

Split-Decomposition Trees with Prime Nodes: Enumeration and Random Generation of Cactus Graphs

Maryam Bahrani*

Jérémie Lumbroso*

Abstract

In this paper, we build on recent results by Chauve *et al.* and Bahrani and Lumbroso, which combined the split-decomposition, as exposed by Gioan and Paul, with analytic combinatorics, to produce new enumerative results on graphs—in particular the enumeration of several subclasses of perfect graphs (distance-hereditary, 3-leaf power, ptolemaic).

Our goal was to study a simple family of graphs, of which the split-decomposition trees have prime nodes drawn from an enumerable (and manageable!) set of graphs. Cactus graphs, which we describe in more detail further down in this paper, can be thought of as trees with their edges replaced by cycles (of arbitrary lengths). Their split-decomposition trees contain prime nodes that are cycles, making them ideal to study.

We derive a characterization for the split-decomposition trees of cactus graphs, produce a general template of symbolic grammars for cactus graphs, and implement random generation for these graphs, building on work by Iriza.

Introduction

In general, counting graphs is much more difficult than counting trees—the latter are completely recursive, and as such, there are many generic results and theorems that can be applied straightforwardly [18]. Graphs, on the other hand, can have cycles and can thus not be specified recursively in the general case. A *tree decomposition* is a method by which graphs are put in correspondence with rooted trees (usually bijectively), which are easier to specify and study; to convert a graph to a tree, the decomposition makes certain hypotheses on the structure of the graphs.

There exists a number of tree decompositions, and several have been used in the context of analytic combinatorics. Most recently, Chauve *et al.* [11] have used the split decomposition to provide the first exact enumeration of distance-hereditary graphs (the largest subset of graphs that can be entirely decomposed with the split-decomposition).

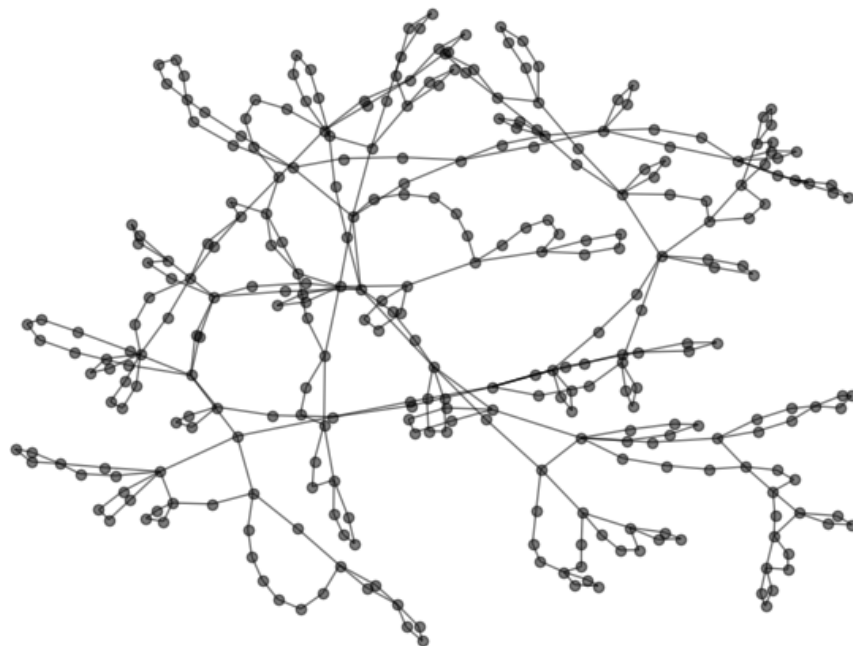
The split decomposition and prime nodes. Informally, a *split* in a graph is a bipartition of the vertices into two sets of size at least 2, such that the edges between the sets form a complete bipartite graph. All splits of a graph form a tree-like structure, where each internal node of the tree is labeled with a graph. The split decomposition [25] refers to the process of transforming a graph into a *graph-labeled tree* using splits of the original graph.

The split decomposition was first introduced by Cunningham [12] and later reformulated by Gioan and Paul [25], who introduced the concept of *graph-labeled trees*. More recently, Chauve *et al.* [10, 11] observed that these graph-labeled trees are well-suited for recursive symbolic specification, which they used to provide, among other things, the first exact enumeration of distance-hereditary graphs. Distance-hereditary graphs are the totally decomposable graphs for the split decomposition [25, 12], and therefore a natural first class of graphs to study when considering the split decomposition for enumerating graphs.

The paper by Chauve *et al.* [11] was first in a line of work on the application of the split decomposition to graph enumeration. Bahrani and Lumbroso [2] used the same framework to give grammars for *subsets* of distance-hereditary graphs that are specified in terms of forbidden induced subgraphs. They showed that constraining a split decomposition tree to avoid certain patterns can help avoid corresponding induced subgraphs in the original graph. The ability to *forbid* induced subgraphs led to grammars and full enumeration for a wide set of graph classes: ptolemaic, block, and variants of cactus graphs (2,3-cacti, 3-cacti and 4-cacti). In certain cases, no enumeration was known (ptolemaic, 4-cacti); in other cases, although the enumerations were known, an abundant potential is unlocked by the provided grammars (in terms of asymptotic analysis, random generation, and parameter analysis, *etc.*).

The methodology has shown its potential in this line of work, specially in providing grammars for totally decomposable graph classes, *i.e.* distance-hereditary graphs and their subfamilies. As stated by Bahrani and Lumbroso [2], the natural next step to develop its reach is to study split decomposition trees that are not fully decomposable and contain prime nodes. This is part of the motivation behind studying cactus graphs, which result in split decomposition trees

*Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540, USA. mbahrani@princeton.edu and lumbroso@cs.princeton.edu



(a) A random mixed cactus graph with 309 vertices and 80 cycles.



(b) A random mixed cactus graph with 933 vertices and 239 cycles.

Figure 1. Our symbolic grammars can be used to create Boltzmann samplers for the uniform random generation of large cactus graphs. Here we have included two cactus graphs, each of which has been drawn uniformly at random from the family of unlabeled, plane, rooted cacti where each cycle has at least 4 nodes.

with prime polygonal nodes. As such, this paper is the first work in the line of papers introduced above to examine split decomposition trees that are not fully decomposable.

Cactus graphs A cactus is a graph in which no two cycles share an edge¹. It can be thought of as a collection of polygons and bridges that pairwise share at most one vertex. Cactus graphs come in many flavors (*e.g.* see Figure 2):

- A cactus graph is *pure* if all its cycles have the same size; it is *mixed* otherwise.
- A graph is *labeled* if its vertices are distinguished for the purpose of enumeration; it is *unlabeled* if the vertices have no distinct identification besides their adjacencies.
- A *rooted* graph has one distinguished vertex; it is otherwise called *unrooted*.
- A *plane* cactus graph is embedded in the plane so that every vertex is paired with a circular permutation of the edges incident to it; when this permutation is insignificant, the cactus graph is called *non-plane* or *free*.

Besides being objects of interest in combinatorics, cactus graphs have applications in other fields. For example, many hard problems on general graphs have efficient algorithms on cactus graphs [3, 9, 13]. Cactus graphs are also used in modeling electronic circuits [30], networks [1], and comparative genomics [31].

The problem of enumerating cactus graphs was first proposed in 1950 in a lecture by Uhlenbeck [35], where these graphs were referred to as *Husimi Trees*. Subsequently, Harry and Uhlenbeck [27] as well as Ford and coauthors [21, 20, 22, 23] and Bergeron *et al.* [4] have examined the enumeration of unlabeled and labeled cactus graphs in a series of notes. These works derive functional equations for the generating functions of cactus graphs, with no convenient way of obtaining a few hundred terms of the enumeration or the prospect of building random generators. Additionally, their methodology is dependent upon the symmetric structure of cycles in cactus graphs and therefore hard to generalize to other graph classes. More recently, Bona *et al.* [8] derived the enumeration of *plane* cactus graphs, but their methodology does not seem to generalize to non-plane cacti.

The goal of this paper is to develop a simple, extendable framework that not only allows for enumerating all aforementioned varieties of cactus graphs but also is amenable to imposing arbitrary restrictions on the structure of the graphs² as well as random generation. This goal is achieved by developing symbolic grammars for different varieties of cactus graphs.

