Tracelets and Tracelet Analysis
Of Compositional Rewriting Systems

Nicolas Behr

Université de Paris, IRIF, CNRS, F-75013 Paris, France

Taking advantage of a recently discovered associativity property of rule compositions, we extend the classical concurrency theory for rewriting systems over adhesive categories. We introduce the notion of tracelets, which are defined as minimal derivation traces that universally encode sequential compositions of rewriting rules. Tracelets are compositional, capture the causality of equivalence classes of traditional derivation traces, and intrinsically suggest a clean mathematical framework for the definition of various notions of abstractions of traces. We illustrate these features by introducing a first prototype for a framework of tracelet analysis, which as a key application permits to formulate a first-of-its-kind algorithm for the static generation of minimal derivation traces with prescribed terminal events.

1 Motivation and relation to previous works

The analysis of realistic models of complex chemical reaction systems in organic chemistry and in systems biology poses considerable challenges, both in theory and in terms of algorithmic implementations. Two major classes of successful approaches include *chemical graph rewriting* [2, 6, 15, 20], and the *rule-based modeling frameworks* Kappa [27–31] and BioNetGen [21, 45], respectively. These approaches utilize well-established modern variants of *Double-Pushout (DPO)* [37, 39] and *Sesqui-Pushout (SqPO)* [26] rewriting frameworks over suitably chosen adhesive categories [47] (and with additional constraints [39, 44] on objects and transitions for consistency). The sheer complexity of the spaces of distinct classes of objects and of active transitions thereof necessitated the development of specialized and highly optimized variants of static analysis techniques for these types of systems. As we will demonstrate in this paper, a novel class of such techniques is found to arise from a refocusing of the analysis from *derivation traces* to so-called *tracelets*.

To provide some context, we briefly recall some basic notions of rewriting based

---

Nicolas Behr: nicolas.behr@irif.fr, http://nicolasbehr.com, This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 753750.

upon a finitary adhesive category $\mathbf{C}$ [47], such as e.g. the category $\mathbf{FinGraph}$ of finite directed multigraphs. Objects of this category provide the possible configurations or states of the rewriting system (typically considered up to isomorphism), while partial maps between objects (encoded as spans of monomorphisms) will provide the possible transitions, referred to as *(linear) rules*. The application of a rule $O \xleftharpoondown{r} I$ to some object X then requires the choice of an instance of a subobject I within X, established via a monomorphism $m : I \hookrightarrow X$ called a *match*, followed by replacing $m(I) \subset X$ with an instance $m^*(O)$ of O, where the precise details depend on the chosen rewriting semantics (i.e. *Double-Pushout (DPO)* [37, 39] or *Sesqui-Pushout (SqPO)* [26] semantics). This process of rule application is traditionally referred to as a *(direct) derivation*. The central structure studied in the concurrency theory and static analysis of the rewriting system consists in so-called *derivation traces*:

$$\begin{tikzcd}
& O_n \arrow[dl, hook', "m_n^*"] & I_n \arrow[l, harpoon, "r_n"'] \arrow[dr, hook, "m_n"'] & & & & O_2 \arrow[dl, hook', "m_2^*"] & I_2 \arrow[l, harpoon, "r_2"'] \arrow[dr, hook, "m_2"'] & & O_1 \arrow[dl, hook', "m_1^*"] & I_1 \arrow[l, harpoon, "r_1"'] \arrow[dr, hook, "m_1"'] & \\
X_n & & & X_{n-1} \arrow[lll, Rightarrow, "{r_n,m_n}"] & \cdots & X_2 & & & X_1 \arrow[lll, Rightarrow, "{r_2,m_2}"] & & & X_0 \arrow[lll, Rightarrow, "{r_1,m_1}"]
\end{tikzcd} \tag{1}$$

Each transition in such a derivation trace from a state X_i to a state X_{i+1} is thus given by a direct derivation via a linear rule r_i at a match m_i. A typical abstract encoding of rewriting systems is then provided in the form of a *rewriting grammar*, whose data consists of an initial state X_0 and a set of linear rewriting rules, from which all possible derivation traces starting at X_0 are constructed.

Static analysis of rewriting systems is traditionally based upon several notions of *abstractions* of derivation traces. At a fundamental level, the category-theoretical definitions of rewriting are inherently invariant under various types of isomorphisms, which suggests a form of equivalence on derivation traces induced by isomorphisms referred to as *abstraction equivalence* [25]. The second major source of equivalences is based upon so-called *sequential independence* of derivations [14, 26, 37, 39]: again leaving technicalities aside, if two "adjacent" direct derivations $X_{i+1} \xLeftarrow{r_{i+1},\, m_{i+1}} X_i$ and $X_i \xLeftarrow{r_i,\, m_i} X_{i-1}$ in a given derivation trace are sequentially independent, there exist matches m_i' and m_{i+1}' so that $X_{i+1} \xLeftarrow{r_i,\, m_{i'}} X_i$ and $X_i \xLeftarrow{r_{i+1},\, m_{i+1}'} X_{i-1}$ constitute sequential derivations in the opposite order of application. Lifting this notion to sequences of an arbitrary finite number of consecutive derivations yields an abstraction equivalence called *shift equivalence* [36, 37, 46]. Quotienting a given grammar by a combination of abstraction and shift equivalence leads to the sophisticated frameworks of occurrence grammars [7, 9] as well as (equivalently [10, 12, 13]) of processes and unfoldings [8, 11, 14]. Quintessentially, since sequential commutativity induces a preorder on derivations of a grammar, the aforementioned well-established static analysis techniques encode the causal relationships of derivations according to this preorder.

Of particular interest in view of practical applications of such techniques to chemical and biochemical reaction systems (via *chemical graph rewriting* [2, 6, 15, 20], and via the *rule-based modeling* frameworks Kappa [27–31] and BioNetGen [21, 45]) are concepts that permit to extract high-level information on the causal properties of the typically immensely complex transition sets and state spaces encountered in real-life reaction systems in an automated fashion. In the setting of systems chemistry, taking full advantage of the highly constrained type of rewriting relevant to model molecules and possible reactions (i.e. a flavor of DPO rewriting in which vertices modeling atoms are preserved throughout transitions), a highly efficient analysis technique based upon mapping of reaction networks into multi-hypergraphs and modeling pathways as integer hyperflows has been developed in [3, 5, 40]. An essential role in this framework is played by compositions of chemical graph rewriting rules [1, 4, 6], which have been implemented algorithmically in [2]. The tracelets as introduced in this paper may be seen as a formalization of these ideas of understanding pathways as particular rule compositions, which in particular answers an open question on the associativity of compositions of such pathways to the affirmative.

In the biochemistry setting, important developments include sophisticated specializations of the aforementioned static analysis techniques for general rewriting systems to the relevant setting of site-graph rewriting in order to extract information on cellular signaling pathways [30, 31, 33], the notion of refinements [32], techniques of model reduction based on the differential semantics of the stochastic transition systems [34] and notions of trace compression [35]. In particular, so-called *strong compression* as introduced in [35] will play an interesting role also in our tracelet framework. While the theory of static analysis of such complex rewriting systems is thus rather well-developed, several open problems remain. Referring to [23] for a recent review, at present the established approach to the generation of pathways for biochemical reaction systems passes through extensive simulation runs in order to generate large ensembles of derivation traces of the given system, which then have to be curated and suitably compressed in order to extract the static information constituting the pathways of interest. This dependence on a posteriori analyses of derivation traces hinders the efficiency of the algorithms considerably, since typically only a small portion of the information contained in a given trace gives rise to useful information on pathways. We will develop in the following an alternative approach to the static analysis of rewriting systems that aims to avoid precisely this bottleneck in the synthesis of pathways.

The **main contribution** of this paper consists in an alternative paradigm for the static analysis of rewriting systems, which emphasizes the notion of sequential rule compositions over that of derivation traces. Our development hinges on two central theorems of rewriting theory: a theorem describing the relationship between two-step sequences of direct derivations and the underlying rule compositions, the so-called *concurrency theorem* (well-known in the DPO setting [36, 37, 39], only recently established in the SqPO setting [16, 17]), and an equally recently proved [16–19] theorem establishing a form of *associativity* of the operation of rule compositions. The combination of these two results admits to characterize derivation traces *universally* via so-called *tracelets*, in the sense that each trace of length n applied to an initial object X_0 may be obtained as the extension of a *minimal* derivation trace of length n into the context of the object X_0. Referring to Figure 2 for an overview, one may shift focus onto the tracelets themselves as the objects to analyze in a given rewriting system, since they encode all relevant information in terms of the causality of derivation traces. From a technical perspective, since a tracelet is nothing but a special type of derivation trace, all of the traditional analysis techniques on derivation traces remain applicable. At the same time, tracelets may be naturally equipped with a notion of associative composition, which opens novel possibilities in view of static pathway generation in the aforementioned (bio-) chemical rewriting system settings.

Plan of the paper: In Section 2, the core tracelet formalism is established, providing the precise definitions of the concepts summarized in the schematic Figure 2. Section 3 is devoted to developing *tracelet analysis*, based in part upon the aforementioned static analysis techniques for derivation traces. As a first application of our framework, we present a prototypical *Feature-driven Explanatory Tracelet Analysis (FETA)* algorithm in Section 4. Since our framework is heavily based upon our very recent developments in the field of compositional rewriting, we provide a technical appendix containing a collection of illustrative figures and of requisite technical definitions and results.

2 Tracelets for compositional rewriting theories

Assumption 1 *Throughout this paper, we fix*[1] *a category* $\mathbf{C}$ *that satisfies:*

- $\mathbf{C}$ *is* ***adhesive*** *[47]*
- $\mathbf{C}$ *possesses an* ***epi-mono-factorization*** *[44] (i.e. every morphism* $f \in \mathsf{mor}(\mathbf{C})$ *can be factorized into the form* $f = m \circ e$*, with* $m \in \mathsf{mono}(\mathbf{C})$ *and* $e \in \mathsf{epi}(\mathbf{C})$*)*
- $\mathbf{C}$ *possesses a* ***strict initial object*** $\varnothing \in \mathsf{obj}(\mathbf{C})$ *[47] (i.e. an object such that for every* $X \in \mathsf{obj}(\mathbf{C})$*, there exists a unique monomorphism* $\varnothing \to X$*, and for every* $Y \in \mathsf{obj}(\mathbf{C})$*, if there exists a morphism* $Z \to \varnothing$*, then it is an isomorphism).*
- $\mathbf{C}$ *is* ***finitary****, i.e. for every object* $X \in \mathsf{obj}(\mathbf{C})$*, there exist only finitely many monomorphisms* $Z \to X$ *into* X *(and thus only finitely many subobjects of* X*).*

Categories satisfying Assumption 1 have a number of properties that are of particular importance in view of compositionality of rewriting rules (cf. Appendix A.1). A

---

[1] Although especially in the DPO-type rewriting case more general settings would be admissible while retaining compositionality of the rewriting (see [17] for further details), the present choice covers many cases of interest, is a sufficient setting also for compositional Sesqui-Pushout (SqPO) rewriting, and overall strikes a good balance of generality vs. simplicity.

prototypical example of a category satisfying all of the assumptions above is the finitary restriction **FinGraph** of the *category of directed multigraphs* **Graph** [24]. We collect in Appendix A the necessary background material on compositional DPO- and SqPO-type rewriting for rules with conditions [16–19], and will freely employ the standard notations therein.

Definition 1 (Tracelets) *Let $\mathbb{T} \in \{DPO, SqPO\}$ be the type of rewriting, and let $\overline{\mathsf{Lin}}(\mathbf{C})$ denote the set of linear rules with conditions over $\mathbf{C}$ (cf. Definition 14).*

- ***Tracelets of length** 1: the set $\mathcal{T}_1^{\mathbb{T}}$ of type $\mathbb{T}$ tracelets $T(R)$ of length 1 is defined as*

$$\mathcal{T}_1^{\mathbb{T}} := \left\{ T(R) = \begin{tikzcd} O \arrow[d, equal] \arrow[rd, phantom, "\mathbb{T}"] & I \arrow[l, harpoon', "r"'] \arrow[d, equal] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \\ O & I \arrow[l, Rightarrow] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \end{tikzcd} \;\middle|\; R = (r, \mathsf{c}_I) \in \overline{\mathsf{Lin}}(\mathbf{C}) \right\} . \tag{2}$$

- ***Tracelets of length** $n+1$: given tracelets $T_{n+1} \in \mathcal{T}_1^{\mathbb{T}}$ of length 1 and $T_{n\cdots 1} \in \mathcal{T}_n^{\mathbb{T}}$ of length n (for $n \geq 1$), we define a span of monomorphisms $\mu = (I_{n+1} \hookleftarrow M \hookrightarrow O_{n\cdots 1})$ as $\mathbb{T}$-*admissible*, denoted $\mu \in \mathsf{MT}_{T_1}^{\mathbb{T}}(T_{n\cdots 1})$, if the following diagram is constructible:*

$$\begin{tikzcd}[column sep=small]
O_{n+1} \arrow[d, equal] \arrow[rd, phantom, "\mathbb{T}"] & I_{n+1} \arrow[l, harpoon', "r_{n+1}"'] \arrow[d, equal] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{n+1}} & O_n \arrow[d, hook] \arrow[rd, phantom, "\mathbb{T}"] & I_n \arrow[l, harpoon', "r_n"'] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_n} & O_1 \arrow[d, hook] \arrow[rd, phantom, "\mathbb{T}"] & I_1 \arrow[l, harpoon', "r_1"'] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_1} \\
O_{n+1} \arrow[d, hook] \arrow[rd, phantom, "\mathbb{T}"] & I_{n+1} \arrow[l, Rightarrow] \arrow[rd, hook] & M \arrow[l, hook'] \arrow[r, hook] \arrow[d, phantom, "\mathsf{PO}"] & O_{n\cdots 1} \arrow[ld, hook] \arrow[rd, phantom, "DPO^{\dagger}"] & Y_{n,n-1}^{(n)} \arrow[l, Rightarrow] \arrow[d, hook] & \cdots & Y_{2,1}^{(n)} \arrow[d, hook] \arrow[rd, phantom, "DPO^{\dagger}"] & I_{n\cdots 1} \arrow[l, Rightarrow] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{n\cdots 1}} \\
O_{(n+1)\cdots 1} & & Y_{n+1,n}^{(n+1)} \arrow[ll, Rightarrow] & & Y_{n,n-1}^{(n+1)} \arrow[ll, Rightarrow] & \cdots & Y_{2,1}^{(n+1)} & I_{(n+1)\cdots 1} \arrow[l, Rightarrow] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{(n+1)\cdots 1}}
\end{tikzcd} \tag{3}$$

Here, the square marked PO *is constructed as a pushout, followed by performing the $\mathbb{T}$- and $DPO^{\dagger}$-type direct derivations as indicated to form the lower part of the diagram. The latter operation may fail, either by non-existence of the requisite pushout complements (cf. Definition 12), or, if all POCs exist, because the tentative composite condition $\mathsf{c}_{I_{(n+1)\cdots 1}}$ might evaluate to* false, *with computed as $\mathsf{c}_{I_{(n+1)\cdots 1}}$*

$$\begin{aligned} \mathsf{c}_{I_{(n+1)\cdots}} := {} & \mathsf{Shift}(I_{n\cdots 1} \hookrightarrow I_{(n+1)\cdots 1}, \mathsf{c}_{I_{n\cdots 1}}) \\ & \bigwedge \mathsf{Trans}(Y_{n+1,n}^{(n+1)} \Leftarrow I_{(n+1)\cdots 1}, \mathsf{Shift}(I_{n+1} \hookrightarrow Y_{n+1,n}^{(n+1)}, \mathsf{c}_{I_{n+1}})) \, . \end{aligned} \tag{4}$$

