Treewidth-Two Graphs as a Free Algebra

Christian Doczkal, Damien Pous

Treewidth-Two Graphs as a Free Algebra*

Christian Doczkal
Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, France
christian.doczkal@ens-lyon.fr

Damien Pous
Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, France
damien.pous@ens-lyon.fr

Abstract

We give a new and elementary proof that the graphs of treewidth at most two can be seen as a free algebra. This result was originally established through an elaborate analysis of the structure of K_4-free graphs, ultimately reproving the well-known fact that the graphs of treewidth at most two are precisely those excluding K_4 as a minor. Our new proof is based on a confluent and terminating rewriting system for term-labeled graphs and does not involve graph minors anymore. The new strategy is simpler and robust in the sense that it can be adapted to subclasses of treewidth-two graphs, e.g., graphs without self-loops.

2012 ACM Subject Classification Mathematics of computing → Graph theory · Theory of computation → Rewrite systems · Computing methodologies → Symbolic and algebraic manipulation

Keywords and phrases Treewidth, Universal Algebra, Rewriting

Supplement Material `http://perso.ens-lyon.fr/damien.pous/covece/tw2rw`

Funding This work has been funded by the European Research Council (ERC) under the European Union's Horizon 2020 programme (CoVeCe, grant agreement No 678157). This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

1 Introduction

The notion of *treewidth* [9] is a cornerstone of algorithmic graph theory and parameterised complexity: treewidth measures how close a graph is to a forest, and many problems that can be solved in polynomial time on forests but are NP-complete on arbitrary graphs remain polynomial on classes of graphs of bounded treewidth. This is the case for instance for the graph homomorphism problem (and thus k-coloring) [13, 5, 14].

Similar to trees, graphs of bounded treewidth can be described by a variety of syntaxes [8]. Among the open problems, there is the question, for graphs of a given treewidth, of finding a syntax making it possible to get a finite and equational axiomatisation of graph isomorphism [8, page 118]. This question was recently answered positively for directed multigraphs of treewidth at most two [7].

The syntax used in [7] is comprised of two binary operations: *series* and *parallel* composition [12], their neutral elements, and a unary *converse* operation. In this syntax, several terms may denote the same graph (up-to isomorphism); the key result of [7] is that the corre-

* Extended version of the paper that appeared in Proc. MFCS 2018 [11], with additional proofs.

sponding equational theory is characterized by twelve equational axioms, defining so-called 2p-algebras.

To get this result, the authors define a function $\mathtt{t}$ from graphs to terms and establish that $\mathtt{t}$ is a isomorphism of 2p-algebras. The function $\mathtt{t}$ is defined using an elaborate analysis of the structure of treewidth-two graphs, which requires complicated graph-theoretical arguments that are not directly related to the proposed axiom system. For instance they ultimately reprove the well-known fact that the graphs of treewidth at most two are precisely those graphs excluding K_4 (the complete graph with four vertices) as a minor [12]. The authors also make $\mathtt{t}$ as canonical as possible in order to facilitate the proof that on isomorphic graphs, $\mathtt{t}$ returns terms that are congruent modulo the axioms. This comes at the price of complicating the proofs that $\mathtt{t}$ is a homomorphism of 2p-algebras.

In the present paper, we reprove the result from [7] using a completely different approach inspired by [3]: instead of using an elaborate top-down analysis, we design a graph rewriting system on term-labeled graphs and use it to reduce graphs, in a bottom-up fashion, to a shape where a term can be read off. This process is highly nondeterministic but can be shown confluent modulo the axioms. This results in big simplifications: tree decompositions are only used to show that all treewidth-two graphs can be reduced to the point where a term can be read off, and minors are not used at all in this new approach.

Another important feature of this new proof is that it makes it possible to discover the required axioms almost automatically, mainly during the confluence proof. It is also more robust: it allows us to solve two problems left open in [7], characterizing connected graphs as a free-algebra, and characterizing self-loop free graphs as a free-algebra, in both cases for graphs of treewidth at most two.

The first problem was solved recently [16] using a purely model-theoretic argument: 2p-algebras form a conservative extension of *2pdom-algebras*, the counterpart of 2p-algebras for connected graphs. Our strategy makes it possible to proceed the other way around: we prove the main result for connected graphs and 2pdom-algebras (Sections 3 to 5), before extending it to potentially disconnected graphs and 2p-algebras using a simple and mainly algebraic argument (Section 6).

The second problem was still open. We solve it using a slight variation of the presented proof, which actually leads us to the discovery of the required axioms (Section 7).

2 Preliminaries: 2p- and 2pdom-algebras

We recall the definitions of 2p- and 2pdom-algebras [7, 16]. We let $a, b \dots$ range over the *letters* of a fixed alphabet $\mathcal{A}$. We consider labeled directed graphs with two designated vertices. We just call them graphs in the sequel.

▶ **Definition 1.** A *graph* is a tuple $G = \langle V, E, s, t, l, \iota, o\rangle$, where V is a finite set of *vertices*, E is a finite set of *edges*, $s, t : E \to V$ are maps indicating the *source* and *target* of each edge, $l : E \to \mathcal{A}$ is a map indicating the *label* of each edge, and $\iota, o \in V$ are the designated vertices, respectively called *input* and *output*.

Note that we allow multiple edges between two vertices, as well as self-loops.

▶ **Definition 2.** A *homomorphism* from $G = \langle V, E, s, t, l, \iota, o\rangle$ to $G' = \langle V', E', s', t', l', \iota', o'\rangle$ is a pair $h = \langle f, g\rangle$ of functions $f : V \to V'$ and $g : E \to E'$ that respect the various components: $s' \circ g = f \circ s$, $t' \circ g = f \circ t$, $l = l' \circ g$, $\iota' = f(\iota)$, and $o' = f(o)$.

A *(graph) isomorphism* is a homomorphism whose two components are bijective functions. We write $G \simeq G'$ when there exists an isomorphism between graphs G and G'.

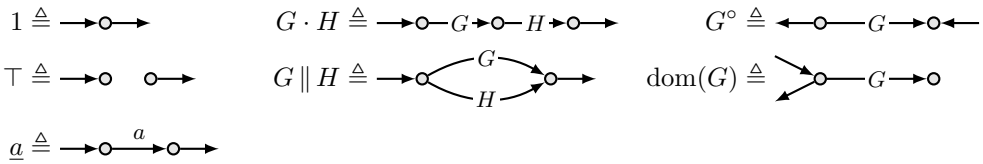


Figure 1 Graph operations

$$u \parallel (v \parallel w) = (u \parallel v) \parallel w \quad \text{(A1)}$$
$$u \parallel v = v \parallel u \quad \text{(A2)}$$
$$u \cdot (v \cdot w) = (u \cdot v) \cdot w \quad \text{(A4)}$$
$$u \cdot 1 = u \quad \text{(A5)}$$
$$u^{\circ\circ} = u \quad \text{(A6)}$$
$$(u \parallel v)^\circ = u^\circ \parallel v^\circ \quad \text{(A7)}$$
$$(u \cdot v)^\circ = v^\circ \cdot u^\circ \quad \text{(A8)}$$
$$1 \parallel 1 = 1 \quad \text{(A9)}$$
$$\mathrm{dom}(u \parallel v) = 1 \parallel u \cdot v^\circ \quad \text{(A10)}$$
$$u \parallel \top = u \quad \text{(A3)}$$
$$u \cdot \top = \mathrm{dom}(u) \cdot \top \quad \text{(A11)}$$
$$(1 \parallel u) \cdot v = (1 \parallel u) \cdot \top \parallel v \quad \text{(A12)}$$
$$\mathrm{dom}(u \cdot v) = \mathrm{dom}(u \cdot \mathrm{dom}(v)) \quad \text{(A13)}$$
$$\mathrm{dom}(u) \cdot (v \parallel w) = \mathrm{dom}(u) \cdot v \parallel w \quad \text{(A14)}$$

Figure 2 Axioms of 2p-algebras (A1-A12) and 2pdom-algebras (A1,A2,A4-A10,A13,A14)

We consider the following signatures for terms and algebras:

$$\Sigma = \{\cdot_2, \parallel_2, \_^\circ_1, 1_0\} \qquad \Sigma_\top = \Sigma \cup \{\top_0\} \qquad \Sigma_{\mathrm{dom}} = \Sigma \cup \{\mathrm{dom}_1\}$$

We usually omit the $\cdot$ symbol and we assign priorities so that the term $(a \cdot (b^\circ)) \parallel c$ can be written just as $ab^\circ \parallel c$.

Graphs form algebras for those signatures by considering the operations depicted in Figure 1, where input and outputs are represented by unlabelled ingoing and outgoing arrows. The binary operations $(\cdot)$ and $(\parallel)$ respectively correspond to series and parallel composition, converse $(\_^\circ)$ just exchanges input and output, and *domain* $(\mathrm{dom}(\_))$ relocates the output to the input.

