear; (iii) is a consequence of having an empty match relation in \( \mathcal{C}(d) \); (iv) follows from Defs. \( \mathcal{I} \) and \( \mathcal{O} \).

Normalization is compatible with composition:

Proposition 10. Let \( d, d' \) be pdds: (i) \( M_0(d, d') = M_0(\mathcal{C}(d), \mathcal{C}(d')) \) and (ii) for all \( n \) in \( M_0(d, d') \), \( \mathcal{C}(d \rhd_n d') = \mathcal{C}(d) \rhd_n \mathcal{C}(d) \).

Proof. (i) By Lemma \( \mathcal{I} \) interfaces are preserved by reduction therefore \( M_0(d, d') = M_0(\mathcal{C}(d), \mathcal{C}(d')) \). (ii) It is sufficient to check equality of the reduced rule relation. Observe that

\[
\begin{align*}
\mathcal{L}(d \rhd_n d' & \rhd_w d'') = \mathcal{L}(d) \cup \mathcal{L}(d' \rhd_w d'') \cup \mathcal{L}(d' \rhd_w d'') \\
\mathcal{O}(d \rhd_n d' & \rhd_w d'') = \mathcal{O}(d) \cup \mathcal{O}(d') \cup \mathcal{O}(d'')
\end{align*}
\]

Let \( R = \mathcal{L}(d)^{-1} \cup \mathcal{O}(d) \) and \( R' = \mathcal{L}(d')^{-1} \cup \mathcal{O}(d') \) be the “interior” worldline relations. Since \( d \) and \( d' \) are disjoint, we can expand the rule structure of \( \mathcal{C}(d \rhd_n d') \) as follows:

\[
\begin{align*}
\mathcal{L}(d \rhd_n d' & \rhd_w d'') \cup \mathcal{L}(d \rhd_n d') \cup \mathcal{O}(d \rhd_n d') \\
\mathcal{O}(d \rhd_n d' & \rhd_w d'') \cup \mathcal{O}(d \rhd_n d') \cup \mathcal{O}(d \rhd_n d')
\end{align*}
\]

By Lemma \( \mathcal{I} \) iv, \( R = \mathcal{L}(\mathcal{C}(d \rhd_n d'))^{-1} \cup \mathcal{L}(\mathcal{C}(d \rhd_n d')) \) and \( R' = \mathcal{O}(\mathcal{C}(d \rhd_n d'))^{-1} \cup \mathcal{O}(\mathcal{C}(d \rhd_n d')) \), so we can substitute these in the expansion above, and fold the expansion back to conclude.

For the construction of the rule and rule diagram algebras, we need to consider diagram only up to isomorphisms:

Definition 11 (Isomorphism of pdds). Let \( d = (i, o, r, m) \) and \( d' = (i', o', r', m') \) be pdds. An isomorphism from \( d \) to \( d' \) is a pair of bijections \( f = (f_1 : i \rightarrow i', f_2 : o \rightarrow o') \) such that \( f_1 \circ r = r \circ f_2 \) and \( m_1 = f_2 \circ m \). This will be denoted by \( f : d \cong d' \).

3.2 Rule diagrams

A rule diagram is a suitable coupling of a pair of pdds supported by respectively a set of vertices and a set of edges. The set of rule diagrams will be denoted by \( \mathcal{D} \).

Definition 12 (Rule pre-diagram). A rule pre-diagram is a tuple \( d = (dv, dv, s_i, t_i, s_o, t_o) \) where \( dv' = (iv, ov, rv, mv) \) and \( dv = (iv, ov, rv, mv) \) are pdds verifying:

1. \( G_i(d) := (iv, ie, s_i, t_i) \) and \( G_o(d) := (ov, oe, s_o, t_o) \) are graphs;
2. \( (rv, re) \) is a partial graph morphism from \( G_i(d) \) to \( G_o(d) \).

Conditions 1 and 2 can be summarized in the following diagram, where the horizontal components are pdds and the vertical ones are graphs:
Definition 13 (Rule diagram). A rule diagram is a rule pre-diagram $d$ (same notations as above) such that:

3. $d$ fulfills the delayed morphism condition: if $(e, e')$ is in $\omega(d_E)$, then $(s(e), s(e'))$ and $(t(e), t(e'))$ are in $\omega(d_V)$, with $s = s_1 \circ \omega s_0$ and $t = t_1 \circ \omega t_0$.
4. $d$ is globally acyclic: i.e. $id_{\omega(d_E)} \cap (\nu_V; \tau_V)' = \emptyset$ with $\tau_V' = \text{ec}(G_1(d)) \cap \nu_V$.
5. $d$ verifies the totality condition, meaning $\text{id}_{\omega(d_E)}; s_1; \omega(s_0(d_V))$ and $\text{id}_{\omega(d_E)}; s_0; \omega(s_1(d_V))$ are total functions from $\omega(d_E)$ to $Z(d_V)$ and to $O(d_V)$, similarly for $t_1$ and $t_0$.

We say a rule diagram $d$ is normal, or simply that it is a rule, when both $d_V$ and $d_E$ are normal pdls. In which case, we simply write $d = G_0(d) = G_1(d)$, with $r = (\tau_V, \tau_E)$ the partial injective graph morphism from $G_1(d)$ to $G_0(d)$ (see condition 2 above). Rules are isomorphic to the traditional concept of rules in graph rewriting.

Let us discuss the conditions listed in Def. 13. The delayed morphism condition is key to proving that composition of rule diagrams is associative (Prop. 19). Global acyclicity allows one to have a sequential interpretation of rule diagrams as compositions of rules (Def. 17). The totality condition enforces that the source and target maps in any normalized diagram are total functions. Note that it is possible to drop this axiom from the definition of rule diagrams at the cost of having to define a more complicated form of reduction, but at the benefit of being able to implement the algebraic structures for types of graph rewriting more general than DPO.