If $\mu \in \mathsf{MT}_{T_1}^{\mathbb{T}}(T_{n\cdots 1})$, we define a tracelet $T_{n+1}{}^{\mu}\!\angle_{\mathbb{T}} T_{n\cdots 1}$ of length $n+1$ as

$$T_{n+1}{}^{\mu}\!\angle_{\mathbb{T}} T_{n\cdots 1} := \begin{tikzcd}[column sep=small]
O_{n+1} \arrow[d] \arrow[rd, phantom, "\mathbb{T}"] & I_{n+1} \arrow[l, harpoon', "r_{n+1}"'] \arrow[rd] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{n+1}} & O_n \arrow[ld] \arrow[rd, phantom, "\mathbb{T}"] & I_n \arrow[l, harpoon', "r_n"'] \arrow[d] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_n} & O_1 \arrow[d] \arrow[rd, phantom, "\mathbb{T}"] & I_1 \arrow[l, harpoon', "r_1"'] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_1} \\
O_{(n+1)\cdots 1} & & Y_{n+1,n}^{(n+1)} \arrow[ll, Rightarrow] & & Y_{n,n-1}^{(n+1)} \arrow[ll, Rightarrow] & \cdots & Y_{2,1}^{(n+1)} & I_{(n+1)\cdots 1} \arrow[l, Rightarrow] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{(n+1)\cdots 1}}
\end{tikzcd} \tag{5}$$

We define the set $\mathcal{T}_{n+1}^{\mathbb{T}}$ of type $\mathbb{T}$ tracelets of length $n+1$ *as*

$$\mathcal{T}_{n+1}^{\mathbb{T}} := \{ T_{n+1}{}^{\mu}\!\angle_{\mathbb{T}} T_{n\cdots 1} \,|\, T_{n+1} \in \mathcal{T}_1^{\mathbb{T}} \, , \; T_{n\cdots 1} \in \mathcal{T}_n^{\mathbb{T}} \, , \; \mu \in \mathsf{MT}_{T_1}^{\mathbb{T}}(T_{n\cdots 1}) \Big\} \, . \tag{6}$$

For later convenience, we introduce the tracelet evaluation operation $[[.]]$,

$$[[.]] : \mathcal{T}^{\mathbb{T}} \to \overline{\mathsf{Lin}}(\mathbf{C}) : \mathcal{T}_n^{\mathbb{T}} \ni T \mapsto [[T]] := ((O_{n\cdots 1} \leftharpoonup I_{n\cdots 1}), \mathsf{c}_{I_{n\cdots 1}}) \, , \tag{7}$$

with $\mathcal{T}^{\mathbb{T}} := \bigcup_{n\geq 1}\mathcal{T}_n^{\mathbb{T}}$*, and where* $(O_{n\cdots 1} \leftharpoondown I_{n\cdots 1})$ *denotes the span composition (cf. (3))*

$$(O_{n\cdots 1} \leftharpoondown I_{n\cdots 1}) := (O_{n\cdots 1} \Leftarrow Y^{(n)}_{n,n-1}) \circ \cdots \circ (Y^{(n)}_{2,1} \Leftarrow I_{n\cdots 1})\,. \tag{8}$$

A first example of a tracelet of length 3 generated iteratively from tracelets of length 1 is given in Figure 1c, with the relevant computation presented in (the top half of) Figure 1b. The example illustrates a sequential composition of graph rewriting rules, with vertex symbols and edge colors used purely to encode the structure of the various morphisms and rules, i.e. repeated symbols mark objects identified by the partial morphisms. Note that since in this example no vertices are deleted without explicitly deleting the incident edges, too, this example constitutes a valid composition in both the DPO- and the SqPO-type frameworks.

Another very important aspect visualized in Figure 1b is the *associativity* property of the underlying rule compositions: the top half of the figure represents a composition of r_2 with r_1 (yielding the tracelet of length 2 highlighted in blue), followed by a further composition of r_3 with the composite of r_2 and r_1. By the associativity theorem for compositional rewriting theories (Theorem 6), there exist suitable overlaps such that the outcome of the aforementioned operation may be equivalently obtained by composing r_3 with r_2 (yielding the tracelet of length 2 highlighted in yellow), and by pre-composing the composite with r_1. Vertically composing squares in each half of Figure 1b, one may verify that this associativity property on rule compositions extends to an associativity property on tracelet compositions, as both halves of the figure yield the same tracelet of length 3. These observations motivate the following extension of the definition of $._{\mathbb{T}}\!\angle.$:

Definition 2 (Tracelet composition) *For tracelets* $T', T \in \mathcal{T}^{\mathbb{T}}$ *of lengths* m *and* n*, respectively, a span of monomorphisms* $\mu = (I'_{m\cdots 1} \hookleftarrow M \hookrightarrow O_{n\cdots 1})$ *is defined to be an* admissible match of T into T'*, denoted* $\mu \in \mathsf{MT}^{\mathbb{T}}_{T'}(T)$*, if (i) all requisite pushout complements exist to form the type* $DPO^\dagger$ *derivations (in the sense of rules without conditions) to construct the diagram in (9a) below, where* $p := m + n + 1$*,*

$$\begin{tikzcd}
O'_m \arrow[dd, hook] & & I'_m \arrow[ll, harpoon, "r'_m"'] \arrow[dd, hook] & \mathsf{c}_{I'_m} \arrow[l, phantom, "\blacktriangleleft"] & & O'_1 \arrow[dd, hook] & & I'_1 \arrow[ll, harpoon, "r'_1"'] \arrow[dd, hook] & \mathsf{c}_{I'_1} \arrow[l, phantom, "\blacktriangleleft"] & & O_n \arrow[dd, hook] & & I_n \arrow[ll, harpoon, "r_n"'] \arrow[dd, hook] & \mathsf{c}_{I_n} \arrow[l, phantom, "\blacktriangleleft"] & & O_1 \arrow[dd, hook] & & I_1 \arrow[ll, harpoon, "r_1"'] \arrow[dd, hook] & \mathsf{c}_{I_1} \arrow[l, phantom, "\blacktriangleleft"] \\
 & \mathbb{T} & & & & & \mathbb{T} & & & M \arrow[dll, hook'] \arrow[dr, hook] & & \mathbb{T} & & & & & \mathbb{T} & & \\
O'_{m\cdots 1} \arrow[dd, hook] & & Y^{(m)}_{m,m-1} \arrow[ll, Rightarrow] \arrow[dd, hook] & & \cdots & Y^{(m)}_{2,1} \arrow[dd, hook] & & I'_{m\cdots 1} \arrow[ll, Rightarrow] \arrow[ddrr, hook] & & & O_{n\cdots 1} \arrow[ddl, hook'] & & Y^{(n)}_{n,n-1} \arrow[ll, Rightarrow] \arrow[dd, hook] & & \cdots & Y^{(n)}_{2,1} \arrow[dd, hook] & & I_{n\cdots 1} \arrow[ll, Rightarrow] \arrow[dd, hook] & \mathsf{c}_{I_{n\cdots 1}} \arrow[l, phantom, "\blacktriangleleft"] \\
 & \mathbb{T} & & & & & \mathbb{T} & \mathsf{c}_{I'_{m\cdots 1}} \arrow[u, phantom, "\blacktriangle"] & & \mathsf{PO} & & DPO^\dagger & & & & & DPO^\dagger & & \\
O_{p\cdots 1} & & Y^{(p)}_{p,p-1} \arrow[ll, Rightarrow] & & \cdots & Y^{(p)}_{n+2,n+1} & & & & Y^{(p)}_{n+1,n} \arrow[llll, Rightarrow] & & & Y^{(p)}_{n,n-1} \arrow[lll, Rightarrow] & & \cdots & Y^{(p)}_{2,1} & & I_{p\cdots 1} \arrow[ll, Rightarrow] & \mathsf{c}_{I_{p\cdots 1}} \arrow[l, phantom, "\blacktriangleleft"]
\end{tikzcd} \tag{9a}$$

and if (ii) the condition $\mathsf{c}_{I_{(m+n+1)\cdots 1}}$ *as in (9b) below does not evaluate to* false*:*

$$\begin{aligned}
\mathsf{c}_{I_{(m+n+1)\cdots}} &:= \mathsf{Shift}(I_{n\cdots 1} \hookrightarrow I_{(m+n+1)\cdots 1}, \mathsf{c}_{I_{n\cdots 1}}) \\
&\qquad \bigwedge \mathsf{Trans}(Y^{(m+n+1)}_{n+1,n} \Leftarrow I_{(m+n+1)\cdots 1}, \mathsf{Shift}(I_{m\cdots 1} \hookrightarrow Y^{(n+1)}_{n+1,n}, \mathsf{c}_{I_{m\cdots 1}}))\,.
\end{aligned} \tag{9b}$$

Then for $\mu \in \mathsf{MT}^{\mathbb{T}}_{T'}(T)$, *we define the* type $\mathbb{T}$ tracelet composition of T' with T along μ *as*

$$T'{}^{\mu}\angle_{\mathbb{T}}T := \begin{tikzcd}
O'_m \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & I'_m \arrow[l, hook', "r'_m"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[dr, hook] & \mathsf{c}_{I'_m} & & & O_1 \arrow[dl, hook] \arrow[d, phantom, "\mathbb{T}"] & I_1 \arrow[l, hook', "r_1"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[d, hook] & \mathsf{c}_{I_1} \\
O_{p\cdots 1} & & Y^{(p)}_{p,p-1} \arrow[ll, Rightarrow] & \cdots & Y^{(p)}_{2,1} & & I_{p\cdots 1} \arrow[ll, Rightarrow] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_{I_{p\cdots 1}}
\end{tikzcd}\,. \tag{10}$$

Next, the precise relationship between $\mathbb{T}$-type rule and tracelet compositions is clarified.

Theorem 1 *Let* $.{\lhd_{\mathbb{T}}}.$ *denote the* $\mathbb{T}$*-type rule composition (Definition 14), and let the set of* $\mathbb{T}$*-admissible matches be denoted by* $\mathsf{M}^{\mathbb{T}}_{r_2}(r_1)$ *(for* $r_2, r_1 \in \overline{\mathsf{Lin}}(\mathbf{C})$*).*

(i) For all $T', T \in \mathcal{T}^{\mathbb{T}}$, $\mathsf{MT}^{\mathbb{T}}_{T'}(T) = \mathsf{M}^{\mathbb{T}}_{[[T']]}([[T]])$.

(ii) For all $T', T \in \mathcal{T}^{\mathbb{T}}$ *and* $\mu \in \mathsf{MT}^{\mathbb{T}}_{T'}(T)$, $[[T'{}^{\mu}\angle_{\mathbb{T}}T]] = [[T']]^{\mu}\lhd_{\mathbb{T}}[[T]]$.

(iii) The $\mathbb{T}$*-type tracelet composition is* **associative**, *i.e. for any three tracelets* $T_1, T_2, T_3 \in \mathcal{T}^{\mathbb{T}}$, *there exists a bijection* $\varphi : S_{3(21)} \xrightarrow{\cong} S_{(32)1}$ *between the sets pairs of* $\mathbb{T}$*-admissible matches of tracelets (with* $T_{ji} := T_j{}^{\mu_{ji}}\angle_{\mathbb{T}}T_i$ *and using property (i))*

$$\begin{aligned}
S_{3(21)} &:= \{(\mu_{21}, \mu_{3(21)}) | \mu_{21} \in \mathsf{M}^{\mathbb{T}}_{[[T_2]]}([[T_1]])\,,\ \mu_{3(21)} \in \mathsf{M}^{\mathbb{T}}_{[[T_3]]}([[T_{21}]])\\
S_{(32)1} &:= \{(\mu_{32}, \mu_{(32)1}) | \mu_{32} \in \mathsf{M}^{\mathbb{T}}_{[[T_3]]}([[T_2]])\,,\ \mu_{(32)1} \in \mathsf{M}^{\mathbb{T}}_{[[T_{32}]]}([[T_1]])\}
\end{aligned} \tag{11}$$

such that for all $(\mu'_{32}, \mu'_{(32)1}) = \varphi((\mu_{21}, \mu_{3(21)}))$

$$T_3{}^{\mu_{3(21)}}\angle_{\mathbb{T}}\left(T_2{}^{\mu_{21}}\angle_{\mathbb{T}}T_1\right) \cong \left(T_3{}^{\mu'_{32}}\angle_{\mathbb{T}}T_2\right){}^{\mu'_{(32)1}}\angle_{\mathbb{T}}T_3\,. \tag{12}$$

Moreover, the bijection φ *coincides with the corresponding bijection provided in the associativity theorem for* $\mathbb{T}$*-type rule compositions (Theorem 6). (**Proof:** Appendix A.7)*

Finally, combining the associativity results for rule and tracelet compositions with the so-called concurrency theorems for compositional rewriting theories, we find a characterization of derivation traces via tracelets and vice versa:

Theorem 2 (Tracelet characterization) *For all type-*$\mathbb{T}$ *tracelets* $T \in \mathcal{T}^{\mathbb{T}}_n$ *of length* n*, for all objects* X_0 *of* $\mathbf{C}$*, and for all monomorphisms* $(m : I_{n\cdots 1} \hookrightarrow X_0)$ *such that* $m \in \mathsf{M}^{\mathbb{T}}_{[[T]]}(X_0)$*, there exists a type-*$\mathbb{T}$ *direct derivation* $D = T_m(X_0)$ *obtained via vertically composing the squares in each column of the diagram below:*

$$\begin{tikzcd}
O_n \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & I_n \arrow[l, hook', "r_n"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[dr, hook] & \mathsf{c}_{I_n} & & & O_1 \arrow[dl, hook] \arrow[d, phantom, "\mathbb{T}"] & I_1 \arrow[l, hook', "r_1"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[d, hook] & \mathsf{c}_{I_1} \\
O_{n\cdots 1} \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & & Y^{(n)}_{n,n-1} \arrow[ll, Rightarrow] \arrow[d, hook] & \cdots & Y^{(n)}_{2,1} \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & & I_{n\cdots 1} \arrow[ll, Rightarrow] \arrow[r, phantom, "\blacktriangleleft"] \arrow[d, hook] & \mathsf{c}_{I_{n\cdots 1}} \\
X_n & & X_{n-1} \arrow[ll, Rightarrow] & \cdots & X_1 & & X_0 \arrow[ll, Rightarrow] &
\end{tikzcd}
\leftrightsquigarrow
\begin{tikzcd}
O_n \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & I_n \arrow[l, hook', "r_n"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[dr, hook] & \mathsf{c}_{I_n} & & & O_1 \arrow[dl, hook] \arrow[d, phantom, "\mathbb{T}"] & I_1 \arrow[l, hook', "r_1"'] \arrow[r, phantom, "\blacktriangleleft"] \arrow[d, hook] & \mathsf{c}_{I_1} \\
X_n & & X_{n-1} \arrow[ll, Rightarrow] & \cdots & X_1 & & X_0 \arrow[ll, Rightarrow] &
\end{tikzcd} \tag{13}$$

Conversely, every $\mathbb{T}$*-direct derivation* D *of length* n *along rules* $R_j = (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *starting at an object* X_0 *of* $\mathbf{C}$ *may be cast into the form* $D = T_m(X_0)$ *for some tracelet* T *of length* n *and a* $\mathbb{T}$*-admissible match* $m \in \mathsf{M}^{\mathbb{T}}_{[[T]]}(X_0)$ *that are uniquely determined from* D *(up to isomorphisms). (**Proof:** Appendix A.8)*

3 Tracelet Analysis

Let us first introduce a convenient shorthand notation for tracelets, which emphasizes that by definition, every tracelet is a type of commutative diagram of consecutive direct derivations "glued" at common interface objects (compare Figure 2a):

Definition 3 *For a tracelet $T \in \mathcal{T}_n^{\mathbb{T}}$ of length $n \geq 1$, let symbols t_j for $1 \leq j \leq n$ denote j-th* subtracelets *of T, so that $T \equiv t_n|t_{n-1}|\dots|t_1$ is a concatenation of its subtracelets, with*

$$
t_j := \begin{tikzcd} O_j \arrow[d, hook] & I_j \arrow[l, harpoon', "r_j"'] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] \arrow[dl, phantom, "\mathbb{T}"] & \mathsf{c}_{I_j} \\ Y_{j+1,j}^{(n)} & Y_{j,j-1}^{(n)} \arrow[l, Leftarrow] \end{tikzcd} \,, \qquad Y_{n+1,n}^{(n)} := O_{n\cdots 1}\,,\; Y_{1,0}^{(n)} := I_{n\cdots 1}\,. \tag{14}
$$

