



**HAL**  
open science

# Treewidth-Two Graphs as a Free Algebra

Christian Doczkal, Damien Pous

► **To cite this version:**

| Christian Doczkal, Damien Pous. Treewidth-Two Graphs as a Free Algebra. 2018. hal-01780844v1

**HAL Id: hal-01780844**

**<https://hal.science/hal-01780844v1>**

Preprint submitted on 28 Apr 2018 (v1), last revised 12 Sep 2018 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Treewidth-Two Graphs as a Free Algebra

**Christian Doczkal**

CNRS, Plume team, LIP, ENS de Lyon, France  
christian.doczkal@ens-lyon.fr

**Damien Pous**

CNRS, Plume team, LIP, ENS de Lyon, 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 presented last year at MFCS and was proved 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 treewidth-two 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. It could also be amenable to handle graphs of larger treewidth, where the excluded minors are typically unknown.

**2012 ACM Subject Classification** Algebra, Graph theory

**Keywords and phrases** Treewidth, Universal algebra, Rewriting

## 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) [12, 5, 13].

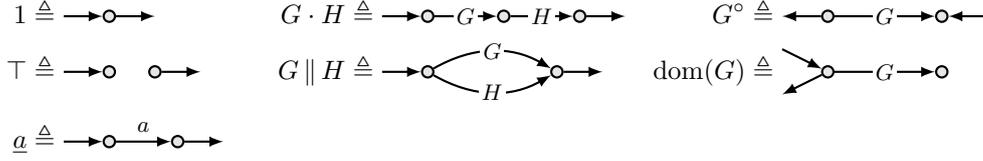
Similar to trees, graphs of bounded treewidth can be described by a variety of syntaxes [8]. Among the open problems, there is the question, for graphs of a given treewidth, of finding a syntax making it possible to get a finite and equational axiomatisation of graph isomorphism [8, page 118]. This question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

The syntax used in [7] is comprised of two binary operations: *series* and *parallel* composition [11], 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 corresponding equational theory is characterized by twelve equational axioms, defining so-called 2p-algebras.

To get this result, the authors define a function  $t$  from graphs to terms and establish that  $t$  is a isomorphism of 2p-algebras. The function  $t$  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 [11]. The authors also make  $t$  as canonical as possible in order to facilitate the proof that on isomorphic graphs,  $t$  returns terms that are congruent modulo the axioms. This comes at the price of complicating the proofs that  $t$  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





■ **Figure 1** Graph operations

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

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

graph  $\underline{a} \parallel 1$  consists of a single vertex with a self-loop labeled with  $a$ . Also note that the graph  $\text{dom}(G)$  is isomorphic to the graph  $G \cdot \top \parallel 1$ . For  $\Sigma_\top$ -terms, we will therefore consider  $\text{dom}(u)$  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  $\Sigma_{\text{dom}}$ -algebra satisfying axioms A1,A2,A4-A10,A13,A14 from Figure 2.

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

**Proof.** This easy result is implicitly proved in [15]; 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  $\Sigma_{\text{dom}}$ -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  $\Sigma_{\text{dom}}$ -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  $\mathbf{g}(u)$  to every term over the considered signatures. By Proposition 5,  $2p \vdash u = v$  entails  $\mathbf{g}(u) \simeq \mathbf{g}(v)$  for all  $\Sigma_\top$ -terms  $u, v$ . Similarly for  $\Sigma_{\text{dom}}$ -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  $\Sigma_{\text{dom}}$ -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  $\alpha, \beta$ , and  $\gamma$  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  $\text{dom}(u)$  with  $\text{dom}(u^\circ)$ .

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

1.  $\text{dom}(u) = \text{dom}(u) \parallel 1$ , i.e.,  $\text{dom}(u)$  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.** See Appendix A. ◀

► **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.  $\top u^\circ\top = \top u\top$

► **Lemma 10.** *A  $\Sigma_{\text{dom}}$ - or  $\Sigma_{\top}$ -term  $u$  is a test iff  $\mathbf{g}(u)$  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. ◀

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.

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  $\iota$  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 bag  $B_t$  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  $\text{TW}_2$  for the collection of graphs whose strong skeleton has treewidth at most two.

► **Proposition 13** ([7]).  *$\text{TW}_2$  forms a subalgebra of the  $\Sigma_{\top}$ -algebra of graphs.*

► **Corollary 14.** *For every term  $u$ ,  $\mathbf{g}(u) \in \text{TW}_2$ .*

The main results about 2p- and 2pdom-algebras, which we reprove in this paper, are that  $\text{TW}_2$  (up to isomorphism) forms the free 2p-algebra [7] and that the connected graphs in  $\text{TW}_2$  form the free 2pdom-algebra [15]. 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  $\Sigma_{\text{dom}}$ -terms and connected graphs in Sections 3 to 5; for these sections we thus abbreviate  $2\text{pdom} \vdash u = v$  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  $\Sigma_{\text{dom}}$ -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  $\text{exp}(G)$  for the *expansion* of  $G$  obtained by replacing every edge  $e$  with  $\mathbf{g}(l(e))$  and every vertex  $x$  with  $\mathbf{g}(l(x))$ .