A graph is called a *test* if its input and output coincide. The parallel composition of a graph with a test merges the input and output of the former graph. For instance, the graph $\underline{a} \parallel 1$ consists of a single vertex with a self-loop labeled with a. Also note that the graph $\mathrm{dom}(G)$ is isomorphic to the graph $G \cdot \top \parallel 1$. For $\Sigma_\top$-terms, we will therefore consider $\mathrm{dom}(u)$ to be an abbreviation for $u\top \parallel 1$.

▶ **Definition 3.** A 2p-algebra is a $\Sigma_\top$-algebra satisfying axioms A1-A12 from Figure 2. A 2pdom-algebra is a Σ_{dom}-algebra satisfying axioms A1,A2,A4-A10,A13,A14 from Figure 2.

▶ **Lemma 4.** *Every 2p-algebra is a 2pdom-algebra (with* $\mathrm{dom}(u) \triangleq u\top \parallel 1$*).*

Proof. This easy result is implicitly proved in [16]; Coq proofs scripts are available [10]. ◀

▶ **Proposition 5.** *Graphs (up to isomorphism) form a 2p-algebra.*

▶ **Proposition 6.** *Connected graphs form a subalgebra of the* Σ_{dom}*-algebra of graphs.*

Given $\Sigma_\top$-terms u, v with variables in $\mathcal{A}$, we write $2p \vdash u = v$ when the equation is derivable from the axioms of 2p-algebra (equivalently, when the equation universally holds in all 2p-algebras). Similarly for Σ_{dom}-terms and 2pdom-algebras.

By interpreting a letter $a \in \mathcal{A}$ as the graph $\underline{a}$ in Figure 1, we can associate a graph $\mathsf{g}(u)$ to every term over the considered signatures. By Proposition 5, $2p \vdash u = v$ entails $\mathsf{g}(u) \simeq \mathsf{g}(v)$ for all $\Sigma_\top$-terms u, v and similarly for Σ_{dom}-terms and $2pdom$-algebras (using Lemma 4).

▶ **Definition 7.** A $\Sigma_\top$-term u is called a *test* if $2p \vdash u \parallel 1 = u$. A Σ_{dom}-term u is called a *test* if $2pdom \vdash u \parallel 1 = u$. We write $\mathcal{T}$ for the set of tests and $\mathcal{N}$ for the set of non-tests. We let α, β, and γ range over terms that are tests.

Thanks to converse being an involution, there is a notion of duality in 2p-algebras: a valid law remains so when swapping the arguments of products and replacing $\mathrm{dom}(u)$ with $\mathrm{dom}(u^\circ)$.

▶ **Lemma 8.** *The following laws hold in all 2pdom-algebras.*

1. $\mathrm{dom}(u) \parallel 1 = \mathrm{dom}(u)$ *(*$\mathrm{dom}(u)$ *is a test)*
2. $\alpha^\circ = \alpha$
3. $\alpha\beta = \alpha \parallel \beta = \beta\alpha$
4. $(u \parallel v)\alpha = u \parallel v\alpha$

Proof. As is standard for involutive monoids, we have $1^\circ = 1$ and, therefore, $\mathrm{dom}(\alpha) = \alpha$ by (A10). We then reason as follows:

1. $\mathrm{dom}(u) = \mathrm{dom}(u)(1 \parallel 1) = \mathrm{dom}(u)1 \parallel 1 = \mathrm{dom}(u) \parallel 1$
2. $\alpha^\circ = (\alpha \parallel 1)^\circ = 1 \parallel 1\alpha^\circ = \mathrm{dom}(1 \parallel \alpha) = \mathrm{dom}(\alpha) = \alpha$
3. By commutativity of $\parallel$, it suffices to prove the first equation (using Claim (4)): $\alpha\beta = (\alpha \parallel 1)\beta = \alpha \parallel 1\beta = \alpha \parallel \beta$
4. We show the dual: $\alpha(u \parallel v) = \mathrm{dom}(\alpha)(u \parallel v) = \mathrm{dom}(\alpha)u \parallel v = \alpha u \parallel v$ ◀

▶ Remark. Lemma 8 is implicitly proved in [16], and Coq proofs scripts are available [10]. We nevertheless provide a proof here for the sake of completeness. Also note that similar facts were already proven for 2p-algebra in [7] but that we need proofs in 2pdom-algebras here.

▶ **Lemma 9** ([7, Proposition 1])**.** *The following laws hold in all 2p-algebras*

1. $u\top v \parallel \top w\top = u\top w\top v$
2. $uv \parallel \top w\top = (u \parallel \top w\top)v$
3. $\top u^\circ\top = \top u\top$
4. $\alpha\top\beta \parallel u = \alpha u\beta$

▶ **Lemma 10.** *A* Σ_{dom}*- or* $\Sigma_\top$*-term* u *is a test iff* $\mathsf{g}(u)$ *is a test.*

Proof. The direction from left to right follows with Proposition 5. The converse direction follows by induction on u using the lemmas above. We show the case for uv. If $\mathsf{g}(uv)$ is test, then both $\mathsf{g}(u)$ and $\mathsf{g}(v)$ must be tests and we have $2pdom \vdash u \parallel 1$ and $2pdom \vdash v \parallel 1 = v$. Thus $2pdom \vdash uv \parallel 1 = (u \parallel 1)(v \parallel 1) \parallel 1 = (u \parallel 1) \parallel (v \parallel 1) \parallel 1 = (u \parallel 1) \parallel (v \parallel 1) = uv$ ◀

One useful consequence of the lemma above is that uv is a test iff both u and v are tests and $u \parallel v$ is test if either u or v is a test. Further, $\mathcal{A} \subseteq \mathcal{N}$, i.e., letters are non-tests.

We conclude this preliminary section by defining the subalgebra of treewidth-two graphs.

▶ **Definition 11.** A *simple graph* is a pair $\langle V, R\rangle$ consisting of a finite set V of vertices and an irreflexive and symmetric binary relation R on V. The *skeleton* of a graph G is the simple graph obtained from G by forgetting input, output, labeling, self loops, and edge directions and multiplicities. The *strong skeleton* of a graph is the skeleton of G with an additional edge connecting ι and o.

▶ **Definition 12** ([9])**.** Let G be a simple graph. A *tree decomposition* of G is a tree T where each node $t \in T$ is labeled with a set of vertices B_t such that:

1. For every vertex x of G, the set of nodes t such that $x \in B_t$ is nonempty and connected in T (i.e., forms a subtree)
2. For every xy-edge, there exists some t such that $\{x, y\} \subseteq B_t$.

The *width* of a tree decomposition is the size of its largest set B_t minus one, and the *treewidth* of a graph is the minimal width of a tree decomposition for this graph. The simple graphs of treewidth at most one are the forests. We write TW_2 for the collection of graphs whose strong skeleton has treewidth at most two.

▶ **Proposition 13** ([7]). *TW_2 forms a subalgebra of the $\Sigma_\top$-algebra of graphs.*

▶ **Corollary 14.** *For every term u, $\mathsf{g}(u) \in \mathsf{TW}_2$.*

The main results about 2p- and 2pdom-algebras, which we reprove in this paper, are that TW_2 (up to isomorphism) forms the free 2p-algebra (over $\mathcal{A}$) [7] and that the connected graphs in TW_2 form the free 2pdom-algebra [16]. As explained in the introduction, we start with the connected case, which we then extend to deal with disconnected graphs.

3 A Confluent Rewriting System for Term-labeled Graphs

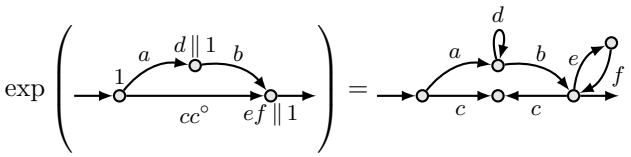
The rewriting system we define to extract terms from graphs works on a generalised form of graphs, whose edges are labeled by terms rather than just letters, and whose vertices are labeled by tests.

We work exclusively with Σ_{dom}-terms and connected graphs in Sections 3 to 5; for these sections we thus abbreviate $2pdom \vdash u = v$ as $u \equiv v$.

▶ **Definition 15.** A *term-labeled graph* is a tuple $G = \langle V, E, s, t, l, \iota, o\rangle$ that is a graph except that l is a function from $V \uplus E$ to Σ_{dom}-terms (we assume V and E to always be disjoint) such that $l(x) \in \mathcal{T}$ for vertices x and $l(e) \in \mathcal{N}$ for edges e. We write $\exp(G)$ for the *expansion* of G obtained by replacing every edge e with $\mathsf{g}(l(e))$ and every vertex x with $\mathsf{g}(l(x))$.