Based upon the concurrency theorem for compositional rewriting theories (Theorem 5), one may define an operation that provides the starting point of our tracelet analysis framework:

Corollary 1 (Tracelet surgery) *Let $T \in \mathcal{T}_n^{\mathbb{T}}$ a $\mathbb{T}$-type tracelet of length n, so that $T \equiv t_n|\dots|t_1$. Then for any consecutive subtracelets $t_j|t_{j-1}$ in T, one may uniquely (up to isomorphisms) construct a diagram $t_{(j|j-1)}$ and a tracelet $T_{(j|j-1)}$ of length 2 as follows:*

$$
\begin{tikzcd}[column sep=small] & \mathsf{c}_{I_j} \arrow[d, phantom, "\blacktriangledown"] & & & \mathsf{c}_{I_{j-1}} \arrow[d, phantom, "\blacktriangledown"] & \\ O_j \arrow[d] & I_j \arrow[l, harpoon', "r_j"'] \arrow[dr] & & O_{j-1} \arrow[dl] & I_{j-1} \arrow[l, harpoon', "r_{j-1}"'] \arrow[dr] & \\ Y_{j+1,j}^{(n)} & \mathbb{T} & Y_{j,j-1}^{(n)} \arrow[ll, Leftarrow] & \mathbb{T} & & Y_{j-1,j-2}^{(n)} \arrow[lll, Leftarrow] \end{tikzcd}
\rightsquigarrow
\begin{tikzcd}[column sep=small] & & \mathsf{c}_{I_j} \arrow[d, phantom, "\blacktriangledown"] & M \arrow[dl, hook] \arrow[dr, hook] & & \mathsf{c}_{I_{j-1}} \arrow[d, phantom, "\blacktriangledown"] & & \\ & O_j \arrow[dl] \arrow[ddl, bend left] & I_j \arrow[l, harpoon', "r_j"'] \arrow[dr, hook] \arrow[ddr, bend right] \arrow[dl, phantom, "\mathbb{T}"] & \mathsf{PO} & O_{j-1} \arrow[dl, hook] \arrow[ddl, bend left] & I_{j-1} \arrow[l, harpoon', "r_{j-1}"'] \arrow[dr] \arrow[ddr, bend right] \arrow[dl, phantom, "DPO^{\dagger}"] & & \\ O_{j|j-1} \arrow[d, hook] & & \mathbb{T} & Y_{j,j-1}^{(2)} \arrow[lll, Leftarrow] \arrow[d, hook] & & \mathbb{T} & I_{j|j-1} \arrow[lll, Leftarrow] \arrow[d, hook] & \mathsf{c}_{I_{j|j-1}} \arrow[l, phantom, "\blacktriangleleft"] \\ Y_{j+1,j}^{(n)} & & & Y_{j,j-1}^{(n)} \arrow[lll, Leftarrow] & & & Y_{j-1,j-2}^{(n)} \arrow[lll, Leftarrow] & \end{tikzcd}
\tag{15a}
$$

$$
t_{(j|j-1)} := \begin{tikzcd} O_{j|j-1} \arrow[d, hook] & I_{j|j-1} \arrow[l] \arrow[d, hook] \arrow[r, phantom, "\blacktriangleleft"] \arrow[dl, phantom, "\mathbb{T}"] & \mathsf{c}_{I_{j|j-1}} \\ Y_{j+1,j}^{(n)} & Y_{j-1,j-2}^{(n)} \arrow[l, Leftarrow] \end{tikzcd} \,, \qquad T_{(j|j-1)} := T(r_j, \mathsf{c}_{I_j})^{\mu}\!\angle_{\mathbb{T}} T(r_{j-1}, \mathsf{c}_{I_{j-1}}) \tag{15b}
$$

Here, $\mu = (I_j \hookleftarrow M \hookrightarrow O_{j-1})$ is the span of monomorphisms obtained by taking the pullback of the cospan $(I_j \hookrightarrow Y_{j,j-1}^{(n)} \hookleftarrow O_{j-1})$, and this μ is always a $\mathbb{T}$-admissible match. By associativity of the tracelet composition, this extends to consecutive sequences $t_j|\dots|t_{j-k}$ of subtracelets in T inducing diagrams $t_{(j|\dots|j-k)}$ and tracelets of length 1 $T_{(j|\dots|j-k)}$, where for $k = 0$, $t_{(j)} = t_j$ and $T_{(j)} = T(r_j, \mathsf{c}_{I_j})$.

Proof: The proof follows by invoking Theorem 2 in order to convert the derivation trace encoded in a given the subdiagram $t_j|\dots|t_{j-k}$ into the application of a tracelet $T_{(j|\dots|j-k)}$ of length k onto the initial object $Y_{j-k,j-k-1}^{(n)}$. □

It is via these tracelet surgery operations that we may lift the theory of static analysis of derivation traces to our alternative setting of tracelets. We define in the following two notions of equivalence that have analogues also in the traditional theory of rewriting systems.

Definition 4 (Tracelet abstraction equivalence) *Two tracelets $T, T' \in \mathcal{T}_n^{\mathbb{T}}$ of the same length $n \geq 1$ are defined to be* abstraction equivalent, *denoted $T \equiv_A T'$, if there exist suitable isomorphisms on the objects in T in order to transform T into T' (with transformations on morphisms induced by object isomorphisms).*

Due to the intrinsic invariance of all category-theoretical constructions pertaining to rewriting rules as well as tracelets up to universal isomorphisms, it is clear that abstraction equivalence is a very natural[2], or even essential type of equivalence.

Definition 5 (Tracelet shift equivalence) *Let $T, T' \in \mathcal{T}_n^{\mathbb{T}}$ be two tracelets of the same length $n \geq 1$. If there exist subtracelets $t_j|\ldots|t_{j-k}$ and $t'_j|\ldots|t'_{j-k}$ such that*

(i) the subtracelets have the same rule content (up to isomorphisms), i.e. there exists a permutation $\sigma \in S_k$ such that $[[T_{(p)}]] \cong [[T'_{(\sigma(p))}]]$ for all $j-k \leq p \leq j$, and

(ii) the diagrams $t_1|\ldots|t_{(j|\ldots|j-k)}|\ldots|t_n$ and $t'_1|\ldots|t'_{(j|\ldots|j-k)}|\ldots|t'_n$ are isomorphic,

then T and T' are defined to be shift equivalent, *denoted $T \equiv_S T'$. Extending $\equiv_S$ by transitivity then yields an equivalence relation on $\mathcal{T}_n^{\mathbb{T}}$ for every $n \geq 1$.*

Referring to Appendix B.1 for the precise details, one may for example verify that the tracelet $t_3|t_2|t_1$ of length 3 depicted in Fig. 1b is shift equivalent to a tracelet $t'_2|t'_3|t_1$, with the order of the applications of the rules r_3 and r_2 (contained in the yellow box in Fig. 1b) reversed. Notably, while our definition of tracelet abstraction equivalence follows precisely the same methodology as its analogous notion in rewriting theory, our definition of tracelet shift equivalence is strictly more general than the notion of shift equivalence in rewriting theories according to the standard literature [36, 37, 46]. More precisely, the latter concept is based upon so-called *sequential independence* for derivation sequences [14, 26, 37, 39], which would induce a notion of tracelet shift equivalence strictly less permissive than our requirements described in Definition 5. As this difference is of crucial importance to the design of static analysis algorithms, we provide the precise technical relationship in Theorem 7 of Appendix B for clarification.

4 Application: a prototype for a Feature-driven Explanatory Tracelet Analysis (FETA) algorithm

A major motivation behind the development of the tracelet analysis framework has been the desire to improve upon (and, to an extent, also formalize) existing static analysis techniques for rewriting systems in the application areas of bio- and organic

---

[2] While we will typically consider tracelets by default only up to abstraction equivalence, the definitions provided thus far may nevertheless still be interpreted as concrete operations if suitable "standard representatives" are chosen for pushouts, pullbacks etc. — for instance, an extensive discussion of such an interplay of concrete representatives vs. universal structures for the special case of graph rewriting systems may be found in [7].

chemical reaction systems (see also Section 1). An application of our framework to the static generation of so-called *pathways* appears to be particularly promising:

Definition 6 (Pathways (sketch)) *Let $\mathcal{R} = \{R_j \in \overline{\mathsf{Lin}}(\mathbf{C})\}_{j\in J}$ a (finite) set of rules with conditions over* $\mathbf{C}$*, which model the* transitions *of a rewriting system. We designate a rule $E \in \overline{\mathsf{Lin}}(\mathbf{C})$ as modeling a* "target event"*, i.e. E must be the last rule applied in the derivation traces we will study. Let moreover $\equiv_C$ be an equivalence relation on derivation traces such as abstraction or shift equivalences, or combinations thereof. Then the task of* pathway generation *or* explanatory synthesis *for the type-$\mathbb{T}$ rewriting system based upon the set of rules $\mathcal{R}$ is defined as follows: synthesize the* maximally compressed *derivation traces ending in an application of E* such that *"E cannot occur at an earlier position in a given trace". Here, compression refers to retaining only the smallest traces in a given $\equiv_C$ equivalence class, while the last part of the statement needs to be made precise in a specific application (as it depends on the chosen framework).*

A standard approach to this type of task consists in generating a large number of random generic derivation traces first, followed by static analysis type operations performed on these traces in order to extract pathways (see e.g. the recent review [23]). This type of approach typically suffers from two disadvantages: (i) depending on the complexity of the rule set $\mathcal{R}$ and of the target event E, it may be difficult to find suitable choices of initial objects X_0 as an input to the simulation algorithms, and (ii) the extraction of compressed pathways from typically quite extensive datasets of simulator outputs may be computationally rather intense. We thus propose an alternative pathway generation approach based upon tracelets, which avoids the first problem by design (since tracelets are composable with themselves and yield the minimal derivation traces for entire classes of derivations according to Theorem 2).

Definition 7 (Algorithm 1: FETA) *With input data as described in Algorithm 1, let $\equiv_C$ be the equivalence relation obtained by conjunction of the tracelet abstraction and tracelet shift equivalences $\equiv_A$ and $\equiv_S$, respectively. Then for a tracelet $T \in \mathcal{T}^{\mathbb{T}}_{n+1}$ of the structure $T = t_E|t_n|\dots|t_1$ (for some finite value $n \geq 0$, and with t_E containing the rule E, $[[T_{(E)}]] \cong E$), we let $E \prec_C T$ denote the following property: there exist no tracelets $T' \in \mathcal{T}^{\mathbb{T}}_{n+1}$*

$$t_E|t_n|\dots|t_1 \equiv_C t'_{n+1}|t'_n|\dots|t'_1 \quad \text{with} \;\; [[T'_{(k)}]] \cong E \text{ for an index } k < n+1\,. \tag{16}$$

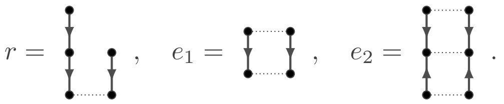
We refer to the set of such tracelets modulo $\equiv_C$ as the set of strongly compressed pathways.

Since length limitations preclude presenting an application example of realistic complexity in one of the chemical reactions system frameworks, we will present here only a first proof of concept for an application of the FETA algorithm, which nevertheless illustrates in which sense the above algorithm synthesizes "explanations".

Example 1 *Let $\mathbf{C} = \mathbf{FinGraph}$ be the category of finite directed multigraphs. For compactness of graphical illustrations and to enhance intuitions, we will present linear rules $r = (O \hookleftarrow$*

```
Algorithm 1: Feature-driven Explanatory Tracelet Analysis (FETA)
  Data: $N_{max} \geq 2 \leftarrow$ maximal length of tracelets to be generated
  $T_E := T(E) \leftarrow$ tracelet of length 1 associated to the rule $E$
  $\mathsf{T}_1 := \{T(R_j) \mid j \in J\} \leftarrow$ set of tracelets of length 1 associated to the transitions
  Result: sets $\mathsf{P}_i$ $(i = 2, \dots, N_{max})$ of strongly compressed pathways
  begin
      $P_1 := \{T_E\} \leftarrow$ the only pathway of length 1;
      for $2 < n \leq N_{max}$ do
          $\mathsf{pre}_n := \left\{P^{\mu}\angle_{\mathbb{T}}T \mid P \in \mathsf{P}_{n-1}, T \in \mathsf{T}_1\,,\ \mu \in \mathsf{MT}^{\mathbb{T}}_{P}(T)\right\}$;
          $\mathsf{P}_n := \{T' \in \mathsf{pre}_n \mid E \prec_C T'\}\diagup_{\equiv_C}$;
      end
  end
```

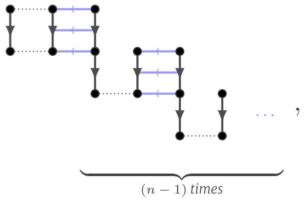
$K \hookrightarrow I) \in \mathsf{Lin}(\mathbf{FinGraph})$ in a diagrammatic form, where graphs O and I are depicted to the left and to the right, respectively, and where dotted lines connecting elements of I with elements of O indicate the structure of the partial map encoded in the span r. Let thus $\mathcal{R} = \{r\}$ be a one-element transition set (for a rule $r \in \mathsf{Lin}(\mathbf{FinGraph})$ without conditions), and let $e_1, e_2 \in \mathsf{Lin}(\mathbf{FinGraph})$ be two rules modeling alternative target events:



(17)

If we consider DPO-type rewriting, the FETA algorithm produces the following strongly compressed pathways for $n \geq 2$ (with light blue arrows indicating the relative overlap structure within the tracelets):

$$\mathsf{P}_n = \{S_n\}\,, \qquad S_n = t_E|\underbrace{t_r|\dots|t_r}_{(n-1)\ \textit{times}} \ =$$



(18)

while for the target event e_2 the algorithm detects no pathways P'_n for $n \geq 2$. This result may indeed be interpreted as expressing a high-level causal structure or explanation about this simple rewriting system. As for e_1, the pathways P_n are seen to effectively encode those possibilities of sequential rule compositions that ensure that the edge eventually matched by e_1 had not already been present in any of the first $n-2$ steps of rule applications. This leaves only the pathways of type S_n as options, since for any other match of the tracelet T_E within a candidate tracelet T of length n, one finds a violation of the condition $E \prec_C T$. On the other hand, the fact that there are no pathways of length $n \geq 2$ for the target event encoded by e_2 signifies that the rule r acting on some initial graph X_0 can in fact not generate any occurrences of the shape of the input of e_2 (two edges with a shared vertex pointing towards each other) that had not *already been present in X_0. Note that we have obtained this result* statically, *and without ever evaluating any concrete direct derivation on initial graphs X_0.*

5 Conclusion and Outlook

Many of the standard constructions in the concurrency theory and the theory of static analysis of rewriting systems over adhesive categories are ultimately based upon one of the central theorems of rewriting theory, which is known fittingly as the *concurrency theorem* [36, 38, 49]. The essential property provided by this theorem is a form of compatibility between (i) sequential applications of rewriting rules starting at some initial object X_0, and (ii) a one-step application of a *composition* of the rewriting rules involved, and with both descriptions in a (constructive) bijective correspondence. As outlined in Section 1, it is then precisely this correspondence which allows to develop various abstractions and analysis techniques for derivation traces of a given rewriting system [7, 8, 11, 12, 14, 35]. However, as has been only very recently discovered [16–19], both Double-Pushout (DPO) and Sesqui-Pushout (SqPO) rewriting theories over suitable adhesive categories carry an additional important structure, namely on the operation of *composing rules* itself: in a certain sense, rule compositions are *associative* (with a concrete example provided in Figure 1).

