TERMINATION OF GRAPH TRANSFORMATION SYSTEMS USING WEIGHTED SUBGRAPH COUNTING [*]

ROY OVERBEEK ⬤ AND JÖRG ENDRULLIS ⬤

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam
e-mail address: r.overbeek@vu.nl, j.endrullis@vu.nl

ABSTRACT. We introduce a termination method for the algebraic graph transformation framework PBPO$^+$, in which we weigh objects by summing a class of weighted morphisms targeting them. The method is well-defined in rm-adhesive quasitoposes (which include toposes and therefore many graph categories of interest), and is applicable to non-linear rules. The method is also defined for other frameworks, including SqPO and left-linear DPO, because we have previously shown that they are naturally encodable into PBPO$^+$ in the quasitopos setting. We have implemented our method, and the implementation includes a REPL that can be used for guiding relative termination proofs.

1. INTRODUCTION

Many fields of study related to computation have mature termination theories. See, for example, the corpus for (first-order) term rewriting systems [Ter03, Chapter 6] (for a more recent (but less systematic) discussion, see, e.g., [Yam22]).

For the study of graph transformation, by contrast, not many termination methods exist, and the ones that do exist are usually defined for rather specific notions of graphs. Although the techniques themselves can be interesting, the latter observation fits somewhat uneasily with the general philosophy of the algebraic graph transformation tradition, in which graph transformations are defined and studied in a graph-agnostic manner.

In this chapter, we introduce a termination method for PBPO$^+$. We weigh objects G by summing a class of weighted elements (i.e., morphisms of the form $T \to G$), and construct a decreasing measure. Our method enjoys generality across two dimensions:

(1) The method is formulated completely in categorical terms, and is well-defined in (locally finite) rm-adhesive quasitoposes.
(2) The method is also defined for SqPO [CHHK06], AGREE [CDE$^+$20], PBPO [CDE$^+$19], and left-linear DPO [EPS73]. This is because we have recently shown that, in the quasitopos setting, each rule of these formalisms can be straightforwardly encoded as a PBPO$^+$ rule that generates the same rewrite relation [OER23, Theorem 73].

Key words and phrases: graph transformation, termination, pullback pushout.

[*] This paper extends and improves the identically titled paper [OE23b] published at ICGT2023.

To the best of our knowledge, this is the first termination method applicable in such a broad setting; and the first method that is automatically defined for a variety of well-known algebraic graph transformation frameworks. In addition, the termination method can be applied to non-linear (duplicating) rules.

This paper is structured as follows. We summarize some basic categorical and termination preliminaries (Section 2). We then provide some useful background on quasitoposes, and required background on PBPO$^+$ (Section 3). Next, we explain and prove our termination method (Section 4). After, we illustrate our method with a variety of examples (Section 5), and compare our approach to related work (Section 6). We then present an implementation of our method, which includes a REPL (read-eval print loop) that can be used for guiding relative termination proofs (Section 7). We close with some concluding remarks and pointers for future work (Section 8).

Remark 1.1. This paper extends and improves the identically titled conference paper [OE23b]. The most noteworthy differences are as follows:

(1) All previously elided proofs, Example 5.8, and Example 5.9 are now included. Previously they were available on arXiv [OE23c] only due to space restrictions.
(2) The background survey on quasitoposes (Section 3.1) is newly included, in order to make the paper more self-contained, and to improve the motivation.
(3) Instead of requiring the existence of (essentially unique) $\mathcal{A}$-factorizations [OE23b, Definition 9], we use a simpler and weaker condition (Definition 4.16), which we then relate to standard factorization systems (Proposition 4.18).
(4) Section 7 contains completely new material (and the implementation described therein is also new).

The new additions include previously unpublished material taken from the PhD thesis of the first author [Ove24].

2. Preliminaries

The preliminaries for this paper include basic categorical and graph notions (Section 2.1), and a basic understanding of termination (Section 2.2).

2.1. **Basic Notions.** We assume familiarity with basic categorical notions such as (regular) monomorphisms, pullbacks and pushouts [BW90, Pie91]. We write $\rightarrowtail$ for monos; and $\mathrm{Hom}(\mathbf{C})$, $\mathrm{mono}(\mathbf{C})$, $\mathrm{rm}(\mathbf{C})$ and $\mathrm{iso}(\mathbf{C})$ for the classes of morphisms, monomorphisms, regular monomorphisms and isomorphisms in $\mathbf{C}$, respectively.

Notation 2.1 (Nonstandard Notation). Given a class of morphisms $\mathcal{A}(\mathbf{C})$, we write $\mathcal{A}(A, B)$ to denote the collection of $\mathcal{A}$-morphisms from A to B, leaving $\mathbf{C}$ implicit. For sets of objects S, we overload $\mathcal{A}(S, A)$ to denote $\bigcup_{X \in S} \mathcal{A}(X, A)$. If $\mathcal{A}(\mathbf{C})$ is a generic class in lemmas, we use $\rightsquigarrow$ to denote $\mathcal{A}$-morphisms.

For cospans $A \xrightarrow{f} C \xleftarrow{g} D$, we write $\langle f \mid g \rangle$ to denote the arrow $B \to D$ obtained by pulling f back along g.

Definition 2.2 ($\mathcal{A}$-Local Finiteness). Let $\mathcal{A}(\mathbf{C})$ be a class of morphisms. A category $\mathbf{C}$ is $\mathcal{A}$-*locally finite* if $\mathcal{A}(A, B)$ is finite for all $A, B \in \mathrm{Ob}(\mathbf{C})$.

Lemma 2.3 (Pullback Lemma [Bor94a, Proposition 2.5.9]). *Assume the right square of*

$$\begin{array}{ccc} A \longrightarrow B \longrightarrow C \\ \downarrow \quad\quad \downarrow \text{ PB } \downarrow \\ D \longrightarrow E \longrightarrow F \end{array}$$

is a pullback and the left square commutes. Then the outer rectangle (obtained by composing the horizontal morphisms) is a pullback iff the left square is a pullback. □

Definition 2.4 (Split Epimorphism)**.** An epimorphism $e : A \twoheadrightarrow B$ is *split* if it has a right inverse, i.e., if there exists an $f : B \to A$ such that $e \circ f = 1_B$.

For split epimorphisms e, we let $e^{\leftarrow}$ denote an arbitrary right inverse of e.

Observe that right inverses need not be unique. In **Set**, for instance, the constant function $k : \{0,1\} \to \{0\}$ has two right inverses.

Proposition 2.5 [AHS06, Prop. 7.59]**.** *If e is a split epi, then $e^{\leftarrow} \in \mathrm{rm}(\mathbf{C})$.* □

Our method is defined fully in categorical terms. For examples, and to guide intuition, we will use the category of edge-labeled multigraphs.

Definition 2.6 (Graph Notions)**.** Let a finite label set $\mathcal{L}$ be fixed. An (edge-labeled) *(multi)graph* G consists of a set of vertices V, a set of edges E, source and target functions $s, t : E \to V$, and an edge label function $\ell^E : E \to \mathcal{L}$. A graph is *unlabeled* if $\mathcal{L}$ is a singleton.

A *homomorphism* between graphs G and G' is a pair of maps $\phi = (\phi_V : V_G \to V_{G'}, \phi_E : E_G \to E_{G'})$ satisfying $(s_{G'}, t_{G'}) \circ \phi_E = \phi_V \circ (s_G, t_G)$ and $\ell^E_{G'} \circ \phi_E = \ell^E_G$.

Definition 2.7 [EEPT06]**.** The category **Graph** has graphs as objects, parameterized over some global (and usually implicit) label set $\mathcal{L}$, and homomorphisms as arrows. The subcategory **FinGraph** restricts to graphs with finite V and E.

Although our termination method is defined for PBPO$^+$ (Section 3.2), we will state a key result (Theorem 3.14) that involves (left-linear) DPO.

Definition 2.8 (DPO Rewriting [EPS73])**.** In its most general formulation, a *DPO rewrite rule* ρ is a span $L \xleftarrow{l} K \xrightarrow{r} R$, and a diagram

$$\begin{array}{ccccc} L & \xleftarrow{\;l\;} & K & \xrightarrow{\;r\;} & R \\ \Big\downarrow{\scriptstyle m} & \text{PO} & \Big\downarrow & \text{PO} & \Big\downarrow \\ G_L & \longleftarrow & G_K & \longrightarrow & G_R \end{array}$$

defines a *DPO rewrite step* $G_L \Rightarrow^{\rho,m}_{\mathrm{DPO}} G_R$, i.e., a step from G_L to G_R using rule ρ and match $m : L \to G_L$.

In some versions of DPO, morphism l is required to belong to some class of monomorphisms $\mathcal{M}$, to ensure that pushout complements exist uniquely. We will refer to this restricted version of DPO as *left-linear DPO*, and denote it by DPO$^{\mathcal{M}}$. Additionally, morphism m is often required to be in $\mathcal{M}$ because it increases expressivity [HMP01].

2.2. **Termination.** The topic of termination dates back at least to Turing, and is studied in many different settings. For a systematic overview for term rewriting systems (not yet existent for graph transformation systems), see [Ter03, Chapter 6] (for a more recent discussion, see, e.g., [Yam22]). Plump has shown that termination of graph rewriting is undecidable [Plu98].

Definition 2.9. Define $I_n = \{m \in \mathbb{N} \mid m < n\}$ for $n \in \mathbb{N} \cup \{\omega\}$, where ω extends the natural ordering of $\mathbb{N}$ with $k < \omega$ for all $k \in \mathbb{N}$.

Let $R \subseteq A \times A$ be given. An *R-sequence (of length n)* is a function $f : I_n \to A$ such that for all $i < n$, $(f(i), f(i+1)) \in R$. The pair $(f(i), f(i+1))$ is said to be the $(i+1)$-*th step* in this sequence. The sequence is *infinite* if $n = \omega$.

Definition 2.10. A binary relation R is *terminating* if there does not exist an infinite R-sequence.

Definition 2.11 [BD86,Klo87]. Let $R, S \subseteq A \times A$ be binary relations. Then R is *terminating relative to S* if every infinite $R \cup S$-sequence contains a finite number of R steps.

For our purposes, it suffices to measure objects as natural numbers (instead of a general well-founded order).

Definition 2.12 (Measure). A *measure* is a function $\mathbf{wt} : A \to \mathbb{N}$. The measure $\mathbf{wt}$ is *decreasing* for a binary relation $R \subseteq A \times A$ if for all $(x, y) \in R$, $\mathbf{wt}(x) > \mathbf{wt}(y)$, and it is *non-increasing* for R if for all $(x, y) \in R$, $\mathbf{wt}(x) \geq \mathbf{wt}(y)$.

Proposition 2.13 [BD86, Klo87]. *Let $R, S \subseteq A \times A$ be binary relations. Assume that there exists a measure $\mathbf{wt}$ that is decreasing for R and non-increasing for S. Then R is terminating relative to S. Consequently $R \cup S$ is terminating iff S is.* □

In a framework agnostic setting, a rule ρ is a mathematical object that induces a binary relation $\Rightarrow_\rho \subseteq A \times A$. We say that a rule (or a system of rules) is terminating, decreasing or non-increasing if the induced rewrite relations have the respective property (and analogously for relative termination). Note that also Proposition 2.13 can then be applied to systems of rules in place of relations.

3. Background

3.1. **Quasitoposes.** Our termination method makes a number of assumptions about the underlying category and about certain morphism classes. One natural and relevant setting in which these assumptions are satisfied is an rm-adhesive quasitopos. For motivational reasons, we therefore provide a brief background on quasitoposes.

Definition 3.1 (Quasitopos [Wyl91, AHS06, Joh02]). A category $\mathbf{C}$ is a *quasitopos* if it has all finite limits and colimits, it is locally cartesian closed, and it has a regular-subobject classifier.

The definition of a quasitopos itself need not be understood: we are interested in specific properties of quasitoposes, and in examples. Quasitoposes can also be understood in terms of the more familiar toposes.

Proposition 3.2 ([Wyl91, Proposition 19.6], [AHS06, Proposition 28.6(2)]). $\mathbf{C}$ *is a topos iff* $\mathbf{C}$ *is a quasitopos and* $\mathrm{rm}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. □

In the graph transformation literature there is a variety of adhesivity properties, which depend on the notion of a Van Kampen square.

Definition 3.3 (Van Kampen Square [LS04]). A pushout square is *Van Kampen (VK)* if, whenever it lies at the bottom of a commutative cube

$$\begin{array}{ccc}
F & \longrightarrow & G \\
& & \\
E & \longrightarrow & H \\
& & \\
& B & \longrightarrow C \\
& & \\
A & \longrightarrow & D
\end{array}$$
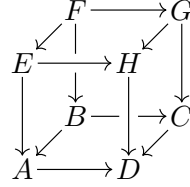
of which the back faces $FBAE$ and $FBCG$ are pullbacks, this implies that

the top face is a pushout $\iff$ the front faces $EHDA$ and $GHDC$ are pullbacks.

Definition 3.4 (Rm-Adhesive Category [LS05]). A category is *rm-adhesive* (a.k.a. *quasiadhesive*) if pushouts along regular monomorphisms exist and are VK.

Figure 1 situates quasitoposes among toposes and various adhesivity properties. Observe in particular that not all quasitoposes are rm-adhesive. A counterexample is found in the category of simple graphs [JLS07, Corollary 20].

The following properties of quasitoposes are relevant for this paper (either as requirements for the termination method or for exposition). Extended (and overlapping) summaries relevant for graph transformation are provided by Behr et al. [BHK23, Corollary 5.15] and by the authors [OER23, Proposition 36].

Proposition 3.5 (Relevant Quasitopos Properties). *A quasitopos* **C**

(1) *has (by definition) all pushouts and pullbacks;*
(2) *satisfies*
   (a) $g, f \in \mathrm{rm}(\mathbf{C}) \implies g \circ f \in \mathrm{rm}(\mathbf{C})$; *and*
   (b) $g \circ f \in \mathrm{rm}(\mathbf{C}) \implies f \in \mathrm{rm}(\mathbf{C})$;
   for all morphisms $g : B \to C$ and $f : A \to B$ [AHS06, Proposition 7.62(2), Corollary 28.6(2-3)];
(3) *pushouts along regular monomorphisms are pullbacks [Joh02, Lemma A2.6.2];*
(4) *regular monomorphisms are stable under pushout [Joh02, Lemma A.2.6.2];*
(5) *has essentially unique (epi, regular mono) factorizations for every morphism [AHS06, Proposition 28.10], i.e., every morphism $f : A \to B$ factors uniquely as $A \xrightarrow{e} C \xhookrightarrow{m} B$ (up to isomorphism in C), where e is epic and m regular monic;*
(6) *has essentially unique (regular epi, mono) factorizations for every morphism [JLS07, Proposition 11], i.e., every morphism $f : A \to B$ factors uniquely as $A \xrightarrow{e} C \xrightarrow{m} B$ (up to isomorphism in C), where e is regular epic and m is monic; and*
(7) *is stable under slicing, i.e., every slice category $\mathbf{C}/X$ (for $X \in \mathrm{Ob}(\mathbf{C})$) is a quasitopos [Wyl91, Theorem 19.4].*

Many structures of interest are (quasi)toposes. We give some motivating examples of rm-adhesive quasitoposes.

Example 3.6 ((Co)presheaf Toposes). If **C** is small, then the functor category $[\mathbf{C}, \mathbf{Set}]$ is a topos [Wyl91, Theorem 26.2], and hence an rm-adhesive quasitopos. Such categories are known as copresheaf toposes, **C**-sets [BPHF23], or graph structures [Löw93] (and $[\mathbf{C}^{\mathrm{op}}, \mathbf{Set}]$ is known as a presheaf category). Many structures that are of interest to the graph transformation community can be defined in this manner. For example, if the label set $\mathcal{L}$ is a singleton (i.e., graphs are unlabeled), then $\mathbf{Graph} \cong [\cdot \rightrightarrows \cdot, \mathbf{Set}])$.