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. See Appendix B for an example.

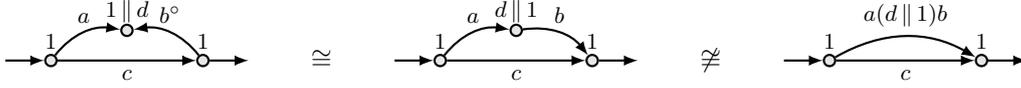
We will compare term-labeled graphs using a notion of isomorphism where labels are compared modulo 2pdom-axioms. A subtlety here is that we should consider as equivalent two graphs where one is obtained from the other by reversing a  $u$ -labeled edge and labeling it with  $u^\circ$  (this operation 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 16.** 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  $f(o) = o'$ .
2. For all vertices  $x \in V$ ,  $l(x) \equiv l'(f(x))$ .
3. For all edges  $e \in E$ ,  $L(s'(g(e)), t'(g(e)), g(e), l(e))$ .

► **Example 17.** 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 18.** 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  $\alpha \cdot \text{dom}(u \cdot \beta)$  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  $(u || v)$ -labeled  $xy$ -edge.
4. If  $s(e) = t(e) = x$ ,  $l(x) = \alpha$  and  $l(e) = u$ , then assign label  $\alpha(u || 1)$  to  $x$  and remove  $e$ .

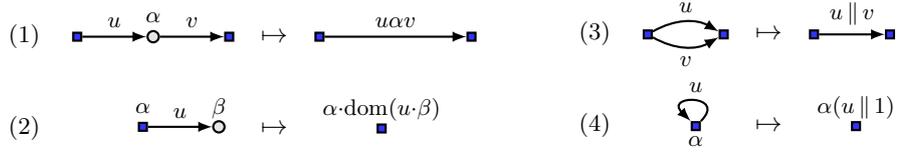
It is straightforward to verify that  $\mapsto$  preserves the requirements on edge and vertex labels from Definition 15. We write  $\rightsquigarrow$  for the transitive closure of  $\mapsto$  up to  $\cong$  (i.e.,  $G \rightsquigarrow H$  iff there exists a sequence  $G \cong G_1 \mapsto G_2 \cong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

► **Lemma 19.** *The relation  $\mapsto$  is terminating.*

► **Lemma 20.** *If  $G \mapsto G'$ , then  $\text{exp}(G) \simeq \text{exp}(G')$ .*

► **Lemma 21.** *If  $G \cong H$  and  $G \mapsto G'$ , then there exists  $H'$  such that  $H \mapsto H'$  and  $G' \cong H'$ .*

## 1:6 Treewidth-Two Graphs as a Free Algebra



■ **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.

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 29 below).

► **Lemma 22 (Local Confluence).** *If  $G_1 \leftarrow G \mapsto G_2$ , 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:

■ Rules 1 and 2 can interact as follows:

$$\begin{array}{c} \gamma \\ \square \end{array} \xrightarrow{u\alpha v} \begin{array}{c} \beta \\ \circ \end{array} \leftarrow \begin{array}{c} \gamma \\ \square \end{array} \xrightarrow{u} \begin{array}{c} \alpha \\ \circ \end{array} \xrightarrow{v} \begin{array}{c} \beta \\ \circ \end{array} \mapsto \begin{array}{c} \gamma \\ \square \end{array} \xrightarrow{u \ \alpha \text{dom}(u\beta)} \begin{array}{c} \circ \end{array}$$

After applying rule 2 on both sides, it suffices to show  $\text{dom}(u\alpha v\beta) \equiv \text{dom}(u\alpha \text{dom}(v\beta))$ , which is an instance of (A13).