Restricting edge labels to non-tests ensures that replacing edges by the graphs described by their labels does not collapse source and target of the edge. Similarly, replacing vertices by graphs is only meaningful if the replacement is a test.

▶ **Example 16.** A term-labeled graph and its expansion:



We will compare term-labeled graphs using a notion of isomorphism where labels are compared modulo 2pdom-axioms. A subtlety here is that we should consider as equivalent two graphs where one is obtained from the other by reversing a u-labeled edge and labeling it with u° (this operation preserves the expansion). The following predicate, which we use in the definitions below, captures this idea in a formal way: $L(x, y, e, u)$ means that e can be seen as a u-labeled xy-edge, up to $\equiv$.

$$L(x, y, e, u) \triangleq (s(e) = x \wedge t(e) = y \wedge l(e) \equiv u) \vee (s(e) = y \wedge t(e) = x \wedge l(e) \equiv u^\circ)$$

▶ **Definition 17.** Two term-labeled graphs $G = \langle V, E, s, t, l, \iota, o\rangle$ and $H = \langle V', E', s', t', l', \iota', o'\rangle$ are *weakly isomorphic*, written $G \cong H$, if there is a pair of bijective functions $\langle f, g\rangle$ satisfying

1. $f(\iota) = \iota'$ and $f(o) = o'$.

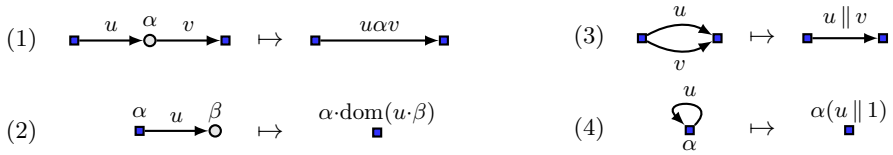
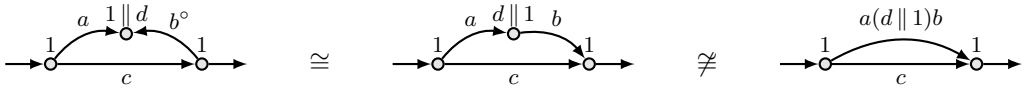


Figure 3 Rewriting system for term-labeled graphs. The square vertices may have additional incident edges. The circular vertices (i.e., those that are removed) must be distinct from input and output and may not have other incident edges.

2. For all vertices $x \in V$, $l(x) \equiv l'(f(x))$.
3. For all edges $e \in E$ and $e' \in E'$ such that $g(e) = e'$, $L(s'(e'), t'(e'), e', l(e))$.

▶ **Example 18.** Weakly isomorphic graphs always have isomorphic expansions. However, the converse is not true: all three graphs below have isomorphic expansions, but only the first two are weakly isomorphic.



We now define the rewriting system on term-labeled graphs, as depicted in Figure 3.

▶ **Definition 19.** Let $G = \langle V, E, s, t, l, \iota, o\rangle$ be a term-labeled graph. We write $G \mapsto G'$ if G' can be obtained from G by applying one of the following rules.

1. If $l(x) = \alpha$, $L(x_1, x, e_1, u)$ and $L(x, x_2, e_2, v)$ where $x \notin \{\iota, o, x_1, x_2\}$ and e_1 and e_2 are the only incident edges of x, then replace e_1 and e_2 with an $u\alpha v$-labeled edge from x_1 to x_2 and remove x.
2. If $l(x) = \alpha$, $l(y) = \beta$ and $L(x, y, e, u)$ where $y \notin \{\iota, o\}$ and e is the only edge incident to y, then change the label of x to $\alpha{\cdot}\mathrm{dom}(u{\cdot}\beta)$ and remove y and e.
3. If $L(x, y, e_1, u)$ and $L(x, y, e_2, v)$ then replace e_1 and e_2 with a $(u \parallel v)$-labeled xy-edge.
4. If $s(e) = t(e) = x$, $l(x) = \alpha$ and $l(e) = u$, then assign label $\alpha(u \parallel 1)$ to x and remove e.

It is straightforward to verify that $\mapsto$ preserves the requirements on edge and vertex labels from Definition 15. We write $\rightsquigarrow$ for the reflexive transitive closure of $\mapsto$ up to $\cong$ (i.e., $G \rightsquigarrow H$ iff either $G \cong H$ or there exists a sequence $G \cong G_1 \mapsto G_2 \cong G_3 \mapsto \ldots \mapsto G_n \cong H$).

▶ **Lemma 20.** *The relation $\mapsto$ is terminating.*

▶ **Lemma 21.** *If $G \mapsto G'$, then $\exp(G) \simeq \exp(G')$.*

▶ **Lemma 22.** *If $G \cong H$ and $G \mapsto G'$, then there exists H' such that $H \mapsto H'$ and $G' \cong H'$.*

We now show that the relation $\mapsto$ is locally confluent up to weak isomorphism. The proof is fundamental: while closing the various critical pairs, we rediscover most of the axioms of 2pdom-algebras. Note that for rules (1) and (3) we do *not* assume that the square vertices are distinct. This introduces some critical pairs (e.g, between rules 3 and 4), but ensures that reductions are preserved in contexts that collapse input and output (Lemma 30 below).

▶ **Lemma 23** (Local Confluence). *If $G_1 \leftarrow\!\shortmid G \mapsto G_2$, then there exist G_1' and G_2' such that $G_1 \mapsto G_1'$, $G_2 \mapsto G_2'$ and $G_1' \cong G_2'$.*

Proof. If the redexes do not overlap, we can reduce G_1 and G_2 to the same graph in one step. It remains to analyze the critical pairs. The nontrivial interactions are as follows:

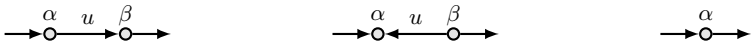
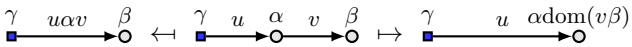
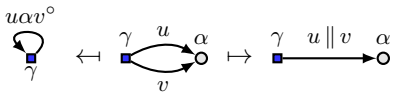


Figure 4 Atomic Graphs

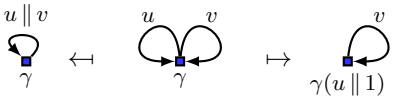
- Rules 1 and 2 can interact as follows:



  After applying rule 2 on both sides, it suffices to show $\mathrm{dom}(u\alpha v\beta) \equiv \mathrm{dom}(u\alpha\mathrm{dom}(v\beta))$, which is an instance of (A13).
- Rules 1 and 3 can interact as follows:



  After applying rule 4 on the left and rule 2 on the right, it suffices to show that we have $u\alpha v^\circ \parallel 1 \equiv \mathrm{dom}((u \parallel v)\alpha)$. We prove it as follows using Lemma 8(4) and (A10): $\mathrm{dom}((u \parallel v)\alpha) \equiv \mathrm{dom}(u \parallel v\alpha) \equiv 1 \parallel u(v\alpha)^\circ \equiv u\alpha v^\circ \parallel 1$.
- Rules 3 and 4 can interact as follows:



  After applying rule 4 on both sides, it suffices to show $u \parallel v \parallel 1 \equiv (u \parallel 1)(v \parallel 1)$. Since $v \parallel 1$ is a test, this follows with Lemma 8(4) and (A5).

There are a number of other critical pairs that can easily be resolved using (A1)-(A8) (e.g., two overlapping instances of rule 1 that differ in the direction the edges are matched, or overlapping instances of rule 3). Similarly, overlapping instances of rules 2 and 4 remain instances after the first rule has been applied and the resulting graphs only differ in the order of the tests being generated; thus they are weakly isomorphic by Lemma 8(3). ◀

▶ **Proposition 24** (Confluence). *If $G_1 \leftsquigarrow G \rightsquigarrow G_2$, then there exists H such that $G_1 \rightsquigarrow H \leftsquigarrow G_2$.*

▶ **Definition 25.** We call a term-labeled graph *atomic* if it consists of either a single vertex and no edges or two vertices connected by a single edge as depicted in Figure 4. If A is atomic, we write $\underline{A}$ for the term that can extracted from A, i.e., $\alpha u \beta$, $\alpha u^\circ \beta$, or α for the atoms in Figure 4, from left to right.

▶ **Lemma 26.** *If $A \leftsquigarrow G \rightsquigarrow B$ for some atomic graphs A, B, then $\underline{A} \equiv \underline{B}$.*

Proof. We have $A \cong B$ by Proposition 24, since atomic graphs are irreducible. The claim then follows by case analysis on A and B. ◀

4 Reducibility of Term-Graphs

We now show that the rewriting system from the previous section can be used to reduce graphs of the shape $\mathsf{g}(u)$ to atomic graphs. As a consequence, we obtain that $u \equiv v$ iff $\mathsf{g}(u) \simeq \mathsf{g}(v)$ and, hence, that equivalence of Σ_{dom}-terms is decidable.