The following proposition assures us that such toposes are closed under finite restrictions. We are not aware of a similar principle for quasitoposes.
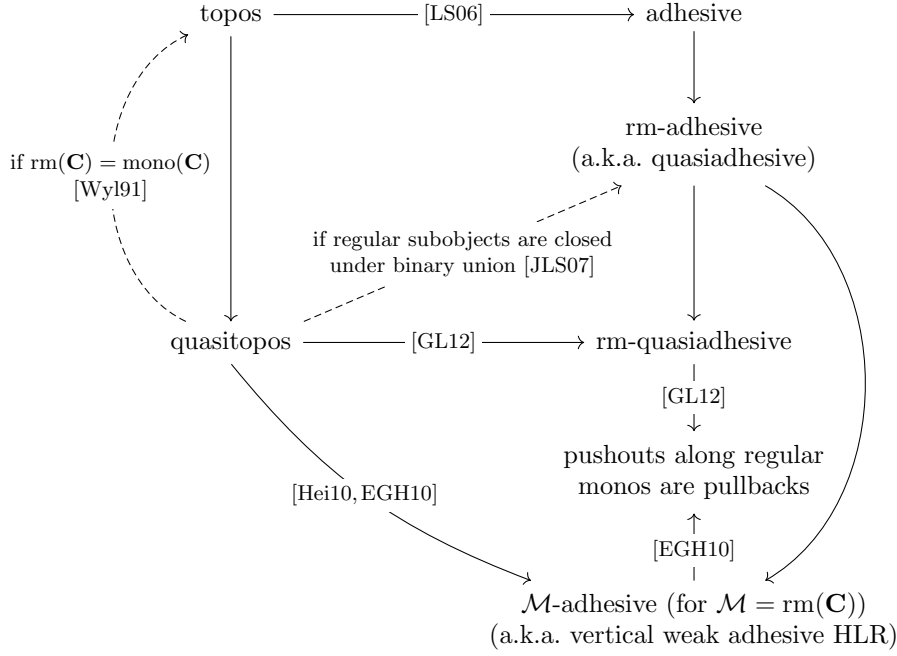
FIGURE 1. Implications between (quasi)toposes and adhesivity properties.

Proposition 3.7. *If* **I** *is finite and* $\mathbf{C} \cong [\mathbf{I}, \mathbf{FinSet}]$, *then* **C** *is a* Hom-*locally finite topos, and so an rm-adhesive quasitopos that is* $\mathcal{A}$-*locally finite for any* $\mathcal{A}(\mathbf{C})$.

Proof. **C** is a topos [Bor94b, Example 5.2.7], and it is locally finite because **FinSet** is Hom-locally finite. And any topos is an rm-adhesive quasitopos (Figure 1). □

Example 3.8 (Typed Graphs). Given the category of unlabeled graphs **Graph**, a category of graphs typed over $X \in \mathrm{Ob}(\mathbf{Graph})$ can be constructed through a slice construction **Graph**$/X$, which yields a quasitopos (Proposition 3.5). Typed graphs are subsumed by **C**-sets [BPHF23, Section 2.3].

The following is an example of an rm-adhesive quasitopos that is not a topos.

Example 3.9 (Fuzzy Presheaves). If $(\mathcal{L}, \leq)$ is a complete Heyting algebra, then the category of $(\mathcal{L}, \leq)$-fuzzy presheaves is an rm-adhesive quasitopos [ROE23, Theorems 15 and 41]. This includes the category of fuzzy graphs over a complete Heyting algebra $(\mathcal{L}, \leq)$ [OER23, Definition 76]. We have proposed fuzzy graphs as a mechanism for relabeling labeled graphs [OER23, Section 6]. In such a category, mono(**C**) contains all injective graph homomorphisms such that labels are non-decreasing ($\leq$) w.r.t. the given order, and rm(**C**) contains only the injective graph homomorphisms that preserve labels ($=$).

3.2. **PBPO$^+$.** PBPO$^+$ is short for *Pullback-Pushout with strong matching*. It is obtained by strengthening the matching mechanism of PBPO [CDE$^+$19] by Corradini et al.

We provide the necessary definitions and results on PBPO$^+$. See Section 5 for many examples of rules. For a gentler introduction to PBPO$^+$, with examples of rewrite steps, see the tutorial [OE23a] or the PhD thesis of the first author [Ove24].

Definition 3.10 (PBPO$^+$ Rewriting [CDE$^+$19, OER23]). A *PBPO$^+$ rule* ρ is a diagram as shown on the left of:

$$
\begin{array}{ccccc}
L & \xleftarrow{\; l \;} & K & \xrightarrow{\; r \;} & R \\
{\scriptstyle t_L}\downarrow & \text{PB} & {\scriptstyle t_K}\downarrow & & \\
L' & \xleftarrow{\; l' \;} & K' & &
\end{array}
\qquad\qquad
\begin{array}{ccccccc}
 & & & K & \xrightarrow{\; r \;} & R & \\
 & & & {\scriptstyle !u}\downarrow & \text{PO} & {\scriptstyle w}\downarrow & \\
L & \xrightarrow{\; m \;} & G_L \xleftarrow{\; g_L \;} & G_K & \xdashrightarrow{\; g_R \;} & G_R & \\
\| & \text{PB} & {\scriptstyle \alpha}\downarrow \quad \text{PB} & {\scriptstyle u'}\downarrow\; {\scriptstyle t_K} & & & \\
L & \xrightarrow{\; t_L \;} & L' \xleftarrow{\; l' \;} & K' & & &
\end{array}
$$

L is the *lhs pattern* of the rule, L' its *(context) type* and t_L the *(context) typing* of L. Likewise for the *interface* K. R is the *rhs pattern* or *replacement for L*.

Rule ρ, *match morphism* $m : L \to G_L$ and *adherence morphism* $\alpha : G_L \to L'$ induce a *rewrite step* $G_L \Rightarrow_\rho^\alpha G_R$ on arbitrary objects G_L and G_R if the properties indicated by the commuting diagram on the right hold, where $u : K \to G_K$ is the unique morphism satisfying $t_K = u' \circ u$ [OER23, Lemma 15].

Lemma 7.2 suggests how to compute a rewrite step: a search for u is in fact not required.

Assumption 3.11. We assume that for any rule ρ analyzed for termination, the pushout of t_K along r exists.

Remark 3.12. If Assumption 3.11 holds for a rule ρ, then any ρ step defines (up to isomorphism) a diagram

$$
\begin{array}{ccccccc}
\mathbf{L} & \xleftarrow{\; \iota \;} & \mathbf{K} & \xrightarrow{\; r \;} & \mathbf{R} & & \\
{\scriptstyle m}\downarrow & \text{PB} & {\scriptstyle u}\downarrow & \text{PO} & {\scriptstyle w}\downarrow & & \\
G_L \xleftarrow{} & \!\!-\, g_L \, & G_K & \!\!-\, g_R \to & G_R & & \\
{\scriptstyle \alpha}\downarrow\; {\scriptstyle t_L} & \text{PB} & {\scriptstyle u'}\downarrow\; {\scriptstyle t_K} & \text{PO} & {\scriptstyle w'}\downarrow\; {\scriptstyle t_R} & & \\
\mathbf{L'} & \xleftarrow{\; l' \;} & \mathbf{K'} & \xrightarrow{\; r' \;} & \mathbf{R'} & &
\end{array}
$$

where the bold diagram is ρ ($t_L \circ l = l' \circ t_K$ a pullback and $t_R \circ r = r' \circ t_K$ a pushout) [OER23, Section 3]. Our method uses the extra pushout to analyze how rewritten objects (the middle span) relate to the context types (the bottom span).

Remark 3.13. This assumption is not restrictive. All colimits (and thus all pushouts) exist in quasitoposes, our main setting. Moreover, pushouts are guaranteed to exist along $\mathcal{M}$-morphisms in $\mathcal{M}$-adhesive categories, and morphism t_K is usually in $\mathcal{M}$.

Theorem 3.14 [OER23, Theorem 73]. *Let $\mathbf{C}$ be a quasitopos, and let matches $m \in \mathrm{rm}(\mathbf{C})$. For rewriting formalisms $\mathcal{F}$ and $\mathcal{G}$, let $\mathcal{F} \prec \mathcal{G}$ express that in $\mathbf{C}$, for any $\mathcal{F}$ rule ρ, there exists a $\mathcal{G}$ rule τ such that $\Rightarrow_\mathcal{F}^\rho = \Rightarrow_\mathcal{G}^\tau$. We have:*

$$
SqPO \prec AGREE \overset{\succ}{} \begin{array}{c} PBPO^+ \\ \curlyvee \\ PBPO \end{array} \overset{\prec}{} DPO^{\mathrm{rm}(\mathbf{C})} \qquad\qquad \square
$$

Observe that $\prec$ is transitive. As the constructive proofs in [OER23] show, the procedures to encode the mentioned formalisms into PBPO$^+$ are straightforward.[1] We moreover conjecture SPO $\prec$ PBPO$^+$ [OER23, Remark 26].

---

[1] More precise statements and proofs can be found in the first author's PhD thesis [Ove24, Section 4.5].

4. Decreasingness by Counting Weighted Elements

We start with an explanation of the general idea behind our termination approach. Given a set of rules $\mathcal{T}$, we seek to construct a measure $\mathbf{wt}$ such that for all PBPO$^+$ steps $G_L \Rightarrow_\rho G_R$ generated by a rule $\rho \in \mathcal{T}$, $\mathbf{wt}(G_L) > \mathbf{wt}(G_R)$. Then $\mathbf{wt}$ is a decreasing measure for the rewrite relation generated by $\mathcal{T}$, such that $\mathcal{T}$ is terminating. We construct such a measure $\mathbf{wt}$ by weighing objects as follows.

Definition 4.1 (Weight Functions). Given a set of objects $\mathbb{T}$, *weight function* $\mathbf{wt} : \mathbb{T} \to \mathbb{N}$, and class of morphisms $\mathcal{A}(\mathbf{C})$, we define the *tiling weight function*

$$\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(X) \qquad = \qquad \sum_{t \in \mathcal{A}(\mathbb{T}, X)} \mathbf{wt}(dom(t))$$

for objects $X \in \mathrm{Ob}(\mathbf{C})$. In this context, we refer to the objects of $\mathbb{T}$ as *tiles*.

Assumption 4.2. We assume that $\mathbb{T}$ is finite and $\mathbf{C}$ is $\mathcal{A}$-locally finite, such that $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}$ is well defined.

Example 4.3. Let $\mathbf{C} = \mathbf{FinGraph}$ with singleton label set $\mathcal{L}$, and G an arbitrary graph. Some basic examples of tile sets and parameters are as follows.

- Let $\cdot$ represent the graph consisting of a single node. If $\mathbb{T} = \{\cdot\}$, $\mathbf{wt}(\cdot) = 1$, and $\mathcal{A}(\mathbf{C}) \in \{\mathrm{Hom}(\mathbf{C}), \mathrm{mono}(\mathbf{C}), \mathrm{rm}(\mathbf{C})\}$, then $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G) = |V_G|$.
- Let $\cdot \to \cdot$ represent the graph consisting of a single edge with distinct endpoints. If $\mathbb{T} = \{\cdot \to \cdot\}$, $\mathbf{wt}(\cdot \to \cdot) = 1$ and $\mathcal{A}(\mathbf{C}) = \mathrm{Hom}(\mathbf{C})$, then $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G) = |E_G|$. If instead $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$, then $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G)$ counts the number of subgraph occurrences isomorphic to $\cdot \to \cdot$ in G (loops are not counted). (See also Example 5.3.)
- If $\mathbb{T} = \{\cdot, \cdot \to \cdot\}$, $\mathbf{wt}(\cdot) = 2$, $\mathbf{wt}(\cdot \to \cdot) = 1$ and $\mathcal{A}(\mathbf{C}) = \mathrm{Hom}(\mathbf{C})$, then $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G) = 2 \cdot |V_G| + |E_G|$.

Our goal is to use $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(\cdot)$ as a decreasing measure. This gives rise to two main challenges: finding a suitable $\mathbb{T}$ (if it exists), and determining whether $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(\cdot)$ is decreasing. In this paper, we focus exclusively on the second problem, and show that the matter can be decided through a finite rule analysis.

Certain assumptions on $\mathcal{A}(\mathbf{C})$ will be needed. To prevent clutter and to help intuition, we state them now, valid for the remainder of this paper. In the individual proofs, we clarify which assumptions on $\mathcal{A}(\mathbf{C})$ are used.

Assumption 4.4. We assume the following about $\mathcal{A}(\mathbf{C})$:

- $\mathrm{rm}(\mathbf{C}) \subseteq \mathcal{A}(\mathbf{C})$; and
- $\mathcal{A}(\mathbf{C})$ is stable under:
  - pullback;
  - composition ($g, f \in \mathcal{A}(\mathbf{C}) \implies g \circ f \in \mathcal{A}(\mathbf{C})$); and
  - decomposition ($g \circ f \in \mathcal{A}(\mathbf{C}) \implies f \in \mathcal{A}(\mathbf{C})$).[2]

Note that $\mathrm{iso}(\mathbf{C}) \subseteq \mathcal{A}(\mathbf{C})$, because $\mathrm{iso}(\mathbf{C}) \subseteq \mathrm{rm}(\mathbf{C})$ (see, e.g., [Ove24, Proposition 4.29]).

---

[2]More precisely, the notion used here is stability under decomposition for a right class $\mathcal{M}$ of morphisms of some factorization system $(\mathcal{E}, \mathcal{M})$. Stability under decomposition for the left class would mean $g \circ f \in \mathcal{E} \implies g \in \mathcal{E}$. This would hold, for instance, for $\mathcal{E}$ the class of epimorphisms. We will only use stability under decomposition in the sense defined here, and so we do not make the distinction in the definition.

Proposition 4.5. *In any category, the classes* $\mathrm{Hom}(\mathbf{C})$ *and* $\mathrm{mono}(\mathbf{C})$ *satisfy Assumption 4.4. Likewise for* $\mathrm{rm}(\mathbf{C})$ *if* $\mathbf{C}$ *is a quasitopos.*

Proof. The statement is vacuously true in any category for $\mathrm{Hom}(\mathbf{C})$.

The properties for $\mathrm{mono}(\mathbf{C})$ are well known. Some references are [Awo10, Proposition 3.16] for $\mathrm{rm}(\mathbf{C}) \subseteq \mathrm{mono}(\mathbf{C})$, [Bor94a, Proposition 2.5.2] for pullback stability, and [Awo10, Exercise 2.8.4(a,b)] for stability under composition and decomposition.

Class $\mathrm{rm}(\mathbf{C})$ is stable under pullback in any category [Bor94a, Proposition 4.3.8(2)]. See the provided summary (Proposition 3.5) for the remaining properties if $\mathbf{C}$ is a quasitopos. $\square$

Now suppose that a rule ρ generates a rewrite step diagram. This defines a factorization $t_R = R \overset{w}{\to} G_R \overset{w'}{\to} R'$ (Remark 3.12). Any tiling of G_R can be partitioned into two using the following definition.

Definition 4.6. For arrows $f : A \to B$ and sets S of arrows with codomain B we define the partitioning $S = S^f_{\cong} \uplus S^f_{\not\cong}$ where $S^f_{\cong} = \{g \in S \mid \langle f \mid g \rangle \in \mathrm{iso}(\mathbf{C})\}$ and $S^f_{\not\cong} = \{g \in S \mid \langle f \mid g \rangle \notin \mathrm{iso}(\mathbf{C})\}$.

Intuitively,
$$\mathcal{A}(\mathbb{T}, G_R)^w_{\cong} = \{t \in \mathcal{A}(\mathbb{T}, G_R) \mid \langle w \mid t \rangle \in \mathrm{iso}(\mathbf{C})\}$$
contains all tilings that lie isomorphically in the pattern $w(R)$, and
$$\mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong} = \{t \in \mathcal{A}(\mathbb{T}, G_R) \mid \langle w \mid t \rangle \notin \mathrm{iso}(\mathbf{C})\}$$
the remaining tilings, which overlap partially or fully with the context. The remainder of this section is structured as follows.

We will start by centrally identifying some key assumptions and properties that we need in order to reason on the level of the rule (Section 4.1).

We then prove that there exists a domain-preserving bijection between $\mathcal{A}(\mathbb{T}, G_R)^w_{\cong}$ and $\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}$, allowing us to determine $\mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\cong})$ on the level of the rule (Section 4.2).

Determining $\mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong})$ on the level of the rule is in general impossible, because usually G_R can have an arbitrary size. Instead, we give precise conditions, formulated on the level of the rule, that ensure that there exists a domain-preserving injection $\xi : \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong} \rightarrowtail \mathcal{A}(\mathbb{T}, G_L)$ across the rewrite step diagram, such that $\mathbf{wt}(\xi \circ \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}) = \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong})$ (Section 4.3). Such injections often exist in the usual categories of interest, in which the context of G_R is roughly inherited from the left.