As a first example, we consider a normal diagram corresponding to a rule which acts identically on an edge, see (4). We use the same pictorial conventions as for pdls for vertices and we use $\circ$ to denote input edges and $\ast$ for output edges. The concrete presentation for this diagram is $d = (d_V, d_E, s_1, t_1, s_0, t_0)$ with $d_V = \{(c_0, \ast_0), \{(\ast_0, \ast_1), (\ast_1, \ast_2)\}, \emptyset\}$ for vertices, $d_E = \{(\ast_0, \ast_1), \{(\ast_0, \ast_2)\}, \emptyset\}$ for edges and the obvious maps for $s_1, t_1, s_0, t_0$. Rule diagrams admit a notion of isomorphism:

Definition 14 (Isomorphism of rule diagrams). Let $d = (d_V, d_E, s_1, t_1, s_0, t_0)$ and $d' = (d'_V, d'_E, s'_1, t'_1, s'_0, t'_0)$ be rule diagrams. An isomorphism from $d$ to $d'$ is a pair of discrete diagram isomorphisms (Def. 7)

\[ f_V = (f_{V,1}, f_{V,0}) : d_V \cong d'_V, \quad f_E = (f_{E,1}, f_{E,0}) : d_E \cong d'_E \]

such that $(f_{V,1}, f_{V,0})$ is a graph isomorphism from $G_1(d)$ to $G_1(d')$ and $(f_{E,1}, f_{E,0})$ is a graph isomorphism from $G_0(d)$ to $G_0(d')$. The set of isomorphism classes of rule diagrams will be denoted $D_E$. Isomorphism classes of rules diagrams will be denoted $R_E$.

Normalization of rule diagrams is defined as the componentwise normalization of the vertices and edges pdls:

Definition 15 (Normalization of rule diagrams). Let us consider a rule diagram $d = (d_V, d_E, s_1, t_1, s_0, t_0)$. We define its normal form as $\check{d}(d) := (\check{v}(d_V), \check{v}(d_E), \check{s}_0, \check{s}_1, \check{t}_0, \check{t}_1)$ where

\[ \check{s}_i := \text{id}_{\omega(d_E)}; s_i; \omega(s_0(d_V)), \quad \check{s}_0 := \text{id}_{\omega(d_E)}; s_0; \omega(s_1(d_V)) \]
\[ \check{t}_i := \text{id}_{\omega(d_E)}; t_i; \omega(t_0(d_V)), \quad \check{t}_0 := \text{id}_{\omega(d_E)}; t_0; \omega(t_1(d_V)) \]

The following proposition states that normalization preserves the structure of rule diagrams:

Proposition 16. (i) For all $d \in D$, $\check{d}(d)$ is a rule, and (ii) $\check{d} \circ \check{d} = \check{d}$.

Proof. (i) Let us show that the conditions listed in Def. 13 are verified by $\check{d}(d)$.

5. Totality is guaranteed by construction.

1. Totality directly implies that $G_1(\check{d}(d))$ and $G_0(\check{d}(d))$ are graphs.

2. Let us prove that the rule structure $(\tau_V', \tau_E')$ of $\check{d}(d)$ induces a partial graph morphism. Let $(e, e_0) \in \tau_E$; then by virtue of the delayed morphism condition, $(s(e), s_0(e_0)) \in \omega(d_V)$, therefore $(\check{s}(\check{e}), \check{s}_0(\check{e}_0)) \in \omega(d_V)$. By Def. 13 this pair of vertices is in $r_V'$. The same argument holds for $t_0$.

3. By the previous point and using that the match relation is empty, the delayed morphism condition is trivially verified.

4. Global acyclicity follows trivially by emptiness of the match relation of $\check{d}(d)$.

(ii) follows trivially from normality of $\check{d}(d)$.

As for pdls, rule diagrams admit a notion of binary composition along a match:

Definition 17 (Composition). Consider two disjoint rule diagrams

\[ d = (d_V, d_E, s_1, t_1, s_0, t_0), \quad d' = (d'_V, d'_E, s'_1, t'_1, s'_0, t'_0) \]

and a pair of matches

\[ n = (n_V, n_E) \in M_0(d_V, d'_V) \times M_0(d_E, d'_E) \]

on vertices and edges, respectively. Whenever the object defined by

\[ d \circ_{D, n} d' = (d_V \circ_{D, n_V} d'_V, d_E \circ_{D, n_E} d'_E, s_1 \circ_{E, n} s'_1, t_0 \circ_{E, n} t'_0, s_0 \circ_{E, n} s'_0, t_0 \circ_{E, n} t'_0) \]

is in $D$, we call it the composition of $d$ and $d'$ along $n$.

Example 4 corresponds to a vertex-preserving rule precomposed with an edge-preserving rule.

This example highlights in a striking way the delayed morphism condition: here, $(\ast_0, \ast_0)$ are in the match relation but $(t_0(\ast_0), t_1(\ast_0))$ are not. This makes the following proposition, which characterizes the admissible matches more concretely, not entirely trivial:

Proposition 18 (Admissible matches). Let $d, d', n$ be as in Def. 7. $d \circ_{D, n} d'$ is a rule diagram if and only if (i) $n$ is a partial injective morphism of graphs from $G_0(\check{d}(d))$ to $G_1(\check{d}'(d'))$ and (ii) $d \circ_{n} d'$ verifies the totality condition. Such an $n$ will be called admissible. We denote the set of admissible matches from $d$ to $d'$ by $M(d, d')$.

Proof. Assume $d \circ_{D, n} d'$ is in $D$ with $n = (n_V, n_E)$. First, observe that this implies trivially that $d \circ_{D, n} d' \in D$ verifies the totality condition. Now observe that by construction, $n_V \in \text{Rel}_1(\omega(d_V), Z(d'_V))$ and $n_E \in \text{Rel}_1(\omega(d_E), Z(d'_E))$. Consider $e, e'$ such that $e = n_V(e)$. We have to prove that $n$ is a partial injective morphism of graphs from $G_0(\check{d}(d))$ to $G_1(\check{d}'(d'))$ (see Prop. 16 for their definitions), i.e. that $n_V(\check{s}(e)) = \check{s}'(e')$.