In this paper, we demonstrate that combining the concurrency with the associativity theorems, one is naturally led to the concept of *tracelets* (Section 2), which may be intuitively understood as a form of *minimal derivation traces* that generate all derivations that are based upon the same sequential rule compositions (Theorem 2). Owing to the associativity theorem for rule compositions, tracelets are on the one hand by definition instances of derivation traces themselves and thus admit all aforementioned standard static analysis techniques, but importantly in addition afford certain universal properties: an associative notion of composition directly on tracelets, certain types of "surgery" operations, and finally various forms of equivalence relations that may be employed to develop compressions and other abstractions of tracelets (Section 3).

In view of practical applications, we have proposed a first prototypical tracelet-based static analysis algorithm, the so-called *Feature-driven Explanatory Tracelet Analysis (FETA)* algorithm (Section 4). As illustrated in Example 1, this algorithm permits to extract high-level causal information on the "pathways" or minimal derivation traces that can lead to the ultimate application of the rule that models a target event. We believe that our methodology may provide a significant contribution to the static analysis toolset in the fields of chemical graph transformation systems [2, 6, 15, 20] and of rule-based modeling approaches to biochemical reaction systems such as the Kappa [23, 28, 30, 31] and the BioNetGen [21, 45] frameworks.

References

[1] Jakob L Andersen, Christoph Flamm, Daniel Merkle, and Peter F Stadler. Inferring chemical reaction patterns using rule composition in graph grammars. *Journal of Systems Chemistry*, 4(1):4, 2013. DOI: 10.1186/1759-2208-4-4.

[2] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. A Software Package for Chemically Inspired Graph Transformation. In *Graph Transformation*, pages 73–88. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-40530-8_5.

[3] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Chemical Transformation Motifs — Modelling Pathways as Integer Hyperflows. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(2):510–523, 2019. DOI: 10.1109/tcbb.2017.2781724.

[4] Jakob Lykke Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. 50 Shades of Rule Composition. In *Formal Methods in Macro-Biology*, pages 117–135. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-10398-3_9.

[5] Jakob Lykke Andersen, Rolf Fagerberg, Christoph Flamm, Rojin Kianian, Daniel Merkle, and Peter F Stadler. Towards mechanistic prediction of mass spectra using graph transformation. *MATCH Commun. Math. Comput. Chem*, 80:705–731, 2018.

[6] Jakob Lykke Andersen, Christoph Flamm, Daniel Merkle, and Peter F Stadler. Rule composition in graph transformation models of chemical reactions. *MATCH Commun. Math. Comput. Chem*, 80(661-704):45, 2018.

[7] Paolo Baldan. Modelling Concurrent Computations: from Contextual Petri Nets to Graph Grammars. *PhD thesis, University of Pisa*, 2000.

[8] Paolo Baldan, Andrea Corradini, and Ugo Montanari. Concatenable graph processes: Relating processes and derivation traces. *Lecture Notes in Computer Science*, pages 283–295, 1998. DOI: 10.1007/bfb0055061.

[9] Paolo Baldan, Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, and Michael Löwe. Concurrent Semantics of Algebraic Graph Transformations. *Handbook of Graph Grammars and Computing by Graph Transformation*, pages 107–187, 1999. DOI: 10.1142/9789812814951_0003.

[10] Paolo Baldan, Andrea Corradini, and Ugo Montanari. Unfolding of Double-Pushout Graph Grammars is a Coreflection. In *Theory and Application of Graph Transformations*, pages 145–163. Springer Berlin Heidelberg, 2000. DOI: 10.1007/978-3-540-46464-8_11.

[11] Paolo Baldan, Andrea Corradini, Tobias Heindel, Barbara König, and Paweł Sobociński. Processes for Adhesive Rewriting Systems. *Lecture Notes in Computer Science*, pages 202–216, 2006. DOI: 10.1007/11690634_14.

[12] Paolo Baldan, Andrea Corradini, Ugo Montanari, and Leila Ribeiro. Unfolding

semantics of graph transformation. *Information and Computation*, 205(5):733–782, 2007. DOI: 10.1016/j.ic.2006.11.004.

[13] Paolo Baldan, Andrea Corradini, Tobias Heindel, Barbara König, and Paweł Sobociński. Unfolding Grammars in Adhesive Categories. *Lecture Notes in Computer Science*, pages 350–366, 2009. DOI: 10.1007/978-3-642-03741-2_24.

[14] Paolo Baldan, Andrea Corradini, Tobias Heindel, Barbara König, and Paweł Sobociński. Processes and unfoldings: concurrent computations in adhesive categories. *Mathematical Structures in Computer Science*, 24(04), jun 2014. DOI: 10.1017/s096012951200031x.

[15] Wolfgang Banzhaf, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Algorithmic Cheminformatics (Dagstuhl Seminar 14452). *Dagstuhl Reports*, 4(11): 22–39, 2015. DOI: 10.4230/DagRep.4.11.22.

[16] Nicolas Behr. Sesqui-Pushout Rewriting: Concurrency, Associativity and Rule Algebra Framework. *arXiv preprint 1904.08357*, 2019.

[17] Nicolas Behr and Jean Krivine. Compositionality of Rewriting Rules with Conditions. *arXiv preprint 1904.09322*, 2019.

[18] Nicolas Behr and Pawel Sobocinski. Rule Algebras for Adhesive Categories. In Dan Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-088-0. DOI: 10.4230/LIPIcs.CSL.2018.11.

[19] Nicolas Behr, Vincent Danos, and Ilias Garnier. Stochastic mechanics of graph rewriting. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science - LICS '16*. ACM Press, 2016. DOI: 10.1145/2933575.2934537.

[20] Gil Benkö, Christoph Flamm, and Peter F. Stadler. A Graph-Based Toy Model of Chemistry. *Journal of Chemical Information and Computer Sciences*, 43(4):1085–1093, 2003. DOI: 10.1021/ci0200570.

[21] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004. DOI: 10.1093/bioinformatics/bth378.

[22] Paul Boehm, Harald-Reto Fonio, and Annegret Habel. Amalgamation of graph transformations: A synchronization mechanism. *Journal of Computer and System Sciences*, 34(2-3):377–408, 1987. DOI: 10.1016/0022-0000(87)90030-4.

[23] Pierre Boutillier, Mutaamba Maasha, Xing Li, Héctor F Medina-Abarca, Jean Krivine, Jérôme Feret, Ioana Cristescu, Angus G Forbes, and Walter Fontana. The kappa platform for rule-based modeling. *Bioinformatics*, 34(13):i583–i592, 2018. DOI: 10.1093/bioinformatics/bty272.

[24] Benjamin Braatz, Hartmut Ehrig, Karsten Gabriel, and Ulrike Golas. Finitary $\mathcal{M}$-Adhesive Categories. *Lecture Notes in Computer Science*, pages 234–249, 2010. DOI: 10.1007/978-3-642-15928-2_16.

[25] A. Corradini, H. Ehrig, M. Löwe, U. Montanari, and F. Rossi. Abstract graph derivations in the double pushout approach. In *Graph Transformations in Computer Science*, pages 86–103. Springer Berlin Heidelberg, 1994. DOI: 10.1007/3-540-57787-4_6.

[26] Andrea Corradini, Tobias Heindel, Frank Hermann, and Barbara König. Sesqui-Pushout Rewriting. In *Lecture Notes in Computer Science*, pages 30–45. Springer Berlin Heidelberg, 2006. DOI: 10.1007/11841883_4.

[27] Vincent Danos and Cosimo Laneve. Graphs for Core Molecular Biology. *Lecture Notes in Computer Science*, pages 34–46, 2003. DOI: 10.1007/3-540-36481-1_4.

[28] Vincent Danos and Cosimo Laneve. Core Formal Molecular Biology. *Lecture Notes in Computer Science*, pages 302–318, 2003. DOI: 10.1007/3-540-36575-3_21.

[29] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004. DOI: 10.1016/j.tcs.2004.03.065.

[30] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-Based Modelling of Cellular Signalling. *Lecture Notes in Computer Science*, pages 17–41, 2007. DOI: 10.1007/978-3-540-74407-8_3.

[31] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable Simulation of Cellular Signaling Networks. *Lecture Notes in Computer Science*, pages 139–157, 2007. DOI: 10.1007/978-3-540-76637-7_10.

[32] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling, symmetries, refinements. In Jasmin Fisher, editor, *Formal Methods in Systems Biology*, pages 103–122, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-68413-8_8.

[33] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Abstract Interpretation of Cellular Signalling Networks. *Lecture Notes in Computer Science*, pages 83–97, 2008. DOI: 10.1007/978-3-540-78163-9_11.

[34] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Abstracting the differential semantics of rule-based models: exact and automated model reduction. In *Logic in Computer Science (LICS), 2010 25th Annual IEEE Symposium on*, pages 362–381. IEEE, 2010.

[35] Vincent Danos, Jerome Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Chris Thompson-Walsh, and Glynn Winskel. Graphs, Rewriting and Pathway Reconstruction for Rule-Based Models. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 276–288,

Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. DOI: 10.4230/LIPIcs.FSTTCS.2012.276.

[36] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. Fundamentals of Algebraic Graph Transformation. *Monographs in Theoretical Computer Science (An EATCS Series)*, 2006. DOI: 10.1007/3-540-31188-2.

[37] Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1-3. world Scientific, 1997. DOI: 10.1142/3303.

[38] Hartmut Ehrig, Ulrike Golas, Annegret Habel, Leen Lambers, and Fernando Orejas. $\mathcal{M}$-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation. *Mathematical Structures in Computer Science*, 24(04), 2014. DOI: 10.1017/s0960129512000357.

[39] Hartmut Ehrig, Ulrike Golas, Annegret Habel, Leen Lambers, and Fernando Orejas. $\mathcal{M}$-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation. *Mathematical Structures in Computer Science*, 24(04), jun 2014. DOI: 10.1017/s0960129512000357. URL `https://doi.org/10.1017%2Fs0960129512000357`.

[40] Rolf Fagerberg, Christoph Flamm, Rojin Kianian, Daniel Merkle, and Peter F. Stadler. Finding the K best synthesis plans. *Journal of Cheminformatics*, 10(1), 2018. DOI: 10.1186/s13321-018-0273-z.

[41] Ulrike Golas, Hartmut Ehrig, and Annegret Habel. Multi-Amalgamation in Adhesive Categories. *Lecture Notes in Computer Science*, pages 346–361, 2010. DOI: 10.1007/978-3-642-15928-2_23.

[42] Ulrike Golas, Annegret Habel, and Hartmut Ehrig. Multi-amalgamation of rules with application conditions in $\mathcal{M}$-adhesive categories. *Mathematical Structures in Computer Science*, 24(04), jun 2014. DOI: 10.1017/s0960129512000345.

[43] Ulrike Golas, Annegret Habel, and Hartmut Ehrig. Multi-amalgamation of rules with application conditions in $\mathcal{M}$-adhesive categories. *Mathematical Structures in Computer Science*, 24(04), 2014. DOI: 10.1017/s0960129512000345.

[44] Annegret Habel and Karl-Heinz Pennemann. Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science*, 19(02):245, jan 2009. DOI: 10.1017/s0960129508007202. URL `https://doi.org/10.1017%2Fs0960129508007202`.

[45] Leonard A. Harris, Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P. Sheehan, and James R. Faeder. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, 32(21): 3366–3368, 2016. DOI: 10.1093/bioinformatics/btw469.

[46] Hans-Jörg Kreowski. Is parallelism already concurrency? Part 1: Derivations in graph grammars. In *Lecture Notes in Computer Science*, pages 343–360. Springer Berlin Heidelberg, 1987. DOI: 10.1007/3-540-18771-5_63.

[47] Stephen Lack and Paweł Sobociński. Adhesive and quasiadhesive categories. *RAIRO - Theoretical Informatics and Applications*, 39(3):511–545, 2005. DOI: 10.1051/ita:2005028.

[48] Karl-Heinz Pennemann and Karl-Heinz Pennemann. Resolution-Like Theorem Proving for High-Level Conditions. *Lecture Notes in Computer Science*, pages 289–304, 2008. DOI: 10.1007/978-3-540-87405-8_20.

[49] Grzegorz Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997. ISBN 9810228848. DOI: 10.1142/9789812384720.

A Background material: compositional rewriting theories

For the readers' convenience, we collect in this section a number of technical results and details on rewriting theories, most of which is either standard material from the rewriting theory, or quoted from our recent series of works [16–18].

A.1 Properties of adhesive categories

Theorem 3 ([17]) *Let $\mathbf{C}$ be a category satisfying Assumption 1. Then the following properties hold:*

(i) $\mathbf{C}$ has effective unions *(compare [47]): given a commutative diagram as in the left of Fig. 3, if the (b', c') is the pullback of the cospan of monomorphisms (b, c) (which by stability of monos under pullback entails that $b', c' \in \mathsf{mono}(\mathbf{C})$), and if (e, f) is the pushout of the span (b', c') (with $e, f \in \mathsf{mono}(\mathbf{C})$ by stability of monos under pushout), then the morphism d which exists by the universal property of pushouts is also a monomorphism.*

(ii) properties of final pullback complements (FPCs)[3] *in $\mathbf{C}$ (cf. the middle diagram in Fig. 3): for every pair of composable monomorphisms (c, a), there exists an FPC (d, b), and moreover $b, d \in \mathsf{mono}(\mathbf{C})$.*

(iii) characterization of epimorphisms via pushouts*: given a diagram such as on the right of Fig. 3, where all morphisms except e are monomorphisms, where the square (1) is a pushout, $e \circ d_i = e_i$ for $i = 1, 2$, and where (b_1, b_2) is the pullback of (e_1, e_2). Then the morphism e is an epimorphism if and only if the exterior square is a pushout.*

A.2 Conditions, shift and transport constructions

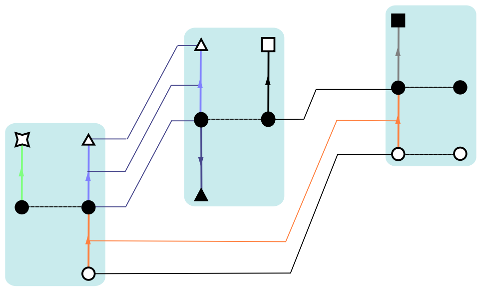
Definition 8 (Conditions) *Let $\mathbf{C}$ be a category satisfying Assumption 1. Then a* condition over an object *$X \in \mathsf{obj}(\mathbf{C})$, denoted c_X, is inductively defined as follows:*

- *$\mathsf{c}_X = \mathsf{true}$ is a condition over X.*
- *For every $(a : X \to A) \in \mathsf{mono}(\mathbf{C})$ and for every condition c_A over $A \in \mathsf{obj}(\mathbf{C})$, $\exists(a : X \to A, \mathsf{c}_A)$ is a condition over X.*
- *If c_X is a condition over X, so is its negation $\neg\mathsf{c}_X$.*
- *If $\mathsf{c}_X^{(i)}$ are conditions over X (for indices $i \in I$), then $\wedge_{i\in I}\mathsf{c}_X^{(i)}$ is a condition over X.*

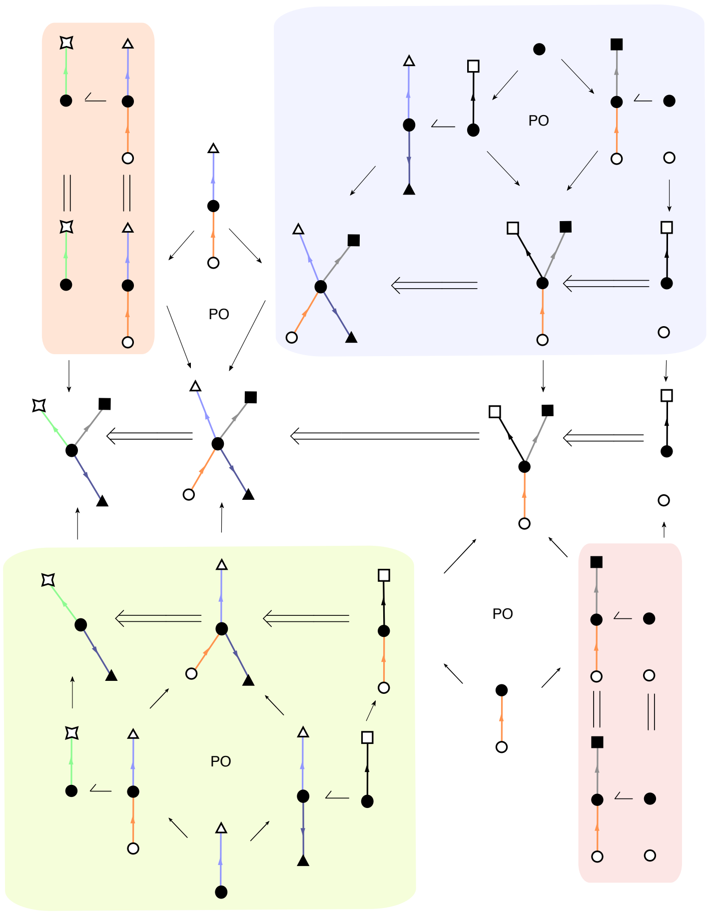
A concrete interpretation of conditions is provided by the accompanying definition of *satisfaction* of conditions.