The two results are then combined as follows. If we additionally find a tiling $\Delta \subseteq \mathcal{A}(\mathbb{T}, L)$ such that for the given match $m : L \to G_L$,

(1) $m \circ \Delta \subseteq \mathcal{A}(\mathbb{T}, G_L)$;
(2) $\mathbf{wt}(m \circ \Delta) > \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\cong})$; and
(3) $(m \circ \Delta) \cap (\xi \circ \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}) = \varnothing$;

then

$$
\begin{aligned}
\mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_L) &\geq \mathbf{wt}(m \circ \Delta) + \mathbf{wt}(\xi \circ \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}) \\
&> \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\cong}) + \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}) \\
&= \mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_R)
\end{aligned}
$$

and we will have successfully proven that $\mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(\cdot)$ is a decreasing measure. This is the main result of this section (Section 4.4).

4.1. **Relating Rule and Step.** In order to reason about steps on the level of rules, the following variant of adhesivity is needed. It does not yet occur in the literature.

Definition 4.7 (PBPO$^+$-Adhesive). A pushout square $r' \circ t_K = t_R \circ r$ is *PBPO$^+$-adhesive* if, whenever it lies at the bottom of a commutative cube
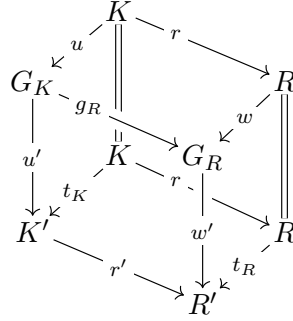
where the top face is a pushout and the back faces are pullbacks, we have that the front faces are pullbacks.

Corollary 4.8. *If* **C** *is rm-adhesive, pushouts* $r' \circ t_K = t_R \circ r$ *with* $t_K \in \mathrm{rm}(\mathbf{C})$ *are PBPO$^+$-adhesive.* $\square$

Remark 4.9. Not all quasitoposes are PBPO$^+$-adhesive: the counterexample by Johnstone et al. [JLS07, Fig. 1], which shows that the category of simple graphs is not rm-adhesive, is also a counterexample for PBPO$^+$-adhesivity. An open question is whether there are categories (of interest to the graph transformation community) that are PBPO$^+$-adhesive, but not rm-adhesive.

The following equalities will prove crucial. Recall Notation 2.1.

Lemma 4.10. *Assume* **C** *has pullbacks. Let a rewrite step for a PBPO$^+$ rule ρ be given. If square $r' \circ t_K = t_R \circ r$ is PBPO$^+$-adhesive, then for any ρ-rewrite step and any $t : T \to G_R$*

(1) $\langle g_R \mid t \rangle = \langle r' \mid w' \circ t \rangle$;
(2) $u' \circ \langle t \mid g_R \rangle = \langle w' \circ t \mid r' \rangle$;
(3) $\langle w \mid t \rangle = \langle t_R \mid w' \circ t \rangle$; *and*
(4) $\langle t \mid w \rangle = \langle w' \circ t \mid t_R \rangle$.

Proof. In the diagram on the right, the bottom face of the bottom cube is PBPO$^+$-adhesive by assumption, its top face is a pushout, and its back faces are pullbacks in any category [OER23, Lemma 15]. Hence its front faces are pullbacks by PBPO$^+$-adhesivity. Then all claims follow by composing pullback squares, using the pullback lemma. $\square$
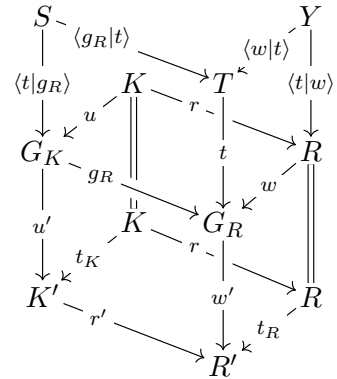
Remark 4.11. Because every $t \in \mathcal{A}(T, G_R)$ defines an arrow $w' \circ t \in \mathrm{Hom}(T, R')$, we can overapproximate $\mathcal{A}(T, G_R)$ using $\mathrm{Hom}(T, R')$. The equalities of Lemma 4.10 will then be used as follows.

(1) We will slide morphisms $t \in \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}$ to the left. If $\langle g_R \mid t \rangle$ is invertible, then $g_L \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow} : T \to G_L$ is an arrow towards the left. Lemma 4.10.1 implies that invertibility of $\langle g_R \mid t \rangle$ can be verified on the level of the rule.
(2) Although we cannot deduce $\langle t \mid g_R \rangle$, Lemma 4.10.2 implies that we can at least deduce how it is mapped into K'.

(3) Lemma 4.10.3 implies that it suffices to restrict the overapproximation of $\mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong}$ to $\mathrm{Hom}(\mathbb{T}, R')^{t_R}_{\not\cong}$.

(4) If $t \in \mathcal{A}(\mathbf{C})$, then $\langle t \mid w \rangle \in \mathcal{A}(\mathbf{C})$ by the pullback stability assumption. Thus, Lemma 4.10.4 implies that it suffices to restrict the overapproximation even further to $\{f \in \mathrm{Hom}(\mathbb{T}, R')^{t_R}_{\not\cong} \mid \langle f \mid t_R \rangle \in \mathcal{A}(\mathbf{C})\}$.

4.2. Determining $\mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\cong})$.

In this section we show that the weight of $\mathcal{A}(\mathbb{T}, G_R)^w_{\cong}$ can be determined under minimal assumptions.

Proposition 4.12. *If f and g have a pullback of the form on the left of:*

$$
\begin{array}{ccc}
C \xleftarrow{\ f'\ } A' & \qquad & C \xleftarrow{f'i^{-1}} A \\
g\downarrow \quad \mathrm{PB} \quad \downarrow i & \qquad & g\downarrow \quad \mathrm{PB} \quad \| \\
B \xleftarrow{\ f\ } A & \qquad & B \xleftarrow{\ f\ } A
\end{array}
$$

where $i \in \mathrm{iso}(\mathbf{C})$, then the square on the right is also a pullback. Moreover, $f'i^{-1}$ is a unique solution for h in $f = g \circ h$.

Proof. First, $gf' = fi \implies gf'i^{-1} = fii^{-1} \implies gf'i^{-1} = f$. So the right square commutes.

For the pullback property, let $gx = fy$ commute for a span $C \xleftarrow{x} X \xrightarrow{y} A$. Then using the pullback property of the left square, there exists a unique z such that

commutes. Then iz is a solution for w in $f'i^{-1}w = x$ and $1_A w = y$, and it is in fact a unique solution by the second equation. So the commuting square is indeed a pullback.

Finally, that $f'i^{-1}$ is a unique solution for h in $f = g \circ h$ follows easily from the established pullback property [OER23, Proposition 13]. $\qquad\square$

Definition 4.13. For cospans $A \xrightarrow{f} B \xleftarrow{g} C$ with $\langle g \mid f \rangle \in \mathrm{iso}(\mathbf{C})$ iso, we let $\langle f \mid g \rangle^=$ denote the unique h satisfying $f = g \circ h$ (well-defined by Proposition 4.12).

Lemma 4.14. *Let the pullback*

$$
\begin{array}{ccc}
R & = & R \\
t_R\downarrow & \mathrm{PB} & \downarrow w \\
R' & \xleftarrow{\ w'\ } & G_R
\end{array}
$$

be given with $t_R \in \mathcal{A}(\mathbf{C})$. Let $\mathbf{C}$ be $\mathcal{A}$-locally finite and $\mathbb{T}$ a set of objects. Then $\chi : \mathcal{A}(\mathbb{T}, G_R)^w_{\cong} \to \mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}$ defined by $\chi(t) = w' \circ t$ is a domain-preserving bijection.

Proof. That χ is domain-preserving is immediate from the definition.

We show that χ is well-typed. Let $t \in \mathcal{A}(\mathbb{T}, G_R)^w_{\cong}$ and let $dom(t) = T \in \mathbb{T}$. We have

$$
\begin{array}{ccccc}
R & =\!=\!=\!=\!= & R & \leftsquigarrow \langle t|w\rangle^= \rightsquigarrow & T \\
{\scriptstyle\wr} & & \big| & & \| \\
{\scriptstyle t_R} & \text{PB} & {\scriptstyle w} & \text{PB} & \| \\
\downarrow & & \downarrow & & \| \\
R' & \longleftarrow w' \longrightarrow & G_R & \leftsquigarrow t \rightsquigarrow & T
\end{array}
\tag{4.1}
$$

using Proposition 4.12. By the pullback lemma, the outer square is a pullback, and so $\langle t_R \mid w' \circ t \rangle \in \mathrm{iso}(\mathbf{C})$. Moreover, $w' \circ t \in \mathcal{A}(T, R')$, because $w' \circ t = t_R \circ \langle t \mid w \rangle^=$, and $t_R \circ \langle t \mid w \rangle^= \in \mathcal{A}(T, R')$ by stability under composition (Assumption 4.4). Thus, the image of χ lies in $\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}$, and so $\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}$ is a valid codomain.

For injectivity of χ, assume $w' \circ s = w' \circ t$. We then have the diagram

$$
\begin{array}{cccccc}
& & \rightsquigarrow \langle t|w\rangle^= \rightsquigarrow & & & \\
R & \Leftarrow\!=\!=\!= & R & \leftsquigarrow \langle s|w\rangle^= \rightsquigarrow & T & \xleftarrow{\;\;!x\;\;} T \\
{\scriptstyle\wr} & & \big| & & \| & \nearrow \\
{\scriptstyle t_R} & \text{PB} & {\scriptstyle w} & \text{PB} & \| & \\
\downarrow & & \downarrow & & \| & \\
R' & \longleftarrow w' \longrightarrow & G_R & \leftsquigarrow s \rightsquigarrow & T &
\end{array}
\tag{4.2}
$$

where the outer diagram commutes by $w' \circ s = w' \circ t$ and Diagram (4.1). Hence there exists a unique $x : T \to T$ such that $\langle t \mid w \rangle^= = 1_R \circ \langle s \mid w \rangle^= \circ x$ and $1_T = 1_T \circ x$. Hence $x = 1_T$, and by canceling identities, $\langle t \mid w \rangle^= = \langle s \mid w \rangle^=$. Thus $s = w \circ \langle s \mid w \rangle^= = w \circ \langle t \mid w \rangle^= = t$.

For surjectivity of χ, let $f \in \mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}$. We have the diagram

$$
\begin{array}{ccccc}
& & \rightsquigarrow \langle f|t_R\rangle^= \rightsquigarrow & & \\
R & \Leftarrow\!=\!=\!= & R & \leftsquigarrow \langle f|t_R\rangle^= \rightsquigarrow & T \\
{\scriptstyle\wr} & & \big| & & \| \\
{\scriptstyle t_R} & \text{PB} & {\scriptstyle w} & = & \| \\
\downarrow & & \downarrow & & \| \\
R' & \Leftarrow\!=\!=\!= w' =\!=\!= & G_R & \leftsquigarrow w\circ\langle f|t_R\rangle^= \rightsquigarrow & T \\
& & \rightsquigarrow f \rightsquigarrow & &
\end{array}
\tag{4.3}
$$

where the outer square is a pullback. We have $t_R \circ \langle f \mid t_R \rangle^= \in \mathcal{A}(\mathbf{C})$ by stability under composition, and hence $w \circ \langle f \mid t_R \rangle^= \in \mathcal{A}(\mathbf{C})$ by stability under decomposition, using $t_R = w' \circ w$. The right square is moreover a pullback by the pullback lemma. Hence $w \circ \langle f \mid t_R \rangle^= \in \mathcal{A}(\mathbb{T}, G_R)^w_{\cong}$. It then follows that $f = w' \circ w \circ \langle f \mid t_R \rangle^= = \chi(w \circ \langle f \mid t_R \rangle^=)$ lies in the image of χ. $\qquad\square$

Corollary 4.15. *If the conditions of Lemma 4.14 are met, then*

$$
\mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^w_{\cong}) = \mathbf{wt}(\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}). \qquad\qquad\square
$$

4.3. **Sliding Tiles Injectively.** In this section we establish conditions for the existence of a domain-preserving injection $\xi : \mathcal{A}(\mathbb{T}, G_R)^w_{\not\cong} \rightarrowtail \mathcal{A}(\mathbb{T}, G_L)$. Intuitively, one can think of ξ as sliding tiles from right to left across the rewrite step diagram.

If $l' \in \mathrm{rm}(\mathbf{C})$, then ξ will be seen to exist rather straightforwardly. However, in general it suffices to require more weakly that l' preserves any tiles to be slid (and distinctly so). Definitions 4.16 and 4.20 help capture such a weaker requirement. With these definitions, ξ can be shown to exist even for non-trivial rules with non-monic l'.

Definition 4.16. A morphism $g : B \to C$ *preserves an $\mathcal{A}$-factorization* of $f : A \to B$ if there exists a factorization $f = f'' \circ f'$ of f satisfying

- $f'' \in \mathcal{A}(\mathbf{C})$; and
- $g \circ f'' \in \mathcal{A}(\mathbf{C})$.

Lemma 4.17. *Assume $\mathbf{C}$ has pullbacks. Let the diagram*

$$
\begin{array}{ccccc}
A & \leftarrow f' - B & \leftsquigarrow x \leftsquigarrow & X \\
\downarrow g & \mathrm{PB} \quad \downarrow g' & & \\
C & \leftarrow f - D & &
\end{array}
$$

be given, with $x \in \mathcal{A}(\mathbf{C})$. If f preserves an $\mathcal{A}$-factorization of $g' \circ x$, then $f' \circ x \in \mathcal{A}(\mathbf{C})$.

Proof. By assumption, there exists an $\mathcal{A}$-factorization $k' \circ k$ of $g' \circ x$ with $k', (f \circ k') \in \mathcal{A}(\mathbf{C})$. Construct the pullback of $B \xrightarrow{g'} D \xleftarrow{k'} K$ as depicted in diagram

$$
\begin{array}{cccccc}
A & \leftarrow f' - B & \leftsquigarrow k'' \leftsquigarrow & E & \leftarrow y \cdots & X \\
\downarrow g & \mathrm{PB} \quad \downarrow g' & \mathrm{PB} \quad \downarrow g'' & & & \\
C & \leftarrow f - D & \leftarrow k' \leftsquigarrow & K & \leftarrow k &
\end{array}
$$

where $k'' \in \mathcal{A}(\mathbf{C})$ by pullback stability. By the pullback property, there exists a morphism $y : X \to E$ such that $x = k'' \circ y$ and $k = g'' \circ y$. By $x \in \mathcal{A}(\mathbf{C})$ and the decomposition property, $y \in \mathcal{A}(\mathbf{C})$. By the pullback lemma, the two squares compose to form a larger pullback square so that $f' \circ k'' \in \mathcal{A}(\mathbf{C})$ by pullback stability. Finally, by stability under composition, $f' \circ x = (f' \circ k'') \circ y \in \mathcal{A}(\mathbf{C})$. $\qquad\square$

Proposition 4.18. *If $\mathbf{C}$ has (up to isomorphism) unique $(\mathcal{E}, \mathcal{A})$-factorizations (for some class of morphisms $\mathcal{E}$), then the following statements are equivalent:*

(1) $g : B \to C$ *preserves an $\mathcal{A}$-factorization of $f : A \to B$.*
(2) $g \circ a \in \mathcal{A}(\mathbf{C})$ *for $f = a \circ e$ an $(\mathcal{E}, \mathcal{A})$-factorization of f, i.e., $a \in \mathcal{A}$ and $e \in \mathcal{E}$.*

Proof. $1 \implies 2$: let $f = f'' \circ f'$ be given with $f'', (g \circ f'') \in \mathcal{A}(\mathbf{C})$. $(\mathcal{E}, \mathcal{A})$-factorize both f' and f as in the following diagram:

The dashed arrows are in $\mathcal{E}$. Then $(e', f'' \circ a')$ is an $(\mathcal{E}, \mathcal{A})$-factorization of f (using that $\mathcal{A}(\mathbf{C})$ is stable under composition), so that by essential uniqueness of such factorizations, there exists an $(i : X \to Y) \in \mathrm{iso}(\mathbf{C}) \subseteq \mathcal{A}(\mathbf{C})$ with $a = (f'' \circ a') \circ i$. Then $g \circ a \in \mathcal{A}(\mathbf{C})$ by $g \circ a = (g \circ f'') \circ a' \circ i$ and the assumption that $\mathcal{A}(\mathbf{C})$ is stable under composition.
    $2 \implies 1$: immediate by definition. $\qquad\square$

Remark 4.19. In any category $\mathbf{C}$, morphisms $f : A \to B$ trivially admit a unique (identity, morphism)-factorization, namely $(1_{dom(f)}, f)$. So if $\mathcal{A}(\mathbf{C}) = \mathrm{Hom}(\mathbf{C})$, verifying whether $g : B \to C$ preserves an $\mathcal{A}$-factorization of f reduces to verifying whether $g \circ f \in \mathrm{Hom}(\mathbf{C})$, which is immediate. Moreover, a quasitopos admits both (essentially) unique (epi, regular mono)-factorizations and (essentially) unique (regular epi, mono)-factorizations (Proposition 3.5). Thus, if $\mathbf{C}$ is a quasitopos and $\mathcal{A}(\mathbf{C}) = \mathrm{rm}(\mathbf{C})$ or $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$, verifying whether any morphism $g : B \to C$ preserves an $\mathcal{A}$-factorization of f reduces to verifying a property for the respective factorizations.