By definition of reduction, $(\check{s}(e), \check{s}_0(e)) \in \omega(d_V)$ and symmetrically, $(\check{s}'(e'), \check{s}'_0(e')) \in \omega(d'_V)$; while the delayed morphism property (dmp) ensures that $(\check{s}_0(e), \check{s}_0(e')) \in \omega(d_V \circ D, n_V d'_V)$. Since $d$ and $d'$ are disjoint and $n$ is by assumption one-to-one, this is only
possible if \( nV(s_0(e)) = s'_1(e') \) is verified. Mirroring this argument for \( t', t'' \) yields the result that \( n \) is a partial injective morphism of graphs.

Conversely, assume \( n \) as stated. We only prove that \( d \triangleright_n d' \) verifies the \( dmp \) (the other defining properties of rule diagrams are easily verified). It is enough to exhibit the \( dmp \) for a pair of edges \((e, e')\) with \( e \in d \) and \( e' \in d' \), otherwise it is satisfied by assumption. Assume that \((e, e') \in \omega(d_E) \triangleright_n E \omega(d_E)\). Using one-to-one-ness and acyclicity of \( dmp \), we obtain that \((e, e') \in \omega(d_E)\). For \( e \in E \omega(d_E) \), there exists \( e_0 \in E \omega(d_E) \) such that \((e, e_0) \in \omega(d_E)\) and \((e, e') \in \omega(d_E)\) and \( e \) is an edge of \( G_1(\omega(d_E))\); also, one can apply the \( dmp \) on \((e, e_0)\) and \((e', e')\), by assumption. The goal is reduced to proving that \((s_0(e_0), s_1(e_1)) \in \omega(d_1 \triangleright_n d_1)\). Since \( d \) and \( d' \) are disjoint, this can only be satisfied if the vertices related by \( \omega(d_1) \) in the interface are related through \( nV \). It is thus enough to prove that

\[
(s_1; \omega(d_1); i_d \omega(d_1); nV)(e_0) = (s_1; \omega(d_1); i_d \omega(d_1); e_1'),
\]

which by definition of \( s \) and \( s' \) corresponds to having \( nV(s_0(e_0)) = s'_1(e_1') \). Since \( e_0 \) is an edge of \( G_1(\omega(d)) \) and \( e'_1 \) is an edge of \( G_1(\omega(d')) \), \( nV(s_0(e_0)) = s'_1(e_1') \). The exact same argument for the target maps \( t_0', t_1' \) concludes the proof. \( \square \)

The following is analogous to Prop. 19:

**Proposition 20.** Let \( d, d', d'' \in D \). There exists a bijection \( \alpha_{d, d', d''} \) between the following two sets:

\[
\{ (n, n') \mid n \in M(d, d'), n' \in M(d \triangleright_n d', d'') \}
\]

\[
\{ (w, w') \mid w' \in M(d', d''), w \in M(d, d' \triangleright_n d'') \}
\]

such that, for all \( (w, w') \in \alpha_{d, d', d''} \),

\[
(d \triangleright_n d') \triangleright_n d'' = d \triangleleft w \triangleright_n (d' \triangleright_n d'').
\]

**Proof:** The source and target maps \( s, t, s_0, t_0 \) of a triple composite are given by the union of the corresponding data from each component, independently of the chosen matches. Therefore, it is enough to apply Prop. 19 to conclude.

Moreover, normalization respects composition:

**Proposition 21.** Let \( d, d' \in D \) and \( n \in M(d, d') \). One has \( n \in M(\tilde{\alpha}(d), \tilde{\alpha}(d')) \) and \( \tilde{\alpha}(d \triangleright_n d') = \tilde{\alpha}(d \triangleright_n d) \).

**Proof:** (i) Using the assumption \( n \in M(d, d') \), by Prop. 18 \( n \) is an injective morphism of graphs from \( G_0(\tilde{\alpha}(d)) \) to \( G_1(\tilde{\alpha}(d')) \) and therefore \( n \in M(\tilde{\alpha}(d), \tilde{\alpha}(d')) \). The proof of (ii) follows the same pattern as the proof of Prop. 10.

### 3.3 The rule diagram and rule algebra

Rule diagrams span a vector space that may be endowed via the composition operation with the structure of an algebra. In the following, we denote by \((\text{span}(X), +, \cdot)\) the formal vector space of finite linear combinations with real coefficients over a set \( X \), where \( v + v' \) is the vector addition, and where \( \lambda \cdot v \) is the scalar multiplication. We let \( \delta : X \rightarrow \text{span}(X) \) be the map associating \( x \in X \) to the basis vector \( \delta(x) \). However, where the context allows it, we will drop \( \delta \) and denote a basis element by its index in \( X \). In the remainder of this paper, we will only deal explicitly with isomorphism classes of combinatorial structures where required.

**Definition 22** (Vector spaces of rule diagrams and rules). Let \( D = (\text{span}(D_0), +, \cdot) \). Since \( R_{D_0} \subseteq D_{D_0} \), there exists a subspace vector of \( D \) spanned by (isomorphism classes of) rules which will be denoted by \( D \), together with a canonical inclusion \( \psi : D \rightarrow D \).

\( D \) admits an algebra structure induced by diagram composition. Let us define the product:

**Definition 23** (Product in \( D \)). Let \( d, d' \in D \) be two basis vectors for \( d, d' \in D_{D_0} \). We define their product as:

\[
\delta(d') \triangleright \delta(d) := \sum_{n \in M(d, d')} \delta(d' \triangleright_n d).
\]

This extends to arbitrary elements of \( D \) by linearity:

\[
\left( \sum_{d'} \beta_d \delta(d') \right) \triangleright \left( \sum_{d''} \alpha_{d, d'} \delta(d') \right) := \sum_{d''} \alpha_{d, d'} \beta_d \delta(d') \triangleright \delta(d)
\]

**Theorem 24.** \( \triangleright \) turns \( D \) into an associative algebra with unit \( 1_D = d_0 \). We call \( (D, \triangleright, 1_D) \) the rule diagram algebra.