---

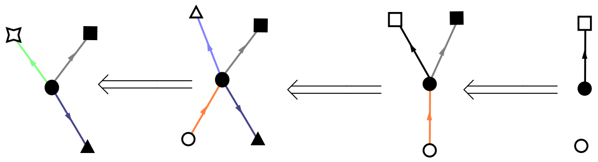
[3] Recall e.g. from [26] that for a pair of composable morphisms (c, a) such as in the middle part of Fig. 3, a pair of composable morphisms (d, b) is an FPC of (c, a) if (a, b) is the pullback of (c, d), and if the following universal property holds: given a cospan (c, z) such that (x, y) is the pullback of (c, z) and such that there exists a morphism w satisfying $z = a \circ w$, then there exists a unique (up to isomorphism) w^* such that $z = d \circ w^*$.



(a) Three rules sequentially composed (from right to left): input and output interfaces are drawn explicitly, while the context graphs K_j are implicitly encoded as subgraphs of O_j and I_j joined by dotted lines (for $j = 1, 2, 3$). The structure of the matches of the rules is indicated via lines connecting elements of outputs to elements of inputs of rules.

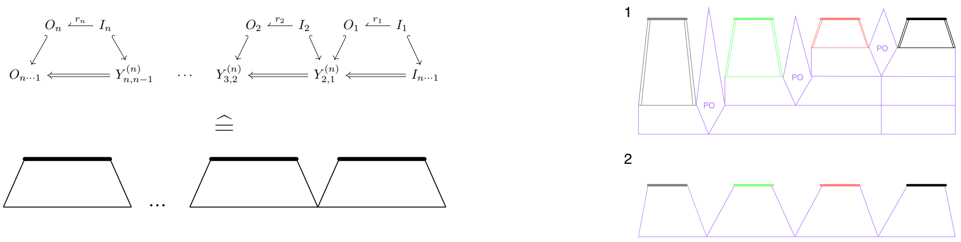


(b) Explicit demonstration of the associativity property of the rule composition operation: the top half of the diagram encodes a composition of the shape $r_3 \triangleleft (r_2 \triangleleft r_3)$, while the bottom half encodes $(r_3 \triangleleft r_2) \triangleleft r_1$, with both operations for the overlaps depicted leading to the same minimal trace (up to isomorphisms). The tracelet of length 3 equivalently encoded by both halves of the diagram is obtained by composition of squares.



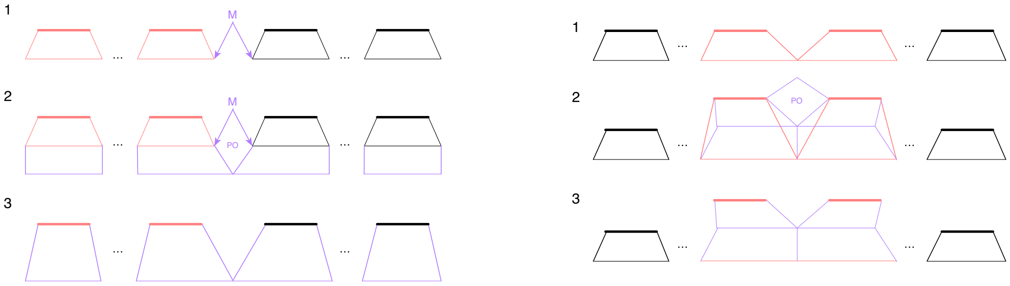
(c) The minimal trace of length 3 encoded in Fig. 1b.

Figure 1: Illustration of the relationship between associativity and tracelets.



(a) Tracelets as (minimal) derivation traces.

(b) Tracelet generation (Definition 1).



(c) Tracelet composition (Definition 2).

(d) Tracelet analysis (Section 3).

Figure 2: Schematic overview of the tracelet and tracelet analysis framework.

$$\begin{tikzcd}
 & A \arrow[dl, "b'"'] \arrow[dr, "c'"] & \\
B \arrow[dr, "e"] \arrow[ddr, bend right, "b"'] & & C \arrow[dl, "f"'] \arrow[ddl, bend left, "c"] \\
 & D \arrow[d, dotted, "\exists!\, d"] & \\
 & E &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & P \arrow[dll, bend right, "x"'] \arrow[dl, "w"] \arrow[ddd, "y"] \\
B \arrow[d, "c"'] & A \arrow[l, "a"'] \arrow[d, "b"] & \\
C & D \arrow[l, "d"] & \\
 & & Q \arrow[ul, dotted, "\exists!\, w^*"'] \arrow[ull, bend left, "z"]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & & X \arrow[dll, bend right, "b_1"'] \arrow[ddl, bend left, "b_2"] \\
 & B_1 \arrow[d, "d_1"'] \arrow[ddl, bend right, "e_1"'] \arrow[dr, phantom, "(1)"] & A \arrow[l, "a_1"'] \arrow[d, "a_2"] \arrow[ur, dotted, "x"] & \\
 & D \arrow[dl, dashed, "e"'] & B_2 \arrow[l, "d_2"] \arrow[dll, bend left, "e_2"] & \\
E & & &
\end{tikzcd}$$

Figure 3: from [17]: Effective unions (left), final pullback complements (FPCs) and their universal property (middle), and the epimorphism-pushout correspondence (right).

Definition 9 (Satisfaction) *Let $X \in \mathsf{obj}(\mathbf{C})$ be an object and c_X a condition over X. Then the* satisfaction of c_X by a monomorphism $(m : X \to Y) \in \mathsf{mono}(\mathbf{C})$*, denoted $m \vDash \mathsf{c}_X$, is inductively defined as follows:*

- *Every morphism satisfies $\mathsf{c}_X = \mathsf{true}$.*
- *For every $(a : X \to A) \in \mathsf{mono}(\mathbf{C})$ and for every condition c_A over $A \in \mathsf{obj}(\mathbf{C})$, the morphism $m : X \to Y$ satisfies $\exists(a : X \to A, \mathsf{c}_A)$ if there exists $(q : A \to Y) \in \mathsf{mono}(\mathbf{C})$ such that $m = q \circ a$ and $q \vDash \mathsf{c}_A$.*
- *m satisfies $\neg\mathsf{c}_X$ if it does not satisfy c_X.*
- *If $\mathsf{c}_X^{(i)}$ are conditions over X (with $i \in I$), m satisfies $\wedge_{i\in I}\mathsf{c}_X^{(i)}$ if $m \vDash \mathsf{c}_X^{(i)}$ for all $i \in I$.*

The notion of satisfaction of conditions permits to reason on equivalences of conditions:

Definition 10 (Equivalence) *Let $X \in \mathsf{obj}(\mathbf{C})$ be an object, and let $\mathsf{c}_X^{(1)}$ and $\mathsf{c}_X^{(2)}$ be two conditions over X. Then the two conditions are* equivalent*, denoted $\mathsf{c}_X^{(1)} \equiv \mathsf{c}_X^{(2)}$, iff*

$$\forall (m : X \to Y) \in \mathsf{mono}(\mathbf{C}) : \; m \vDash \mathsf{c}_X^{(1)} \Leftrightarrow m \vDash \mathsf{c}_X^{(2)} \,. \tag{19}$$

Besides the evident equivalences that arise from isomorphisms of rules and objects, some important classes of equivalences are implemented by the following two constructions quoted from [17], which are essential in our compositional rewriting framework (cf. Section A.3).

Definition 11 (Shift and Transport) *Let $\mathbf{C}$ be a category satisfying Assumption 1. Then for every condition c_A over $A \in \mathsf{obj}(\mathbf{C})$ and for every $(a_1 : A \to B_1)$, the* shift of the condition c_A over a_1*, denoted $\mathsf{Shift}(a_1, \mathsf{c}_A)$, is defined inductively as follows:*

- *$\mathsf{Shift}(a_1, \mathsf{true}) := \mathsf{true}$ (over B_1).*
- *If $\mathsf{c}_A = \exists(a_2 : A \to B_2, \mathsf{c}_{B_2})$ for some $(a_2 : A \to B_2) \in \mathsf{mono}(\mathbf{C})$, then*

  $$\mathsf{Shift}(a_1 : A \to B_1, \exists(a_2, \mathsf{c}_{B_2})) := \bigvee_{(b_1,b_2)\in\mathcal{X}} \exists(e_1 : B_1 \to E, \mathsf{Shift}(e_2 : B_2 \to E, \mathsf{c}_{B_2}))\,, \tag{20}$$

  where the set $\mathcal{X}$ is the set of all isomorphism classes[4] *of spans (b_1, b_2) as in the right part of Figure 3, where (e_1, e_2) are constructed as the pushout of (b_1, b_2).*
- *$\mathsf{Shift}(a_1, \neg\mathsf{c}_A) := \neg\mathsf{Shift}(a_1, \mathsf{c}_A)$ and $\mathsf{Shift}(a_1, \vee_{i\in I}\mathsf{c}_A^{(i)}) := \vee_{i\in I}\mathsf{Shift}(a_1, \mathsf{c}_A^{(i)})$.*

---

[4] Note that our improvement over the original variant of this construction as presented in [38, 44] consists in the precise characterization of the contributions to $\mathsf{Shift}(\exists(a_2, \mathsf{c}_{B_2}))$ via constructing pushouts of the possible spans (b_1, b_2) (rather than via the original indirect characterization in terms of listing the possible epimorphisms $(e : D \to E) \in \mathsf{epi}(\mathbf{C})$), a result which relies on Theorem 3(iii), and which is of central importance to proving the compositionality and associativity of rules with conditions.

We also define the operations of transporting a condition c_O over the output object O of a linear rule $r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})$ to the input object I of r. *The construction is denoted* $\mathsf{Trans}(r, \mathsf{c}_O)$ *and is defined inductively as follows:*

- $\mathsf{Trans}(r, \mathsf{true}) := \mathsf{true}$ *(over I).*
- *If* $\mathsf{c}_O = \exists(b : O \rightarrow B, \mathsf{c}_B)$ *for some* $b \in \mathsf{mono}(\mathbf{C})$, *then if* $b \notin \mathsf{M}_r^{DPO^\dagger}(B)$, *we define* $\mathsf{Trans}(r, \exists(b : O \rightarrow B, \mathsf{c}_B)) := \mathsf{false}$ *(as a condition over I). Otherwise, i.e. if* $b \in \mathsf{M}_r^{DPO^\dagger}(B)$, *we let (referring to Definition 12 for the definition of* $DPO^\dagger$*)*

$$\begin{aligned}&\mathsf{Trans}(r, \exists(b : O \rightarrow B, \mathsf{c}_B))\\ &\quad := \exists(b^* : I \rightarrow B', \mathsf{Trans}(B \Leftarrow B', \mathsf{c}_B))\end{aligned}\,, \quad \textit{where} \tag{21}$$

$$\begin{tikzcd} O \arrow[d, "b"'] \arrow[dr, phantom, "DPO^\dagger"] & I \arrow[l, harpoon, "r"'] \arrow[d, "b^*"] \\ B & B' \arrow[l, Rightarrow] \end{tikzcd}\,.$$

- $\mathsf{Trans}(r, \neg\mathsf{c}_O) := \neg\mathsf{Trans}(r, \mathsf{c}_O)$ *and* $\mathsf{Trans}(r, \vee_{i \in I}\mathsf{c}_O^{(i)}) := \vee_{i \in I}\mathsf{Trans}(r, \mathsf{c}_O^{(i)})$.

Theorem 4 (Properties of shift and transport constructions [17]; compare [43]) *Let* $\mathbf{C}$ *be a category satisfying Assumption 1.*

(i) Shift and satisfaction: *for* $X \in \mathsf{obj}(\mathbf{C})$, c_X *a condition over X and* $(m : X \rightarrow Y) \in \mathsf{mono}(\mathbf{C})$, *then for monomorphisms* $(q : Y \rightarrow Z) \in \mathsf{mono}(\mathbf{C})$ *it holds that*

$$q \vDash \mathsf{Shift}(m : X \rightarrow Y, \mathsf{c}_X) \;\Leftrightarrow\; q \circ m \vDash \mathsf{c}_X\,. \tag{22}$$

(ii) Unit for Shift: *for every object* $X \in \mathsf{obj}(\mathbf{C})$ *and for every* $(X \xrightarrow{\cong} X') \in \mathsf{iso}(\mathbf{C})$,

$$\mathsf{Shift}(X \xrightarrow{\cong} X', \mathsf{c}_X) \equiv \mathsf{c}_X\,. \tag{23}$$

(iii) Compositionality of Shift: *given composable monomorphisms* $(f : X \rightarrow Y), (g : Y \rightarrow Z) \in \mathsf{mono}(\mathbf{C})$ *and a condition* c_X *over* X,

$$\mathsf{Shift}(g : Y \rightarrow Z, \mathsf{Shift}(f : X \rightarrow Y, \mathsf{c}_X)) \equiv \mathsf{Shift}(g \circ f : X \rightarrow Z, \mathsf{c}_X)\,. \tag{24}$$

(iv) Satisfiability for Trans: *for* $r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})$, c_O *a condition over O and* $X \in \mathsf{obj}(\mathbf{C})$, *denoting for an admissible match* $m \in \mathsf{M}_r^{\mathbb{T}}(X)$ *(for* $\mathbb{T} \in \{DPO, SqPO\}$*) the comatch for m under a $\mathbb{T}$-type rule application by m^* (cf. (30)),*

$$\forall m \in \mathsf{M}_r^{\mathbb{T}}(X) : \quad m \vDash \mathsf{Trans}(r, \mathsf{c}_O) \Leftrightarrow m^* \vDash \mathsf{c}_O\,. \tag{25}$$

We write $\mathsf{Trans}(r, \mathsf{c}_O) \doteqdot_{\mathbb{T}} \mathsf{c}_O$ *to indicate this* equivalence up to $\mathbb{T}$-type admissibility.

(v) Units for Trans: *for each span of isomorphisms* $(Z \xleftarrow{\cong} Y \xrightarrow{\cong} X) \in \mathsf{Lin}(\mathbf{C})$ *and for each condition* c_Z *over* Z,

$$\mathsf{Trans}((Z \xleftarrow{\cong} Y \xrightarrow{\cong} X), \mathsf{c}_X) \equiv \mathsf{c}_Z\,. \tag{26}$$

(vi) Compositionality of Trans: *for composable spans* $s = (E \leftarrow D \rightarrow C), r = (C \leftarrow B \rightarrow A) \in \mathsf{Lin}(\mathbf{C})$ *and* c_E *a condition over* E,

$$\mathsf{Trans}(s, \mathsf{Trans}(r, \mathsf{c}_E)) \equiv \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,. \tag{27}$$

(vii) Compatibility of Shift with Trans: *for* $r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})$, $(m : I \rightarrow X) \in \mathsf{mono}(\mathbf{C})$, c_O *a condition over* O *and* $\mathbb{T} \in \{DPO, SqPO\}$,

$$m \in \mathsf{M}^{\mathbb{T}}_r(X) \;\Rightarrow \mathsf{Shift}(m, \mathsf{Trans}(r, \mathsf{c}_O)) \doteq_{\mathbb{T}} \mathsf{Trans}((r_m(X) \xLeftarrow[r,m]{\mathbb{T}} X), \mathsf{Shift}(m^*, \mathsf{c}_O))\,. \tag{28}$$

A.3 Compositional Double- and Sesqui-Pushout rewriting

We present here our recently introduced refinement of the DPO-type such framework, as well as the first of its kind such framework for compositional SqPO-type rewriting theories [17]. For the DPO-type case, our developments generalized earlier work by Habel and Pennemann [44, 48] and by Ehrig and collaborators [38, 42], while the SqPO-constructions were new.