Definition 4.20 (Monic For). Morphism $h : B \to C$ is *monic for* morphisms $f, g : A \to B$ if $h \circ f = h \circ g$ implies $f = g$.

Lemma 4.21. *Let the diagram*

$$
\begin{array}{ccc}
A \leftarrow f' - B & \overset{\leftarrow x -}{\underset{\leftarrow y -}{}} & X \\
\downarrow{\scriptstyle g} \quad \mathrm{PB} \quad \downarrow{\scriptstyle g'} & & \\
C \leftarrow f - D & &
\end{array}
$$

be given. If f is monic for $g' \circ x$ and $g' \circ y$, then f' is monic for x and y.

Proof. Assume $(\dagger_1)$ $f' \circ x = f' \circ y$. We must show $x = y$. We have:

$$
\begin{aligned}
f' \circ x &= f' \circ y \\
g \circ f' \circ x &= g \circ f' \circ y \\
f \circ g' \circ x &= f \circ g' \circ y \\
g' \circ x &= g' \circ y \qquad\qquad\qquad (\dagger_2)
\end{aligned}
$$

using $g \circ f' = f \circ g'$ and the assumption that f is monic for $g' \circ x$ and $g' \circ y$. By $g \circ (f' \circ x) = f \circ (g' \circ x)$ and the pullback property, there exists a unique z such that in diagram

$$
\begin{array}{ccc}
& \overset{f' \circ x}{\frown} & \\
A \overset{}{\leftarrow} f' - B & \dashleftarrow !z \dashrightarrow X & \\
\downarrow{\scriptstyle g} \quad \mathrm{PB} \quad \downarrow{\scriptstyle g'} & \swarrow{\scriptstyle g' \circ x} & \\
C \leftarrow f - D & &
\end{array}
$$

$f' \circ x = f' \circ z$ and $g' \circ x = g' \circ z$. Solution $z = x$ is trivial, and $z = y$ is a solution by $(\dagger_1)$ and $(\dagger_2)$. Hence $x = y$. $\qquad\square$

The morphism $g_R : G_K \to G_R$ of the rewrite step may identify elements. So for the injection ξ from right to left to exist, we must be able to go into the inverse direction, without identifying tiles. To this end, the following lemma will prove useful.

Lemma 4.22. *If epimorphisms e and e' in diagram*

$$
\begin{array}{ccccc}
A' & - f' \to & B' & \leftarrow g' - & C' \\
\downarrow{\scriptstyle e} & = & \downarrow{\scriptstyle h} & = & \downarrow{\scriptstyle e'} \\
A & - f \to & B & \leftarrow g - & C
\end{array}
$$

are split, then for right inverses $e^{\leftarrow}$ and $e'^{\leftarrow}$, $f'e^{\leftarrow} = g'e'^{\leftarrow} \implies f = g$.

Proof. $f'e^{\leftarrow} = g'e'^{\leftarrow} \implies hf'e^{\leftarrow} = hg'e'^{\leftarrow} \implies fee^{\leftarrow} = ge'e'^{\leftarrow} \implies f = g$. $\qquad\square$

We are now ready for the main theorem of this subsection. We recommend keeping the diagram for Lemma 4.10 alongside it.

Theorem 4.23. *Assume* **C** *has pullbacks. Let a rewrite rule ρ be given with square $r' \circ t_K = t_R \circ r$ PBPO$^+$-adhesive. Fix a class $\mathcal{A}(\mathbf{C})$ and a set of objects $\mathbb{T}$. Define*

$$\Phi = \{f \in \mathrm{Hom}(\mathbb{T}, R')_{\not\cong}^{t_R} \mid \langle f \mid t_R \rangle \in \mathcal{A}(\mathbf{C})\}.$$

If

(1) *for all $f \in \Phi$, $\langle r' \mid f \rangle$ is a split epimorphism; and*
(2) *for some right inverse choice function $(\cdot)^{\leftarrow}$ and for all $f, g \in \Phi$,*
    (a) *l' preserves an $\mathcal{A}$-factorization of $\langle f \mid r' \rangle \circ \langle r' \mid f \rangle^{\leftarrow}$; and*
    (b) *l' is monic for $\langle f \mid r' \rangle \circ \langle r' \mid f \rangle^{\leftarrow}$ and $\langle g \mid r' \rangle \circ \langle r' \mid g \rangle^{\leftarrow}$;*

then for any rewrite step diagram induced by ρ, the function

$$\xi(t) = g_L \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow}$$

defines an injection $\mathcal{A}(\mathbb{T}, G_R)_{\not\cong}^w \rightarrowtail \mathcal{A}(\mathbb{T}, G_L)$.

Proof. We first argue that the use of $(\cdot)^{\leftarrow}$ in $\xi(t)$ is well-defined. Because $t \in \mathcal{A}(\mathbf{C})$, we have $\langle t \mid w \rangle \in \mathcal{A}(\mathbf{C})$ by pullback stability. And $\langle t \mid w \rangle = \langle w' \circ t \mid t_R \rangle$ by Lemma 4.10.4. Moreover, $\langle w \mid t \rangle \notin \mathrm{iso}(\mathbf{C})$, and $\langle w \mid t \rangle = \langle t_R \mid w' \circ t \rangle$ by Lemma 4.10.3. So $\langle w' \circ t \mid t_R \rangle \in \Phi$. Then by local assumption 1, $\langle r' \mid w' \circ t \rangle$ is a split epimorphism. And $\langle r' \mid w' \circ t \rangle = \langle g_R \mid t \rangle$ by Lemma 4.10.1. So $\langle g_R \mid t \rangle^{\leftarrow}$ is well-defined.

We next argue that $\xi(t) \in \mathcal{A}(\mathbb{T}, G_L)$. As established, $\langle r' \mid w' \circ t \rangle = \langle g_R \mid t \rangle$, and by Lemma 4.10.2, $u' \circ \langle t \mid g_R \rangle = \langle w' \circ t \mid r' \rangle$. So the diagram

$$
\begin{array}{ccccc}
G_L & \xleftarrow{\hspace{1.2cm} g_L \hspace{1.2cm}} & G_K & \xleftsquigarrow{\langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow}} & T \\
{\scriptstyle\alpha}\big\downarrow & \text{PB} & {\scriptstyle u'}\big\downarrow & {\scriptstyle \langle w' \circ t \mid r' \rangle \circ \langle r' \mid w' \circ t \rangle^{\leftarrow}} & \\
L' & \xleftarrow{\hspace{1cm} l' \hspace{1cm}} & K' & \xleftarrow{\hspace{1cm}} & \\
\end{array}
$$

commutes, where the pullback square is given by the rewrite step. Moreover, $\langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow} \in \mathcal{A}(\mathbf{C})$ (as indicated in the diagram) by stability under composition, using $\langle t \mid g_R \rangle \in \mathcal{A}(\mathbf{C})$ (by pullback stability and $t \in \mathcal{A}(\mathbf{C})$) and $\langle g_R \mid t \rangle^{\leftarrow} \in \mathrm{rm}(\mathbf{C}) \subseteq \mathcal{A}(\mathbf{C})$ (using Proposition 2.5 and Assumption 4.4). By local assumption 2a and the commuting triangle of the diagram, l' preserves an $\mathcal{A}$-factorization of $u' \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow}$. So by Lemma 4.17, $\xi(t) \in \mathcal{A}(\mathbf{C})$ and consequently $\xi(t) \in \mathcal{A}(\mathbb{T}, G_L)$.

For injectivity of ξ, assume $\xi(t) = \xi(s)$ for $t, s \in \mathcal{A}(\mathbb{T}, G_R)_{\not\cong}^w$. By local assumption 2b, Lemma 4.10.1, and Lemma 4.10.2, l' is monic for

$$\langle w' \circ t \mid r' \rangle \circ \langle r' \mid w' \circ t \rangle^{\leftarrow} = u' \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow}$$

and

$$\langle w' \circ s \mid r' \rangle \circ \langle r' \mid w' \circ s \rangle^{\leftarrow} = u' \circ \langle s \mid g_R \rangle \circ \langle g_R \mid s \rangle^{\leftarrow}.$$

So by Lemma 4.21, g_L is monic for $\langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow}$ and $\langle s \mid g_R \rangle \circ \langle g_R \mid s \rangle^{\leftarrow}$. Then because $\xi(t) = \xi(s)$, $\langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow} = \langle s \mid g_R \rangle \circ \langle g_R \mid s \rangle^{\leftarrow}$. Then finally, $t = s$ by Lemma 4.22. $\square$

4.4. **The Main Result.** We are now ready to prove the main result of this paper (Theorem 4.24) and its corollary (Corollary 4.26). We also show that in rather common settings, many technical conditions of the theorem are met automatically (Lemma 4.28 and Propositions 4.29 and 3.7). We close with a complementary lemma that establishes decreasingness for deleting rules (Lemma 4.30). Examples of applications will be given in Section 5.

Theorem 4.24 (Decreasingness by Element Counting). *Let $\mathcal{T}$ and $\mathcal{T}'$ be disjoint sets of PBPO$^+$ rules. Assume $\mathbf{C}$ has pullbacks and let $\mathcal{A}(\mathbf{C})$ be a class such that $\mathbf{C}$ is $\mathcal{A}$-locally finite. Let $\mathbb{T}$ be a set of objects and $\mathbf{wt} : \mathbb{T} \to \mathbb{N}$ a weight function such that, for every $\rho \in \mathcal{T} \uplus \mathcal{T}'$, the following conditions hold:*

- *ρ's pushout square $r' \circ t_K = t_R \circ r$ is PBPO$^+$-adhesive; and*
- *$t_R \in \mathcal{A}(\mathbf{C})$; and*
- *set $\Phi_\rho = \{ f \in \mathrm{Hom}(\mathbb{T}, R')^{t_R}_{\ncong} \mid \langle f \mid t_R \rangle \in \mathcal{A}(\mathbf{C}) \}$ meets the conditions of Theorem 4.23 for some right inverse choice function $(\cdot)^{\leftarrow}$; and*
- *there exists a set $\Delta_\rho \subseteq \mathcal{A}(\mathbb{T}, L)$ such that*
  - *for all $f \in \Phi_\rho$ and $t \in \Delta_\rho$, $l' \circ \langle f \mid r' \rangle \circ \langle r' \mid f \rangle^{\leftarrow} \neq t_L \circ t$;*
  - *t_L is monic for all $t, t' \in \Delta_\rho$; and*
  - *$\mathbf{wt}(\Delta_\rho) > \mathbf{wt}(\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong})$ if $\rho \in \mathcal{T}$ and $\mathbf{wt}(\Delta_\rho) \geq \mathbf{wt}(\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong})$ if $\rho \in \mathcal{T}'$.*

Then for any rewrite step with match $m \in \mathcal{A}(\mathbf{C})$, induced by a rule $\rho \in \mathcal{T} \uplus \mathcal{T}'$, we have $\mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_L) > \mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_R)$ if $\rho \in \mathcal{T}$ and $\mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_L) \geq \mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_R)$ if $\rho \in \mathcal{T}'$.

Proof. Let a step induced by a $\rho \in \mathcal{T} \uplus \mathcal{T}'$ be given.

By Corollary 4.15, $\mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\cong}) = \mathbf{wt}(\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong})$.

By Theorem 4.23, we obtain an injection $\xi : \mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong} \rightarrowtail \mathcal{A}(\mathbb{T}, G_L)$ with $dom(\xi(t)) = dom(t)$, using the assumption on Φ_ρ. So $\mathbf{wt}(\xi \circ \mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong}) = \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$.

Moreover, by $m \in \mathcal{A}(\mathbf{C})$ and stability under composition, we have $(m \circ \Delta_\rho) \subseteq \mathcal{A}(\mathbb{T}, G_L)$. And by t_L monic for all $t, t' \in \Delta_\rho$, we have m monic for all $t, t' \in \Delta_\rho$, and so $\mathbf{wt}(m \circ \Delta_\rho) = \mathbf{wt}(\Delta_\rho)$. It remains to show that $(m \circ \Delta_\rho)$ and $(\xi \circ \mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$ are disjoint. If for a $t' \in \Delta_\rho$ and $t \in \mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong}$, $m \circ t' = \xi(t)$, then $t_L \circ t' = \alpha \circ m \circ t' = \alpha \circ \xi(t) = \alpha \circ g_L \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow} = l' \circ u' \circ \langle t \mid g_R \rangle \circ \langle g_R \mid t \rangle^{\leftarrow} = l' \circ \langle w' \circ t \mid r' \rangle \circ \langle r' \mid w' \circ t \rangle^{\leftarrow}$, using Lemma 4.10.(1–2) and $\alpha \circ g_L = l' \circ u$, which contradictingly implies $t' \notin \Delta_\rho$ by the definition of Δ_ρ and $w' \circ t \in \Phi$. Thus $\xi(t) \neq m \circ t'$.

In summary,

$$\mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_L) \geq \mathbf{wt}(m \circ \Delta_\rho) + \mathbf{wt}(\xi \circ \mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$$
$$= \mathbf{wt}(\Delta_\rho) + \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$$
$$\succ \mathbf{wt}(\mathcal{A}(\mathbb{T}, R')^{t_R}_{\cong}) + \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$$
$$= \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\cong}) + \mathbf{wt}(\mathcal{A}(\mathbb{T}, G_R)^{w}_{\ncong})$$
$$= \mathbf{wt}^{\mathcal{A}}_{\mathbb{T}}(G_R)$$

for $\succ \; = \; >$ if $\rho \in \mathcal{T}$ and $\succ \; = \; \geq$ if $\rho \in \mathcal{T}'$, completing the proof. $\qquad\square$

Remark 4.25. The requirement $m \in \mathcal{A}(\mathbf{C})$ puts a lower bound on what one can choose for $\mathcal{A}(\mathbf{C})$ in a termination proof. Usually two factors are relevant: the class of t_L, and match restrictions imposed by the setting. More precisely, let $X(\mathbf{C})$ and $Y(\mathbf{C})$ be classes of morphisms. If $t_L \in X(\mathbf{C})$, where $X(\mathbf{C})$ satisfies the decomposition property (meaning

$m \in X(\mathbf{C})$ by $t_L = \alpha \circ m$), and the setting imposes $m \in Y(\mathbf{C})$, then the choice of $\mathcal{A}(\mathbf{C})$ must satisfy $X(\mathbf{C}) \cap Y(\mathbf{C}) \subseteq \mathcal{A}(\mathbf{C})$.

From Theorem 4.24 and Remark 4.25 the following is immediate.