To show that $\mathsf{g}(u)$ reduces to an atomic graph, we define a function computing for every u an equivalent term that can be obtained as $\underline{A}$ for some atomic graph A. In particular, if u is not a test, it computes "maximal" tests α and β and a non-test v such that $u \equiv \alpha v \beta$.

▶ **Definition 27.** We define a function f from Σ_{dom}-terms to $\mathcal{T} \cup \mathcal{T} \times \mathcal{N} \times \mathcal{T}$ as depicted in Figure 5, as well as functions $\lceil \_ \rceil$ and $\lfloor \_ \rfloor$ interpreting elements of $\mathcal{T} \cup \mathcal{T} \times \mathcal{N} \times \mathcal{T}$ as atomic

$$
\begin{aligned}
\mathsf{f}(u^\circ) = {} & \mathsf{match}\ \mathsf{f}(u)\ \mathsf{with} && \mathsf{f}(1) = (1)\\
& (\alpha_u, u', \beta_u) \Rightarrow (\beta_u, u'^\circ, \alpha_u)\\
& (\gamma) \Rightarrow (\gamma)\\
\mathsf{f}(u \parallel v) = {} & \mathsf{match}\ \mathsf{f}(u), \mathsf{f}(v)\ \mathsf{with} && \mathsf{f}(a) = (1, a, 1)\\
& (\alpha_u, u', \beta_u), (\alpha_v, v', \beta_v) \Rightarrow (\alpha_u\alpha_v, u' \parallel v', \beta_u\beta_v)\\
& (\alpha_u, u', \beta_u), (\gamma) \Rightarrow (\alpha_u u'\beta_u \parallel \gamma)\\
& (\gamma), (\alpha_v, v', \beta_v) \Rightarrow (\gamma \parallel \alpha_v v'\beta_v)\\
& (\gamma_1), (\gamma_2) \Rightarrow (\gamma_1\gamma_2)\\
\mathsf{f}(u{\cdot}v) = {} & \mathsf{match}\ \mathsf{f}(u), \mathsf{f}(v)\ \mathsf{with} && \mathsf{f}(\mathrm{dom}(u)) = (\mathrm{dom}(u))\\
& (\alpha_u, u', \beta_u), (\alpha_v, v', \beta_v) \Rightarrow (\alpha_u, u'\beta_u\alpha_v v', \beta_v)\\
& (\alpha_u, u', \beta_u), (\gamma) \Rightarrow (\alpha_u, u', \beta_u\gamma)\\
& (\gamma), (\alpha_v, v', \beta_v) \Rightarrow (\gamma\alpha_v, v', \beta_v)\\
& (\gamma_1), (\gamma_2) \Rightarrow (\gamma_1\gamma_2)
\end{aligned}
$$

Figure 5 Test analysis for Σ_{dom}-terms

$$
\begin{array}{ccc}
\Sigma_{\mathrm{dom}}\text{-terms} & \xrightarrow{\quad\mathsf{g}\quad} & \text{graphs}\\
{\scriptstyle\lfloor\_\rfloor}\uparrow\downarrow{\scriptstyle\mathsf{f}} & & \exp(\_)\uparrow\downarrow\\
\mathcal{T} \cup \mathcal{T} \times \mathcal{N} \times \mathcal{T} & \xrightarrow{\quad\lceil\_\rceil\quad} & \text{term-labeled graphs}
\end{array}
$$

Figure 6 Summary of functions between terms and graphs

graphs and terms, respectively: $\lceil(\alpha, u, \beta)\rceil$ is the graph on the left in Figure 4 and $\lceil(\alpha)\rceil$ is the graph on the right, $\lfloor(\alpha, u, \beta)\rfloor \triangleq \alpha u \beta$, and $\lfloor(\gamma)\rfloor \triangleq \gamma$. Note that $\underline{\lceil(\alpha, u, \beta)\rceil} = \lfloor(\alpha, u, \beta)\rfloor$.

A summary of the functions defined so far is given in Figure 6.

▶ **Lemma 28.** $u \equiv \lfloor \mathsf{f}(u) \rfloor$.

Proof. By induction on u. The cases for a, 1, and $\mathrm{dom}(u)$ are trivial. The case for u° follows with Lemma 8(2). We show the case for $\mathsf{f}(u \parallel v)$ where $\mathsf{f}(u) = (\alpha_u, u', \beta_u)$ and $\mathsf{f}(v) = (\alpha_v, v', \beta_v)$.

$$
\begin{aligned}
\lfloor \mathsf{f}(u \parallel v) \rfloor &= \alpha_u\alpha_v(u' \parallel v')\beta_u\beta_v\\
&\equiv (\alpha_u u'\beta_u) \parallel (\alpha_v v'\beta_v) && \text{Lemma 8(4) and commutativity of } \parallel\\
&\equiv \lfloor \mathsf{f}(u) \rfloor \parallel \lfloor \mathsf{f}(v) \rfloor\\
&\equiv u \parallel v && \text{induction hypothesis}
\end{aligned}
$$

The remaining cases are straightforward. ◀

We now show that $\mathsf{g}(u)$ (seen as a term-labeled graph) reduces to $\lceil \mathsf{f}(u) \rceil$. To do so, we first extend the graph operations to term-labeled graphs and prove the context lemma below.

▶ **Definition 29.** If a graph G occurs as a term-labeled graph, it is to be read as the graph where every vertex is labeled with 1 (and every edge is labeled with a single letter as before).

We extend the operations $\cdot$, $\|$ and $\mathrm{dom}(\_)$ to term-labeled graphs. If two vertices x and y are identified by an operation, we label the resulting vertex with $l(x)\cdot l(y)$.

▶ **Lemma 30** (Context Lemma). *If $G \rightsquigarrow G'$, then $G \,\|\, H \rightsquigarrow G' \,\|\, H$, $G{\cdot}H \rightsquigarrow G'{\cdot}H$, $H{\cdot}G \rightsquigarrow H{\cdot}G'$, and $\mathrm{dom}(G) \rightsquigarrow \mathrm{dom}(G')$.*

Proof. By induction on $G \rightsquigarrow G'$. First, all operations preserve weak isomorphisms. Second, a redex in G is still a redex in $G{\cdot}H$, $H{\cdot}G$, and $\mathrm{dom}(G)$ since G remains unchanged except that one of its nodes may cease to be input or output. Similarly, a redex in G is also a redex in $G \,\|\, H$ (even if H is a test: we do not require the square vertices in rules 1 and 3 to be distinct so that redexes are preserved under collapsing input and output). ◀

Note that the converse of Lemma 30 does not hold. For instance, $\mathrm{dom}(\underline{a})$ (cf. Figure 1) reduces by rule 2 to a graph G with a single node labeled $1\mathrm{dom}(a1)$. Hence, $\mathrm{dom}(\underline{a}) \rightsquigarrow \mathrm{dom}(G)$ since $G \cong \mathrm{dom}(G)$, but $\underline{a}$ is an atom and thus irreducible.

▶ **Proposition 31** (Reducibility). $\mathsf{g}(u) \rightsquigarrow \lceil \mathsf{f}(u) \rceil$

Proof. The proof proceeds by induction on u.
case $u \,\|\, v$: We first reduce $\mathsf{g}(u)$ and $\mathsf{g}(v)$ using the context lemma.

$$\begin{aligned} \mathsf{g}(u \,\|\, v) &= \mathsf{g}(u) \,\|\, \mathsf{g}(v) && \text{def. } \mathsf{g} \\ &\rightsquigarrow \lceil \mathsf{f}(u) \rceil \,\|\, \mathsf{g}(v) && \text{IH, Lemma 30} \\ &\rightsquigarrow \lceil \mathsf{f}(u) \rceil \,\|\, \lceil \mathsf{f}(v) \rceil && \text{IH, Lemma 30, comm. } \| \end{aligned}$$

We consider several cases.