Since cactus graphs are very similar to trees, the same grammars can be developed more directly by exploiting this apparent recursivity. However, while the entire machinery introduced in this paper is not necessary to obtain grammars for cactus graphs, these tools belong to a more general and powerful framework developed by Chauve *et al.* [11] and extended by Bahrani and Lumbroso [2]. This framework is highly flexible and can be generalized to obtain grammars for many other classes of cactus graphs. We therefore believe it is informative to state our results within this framework, specially since our original investigation was to look at identifying characterizable families of graphs of which the split decomposition trees contain a manageable family of prime nodes (in this case: various cycles and sequences).

1 Definitions and Preliminaries

In this section, we introduce standard definitions from graph theory (1.1), followed by a formal introduction to the split-decomposition expressed in terms of graph-labeled trees (1.2 and 1.3), which are the tools also used by Chauve *et al.* [11] and Bahrani and Lumbroso [2].

1.1 Graph definitions. For a graph G , we denote by $V(G)$ its vertex set and $E(G)$ its edge set.

Moreover, for a vertex x of a graph G , we denote by $N(x)$ the neighborhood of x , that is the set of vertices y such that $\{x, y\} \in E(G)$; this notion extends naturally to vertex sets: if $V_1 \subseteq V(G)$, then $N(V_1)$ is the set of vertices defined by the (non-disjoint) union of the neighborhoods of the vertices in V_1 . The subgraph of G induced by a subset V_1 of vertices is denoted by $G[V_1]$.

A clique on k vertices, denoted by K_k , is the complete graph on k vertices (*i.e.*, there exists an edge between every pair of vertices). A star on k vertices, denoted by S_k , is the graph with one vertex of degree $k - 1$ (the *center* of the star) and $k - 1$ vertices of degree 1 (the *extremities* of the star).

A closed walk in a graph is an alternating sequence of vertices and edges, starting and ending at the same vertex, where each edge is adjacent in the sequence to its endpoints. A cycle is a closed walk in which no repetitions of vertices and edges are allowed, except for the starting and ending point. We use the terms *cycles* and *polygons* interchangeably. In this paper we work with simple graphs (*i.e.* graphs without self-loops or multi-edges), so all cycles have at least 3 distinct vertices.

1.2 Graph-labeled trees. We first introduce the notion of *graph-labeled tree*, due to Gioan and Paul [25], then define the split-decomposition and finally give the characterization of a *reduced* split-decomposition tree, described as a graph-labeled tree.

¹In this paper, it is assumed that cactus graphs are connected.

²For example, one could enumerate cactus graphs where the length of each cycle is a prime number.

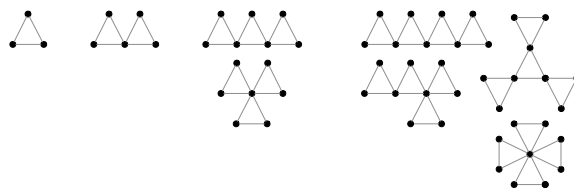


Figure 2. Small (unrooted, unlabeled, pure, free) 3-cacti

DEFINITION 1. A graph-labeled tree (T, \mathcal{F}) is a tree T in which every internal node v of degree k is labeled by a graph $G_v \in \mathcal{F}$ on k vertices, called *marker vertices*, such that there is a bijection ρ_v between the edges of T incident to v and the vertices of G_v .

For example, in Figure 3 the internal nodes of T are denoted by large circles, the marker vertices are denoted by small hollow circles, the leaves of T are denoted by small solid circles, and the bijection ρ_v is denoted by each edge that crosses the boundary of an internal node and ends at a marker vertex.

These graph-labeled trees are a powerful tool for studying the structure of the original graph they describe. Some elements of the terminology have been summarized in Figure 4 (reproduced from Bahrani and Lumbroso [2]), as they are frequently referenced in the proofs of Section 2.

DEFINITION 2. Let (T, \mathcal{F}) be a graph-labeled tree and let $\ell, \ell' \in V(T)$ be leaves of T . We say that there is an *alternated path* between ℓ and ℓ' , if there exists a path from ℓ to ℓ' in T such that for any adjacent edges $e = \{u, v\}$ and $e' = \{v, w\}$ on the path, $\{\rho_v(e), \rho_v(e')\} \in E(G_v)$.

DEFINITION 3. The *original graph*, also called *accessibility graph*, of a graph-labeled tree (T, \mathcal{F}) is the graph G where $V(G)$ is the leaf set of T and, for $x, y \in V(G)$, $\{x, y\} \in E(G)$ iff there is an alternated path between x and y in T .

Figures 3 and 4 (reproduced from Bahrani and Lumbroso [2]) illustrate the concept of an alternated path: it is, more informally, a path that only ever uses at most one interior edge of any graph label.

1.3 Split-decomposition. In this section we summarize various concepts, a more formal and comprehensive treatment of which can be found in previous work [11, 2].

A *split* in a graph is a bipartition of the vertices into two sets of size at least 2, such that the edges between the two sets form a complete bipartite graph. More formally,

DEFINITION 4. A *split* [12] of a graph G with vertex set V is a bipartition (V_1, V_2) of V (i.e., $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$) such that

- (a) $|V_1| \geq 2$ and $|V_2| \geq 2$;
- (b) every vertex of $N(V_1)$ is adjacent to every vertex of $N(V_2)$.

A graph without any split is called a *prime* graph. A graph is *degenerate* if any bipartition of its vertices into sets of size at least 2 is a split: cliques and stars are the only such graphs.

Informally, the split-decomposition of a graph G consists of finding a split (V_1, V_2) in G , followed by decomposing G into two graphs $G_1 = G[V_1 \cup \{x_1\}]$, where $x_1 \in N(V_1)$, and $G_2 = G[V_2 \cup \{x_2\}]$, where $x_2 \in N(V_2)$, and then recursively decomposing G_1 and G_2 . This decomposition naturally defines an unrooted tree structure, called a *split-decomposition tree*, in which the internal vertices are labeled by degenerate or prime graphs and the leaves are in bijection with $V(G)$.

The structure of the split decomposition tree of a graph may depend on the specific sequence of split operations performed on it. However, Cunningham [12] provided a set of criteria that are satisfied by exactly one split decomposition tree for every graph. This uniqueness result is reformulated below in terms of graph-labeled trees by Gioan and Paul [25].

THEOREM (Cunningham [12]). *For every connected graph G , there exists a unique split-decomposition tree such that:*

1. every non-leaf node has degree at least three;
2. no tree edge links two vertices with clique labels;
3. no tree edge links the center of a star to the extremity of another star.

Such a tree is called *reduced*, and this theorem establishes a bijection between graphs and their reduced split decomposition trees.

2 Split-Decomposition of Cactus Graphs

In this section, we first introduce a bijective split decomposition tree characterization of general cactus graphs. We then make certain simplifications to that characterization, which, while respecting its bijective nature, make its correspondence with our cactus grammars more apparent. Additionally, these characterizations make apparent how arbitrary size constraints on cycles of a cactus graph translate into modifications in its split-decomposition tree characterization, which can then be directly translated into modifications in the grammar.

First, we provide a characterization for the reduced

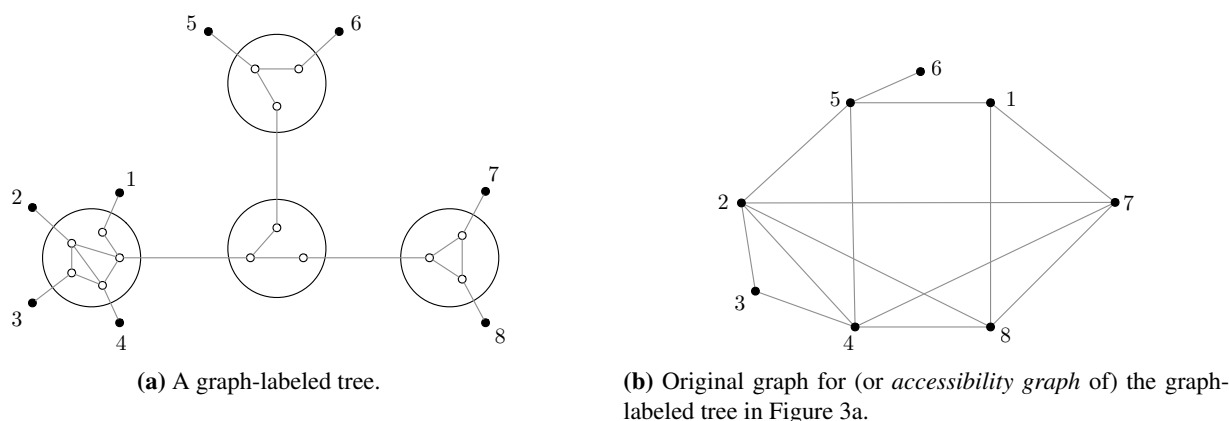


Figure 3. Two leaves of the split-decomposition graph-labeled tree (left) correspond to adjacent vertices in the original graph that was decomposed (right) if there exists an *alternated path*: a path between those leaves, which uses at most one interior edge of any graph label. For example, vertex 5 is adjacent to vertex 4 in the original graph, because there is an alternated path between the two corresponding leaves in the split-decomposition tree; vertex 5 is not adjacent to vertex 3 however, because that would require the path to take two interior edges of the (*prime*) leftmost graph-label.