Corollary 4.26 (Termination by Element Counting)**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be disjoint sets of PBPO$^+$ rules. Let $\mathcal{A}(\mathbf{C})$ be a class such that for all rules $\rho \in (\mathcal{T} \uplus \mathcal{T}')$, $t_L(\rho) \in \mathcal{A}(\mathbf{C})$ or matching of ρ is restricted to a class $X \subseteq \mathcal{A}(\mathbf{C})$. If the conditions of Theorem 4.24 are met, then $\mathcal{T}$ terminates relative to $\mathcal{T}'$. Hence $\mathcal{T} \uplus \mathcal{T}'$ is terminating iff $\mathcal{T}'$ is.* $\square$

Remark 4.27 (Choosing $\mathcal{A}(\mathbf{C})$ Per Tile)**.** Theorem 4.24 and the results it depends on still hold if in instead of having $\mathcal{A}(\mathbf{C})$ globally fixed, a class $\mathcal{A}(\mathbf{C})$ is fixed for each individual $T \in \mathbb{T}$, and match morphism m is in the intersection of every class. This for instance allows counting some tiles monically, and others non-monically.

The following lemma implies that in many categories of interest, tilings of L and slid tiles never collide, in which case one condition on Δ_ρ from Theorem 4.24 is vacuously satisfied. For instance, in quasitoposes, pushouts along t_K are pullbacks if $t_K \in \mathrm{rm}(\mathbf{C})$ [Joh02, Lemma A2.6.2].

Lemma 4.28. *Let $\mathbf{C}$ have pullbacks, and let a rule ρ be given. If $t_K \in \mathrm{mono}(\mathbf{C})$ and pushout $r' \circ t_K = t_R \circ r$ is a pullback, then for all $t \in \mathrm{Hom}(T, R')^{t_R}_{\not\cong}$ with $\langle r' \mid t \rangle$ a split epi, we have*

$$l' \circ \langle t \mid r' \rangle \circ \langle r' \mid t \rangle^{\leftarrow} \neq t_L \circ t'$$

for all $T' \in \mathrm{Ob}(\mathbf{C})$ and $t' \in \mathrm{Hom}(T', L)$.

Proof. By contradiction. Assume that for some $t \in \mathrm{Hom}(T, R')^{t_R}_{\not\cong}$ with $\langle r' \mid t \rangle$ a split epi and some $t' \in \mathrm{Hom}(T', L)$, $l' \circ \langle t \mid r' \rangle \circ \langle r' \mid t \rangle^{\leftarrow} = t_L \circ t'$. Then we have the commuting diagram

where
- squares PB(2) and PO + PB are given by ρ;
- squares PB(1) and PB(4) are constructed;
- square PB(3) follows from PB(1) and PB(2), using the pullback lemma;
- morphism x exists by the hypothesis and the pullback property; and
- $\langle r' \mid t \rangle \circ \langle r' \mid t \rangle^{\leftarrow} = 1_T$ by the right inverse property.

By a diagram chase we thus have $t = r' \circ t_K \circ \langle t' \mid l \rangle \circ x$. Then from

$$R \xleftarrow{\quad r \quad} K =\!=\!=\!= K \xleftarrow{\langle t'|l\rangle \circ x} T$$
$$\downarrow t_R \quad \text{PB} \quad \downarrow t_K \quad \text{PB} \quad \| \quad \text{PB} \quad \|$$
$$R' \xleftarrow{\quad r' \quad} K' \xleftarrow{\ t_K\ } K \xleftarrow{\langle t'|l\rangle \circ x} T$$
$$\underbrace{\qquad\qquad\qquad}_{t}$$

and two applications of the pullback lemma, we have $\langle t_R \mid t \rangle \in \mathrm{iso}(\mathbf{C})$, contradicting $t \in \mathrm{Hom}(T, R')^{t_R}_{\not\cong}$. $\qquad\square$

The proposition below states a strong sufficient condition for satisfying the termination method's preconditions. Many graph categories of interest meet these conditions.

Proposition 4.29. *If $\mathbf{C}$ is an rm-locally finite, rm-adhesive quasitopos, then $\mathrm{rm}(\mathbf{C})$ satisfies Assumption 4.4. $\mathbf{C}$ also has all pullbacks and all pushouts, and so in particular the required pushouts described in Assumption 3.11. If moreover $t_L(\rho) \in \mathrm{rm}(\mathbf{C})$, then $m, t_K, t_R \in \mathrm{rm}(\mathbf{C})$, ρ's pushout square is $PBPO^+$-adhesive, and t_L is monic for $\mathcal{A}(\mathbb{T}, L(\rho))$.*

Proof. See Proposition 3.5 for a summary of quasitopos properties. That $\mathrm{rm}(\mathbf{C})$ satisfies Assumption 4.4 was stated in Proposition 4.5. If $t_L \in \mathrm{rm}(\mathbf{C})$, and $m, t_K \in \mathrm{rm}(\mathbf{C})$ by stability under decomposition and pullback, respectively. That $t_R \in \mathrm{rm}(\mathbf{C})$ subsequently follows from pushout stability in quasitoposes [Joh02, Lemma A.2.6.2]. Because of the assumed rm-adhesivity and $t_K \in \mathrm{rm}(\mathbf{C})$, ρ's pushout square is $PBPO^+$-adhesive (Corollary 4.8). Finally, that t_L is monic for $\mathcal{A}(\mathbb{T}, L(\rho))$ follows trivially from the fact that t_L is monic. $\qquad\square$

Finally, we have the following general principle, which does not require any assumptions on t_K and t_R, nor any adhesivity assumptions.

Lemma 4.30 (Deleting Rules Are Decreasing)**.** *Assume $\mathbf{C}$ has pullbacks and is mono-locally finite. Suppose that for a $PBPO^+$ rule ρ, l' is monic, l is not epic, and r is iso; and that for any matches m for ρ, m is monic. Then ρ is decreasing for $\mathbb{T} = \{L\}$, $\mathbf{wt}(L) > 0$, $\Delta_\rho = \{1_L\}$ and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$.*

Proof. Let a rewrite step diagram be given. By stability properties, morphism g_L is monic because l' is monic, and g_R is iso because r is iso. Then

$$\xi : \mathrm{mono}(L, G_R) \rightarrowtail \mathrm{mono}(L, G_L)$$

defined by $\xi(t) = g_L \circ g_R^{-1} \circ t$ is an injection because $g_L \circ g_R^{-1}$ is monic. But ξ is not surjective. Because if $m = \xi(t)$ for some $t \in \mathrm{mono}(L, G_R)$, then in diagram

$$L \xleftarrow{\ l\ } K \xleftarrow{\quad !x \quad} L$$
$$\downarrow m \quad \text{PB} \quad \downarrow u \quad\quad \|$$
$$G_L \xleftarrow{\ g_L\ } G_K \xleftarrow{g_R^{-1}} G_R \xleftarrow{\ t\ } L$$
$$\underbrace{\qquad\qquad\qquad}_{\xi(t)}$$

a unique $x : L \to K$ is obtained that makes the diagram commute, using the pullback square on the left. Thus $l \circ x$ is an isomorphism, and hence l an epimorphism, which contradicts our assumption about l. So $m \in \mathrm{mono}(L, G_L)$ does not lie in the image of ξ. It follows that $\mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G_L) > \mathbf{wt}_{\mathbb{T}}^{\mathcal{A}}(G_R)$. $\qquad\square$
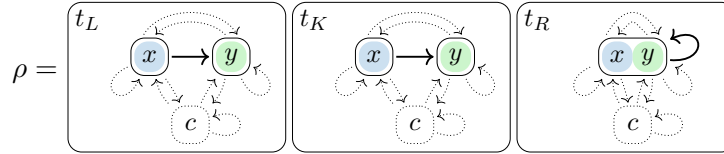
5. Examples

We give a number of examples of applying Theorem 4.24 in category $\mathbf{C} = \mathbf{FinGraph}$ (Definition 2.7), each demonstrating new features. For each example, we will fix $\mathbb{T}$, $\mathbf{wt}$, and $\mathcal{A}(\mathbf{C})$, and usually some properties of the relevant morphism sets (such as cardinalities) or related comments. The remaining details of the proofs are routine. Note that in $\mathbf{FinGraph}$ (and more generally in any topos), $\mathrm{rm}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$, and because the rules in examples satisfy $t_L \in \mathrm{mono}(\mathbf{C})$, we are in each case free to choose $\mathrm{mono}(\mathbf{C})$ or $\mathrm{Hom}(\mathbf{C})$ for $\mathcal{A}(\mathbf{C})$ (Remark 4.25).

Notation 5.1 (Visual Notation). In our examples of rules, the morphisms $t_X : X \hookrightarrow X'$ ($X \in \{L, K, R\}$) of rules are regular monos (embeddings). We depict t_X by depicting the graph X', and then let solid, colored vertices and solid edges denote $t_X(X)$, with dotted blank vertices and dotted edges the remainder of X'. For example, in Example 5.2 below, subgraph L of $t_L : L \hookrightarrow L'$ is $\boxed{x} \longrightarrow \boxed{y}$.
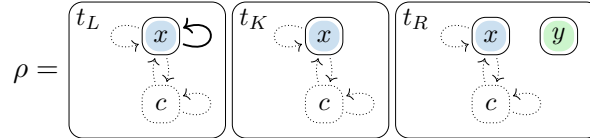
The vertices of graphs are non-empty sets $\{x_1, \ldots, x_n\}$ depicted by boxes $\boxed{x_1 \;\cdots\; x_n}$. When depicting the homomorphism $(r' = (r'_V, r'_E)) : K' \to R'$ we will choose the vertices of K' and R' in such a way that component r'_V is fully determined by $S \subseteq r'_V(S)$ for all $S \in V_{K'}$. For example, for nodes $\{x\}, \{y\} \in V_{K'}$ of Example 5.2 below (in which morphism r' is implicit), $r'(\{x\}) = r'(\{y\}) = \{x, y\} \in V_{R'}$. If component r'_E is not uniquely determined by r'_V, then let r'_E preserve the relative positioning of the edges (although normally, this choice will be inconsequential). Morphism $l' : K' \to L'$ is depicted similarly.
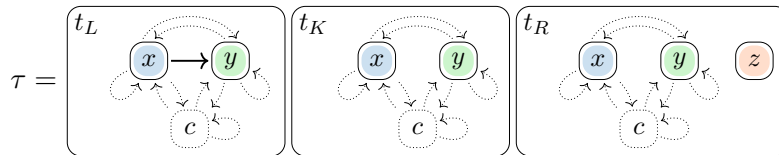
Example 5.2 (Folding an Edge). The rule

$$\rho = \quad \boxed{\begin{array}{c} t_L \\ \boxed{x} \longrightarrow \boxed{y} \\ c \end{array}} \quad \boxed{\begin{array}{c} t_K \\ \boxed{x} \longrightarrow \boxed{y} \\ c \end{array}} \quad \boxed{\begin{array}{c} t_R \\ \boxed{x\;y} \circlearrowright \\ c \end{array}}$$

folds a non-loop edge $\cdot \to \cdot$ into a loop $\cdot \circlearrowleft$ (in any context). Define tile set $\mathbb{T} = \{\cdot \to \cdot\}$ with $\mathbf{wt}(\cdot \to \cdot) = 1$ and fix $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. Then $|\Phi_\rho| = 5$ (every non-loop dotted edge in R', and the loop on c). For every $f \in \Phi_\rho$, $\langle r' \mid f \rangle$ is iso and hence a regular epi with a unique right inverse. Because l' is monic (even iso), the remaining details of Theorem 4.23 are immediate. Finally, for the only choice of Δ_ρ, $\mathbf{wt}(\Delta_\rho) = 1 > \mathbf{wt}(\mathrm{mono}(\mathbb{T}, R')^{t_R}_{\cong}) = \mathbf{wt}(\varnothing) = 0$. So ρ is terminating.

Example 5.3. Let the rewrite system $\mathcal{T}$ consist of rules

$$\rho = \quad \boxed{\begin{array}{c} t_L \\ \boxed{x} \circlearrowright \\ c \end{array}} \quad \boxed{\begin{array}{c} t_K \\ \boxed{x} \\ c \end{array}} \quad \boxed{\begin{array}{c} t_R \\ \boxed{x} \quad \boxed{y} \\ c \end{array}}$$

and

$$\tau = \quad \boxed{\begin{array}{c} t_L \\ \boxed{x} \longrightarrow \boxed{y} \\ c \end{array}} \quad \boxed{\begin{array}{c} t_K \\ \boxed{x} \quad \boxed{y} \\ c \end{array}} \quad \boxed{\begin{array}{c} t_R \\ \boxed{x} \quad \boxed{y} \quad \boxed{z} \\ c \end{array}}$$

Rule ρ deletes a loop in any context, and adds a node; and rule τ deletes a non-loop edge in any context, and adds a node.

Because the r morphisms of ρ and τ are not iso, Lemma 4.30 cannot be applied. But for $T = \cdot \to \cdot$, $\mathbb{T} = \{T\}$, $\mathbf{wt}(T) = 1$, $\mathcal{A}(\mathbf{C}) = \mathrm{Hom}(\mathbf{C})$, and Δ_{ρ_1} and Δ_{ρ_2} the singleton sets containing the unique morphisms $T \to L(\rho)$ and $T \to L(\tau)$, respectively, $\mathcal{T}$ is proven decreasing and thus terminating by Theorem 4.24. This argument captures the natural argument: "the number of edges is decreasing".

An alternative argument lets $\mathbb{T} = \{T, T'\}$, with $T' = \cdot \circlearrowleft$, $\mathbf{wt}(T) = \mathbf{wt}(T') = 1$, and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. Then Δ_{ρ} contains the unique mono $T' \rightarrowtail L(\rho)$ and Δ_{τ} the unique mono $T \rightarrowtail L(\tau)$. This captures the argument: "the sum of loop and non-loop edges is decreasing".

Remark 5.4 (Element Counting in Fuzzy Presheaves). In a fuzzy graph category (Example 3.9), rules that change labels (but leave the structure of the graph unchanged) can be proven terminating by using $\mathcal{A}(\mathbf{C}) = \mathrm{rm}(\mathbf{C})$, but not always by using $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. For instance, a rule that increases a loop edge label a into label $b > a$, is shown terminating by $\mathbb{T} = \{\cdot \circlearrowleft a\}$ and $\mathcal{A}(\mathbf{C}) = \mathrm{rm}(\mathbf{C})$, but no proof exists for $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$, because $\cdot \circlearrowleft b$ has strictly more monic elements than $\cdot \circlearrowleft a$.

Example 5.5. The rule
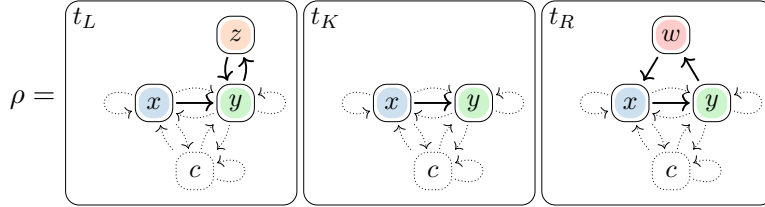
is proven terminating by $\mathbb{T} = \{\cdot \rightleftarrows \cdot\}$, $\mathbf{wt}(\cdot \rightleftarrows \cdot) = 1$, and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$.

This example can be used to show that applicability of our technique is not invariant under rule equivalence. Let ρ' be obtained from ρ by dropping the solid edge from x to y in K and K'. The rules ρ and ρ' are equivalent: they induce the same rewrite relation. However, our termination technique fails for ρ', because not all tiles can be intactly transferred. We therefore pose the following open question. *Does there exist a procedure that maps a rule τ onto an equivalent "standard" representative $\mathrm{standard}(\tau)$, such that the termination method fails on $\mathrm{standard}(\tau)$ iff it fails on all rules in the equivalence class? And if it exists, can it be formulated on the level of (rm-adhesive) quasitoposes?*

**Example 5.6.** Consider the rules ρ and τ, respectively:
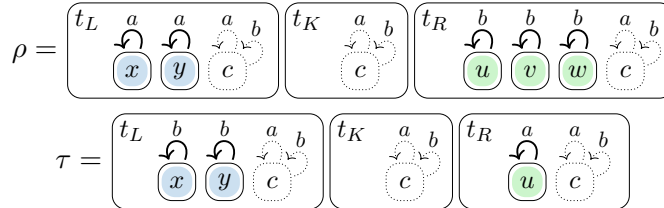
Intuitively, these rules model replacements in multisets over $\{a, b\}$. The elements of the multiset are modeled by nodes with loops that carry the label a or b, respectively. Rule ρ replaces two a's by three b's, and τ replaces two b's by one a.