The core definitions of rewriting are the following two sets of definitions, which provide the semantics of rule applications to objects and compositions of linear rules:

Definition 12 (Rules and rule applications) *Let* $\mathbf{C}$ *be a category satisfying Assumption 1. Denote by* $\mathsf{Lin}(\mathbf{C})$ *the set of isomorphism classes of spans of monomorphisms,*

$$\mathsf{Lin}(\mathbf{C}) := \left\{ O \xleftarrow{o} K \xrightarrow{i} I \,\middle|\, o, i \in \mathsf{mono}(\mathbf{C}) \right\} \big/ _{\cong}\,, \tag{29}$$

referred to henceforth as the set of linear rules (on $\mathbf{C}$)*. Here, two rules* $r_i = (O_j \leftarrow K_j \rightarrow I_j) \in \mathsf{Lin}(\mathbf{C})$ *(for* $j = 1, 2$*) are representatives of the same isomorphism class if there exist isomorphisms* $\omega : O_1 \rightarrow O_2$, $\kappa : K_1 \rightarrow K_2$ *and* $\iota : I_1 \rightarrow I_2$ *that make the evident diagram commute*[5]*. Let* $r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})$ *be a linear rule,* $X \in \mathsf{obj}(\mathbf{C})$ *an object, and* $(m : I \rightarrow X) \in \mathsf{mono}(\mathbf{C})$ *be a monomorphism. A* rule application *of the rule* r *to the object* X *along an admissible match* m *is defined via the following type of commutative diagram (referred to as* a direct derivation *in the literature):*

$$\begin{tikzcd} O \arrow[d] & I \arrow[l, "r"'] \arrow[d] \arrow[dl, phantom, "\mathbb{T}"] \\ r_m(X) & X \arrow[l, Leftarrow] \end{tikzcd} \quad := \quad \begin{tikzcd} O \arrow[d, dotted, "m^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, dashed] \arrow[dl, phantom, "(B)"] & I \arrow[d, "m"] \arrow[dl, phantom, "(A)"] \\ Y & \overline{K} \arrow[l, dotted] \arrow[r, dashed] & X \end{tikzcd} \;. \tag{30}$$

Here, the precise construction depends on the type $\mathbb{T}$ *of rewriting:*

- *(i)* $\mathbb{T} = DPO$ (**Double-Pushout (DPO) rewriting**): *given* $(m : I \rightarrow X) \in \mathsf{mono}(\mathbf{C})$, m *is defined to be an* admissible match *if the square marked* (A) *in* (30) *is constructible as a* pushout complement (POC)*; in this case, the square marked* (B) *is constructed as a* pushout (PO).
- *(ii)* $\mathbb{T} = DPO^{\dagger}$ (**"opposite" of Double-Pushout rewriting**): *for this auxiliary rewriting semantics, given* $(m^* : O \rightarrow Y) \in \mathsf{mono}(\mathbf{C})$, m^* *is defined to be an* admissible match *if*

---

[5] We will henceforth speak about isomorphism classes of rules and objects and their concrete representatives interchangeably, since all constructions presented afford a clear notion of invariance under isomorphisms. This feature of the constructions also motivates the notion of abstraction equivalence standard in the rewriting theory (cf. Section 3).

the square marked (B) in ([30]) is constructible as a pushout complement (POC)*; in this case, the square marked (A) is constructed as a* pushout (PO)*. Coincidentally, a $DPO^{\dagger}$-admissible match m^* uniquely induces a DPO-admissible match m for r applied to X, which is why m^* is sometimes referred to as the* comatch of m*.*

(iii) $\mathbb{T} = SqPO$ (**Sesqui-Pushout (SqPO) rewriting**)*: given $(m : I \to X) \in \mathsf{mono}(\mathbf{C})$, the square (A) is constructed as a* final pullback complement (FPC)*, followed by constructing (B) as a pushout.*

We denote the set of $\mathbb{T}$-admissible matches *by $\mathsf{M}^{\mathbb{T}}_{r}(X)$. We will moreover adopt the traditional "direct derivation" notation $r_m(X) \xLeftarrow[r,m]{\mathbb{T}} X$ for $\mathbb{T} = DPO$ or $\mathbb{T} = SqPO$ as a compact notation for the process of applying rule r to X along admissible match m.*

Crucially, linear rules in both types of rewriting semantics admit a *composition operation*:

Definition 13 (Rule composition) *Let $\mathbf{C}$ be a category satisfying Assumption [1]. Let $r_1, r_2 \in \mathsf{Lin}(\mathbf{C})$ be two linear rules. We define the* set of $\mathbb{T}$-type admissible matches of r_2 into r_1 *for $\mathbb{T} \in \{DPO, SqPO\}$, denoted $\mathsf{M}^{\mathbb{T}}_{r_2}(r_1)$, as*

$$\begin{aligned} \mathsf{M}^{\mathbb{T}}_{r_2}(r_1) := \big\{\mu_{21} = (I_2 \leftarrow M_{21} \rightarrow O_2) \big| n_1, n_2 \text{ in } \mathsf{PO}(\mu_{21}) = (I_2 \xrightarrow{n_2} N_{21} \xleftarrow{n_1} O_1) \\ \textit{satisfy } n_2 \in \mathsf{M}^{\mathbb{T}}_{r_2}(N_{21}) \ \wedge \ n_1 \in \mathsf{M}^{DPO^{\dagger}}_{r_1}(N_{21})\big\}\,. \end{aligned} \tag{31}$$

For a $\mathbb{T}$-type admissible match $\mu_{21} = (I_2 \leftarrow M_{21} \rightarrow O_2) \in \mathsf{M}^{\mathbb{T}}_{r_2}(r_1)$, construct the diagram

$$\begin{tikzcd}
O_2 \arrow[d] \arrow[dr, phantom, "\mathbb{T}"] & I_2 \arrow[l, harpoon', "r_2"'] \arrow[r, equal] \arrow[dr, "n_2"'] & M_{21} \arrow[r, equal] \arrow[d, phantom, "\mathsf{PO}"] & O_1 \arrow[dl, "n_1"] \arrow[dr, phantom, "DPO^{\dagger}"] & I_1 \arrow[l, harpoon', "r_1"'] \arrow[d] \\
O_{21} & & N_{21} \arrow[ll, Rightarrow] & & I_{21} \arrow[ll, Rightarrow]
\end{tikzcd} \;. \tag{32}$$

From this diagram, one may compute (via pullback composition $\circ$ of the two composable spans in the bottom row) a span of monomorphisms $(O_{21} \Leftarrow I_{21}) \in \mathsf{Lin}(\mathbf{C})$, which we define to be the $\mathbb{T}$-type composition of r_2 with r_1 along μ_{21} *(for $\mathbb{T} \in \{DPO, SqPO\}$ as in ([32])):*

$$r_2{}^{\mu_{21}}\!\triangleleft_{\mathbb{T}} r_1 := (O_{21} \Leftarrow I_{21}) = (O_{21} \Leftarrow N_{21}) \circ (N_{21} \Leftarrow I_{21})\,. \tag{33}$$

We refer the interested readers to [16, 18] for the precise derivations of these notions of rule compositions, and note here that the definitions are justifiable a posteriori via the concurrency theorems as presented in Section A.5.

The compositional rewriting framework may be extended to the setting of rules with conditions as follows[6]:

Definition 14 (Rewriting with conditions [17]) *Let $\mathbf{C}$ be a category satisfying Assumption [1]. We denote by $\overline{\mathsf{Lin}}(\mathbf{C})$ the* set of linear rules with conditions, *defined as*

$$\overline{\mathsf{Lin}}(\mathbf{C}) := \{R = (r, \mathsf{c}_I) \mid r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})\}\,. \tag{34}$$

---

[6] Referring to [17] for the precise details, based upon the transport construction we may from hereon without loss of generality consider rewriting rules with conditions over the input objects only.

We extend the definitions of rule applications (Definition 12) and rule compositions (Definition 13) to the setting of rules with conditions as follows: for $R = (r, \mathsf{c}_I) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *and* $X \in \mathsf{obj}(\mathbf{C})$, *define the* sets of $\mathbb{T}$-type admissible matches of R into X *(for* $\mathbb{T} \in \{DPO, SqPO\}$*), denoted* $\mathsf{M}^{\mathbb{T}}_R(X)$*, as*

$$\mathsf{M}^{\mathbb{T}}_R(X) := \{m \in \mathsf{M}^{\mathbb{T}}_r(X) \mid m \vDash \mathsf{c}_I\}\,. \tag{35}$$

Then the $\mathbb{T}$-type rule application of R along $m \in \mathsf{M}^{\mathbb{T}}_R(X)$ to X *is defined as* $r_m(X) \overset{\mathbb{T}}{\Leftarrow} X$.

As for the rule compositions, we define for two rules with application conditions $R_j = (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ $(j = 1, 2)$ *the* sets of $\mathbb{T}$-admissible matches of R_2 into R_1 *as*

$$\mathsf{M}^{\mathbb{T}}_{R_2}(R_1) := \{\mu_{21} \in \mathsf{M}^{\mathbb{T}}_{r_2}(r_1) \mid \mathsf{c}_{I_{21}} \not\equiv \mathsf{false}\}\,, \tag{36}$$

where the condition $\mathsf{c}_{I_{21}}$ *for a given rule composite is defined as (compare (32) for the defining construction of the various morphisms)*

$$\mathsf{c}_{I_{21}} := \mathsf{Shift}(I_1 \to I_{21}, \mathsf{c}_{I_1}) \bigwedge \mathsf{Trans}(N_{21} \leftarrow \overline{K}_1 \to I_{21}, \mathsf{Shift}(I_2 \to N_{21}, \mathsf{c}_{I_2}))\,. \tag{37}$$

Then we define for admissible matches μ_{21} *the compositions as*

$$\forall \mu_{21} \in \mathsf{M}^{\mathbb{T}}_{R_2}(R_1): \quad R_2\,{}^{\mu_{21}}\!\!\lhd_{\mathbb{T}} R_1 := (r_2\,{}^{\mu_{21}}\!\!\lhd_{\mathbb{T}} r_1, \mathsf{c}_{I_{21}})\,. \tag{38}$$

A.4 Auxiliary properties of direct derivations

Lemma 1 *Let* $\mathbf{C}$ *be a category satisfying Assumption 1, and let* $\mathbb{T} \in \{DPO, SqPO, DPO^{\dagger}\}$.

(i) For every rule $R = (r, \mathsf{c}_I) \in \overline{\mathsf{Lin}}(\mathbf{C})$*, the diagram below is a* $\mathbb{T}$*-type direct derivation for arbitrary* $\mathbb{T} \in \{DPO, DPO^{\dagger}, SqPO\}$:

$$\left(\begin{tikzcd} O \arrow[d, equal] \arrow[dr, phantom, "(1)"] & K \arrow[l, hook', "o"'] \arrow[r, hook, "i"] \arrow[d, equal] \arrow[dr, phantom, "(2)"] & I \arrow[d, equal] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \\ O & K \arrow[l, hook'] \arrow[r, hook] & I \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \end{tikzcd}\right) = \left(\begin{tikzcd} O \arrow[d, equal] \arrow[dr, phantom, "\mathbb{T}"] & I \arrow[l, harpoon', "r"'] \arrow[d, equal] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \\ O & I \arrow[l, Leftarrow] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \end{tikzcd}\right). \tag{39}$$

(ii) Vertical pasting: *for* $\mathbb{T} \in \{DPO, DPO^{\dagger}, SqPO\}$*, and suppose that in the diagrams below the monomorphism* $(X' \hookleftarrow X) \circ (X \hookleftarrow I)$ *satisfies the condition* c_I *(not explicitly drawn for clarity). Then the following properties hold true: composing squares of the underlying commutative diagrams vertically (indicated by the* $\rightsquigarrow$ *notation), one obtains (a) for all combinations of types of rewriting with* $\mathbb{T} = \mathbb{T}'$*, or (b) for* $\mathbb{T}$ *arbitrary and* $\mathbb{T} = DPO$ *or* $\mathbb{T} = DPO^{\dagger}$*, the following* $\mathbb{T}$*-type direct derivations:*

$$\begin{tikzcd} O \arrow[d] \arrow[dr, phantom, "\mathbb{T}"] & I \arrow[l, harpoon', "r"'] \arrow[d] \\ Y \arrow[d] \arrow[dr, phantom, "\mathbb{T}'"] & X \arrow[l, Leftarrow] \arrow[d] \\ Y' & X' \arrow[l, Leftarrow] \end{tikzcd} \quad \rightsquigarrow \quad \begin{tikzcd} O \arrow[d] \arrow[dr, phantom, "\mathbb{T}"] & I \arrow[l, harpoon', "r"'] \arrow[d] \\ Y' & X' \arrow[l, Leftarrow] \end{tikzcd} \tag{40}$$

Proof: The first statement follows since the squares marked (1) and (2) are pushouts for arbitrary morphisms o and i. The second property follows by invoking various elementary square composition lemmata for pushouts and FPCs (see e.g. the list of technical Lemmas provided in the appendix of [16]). □

A.5 Compositional concurrency

A central role in our rewriting frameworks is played by the notion of certain concurrency theorems, which entail an equivalence between (i) sequential applications of rewriting rules along admissible matches and (ii) application of *composites* of rewriting rules along admissible matches. This structure is in turn intimately related to notions of *traces* and analyses thereof in rewriting theory. In the form as presented below (which is compatible with the notion of compositionality of rewriting rules), both the DPO- and the SqPO-type concurrency theorems were first introduced in [17] (with some earlier results in the DPO-setting under weaker assumptions reported in [38, 44]).