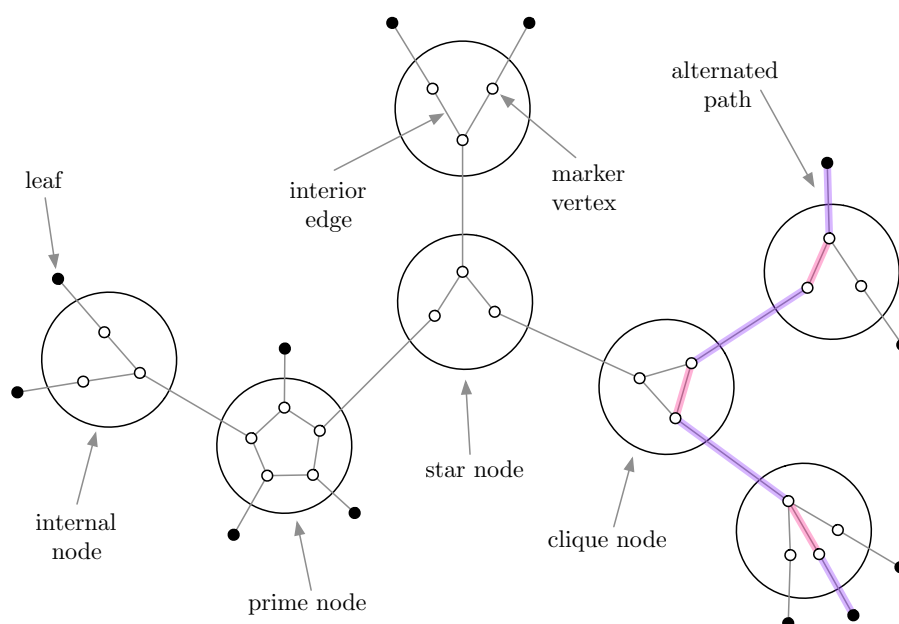


Figure 4. In this figure, we present a few terms that we use a lot in this article.

split decomposition trees of general³ cactus graphs. The following lemma will be used in the proof of Theorem 1.

LEMMA 1 (Polygon primality). *Polygons of size at least 5 are prime with respect to the split decomposition.*

Proof. We will show that a cycle with at least 5 vertices has no splits. Suppose, on the contrary, that such a split exists.

³By *general*, we mean all cactus graphs with no restriction on the size of each cycle. In other words, *general* cactus graphs is the family of all connected graphs in which no two cycles share an edge.

We claim that there are at least two disjoint edges crossing the split. Any bipartition of a cycle has at least two edges crossing it. Suppose all edges crossing the bipartition are non-disjoint. Then they must all be incident to the same vertex v in one side of the split. Since the degree of every vertex in a cycle is 2, there must be exactly two edges crossing the split, both of which are incident to v . This implies that v is the only vertex in its side of the split: Any other vertex on the same side cannot have a path to v without either crossing the split or causing v to have degree higher than 2. Since a cycle is connected to begin with, no other

vertex besides v can belong to the same side of the split. This contradicts the requirement that each side of a split must have at least 2 vertices.

Let $\{x, u\}$ and $\{y, v\}$ be two disjoint edges crossing a split in a cycle with at least 5 vertices (with x and y on one side of the split and u and v on the other side). Since the edges crossing a split must induce a complete bipartite graph, the edges $\{x, v\}$ and $\{y, u\}$ must also be present in the graph. Therefore, the graph must have a cycle of size 4 as a subgraph, which is not the case for any cycle with at least 5 vertices. \square

THEOREM 1 (*Reduced split tree characterization of general cactus graphs*). *A graph G with the reduced split decomposition tree (T, \mathcal{F}) is a cactus graph if and only if*

1. \mathcal{F} consists of stars and polygons of size 3 or at least 5;
2. the center of every star-node in T is attached to either
 - (a) a leaf, or
 - (b) the center of another star-node, only as long as both star-nodes have exactly two extremities;
3. every extremity of star-nodes in T is attached to either
 - (a) a polygon,
 - (b) a leaf, or
 - (c) an extremity of another star-node;
4. no two polygons are adjacent.

Proof. $[\Rightarrow]$ Let G be a cactus graph with the corresponding reduced split decomposition tree T . We will show that T satisfies all the conditions of the theorem.

A *block* in a graph is a maximal 2-connected subgraph, or a subgraph formed by a bridge or an isolated vertex. In the case of a cactus graph, a *block* is either a cycle, or an edge that does not belong to any cycles. Let a *cluster* in a cactus graph be a maximal set of at least two blocks, all of which share exactly one vertex.

Furthermore, we define *external* blocks and clusters in cactus graphs in a similar way to external nodes in trees. A block in a cactus graph is called *external* if it shares exactly one vertex with other blocks. A cluster is *external* if it contains an external block. Note that this analogy captures the underlying tree structure for every cactus graph, where every cluster is a node in the tree, and two clusters share an edge iff they share a block. It is easy to see that the resulting graph is indeed connected and acyclic, with external clusters of the cactus graph corresponding to external nodes in the underlying tree.⁴ These definitions are illustrated in Figure 5.

We will proceed by induction on the number of clusters in G . For the base case, we consider cactus graphs with zero clusters. In this case, the graph consists of exactly one block, which can be one of the following (also shown in Figure 6):

- an edge: In this case, the corresponding reduced split decomposition tree is just an edge.
- a cycle of size 3 or at least 5: In this case, the corresponding reduced split decomposition tree is an internal node labeled with a cycle of the same size as the block, with a leaf hanging from every vertex of the label.⁵
- a cycle of size 4: In this case, the corresponding reduced split decomposition tree is two internal nodes, each labeled with a star with two extremities. The centers of the stars are connected by an edge, and the extremities of the stars are connected to leaves.

It is easy to confirm that the split decomposition trees above are both correct (by making sure their alternated paths match the adjacencies of the corresponding cluster) and reduced (by making sure the conditions of Theorem 1.3 are met). Furthermore, all the above split decomposition trees satisfy the conditions of this theorem, thus proving our base case.

For the inductive step, let G be a cactus graph with at least one cluster. First, we claim that G must have an external cluster, since the underlying tree must have a leaf. Let C be an external cluster including blocks $S_1 \cdots S_k$, $k > 1$, all of which share the vertex $v \in V(G)$. Finally, observe that at most one of the blocks is internal (the block representing the edge between the node corresponding to C and its neighbor in the underlying tree is internal). Without loss of generality, assume that S_k is the internal block, if one exists.

We split the cluster C from the rest of the cactus graph and create a graph-labeled tree \tilde{T} as follows (Figure 7):

- Create a star-node s with k extremities, one per block, respecting the blocks' circular permutation around v in the case of plane cacti. Attach the center of this star-node to a leaf corresponding to v .
- For every external block S_i , $i < k$,
 - If S_i is a single edge, attach its corresponding extremity to a leaf.
 - If S_i is a cycle of size 4, create two star-nodes with two extremities each, where the pair of extremities of each star represent two non-adjacent vertices in S_i , with the order of the appearance of the extremities respecting the planar embedding of the corresponding vertices in the case of plane cacti. Attach the two stars together at their centers, and connect the extremity corresponding to v with the extremity of s corresponding to S_i , and connect the other three extremities to leaves.
 - Otherwise, create an internal node labeled with S_i . Attach the copy of v in the label of the new internal node to the extremity of s corresponding to S_i . Attach every other vertex in the label of the new internal node to a leaf, representing the vertices of G in S_i , respecting

⁴However, this transformation is not bijective: There can be many cactus graphs with the same underlying tree structure. Therefore, it does not suffice for enumerative purposes.