These rules are terminating. To prove this, we use tiles $\mathbb{T} = \{\cdot \circlearrowleft a, \cdot \circlearrowleft b\}$ together with the weight assignment $\mathbf{wt}(\cdot \circlearrowleft a) = 5$ and $\mathbf{wt}(\cdot \circlearrowleft b) = 3$, and let $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. The context is isomorphically preserved along l' and r', and partial overlaps with the pattern are not possible. So Theorem 4.23 is easily verified for Φ_{ρ} and Φ_{τ}. Then for the obvious largest

choices of Δ_ρ and Δ_τ, we have $\mathbf{wt}(\Delta_\rho) = 2 \cdot 5 = 10 > \mathbf{wt}(\mathrm{mono}(\mathbb{T}, \rho(R'))_{\cong}^{\rho(t_R)}) = 3 \cdot 3 = 9$ for ρ and $\mathbf{wt}(\Delta_\tau) = 2 \cdot 3 = 6 > \mathbf{wt}(\mathrm{mono}(\mathbb{T}, \tau(R'))_{\cong}^{\tau(t_R)}) = 5$ for τ, completing the proof.

The above termination proof works also for vast generalizations of the rules. For instance, rule ρ can be generalized to rule ρ' as follows:
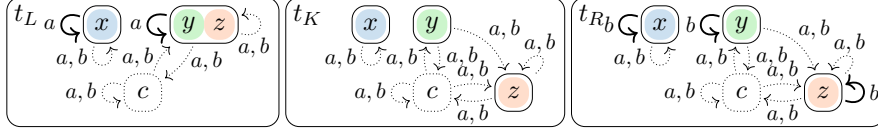
Observe that L' now allows an unbounded number of additional loops on the nodes, and an unbounded number of edges between the nodes and the context. The morphism l' preserves the loops, duplicates a node (including its incident edges from and to the context), and unfolds loops between the duplicated nodes. The system $\{\rho', \tau\}$ is still proven terminating using the argument above.

More generally, as long as l' and r' of a rule do not create new loops other than those specified by l and r, the rule can be proven terminating using the argument above.

Example 5.7. Consider the following rules:
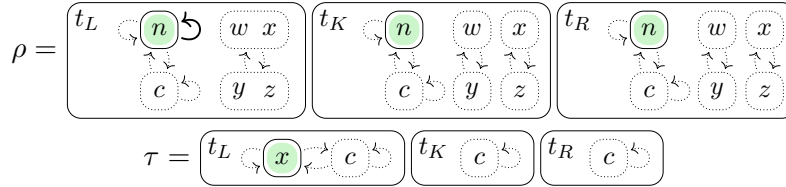
Rule ρ deletes an arbitrary loop, and in doing so, allows arbitrarily many bipartite graph components in the context to duplicate (such components can either be mapped onto node c or onto the right subgraph component). Note that this makes the rule non-deterministic. Rule τ deletes an arbitrary node including incident edges.

Termination of the system can be proven as follows. Let $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. Use the tile set $\mathbb{T} = \{\cdot\circlearrowleft\}$ with the weight assignment $\mathbf{wt}(\cdot\circlearrowleft) = 1$. Then ρ is decreasing and τ is non-increasing, and so it suffices to prove τ terminating, whose termination is immediate from Lemma 4.30.

The derivational complexity[3] of this system is $O(2^m n)$ where $m = |E_G|$ and $n = |V_G|$ of the starting graph G. This can be seen as follows. In the worst case, G consists of

- one distinguished node x with m loops; and
- the remaining $n - 1$ nodes without any incident edges.

In this case, each of the m edges gives rise to an application of rule ρ, and each node forms a bipartite graph component, with the exception of node x. Assuming in the worst case that

- all ρ-steps precede τ-steps; and
- all of the nodes except x are duplicated at every ρ-step;

this scenario gives rise to the following recurrence relation $max\_len(m, n)$ for computing the derivation length:

- $max\_len(0, n) = n$; and
- $max\_len(m + 1, n) = 1 + max\_len(m, 2n - 1)$.

---

[3]The *derivational complexity* of a terminating term rewrite system is a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n)$ is the length of the longest rewrite sequence for any starting term of size n.

Using a routine inductive proof it can be shown that

$$max\_len(m, n) = 2^m(n - 1) + m + 1$$

which is in $O(2^m n)$.

Example 5.8. We consider a system consisting of two DPO rules, given by Plump [Plu18, Example 3], [Plu95, Example 3.8]. The equivalent PBPO$^+$ rules (using the standard encoding [OER23, Definition 71]) are as follows:
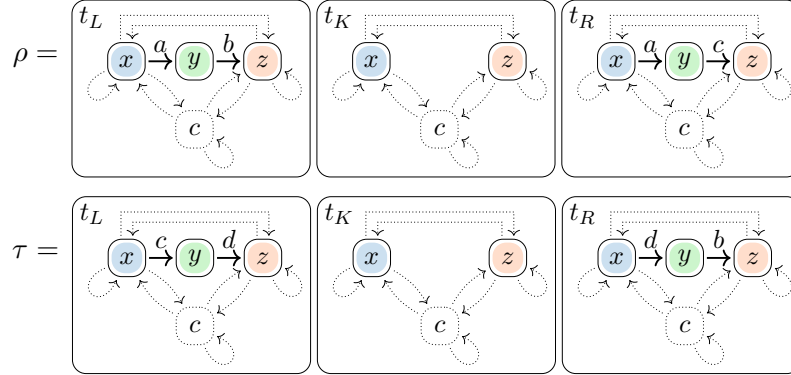
These are graph transformation versions of the string rewrite rules $ab \to ac$ and $cd \to db$, respectively. Note that both rules specify that node y is not allowed to have any incident edges other than those shown in L.

We can use $\mathbb{T} = \{L(\rho)\}$ for our tile set, with $\mathbf{wt}(L(\rho)) = 1$ and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. Then rule ρ is clearly decreasing. Rule τ is non-increasing, because the creation of the b-edge cannot create new $L(\rho)$ subobjects, due to y not having any incoming a-edges. So τ is non-increasing and we can remove ρ by relative termination. Next, we can prove τ terminating using tile with weight 1 and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$, i.e., by counting non-loop c-edges.

We give one example in a hypergraph category that was originally considered by Plump [Plu18, Example 6].

Example 5.9 (Hypergraphs)**.** Let $\mathbf{I}$ be the category freely generated by:

We view V as the type of expression roots. There are three types of hyperedges: $+$, s, and 0, each associated with a root (r_+, r_S, and r_0, respectively). Moreover, $+$ is associated with two (ordered) arguments a_+^1 and a_+^2, and s with one argument a_s^1 (s should be thought of as representing the successor function). The functor category $[\mathbf{I}, \mathbf{FinSet}]$ has hypergraphs as objects and hypergraph homomorphisms as arrows. Moreover, it is a topos (Proposition 3.7).

Not all objects of $[\mathbf{I}, \mathbf{Set}]$ would be considered well-formed under our suggested interpretation of these hypergraphs as expressions (in particular, there could be cycles and parallel hyperedges), but our termination argument below is valid even in the presence of such non-well-formed hypergraphs.

The DPO rules considered by Plump [Plu18, Example 6] in this category are as follows:

$$\rho =$$

$$\tau =$$
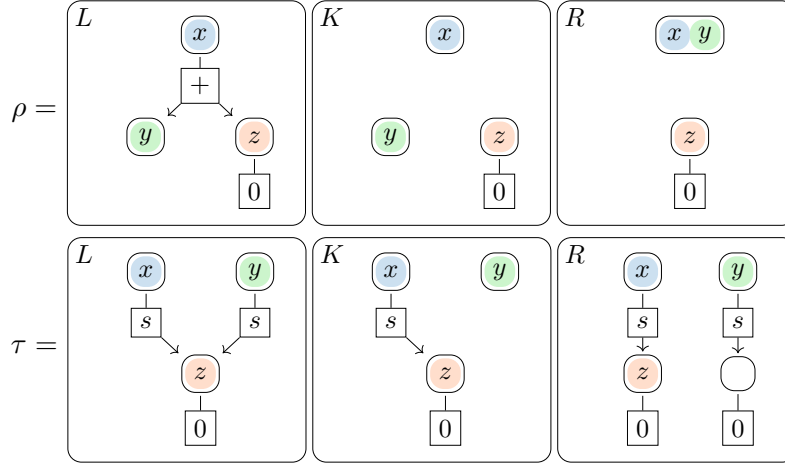
They can be encoded into PBPO$^+$ using the standard encoding [OER23, Definition 71]. That is, $t_X = (\eta_X : X \hookrightarrow T(X))$ for $X \in \{L, K\}$ (with (η, T) the partial map classifier of the category), and R' is obtained by pushing out t_K along r. Note that because vertex w is fresh in R, it does not have any connections in R' other than those shown in R.

To prove termination, we first use tile
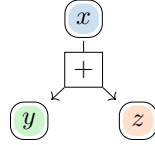
with weight 1 and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. This proves ρ decreasing and τ non-increasing, so that we can remove rule ρ.

We then remove the second rule using tile
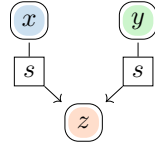
with weight 1 and $\mathcal{A}(\mathbf{C}) = \mathrm{mono}(\mathbf{C})$. Observe that a node in the host graph with n incoming s-edges allows for $n \cdot (n - 1)$ embeddings of this tile: this observation relates our argument to that of Plump [Plu18, Example 6].

6. Related Work

We consider two closely related approaches by Bruggink et al. [BKZ14, BKNZ15] and the approach by Endrullis and Overbeek [EO24] to be the most relevant to our method.[4] These approaches use weighted type graphs T to measure graphs G by means of counting weighted morphisms $G \to T$ (instead of weighted morphisms $T' \to G$ for tiles T'). So the general idea is dual to the one in this paper. Moreover, to our knowledge, these approaches are the only systematic termination methods in the algebraic tradition based on decreasing interpretations.

---

[4]The recently updated arXiv version [BKNZ23] of [BKNZ15] corrects some errors in the theory.

The methods by Bruggink et al. are defined for DPO in the category of edge-labeled multigraphs. The first approach [BKZ14] requires that l and r of DPO rules $L \xleftarrow{l} K \xrightarrow{r} R$, and matches $m : L \rightarrowtail G_L$, are monic. The second approach [BKNZ15] has no such restrictions.

Because our method is applicable in a much broader setting, our method will prove rules terminating that are outside the scope of the methods by Bruggink et al. Nonetheless, it is interesting to ask how the approaches relate in settings where they are all defined.

On the one hand, although Examples 5 and 6 of [BKNZ15] are within the scope of our method, our method cannot prove them terminating. The intuitive reason is that the examples terminate because of global properties, rather than local ones. On the other hand, Example 6.1 below defines a DPO rule that falls inside the scope of all three methods, and only our method can prove it terminating. In conclusion, within the restricted setting, the methods are incomparable.

Example 6.1. Consider the following DPO rule ρ in category **FinGraph**:
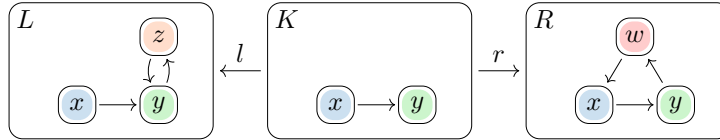
and assume matching is required to be monic. This requirement is often used in practice, because monic matching increases the expressiveness of DPO [HMP01].

The approach in [BKZ14] cannot prove ρ terminating. For establishing termination (on all graphs), the weighted type graph T has to contain a node with a loop (called a *flower node*). The flower node ensures that every graph G can be mapped into T. Then, in particular, the technique requires a weight decrease (from L to R) for the case that the interface K is mapped onto the flower node. However, this makes L and R indistinguishable for the technique in [BKZ14].

Although matches are required to be monic, the method of [BKNZ15] overapproximates for unrestricted matches by design. Observe that if matching is not monic, then graph L of ρ, but with x and y identified, rewrites to itself, meaning ρ is not terminating. As a consequence, the overapproximation of [BKNZ15] causes it to fail in proving ρ terminating for the monic matching setting. (For the same reason, the method of [BKNZ15] fails on the simpler top span of Example 5.2, which is a DPO rule, for the monic matching setting.)

Rule ρ can be proven terminating with our method as follows. Encode ρ into PBPO$^+$ using the standard encoding [OER23, Definition 71].). The resulting rule and a termination proof were given in Example 5.5 above.

Additional examples by Bruggink et al. that our method can prove are Example 4 of [BKZ14] (= Example 2 of [BKNZ15]), and Example 4 of [BKNZ15]. Additional examples that our method cannot prove are Example 1 and the example of Section 3.5 of [BKZ14]. However, unlike the earlier referenced Examples 5 and 6 of [BKNZ15], these examples are in reach if our morphism counting technique can take into account antipatterns (Remark 6.2 below), because they terminate because of local properties.

Remark 6.2 (Antipatterns). A rule that matches an isolated node, and adds a loop, cannot be proven terminating with our method. For this, one must be able to count nodes *without* loops (an *antipattern*), which is currently unsupported. We believe that extending our method with support for such antipatterns is a natural first step for significantly strengthening it.

In [EO24], we generalize and strengthen the approach of [BKNZ15] to (a) be applicable in a wider range of categories and (b) be applicable when termination depends on matching restrictions. This enhanced weighted type graph technique can prove termination of Example 6.1. However, it still fails for other systems which can be proven terminating by the approach presented in the current paper (for instance Example 6 of [Plu18]). Thus, even in settings where both methods are defined, the methods are strictly incomparable.

We discuss some additional related work. An early systematic termination criterion for hypergraph rewriting with DPO, due to Plump, is based on the concept of forward closures [Plu95]. Both of the examples proven terminating with forward closures, Example 3.8 (Example 5.8 above) and Example 4.1 of [Plu95], can be handled with our method.

More recently, Plump formulated a modularity criterion for hypergraph rewriting using DPO [Plu18]: the union of two terminating systems is terminating if there are no sequential critical pairs. Of this paper, our method can prove three out of four examples: Examples 3 (= Example 3.8 of [Plu95]), 5 and 6. The modeling of Example 6 is Example 5.9 above. Our method cannot prove Example 4 (= the already discussed Example 5 of [BKZ14]). It would be interesting to assess the strength of the modularity criterion (especially if generalized to PBPO$^+$) combined with our method.

Bruggink et al. have shown that string rewriting rules are terminating on graphs iff they are terminating on cycles [BKZ14], making cycle rewriting techniques [ZKB14, SZ17] applicable to graph transformation systems consisting of string rewrite rules. Similarly, in another paper [OE21] we have shown that particular PBPO$^+$ encodings of linear term rewrite rules are terminating on graphs iff they are terminating on terms.

There also exist a variety of approaches that generalize TRS methods (such as simplification orderings) to acyclic term graphs [Plu99, Plu97, MS16] and drags [DJ19, DJO23] (that possibly contain cycles) [DJ18].

Remark 6.3 (Relating Subgraph Counting with TRS Termination Methods)**.** An interesting question is how the technique presented in this chapter relates to the many techniques available for term rewriting systems. Zantema [Ter03, Chapter 6] has roughly categorized the available termination techniques for TRSs as follows:

- syntactical methods, which inductively define a well-founded order directly on the terms;
- transformational methods, which define non-termination preserving transformations Φ on rewrite systems (Σ, R), such that termination of (Σ, R) can be proven by applying other techniques to $\Phi((\Sigma, R))$; and
- semantical methods, which interpret terms into some well-founded order using weight functions.

One of the most well-known syntactical methods for TRSs is the recursive path order (RPO) by Dershowitz [Der82]. RPO has been adapted for acyclic term graphs and [Plu99, Plu97, MS16] and drags [DJ18].

Well-known transformational methods for TRSs include dependency pairs [AG00] and semantic labeling [Zan95]. As far as we know, no versions for graphs have been proposed.

For the semantical methods, finally, a fundamental result is that a TRS is terminating iff it admits a compatible well-founded monotone algebra [Ter03, Theorem 6.2.2.]. This of course also implies that finding such an algebra is hard. For TRSs, there exist well-known heuristics for finding polynomial interpretations [CL87] and matrix interpretations [HW06, EWZ06]. The weighted type graph approach by Bruggink et al. [BKZ14, BKNZ15], discussed above, essentially corresponds to the matrix interpretation method.

Our weighted subgraph counting approach is distinctly semantical, but as far as we can see, it does not correspond directly to an existing method for term rewriting systems. Because an analogous approach would not exploit the inductive term structure, we also do not believe that it would be very powerful in the TRS setting.