Theorem 5 (Concurrency theorems [16–18]) *Let $\mathbf{C}$ be a category satisfying Assumption 1, and let $\mathbb{T} \in \{DPO, SqPO\}$. Then there exists a bijection $\varphi : A \xrightarrow{\cong} B$ on pairs of $\mathbb{T}$-admissible matches between the sets A and B,*

$$\begin{aligned} & A = \{(m_2, m_1) \mid m_1 \in \mathsf{M}^{\mathbb{T}}_{R_1}(X_0)\,,; m_2 \in \mathsf{M}^{\mathbb{T}}_{R_2}(X_1)\} \\ \cong \quad & B = \{(\mu_{21}, m_{21}) \mid \mu_{21} \in \mathsf{M}^{\mathbb{T}}_{R_2}(R_1)\,,\ m_{21} \in \mathsf{M}^{\mathbb{T}}_{R_{21}}(X_0)\}\,, \end{aligned} \tag{41}$$

where $X_1 = R_{1_{m_1}}(X_0)$ and $R_{21} = R_2{}^{\mu_{21}}\!\!\lhd_{\mathbb{T}} R_1$ such that for each corresponding pair $(m_2, m_1) \in A$ and $(\mu_{21}, m_{21}) \in B$, it holds that

$$R_{21_{m_{21}}}(X_0) \cong R_{2_{m_2}}(R_{1_{m_1}}(X_0))\,. \tag{42}$$

Proof: See [17] for the full technical details. Let us note here that the bijective correspondence is *constructive*, i.e. there exist explicit algorithms for realizing B from A and vice versa. □

The following technical result is a necessary prerequisite for deriving the proof of Theorem 2:

Corollary 2 *Let $r_{n\cdots 1} = (O_{n\cdots 1} \Leftarrow I_{n\cdots 1})$ be a span of monomorphisms, and let $(Y^{(n)}_{j+1,j} \Leftarrow Y^{(n)}_{j,j-1})$ be n spans of monomorphisms (for $0 \leq j \leq n$) with $Y^{(n)}_{n+1,n} = O_{n\cdots 1}$, $Y^{(n)}_{1,0} = I_{n\cdots 1}$, and such that*

$$(O_{n\cdots 1} \Leftarrow I_{n\cdots 1}) = (O_{n\cdots 1} \Leftarrow Y^{(n)}_{n,n-1}) \circ \ldots \circ (Y^{(n)}_{2,1} \Leftarrow I_{n\cdots 1})\,.$$

Let $\mathsf{c}_{I_{n\cdots 1}}$ be a condition over $I_{n\cdots 1}$. Then for each object X_0 and for each $\mathbb{T}$-admissible match $(I_{n\cdots 1} \hookrightarrow X_0) \in \mathsf{M}^{\mathbb{T}}_{R_{n\cdots 1}}(X_0)$ (for $R_{n\cdots 1} = (r_{n\cdots 1}, \mathsf{c}_{I_{n\cdots 1}})$ and for $\mathbb{T} \in \{DPO, SqPO\}$), the $\mathbb{T}$-type *application of $R_{n\cdots 1}$ to X_0 along this match*

$$\begin{tikzcd}
O_{n\cdots1} \arrow[d, hook] \arrow[dr, phantom, "\mathbb{T}"] & I_{n\cdots1} \blacktriangleleft \mathsf{c}_{I_{n\cdots1}} \arrow[l, Leftarrow] \arrow[d, hook] \\
X_n & X_0 \arrow[l, Leftarrow]
\end{tikzcd} \tag{43a}$$

uniquely (up to isomorphisms) encodes an n-step $\mathbb{T}$-type derivation sequence of the following form, and vice versa:

$$\begin{tikzcd}
O_{n\cdots1} \arrow[d, hook] \arrow[dr, phantom, "\mathbb{T}"] & Y^{(n)}_{n,n-1} \arrow[l, Leftarrow] \arrow[d, hook] & \cdots & Y^{(n)}_{2,1} \arrow[d, hook] \arrow[dr, phantom, "\mathbb{T}"] & I_{n\cdots1} \blacktriangleleft \mathsf{c}_{I_{n\cdots1}} \arrow[l, Leftarrow] \arrow[d, hook] \\
X_n & X_{n-1} \arrow[l, Leftarrow] & \cdots & X_1 & X_0 \arrow[l, Leftarrow]
\end{tikzcd} \tag{43b}$$

Proof: *The statement is trivially true for $n = 1$. For $n = 2$, note first that for any two composable spans of monomorphisms $S_2 = (Z \leftarrow Y)$ and $S_1 = (Y \leftarrow X)$, invoking the definition of rule compositions (Definition 13) allows to verify that considering S_2 and S_1 as linear rules without conditions, and letting $\mu = (Y \hookleftarrow Y \hookrightarrow Y)$ be a span of identity morphisms on Y, the span composition $S_2 \circ S_1$ is in fact computable as a rule composition:*

$$\begin{tikzcd}
Z \arrow[d, equal] \arrow[dr, phantom, "\mathbb{T}"] & Y \arrow[l, Leftarrow] \arrow[r, equal] \arrow[dr, equal] & Y \arrow[d, phantom, "\mathsf{PO}"] & Y \arrow[l, equal] \arrow[dl, equal] \arrow[dr, phantom, "DPO^{\dagger}"] & X \arrow[l, Leftarrow] \arrow[d, equal] \\
Z & & Y \arrow[ll, Leftarrow] & & X \arrow[ll, Leftarrow]
\end{tikzcd} \quad \rightsquigarrow \quad S_2 \circ S_1 = S_2{}^{\mu}\!\!\triangleleft_{\mathbb{T}} S_1\,. \tag{44}$$

Here, according to Lemma 1(i) the $\mathbb{T}$- and $DPO^{\dagger}$-type direct derivation subdiagrams as indicated always exist. Consequently, the claim of the corollary for $n = 2$ follows by invoking the concurrency theorem (Theorem 5) for the special case of $r_{2\cdots1} = (O_{2\cdots1} \Leftarrow I_{2\cdots1})$, $(O_{2\cdots1} \Leftarrow I_{2\cdots1}) = (O_{2\cdots1} \Leftarrow Y^{(2)}_{2,1}) \circ (Y^{(2)}_{2,1} \Leftarrow I_{2\cdots1})$ and for μ_{21} a span of identity morphisms of $Y^{(2)}_{2,1}$. The proof for the case $n \geq 2$ may then be obtained via induction over n. $\square$

A.6 Compositional associativity

The second main ingredient of our novel *compositional* rewriting frameworks is the notion of *associativity*. Intuitively, if one wishes to extend the analysis of traces as suggested via the concurrency theorems to a full-fledged analysis that is centered on compositions of rewriting rules (which constitutes the main contribution of this paper in the form of the tracelet framework), one must necessarily have a certain property fulfilled, in that multiple sequential compositions of rewriting rules may be computed in any admissible order of pairwise compositions. The latter feature is crucial for the purposes of analysis of classes of traces, since the traditional interpretation of the concurrency theorem would only permit to reason on pairwise sequential compositions (but not on extension thereof to higher order composites).

Theorem 6 (Associativity of rule compositions [16–18]) *Let* $\mathbf{C}$ *be a category satisfying Assumption 1. let* $R_1, R_2, R_3 \in \overline{\mathsf{Lin}}(\mathbf{C})$ *be linear rules with conditions, and let* $\mathbb{T} \in \{DPO, SqPO\}$*. Then there exists a bijection* $\varphi : A \xrightarrow{\cong} B$ *of sets of pairs of* $\mathbb{T}$*-admissible matches* A *and* B*, defined as*

$$\begin{aligned} A &:= \{(\mu_{21}, \mu_{3(21)}) \mid \mu_{21} \in \mathsf{M}^{\mathbb{T}}_{R_2}(R_1)\,,\ \mu_{3(21)} \in \mathsf{M}^{\mathbb{T}}_{R_3}(R_{21})\} \qquad (R_{21} = R_2{}^{\mu_{21}}\!\lhd_{\mathbb{T}} R_1) \\ B &:= \{(\mu_{32}, \mu_{(32)1}) \mid \mu_{32} \in \mathsf{M}^{\mathbb{T}}_{R_3}(R_2)\,,\ \mu_{(32)1} \in \mathsf{M}^{\mathbb{T}}_{R_{32}}(R_1)\} \qquad (R_{32} = R_3{}^{\mu_{32}}\!\lhd_{\mathbb{T}} R_2) \end{aligned} \tag{45}$$

such that for each corresponding pair $(\mu_{21}, \mu_{3(21)}) \in A$ *and* $\varphi(\mu_{21}, \mu_{3(21)}) = (\mu'_{32}, \mu'_{(32)1}) \in B$*,*

$$R_3{}^{\mu_{3(21)}}\!\lhd_{\mathbb{T}} \left(R_2{}^{\mu_{21}}\!\lhd_{\mathbb{T}} R_1\right) \cong \left(R_3{}^{\mu'_{32}}\!\lhd_{\mathbb{T}} R_2\right){}^{\mu'_{(32)1}}\!\lhd_{\mathbb{T}} R_1\,. \tag{46}$$

In this particular sense, the composition operations $.\lhd_{\mathbb{T}}.$ *are* ***associative****.*

A.7 Proof of Theorem 1

Ad part (i): Note first that by virtue of Corollary 2, the composition of two $\mathbb{T}$-type tracelets $T', T \in \mathcal{T}^{\mathbb{T}}$ of lengths m and n, respectively, along a $\mathbb{T}$-admissible match $\mu = (I'_{m\cdots1} \hookleftarrow M \hookrightarrow O_{n\cdots1}) \in \mathsf{MT}^{\mathbb{T}}_{T'}(T)$ encodes a $\mathbb{T}$-type composition of linear rules $R'_{m\cdots1} = (r'_{m\cdots1}, \mathsf{c}_{I'_{m\cdots1}})$ and $R_{n\cdots1} = (r_{n\cdots1}, \mathsf{c}_{I_{n\cdots1}})$ (with $r'_{m\cdots1} = (O'_{m\cdots1} \Leftarrow I'_{m\cdots1})$ and $r_{n\cdots1} = (O_{n\cdots1} \Leftarrow I_{n\cdots1})$) of the following form:

$$\begin{tikzcd} & \mathsf{c}_{I'_{m\cdots1}} \arrow[d, phantom, "\blacktriangledown"] & M \arrow[dl, hook'] \arrow[dr, hook] & & \mathsf{c}_{I_{n\cdots1}} \arrow[d, phantom, "\blacktriangledown"] & \\ O'_{m\cdots1} \arrow[d, hook] \arrow[drr, phantom, "\mathbb{T}"] & I'_{m\cdots1} \arrow[l, hook', "r'_{m\cdots1}"'] \arrow[dr] & \mathsf{PO} & O_{n\cdots1} \arrow[dl] & I_{n\cdots1} \arrow[l, hook', "r_{n\cdots1}"'] \arrow[d, hook] \arrow[dll, phantom, "DPO^{\dagger}"] & \\ O_{p\cdots1} & & Y^{(p)}_{n+1,n} \arrow[ll, Rightarrow] & & I_{p\cdots1} \arrow[ll, Rightarrow] & \mathsf{c}_{I_{p\cdots1}} \arrow[l, phantom, "\blacktriangleleft"] \end{tikzcd} \tag{47}$$

Consequently, the constructibility of this diagram indeed hinges on whether or not μ is a $\mathbb{T}$-type admissible match of $r'_{m\cdots1}$ into $r_{n\cdots1}$, thus proving the statement of part (i).
Ad part (ii): The latter argument has the additional consequence that for all tracelets $T', T \in \mathcal{T}^{\mathbb{T}}$ and $\mu \in \mathsf{MT}^{\mathbb{T}}_{T'}(T)$, equation (47) demonstrates explicitly that $[[T'{}^{\mu}\angle_{\mathbb{T}}T]] = [[T']]{}^{\mu}\!\lhd_{\mathbb{T}}[[T]]$, thus proving part (ii) of the theorem statement.
Ad part (iii): The proof of part (iii) of Theorem 1 follows by combining the statements of the first two parts with the associativity theorem for $\mathbb{T}$-type rule compositions (Theorem 6).

A.8 Proof of Theorem 2

The first part of the claim follows by applying a corollary of the concurrency theorem for rules with conditions (Corollary 2 of Appendix A.5) in order to construct the lower row in the left diagram of (13), followed by vertically composing squares (Lemma 1 of Appendix A.4) in each column of the diagram in order to obtain the derivation trace shown on the right of (13). The second part of the statement follows by an inductive application of the concurrency theorem: the case $n = 1$ coincides with the definition of a direct derivation, while for $n = 2$ Theorem 5 precisely describes the transition from a length 2 derivation trace to a length 1 derivation trace along the composite rule. The induction step $n \to n + 1$ is then verified by applying the concurrency theorem to the derivation trace $X_{n+1} \Leftarrow X_n \Leftarrow X_0$ along the rules r_{n+1} and $(O_{n\ldots 1} \Leftarrow I_{n\ldots 1})$.

B Compositional sequential independence

A key role in the analysis of rewriting theories is played by the notion of sequential independence, which we first recall in its traditional form as known from the rewriting literature:

Definition 15 (cf. e.g. [38], Def. 4.3 (DPO) and [14], Def. 2.15 (SqPO)) *Consider a two-step sequence of rule applications of type $\mathbb{T}$ to an initial object $X_0 \in \mathbf{C}$ along admissible matches,*

$$\begin{tikzcd}
O_2 \arrow[d, "m_2^*"'] \arrow[dr, phantom, "\mathsf{PO}"] & K_2 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathbb{t}"] & I_2 \arrow[drr, "m_2"] \arrow[rrrd, dashed, "n_2"] & & O_1 \arrow[dl, "m_1^*"'] \arrow[llld, dashed, "n_1"'] \arrow[dr, phantom, "\mathsf{PO}"] & K_1 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathbb{t}"] & I_1 \arrow[d, "m_1"] \\
X_2 & K_2' \arrow[l, "d_2^*"] \arrow[rr, "d_2"'] & & X_1 & & K_1' \arrow[ll, "d_1^*"] \arrow[r, "d_1"'] & X_0
\end{tikzcd}, \tag{48}$$

where the squares marked $\mathbb{t}$ for $\mathbb{T} = DPO$ are pushout complements and for $\mathbb{T} = SqPO$ FPCs. The two rule applications are called sequentially independent *if there exist monomorphisms $(n_1 : O_1 \to K_2'), (n_2 : I_2 \to K_1') \in \mathsf{mono}(\mathbf{C})$ such that*

$$d_2 \circ n_1 = m_1^* \quad \wedge \quad d_1^* \circ n_2 = m_2 \,. \tag{49}$$

Based upon the concurrency theorems for the DPO- and SqPO-type compositional rewriting theories, one may develop the following refined variant of the above definition, as was anticipated e.g. in [22] for the DPO-type setting:

Lemma 2 (Compositional sequential independence) *In the setting of Def. 15, the rule applications are sequentially independent if and only if there exist monomorphisms $(O_1 \to \overline{K}_2), (I_2 \to \overline{K}_1) \in \mathsf{mono}(\mathbf{C})$ such that*

$$(N_{21} \leftarrow \overline{K}_2) \circ (\overline{K}_2 \leftarrow O_1) = (N_{21} \leftarrow O_1) \quad \wedge \quad (N_{21} \leftarrow \overline{K}_1) \circ (\overline{K}_1 \leftarrow I_2) = (N_{21} \leftarrow I_2)\,, \tag{50}$$

with notations as in the explicit version of diagram (32) (see the proof). For the case of rules with conditions, it is in addition required that $\mathsf{c}_{I_{21}} \doteq \mathsf{c}_{I_{12}}$.