⁵Note that this tree is indeed reduced: A cycle of size 3 is also a clique and therefore degenerate with respect to the split decomposition, and a cycle of size at least 5 is prime with respect to the split decomposition by Lemma 1.

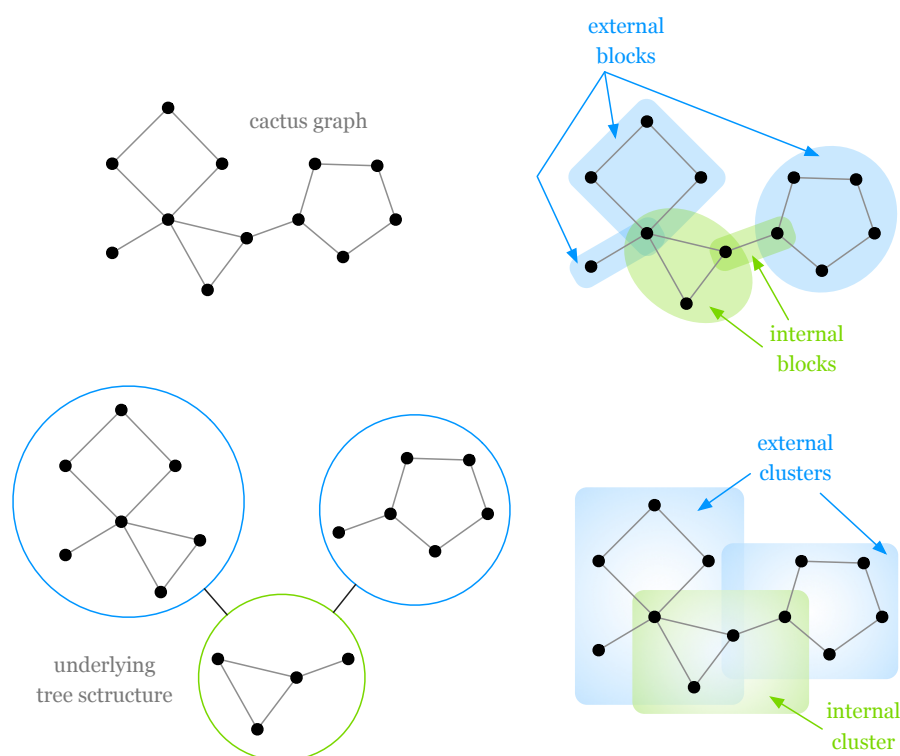


Figure 5. This figure illustrates the terminology introduced as part of the proof of Theorem 1. The top left graph is a cactus graph. The top right figure highlights all the blocks of this cactus graph, and the bottom right figure shows all its clusters. The bottom left figure shows the underlying tree structure of this cactus graph, where each node corresponds to a cluster and each edge represents a pair of clusters sharing a block. In all these figures, blue denotes *external* units (*i.e.* blocks, clusters, underlying tree nodes) whereas green indicates *internal* units.

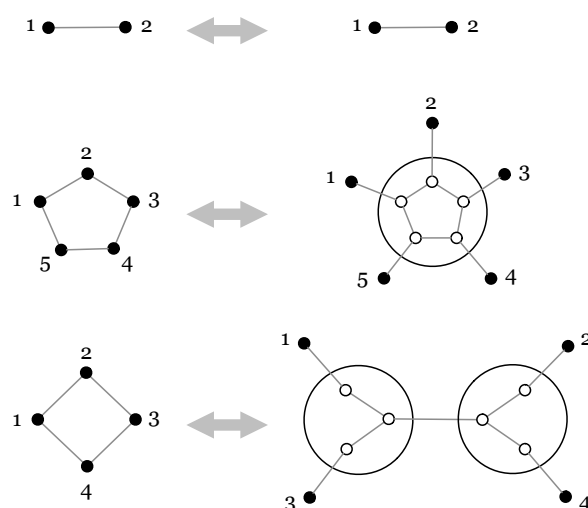


Figure 6. This figure shows the base case in part of the proof of Theorem 1, *i.e.* split decomposition trees of cactus graphs with exactly one block (and thus zero clusters). These graphs are partitioned into three disjoint subsets: an edge, a cycle of size 3 or at least 5, and a cycle of size 4. Each subset is treated differently as illustrated above. Note that the numbers here serve to specify the bijection between the vertices of each cactus graph and the leaves of its split decomposition tree. They do not serve to distinguish the vertices and do not imply that these graphs are labeled.

their planar ordering.

- Next, consider G' , the remainder of G excluding the external blocks $S_i, i < k$ but including S_k . This is a cactus graph with fewer clusters than G . Applying the inductive hypothesis, we obtain a reduced split-decomposition tree \tilde{T}' satisfying the conditions of this theorem. Finally, we attach \tilde{T}' to s by removing the leaf corresponding to v in $V(\tilde{T}')$ and connecting the edge incident to that leaf with the extremity of s corresponding to S_k .

Note that this procedure produces a split-decomposition tree \tilde{T} that satisfies the conditions of this theorem. In every step, the new star-node and the nodes corresponding to external blocks satisfy the conditions by construction, and the nodes within G' satisfy the conditions by the inductive hypothesis. It remains to check that nothing is violated upon attaching \tilde{T}' to an extremity of the new star-node. Since v belongs to a single external block in G' , its corresponding leaf in \tilde{T}' can only be attached to either a star extremity or a polygon; this is because the only time a leaf is attached to a star center is when the leaf corresponded to the common vertex of a cluster in the inductive step, in which case it was shared between more than one block. Therefore, only conditions 3 and 4 are relevant, both of which are satisfied when attaching the extremity of s to v 's neighbor in \tilde{T}' .

Finally, we need to show that this procedure preserves the adjacencies of G .

First, we show that every step in the procedure above preserves the adjacencies of G via alternated paths in \tilde{T} . At every step, any edge $e = \{x, y\} \in E(G)$ must belong to one of the following cases.

- e is incident to v : In this case, e must belong to a block $S_i, i \leq k$. There is an alternated path from v to the center of the star-node attached to v , then to the extremity corresponding to S_i and then out of the star-node. From there,
 - if S_i is an edge, we arrive at a leaf representing the other end of e , completing our alternated path;
 - if S_i is an external cycle of size 4, we next arrive at the extremity of a star-node and can continue to its center, then to the center of the neighboring star-node, then to the extremity of the current star-node corresponding to the other end of e , and finally to the leaf corresponding to the other end of e ;
 - finally, if S_i is an external cycle of size 3 or at least 5 or an internal cycle, we next arrive at an internal node labeled with either S_i or G' , both of which include e as an edge within the label. We can therefore continue the alternated path by taking the edge corresponding to e from within the label and then to the leaf corresponding to the other end of e .
- e is not incident to v : In this case, e belongs to either $S_i, i < k$ or G' . Note that in the first case S_i cannot be an edge since $v \notin e$.

- If S_i is a cycle of size 4, there is an alternated path representing e starting from the leaf corresponding to x to an extremity of the neighboring star, then to its center, then to the center of the next star, and finally to the extremity of the same star corresponding to the leaf y and lastly to the leaf corresponding to y itself.
- Otherwise, S_i is either cycle of size 3 or at least 5, or G' . In all these cases, e is present as within the label of an internal node, and thus there is an alternated path between the leaves representing x and y that takes that edge within a graph label corresponding to e .

Therefore, at every step, including the base case, for every edge in G there exists an alternated path in \tilde{T} .

In the other direction, we have to show that in every step, for every alternated path in \tilde{T} there is an edge in G . Take a pair of vertices $x, y \in V(G), \{x, y\} \notin E(G)$. It is easy to check that if x and y belong to the same block S_i of size at least 5, any path between the leaves representing them in \tilde{T} uses more than one edge from the internal node labeled with S_i . Furthermore, if x and y belong to the same block of size 4 or different blocks, any path connected the leaves representing them uses two edges from the same star-node.

Finally, note that the conditions of this theorem imply that T is *reduced*: Cunningham's theorem's (Section 1.3) criterion 1 for being reduced is satisfied by condition 1 of this theorem, criterion 2 by condition 4 of this theorem, and criterion 3 by conditions 2 and 3 of this theorem. Therefore, the procedure above produces a reduced split-decomposition tree, which is guaranteed to be unique by Cunningham's theorem. Thus, \tilde{T} and T are the same split-decomposition tree.

Therefore, the adjacencies of G correspond bijectively to the alternated paths of T . Furthermore, the plane embedding of a cactus graph can be preserved if the cyclic order of the new star-node in every step respects the plane embedding of the corresponding blocks in G .