■ Rules 1 and 3 can interact as follows:

$$\begin{array}{c} u\alpha v^\circ \\ \square \\ \gamma \end{array} \leftarrow \begin{array}{c} \gamma \\ \square \end{array} \xrightarrow{u} \begin{array}{c} \alpha \\ \circ \end{array} \xrightarrow{v} \begin{array}{c} \beta \\ \circ \end{array} \mapsto \begin{array}{c} \gamma \\ \square \end{array} \xrightarrow{u \parallel v} \begin{array}{c} \alpha \\ \circ \end{array}$$

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

■ Rules 3 and 4 can interact as follows:

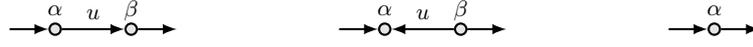
$$\begin{array}{c} u \parallel v \\ \square \\ \gamma \end{array} \leftarrow \begin{array}{c} u \quad v \\ \square \\ \gamma \end{array} \mapsto \begin{array}{c} v \\ \square \\ \gamma(u \parallel 1) \end{array}$$

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 23 (Confluence).** *If  $G_1 \rightsquigarrow G \rightsquigarrow G_2$ , then there exists  $H$  such that  $G_1 \rightsquigarrow H \rightsquigarrow G_2$ .*

► **Definition 24.** 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 be extracted from  $A$ , i.e.,  $\alpha u\beta$ ,  $\alpha u^\circ\beta$ , or  $\alpha$  for the atoms in Figure 4, from left to right.



■ **Figure 4** Atomic Graphs

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

■ **Figure 5** Test analysis for  $\Sigma_{\text{dom}}$ -terms

► **Lemma 25.** *If  $A \rightsquigarrow G \rightsquigarrow B$  for some atomic graphs  $A, B$ , then  $\underline{A} \equiv \underline{B}$ .*

**Proof.** We have  $A \cong B$  by Proposition 23, 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  $\mathbf{g}(u)$  to atomic graphs. As a consequence, we obtain that  $u \equiv v$  iff  $\mathbf{g}(u) \simeq \mathbf{g}(v)$  and, hence, that equivalence of  $\Sigma_{\text{dom}}$ -terms is decidable.

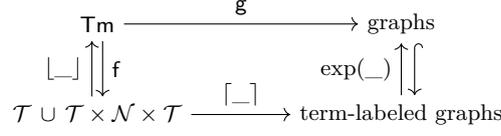
To show that  $\mathbf{g}(u)$  reduces to an atomic graph, we define a function computing for every  $u$  an equivalent term that can be obtained as  $\underline{A}$  for some atomic graph  $A$ . In particular, if  $u$  is not a test, it computes “maximal” tests  $\alpha$  and  $\beta$  and a non-test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

► **Definition 26.** We define a function  $f : \text{TM} \rightarrow \mathcal{T} \cup \mathcal{T} \times \mathcal{N} \times \mathcal{T}$  as depicted in Figure 5, as well as functions  $\llbracket \_ \rrbracket$  and  $\llbracket \_ \rrbracket$  interpreting elements of  $\mathcal{T} \cup \mathcal{T} \times \mathcal{N} \times \mathcal{T}$  as atomic graphs and terms, respectively:  $\llbracket (\alpha, u, \beta) \rrbracket$  is the graph on the left in Figure 4 and  $\llbracket (\alpha) \rrbracket$  is the graph on the right,  $\llbracket (\alpha, u, \beta) \rrbracket \triangleq \alpha u \beta$ , and  $\llbracket (\gamma) \rrbracket \triangleq \gamma$ . Note that  $\llbracket (\alpha, u, \beta) \rrbracket = \llbracket (\alpha, u, \beta) \rrbracket$ .

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

► **Lemma 27.**  $u \equiv \llbracket f(u) \rrbracket$ .

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



■ **Figure 6** Summary of functions between terms and graphs

$$f(v) = (\alpha_v, v', \beta_v).$$

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

The remaining cases are straightforward. ◀

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

► **Definition 28.** 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  $\cdot$ ,  $\parallel$  and  $\text{dom}(\_)$  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 29 (Context Lemma).** *If  $G \rightsquigarrow G'$ , then  $G \parallel H \rightsquigarrow G' \parallel H$ ,  $G \cdot H \rightsquigarrow G' \cdot H$ ,  $H \cdot G \rightsquigarrow H \cdot G'$ , and  $\text{dom}(G) \rightsquigarrow \text{dom}(G')$ .*

**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  $\text{dom}(G)$  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  $G \parallel H$  (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 29 does not hold. For instance,  $\text{dom}(\underline{a})$  (cf. Figure 1) reduces by rule 2 to a single node labeled  $1\text{dom}(a1)$ , but  $\underline{a}$  is an atom.

► **Proposition 30 (Reducibility).**  $\mathbf{g}(u) \rightsquigarrow [f(u)]$

**Proof.** By induction on  $u$ . The base cases are trivial. For the inductive cases, we use Lemma 29 and the induction hypothesis to reduce the respective subgraphs to atomic graphs. The resulting graphs always reduce to atomic graphs in a single step. See Appendix C. ◀

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

► **Theorem 31.**  $2\text{pdom} \vdash u = v$  iff  $\mathbf{g}(u) \simeq \mathbf{g}(v)$ .