- If $\mathsf{f}(u) = (\gamma)$ and $\mathsf{f}(v) = (\alpha_v, v', \beta_v)$, then $\lceil \mathsf{f}(u) \rceil \,\|\, \lceil \mathsf{f}(v) \rceil$ has a single node labeled $\gamma\alpha_v\beta_v$ (up to commutativity of tests) and a single self loop labeled v'. Hence, we can apply rule 4:

  $$\lceil \mathsf{f}(u) \rceil \,\|\, \lceil \mathsf{f}(v) \rceil \rightsquigarrow \lceil (\gamma\alpha_v\beta_v(v' \,\|\, 1)) \rceil \cong \lceil (\gamma \,\|\, \alpha_v v' \beta_v) \rceil \cong \lceil \mathsf{f}(u \,\|\, v) \rceil$$

- If $\mathsf{f}(u) = (\alpha_u, u', \beta_u)$ and $\mathsf{f}(v) = (\alpha_v, v', \beta_v)$ then $\lceil \mathsf{f}(u) \rceil \,\|\, \lceil \mathsf{f}(v) \rceil$ is an instance of rule 3 and we have:

  $$\lceil \mathsf{f}(u) \rceil \,\|\, \lceil \mathsf{f}(v) \rceil \rightsquigarrow \lceil (\alpha_u\alpha_v, u' \,\|\, v', \beta_u\beta_v) \rceil \cong \lceil \mathsf{f}(u \,\|\, v) \rceil$$

The remaining cases are analogous.
case $u{\cdot}v$:

$$\begin{aligned} \mathsf{g}(u{\cdot}v) &= \mathsf{g}(u){\cdot}\mathsf{g}(v) && \text{def. } \mathsf{g} \\ &\cong \lceil \mathsf{f}(u) \rceil{\cdot}\lceil \mathsf{f}(v) \rceil && \text{IH, Lemma 30 (twice)} \end{aligned}$$

If $\mathsf{f}(u) = (\alpha_u, u', \beta_u)$ and $\mathsf{f}(v) = (\alpha_v, v', \beta_v)$ then $\lceil \mathsf{f}(u) \rceil \cdot \lceil \mathsf{f}(v) \rceil$ is an instance of rule 1 with the intermediate vertex labeled $\beta_u\alpha_v$. Hence,

$$\lceil \mathsf{f}(u) \rceil{\cdot}\lceil \mathsf{f}(v) \rceil \rightsquigarrow \lceil (\alpha_u, u'\beta_u\alpha_v v', \beta_v) \rceil \cong \lceil \mathsf{f}(u{\cdot}v) \rceil$$

The cases where either $\mathsf{f}(u) = (\gamma)$ or $\mathsf{f}(v) = (\gamma')$ are straightforward.
case $\mathrm{dom}(u)$:

$$\begin{aligned} \mathsf{g}(\mathrm{dom}(u)) &= \mathrm{dom}(\mathsf{g}(u)) && \text{def. } \mathsf{g} \\ &\rightsquigarrow \mathrm{dom}(\lceil \mathsf{f}(u) \rceil) && \text{IH, Lemma 30} \end{aligned}$$

If $\mathsf{f}(u) = (\gamma)$, there is nothing to show. Otherwise we have $\mathsf{f}(u) = (\alpha_u, u', \beta_u)$ and therefore $\mathrm{dom}(\lceil \mathsf{f}(u) \rceil)$ is an instance of rule 1:

$$\mathrm{dom}(\lceil \mathsf{f}(u) \rceil) \rightsquigarrow \lceil (\alpha_u \mathrm{dom}(u' \beta_u)) \rceil \cong \lceil \mathsf{f}(\mathrm{dom}(u)) \rceil$$

The cases for a and 1 are straightforward (no reduction required). ◀

We can finally characterise the equational theory of 2pdom-algebras:

▶ **Theorem 32.** *$2pdom \vdash u = v$ iff $\mathsf{g}(u) \simeq \mathsf{g}(v)$.*

Proof. The direction from left to right follows with Proposition 5. For the converse direction, assume $\mathsf{g}(u) \simeq \mathsf{g}(v)$. Then $\mathsf{g}(u) \cong \mathsf{g}(v)$ and therefore $\underline{\lceil \mathsf{f}(u) \rceil} \equiv \underline{\lceil \mathsf{f}(v) \rceil}$ by Proposition 31 and Lemma 26. Hence, $u \equiv \lfloor \mathsf{f}(u) \rfloor = \underline{\lceil \mathsf{f}(u) \rceil} \equiv \underline{\lceil \mathsf{f}(v) \rceil} = \lfloor \mathsf{f}(v) \rfloor \equiv v$ using Lemma 28 twice. ◀

As explained in the introduction we did not use minors to obtain this result. Actually, we did not use tree decompositions either: those arise only in the following section, where we need to characterize the image of the function g. This sharply contrasts with the approach from [7], where both tree decompositions and minors are used to obtain the above characterization.

5 The free 2pdom-algebra

In order to show that the connected graphs in TW_2 form the free 2p-algebra, it remains to obtain an inverse to g (up to $\equiv$), i.e., we need to extract terms from such graphs. We again make use of the rewriting system.

In a slight abuse of notation, we also write $G \in \mathsf{TW}_2$ to denote that (the strong skeleton of) a term-labeled graph G has treewidth at most two.

▶ **Lemma 33** (Preservation). *If $G \in \mathsf{TW}_2$ is a connected term-labeled graph and $G \mapsto G'$, then $G' \in \mathsf{TW}_2$ and G' is connected.*

Proof. Clearly, no rule disconnects the graph. Also, if $G \mapsto G'$, then the strong skeleton of G' is a minor of the strong skeleton of G and taking minors does not increase treewidth. ◀

▶ **Lemma 34** (Progress). *If $G \in \mathsf{TW}_2$ is a connected term-labeled graph, then either there exists some G' such that $G \mapsto G'$ or G is atomic.*

Proof. W.l.o.g., we can assume that rules 3 and 4 do not apply. Thus, it suffices to show that either ι and o are the only vertices of G or that there is some vertex distinct from input and output that has at most two neighbors. Let T be a tree decomposition of the strong skeleton of G of width at most two, and remove leafs of T that are included in their unique neighbor (T remains a tree-decomposition). If T has only one node (say t) then $\{\iota, o\} \subseteq B_t$. Hence, if there is another vertex, it has degree at most two. Otherwise, let t be a leaf and let z be a vertex appearing only in B_t. Without loss of generality, we can assume $z \notin \{\iota, o\}$. (If $z = \iota$ then $o \in B_t$, due to the ιo-edge in the strong skeleton; hence, for any other leaf, neither ι nor o can be the vertex unique to that leaf.) Since z appears only on B_t it has at most two neighbors. ◀

▶ **Definition 35.** We define a function t' from connected term-labeled graphs of treewidth at most two to terms as follows: $\mathsf{t}'(G) \triangleq \underline{A}$ for some atomic graph such that $G \mapsto^* A$. A suitable atomic graph A can be computed by blindly applying the rules (Lemmas 33 and 34): all choices lead to equivalent terms (Lemma 26). For connected (standard) graphs G, we write $\mathsf{t}(G)$ for $\mathsf{t}'(G')$ where G' is G seen as a term-labeled graph.

▶ **Lemma 36.** *If $G \in \mathsf{TW}_2$ is a term-labeled graph and $G \rightsquigarrow H$, then $\mathsf{t}'(G) \equiv \mathsf{t}'(H)$.*

Proof. Follows with Lemma 26. ◀

As an immediate consequence of the lemma above we also have:

▶ **Proposition 37.** *If $G, H \in \mathsf{TW}_2$ are are connected and $G \simeq H$, then $\mathsf{t}(G) \equiv \mathsf{t}(H)$.*

We now show that t and g are inverses up to term equivalence and isomorphism respectively.

▶ **Proposition 38.** *For all Σ_{dom}-terms u, $\mathsf{t}(\mathsf{g}(u)) \equiv u$.*

Proof. We have $\mathsf{t}(\mathsf{g}(u)) \equiv \underline{\lceil \mathsf{f}(u) \rceil} = \lfloor \mathsf{f}(u) \rfloor$ by Proposition 31 and Lemma 36. The claim then follows with Lemma 28. ◀

▶ **Proposition 39.** *If $G \in \mathsf{TW}_2$ is connected, then $\mathsf{g}(\mathsf{t}(G)) \simeq G$.*

Proof. We have $\mathsf{t}(G) = \underline{A}$ for some A such that $G \mapsto^* A$. Hence, $\exp(A) \simeq G$ (Lemma 21). The claim follows since $\mathsf{g}(\underline{A}) \simeq \exp(A)$ for all atoms A. ◀

The function g is a Σ_{dom}-homomorphism by definition. By the above results, this is actually an isomorphism between the *2pdom*-algebra of connected graphs in TW_2 and the (canonically) free *2pdom*-algebra of Σ_{dom}-terms quotiented by $\equiv$:

▶ **Theorem 40** ([16])**.** *The connected graphs in TW_2 (with labels in $\mathcal{A}$) form the free 2pdom-algebra (over $\mathcal{A}$).*

6 The free 2p-algebra

We now extend the results from the previous section to disconnected graphs. That is, we show that the class of all graphs in TW_2 forms the free 2p-algebra [7]. We use for that the previous function t to extract terms from the various connected components of a graph.

In this section, we take $u \equiv v$ to mean $2p \vdash u = v$. Recall that $2p \vdash u = v$ whenever $2pdom \vdash u = v$ (Lemma 4). Hence, all the lemmas from the previous section still apply.