**Proof:** Bilinearity of \( \triangleright \) is straightforward. Let us prove associativity. Clearly it is enough to consider basis vectors. We have:

\[
\begin{align*}
(\delta(d') \triangleright \delta(d)) \triangleright \delta(d) &= \sum_{n' \in M(d', d'')} \delta(d' \triangleright_n d') \triangleright \delta(d') \\
&= \sum_{n'' \in M(d, d'')} \sum_{n' \in M(d', d'')} \delta(d' \triangleright_n d') \triangleright \delta(d) \\
&= \sum_{n'' \in M(d, d'')} \sum_{n' \in M(d', d'')} \delta(d' \triangleright_n d'),
\end{align*}
\]

where we applied Prop. 19 in the final step.

Let us check the unit law. Observe that for all \( d \in D \), \( 0 \) is the only element in \( M(d_0, d) \) and in \( M(d, d_0) \), therefore

\[
\delta(d_0) \triangleright \delta(d) = \delta(d_0 \triangleright d) = \delta(d_0) = \delta(d) \triangleright \delta(d_0).
\]

This lifts trivially to arbitrary vectors. \( \square \)

The normalization map extends by linearity to a linear map from \( D \) to \( \mathbb{R} \) that we call the reduction map:

**Definition 25** (Reduction map). The function \( \overline{\phi} \) defined on basis vectors as

\[
\overline{\phi}(\delta(d)) := \overline{\delta}(\tilde{\alpha}(d))
\]

extends straightforwardly to a linear map \( \overline{\phi} : D \rightarrow \mathbb{R} \).

The unital associative algebra structure on \( D \) can be pushed forward to \( \mathbb{R} \) by composing rules and normalizing back their composition:

**Definition 26** (Product in \( \mathbb{R} \)). Let \( v, v' \in \mathbb{R} \) be given. We define their product as:

\[
v' \triangleright \mathbb{R} v := \overline{\phi} \left( \phi(v') \triangleright \mathbb{R} \phi(v) \right)
\]

**Theorem 27.** \( \overline{\phi} \) is a homomorphism of unital associative algebras from \((D, \triangleright, 1_D)\) to \((\mathbb{R}, \triangleright, 1_D) := d_0 \). We call \((\mathbb{R}, \triangleright, 1_D)\) the rule algebra.

**Proof:** Let us first prove that \( \overline{\phi} \) is an algebra homomorphism. By bilinearity of \( \triangleright \) and \( \triangleright \), it is enough to consider basis vectors. We have to prove \( \overline{\phi}(\delta(d') \triangleright \mathbb{R} \delta(d)) = \overline{\phi}(\delta(d')) \triangleright \mathbb{R} \phi(\delta(d)) \). Unfolding the definition of \( \triangleright \), we get:

\[
\overline{\phi}(\delta(d')) \triangleright \mathbb{R} \phi(\delta(d)) = \overline{\phi}(\phi(\tilde{\alpha}(d'))) \triangleright \mathbb{R} \phi(\tilde{\alpha}(d))
\]

therefore, the goal reduces to proving

\[
\overline{\phi}(\delta(d')) \triangleright \mathbb{R} \phi(\delta(d)) = \overline{\phi}(\phi(\tilde{\alpha}(d'))) \triangleright \mathbb{R} \phi(\tilde{\alpha}(d))
\]
Then \( \psi(\varphi(\delta(d))) = \delta(\varphi(d)) \) and \( \psi(\varphi(\delta(d))) = \delta(\varphi(d')) \), and it is sufficient to prove that \( \varphi(\delta(d) \ast \delta(d')) = \varphi(\delta(d') \ast \delta(d)) \), which follows from point 1. of Prop. 21. The unitality part is trivial. 

As for associativity, it suffices to apply the homomorphism property to \( \varphi(\psi(x) \ast \psi(y) \ast \psi(z)) \) in the two possible ways to obtain the sought after equality. 

An important subalgebra of \( R \) is that of observables, which will be denoted by \( G \). Its elements are (linear combinations of) rules \( g' \sim \overset{g'}{\ast} g \) where \( g \) and \( g' \) are isomorphic. In a slight abuse of notation, we will denote such a diagram by \( g \sim \overset{g}{\ast} g \). As we shall see, such diagrams correspond (via the representation to be defined below) to functions on graphs counting the number of occurrences of a given graph. If \( r \) is not an isomorphism, but simply a partial injective morphism, then we are only counting such occurrences where nodes deleted by \( r \) (and recreated subsequently) are sent to nodes of same degree in the target graph.

**Proposition 28.** Let \( \mathcal{O} \) be the linear subspace spanned by the family \( \{ g \sim \overset{g}{\ast} g \}_{g \in G} \); \( \mathcal{O} \) is a commutative subalgebra of \( R \).

**Proof.** Let \( d = (g \sim \overset{g}{\ast} g), d' = (g' \sim \overset{g'}{\ast} g') \) and \( n = (n_1, n_2) \in \mathcal{M}(d, d') \) be given as in Def. 17. By Prop. 18 \( n \) is a partial graph morphism from \( g \) to \( g' \). Let us write \( d = d \triangleright_n d' \). We prove that there is a graph isomorphism from \( G_n(\vec{c}(d)) \) to \( G_n(\vec{c}(d')) \). By assumption, \( G_n(d) \cong G_n(d) \) and \( G_n(d) \cong G_n(d') \), and evidently \( dom(n) \cong cod(n) \). Moreover, 

\[
G_n(\vec{c}(d)) = G_n(\vec{c}(d)) \cup cod(n_A) \cup (G_n(\vec{c}(d')) \setminus cod(n_A)) \\
G_n(\vec{c}(d')) = G_n(\vec{c}(d')) \cup dom(n_B) \setminus (G_n(\vec{c}(d')) \setminus cod(n_B)) \\
A = id_{\triangleright_n}; n_A = n \setminus n_B \\
B = n; id_{\triangleright_n}; n_B = n \setminus n_B.
\]

where the notation \( A \cup B \) denotes gluing graph \( A \) to graph \( B \) along the overlap \( G \). The claim then follows from \( cod(n_A) \cup cod(n_A) \cong dom(n_B) \cup cod(n_B) \). Commutativity of \( \mathcal{O} \) follows from the same argument.