[\Leftarrow] Let T be a reduced split-decomposition tree satisfying the conditions of this theorem. We should show that the original graph corresponding to T is indeed a cactus.

In the first part of the proof, we introduced a procedure for constructing the reduced split-decomposition tree of any cactus graph. The reverse of the same procedure can be used to build the original cactus graph from a split-decomposition tree. More specifically, pairs of star-nodes connected via their centers can be combined to form cycles of size 4, and star-nodes with a leaf hanging from their center can be combined with their neighbors in the reverse direction of the procedure as well. It is easy to check that as long as T has more than one internal node, it is possible to combine some internal nodes into one. Furthermore, these reverse steps maintain the invariant that in all intermediate split-decomposition trees, all the created labels are cactus graphs. Therefore, at the end of this procedure, a split decomposition

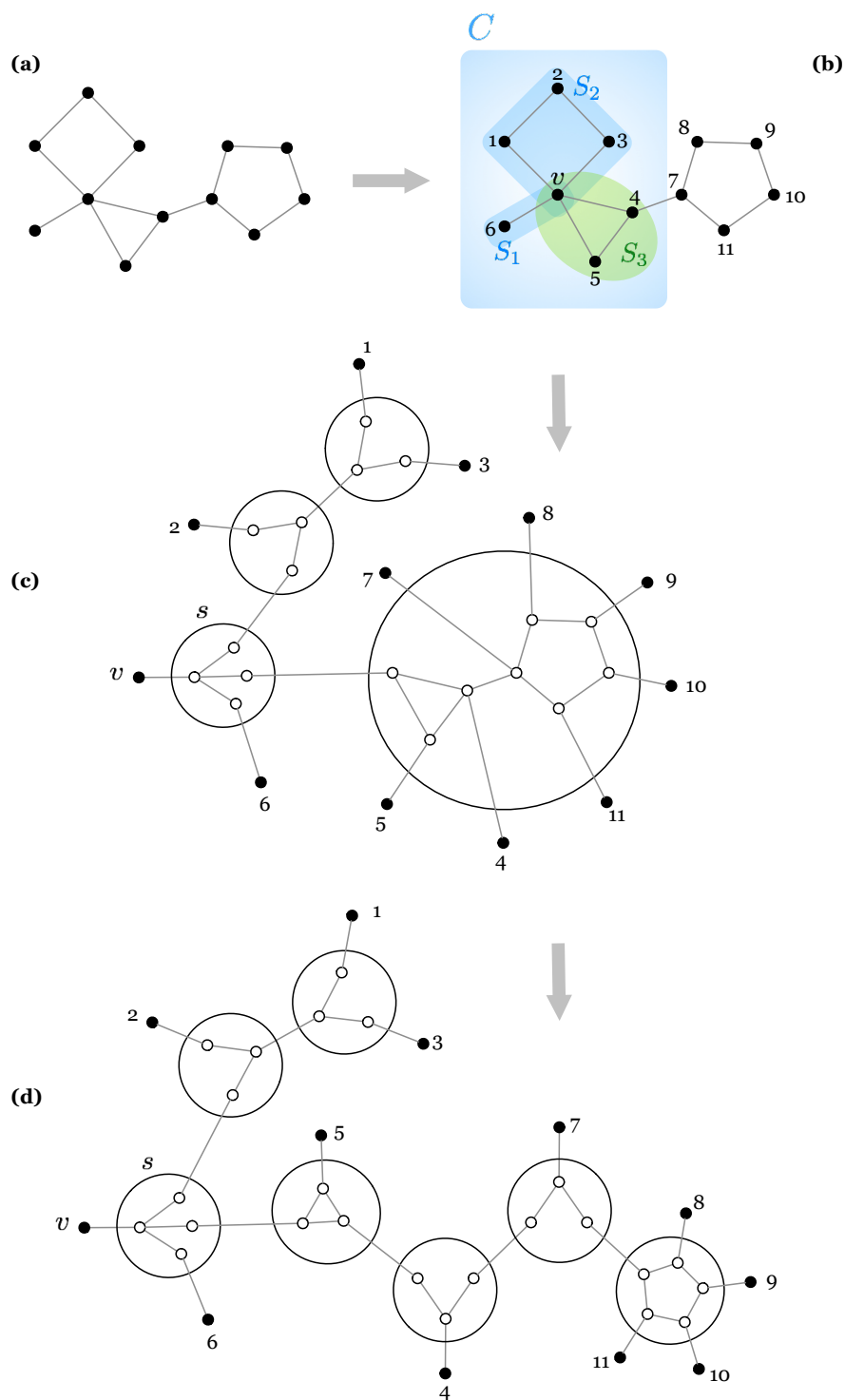


Figure 7. This figure shows the inductive step used in the proof of Theorem 1, *i.e.*. The inductive step begins by identifying an external cluster C , which consists of blocks $S_1 \cdots S_k$, all of which share a vertex v . The external blocks (here S_1 and S_2) are then separated via a star node s , the center of which is attached to v . The rest of the cactus graph forms a large super node hanging from s , and will be decomposed using the inductive hypothesis. Furthermore, the specific plane embedding of the original cactus graph around v can be respected by the choice of the order of the extremities of s . Similar to Figure 6, the numbers here are merely for illustrative purposes and do not imply that the cactus graph is labeled.

tree with a single internal node remains, the label of which must also be a cactus graph. \square

Theorem 1 gives a bijective characterization of cactus graphs in terms of *reduced* split-decomposition trees. In its proof we utilize Cunningham's Theorem, which guarantees uniqueness of the reduced split-decomposition tree for any graph. It turns out, however, that the characterization for general cactus graphs can be simplified to the form in Theorem 2, using a bijection between the reduced split-decomposition trees in Theorem 1 and a simpler, yet not reduced, set of split-decomposition trees.

THEOREM 2 (Split tree characterization of general cactus graphs). *Cactus graphs are in bijection with split-decomposition trees where*

- (a) *internal nodes are stars and polygons;*
- (b) *the centers of all star-nodes are attached to leaves;*
- (c) *the extremities of star-nodes are attached to leaves or polygons.*

Proof. Since the characterization in Theorem 1 describes a *reduced* tree, a direct application of Cunningham's Theorem guarantees a one-to-one correspondence between cactus graphs and those split decomposition trees. It suffices to show a bijection between the characterization given in Theorem 1 and the characterization in this Theorem. This is easily achieved by combining every pair of star-nodes with their centers attached into cycles of size 4, while leaving everything else intact. \square

3 Generating Cactus Grammars

In this section we provide guidelines on how to produce grammars⁶ for different varieties of cactus graphs. These guidelines do not assume familiarity with the technical details of the proofs in the previous sections and can be applied as a black box. The following subsections describe how a grammar can be obtained from a characterization of a split-decomposition tree, such as the one given in Theorem 2. For the black box version of the grammars, the reader can skip to Section 3.3.

3.1 Decomposable structures. In order to enumerate classes of split-decomposition trees, we use the framework of decomposable structures, described by Flajolet and Sedgewick [18]. We refer the reader to this book for details and outline below the basic idea.

We denote by \mathbb{Z} the combinatorial family composed of a single object of size 1, usually called *atom* (in our case, these

refer to a leaf of a split-decomposition tree, *i.e.*, a vertex of the corresponding graph).

Given two disjoint families \mathcal{A} and \mathcal{B} of combinatorial objects, we denote by $\mathcal{A} + \mathcal{B}$ the *disjoint union* of the two families and by $\mathcal{A} \times \mathcal{B}$ the *Cartesian product* of the two families.

We denote by $\text{SET}(\mathcal{A})$ (resp. $\text{SET}_\Omega(\mathcal{A})$ where $\Omega \subseteq \mathbb{Z}_{\geq 0}$) the family defined as all sets (resp. sets of sizes that belong to Ω) of objects from \mathcal{A} ⁷. In a similar fashion, we define $\text{SEQ}(\mathcal{A})$ (resp. $\text{SEQ}_\Omega(\mathcal{A})$) as the family of all (ordered) sequences of objects from \mathcal{A} (resp. a sequence the number of objects in which belongs to Ω). Finally, we define cycles $\text{CYC}(\mathcal{A})$, undirected sequences $\text{USEQ}(\mathcal{A})$, and undirected cycles $\text{UCYC}(\mathcal{A})$ similarly.