**Proof.** The direction from left to right follows with Proposition 5. For the converse direction, assume  $\mathbf{g}(u) \simeq \mathbf{g}(v)$ . Then  $\mathbf{g}(u) \cong \mathbf{g}(v)$  and therefore  $[f(u)] \equiv [f(v)]$  by Proposition 30 and Lemma 25. Hence,  $u \equiv [f(u)] = [f(u)] \equiv [f(v)] = [f(v)] \equiv v$  using Lemma 27 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  $\mathbf{g}$ . 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  $\text{TW}_2$  form the free 2p-algebra, it remains to obtain an inverse to  $\mathbf{g}$  (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  $G \in \text{TW}_2$  to denote that (the strong skeleton of) a term-labeled graph  $G$  has treewidth at most two.

► **Lemma 32** (Preservation). *If  $G \in \text{TW}_2$  is a connected term-labeled graph and  $G \mapsto G'$ , then  $G' \in \text{TW}_2$  and  $G'$  is connected.*

► **Lemma 33** (Progress). *If  $G \in \text{TW}_2$  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  $\iota$  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  $\iota o$ -edge in the strong skeleton; hence, for any other leaf, neither  $\iota$  nor  $o$  can be the vertex unique to that leaf.) Since  $z$  appears only on  $B_t$  it has at most two neighbors. ◀

► **Definition 34.** We define a function  $\mathbf{t}'$  from connected term-labeled graphs of treewidth at most two to terms as follows:  $\mathbf{t}'(G) \triangleq \underline{A}$  for some atomic graph such that  $G \mapsto^* A$ . A suitable atomic graph  $A$  can be computed by blindly applying the rules (Lemmas 32 and 33): all choices lead to equivalent terms (Lemma 25). For connected (standard) graphs  $G$ , we write  $\mathbf{t}(G)$  for  $\mathbf{t}'(G')$  where  $G'$  is  $G$  seen as a term-labeled graph.

► **Lemma 35.** *If  $G \in \text{TW}_2$  is a term-labeled graph and  $G \rightsquigarrow H$ , then  $\mathbf{t}'(G) \equiv \mathbf{t}'(H)$ .*

**Proof.** Follows with Lemma 25. ◀

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

► **Proposition 36.** *If  $G, H \in \text{TW}_2$  and  $G \simeq H$ , then  $\mathbf{t}(G) \equiv \mathbf{t}(H)$ .*

We now show that  $\mathbf{t}$  and  $\mathbf{g}$  are inverses up to term equivalence and isomorphism respectively.

► **Proposition 37.**  $\mathbf{t}(\mathbf{g}(u)) \equiv u$

**Proof.** We have  $\mathbf{t}(\mathbf{g}(u)) \equiv \underline{[f(u)]} = [f(u)]$  by Proposition 30 and Lemma 35. The claim then follows with Lemma 27. ◀

► **Proposition 38.** *If  $G \in \text{TW}_2$  then  $\mathbf{g}(\mathbf{t}(G)) \simeq G$ .*

**Proof.** We have  $\mathbf{t}(G) = \underline{A}$  for some  $A$  such that  $G \mapsto^* A$ . Hence,  $\exp(A) \simeq G$  (Lemma 20). The claim follows since  $\mathbf{g}(\underline{A}) \simeq \exp(A)$  for all atoms  $A$ . ◀

The function  $\mathbf{g}$  is a  $\Sigma_{\text{dom}}$ -homomorphism by definition. By the above results, this is actually an isomorphism between the 2pdom-algebra of connected graphs in  $\text{TW}_2$  and the (canonically) free 2pdom-algebra of  $\Sigma_{\text{dom}}$ -terms quotiented by  $\equiv$ :

► **Theorem 39** ([15]). *The connected graphs in  $\text{TW}_2$  form the free 2pdom-algebra.*

**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  $\text{TW}_2$  forms the free 2p-algebra [7]. We use for that the previous function  $\mathbf{t}$  to extract terms from the various connected components of a graph.

From now on, we take  $u \equiv v$  to mean  $2p \vdash u = v$ . Recall that  $2p \vdash u = v$  whenever  $2p\text{dom} \vdash u = v$  (Lemma 4). Hence, all the lemmas from the previous section still apply.

► **Definition 40.** 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 41.** 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  $\iota$  nor  $o$ . We define a function  $\mathbf{t}^\top$  extracting terms from (possibly disconnected) graphs as follows:

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

Note that the function  $\mathbf{t}^\top$  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 42.** Let  $G \in \text{TW}_2$  be a connected test and let  $x$  be a neighbor of  $\iota$  in  $G$ . We have  $\mathbf{t}(G) \cdot \top \equiv \mathbf{t}(G[\iota, x]) \cdot \top$ .