**Definition 29 (Representation map).**

\[
\rho(g' \sim \overset{g'}{\ast} g)(\emptyset) := \begin{cases} 
0 & \text{if } g = \emptyset \\
\rho(g' \sim \overset{g'}{\ast} g)(\rho(g' \sim \overset{g'}{\ast} g)(\emptyset)) & \text{else}
\end{cases}
\]

This extends to a linear operator \( \rho : R \rightarrow End(G) \). Note that the first definition implies the equation \( \rho(g' \sim \overset{g'}{\ast} g)(\emptyset) = \rho(g' \sim \overset{g'}{\ast} g)(\emptyset) \) for all \( g \in G \). We have:

**Theorem 30.** \( \rho \) is a homomorphism of associative unital algebras.

**Proof.** Let \( d, d' \in R \) be given. By linearity, it suffices to prove \( \rho(d' \ast \rho d) = \rho(d' \ast \rho d) \) and \( \rho(1) = \rho(1) \). It is enough to test these equalities on basis vectors of \( G \). We first consider the case of the basis vector \( |0\rangle \) and then proceed to the case of \( |g\rangle \neq |0\rangle \).

By definition of \( \rho \), we trivially have

\[
\rho(1)|0\rangle = \rho(1)|0\rangle = \rho(0)|0\rangle = |0\rangle.
\]

Let us test the homomorphism property on \(|0\rangle\): if \( d \) is not of the form \( d = (g \sim \overset{g}{\ast} g) \), then \( \rho(d') \rho(d)(|0\rangle) = 0 \). Since \( \forall n, G_n(d) \subseteq G_n(d \triangleright_n d') \), one also has \( \rho(d' \ast \rho d)(|0\rangle) = 0 \) and the equality holds. Assume now that \( d \) is of the form \( d = (g \sim \overset{g}{\ast} g) \). Then by definition,

\[
\rho(d' \rho(g \sim \overset{g}{\ast} g)(|0\rangle) = \rho(d' \rho(d \triangleright_n (g \sim \overset{g}{\ast} g)(|0\rangle)) = \rho(d' \rho(d \triangleright_n (g \sim \overset{g}{\ast} g)(|0\rangle)).
\]

Let us proceed to the case of a basis vector \( |g\rangle \neq |0\rangle \). We have due to the previous result \( \rho(1)|g\rangle = \rho(1)(g \sim \overset{g}{\ast} g)(|0\rangle)\) \( |g\rangle \).

Using the previous results together with the associativity of \( \ast \),

\[
\rho(d' \ast \rho d)|g\rangle = \rho(d' \ast \rho d \ast \rho d)(|0\rangle) = \rho(d' \rho(d \triangleright_n (g \sim \overset{g}{\ast} g)(|0\rangle)) = \rho(d') \rho(d \triangleright_n (g \sim \overset{g}{\ast} g)(|0\rangle)).
\]

The following result will be useful in constructing a stochastic dynamics. We recall that an operator \( A \in End(G) \) is row-finite if for all \( h \), there are only finitely many \( g \) such that \( (A|g\rangle)_h \) is nonzero.

**Lemma 31.** \( \rho \) ranges in row-finite operators.

**Proof.** It is enough to consider \( d = (f' \sim f) \). We have to prove that for all \( h \), there are finitely many \( g \) such that \( (\rho(d)|g\rangle)_h \) is nonzero, i.e. such that

\[
\rho((f' \sim f) \ast \rho d)(|g\rangle)_h = \rho((f' \sim f) \ast \rho d)(|g\rangle)_h
\]

has a strictly positive component in \(|h\rangle\). But since \( h \) is a finite graph, there are only finitely many \( g \) and \( n \) such that

\[
\vec{c}((f' \sim f) \sim \overset{g}{\ast} g)(|h\rangle) = |h\rangle\).
\]

**5. Stochastic mechanics of graph rewriting**

**5.1 Stochastic mechanics in a nutshell**

We are interested in describing the time evolution of probability distributions over \( G \). As these are not necessarily finitely supported, they do not fit in \( G = span(G) \). Therefore, we define our space of states to be the real Fréchet space \( \tilde{G} := \mathbb{R}^{G} \) endowed with the product topology. The subspace of finite sequences is isomorphic to \( G \). The convex subset of subprobability states \( \text{Prob} \subset \tilde{G} \) contains all states \( f \) in \( G \) that are (i) positive, i.e. for all \( g \in G \), \( f(g) \geq 0 \) and (ii) subnormalized, i.e. \( \sum g f(g) = 1 \). We say an operator \( A \in \tilde{G} \) is substochastic, if \( A(\text{Prob}) \subset \text{Prob} \). We write \( \text{Stoch}G \) for such operators.
A stochastic dynamics in our setting will be a continuous-time Markov chain given by a Hamiltonian $H$ in $End(\mathcal{G})$. We require this operator to be infinitesimal stochastic, meaning that $H = (h_{g\rightarrow g'})$ for $g, g' \in G_\Sigma$ with: (i) $h_{g\rightarrow g'} \geq 0$ for all $g \neq g'$ and, (ii) $\sum_{g'} h_{g\rightarrow g'} = 0$ for all $g$. The stochastic dynamics induced by a Hamiltonian is a semigroup $P \in [0, \infty) \rightarrow Stoch(G_\Sigma)$ of substochastic operators (i.e. $P(x)P(t) = P(s + t)$ for all $s, t \geq 0$) which is the pointwise minimal non-negative solution of the (backward) master equation:

$$\frac{dP}{dt} = HP$$

(5)

Given an initial state $f$ in $Prob$, the corresponding trajectory is given by $t \mapsto P(t)(f)$. See Norris [13] for a thorough treatment of the general subject. Note that the above only makes formal sense whenever $H \in End(\mathcal{G})$ can be interpreted as an element of $End(\mathcal{G})$. In this paper, as a consequence of Lemma 31, this will always be the case by construction.