7. Implementation

We have implemented our termination method using Scala 3 [OSVS21]. The software package is called `graphTT` [OE24], and it includes a REPL (read-eval print loop) for exploring relative termination proofs iteratively, where in each step detailed feedback about the proof process is provided (Section 7.1). Our algorithm is implemented generically for general categories, and one category is included (together with parsing methods for input files): **FinGraph**, the category of finite edge- and vertex-labeled directed multigraphs. After describing the REPL, we give a brief description of the implementation (Section 7.2), which can be viewed as an example of computational (applied) category theory (see Remark 7.1 below).

Remark 7.1 (Related Work). *Computational (applied) category theory*, initiated by Rydeheard and Burstall [RB88] in 1988, bridges the gap between programming and category theory. More specifically, Rydeheard and Burstall implemented a large body of category theory in ML. For algebraic graph transformation in particular, Minas and Schneider [MS10] proposed a Java implementation in 2010. In more recent years, Brown et al. [BPHF23] have implemented algebraic graph rewriting in Julia, as part of the broader AlgebraicJulia[5] [HPBF20,PLF21] applied category theory project. The work by Brown et al. is especially interesting because the code is both general and performant.

7.1. **REPL.** We describe the REPL used for exploring relative termination proofs by running through a simple example session.

When the REPL is invoked, it starts in system selection mode. One of the available commands is `help`, which shows the available commands. The output is as follows:

```
=== GraphTT REPL ===
>> You are in system selection mode.
>> Type 'help' to view the available commands.
graphTT> help
>> Available commands:
select [n]  : select system n for termination proving
inspect [n] : inspect option (system/tile) n in detail
systems     : list the available systems
help        : print all available commands
exit        : exit the program
```

The list of available systems at the time of writing is as follows:

```
graphTT> systems
>> The following systems were loaded:
(0) multiset_as_graph
(1) delete_loop_and_nonloop
```

---

[5]https://www.algebraicjulia.org

```
(2) unfold_to_triangle
(3) folding_an_edge
(4) duplicating_bipartite_components
(5) generalized_multiset_as_graph
```

When the REPL is started, the available systems are read from relative directory `./ src/main/resources/labeled/systems`. The included systems correspond to examples provided in this paper. The correspondences are indicated in the included files using comments (`/*...*/`). For example, the contents of file `folding_an_edge.pbpop` are as follows:

```
graphTT> inspect 2
SYSTEM: folding_an_edge
/* Example 5.2 of
Termination of Graph Transformation Systems
Using Weighted Subgraph Counting
*/

=== rho ===
L  { x:0 -P:0-> y:0 }
L' { x:0 -P:0-> y:0
       x:0 -XX:0-> x:0
       y:0 -YY:0-> y:0
       x:0 -XY:0-> <-YX:0- y:0
       x:0 -XC:0-> <-CX:0- c:0
       c:0 -CY:0-> <-YC:0- y:0
       c:0 -CC:0-> c:0 }
K  { x:0 -P:0-> y:0 }
K' { x:0 -P:0-> y:0
       x:0 -XX:0-> x:0
       y:0 -YY:0-> y:0
       x:0 -XY:0-> <-YX:0- y:0
       x:0 -XC:0-> <-CX:0- c:0
       c:0 -CY:0-> <-YC:0- y:0
       c:0 -CC:0-> c:0 }
R  { x.y:0 -P:0-> x.y:0 }
```

The file contains the system of Example 5.2, which consists of a single PBPO$^+$ rule ρ over unlabeled graphs. Elements and their labels are written as `id:label`. By convention, we use lowercase for vertex identities, uppercase for edge identities, and we label unlabeled elements with `0`. Edges require two endpoints and an indication of direction, and vertices without incident edges can be listed separately. The rule morphisms are implicitly defined and follow the convention described in Notation 5.1. For instance, vertices `x` and `y` of K are mapped onto vertex `x.y` of R: so `.` is used to compose identities.

Let us prove system `folding_an_edge` terminating. We do this by selecting it, after which the REPL enters into proof mode:

```
graphTT> select 3
>> Entering proof mode for system folding_an_edge.
>> The system consists of 1 rule.
```

```
>> The system is as follows:

[... printed system as above ...]
```

In proof mode, the available commands are as follows:

```
graphTT> help
>> Available commands:
use [i w c]+ :
use tile i with weight w, and count morphisms of class c, where:
- i and w are integers (and w is positive), and
- c is a character (r: regular monos, m: monos, h: homomorphisms)
multiple tiles can be specified.
for example, 'use 3 4 h 5 9 r' uses:
- tile 3 with weight 4 (counting homomorphisms), and
- tile 5 with weight 9 (counting regular monos)
inspect [n] : inspect option (system/tile) n in detail
back : return to system selection mode
help : print all available commands
tiles : list the available tiles
exit : exit the program
```

Tiles are loaded from relative directory `./src/main/resources/labeled/tiles`. At the time of writing, these tiles are available:

```
>> The following tiles were loaded:
(0) two_opposing_edges
(1) single_loop
(2) single_node
(3) single_nonloop_edge
(4) a_loop
(5) b_loop
```

As Example 5.2 suggests, we just need a single unlabeled non-loop edge, which is provided:

```
graphTT> inspect 3
TILE: single_nonloop_edge
x:0 -XY:0-> y:0
```

Example 5.2 also suggests that a weight assignment of 1 is sufficient, and that we should count (regular) monos. Note that our implementation allows choosing a class $\mathcal{A}(\mathbf{C})$ on a tile-by-tile basis (see Remark 4.27). As the description for `use` suggests in the `help` output above, we can use tile 3 with weight 1 (and counting monos) by writing `use 3 1 m`.

Using tile 3 with weight 1 produces the following output (commentary on the output follows afterwards):

```
graphTT> use 3 1 m

=============== SYSTEM TERMINATION REPORT ===============
---------------          SUMMARY          ---------------

The system has 1 rules, named: rho
Was the sliding successful for every rule? yes
```

```
    Provably decreasing rules: rho
    Provably nonincreasing (but not provably decreasing) rules:
    Possibly increasing rules:
    The pruned system contains rules:

    The pruned system is empty, so the system is TERMINATING.

    ---------------   DETAILED RULE REPORTS   ---------------

    >>>>>>>>>>>>>>> rule rho <<<<<<<<<<<<<<<
    Summary:
    - The sliding is SUCCESSFUL.
    - The weight of Delta is 1.
    - The weight of R is 0.
    - Conclusion: the rule is PROVABLY DECREASING.

    The details per tile for this rule now follow.

    ~~~ Tile single_nonloop_edge with weight 1, counting MONOS only
    x:0 -XY:0-> y:0

    - The tiling of R has size:           0
    - Giving a weight of:                 0 * 1 = 0
    - A largest valid tiling of L has size: 1
    - Giving a weight of:                 1 * 1 = 1

    Slide data:

    # morphisms into R':        10
    # of which valid:           5
    # iso in R:                 0
    # noniso in R:              5
    # number of ways to slide:  1

    >> The pruned system is empty!
    >> You have proven system folding_an_edge terminating.
    >> Returning to system selection mode.
```

As can be seen:

(1) a summary is given on the system level, indicating whether the sliding was successful for every rule, and partitioning the rules according to whether they are provably decreasing, provably nonincreasing (but not provably decreasing), or neither (i.e., possibly increasing);

(2) a summary is given for every rule ρ of the system, indicating, among others, the largest possible weight for Δ_ρ (Theorem 4.24), and the weight of R;

(3) for every rule ρ and tile T, detailed data is given for the slide analysis. Here,
   • morphisms into R' indicates $|\mathrm{Hom}(T, R')|$;

- of which valid indicates $|S|$ for $S = \{t \in \mathrm{Hom}(T, R') \mid \langle t \mid t_R \rangle \in \mathcal{A}(\mathbf{C})\}$;
- iso in R indicates $\mathrm{Hom}(T, R')^{t_R}_{\cong} \cap S$;
- noniso in R indicates $\mathrm{Hom}(T, R')^{t_R}_{\not\cong} \cap S$; and
- ways to slide indicates the number of possible slid tile sets into L'. The number of choices for right inverses do not matter here: it is the number of outcomes.

Moreover, the system is automatically pruned if possible. In this example the pruned system is empty, meaning the system is terminating. If the system is not empty, the process repeats, and the user can select a new set of tiles for the pruned system.

7.2. Implementation Details.
The implementation consists of seven packages, named `categorytheory`, `labeledgraph`, `parsing`, `repl`, `rewriting`, `termination`, and `util`. We comment on packages `categorytheory`, `labeledgraph`, `rewriting`, and `termination`.

7.2.1. *Package `categorytheory`.*
Package `categorytheory` contains general purpose categorical definitions, organized in a highly modular fashion. The library is designed to be reusable for other purposes. Some of the more complex definitions include the notions of a presheaf category and a topos.

Most of the definitions in package `categorytheory` are defined as *traits*. A Scala trait can be described as a more powerful version of Java's *interface*. A trait can take type parameters and define (abstract or implemented) members. It can also mix in other traits, which is Scala's way of implementing multiple inheritance. The definition of a topos is given in Listing 1, for instance, where type parameters `O` and `A` represent the collections of objects and arrows, respectively. All members of `Topos` are inherited from the `Quasitopos` trait, and three default implementations are provided. The `RmAdhesive` trait is empty: mixing it in constitutes a promise that the category is rm-adhesive.

LISTING 1. Topos.scala

```scala
trait Topos[O, A] extends Quasitopos[O, A] with RmAdhesive:
  final def isRegularEpic(f: A): Boolean = isEpic(f)
  final def isRegularMonic(f: A): Boolean = isMonic(f)
  final def regularEpiMonoFactorization(f: A): (A, A) =
    epiRegularMonoFactorization(f)
```

We have also defined a shorthand type name `TerminationCategory` for the intersection of the types listed in Listing 2. Our termination method implementation expects that the ambient category is a `TerminationCategory`. Note that this intersection is strictly stronger than the requirements listed in the paper: it could be generalized in the future, and the algorithm made more fine-grained, if the use case arises.

LISTING 2. TerminationCategory.scala

```scala
type TerminationCategory[O, A] = Quasitopos[O, A] & RmAdhesive
```

7.3. Package `labeledgraph`.
Package `labeledgraph` defines labeled graphs and graph homomorphisms as defined in Definition 2.6. It also defines `LabeledGraphCategory` as an instance of the `Topos` trait. Enumerations of homomorphisms are not naively implemented, but also not fully optimized. For our present purposes, this is sufficient.

7.4. **Package `rewriting`.** Package `rewriting` contains, among others, the definitions of a PBPO$^+$ rule and step. For computing the step, an immediate question is how morphism $u : K \to G_K$ is to obtained (Definition 3.10). The lemma below demonstrates a few things of interest:

(1) the pullback square $t_L \circ l = l' \circ t_K$ of the rewrite rule does not need to be used in the construction;

(2) the explicit construction of u can be bypassed; and

(3) no search is needed (for universal morphisms or factorizations): (a) composition of morphisms, (b) construction of pullbacks, (c) construction of pushouts, and (d) verifying that a morphism is an isomorphism are the only required operations.

Lemma 7.2 (Computing a PBPO$^+$ Rewrite Step). *Let a rule ρ and morphism $\alpha : G_L \to L'$ be given. By using the constructions outlined in the diagram:*

$$
\begin{array}{ccccccccccc}
X & \xrightarrow{\,n\,} & G_L & \xleftarrow{\,g_L\,} & G_K & \xleftarrow{\,v\,} & Y & \xrightarrow{\,v\,} & G_K & & \\
\downarrow{\scriptstyle j} & \text{PB(1)} & \downarrow{\scriptstyle \alpha} & \text{PB(2)} & \downarrow{\scriptstyle u'} & \text{PB(3)} & \downarrow{\scriptstyle i} & & \big\downarrow & & \\
L & \xrightarrow{\,t_L\,} & L' & \xleftarrow{\,l'\,} & K' & \xleftarrow{\,t_K\,} & K & \text{PO(4)} & {\scriptstyle g_R} & & (7.1) \\
& & & & & & \downarrow{\scriptstyle r} & & \big\downarrow & & \\
& & & & & & R & \xrightarrow{\,w\,} & G_R & &
\end{array}
$$

and by additionally verifying that j is an isomorphism, the result of the PBPO$^+$ rewrite step induced by α is computed. Moreover, a PBPO$^+$ rewrite step diagram as given in Definition 3.10 can be reconstructed by defining the missing morphisms $m : L \to G_L$ and $u : K \to G_K$ as follows:

- *$m = n \circ j^{-1}$; and*
- *$u = v \circ i^{-1}$.*

Proof. If j is an isomorphism, it is easy to verify that α defines a strong match for match morphism $m = n \circ j^{-1} : L \to G_L$. Moreover, morphism i is known to be an isomorphism [OER23, Lemma 15]. That the correct pushout is computed in the last step can then be observed as follows. First, using PB(3) we have $t_K = t_K \circ i \circ i^{-1} = u' \circ v \circ i^{-1}$, which means that $v \circ i^{-1}$ is the unique $u : K \to G_K$ for which $t_K = u' \circ u$ [OER23, Lemma 15]. Then in diagram

$$
\begin{array}{ccccc}
 & & \overset{u}{\frown} & & \\
K & \overset{i^{-1}}{\rightarrowtail\!\!\!\rightarrow} & Y & \xrightarrow{\,v\,} & G_K \\
\downarrow{\scriptstyle r} & & \Big\updownarrow{\scriptstyle i} & & \big\downarrow{\scriptstyle g_R} \\
 & \text{PO} & K & \text{PO} & \\
 & & \downarrow{\scriptstyle r} & & \\
\downarrow & & \downarrow & & \big\downarrow \\
R & =\!\!=\!\!= & R & \xrightarrow{\,w\,} & G_R
\end{array}
$$

it is easy to verify that the left square is a pushout. By the dual of the pullback lemma, the pushout of the right square is also the pushout of the outer square. $\qquad\square$

The implementation of the construction described in Lemma 7.2, utilizing the library of `categorytheory` library, is given in Listing 3. Given an adherence $\alpha : G_L \to L'$, it computes

the match $m : L \to G_L$ and the result of the rewrite step G_R. Functions `maybePullback` and `maybePushout` are `Option` types: they can return `None` in categories where not all pullbacks and pushouts are defined.

LISTING 3. `PbpoPlusRule.scala`

```scala
def adherenceInducedStep(alpha: A): Option[(O, A)] =
  require(codomain(alpha) == L1,
    "the codomain of alpha needs to be L'")
  for {
    pb0 <- maybePullback(Cospan(tL, alpha))
    if isIso(pb0.left)
    m = pb0.right o inverseOf(pb0.left).get
    pb1 <- maybePullback(Cospan(alpha, l1))
    u1 = pb1.right
    pb2 <- maybePullback(Cospan(u1, tK))
    u = pb2.left
    iso = pb2.right
    po <- maybePushout(Span(r o iso, u))
    gR = po.sink
  } yield (gR, m)
```

Remark 7.3 (Decidability of the Rewrite Relation). Given a PBPO$^+$ rule ρ in an arbitrary category $\mathbf{C}$, from Lemma 7.2 it follows that the following conditions are sufficient for deciding whether $A \Rightarrow^{\rho}_{\text{PBPO}^+} B$ for objects $A, B \in \text{Ob}(\mathbf{C})$:

(1) for all $X, Y \in \text{Ob}(\mathbf{C})$, the hom-set $\text{Hom}(X, Y)$ is finite; and
(2) pushouts, pullbacks, and morphism compositions are computable.

More specifically, one can use the construction given in Lemma 7.2, where the conditions ensure that

- all adherence morphism candidates $\alpha \in \text{Hom}(A, L')$ can be exhaustively considered;
- verifying whether a morphism $f : X \to Y$ is an isomorphism reduces to a finite search for a morphism $f' \in \text{Hom}(Y, X)$ that satisfies $f \circ f' = 1_Y$ and $f' \circ f = 1_X$;
- verifying whether B qualifies as a pushout object in PO(4) of Diagram (7.1) reduces to a finite search for an isomorphism in $\text{Hom}(G_R, B)$.

For the category of finite sets **FinSet**, the sufficient conditions hold. The same is true for functor categories $[I, \mathbf{FinSet}]$ for which the index category I is finite, which includes many graph categories of interest. This is because here the constructions are computed pointwise in **FinSet**.