**Proof.** Since  $G \in \text{TW}_2$ , so is  $G[\iota, x]$ . Hence,  $G[\iota, x] \rightsquigarrow [(\alpha, u, \beta)]$  for some terms  $\alpha, \beta$ , and  $u$ . Since  $G = \text{dom}(G[\iota, x])$ , we also have  $G \rightsquigarrow \text{dom}([\alpha, u, \beta])$  by Lemma 29. Moreover,  $\text{dom}([\alpha, u, \beta]) \rightsquigarrow [(\alpha \cdot \text{dom}(u\beta))]$  by rule 2. We reason as follows:

$$\begin{aligned} \mathbf{t}(G) \cdot \top &\equiv \mathbf{t}'([\alpha \cdot \text{dom}(u\beta)]) \cdot \top && \text{Lemma 35} \\ &= (\alpha \cdot \text{dom}(u\beta)) \cdot \top \\ &\equiv \alpha u \beta \cdot \top && \text{(A11)} \\ &= \mathbf{t}'([\alpha, u, \beta]) \cdot \top \\ &\equiv \mathbf{t}(G[\iota, x]) \cdot \top && \text{Lemma 35} \quad \blacktriangleleft \end{aligned}$$

► **Lemma 43.** Let  $G \in \text{TW}_2$  be a connected graph and let  $x, y$  be vertices of  $G$ . We have  $\top \cdot \mathbf{t}(G[x]) \cdot \top \equiv \top \cdot \mathbf{t}(G[y]) \cdot \top$

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

$$\begin{aligned} \top \cdot \mathbf{t}(G[x]) \cdot \top &\equiv \top \cdot \mathbf{t}(G[x, y]) \cdot \top && \text{Lemma 42} \\ &\equiv \top \cdot \mathbf{t}(G[y, x])^\circ \cdot \top && \mathbf{t} \text{ is a homomorphism} \\ &\equiv \top \cdot \mathbf{t}(G[y, x]) \cdot \top && \text{Lemma 9(3)} \\ &\equiv \top \cdot \mathbf{t}(G[y]) \cdot \top && \text{Lemma 42} \quad \blacktriangleleft \end{aligned}$$

Lemmas 42 and 43 also appear in [7]. For the former, which depends on the definition of  $\mathbf{t}$ , the proof here is considerably simpler, for the latter the proof is essentially the same.

► **Proposition 44.** Let  $G, H \in \text{TW}_2$ . If  $G \simeq H$ , then  $\mathbf{t}^\top(G) \equiv \mathbf{t}^\top(H)$ .

**Proof.** Follows with Proposition 36 and Lemma 43. ◀

► **Proposition 45.**  $g(\mathfrak{t}^\top(G)) \simeq G$ .

► **Lemma 46.**  $\mathfrak{t}^\top$  is a homomorphism of  $2p$ -algebras.

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

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

**$\iota$  and  $o$  disconnected in both  $G$  and  $H$ :** In that case,  $G_o$  and  $H_l$  are merged into one component of  $F$  that is connected neither to the input nor to the output of  $F$ . By Lemma 43 and Proposition 36 we therefore have:  $c_F \equiv (\top \mathfrak{t}(G_o \cdot H_l) \top) \parallel c_G \parallel c_H$ . We reason as follows:

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

**$\iota$  and  $o$  connected in  $G$  but not in  $H$ :** We have  $c_F \equiv c_G \parallel c_H$  by Proposition 36 and Lemma 43.

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

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  $\mathfrak{t}$  being a homomorphism.

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

► **Proposition 47.**  $\mathfrak{t}^\top(g(u)) \equiv u$ .

**Proof.** By induction on  $u$ , using Lemma 46. ◀

► **Theorem 48 ([7]).** *The graphs in  $TW_2$  form the free  $2p$ -algebra.*

## 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  $\text{dom}(\_)$ ) are removed from the signature. We define 1-free  $2p$ -algebras over the signature  $\Sigma_\top^{-1} \triangleq \Sigma_\top \setminus \{1\}$  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.

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

$$(u \cdot v \parallel w) \cdot \top = (u \parallel w \cdot v^\circ) \cdot \top \quad u \cdot (v \cdot \top \parallel w) = (u \parallel \top \cdot v^\circ) \cdot w \quad u \cdot v \parallel \top \cdot w = u \cdot (v \parallel \top \cdot w)$$

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 define a *type of tests* that can be either 1 or  $[u]$  for some term  $u$ . Tests, which are not terms, allow us to describe graphs that are tests. Letting  $\alpha$  and  $\beta$  range over tests, we then define three operations  $\alpha \cdot v$  and  $u \cdot \alpha$  yielding terms and  $\alpha \cdot \beta$  yielding a test. For instance,  $1 \cdot v \triangleq v$  and  $[u] \cdot v \triangleq u \top \parallel v$ . Intuitively, the term  $u$  in  $[u]$  will always be used in contexts where its output is discarded, like in  $u \top$ ; this allows us to treat  $[u]$  essentially like  $\text{dom}(u)$ .