3.2 Split-decomposition trees expressed symbolically.

While approaching graph enumeration from the perspective of tree decomposition is not a new idea (the recursively decomposable nature of trees makes them well suited to enumeration), Chauve *et al.* [11] brought specific focus to Cunningham's split-decomposition. The same framework was used by Bahrani and Lumbroso [2] and is the starting point of this paper, so we briefly outline their method here.

Example. Let us consider the split-decomposition tree drawn in Figure 7 (d), and illustrate how it can be expressed recursively as a rooted tree.

Suppose the tree is rooted at vertex v . Assigning a root immediately defines a direction for all tree edges, which can be thought of as oriented away from the root. Starting from the root, we can set out to traverse the tree in the direction of the edges, one internal node at a time.

We start at the root, vertex v . The first internal node we encounter is a star node, s , and since we are entering it from the star's center, we have to describe what is on each of its three extremities.

Remark. If we are considering a *plane* cactus graph, the cyclic permutation of the edges incident to v in the original graph defines a cyclic permutation on the extremities of s , so these extremities can be described as a *cycle*. On the other hand, if we are considering a *free* cactus graph that is not embedded in the plane, there is no imposed ordering on the extremities of s . In that sense, the extremities of s are indistinguishable and can therefore be described as a *set*. These differences are what differentiates the templates in Section 3.3, more specifically the first row of Table 2.

Next, we will visit the extremities of s (in counter-clockwise order for easier referencing). On the first extremity, there is a leaf, 6, which concludes our journey in that direction; on each of the other two extremities, there is another split-decomposition subtree, which we can continue to explore.

⁶Grammars are a standard concept in computer science, since they are used to specify programming languages. Although we assume the semantics of these grammars is straightforward, we briefly recall the various operators in Subsection 3.1.

⁷Note the SET operator refers to multisets in the unlabeled world and sets in the labeled world.

Symbol	Explanation
\mathcal{P}	a polygon-node entered from one of its vertices (and missing the corresponding subtree)
\mathcal{S}_C	a star-node entered through its <i>center</i> (and missing the corresponding subtree)
\mathcal{S}_X	a star-node entered through one of its (at least two) <i>extremities</i> (and missing the corresponding subtree)
\mathcal{Z}	a leaf of the split-decomposition tree (an atom with unit size)
\mathcal{T}_P	a split-decomposition tree rerooted at a polygon-node (all subtrees are present)
\mathcal{T}_S	a split-decomposition tree rerooted at a star-node (all subtrees are present)
\mathcal{T}_{SP}	a split-decomposition tree rerooted at an <i>edge</i> connecting a star-node to a polygon-node (the edge can either connect the polygon-node to the star-node's center or an extremity; the edge accounts for one subtree of the polygon-node and one subtree of the star-node)

Table 1. The main symbols used to define the split-decomposition tree of the cactus graphs. Refer to Subsection 3.2 for details on the terminology, and to Subsection ?? for details on the dissymmetry theorem, from which all the rerooted trees, denoted by \mathcal{T}_ω , come from.

The second extremity of s leads to a split-decomposition subtree, of which the first internal node we encounter is labeled by a polygon of size 3. Continuing the traversal through this polygon, we need to describe what hangs from each vertex of the polygon. The exploration continues recursively through all the split-decomposition subtrees hanging from the vertices of this polygon until leaves are reached.

Remark. Once again, there is a distinction between the plane and non-plane cases. In the case of a *plane* cactus graph, each vertex of this polygon comes with a cyclic permutation of its edges, which uniquely determines the “next” vertex in the polygon, thus imposing a direction with respect to which the polygon can be traversed. Therefore, the split-decomposition subtrees hanging from the remaining vertices of the current polygon can be described as a *sequence*. On the other hand, in the case of a *free* cactus graph, there is no such imposed direction, and the split-decomposition subtrees can be described as an *undirected sequence*.

Remark. Note that unlike the extremities of a star, which are completely indistinguishable in the non-plane setting, the vertices of a polygon come with a cyclic ordering defined by the edges regardless of plane embedding, and therefore form an undirected sequence rather than a set. This distinction is reflected in the third row of Table 2.

The split-decomposition subtree hanging from the third extremity of s is traversed recursively in a similar manner, covering the whole tree.

Grammar description. The characterization developed in Theorem 2 can be translated to a symbolic grammar following the same approach as outlined in the example above. As an example, consider the class of plane, rooted, pure 5-

cacti. To describe the split-decomposition tree corresponding to this class symbolically, we must give a set of rules that prescribe the possible neighbors for each type of internal node.

Let's consider the rule for star nodes. First, assume like at the beginning of our example, that we enter a star node through its center; we have to describe what the extremities can be connected to.

According to Cunningham's Theorem, we know that there are at least two extremities (since every non-leaf node has degree at least three). Furthermore, by Theorem 2, we know that star extremities can be connected to either leaves or polygons. More specifically, connecting a star extremity to a leaf corresponds to a pendant edge in the original graph, whereas connecting a star extremity to an internal node labeled by a polygon of size k corresponds to a polygon of that size in the original graph. We call \mathcal{S}_C a split-decomposition tree that is traversed starting at a star-node entered through its center. We have

$$\mathcal{S}_C = \text{CYC}_{\geq 2}(\mathcal{P})$$

because indeed, we have at least two extremities, which have a cyclic order since we are describing a *plane* cactus graph, and each of these extremities must lead to an internal node labeled by a polygon, \mathcal{P} .

A split decomposition subtree rooted at an internal node labeled by a polygon \mathcal{P} entered from one of its vertices can in turn be described by a symbolic equation

$$\mathcal{P} = \text{SEQ}_4(\mathcal{Z} + \mathcal{S}_X),$$

where \mathcal{Z} denotes a leaf (an atom of size 1) and \mathcal{S}_X denotes a star node entered through one of its extremities. Indeed,

since we are in a *plane* setting, the split-decomposition subtrees hanging from the remaining vertices of the current polygon can be described as a *sequence*. The cardinality constraint captures the *pure* aspect of these cacti. Finally, by Theorem 2, internal nodes with polygonal labels can be attached to leaves or star extremities, which translates into the disjoint union term $\mathcal{Z} + \mathcal{S}_X$.

The rule for a star node entered through its extremity is derived in a similar manner.

Conventions. As explained above, we use rather similar notations to describe the combinatorial classes that arise from decomposing split-decomposition trees. These notations are summarized in Table 1.

Terminology. In the rest of this paper, we describe the combinatorial class \mathcal{S}_X as representing a “*a star-node entered through an extremity*”, but others may have alternate descriptions: such as “*a star-node linked to its parent by an extremity*”; or such as Iriza [28], “*a star-node with the subtree incident to one of its extremities having been removed*”—all these descriptions are equivalent (but follow different viewpoints).

3.3 General Template. The variations of cactus graphs supported by our grammars include *plane* versus *non-plane* and *rooted* versus *unrooted*. These differences gives rise to different operators in the grammars, which we will outline below.

Another variation in types of cacti is whether the vertices are labeled. It turns out *labeled* versus *unlabeled* cacti have identical grammars but are distinguished by the way their grammars are translated to generating function equations, with exponential generating functions for labeled cacti and ordinary generating functions for unlabeled ones.

A generalization of *pure* versus *mixed* cacti is supported in the form of arbitrary constraints on allowable cycle sizes. More specifically, in our grammars Ω refers to a set of integers each greater than 1, such that the size of each cycle in the cactus graph must belong to that set. For example, in the case of *pure* cacti, Ω contains a single element, and setting $\Omega = \{2\}$ gives the grammar for trees. Furthermore, the grammar for *mixed* cacti can be obtained by letting Ω be the set of all integers greater than 1. Finally, we use the notation $\Omega - 1 := \{x - 1 \text{ for all } x \in \Omega\}$.