**Lemma 32.** For all $H \in End(\mathcal{G})$, if $H$ is row-finite then $H \in End(\mathcal{G})$.

*Proof.* Operators in $End(\mathcal{G})$ must map finite linear combinations to finite linear combinations, therefore they must be column-finite. If such an operator is moreover row-finite, its application is trivially well-defined on all elements of $\mathcal{G}$.

**The projection.** It will be useful to integrate elements of $\mathcal{G}$ against the counting measure. In analogy with the notations of quantum mechanics, we call this the projection and denote this linear (partial) operation by

$$\langle : : \mathcal{G} \rightarrow \mathbb{R} : v \in \mathcal{G} \mapsto \sum_{g \in G_\Sigma} v(g) \rangle.$$

Hamiltonians verify the following special property:

**Lemma 33.** If $H \in End(\mathcal{G})$ is infinitesimal stochastic, $\langle | H = 0$.

*Proof.* By condition (ii) of the definition of infinitesimal stochastic operators, column vectors $H | g \rangle$ of $H$ sum to zero.

### 5.2 Operators for graph observables

The quantities of interest in stochastic graph rewriting-based models are “graph-counting observables”. They correspond to the number of occurrences of some subgraph isomorphic to a pattern $h$ in the graph being rewritten, say $g$ – in other words, the number of injections from $h$ to $g$, denoted by $[h, g]$. In our setting, these quantities are computed by graph-counting operators. A graph observable for a pattern $h \in G_\Sigma$ is an operator $O_h \in End(\mathcal{G})$ which verifies

$$[O_h | g \rangle = | [h, g] | g \rangle,$$

(6)

i.e. every basis vector $| g \rangle$ is an eigenvector with eigenvalue $[h, g]$. Note that one could take this as a definition. However, it will be useful to express these operators in terms of the representations of the elements of the subalgebra of graph observables $\mathfrak{O}$ (elements which are not to be confused with their representations as actual graph observables, see Prop. [23]):

$$O_h := \rho(h \xleftarrow{r_h} g)$$

for some $r_h$

Let us verify that this matches Eq. [6]:

$$\rho(h \xleftarrow{r_h} g) \langle g = \rho(h \xleftarrow{r_h} g) \rho(g \equiv \emptyset) | \emptyset \rangle = \rho(h \xleftarrow{r_h} g) \rho(g \equiv \emptyset) \langle \emptyset \rangle = \rho(h \xleftarrow{r_h} g) \rho(g \equiv \emptyset) \langle \emptyset \rangle$$

(7)

Consider an arbitrary composite $(h \xleftarrow{r_h} h) \xleftarrow{r_n} (g \equiv \emptyset)$ for some admissible match $n \in \mathcal{M}(g \equiv \emptyset, h \xleftarrow{r_h} h)$. By Prop. [18] $n$ must be an injective graph morphism from $g$ to $h$. Assume that $n$ is not surjective, then:

$$\rho(g \rho(h \xleftarrow{r_h} h) \xleftarrow{r_n} (g \equiv \emptyset)) | \emptyset \rangle = 0 | \emptyset \rangle.$$

In other words, the only contributions to Eq. [7] are those where $n$ is an injective and surjective partial map from $g$ to $h$, i.e. an embedding of $h$ in $g$. It follows that

$$\langle \rho(h \xleftarrow{r_h} h) \mid g \rangle = | [h, g] | g \rangle,$$

where $[h, g]_{r_h} \subseteq [h, g]$ for the subset of matches of $h$ in $g$ that are compatible with $r_h$ deletions – meaning each node deleted by $r_h$ (and then recreated) is matched to a node of the same degree. It thus follows that the graph observables typically considered in the graph rewriting literature would be those $\langle \rho(h \xleftarrow{r_h} h) \mid g \rangle$ for which the $r_h$ are isomorphisms.

### 5.3 Hamiltonians for stochastic graph rewriting

We now have all the ingredients required to produce the Hamiltonian corresponding to a stochastic graph rewriting system:

**Proposition 34.** Let $(g_i, r_i \xleftarrow{r_i} g_i; i \in I)$ be a finite family of normal diagrams seen as rules and $\{\kappa_i \in [0, +\infty)\}_{i \in I}$ their associated base rates. Define

$$H = \sum_{i \in I} \kappa_i \rho(g_i \xleftarrow{r_i} g_i) - \rho(g_i \xleftarrow{r_i} g_i),$$

where $g_i, r_i \xleftarrow{r_i} g_i$ is the observable obtained from $g_i \xleftarrow{r_i} g_i$. We have that (i) $H$ is infinitesimal stochastic, (ii) $H \in End(\mathcal{G})$ is row-finite.

We will need the following lemma:

**Lemma 35.** Let $g_1 \xleftarrow{r_1} g, g_2 \xleftarrow{r_2} g$ be rules such that $\text{dom}(r_1) = \text{dom}(r_2)$; one has:

$$\langle \rho(g_1 \xleftarrow{r_1} g) \mid g \rangle = \langle \rho(g_2 \xleftarrow{r_2} g) \mid g \rangle.$$

*Proof.* We start with $| h = \emptyset \rangle$. If $g \neq \emptyset$ then the claim is trivially verified. Let us then assume $g = \emptyset$ (implying $r_1, r_2 = \emptyset$):

$$\langle \rho(g_1 \xleftarrow{r_1} g) \mid \emptyset \rangle = \langle \rho(g_2 \xleftarrow{r_2} g) \mid \emptyset \rangle = 1$$

For $| h \neq \emptyset \rangle$, we have by definition of $\rho$:

$$\langle \rho(g_1 \xleftarrow{r_1} g) \mid h \rangle = \langle \rho(g_1 \xleftarrow{r_1} g) \rho(g_2 \xleftarrow{r_2} g) \mid h \rangle = \sum_n \langle \rho(\phi((g_1 \xleftarrow{r_1} g) \xleftarrow{r_n} (h \equiv \emptyset))) \mid h \rangle \rangle$$

Only admissible matches $n$ which are surjective partial graph morphisms from $h$ to $g$ contribute to this sum. Also, $\mathcal{M}(h \equiv \emptyset, g_1 \xleftarrow{r_1} g) = \mathcal{M}(h \equiv \emptyset, g_2 \xleftarrow{r_2} g)$. Applying reduction we may write:

$$\sum_n \langle \rho(\phi((g_1 \xleftarrow{r_1} g) \xleftarrow{r_n} (h \equiv \emptyset))) \mid h \rangle \rangle = \sum_n \langle \rho(g_1 \xleftarrow{r_1} g) \mid \emptyset \rangle = \sum_n \langle \rho(g_2 \xleftarrow{r_2} g) \mid \emptyset \rangle,$$

where the last line follows by the first case of our analysis. Applying the same reasoning in reverse yields the claim.