Doing so, we eventually obtain a 2-sorted algebra whose sorts are tests and terms. Using these operations and replacing  $\text{dom}(u)$  with  $[u]$ , the rules of the rewriting system in Figure 3 (except rule 4, since  $u \parallel \alpha$  does not make sense) can also be seen as rules for graphs where vertices are labeled with tests and edges are labeled with  $\Sigma_{\top}^{-1}$ -terms. Thus, removing rule 4 and disallowing the instance of rule 1 where the square vertices coincide (as this would introduce a self-loop) yields a rewriting system on connected self-loop free graphs. We prove 2-sorted variants of the relevant 2pdom- and 2p-laws and define a variant of the function  $f$  that always returns a triple of two tests and a term. This allows us to prove confluence and reducibility. The rest of the proofs then carry over to the disconnected case without conceptual changes. For additional detail, see Appendix D.

The axioms we listed for 1-free 2p-algebras are precisely those needed to prove the 2-sorted 2pdom- and 2p-laws. We verified the derivation of these laws in the Coq proof assistant [6], and we used the model generator Mace4 [14] to verify that these axioms are independent. The corresponding scripts can be downloaded from [10].

► **Theorem 49.** *The graphs of treewidth at most two with distinct input and output and without self-loops form the free 1-free 2p-algebra.*

## 8 Conclusion and directions for future work

We have proved that graphs in  $\text{TW}_2$ , connected graphs in  $\text{TW}_2$ , and self-loop free graphs in  $\text{TW}_2$  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  $\text{TW}_2$ , 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  $\text{TW}_2$  (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 [15], so that 2p-algebras form a conservative extension of 2pdom-algebras. As a corollary of Theorem 49, 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 recognizing such graphs [2]. (In contrast, trying to exploit the four excluded minors known to characterize treewidth three [4, 2] seems extremely difficult.) One could also hope to handle the general case since rewriting systems for recognizing a given treewidth always exist [1] (constructively, unlike the existence of a finite set of excluded minors for each treewidth [16]).

---

**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. <https://coq.inria.fr/>.
- 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. <https://perso.ens-lyon.fr/christian.doczkal/mfcs18>.
- 11 R. Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.
- 12 E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proc. NCAI*, pages 4–9. AAAI Press / The MIT Press, 1990.
- 13 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.
- 14 W. McCune. Prover9 and Mace4. <http://www.cs.unm.edu/~mccune/prover9/>, 2005–2010.
- 15 D. Pous and V. Vignudelli. Allegories: decidability and graph homomorphisms. <https://hal.archives-ouvertes.fr/hal-01703906/>, 2018. to appear in Proc. LiCS 2018.
- 16 N. Robertson and P. Seymour. Graph minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325 – 357, 2004.

**A 2pdom-Facts**

The basic laws about 2pdom-algebras we use (Lemma 8) are implicitly proved in [15], and Coq proofs scripts are available [10]. We nevertheless give their proofs 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.

**Proof of Lemma 8.** As is standard for involutive monoids, we have  $1^\circ = 1$  and, therefore,  $\text{dom}(\alpha) = \alpha$  by (A10).

$\text{dom}(u) = \text{dom}(u) \parallel 1$ :

$$\text{dom}(u) = \text{dom}(u)(1 \parallel 1) = \text{dom}(u)1 \parallel 1 = \text{dom}(u) \parallel 1$$

$\alpha^\circ = \alpha$ :

$$\alpha^\circ = (\alpha \parallel 1)^\circ = 1 \parallel 1\alpha^\circ = \text{dom}(1 \parallel \alpha) = \text{dom}(\alpha) = \alpha$$

$(u \parallel v)\alpha = u \parallel v\alpha$ : We show the dual:

$$\alpha(u \parallel v) = \text{dom}(\alpha)(u \parallel v) = \text{dom}(\alpha)u \parallel v = \alpha u \parallel v$$