The grammars for all the various cactus graphs mentioned above roughly follow the same template. We outline this template below, highlighting the minor differences from case to case. The following three equations are common to all grammars, with XXX, YYY, and ZZZ replaced by the ap-

propriate operator as listed in Table 2.

$$\begin{aligned}\mathcal{S}_C &= \text{XXX}_{\geq 2}(\mathcal{P}) \\ \mathcal{S}_X &= \mathcal{Z} \times \text{YYY}_{\geq 1}(\mathcal{P}) \\ \mathcal{P} &= \text{ZZZ}_{\Omega-1}(\mathcal{Z} + \mathcal{S}_X)\end{aligned}$$

The class of rooted cactus graphs \mathcal{G}_\bullet can be specified as

$$\mathcal{G}_\bullet = \mathcal{Z} \times (\mathcal{P} + \mathcal{S}_C).$$

Furthermore, the class of unrooted cactus graphs, \mathcal{G} , can be specified as

$$\begin{aligned}\mathcal{G} &= \mathcal{T}_S + \mathcal{T}_P - \mathcal{T}_{SP} \\ \mathcal{T}_S &= \mathcal{Z} \times \mathcal{S}_C \\ \mathcal{T}_P &= \text{WWW}_{\Omega}(\mathcal{Z} + \mathcal{S}_X) \\ \mathcal{T}_{SP} &= \mathcal{P} \times \mathcal{S}_X,\end{aligned}$$

with WWW replaced by the appropriate operators according to Table 2.

	plane	free
XXX	CYC	SET
YYY	SEQ	SET
ZZZ	SEQ	USEQ
WWW	CYC	UCYC

Table 2. Guidelines for choosing operators in the template grammar for *plane* versus *free* cactus graphs.

Unrooting the grammar from the traversal of the tree.

The grammar template \mathcal{G}_\bullet corresponds to *rooted* cactus graphs, and results directly from applying the constraints on the split-decomposition tree—which we defined in Section 2—by using the same type of process as described earlier in Subsection 3.2. But obtaining grammars for the *unrooted* cactus graphs requires additional tools, as there are symmetries to account for, that depend on the structure of individual graphs.

There are at least two such *unrooting* techniques. We apply here the *dissymmetry theorem [for trees]*, from Bergeron *et al.* [5], with a proof in Drmota [16, §4.3.3, p. 293]. In the context of symbolically specified split-decomposition trees, Chauve *et al.* [11, §2.2 and §3] contain indications on how to apply the theorem concretely, with further remarks by Bahrani *et al.* [2, §1.6].

The dissymmetry theorem is relatively easy to apply, but the equation which has a subtraction is algebraic but not symbolic: the subtraction is meaningful when we are tabulating the enumeration, but it has no combinatorial interpretation. This means such grammars cannot benefit from the direct application of some of the most powerful “last kilometer” results from analytic combinatorics transfer theorems: asymptotics, parameter analysis, random generation (and so empirical analysis).

An alternate tool to unroot combinatorial classes, *cycle-pointing* [7], does not have this issue. It is a combinatorial operation (rather than an algebraic one), and preserves the transfer theorems, in particular the results on random generation—the recursive method [29, 19, 14, 33] and Boltzmann sampling [17, 32, 6]. It involves a more complex decomposition of the grammar and finer understanding of the symmetries of the structure. Iriza [28] has applied it to the distance-hereditary and 3-leaf power grammars of Chauve *et al.* [11]. Iriza's work was the basis of the random samplers used for Figure 1.

3.4 Example We state a grammar for plane unrooted pure 5-cacti (\mathcal{G}_{pu5}) as an example:

$$\begin{aligned}\mathcal{G}_{\text{pu5}} &= \mathcal{T}_S + \mathcal{T}_P - \mathcal{T}_{SP} \\ \mathcal{T}_S &= \mathcal{Z} \times \mathcal{S}_C \\ \mathcal{T}_P &= \text{CYC}_{\{5\}}(\mathcal{Z} + \mathcal{S}_X) \\ \mathcal{T}_{SP} &= \mathcal{P} \times \mathcal{S}_X \\ \mathcal{S}_C &= \text{CYC}_{\geq 2}(\mathcal{P}) \\ \mathcal{S}_X &= \mathcal{Z} \times \text{SEQ}_{\geq 1}(\mathcal{P}) \\ \mathcal{P} &= \text{SEQ}_{\{4\}}(\mathcal{Z} + \mathcal{S}_X).\end{aligned}$$

A symbolic math package can be used to extract the exact enumeration of plane unrooted pure 5-cacti from this grammar in polynomial time (See Figure 8 for a sample implementation in Maple). The resulting counting sequence in this case indeed matches the sequence provided by Bona *et al.* [8]. The first few terms of the enumeration, *EIS* A054365, are listed below.

$$1, 1, 1, 3, 17, 102, 811, 6626, 58385, \\ 532251, 5011934, 48344880, \dots$$

3.5 Simplified grammars In this section, we list simplified versions of the grammars produced by the template in the previous section. The previous template is informative in its maintaining of a general common structure across different sub-classes of cactus graphs. More specifically, in the previous template, the differences between grammars of *similar* classes are relatively small. The following simplified grammars, however, are more concise and therefore amenable to further analysis, such as the extraction of asymptotics and the study of parameters.

- *free, rooted* cactus graphs (denoted by \mathcal{G}_{fr}) admit the following recursive grammar:

$$\mathcal{G}_{\text{fr}} = \mathcal{Z} \times \text{SET}_{\geq 1}(\text{USEQ}_{\Omega-1}(\mathcal{Z} + \mathcal{G}_{\text{fr}}))$$

- *plane, rooted* cactus graphs (denoted by \mathcal{G}_{pr}) can be

specified as

$$\begin{aligned}\mathcal{G}_{\text{pr}} &= \mathcal{Z} \times \text{CYC}_{\geq 1}(\text{SEQ}_{\Omega-1}(\mathcal{Q})) \\ \mathcal{Q} &= \mathcal{Z} \times \text{SEQ}(\text{SEQ}_{\Omega-1}(\mathcal{Q}))\end{aligned}$$

- *free, unrooted* cactus graphs (denoted by \mathcal{G}_{fu}) can be specified as

$$\begin{aligned}\mathcal{G}_{\text{fu}} &= \text{UCYC}_{\Omega}(\mathcal{Q}) - \mathcal{Q} \times \text{USEQ}_{\Omega-1}(\mathcal{Q}) + \mathcal{Q} - \mathcal{Z} \\ \mathcal{Q} &= \mathcal{Z} \times \text{SET}(\text{USEQ}_{\Omega-1}(\mathcal{Q}))\end{aligned}$$

- *plane, unrooted* cactus graphs (denoted by \mathcal{G}_{pu}) can be specified as

$$\begin{aligned}\mathcal{G}_{\text{pu}} &= \text{CYC}_{\Omega}(\mathcal{Q}) + \mathcal{Z} \times \text{CYC}_{\geq 2}(\text{SEQ}_{\Omega-1}(\mathcal{Q})) \\ &\quad - (\mathcal{Q} - \mathcal{Z}) \times \text{SEQ}_{\Omega-1}(\mathcal{Q}) \\ \mathcal{Q} &= \mathcal{Z} \times \text{SEQ}(\text{SEQ}_{\Omega-1}(\mathcal{Q}))\end{aligned}$$

With these grammars, and using existing frameworks and tools from analytic combinatorics, many results on cactus graphs are easily accessible: full exact enumeration (see Figure 8), asymptotics, and exhaustive and random generation [17].

4 Conclusion

This paper continues the line of work on split-decomposition from an analytic combinatorial perspective, initiated by Chauve *et al.* [10, 11] and which we have also explored previously, in the context of subclasses of distance-hereditary graphs defined by forbidden subgraphs [2]. While these works focused on graphs that were fully decomposable by the split-decomposition, we tentatively investigate having a certain type of prime nodes: cycles.⁸

In this paper, we looked at the split-decomposition of cactus graphs, and provided a characterization of the corresponding split-decomposition trees, as Gioan and Paul [25] did for cographs, 3-leaf power graphs and distance-hereditary graphs. We then leveraged this characterization, as Chauve *et al.* [10, 11] did, to obtain a symbolic specification for cactus graphs—and using the same too [15].