We can now prove Prop. [24]:

*Proof.* It suffices to consider the case of one rule $g' \xleftarrow{r'} g$. It is enough to prove that for all $| g \rangle$,

$$\langle \rho(g' \xleftarrow{r'} g) - \rho(g \xleftarrow{r} g) \mid g \rangle = 0,$$
where \( g \mapsto g' \) is the observable obtained from \( g \mapsto g' \). The above statement is thus a straightforward consequence of Lemma 35.

Row-finiteness is a direct consequence of Lemma 37.

### 5.4 Jump-closure for observables

As presented at the beginning of this section, any Hamiltonian (as obtained from Prop. 24) induces a stochastic dynamics, from which one can – in principle – derive all quantities of interest. However, one is typically not interested in the full dynamical system, but only in the expected value of some graph observable (or higher moments thereof). The remainder of this section re-proves and extends in our algebraic setting a series of results [9] which allow to derive from a Hamiltonian a formal (in the sense that solutions do not always exist) system of ordinary differential equations (ODEs) which describes the time evolution of the expected value of graph observables and all higher moments thereof. The key result is jump-closure of observables under the action of a Hamiltonian. In words, this result implies that the time evolution of the expected value of a graph observable \( O \) is a function of the time evolution of the expected value of a finite family of other observables. This induces a coupled system of ODEs which, in good cases, closes on a finite set of variables. Even when that is not the case, this presentation of the dynamics has the quality of being amenable to approximations [9]. Let us prove jump-closure:

**Theorem 36** (Jump-closure for observables). For all Hamiltonians \( H \) as produced in Prop. 24 and all \( g \in G_\mathcal{F} \), there exists a finite family \( \mathcal{F} \subseteq G_\mathcal{F} \) such that

\[
\langle O H = \sum_{h \in \mathcal{F}} a_{g,h} H \delta \langle O h \rangle
\]

for some constants \( \{ a_{g,h} \} \).

**Proof.** By linearity, it is sufficient to consider the case where \( H \) is generated by a single rule \( d = f' \rightarrow f \) with rate \( \kappa \), yielding

\[
H = \kappa (\rho(f' \rightarrow f) - O_f),
\]

where \( O_f = \rho(f \rightarrow f) \). The goal is reduced to exhibiting \( \mathcal{F} \) s.t.

\[
\langle (O_g \rho(f' \rightarrow f) - O_g O_f) \rangle = \sum_{h \in \mathcal{F}} a_{g,h} \langle O h \rangle.
\]

Since observables \( \mathcal{F} \) form a subalgebra of \( \mathcal{F} \) (Prop. 25), \( O_g O_f \) is trivially a finite linear combination of observables. Let us consider the term \( \langle O_g \rho(f' \rightarrow f) \rangle \):

\[
\langle O_g \rho(f' \rightarrow f) \rangle = \langle \rho(g \rightarrow g') \ast \rho(f' \rightarrow f) \rangle = \sum_{n} a_{g,h_n} \delta \langle \rho(h'_n \rightarrow h_n) \rangle,
\]

where \( n \in M(f' \rightarrow f, g \rightarrow g') \). Lemma 35 allows us to write:

\[
\sum_{n} a_{g,h_n} \delta \langle \rho(h'_n \rightarrow h_n) \rangle = \sum_{n} a_{g,h_n,h'} \rho(h'_n \rightarrow h_n) \delta \langle O h_n \rangle = \sum_{n} a_{g,h_n,h} \langle O h_n \rangle,
\]

which concludes the proof.

### 5.5 Jump-closure for products of observables

We will demonstrate, jump-closure for observables corresponds to the data of a system of ODEs describing the time evolution of the expected value of the first “moment” of an observable. The same procedure can be extended to yield ODEs describing the time evolution of higher moments, i.e. expected values of products of observables. The action of a Hamiltonian on a product of observables will be expressed in terms of the commutator of these operators. Let us recall the definition of the commutator.

**Definition 37** (Commutator). The commutator \([A, B]\) of two operators \( A, B \in \text{End}(G) \) is defined by

\[
[A, B] := AB - BA.
\]

It is trivially bilinear.

The commutator of two operators quantifies their lack of commutativity – in this respect, it is a quantitative account of the independence of the processes represented by these operators. In particular, we have:

**Lemma 38.** For all observables \( O_h, O_g \in \mathcal{O} \), \( [O_h, O_g] = 0 \).

**Proof.** Trivial consequence of Prop. 28.
Applying Lemma\textsuperscript{39} one obtains the claim. Let us treat the inductive case \( n = k + 1 \). Given a family \( O_n = \{ O_i \}_{1 \leq i \leq n} \) and writing \( O_i' = O_i \), for all \( i < k \) and \( O_k' = O_k O_n \), we obtain a family of operators \( O_k' \). The result follows easily by applying the induction hypothesis on this family.

Eq. (8) shows that in order to compute higher moments, it is required to compute the full nested commutators \( C(O_n, H) \). This computation can be simplified slightly via the following observation: any element of \( \mathcal{O} \) can be decomposed uniquely as \( v = \hat{v} + \check{v} \) with \( \hat{v} \in \mathcal{O}', \check{v} \in \mathcal{O} \). This decomposition lifts to Hamiltonians by linearity of the representation.