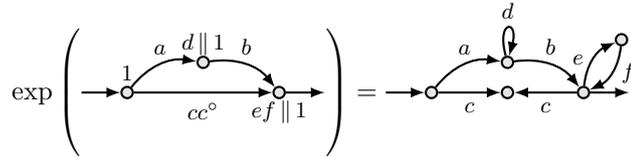
$\alpha\beta = \alpha \parallel \beta = \beta\alpha$ : By commutativity of  $\parallel$ , it suffices to prove the first equation:

$$\alpha\beta = (\alpha \parallel 1)\beta = \alpha \parallel 1\beta = \alpha \parallel \beta$$

This finishes the proof. ◀

**B Graph Expansion**

► **Example 50.** A term-labeled graph and its expansion:



**C Proof of Reducibility**

**Proof of Proposition 30.** The proof proceeds by induction on  $u$ .

case  $u \parallel v$  :

$$\begin{aligned} g(u \parallel v) &= g(u) \parallel g(v) && \text{def. } g \\ &\rightsquigarrow [f(u)] \parallel g(v) && \text{IH, Lemma 29} \\ &\rightsquigarrow [f(u)] \parallel [f(v)] && \text{IH, Lemma 29, comm. } \parallel \end{aligned}$$

We consider several cases.

- If  $f(u) = \gamma$  and  $f(v) = (\alpha_v, v', \beta_v)$ , then  $[f(u)] \parallel [f(v)]$  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:

$$[f(u)] \parallel [f(v)] \rightsquigarrow [(\gamma\alpha_v\beta_v(v' \parallel 1))] \cong [(\gamma \parallel \alpha_v v' \beta_v)] \cong [f(u \parallel v)]$$

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

$$\lceil f(u) \rceil \parallel \lceil f(v) \rceil \rightsquigarrow \lceil (\alpha_u \alpha_v, u' \parallel v', \beta_u \beta_v) \rceil \cong \lceil f(u \parallel v) \rceil$$

The remaining cases are analogous.

**case  $u \cdot v$  :**

$$\begin{aligned} g(u \cdot v) &= g(u) \cdot g(v) && \text{def. } g \\ &\cong \lceil f(u) \rceil \cdot \lceil f(v) \rceil && \text{IH, Lemma 29 (twice)} \end{aligned}$$

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

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

The other cases are straightforward.

**case  $\text{dom}(u)$  :**

$$\begin{aligned} g(\text{dom}(u)) &= \text{dom}(g(u)) && \text{def. } g \\ &\rightsquigarrow \text{dom}(\lceil f(u) \rceil) && \text{IH, Lemma 29} \end{aligned}$$

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

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

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

## D

 1-Free 2p-algebras

We repeat the axioms for 1-free 2p-algebras (in addition to A1-A4, A6-A8) in order to give them names:

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

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

$$u \cdot v \parallel \top \cdot w = u \cdot (v \parallel \top \cdot w) \tag{A17}$$

In this section, we take  $u \equiv v$  to mean that  $u = v$  holds in all 1-free 2p-algebras. Note that, since  $\top^\circ \equiv \top$ ,  $(u \cdot \top \parallel w) \cdot \top \equiv (u \parallel w \cdot \top) \cdot \top$  is (essentially) an instance of (A15); one we will use it repeatedly.

Terms of 1-free 2p-algebras all denote graphs with distinct input and output. To be able to denote tests, we define the following *type of tests*:

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

We let  $\alpha, \beta, \dots$  range over these tests and we extend the definition of  $g$  to tests as follows:

$$g(1) = 1 \qquad g([u]) = \text{dom}(g(u))$$

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 motivates the following notion of equivalence for tests:  $1 \equiv 1$ , and  $[u] \equiv [v]$  if  $u \cdot \top \equiv v \cdot \top$ .

► **Lemma 51.** *If  $\alpha \equiv \beta$  then  $\mathbf{g}(\alpha) \simeq \mathbf{g}(\beta)$ .*

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:

$$\begin{array}{lll} 1 \cdot v \triangleq v & u \cdot 1 \triangleq u & 1 \cdot \alpha \triangleq \alpha \\ [u] \cdot v \triangleq u \top \parallel v & u \cdot [v] \triangleq u \parallel \top v^\circ & [u] \cdot 1 \triangleq [u] \\ & & [u] \cdot [v] \triangleq [u \top \parallel v] \end{array}$$

(Note that  $\alpha \cdot \beta$  is a test whereas all other variants are terms.) Doing so, we obtain a 2-sorted algebra, whose sorts are tests and terms.

The above three operations 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  $\gamma$ ).

We then prove various laws, which cover all axioms of 2pdom-algebras that still make sense in the 2-sorted setting (using  $[\_]$  instead of  $\text{dom}(\_)$ ). All these equivalences have been proved in the interactive theorem prover Coq [10].