Proof: For the "$\Rightarrow$" direction, suppose the $\mathbb{T}$-type rule applications are sequentially independent. Invoking the $\mathbb{T}$-type concurrency theorem, we may construct the following diagram:

$$\begin{tikzcd}
O_2 \arrow[d] \arrow[dd, bend right, "m_2^*"'] \arrow[dr, phantom, "\mathsf{PO}"] & K_2 \arrow[l] \arrow[r] \arrow[d] \arrow[dd, bend right] \arrow[dr, phantom, "\mathbb{t}"] & I_2 \arrow[drr, dashed, bend left=10, "n_2"] \arrow[dr] \arrow[ddr, bend right, "m_2"'] & M_{21} \arrow[l] \arrow[r] & O_1 \arrow[dl] \arrow[ddl, bend left] \arrow[dll, dashed, bend right=10, "n_1"'] \arrow[dr, phantom, "\mathsf{POC}"] & K_1 \arrow[l] \arrow[r] \arrow[d] \arrow[dd, bend left] \arrow[dr, phantom, "\mathsf{PO}"] & I_1 \arrow[d] \arrow[dd, bend left, "m_1"] \\
O_{21} \arrow[d] \arrow[dr, phantom, "\mathsf{PO}"] & \overline{K}_2 \arrow[l] \arrow[rr] \arrow[d] \arrow[drr, phantom, "\mathbb{t}"] & & N_{21} \arrow[d, "m_1^*"] \arrow[drr, phantom, "\mathsf{PO}"] & & \overline{K}_1 \arrow[ll] \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathbb{t}"] & I_{21} \arrow[d] \\
X_2 & K_2' \arrow[l, "d_2^*"] \arrow[rr, "d_2"] & & X_1 & & K_1' \arrow[ll, "d_1^*"] \arrow[r, "d_1"] & X_0
\end{tikzcd} \tag{51}$$

Here, all arrows are monomorphisms, the squares marked $\mathbb{t}$ are pushout complements in the DPO- and FPCs in the SqPO-type rewriting case, while $M_{21} = \mathsf{PB}(I_2 \rightarrow X_1 \leftarrow O_1)$, and $N_{21} = \mathsf{PO}(I_2 \leftarrow M_{21} \rightarrow O_1)$. Since pushouts along monomorphisms are pullbacks in an adhesive category, existence of the morphisms $(I_2 \rightarrow N_{21})$ and $(I_2 \rightarrow K_1')$ (the latter by assumption) entails the existence of a morphisms $(I_2 \rightarrow \overline{K}_2)$ such that

$$(K_1' \leftarrow \overline{K_1}) \circ (\overline{K}_1 \leftarrow I_2) = (K_1' \leftarrow I_2)\,.$$

By the decomposition property of monomorphisms in an adhesive category, $(I_2 \rightarrow \overline{K}_2)$ is a monomorphism, too. Analogously, since POCs and FPCs along monomorphisms are also pullbacks, we may infer the existence of a monomorphism $(O_1 \rightarrow \overline{K}_2)$ such that

$$(K_2' \leftarrow \overline{K_2}) \circ (\overline{K}_2 \leftarrow O_1) = (K_2' \leftarrow O_1)\,.$$

The statement of the "$\Leftarrow$" direction follows by composition of the relevant monomorphisms (to obtain the monomorphisms $(I_2 \rightarrow \overline{K}_2)$ and $(O_1 \rightarrow \overline{K}_2)$) and by composing PO and FPC squares (first row with the second row) to obtain the claim. Finally, the requirement on the conditions as stated arises from the definition of rule compositions, and is necessary so that the composites of the rules in the two sequential orders can give rise to an $\mathbb{T}$-admissible match (in the sense of rules with conditions) $I_{21} \hookrightarrow X_0$. □

The latter result clarifies the precise relationship between compositions and sequential applications of rules on the one hand and the notion of sequential commutativity on the other hand: two rules in a sequential rule application are independent if and only if their underlying concurrent rule composition satisfies a certain property as described above. In other words, we obtain a sharper notion of sequential independence in the latter, compositional form, since this notion characterizes sequential commutativity for an entire class of sequential rule applications (i.e. for all $\mathbb{T}$-type sequential applications that are equivalent to applications of the $\mathbb{T}$-composite rule $r_2{}^{\mu_{21}}\!\triangleleft_{\mathbb{T}} r_1$). This permits us to provide a refinement of the notion of *switching couples* (cf. e.g. [14]) to the level of rule compositions for rules with conditions, which is the first result of this kind for both DPO- and SqPO-rewriting:

Theorem 7 (Compositional concurrent commutativity) *Let* $\mathbf{C}$ *be a category satisfying Assumption 1, let* $R_j = (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *be two rules with conditions, and let* $\mu_{21} = (I_2 \leftarrow$

$M_{21} \rightarrow O_1$) be a $\mathbb{T}$-admissible match of R_2 into R_1 (with $\mathbb{T} \in \{DPO, SqPO\}$). Then if there exist monomorphisms $(O_1 \rightarrow \overline{K}_2), (I_2 \rightarrow \overline{K}_1) \in \mathsf{mono}(\mathbf{C})$ (with notations as in 2), with

$$(N_{21} \leftarrow \overline{K}_2) \circ (\overline{K}_2 \leftarrow O_1) = (N_{21} \leftarrow O_1) \quad \wedge \quad (N_{21} \leftarrow \overline{K}_1) \circ (\overline{K}_1 \leftarrow I_2) = (N_{21} \leftarrow I_2)\,, \tag{52}$$

the following statements hold:

(i) The pullbacks

$$\begin{tikzcd} M_{21} \arrow[d] & M'_{21} \arrow[l] \arrow[d] \arrow[dl, phantom, "\mathsf{PB}"] & M_{21} \arrow[d] & M''_{21} \arrow[l] \arrow[d] \arrow[dl, phantom, "\mathsf{PB}"] \\ O_1 & K_1 \arrow[l] & I_2 & K_2 \arrow[l] \end{tikzcd} \tag{53}$$

satisfy $M'_{21} \cong M_{21}$ and $M''_{21} \cong M_{21}$, thus furnishing monomorphisms $(a_1 : M_{21} \rightarrow K_1)$ and $(a_2 : M_{21} \rightarrow K_2)$.

(ii) The span $\mu_{12} := (I_1 \xleftarrow{i_1 \circ a_1} M_{21} \xrightarrow{o_2 \circ a_2} O_2)$ is a DPO- (and thus SqPO-) admissible match of r_1 into r_2.

(iii) r_1 and r_2 are sequentially independent w.r.t. μ_{12} in both types of rewriting, and with

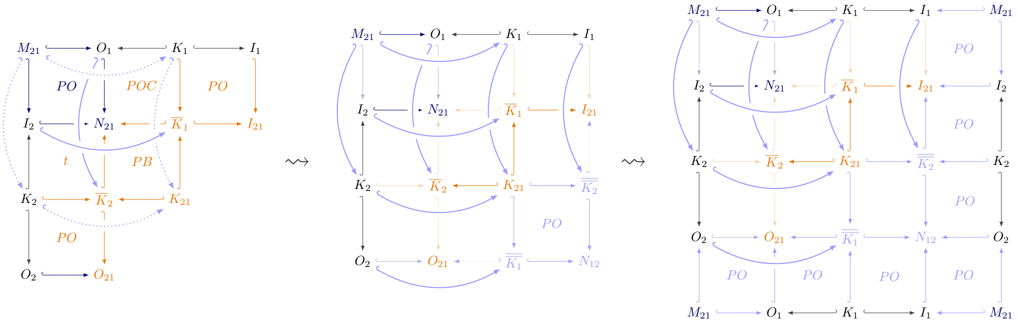
$$(r_2{}^{\mu_{21}}\!\lhd_{SqPO} r_1) \cong (r_2{}^{\mu_{21}}\!\lhd_{DPO} r_1) \cong (r_1{}^{\mu_{12}}\!\lhd_{DPO} r_2) \cong (r_1{}^{\mu_{12}}\!\lhd_{SqPO} r_2)\,. \tag{54}$$

(iv) R_1 and R_2 are sequentially independent only if in addition $\mathsf{c}_{I_{21}} \equiv \mathsf{c}_{I_{12}}$.

Proof: Let K_{21} denote the pullback of $\overline{K}_2 \hookrightarrow N_{21} \hookleftarrow \overline{K}_1$, and let the square marked t in (55) below be a pushout complement (POC) for $\mathbb{T} = DPO$, and a final pullback complement (FPC) for $\mathbb{T} = SqPO$, respectively.

Ad (i) and (ii): By assumption, there exist monomorphisms $O_1 \hookrightarrow \overline{K}_2$ and $I_2 \hookrightarrow \overline{K}_1$. This entails by virtue of the universal property of pullbacks the following structures (compare (55)):

- Commutativity of the diagram and existence of the monomorphisms $\overline{K}_1 \hookleftarrow K_1$ and $(\overline{K}_2 \hookleftarrow O_1) \circ (O_1 \hookleftarrow K_1)$ entail by the universal property of the pullback $\square(K_{21}, \overline{K}_1, N_{21}, \overline{K}_2)$ the existence of a morphism $K_{21} \leftarrow K_1$. Since $(\overline{K}_1 \hookleftarrow K_1) = (\overline{K}_{21} \hookleftarrow K_{21}) \circ (K_{21} \leftarrow K_1)$, with $\hookleftarrow$ indicating monomorphisms, by stability of monomorphisms under decompositions we conclude that $(K_{21} \leftarrow K_1)$ is a monomorphism. In an analogous fashion, the existence of monomorphisms $K_2 \hookrightarrow \overline{K}_2$ and $M_{21} \hookrightarrow K_i$ (for $i = 1, 2$). This proves part (i).
- Invoking *pushout-pullback decomposition* (cf. e.g. [16]) repeatedly (noting that POCs and FPCs along monomorphisms are also pullbacks), we may conclude that the squares marked t and PB in (55) are pushouts, as are all squares formed involving the monomorphisms previously discussed to exist (i.e. the morphisms marked in light blue in (55)).



(55)

From hereon, we may follow the classical strategy for proving sequential commutativity in the DPO-type setting (cf. e.g. [36], proof of Thm. 5.12): first, we form the pushouts $\overline{\overline{K}}_2 = \mathsf{PO}(K_{21} \hookleftarrow K_1 \hookrightarrow I_1)$, $\overline{\overline{K}}_1 := \mathsf{PO}(O_2 \hookleftarrow K_2 \hookrightarrow K_{21})$ and $N_{12} := \mathsf{PO}(\overline{\overline{K}}_1 \hookleftarrow K_{21} \hookrightarrow \overline{\overline{K}}_2)$, which by universal properties of pushouts and stability of monomorphisms under decompositions leads to the existence of monomorphisms $\overline{\overline{K}}_2 \hookrightarrow I_{21}$ and $\overline{\overline{K}}_1 \hookrightarrow O_{21}$. Since moreover by virtue of *pushout-pushout decomposition* the newly formed squares involving the two aforementioned monomorphisms are found to be pushouts, we finally obtain a DPO-type composition of r_1 with r_2 along the span μ_{21} by assembling pushout squares as depicted in the last step of (55). This identifies the span $\mu_{12} := (I_1 \hookleftarrow M_{21} \hookrightarrow O_2)$ as a DPO- (and thus SqPO-) admissible match of r_1 into r_2, which proves part (ii).

Ad (iii): Since we have found in the proof of part (i) that the square marked t in (55) is a pushout whenever the monomorphisms $O_1 \hookrightarrow \overline{K}_2$ and $I_2 \hookrightarrow \overline{K}_1$ exist, note first that sequential compositions of rules r_2 and r_1 along a $\mathbb{T}$-admissible match μ_{21} that are sequentially independent are in fact always DPO-type compositions (which for $\mathbb{T} = SqPO$ is indeed a possible special case, since a pushout complement is also an FPC). Together with the construction of the DPO-type composition of r_1 and r_2 along the uniquely induced span μ_{12} as presented in the proof of part (ii), which in particular entails the existence of monomorphisms $O_2 \hookrightarrow \overline{K}_1$ and $I_1 \hookrightarrow \overline{K}_2$, this provides the proof of part (iii).

Ad (iv): The final claim follows by verifying the well-known fact that there is no guarantee for the conditions $\mathsf{c}_{I_{21}}$ and $\mathsf{c}_{I_{12}}$ of the two composites to coincide, thus concluding the proof. □

Note that the above statements have the peculiar consequence that two sequentially independent rules r_2 and r_1 give rise to a so-called *amalgamated rule* [22], in the sense that

$$\begin{aligned} O_{21} &= \mathsf{PO}(O_2 \hookleftarrow M_{21} \hookrightarrow O_1) \\ K_{21} &= \mathsf{PO}(K_2 \hookleftarrow M_{21} \hookrightarrow K_1) \\ I_{21} &= \mathsf{PO}(I_2 \hookleftarrow M_{21} \hookrightarrow I_1)\,. \end{aligned} \tag{56}$$

Since the theory of amalgamation has been extensively developed in the graph rewriting literature [22, 38, 41, 43], it might well be the case that the above result may be beneficial in the concrete implementations of tracelet analysis algorithms.

B.1 A worked example of tracelet shift equivalence

In order to provide some intuitions for the notion of tracelet shift equivalence, we present in Figure 4 a concrete example of two tracelets of length 3 that are shift equivalent. The bottom half of the diagram coincides with the bottom half of Figure 1b, while the top half of Figure 4 encodes a tracelet where the order of applications of the second and third rules has in effect been reversed. Note in particular that while the rules involved are understood as rewriting rules for finite directed (unlabeled) multigraphs, we have employed vertex symbols and edge colors in order to encode the structure of the various monomorphisms and partial maps (i.e. repeated symbols encode elements related by partial maps). We have moreover chosen representatives for the two tracelets such that the isomorphisms that relate the tracelets are concretely implemented by isomorphisms of the underlying rewriting rules. An essential feature of our definition of shift equivalence (Definition 5) is the following technical detail: for tracelets $T = t_n|\dots|t_1$ and $T' = t'_n|\dots|t'_1$ that are shift equivalent based upon subtracelets $t_j|\dots|t_{j-k}$ and $t'_j|\dots|t'_{j-k}$, part of the definition entails that we demand the existence of an isomorphism between $t_n|\dots|t_{(j|\dots|j-k)}|\dots|t_1$ and $t'_n|\dots|t'_{(j|\dots|j-k)}|\dots|t'_1$. However, we do *not* demand an isomorphism between the original tracelets T and T', which would only exist in the special case where the subtracelets encode sequentially independent derivations in the traditional sense. This feature is illustrated explicitly in Figure 4, where the minimal derivation traces encoded by the two tracelets of length 3 are in fact *not* in isomorphism (due to the non-existence of an isomorphism of the "X-shaped" respective third objects in the minimal traces that would be compatible with the morphism structure of the traces), but only the minimal derivation traces of the tracelets of length 2 given by $t_{(3|2)}|t_1$ and $t'_{(3|2)}|t'_1$, respectively. Here, the tracelet $T'_{(3|2)}$ of length 2 that leads to $t'_{(3|2)}$ is depicted in the light blue box, while $T_{(3|2)}$ leading to $t_{(3|2)}$ is depicted in the light yellow box. It is this particular feature that deserves to refer to the process of abstracting tracelets by means of tracelet shift equivalence as a form of *strong compression* in the sense of [35].

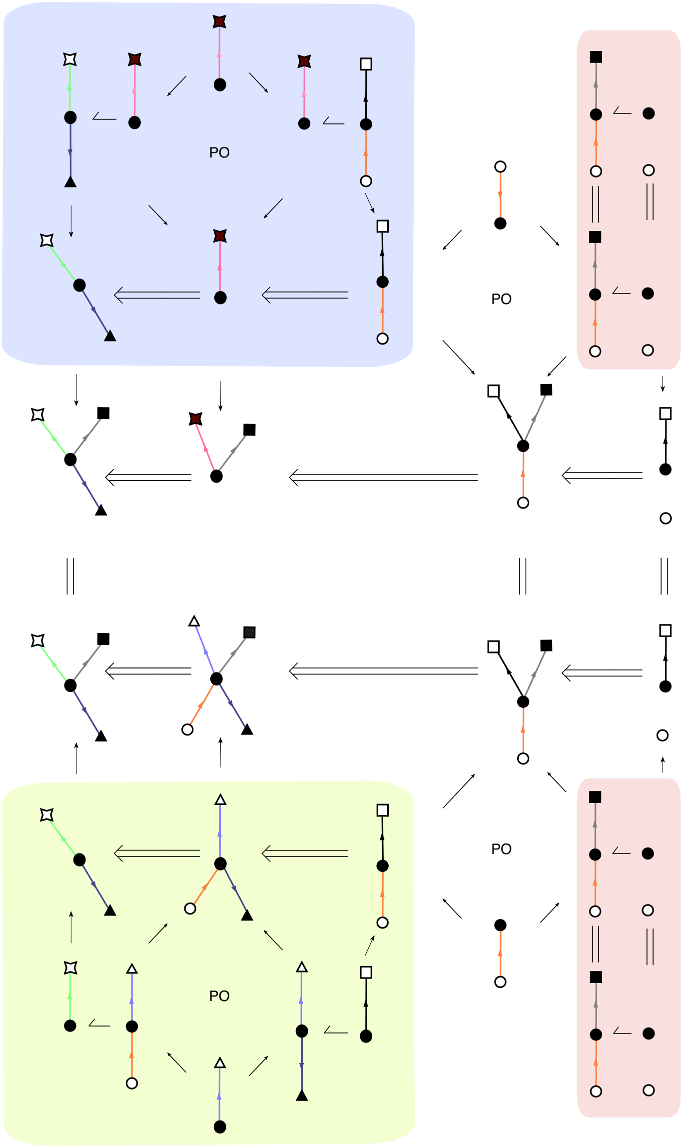


Figure 4: Illustration of a concrete example of tracelet shift equivalence, based upon the tracelet of length 3 as presented in Figure 1. The bottom half of the diagram is identical to the bottom half of Figure 1b, while the top half illustrates a shift-equivalent tracelet of length 3 in which the order of the second and third rule applications has been swapped.