Definition 41 (Non-observable part of a Hamiltonian). Let \( H \) be constructed as in Prop. \textsuperscript{42} \( H \) admits a unique decomposition

\[
H = \hat{H} + \check{H},
\]

where \( \hat{H} = \rho(\hat{v}) \) for \( \hat{v} \in \mathcal{O}' \) and \( \check{H} = \rho(\check{v}) \) for \( \check{v} \in \mathcal{O} \).

The following Lemma takes advantage of this decomposition to simplify commutators:

Lemma 42 (Commutator simplification for Hamiltonians). For all graph observable operators \( O_i \) and all Hamiltonians \( H \) as defined in Prop. \textsuperscript{42} \( [O_i, H] = [O_i, \check{H}] \).

Proof. Trivial, via linearity of the commutator and Lemma\textsuperscript{39}

5.6 Existence of solutions: what we know

We have introduced an algebra of graph-rewriting rules. Rules are seen as normal forms of a combinatorial algebra of diagrams. The diagram algebra is a syntax which we believe has independent interest as it describes what one might call abstract computational traces, or neighbourhoods of such traces. We are particularly interested in investigating prior notions of trace compression \textsuperscript{7} used in causal analysis and diagnosis methods and developed for the case of site-graph rewriting (specifically, the Kappa language) \textsuperscript{6}. Partial evaluation in the diagram algebra (which is permitted by the modularity property of Prop. \textsuperscript{21} should shed light on these notions, especially so in conjunction with the filtered Hopf algebra structure of diagrams (to be explained in Behr et al. \textsuperscript{2}).

With the algebraic part of the paper in place, we turned to actual rewriting which is now seen as a representation of the rule algebra on the vector space spanned by graphs. In the discrete case (no edges), this construction boils down to the Heisenberg-Weyl (HW) algebra and its canonical representation on the Fock space (see references in Ohkubo \textsuperscript{14}), so we are on familiar territory. The fundamental property of this representation is the property of jump-closure, that is to say, we show that observables are closed under (the representation of) rules. This development compares advantageously to Danos et al. \textsuperscript{9} where one obtains jump-closure in a rather ad hoc way and in the simpler setting of Single Pushout (SPO) rewriting (instead of DPO rewriting as in the present article). The actual combinatorial expression of jump-closure reduces in our new framework to a straightforward evaluation in the diagram algebra. Besides the conceptual clarification which the new technique provides, it also marks an improvement as a practical computational tool. It can also be said that the former approach can handle the case of correlators only in an indirect way by using the algebra structure of observables. The direct derivation we propose here is compellingly simple in comparison.

From jump closure, one can immediately derive the so-called rate equations \textsuperscript{10} for graph observables and arbitrary moments thereof. These equations are ubiquitous in the physics and applied mathematical literature. A recent example is Ref. \textsuperscript{11} p21, where

\[
\frac{d}{dt} \langle O_1 \ldots O_n | \psi(t) \rangle = \sum_{\sigma \in S_n} \sum_{m=1}^{n} \frac{C(O_m^\sigma \hat{H}) \prod_{i>m} O_{\sigma(i)} | \psi(t) \rangle}{m!(n-m)!} (10)
\]

Thus, in both cases, we have produced a (potentially infinite) formal system of differential equations – formal in the sense that the following problems might arise:

1. it might not have a unique solution;
2. it might be explosive \textsuperscript{13} \( p \) might not be defined at all times and might range in subprobabilities, in which case the relation of the “solution” with the actual expected value of the observable is subject to caution.

In general, for finite systems (meaning that only finitely many states are accessible from a given reference initial state \( x_0 \) all the above objects make sense, have unique solutions and the meaning of their solution is indeed, as expected, the time-dependent mean value of the associated observable (starting at \( x_0 \). This is also easily seen to be true if the observables are finitely supported. To quote \textsuperscript{15}, “other cases are not quite as clear”.

It remains to be seen whether the few available sufficient conditions on \( H \) and on observables for the derived system of ODEs to have solutions can be exploited to guarantee existence of solutions for a substantial class of dynamics studied in this paper. An adaptation of energy-based graph-rewriting systems (as studied in \textsuperscript{8}) can be a good guess for obtaining such a class. Indeed, in the discrete case (see e.g. \textsuperscript{3}), we know that energy-driven dynamics converges to a multidimensional Poisson distribution and the dynamics is non-explosive, which is a first indication that the equations \textsuperscript{10} should have solutions for a wide class of observables.

6. Outlook
the authors derive the forward equation for a voter model with rewiring (up to order 3). This is still doable by hand, but would become extremely difficult at higher orders or for more complex models. Evidently, it would be interesting to find nontrivial classes of rules and observables for which one can have guarantees on the existence and meaning of solutions to these equations, but, further than the case of finitely supported observables, little seems to be known. In Ref. [15], one finds hard-earned conditions which could allow one some progress, but this remains to be seen. Ergodicity conditions which one can derive from assuming potentials driving the dynamics (perhaps by adapting work done in Danos et al. [8] for site-graphs) offers an interesting and complementary option.

Another interesting avenue is the search for combinatorial applications which parallel those obtained via the HW algebra in the discrete case. Some discrete dynamics, such as multi-type birth-death processes, admit closed forms that one can derive in a systematic way by means of standard analytical combinatorics techniques (umbral calculus [3]). Preliminary results show that we can extend these ideas to graph-based dynamics. Returning to the purely algebraic part of the paper, other types of rewriting follow naturally from the approach. Relaxing condition 5. on diagrams (see Def. [13] in §3) gives rise to a Hopf algebra of diagrams for which one can define four different evaluation morphisms. Each corresponds to a different way to handle worldlines of edges which outlast or predate that of their ends. The simplest evaluation is the only one considered here and corresponds to DPO-rewriting. Other options induce different canonical representations and lead to other types of graph rewriting (among which SPO-rewriting and a hitherto unconsidered dual variant). We will pursue this interesting classification in further work and build the corresponding variants of stochastic mechanical frameworks for each obtained notion of graph-rewriting.

References