▶ **Definition 41.** Let G be a graph. For vertices x, y of G, we write $G[x, y]$ for the graph G with input set to x and output set to y. We abbreviate $G[x, x]$ as $G[x]$. Further, we write G_x for the connected component of x (as a subgraph of G, with input and output set to x).

▶ **Definition 42.** Let $\mathcal{C}(G)$ be the collection of components G_x obtained by choosing some vertex x for every connected component of G containing neither ι nor o. We define a function $\mathsf{t}^\top$ extracting terms from (possibly disconnected) graphs as follows:

$$c_G \triangleq \bigparallel_{H \in \mathcal{C}(G)} \top \cdot \mathsf{t}(H) \cdot \top \qquad \mathsf{t}^\top(G) \triangleq \begin{cases} \mathsf{t}(G_\iota) \cdot \top \cdot \mathsf{t}(G_o) \parallel c_G & \iota \text{ and } o \text{ disconnected} \\ \mathsf{t}(G_\iota[\iota, o]) \parallel c_G & \iota \text{ and } o \text{ connected} \end{cases}$$

Note that the function $\mathsf{t}^\top$ needs to choose shared input/outputs vertices for all disconnected components. For isomorphic arguments, these choices can differ. We begin by showing that this choice does not matter up to term equivalence.

▶ **Lemma 43.** *Let $G \in \mathsf{TW}_2$ be a connected test and let x be a neighbor of ι in G. We have $\mathsf{t}(G) \cdot \top \equiv \mathsf{t}(G[\iota, x]) \cdot \top$.*

Proof. Since $G \in \mathsf{TW}_2$, so is $G[\iota, x]$. Hence, $G[\iota, x] \rightsquigarrow \lceil(\alpha, u, \beta)\rceil$ for some terms α,β, and u. Since $G = \mathrm{dom}(G[\iota, x])$, we also have $G \rightsquigarrow \mathrm{dom}(\lceil(\alpha, u, \beta)\rceil)$ by Lemma 30. Moreover, $\mathrm{dom}(\lceil(\alpha, u, \beta)\rceil) \rightsquigarrow \lceil(\alpha{\cdot}\mathrm{dom}(u\beta))\rceil$ by rule 2. We reason as follows:

$$\begin{aligned}
\mathsf{t}(G){\cdot}\top &\equiv \mathsf{t}'(\lceil(\alpha{\cdot}\mathrm{dom}(u\beta))\rceil){\cdot}\top && \text{Lemma 36}\\
&= (\alpha{\cdot}\mathrm{dom}(u\beta)){\cdot}\top\\
&\equiv \alpha u\beta{\cdot}\top && \text{(A11)}\\
&= \mathsf{t}'(\lceil(\alpha, u, \beta)\rceil){\cdot}\top\\
&\equiv \mathsf{t}(G[\iota, x]){\cdot}\top && \text{Lemma 36}
\end{aligned}$$

◀

▶ **Lemma 44.** *Let $G \in \mathsf{TW}_2$ be a connected graph and let x, y be vertices of G. We have $\top{\cdot}\mathsf{t}(G[x]){\cdot}\top \equiv \top{\cdot}\mathsf{t}(G[y]){\cdot}\top$*

Proof. Since G is connected, it suffices to show the property for all xy-edges of G. We have

$$\begin{aligned}
\top{\cdot}\mathsf{t}(G[x]){\cdot}\top &\equiv \top{\cdot}\mathsf{t}(G[x, y]){\cdot}\top && \text{Lemma 43}\\
&\equiv \top{\cdot}\mathsf{t}(G[y, x])^\circ{\cdot}\top && \mathsf{t} \text{ is a homomorphism}\\
&\equiv \top{\cdot}\mathsf{t}(G[y, x]){\cdot}\top && \text{Lemma 9(3)}\\
&\equiv \top{\cdot}\mathsf{t}(G[y]){\cdot}\top && \text{Lemma 43}
\end{aligned}$$

◀

Lemmas 44 and 43 also appear in [7]. We remark that the proof of Lemma 43 given here, which depends on the definition of t, is considerably simpler than the one in [7]. The proof of Lemma 44 remains essentially unchanged.

▶ **Proposition 45.** *Let $G, H \in \mathsf{TW}_2$. If $G \simeq H$, then $\mathsf{t}^\top(G) \equiv \mathsf{t}^\top(H)$.*

Proof. Follows with Proposition 37 and Lemma 44. ◀

▶ **Proposition 46.** $\mathsf{g}(\mathsf{t}^\top(G)) \simeq G$.

▶ **Lemma 47.** $\mathsf{t}^\top$ *is a homomorphism of 2p-algebras.*

Proof. We already showed that that $\mathsf{t}^\top$ respects graph isomorphisms. It remains to show that $\mathsf{t}^\top$ commutes with all operations.

We show $\mathsf{t}^\top(G{\cdot}H) \equiv \mathsf{t}^\top(G){\cdot}\mathsf{t}^\top(H)$. Let $F \triangleq G{\cdot}H$. We distinguish four cases based on whether ι and o are connected in G and H respectively.

ι and o disconnected in both G and H: In that case, G_o and H_ι are merged into one component of F that is connected neither to the input nor to the output of F. By Lemma 44 and Proposition 37 we therefore have: $c_F \equiv (\top\mathsf{t}(G_o{\cdot}H_\iota)\top) \parallel c_G \parallel c_H$. We reason as follows:

$$\begin{aligned}
\mathsf{t}^\top(F) &\equiv \mathsf{t}(F_\iota){\cdot}\top{\cdot}\mathsf{t}(F_o) \parallel c_F && \iota \text{ and } o \text{ disconnected in F}\\
&\equiv \mathsf{t}(G_\iota){\cdot}\top{\cdot}\mathsf{t}(H_o) \parallel \top{\cdot}\mathsf{t}(G_o{\cdot}H_\iota){\cdot}\top \parallel c_G \parallel c_H && F_\iota \simeq G_\iota,\ F_o \simeq G_o\\
&\equiv \mathsf{t}(G_\iota){\cdot}\top{\cdot}\mathsf{t}(G_o{\cdot}H_\iota){\cdot}\top{\cdot}\mathsf{t}(H_o) \parallel c_G \parallel c_H && \text{Lemma 9(1)}\\
&\equiv \mathsf{t}(G_\iota){\cdot}\top{\cdot}\mathsf{t}(G_o){\cdot}\mathsf{t}(H_\iota){\cdot}\top{\cdot}\mathsf{t}(H_o) \parallel c_G \parallel c_H && \mathsf{t} \text{ is a homomorphim}\\
&\equiv \mathsf{t}^\top(G){\cdot}\mathsf{t}^\top(H) && \text{Lemma 9(2) and its dual}
\end{aligned}$$

***ι* and *o* connected in G but not in *H*:** We have $c_F \equiv c_G \parallel c_H$ by Prop. 37 and Lemma 44.

$$\begin{aligned}
\mathsf{t}^\top(F) &\equiv \mathsf{t}(F_\iota)\cdot\top\cdot\mathsf{t}(F_o) \parallel c_F \\
&\equiv \mathsf{t}(\mathrm{dom}(G_\iota[\iota,o]\cdot H_\iota))\cdot\top\cdot\mathsf{t}(H_o) \parallel c_F && F_\iota \simeq \mathrm{dom}(G_\iota[\iota,o]\cdot H_\iota),\ F_o \simeq H_o \\
&\equiv \mathsf{t}(G_\iota[\iota,o])\cdot\mathsf{t}(H_\iota)\cdot\top\cdot\mathsf{t}(H_o) \parallel c_F && \text{(A11), } \mathsf{t} \text{ is a homomorphism} \\
&\equiv \mathsf{t}(G_\iota[\iota,o])\cdot\mathsf{t}(H_\iota)\cdot\top\cdot\mathsf{t}(H_o) \parallel c_G \parallel c_H \\
&\equiv \mathsf{t}^\top(G)\cdot\mathsf{t}^\top(H) && \text{Lemma 9(2) and its dual}
\end{aligned}$$

The case where input and output are connected only in H is symmetric and the case where they are connected in both graphs follows from t being a homomorphism.

Proving that $\mathsf{t}^\top$ commutes with the other operations is done in a similar manner. ◀

▶ **Proposition 48.** *For all $\Sigma_\top$-terms u, $\mathsf{t}^\top(\mathsf{g}(u)) \equiv u$.*

Proof. By induction on u, using Lemma 47. ◀

▶ **Theorem 49** ([7])**.** *The graphs in TW_2 (with labels in $\mathcal{A}$) form the free 2p-algebra (over $\mathcal{A}$).*

7 1-free 2p-algebras

We now show that the techniques from the previous sections can be adapted to the setting where 1 (and hence dom(_)) are removed from the signature. We define algebras over the signature $\Sigma_\top^{-1} \triangleq \Sigma_\top \setminus \{1\}$, which we call 1-free 2p-algebras, and show that the graphs of treewidth at most two without self-loops and with distinct input and output form the free 1-free 2p-algebra (over $\mathcal{A}$).