7.5. **Package** `termination`. Package `termination`, finally, contains the implementation of the termination method, currently implemented at the level of rm-adhesive quasitoposes. Listing 4 contains a code fragment that illustrates how the implementation utilizes the `categorytheory` library. When given a morphism $f : T \to R'$ (parameter `tiling`) rule ρ (parameter `rule`), it computes set $\{l' \circ \langle f \mid r' \rangle \circ g \mid g \text{ is a right inverse for } \langle r' \mid f \rangle\}$. Then, based on the selected counting class (either $\text{Hom}(\mathbf{C})$, $\text{mono}(\mathbf{C})$, or $\text{rm}(\mathbf{C})$, represented by parameter `countingClass`), it verifies part of the conditions of Theorem 4.23, also utilizing Proposition 4.18.

LISTING 4. Termination.scala

```scala
def slideOptions[O, A](tiling: A, rho: PbpoPlusRule[O, A],
  countingClass: CountingClass)
  (implicit category: TerminationCategory[O, A]): Set[A] =
    import category.{codomain, pullback, rightInversesFor, image, ArrowOps}
    require(codomain(tiling) == rho.R1.get)

    /**
     * L1 <-l1- K1 -r1-> R1
     *           ^          ^
     *           |          |
     *           |    PB   tiling
     *           |          |
     *           X -----> T
     */

    val `K1 <- X -> T` = pullback(Cospan(rho.r1.get, tiling))
    val `K1 <- X`: A = `K1 <- X -> T`.left
    val `X -> T`: A = `K1 <- X -> T`.right
    val `{K1 <- X <- T}`: Set[A] =
      rightInversesFor(`X -> T`).map(`K1 <- X` o _)

    val inClass = countingClass.predicate
    val secondFactor = countingClass.secondFactor

    `{K1 <- X <- T}`.view.filter(f => inClass(rho.l1 o secondFactor(f)))
      .map(rho.l1 o _).toSet
```

8. CONCLUSION AND FUTURE WORK

We have introduced a termination method for graph transformation systems that can be utilized across frameworks, and which is defined in a broad array of categories. Our examples and comparisons with related work show that the method adds considerable value to the study of termination for graph transformation. We have also implemented the method in Scala, together with a REPL with which one can explore relative termination proofs iteratively.

Future work for strengthening the method includes solving the issues raised related to rule equivalence (Example 5.5) and antipatterns (Remark 6.2). Methods for finding $\mathbb{T}$, if it exists, and identifying useful sufficient conditions for the non-existence of $\mathbb{T}$, would also be very useful. A possible metatheoretical direction for future research includes the question posed regarding PBPO$^+$-adhesivity (Remark 4.9).

ACKNOWLEDGMENT

References

[AG00] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sci.*, 236(1-2):133–178, 2000. `doi:10.1016/S0304-3975(99)00207-8`.

[AHS06] J. Adámek, H. Herrlich, and G. E. Strecker. Abstract and concrete categories: The joy of cats. *Reprints in Theory and Applications of Categories*, 17:1–507, 2006. URL: `http://www.tac.mta.ca/tac/reprints/articles/17/tr17.pdf`.

[Awo10] S. Awodey. *Category Theory.* Oxford University Press, 2nd edition, 2010.

[BD86] L. Bachmair and N. Dershowitz. Commutation, transformation, and termination. In *Proc. Conf. on Automated Deduction*, volume 230 of *LNCS*, pages 5–20. Springer, 1986. `doi:10.1007/3-540-16780-3\_76`.

[BHK23] N. Behr, R. Harmer, and J. Krivine. Fundamentals of compositional rewriting theory. *J. Log. Algebraic Methods Program.*, 135:100893, 2023. `doi:10.1016/J.JLAMP.2023.100893`.

[BKNZ15] H. J. S. Bruggink, B. König, D. Nolte, and H. Zantema. Proving termination of graph transformation systems using weighted type graphs over semirings. In *Proc. Conf. on Graph Transformation (ICGT)*, volume 9151 of *LNCS*, pages 52–68. Springer, 2015. `doi:10.1007/978-3-319-21145-9\_4`.

[BKNZ23] H. J. S. Bruggink, B. König, D. Nolte, and H. Zantema. Proving termination of graph transformation systems using weighted type graphs over semirings. *CoRR*, arXiv:1505.01695, 2023. `arXiv:1505.01695`.

[BKZ14] H. J. S. Bruggink, B. König, and H. Zantema. Termination analysis for graph transformation systems. In *Proc. Conf. on Theoretical Computer Science (TCS)*, volume 8705 of *LNCS*, pages 179–194. Springer, 2014. `doi:10.1007/978-3-662-44602-7\_15`.

[Bor94a] F. Borceux. *Handbook of Categorical Algebra 1: Basic Category Theory.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994.

[Bor94b] F. Borceux. *Handbook of Categorical Algebra 3: Categories of Shieves.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994.

[BPHF23] K. Brown, E. Patterson, T. Hanks, and J. P. Fairbanks. Computational category-theoretic rewriting. *J. Log. Algebraic Methods Program.*, 134:100888, 2023. `doi:10.1016/J.JLAMP.2023.100888`.

[BW90] M. Barr and C. Wells. *Category theory for computing science.* Prentice Hall, 1990.

[CDE+19] A. Corradini, D. Duval, R. Echahed, F. Prost, and L. Ribeiro. The PBPO graph transformation approach. *Journal of Logical and Algebraic Methods in Programming*, 103:213–231, 2019. `doi:10.1016/j.jlamp.2018.12.003`.

[CDE+20] A. Corradini, D. Duval, R. Echahed, F. Prost, and L. Ribeiro. Algebraic graph rewriting with controlled embedding. *Theor. Comput. Sci.*, 802:19–37, 2020. `doi:10.1016/j.tcs.2019.06.004`.

[CHHK06] A. Corradini, T. Heindel, F. Hermann, and B. König. Sesqui-Pushout Rewriting. In *Proc. Conf. on Graph Transformation (ICGT)*, volume 4178 of *LNCS*, pages 30–45. Springer, 2006. `doi:10.1007/11841883\_4`.

[CL87] A. B. Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Sci. Comput. Program.*, 9(2):137–159, 1987. `doi:10.1016/0167-6423(87)90030-X`.

[Der82] N. Dershowitz. Orderings for term-rewriting systems. *Theor. Comput. Sci.*, 17:279–301, 1982. `doi:10.1016/0304-3975(82)90026-3`.

[DJ18] N. Dershowitz and J.-P. Jouannaud. Graph path orderings. In *Proc. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*, volume 57 of *EPiC Series in Computing*, pages 307–325. EasyChair, 2018. `doi:10.29007/6hkk`.

[DJ19] N. Dershowitz and J.-P. Jouannaud. Drags: A compositional algebraic framework for graph rewriting. *Theor. Comput. Sci.*, 777:204–231, 2019. `doi:10.1016/j.tcs.2019.01.029`.

[DJO23] N. Dershowitz, J.-P. Jouannaud, and F. Orejas. Drag rewriting. Preprint, hal-04143346v2. `https://inria.hal.science/hal-04143346v2`, 2023.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation.* Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2006. `doi:10.1007/3-540-31188-2`.

[EGH10] H. Ehrig, U. Golas, and F. Hermann. Categorical frameworks for graph transformation and HLR systems based on the DPO approach. *Bull. EATCS*, 102:111–121, 2010.

[EO24] J. Endrullis and R. Overbeek. Generalized weighted type graphs for termination of graph transformation systems. In *Proc. Conf. of Graph Transformation (ICGT)*, volume 14774 of *LNCS*, pages 39–58. Springer, 2024. `doi:10.1007/978-3-031-64285-2_3`.

[EPS73] H. Ehrig, M. Pfender, and H. J. Schneider. Graph-Grammars: An Algebraic Approach. In *Proc. Symp. on on Switching and Automata Theory (SWAT)*, page 167–180. IEEE Computer Society, 1973. `doi:10.1109/SWAT.1973.11`.

[EWZ06] J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. In *Proc. Conf. of Automated Reasoning (IJCAR)*, volume 4130 of *LNCS*, pages 574–588. Springer, 2006. `doi:10.1007/11814771\_47`.

[GL12] R. Garner and S. Lack. On the axioms for adhesive and quasiadhesive categories. *Theory and Applications of Categories*, 27(3):27–46, 2012.

[Hei10] T. Heindel. Hereditary pushouts reconsidered. In *Proc. Conf. on Graph Transformation (ICGT)*, volume 6372 of *LNCS*, pages 250–265. Springer, 2010.

[HMP01] A. Habel, J. Müller, and D. Plump. Double-pushout graph transformation revisited. *Math. Struct. Comput. Sci.*, 11(5):637–688, 2001. `doi:10.1017/S0960129501003425`.

[HPBF20] M. Halter, E. Patterson, A. Baas, and J. P. Fairbanks. Compositional scientific computing with catlab and semanticmodels. *CoRR*, abs/2005.04831, 2020. `arXiv:2005.04831`.

[HW06] D. Hofbauer and J. Waldmann. Termination of string rewriting with matrix interpretations. In *Proc. Conf. of Rewriting Techniques and Applications (RTA)*, volume 4098 of *LNCS*, pages 328–342. Springer, 2006. `doi:10.1007/11805618\_25`.

[JLS07] P. T. Johnstone, S. Lack, and P. Sobocinski. Quasitoposes, quasiadhesive categories and Artin glueing. In *Algebra and Coalgebra in Computer Science (CALCO)*, volume 4624 of *LNCS*, pages 312–326. Springer, 2007. `doi:10.1007/978-3-540-73859-6\_21`.

[Joh02] P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium, Volume 1.* Oxford University Press, 2002.

[Klo87] J. W. Klop. Term rewriting systems: a tutorial. *Bulletin of the European Association for Theoretical Computer Science*, 1987.

[Löw93] M. Löwe. Algebraic approach to single-pushout graph transformation. *Theor. Comput. Sci.*, 109(1&2):181–224, 1993. `doi:10.1016/0304-3975(93)90068-5`.

[LS04] S. Lack and P. Sobociński. Adhesive categories. In *Proc. Conf. on Foundations of Software Science and Computation Structures (FOSSACS)*, volume 2987 of *LNCS*, pages 273–288. Springer, 2004. `doi:10.1007/978-3-540-24727-2\_20`.

[LS05] S. Lack and P. Sobocinski. Adhesive and quasiadhesive categories. *RAIRO Theor. Informatics Appl.*, 39(3):511–545, 2005. `doi:10.1051/ita:2005028`.

[LS06] S. Lack and P. Sobociński. Toposes are adhesive. In *Proc. of Conference on Graph Transformation (ICGT)*, volume 4178 of *LNCS*, pages 184–198. Springer, 2006. `doi:10.1007/11841883\_14`.

[MS10] M. Minas and H. J. Schneider. Graph transformation by computational category theory. In *Graph Transformations and Model-Driven Engineering - Essays Dedicated to Manfred Nagl on the Occasion of his 65th Birthday*, volume 5765 of *LNCS*, pages 33–58. Springer, 2010. `doi:10.1007/978-3-642-17322-6\_3`.

[MS16] G. Moser and M. A. Schett. Kruskal's tree theorem for acyclic term graphs. In *Proc. Workshop on Computing with Terms and Graphs, TERMGRAPH*, volume 225 of *EPTCS*, pages 25–34, 2016. `doi:10.4204/EPTCS.225.5`.

[OE21] R. Overbeek and J. Endrullis. From linear term rewriting to graph rewriting with preservation of termination. In *Proc. Workshop on Graph Computational Models (GCM)*, volume 350 of *EPTCS*, pages 19–34, 2021. `doi:10.4204/EPTCS.350.2`.

[OE23a] R. Overbeek and J. Endrullis. A PBPO$^+$ graph rewriting tutorial. In *Proc. of Workshop on Computing with Terms and Graphs (TERMGRAPH)*, volume 377 of *EPTCS*, pages 45–63. Open Publishing Association, 2023. `doi:10.4204/EPTCS.377.3`.

[OE23b]   R. Overbeek and J. Endrullis. Termination of graph transformation systems using weighted subgraph counting. In *Proc. Conf. of Graph Transformation (ICGT)*, volume 13961 of *LNCS*, pages 81–101. Springer, 2023. `doi:10.1007/978-3-031-36709-0\_5`.

[OE23c]   R. Overbeek and J. Endrullis. Termination of graph transformation systems using weighted subgraph counting. *CoRR*, abs/2303.07812, 2023. `doi:10.48550/arXiv.2303.07812`.

[OE24]   R. Overbeek and J. Endrullis. graphTT: A termination tool for algebraic graph transformation, September 2024. `doi:10.5281/zenodo.13842805`.

[OER23]   R. Overbeek, J. Endrullis, and A. Rosset. Graph rewriting and relabeling with PBPO$^+$: A unifying theory for quasitoposes. *Journal of Logical and Algebraic Methods in Programming*, 2023. `doi:10.1016/j.jlamp.2023.100873`.

[OSVS21]   M. Odersky, L. Spoon, B. Venners, and F. Sommers. *Programming in Scala*. Artima, 5th edition, 2021.

[Ove24]   R. Overbeek. *A Unifying Theory for Graph Transformation*. PhD thesis, Vrije Universiteit Amsterdam, jan 2024. `doi:10.5463/thesis.524`.

[Pie91]   B. C. Pierce. *Basic category theory for computer scientists*. MIT press, 1991. `doi:10.7551/mitpress/1524.001.0001`.

[PLF21]   E. Patterson, O. Lynch, and J. P. Fairbanks. Categorical data structures for technical computing. *CoRR*, abs/2106.04703, 2021. `arXiv:2106.04703`.

[Plu95]   D. Plump. On termination of graph rewriting. In *Proc. Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 1017 of *LNCS*, pages 88–100. Springer, 1995. `doi:10.1007/3-540-60618-1_68`.

[Plu97]   D. Plump. Simplification orders for term graph rewriting. In *Proc. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 1295 of *LNCS*, pages 458–467. Springer, 1997. `doi:10.1007/BFb0029989`.

[Plu98]   D. Plump. Termination of graph rewriting is undecidable. *Fundam. Informaticae*, 33(2):201–209, 1998. `doi:10.3233/FI-1998-33204`.

[Plu99]   D. Plump. Term graph rewriting. *Handbook Of Graph Grammars And Computing By Graph Transformation: Volume 2: Applications, Languages and Tools*, pages 3–61, 1999. URL: `https://www-users.york.ac.uk/~djp10/Papers/tgr_survey.pdf`.

[Plu18]   D. Plump. Modular termination of graph transformation. In *Graph Transformation, Specifications, and Nets*, volume 10800 of *LNCS*, pages 231–244. Springer, 2018. `doi:10.1007/978-3-319-75396-6\_13`.

[RB88]   D. E. Rydeheard and R. M. Burstall. *Computational category theory*. Prentice Hall International Series in Computer Science. Prentice Hall, 1988.

[ROE23]   A. Rosset, R. Overbeek, and J. Endrullis. Fuzzy presheaves are quasitoposes. In *Proc. Conf. of Graph Transformation (ICGT)*, volume 13961 of *LNCS*, pages 102–122. Springer, 2023. `doi:10.1007/978-3-031-36709-0\_6`.

[SZ17]   D. Sabel and H. Zantema. Termination of cycle rewriting by transformation and matrix interpretation. *Log. Methods Comput. Sci.*, 13(1), 2017. `doi:10.23638/LMCS-13(1:11)2017`.

[Ter03]   Terese, editor. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[Wyl91]   O. Wyler. *Lecture Notes on Topoi and Quasitopoi*. World Scientific Publishing Co., 1991.

[Yam22]   A. Yamada. Tuple interpretations for termination of term rewriting. *J. Autom. Reason.*, 66(4):667–688, 2022. `doi:10.1007/S10817-022-09640-4`.

[Zan95]   H. Zantema. Termination of term rewriting by semantic labelling. *Fundam. Informaticae*, 24(1/2):89–105, 1995. `doi:10.3233/FI-1995-24124`.

[ZKB14]   H. Zantema, B. König, and H. J. S. Bruggink. Termination of cycle rewriting. In *Proc. Conf. on Rewriting and Typed Lambda Calculi, RTA-TLCA*, volume 8560 of *LNCS*, pages 476–490. Springer, 2014. `doi:10.1007/978-3-319-08918-8\_33`.