Furthermore, because cactus graphs are akin to trees, of which the nodes have been replaced by cycles, the structure

⁸Even though this class of prime nodes may seem trivial to consider, consider some other examples of graph class that decomposes with a well-defined set of prime nodes. For instance, *circle graphs* [34] have a subset of circle graphs as prime nodes, therefore likely requiring some careful substitution and mutual recursion. Another example are parity graphs, which admit as prime node any bipartite graph; we have known how to tabulate the enumeration of bipartite graph for a long time [26], but it is only recently that Gainer-Dewar and Gessel [24] exactly enumerated bipartite graphs with some preservation of the symmetries, but using sophisticated algebraic tools that do not seem to translate easily to symbolic specifications, or at least not at first glance.

```

> with(combstruct):
> COMMON := {
  P   = Sequence(Union(Z, Sx), card = 4),
  Sx  = Prod(Z, Sequence(P, card >= 1)),
  Sc  = Cycle(P, card >= 2),
  Z   = Atom
}:
> FIVE_CACTUS_UNROOTED_PLANE := COMMON union {
  TPSu = Prod(P, Sx),
  TS   = Prod(Sc, Z),
  TP   = Cycle(Union(Z, Sx), card = 5)
}:
> N := 100:
> GF_TPSu := add(count([TPSu, FIVE_CACTUS_UNROOTED_PLANE, unlabeled], size = n)*x^n, n = 1 .. N):
> GF_TS   := add(count([TS, FIVE_CACTUS_UNROOTED_PLANE, unlabeled], size = n)*x^n, n = 1 .. N):
> GF_TP   := add(count([TP, FIVE_CACTUS_UNROOTED_PLANE, unlabeled], size = n)*x^n, n = 1 .. N):
> G := GF_TS + GF_TP - GF_TPSu:
> [seq(coeff(G, x, i), i = 1 .. 40)]
[0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 17, 0, 0, 0, 102, 0, 0, 0, 811, 0, 0, 0, 6626, 0, 0, 0, 58385, 0, 0, 0, 532251, 0, 0, 0]

```

Figure 8. The enumeration of different varieties of cactus graphs can be derived using our grammars and Maple. This figure shows a Maple worksheet for computing the enumeration of unrooted, unlabeled, pure, plane 5-cacti.

of their split-decomposition tree is very similar to that of the cactus graph it decomposes. It was thus possible to define a convenient template of grammars to symbolically describe any class of rooted cactus graph.

This offers perhaps the possibility of more systematic results on cactus graphs. Perhaps some general theorem on the asymptotic that depends only on the allowed size of cycle. We also hope that the convenient format of the grammars will make certain other applications from analytic combinatorics more accessible. For instance, the random graphs in Figure 1 were produced by our rooted cactus graph sampler.

Along the lines of the split decomposition, it would be interesting to consider more general graphs and split-decomposition trees with other classes of prime nodes, especially in the case of families of graphs that are less "tree-like" than cacti. A good starting point can be considering split-decomposition trees where the labels are stars, cliques, or prime cycles (a superset of distance-hereditary graphs).

Another approach that might hold some interest: define and enumerate graph classes according to the subset of allowable prime nodes, from among specific graph classes—though it is not clear which graph classes those could be.

Acknowledgements

We would like to thank the anonymous reviewers for critically reading the manuscript and making suggestions that led to significant improvements to the content and clarity of the paper.

References

- [1] Murat Arcak. Diagonal stability on cactus graphs and application to network stability analysis. *IEEE Transactions on Automatic Control*, 56(12):2766–2777, 2011.
- [2] Maryam Bahrani and Jérémie Lumbroso. Enumerations, forbidden subgraph characterizations, and the split-decomposition. *arXiv preprint arXiv:1608.01465*, 2016.
- [3] Julien Baste, Didem Gözüpek, Christophe Paul, Ignasi Sau, Mordechai Shalom, and Dimitrios M. Thilikos. Parameterized complexity of finding a spanning tree with minimum reload cost diameter. *arXiv preprint arXiv:1703.01686*, 2017.
- [4] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial species and tree-like structures*, volume 67. Cambridge University Press, 1998.
- [5] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial Species and Tree-like Structures*, volume 67 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1998.
- [6] Olivier Bodini, Jérémie Lumbroso, and Nicolas Rolin. Analytic samplers and the combinatorial rejection method. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, pages 40–50. Society for Industrial and Applied Mathematics, 2015.
- [7] Manuel Bodirsky, Éric Fusy, Mihyun Kang, and Stefan Vigerske. Boltzmann samplers, pólya theory, and cycle pointing. *CoRR*, abs/1003.4546, 2010.
- [8] Miklós Bóna, Michel Bousquet, Gilbert Labelle, and Pierre Leroux. Enumeration of m -ary cacti. *Advances in Applied Mathematics*, 24(1):22–56, 2000.
- [9] L. Sunil Chandran, Anita Das, Davis Issac, and Erik Jan van Leeuwen. Algorithms and bounds for very strong rainbow coloring. *arXiv preprint arXiv:1703.00236*, 2017.
- [10] Cédric Chauve, Éric Fusy, and Jérémie Lumbroso. An enumeration of distance-hereditary and 3-leaf power graphs. Preprint presented at ICGT 2014, 2014.
- [11] Cédric Chauve, Éric Fusy, and Jérémie Lumbroso. An exact enumeration of distance-hereditary graphs. In *Proceedings of the Fourteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2017, Barcelona, Spain, Hotel Porta Fira, January 16–17, 2017*, pages 31–45, 2017.

- [12] William H. Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic Discrete Methods*, 3(2):214–228, 1982.
- [13] Kalyani Das. Cactus graphs and some algorithms. *arXiv preprint arXiv:1408.4005*, 2014.
- [14] Alain Denise and Paul Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science*, 218(2):233–248, 1999.
- [15] Michael Drmota. Systems of functional equations. *Random Structures and Algorithms*, 10(1-2):103–124, 1997.
- [16] Michael Drmota. Trees. In Miklos Bona, editor, *Handbook of Enumerative Combinatorics*, pages 281–334. CRC Press, 2015.
- [17] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann Samplers for the Random Generation of Combinatorial Structures. *Combinatorics, Probability and Computing*, 13(4-5):577–625, 2004. Special issue on Analysis of Algorithms.
- [18] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.
- [19] Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem. A Calculus for the Random Generation of Labelled Combinatorial Structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.
- [20] George W. Ford, Robert Z. Norman, and George E. Uhlenbeck. Combinatorial problems in the theory of graphs. II. *Proceedings of the National Academy of Sciences*, 42(4):203–208, 1956.
- [21] George W. Ford and George E. Uhlenbeck. Combinatorial problems in the theory of graphs. I. *Proceedings of the National Academy of Sciences*, 42(3):122–128, 1956.
- [22] George W. Ford and George E. Uhlenbeck. Combinatorial problems in the theory of graphs. III. *Proceedings of the National Academy of Sciences*, 42(8):529–535, 1956.
- [23] George W. Ford and George E. Uhlenbeck. Combinatorial problems in the theory of graphs. IV. *Proceedings of the National Academy of Sciences*, 43(1):163–167, 1957.
- [24] Andrew Gainer-Dewar and Ira M. Gessel. Enumeration of Bipartite Graphs and Bipartite Blocks. *The Electronic Journal of Combinatorics*, 21(2):P2–40, 2014.
- [25] Emeric Gioan and Christophe Paul. Split decomposition and graph-labelled trees: Characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Applied Mathematics*, 160(6):708–733, 2012. Fourth Workshop on Graph Classes, Optimization, and Width Parameters Bergen, Norway, October 2009 Bergen (GROW) 09.
- [26] Phil Hanlon. The enumeration of bipartite graphs. *Discrete Mathematics*, 28(1):49–57, 1979.
- [27] Frank Harary and George E. Uhlenbeck. On the number of husimi trees i. *Proceedings of the National Academy of Sciences*, 39(4):315–322, 1953.
- [28] Alexander Iriza. Enumeration and random generation of unlabeled classes of graphs: A practical study of cycle pointing and the dissymmetry theorem. *arXiv preprint arXiv:1511.06037*, 2015.
- [29] Albert Nijenhuis and Herbert S. Wilf. *Combinatorial Algorithms*. Academic Press, 1975.
- [30] Tetsuo Nishi and Leon O. Chua. Topological criteria for nonlinear resistive circuits containing controlled sources to have a unique solution. *IEEE transactions on circuits and systems*, 31(8):722–741, 1984.
- [31] Benedict Paten, Mark Diekhans, Dent Earl, John St John, Jian Ma, Bernard Suh, and David Haussler. Cactus graphs for genome comparisons. *Journal of Computational Biology*, 18(3):469–481, 2011.
- [32] Carine Pivoteau, Bruno Salvy, and Michele Soria. Algorithms for combinatorial structures: Well-founded systems and newton iterations. *Journal of Combinatorial Theory, Series A*, 119(8):1711–1773, 2012.
- [33] Yann Ponty, Michel Termier, and Alain Denise. Genrgens: software for generating random genomic sequences and structures. *Bioinformatics*, 22(12):1534–1535, 2006.
- [34] Jeremy Spinrad. Recognition of circle graphs. *Journal of Algorithms*, 16(2):264–282, 1994.
- [35] George E. Uhlenbeck. Gibbs lecture, December 1950.