► **Lemma 52.**

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  $\alpha$  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  $v = \top$ :

$$\begin{aligned} (([u][v])[w])\top &\equiv ((u\top \parallel v)\top \parallel w)\top \equiv ((u\top \parallel v) \parallel w\top)\top \equiv (u\top \parallel (v \parallel w\top))\top \\ &\equiv (u \parallel (v \parallel w\top))\top \equiv (u \parallel (v\top \parallel w))\top \equiv (u\top \parallel (v\top \parallel w))\top \equiv ([u]([v][w]))\top \end{aligned}$$

The remaining cases for associativity are straightforward. Claims (4) and (5) follow with associativity. Claim (7) follows with (A15). The remaining claims are straightforward. ◀

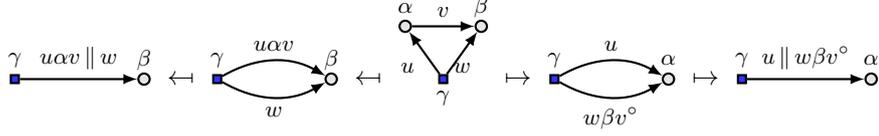
Typical laws that do not make sense in the 2-sorted setting are  $\text{dom}(\alpha) \equiv \alpha$ , and  $\text{dom}(u \parallel v) \equiv 1 \parallel uv^\circ$ .

We define term-labeled graphs as for 2pdom, with the difference that now vertices are labeled with our handcrafted notion of tests and edges are labeled with terms (whose graphs are never tests). The rewriting system on term-labeled graphs (Figure 3) is adapted by removing rule 4, restricting rules 1 and 3 such that the two outer vertices must be distinct, and replacing  $\text{dom}(u)$  with  $[u]$ .

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

► **Lemma 53 (Local Confluence).** *If  $G_1 \leftarrow G \mapsto G_2$ , 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 52(6+7).  $\blacktriangleleft$

For the normalization proof, we employ a function  $f : \mathsf{Tm} \rightarrow \mathsf{Tst} \times \mathsf{Tm} \times \mathsf{Tst}$  that can be seen as a type directed variant of the function in Figure 5.

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

► **Lemma 54.** *If  $u \in \mathsf{Tm}$  and  $\alpha \in \mathsf{Tst}$ , then  $\mathbf{g}(u) \rightsquigarrow [fu]$  and  $\mathbf{g}(\alpha) \rightsquigarrow [(\alpha)]$ .*

**Proof.** We have a context lemma similar to Lemma 29. The proof then proceeds by mutual induction in  $u$  and  $\alpha$ . The cases correspond to those of Proposition 30.  $\blacktriangleleft$

We define two extraction functions  $\mathbf{t}_1$  and  $\mathbf{t}_2$ , where  $\mathbf{t}_1$  extracts syntactic tests from graphs that are tests and  $\mathbf{t}_2$  extracts terms from non-tests. Both functions are defined just like  $\mathbf{t}$  (Definition 34), exploiting the fact that the rewriting system does not merge or delete input and output.

► **Proposition 55.**

1. *If  $G \in \mathsf{TW}_2$  is a test without self loops, then  $\mathbf{g}(\mathbf{t}_1(G)) \simeq G$ .*
2. *If  $G \in \mathsf{TW}_2$  is a non-test without self loops, then  $\mathbf{g}(\mathbf{t}_2(G)) \simeq G$ .*
3.  $\mathbf{t}_1(\mathbf{g}(\alpha)) \equiv \alpha$ .
4.  $\mathbf{t}_2(\mathbf{g}(u)) \equiv u$ .

Using  $\mathbf{t}_1$  and  $\mathbf{t}_2$  we can define a typed variant of  $\mathbf{t}^\top$ .

► **Definition 56.** Let  $\mathcal{C}(G)$  as in Definition 41. We define

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

That  $\mathbf{t}^\top$  respects graph isomorphisms is immediate with Proposition 55. To show  $\mathbf{t}^\top$  is a homomorphism of 1-free 2p-algebras, we require a typed analog to Lemma 9. Again, all the equivalences below have been verified in Coq [10].

► **Lemma 57.**

1.  $\top u^\circ \top \equiv \top u \top$ .

## 1:18 Treewidth-Two Graphs as a Free Algebra

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.  $\alpha \top \beta \parallel \top \gamma \top = \alpha \top \gamma \top \beta$

Note that, due to Lemma 52(1), any equivalence where a test  $\alpha$  appears either only as  $\alpha \top$  or only as  $\top \alpha$  (i.e., in  $\alpha$  and  $\beta$  in Lemma 57(4)), also holds if  $\alpha$  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  $\mathbf{t}_1(G) \top \equiv \mathbf{t}_2(G[\iota, x]) \top$  for neighbors  $x$  of  $\iota$ . Thus we obtain a proof of Theorem 49.