The axioms for 1-free 2p-algebras are A1-A4,A6-A8 plus the following three axioms:

$$(u\cdot v \parallel w)\cdot\top = (u \parallel w\cdot v^\circ)\cdot\top \tag{A15}$$
$$u\cdot(v\cdot\top \parallel w) = (u \parallel \top\cdot v^\circ)\cdot w \tag{A16}$$
$$u\cdot v \parallel \top\cdot w = u\cdot(v \parallel \top\cdot w) \tag{A17}$$

The main complication in adapting our techniques to the 1-free case is that the syntax of 1-free 2p-algebras cannot express tests, even though the algebra of graphs still exhibits tests-like structures. For instance, if G is a test without self-loops, then $G\cdot\top$ is a graph of the proposed free 1-free 2p-algebra. To account for this, we distinguish between the type of $\Sigma_\top^{-1}$-terms, written Tm, and a type of *(syntactic) tests* defined as follows:

$$\alpha \in \mathsf{Tst} ::= 1 \mid [u] \qquad (u \in \mathsf{Tm})$$

Tests, which are not terms, allow us to describe graphs that are tests. We let α,β,... range over tests, and we extend the definition of g to tests by setting $\mathsf{g}(1) = 1$ and $\mathsf{g}([u]) = \mathrm{dom}(\mathsf{g}(u))$. Intuitively, in a test $[u]$, the output of the term u does not matter: u will always be used in contexts where this information disappears, e.g., as in $u\top$; this allows us to treat $[u]$ essentially like $\mathrm{dom}(u)$. It also motivates the following notion of equivalence for tests: $1 \equiv 1$, and $[u] \equiv [v]$ if $u\cdot\top \equiv v\cdot\top$.

▶ **Lemma 50.** *If $\alpha \equiv \beta$ then $\mathsf{g}(\alpha) \simeq \mathsf{g}(\beta)$.*

We extend the sequential composition to take one or two tests as arguments in a manner that 1 is the neutral element on both sides:

$$1{\cdot}v \triangleq v \qquad u{\cdot}1 \triangleq u \qquad 1{\cdot}\alpha \triangleq \alpha$$
$$[u]{\cdot}v \triangleq u\top \parallel v \qquad u{\cdot}[v] \triangleq u \parallel \top v^\circ \qquad [u]{\cdot}1 \triangleq [u]$$
$$[u]{\cdot}[v] \triangleq [u\top \parallel v]$$

Note that $\alpha{\cdot}\beta$ is a test whereas all other variants are terms. The three operations above appropriately preserve test equivalence (e.g., if $\alpha \equiv \beta$, then $u\alpha \equiv u\beta$, $\alpha u \equiv \beta u$, $\gamma\alpha \equiv \gamma\beta$, $\alpha\gamma \equiv \beta\gamma$ for all terms u and tests γ.

The definitions above essentially yield a 2-sorted extension of 1-free 2p-algebras. We prove various laws, including all axioms of 2pdom-algebras that can still be expressed in the 2-sorted setting (using $[\_]$ instead of $\mathrm{dom}(\_)$). Examples of laws that cannot be expressed in the 2-sorted setting are $\mathrm{dom}(\alpha) \equiv \alpha$, and $\mathrm{dom}(u \parallel v) \equiv 1 \parallel uv^\circ$.

▶ **Lemma 51.** *We have the following equivalences:*

1. $u\top \equiv [u]\top$ *and* $\top u \equiv \top[u^\circ]$.
2. $(\alpha u)^\circ \equiv u^\circ\alpha$, $(u\alpha)^\circ \equiv \alpha u^\circ$.
3. $(xy)z \equiv x(yz)$ *(for all x, y, z either test or term).*
4. $[uv] \equiv [u[v]]$
5. $\alpha\beta \equiv \beta\alpha$
6. $\alpha(v \parallel w) \equiv \alpha v \parallel w$.
7. $[uv \parallel w] \equiv [u \parallel wv^\circ]$.

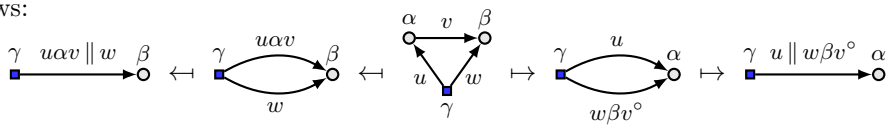
Proof. For all statements involving tests α of unknown shape, we distinguish the cases $\alpha = 1$ (usually trivial) and $\alpha = [w]$ for some w. Claims (1) and (2) are straightforward. By (2) and the laws for converse, we only need to consider 5 of the 8 cases of (3). $(u\alpha)v \equiv u(\alpha v)$ follows with (A16) and $(uv)\alpha \equiv u(\alpha v)$ follows with (A17). For $(\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$ we repeatedly use (A15) with $v = \top$. The remaining cases for associativity are straightforward. Claims (4) and (5) follow with associativity. Claim (6) follows with (A1). Claim (7) follows with (A15). ◀

Having recovered most of the laws of 2pdom-algebras, we adapt the rewriting system for 2pdom-algebras (Figure 3) to the 1-free case. We define term-labeled graphs as for 2pdom, with the difference that now vertices are labeled with syntactic tests and edges are labeled with $\Sigma_\top^{-1}$-terms (whose graphs are never tests). The rewriting system on term-labeled graphs (Figure 3) is adapted by replacing $\mathrm{dom}(u)$ with $[u]$, removing rule 4, and restricting rules 1 and 3 such that the two outer vertices must be distinct. For rule 1, this is necessary to avoid introducing self loops.

Local confluence adapts, although for one of the pairs we now need two reduction steps to join the two alternatives.

▶ **Lemma 52** (Local Confluence). *If $G_1 \hookleftarrow G \mapsto G_2$, then there exist G_1' and G_2' such that $G_1 \rightsquigarrow G_1'$, $G_2 \rightsquigarrow G_2'$ and $G_1' \cong G_2'$.*

Proof. The only interesting (new) critical pair is that of overlapping instances of rule 1, where the outer nodes are the same. Due to the restriction that the outer nodes of rule 1 must be distinct, this pair can no longer be joined by applying rule 1. Instead we use rules 3 and 2 as follows:



After applying rule 2 on both sides, it suffices to show $[(u\alpha v \parallel w)\beta] \equiv [(u \parallel w\beta v^\circ)\alpha]$. This follows with Lemma 51(6+7). ◀

In order to adapt Proposition 31, we need to restrict to terms u such that $\mathsf{g}(u)$ is connected. We write Tm' and Tst' for the set of terms and tests respectively, where tests and the extended sequential composition are treated as primitive and $\top$ does not occur. We then employ a function $\mathsf{f} : \mathsf{Tm}' \to \mathsf{Tst}' \times \mathsf{Tm}' \times \mathsf{Tst}'$ that can be seen as a type directed variant of the function in Figure 5.

$$
\begin{aligned}
\mathsf{f}(a) &= (1, a, 1)\\
\mathsf{f}(u^\circ) &= \mathsf{let}\ (\alpha_u, u', \beta_u) := \mathsf{f}(u)\ \mathsf{in}\ (\beta_u, u'^\circ, \alpha_u)\\
\mathsf{f}(u \parallel v) &= \mathsf{let}\ (\alpha_u, u', \beta_u), (\alpha_v, v', \beta_v) := \mathsf{f}(u), \mathsf{f}(v)\ \mathsf{in}\ (\alpha_u\alpha_v, u' \parallel v', \beta_u\beta_v)\\
\mathsf{f}(u{\cdot}v) &= \mathsf{let}\ (\alpha_u, u', \beta_u), (\alpha_v, v', \beta_v) := \mathsf{f}(u), \mathsf{f}(v)\ \mathsf{in}\ (\alpha_u, u'\beta_u\alpha_v v', \beta_v)\\
\mathsf{f}(\gamma{\cdot}u) &= \mathsf{let}\ (\alpha_u, u', \beta_u) := \mathsf{f}(u)\ \mathsf{in}\ (\gamma\alpha_u, u', \beta_u)\\
\mathsf{f}(u{\cdot}\gamma) &= \mathsf{let}\ (\alpha_u, u', \beta_u) := \mathsf{f}(u)\ \mathsf{in}\ (\alpha_u, u', \beta_u\gamma)
\end{aligned}
$$

▶ **Lemma 53.** *If $u \in \mathsf{Tm}'$ and $\alpha \in \mathsf{Tst}'$, then $\mathsf{g}(u) \rightsquigarrow \lceil \mathsf{f}u \rceil$ and $\mathsf{g}(\alpha) \rightsquigarrow \lceil(\alpha)\rceil$.*

Proof. We have a context lemma similar to Lemma 30. The proof then proceeds by mutual induction on u and α. The cases correspond to those of Proposition 31. We show two representative cases:

Case $u = \alpha \cdot v$: We reason as follows.

$$
\begin{aligned}
\mathsf{g}(\gamma \cdot v) &\cong \mathsf{g}(\gamma) \cdot \mathsf{g}(v) && \text{def. of } \gamma \cdot v\\
&\rightsquigarrow \lceil(\gamma)\rceil \cdot \lceil f\ v\rceil && \text{IH on } \gamma \text{ and } v\text{, context lemma}\\
&= \lceil(\gamma)\rceil \cdot \lceil(\alpha_v, v', \beta_v)\rceil && f\ v = (\alpha_v, v', \beta_v) \equiv v\\
&= \lceil(\gamma\alpha_v, v', \beta_v)\rceil\\
&= \lceil f(\gamma \cdot v)\rceil
\end{aligned}
$$

Case $\alpha = [u]$: We reason as follows.

$$
\begin{aligned}
\mathsf{g}([u]) &= \mathrm{dom}(\mathsf{g}(u)) && \text{def. } \mathsf{g}\\
&\rightsquigarrow \mathrm{dom}(\lceil f\ u\rceil) && \text{IH on } u\text{, context lemma}\\
&= \mathrm{dom}(\lceil(\alpha_u, u', \beta_u)\rceil) && f\ u = (\alpha_u, u', \beta_u) \equiv u\\
&\mapsto \lceil(\alpha_u[u'\beta_u])\rceil && \text{rule 2}\\
&\cong \lceil([u])\rceil
\end{aligned}
$$

The remaining cases are analogous. ◀

We define two extraction functions t_1 and t_2, where t_1 extracts syntactic tests (in Tst') from graphs that are tests and t_2 extracts terms (in Tm') from non-tests. Both functions are defined just like t (Definition 35), exploiting the fact that the rewriting system does not merge or delete input and output. Propositions 38 and 39 then adapt without conceptual changes.

▶ **Proposition 54.** 1. *$\mathsf{t}_1(\mathsf{g}(\alpha)) \equiv \alpha$ for all $\alpha \in \mathsf{Tst}'$ and $\mathsf{t}_2(\mathsf{g}(u)) \equiv u$ for all $u \in \mathsf{Tm}'$.*
2. *If $G \in \mathsf{TW}_2$ is a connected test without self loops, then $\mathsf{g}(\mathsf{t}_1(G)) \simeq G$.*
3. *If $G \in \mathsf{TW}_2$ is a connected non-test without self loops, then $\mathsf{g}(\mathsf{t}_2(G)) \simeq G$.*

Using t_1 and t_2, we define a variant of $\mathsf{t}^\top$ extracting $\Sigma_\top^{-1}$-terms from non-tests without self-loops.

▶ **Definition 55.** Let $\mathcal{C}(G)$ as in Definition 42. We define

$$c_G \triangleq \mathop{\Big\|}_{H \in \mathcal{C}(G)} \top \cdot \mathsf{t}_1(H) \cdot \top \qquad \mathsf{t}^\top(G) \triangleq \begin{cases} \mathsf{t}_1(G_\iota) \cdot \top \cdot \mathsf{t}_1(G_o) \parallel c_G & \iota \text{ and } o \text{ disconnected} \\ \mathsf{t}_2(G_\iota[\iota, o]) \parallel c_G & \iota \text{ and } o \text{ connected} \end{cases}$$

That $\mathsf{t}^\top$ respects graph isomorphisms is immediate with Proposition 54. To show $\mathsf{t}^\top$ is a homomorphism of 1-free 2p-algebras, we require a 2-sorted analog to Lemma 9.

▶ **Lemma 56.** *We have the following equivalences:*

1. $\top u^\circ \top \equiv \top u \top$.
2. $\alpha v \equiv \alpha \top \parallel v$.
3. $\alpha \top \beta \equiv \alpha \top \parallel \top \beta$
4. $uv \parallel \top \alpha \top \equiv (u \parallel \top \alpha \top) v$
5. $\alpha \top \beta \parallel \top \gamma \top \equiv \alpha \top \gamma \top \beta$

Note that, due to Lemma 51(1), any equivalence where a test α appears either only as $\alpha\top$ or only as $\top\alpha$ (i.e., in α in Lemma 56(4)), also holds if α is replaced by a term.

The remaining proofs of Section 6 adapt to the 1-free setting by carefully distinguishing between terms and tests, but without any conceptual changes. For instance, we have $\mathsf{t}_1(G)\top \equiv \mathsf{t}_2(G[\iota, x])\top$ for neighbors x of ι.

▶ **Theorem 57.** *The graphs (with labels in $\mathcal{A}$) of treewidth at most two, with distinct input and output, and without self-loops form the free 1-free 2p-algebra (over $\mathcal{A}$).*

The axioms we listed for 1-free 2p-algebras are precisely those needed to prove the 2-sorted 2pdom- and 2p-laws. The proofs of Lemmas 51 and 56 have been verified in the Coq proof assistant [6]. We also used the model generator Mace4 [15] to verify that the axioms of 1-free 2p-algebras are independent. The corresponding scripts can be downloaded from [10].

8 Conclusion and directions for future work

We have proved that graphs in TW_2, connected graphs in TW_2, and self-loop free graphs in TW_2 with distinct input and output respectively form the free 2p-algebra, the free 2pdom-algebra, and the free 1-free 2p-algebra.

To do so, we used a graph rewriting system that makes it possible to extract terms from connected graphs in TW_2, in a bottom-up fashion. This technique is much easier than the one used in [7] in that it is more local and does not require us to study the precise structure of graphs in TW_2 (i.e., through excluded minors).

As explained in the introduction, the result about connected graphs can be reduced to the one about arbitrary graphs by model-theoretic means: one can easily embed a 2pdom-algebra into a 2p-algebra [16], so that 2p-algebras form a conservative extension of 2pdom-algebras. As a corollary of Theorem 57, we get that 2p-algebras also form a conservative extension of 1-free 2p-algebras. It is however unclear how to prove this result directly, by model-theoretic means: terms which are missing in 1-free 2p-algebras (self-loops) can occur deep inside terms of 2p-algebras, unlike terms which are missing in 2pdom-algebras (disconnected components).

As a natural follow-up to this work, we would like to study whether one can characterize the classes of graphs of higher treewidth as free algebras. The present approach seems promising for treewidth at most three: a reasonable rewriting system is known for recognising such graphs [2]. In contrast, trying to exploit the four excluded minors known to characterize treewidth three [4, 2] seems extremely difficult. For larger treewidth, rewriting systems recognizing graphs of a given treewidth can be shown to exist [1]. However, the result is nonconstructive in the same way as the existence of a finite set of excluded minors for each treewidth [17]).

References

1 S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. *Journal of the ACM*, 40(5):1134–1164, 1993.

2 S. Arnborg and A. Proskurowski. Characterization and recognition of partial 3-trees. *SIAM J. Algebraic Discrete Methods*, 7(2):305–314, Apr. 1986.

3 S. Arnborg and A. Proskurowski. Canonical representations of partial 2- and 3-trees. In *Proc. SAWT*, pages 310–319. Springer, 1990.

4 S. Arnborg, A. Proskurowski, and D. G. Corneil. Forbidden minors characterization of partial 3-trees. *Discrete Mathematics*, 80(1):1–19, 1990.

5 C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000.

6 Coq team. The Coq proof assistant.

7 E. Cosme-Llópez and D. Pous. K_4-free graphs as a free algebra. In *Proc. MFCS*, volume 83 of *LIPIcs*. Schloss Dagstuhl, 2017.

8 B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge Univ. Press, 2012.

9 R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2005.

10 C. Doczkal and D. Pous. Supplementary material accompanying this paper. `http://perso.ens-lyon.fr/damien.pous/covece/tw2rw`.

11 C. Doczkal and D. Pous. Treewidth-two graphs as a free algebra. In *Proc. MFCS*, volume 117 of *LIPIcs*, pages 60:1–60:15. Schloss Dagstuhl, 2018.

12 R. Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.

13 E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proc. NCAI*, pages 4–9. AAAI Press / The MIT Press, 1990.

14 M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1:1–1:24, 2007.

15 W. McCune. Prover9 and Mace4, 2005–2010.

16 D. Pous and V. Vignudelli. Allegories: decidability and graph homomorphisms, 2018. to appear in Proc. LiCS 2018.

17 N. Robertson and P. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325 – 357, 2004.