Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 20, no. 1, pp. 3–22 (2016)
DOI: 10.7155/jgaa.00382

Dichotomy Theorems for Homomorphism Polynomials of Graph Classes

Christian Engels[1]

[1]Department of Computer Science
Saarland University, Germany

Abstract

In this paper, we will show dichotomy theorems for the computation of polynomials corresponding to evaluation of graph homomorphisms in Valiant's model. We are given a fixed graph H and want to find all graphs, from some graph class, homomorphic to this H. These graphs will be encoded by a family of polynomials.

We give dichotomies for the polynomials for cycles, cliques, trees, outerplanar graphs, planar graphs and graphs of bounded genus (for different definitions of geni).

Submitted:	Reviewed:	Revised:	Accepted:	Final:
March 2015	August 2015	November 2015	November 2015	January 2016
		Published:		
		February 2016		

Article type:
Regular paper

Communicated by:
M. S. Rahman and E. Tomita

This work was first published in WALCOM under [12]

E-mail address: engels@cs.uni-saarland.de (Christian Engels)

1 Introduction

Graph homomorphisms are studied because they give important generalizations of many natural questions (k-coloring, acyclicity, binary CSP and many more cf. [18]). One of the first results, given by Hell and Nešetřil [17], was on the decision problem where they gave a dichotomy. The exact result was, that deciding if there exists a homomorphism from some graph G to a fixed undirected graph H is polynomial time computable if H is bipartite and NP-complete otherwise. A different side of graph homomorphisms was looked at by Chekuri and Rajaraman [4], Dalmau et al. [7], Freuder [13] and finally Grohe [15]. They studied the following: Given a graph in a restricted graph class $\mathcal{G}$, decide if G is homomorphic to a given graph H. Later, focus shifted onto the counting versions of these two sides where we have to count the number of homomorphisms. Dyer and Greenhill [11] modified the first problem for the counting case. Here they have a fixed graph H and count all homomorphisms from a given graph G to H. Dalmau and Jonsson [6] modified a version of the second problem. They looked at the complexity of counting the homomorphisms from a fixed graph G in some graph class $\mathcal{G}$ to an input graph H. The first problem was extended by Bulatov and Grohe [1] to graphs with multiple edges. They also notice some interesting connections to statistical physics and constraint satisfaction problems. A good introduction to the history of graph homomorphism was written by Grohe and Thurley [16] and research on these topics continues even today with two noticeable papers being the works by Goldberg et al. [14] and by Cai et al. [3].

However, the arithmetic circuit complexity is seldom studied. Here we want to study the complexity of the following problem. Given a fixed graph H and a graph class $\mathcal{G}$ what graphs $G \in \mathcal{G}$ are homomorphic to H. This can be seen as a generalization to the first decision problem. The study of VNP complete problems and the arithmetic world was started in the seminal paper by Valiant [23]. In this world, we look at the complexity of representing a family of polynomials using a family of arithmetic circuits. Recently, a dichotomy for graph homomorphisms was shown by Rugy-Altherre [8]. Here a graph is encoded by a product of edge variables and sets of graphs as sums over these products. This is known as generating function and a detailed definition will be provided in Section 2. However, his result was for the first side of the graph homomorphism problem.

In this paper we look at a version of the original problem in the arithmetic circuit world. We will encode all graphs $G \in \mathcal{G}$ for some graph class $\mathcal{G}$ where G is homomorphic to a given graph H. While we could not get a general theorem as for the different problem in [6], we show multiple hardness proofs for some classes. We will look at cycles, cliques, trees, outerplanar graphs, planar graphs and graphs of bounded genus (orientable, non-orientable and Euler genus).

Recently, homomorphism polynomials in a different form are even used for giving natural characterizations of VP independent of the circuit definition [10]. Durand et al. showed that all homomorphisms from a balanced binary tree with n leaves to a complete graph on n^6 vertices on specific weights can encode every polynomial in VP. However, their polynomial differs from the one presented here. They looked at a polynomial encoding all homomorphism from a given graph G to a given graph H where they have weights on edges and vertices while our polynomial encodes graphs. While these two problems are very different, we can still see our results as showing that some straightforward candidates originating from the decision problem do not give a characterization of VP.

Section 2 gives a formal introduction to our model, related hard problems and states the problem precisely. We prove our dichotomies in Sections 3.1 to 3.6 where the constructions in Sections 3.4 to 3.6 build on each other. The construction in Section 3.3 will use a slightly different model as the other sections. We will give a brief introduction into concepts from graph genus in Section 3.6 but refer the reader to the textbook by Diestel [9].

2 Model and Definitions

Let us first give a brief introduction to the field of Valiant's classes. For further information the reader is referred to the textbook by Bürgisser [2]. In this theory, we are given an arithmetic circuit (a directed acyclic connected graph) with addition and multiplication gates over some field K. These gates are either connected to other gates or inputs from the set $K \cup X$ for some set of indeterminates X. At the top we have exactly one output gate. An arithmetic circuit computes a polynomial in $K[X]$ at the output gate in the obvious way.

As Valiant's model is non-uniform, a problem consists of families of polynomials. A p-family is a sequence of polynomials (f_n) over $K[X]$ where the number of variables is $q(n)$ for some polynomially bounded function $q(n)$ and the degree is bounded by some polynomial in n. Additionally the family of polynomials (f_n) should be computed by a family of arithmetic circuits (C_n) where f_n is computed by C_n for all n. Valiant's model focuses its study on p-families of polynomials.

We define $L(f)$ to be the number of gates for a minimal arithmetic circuit computing a given polynomial $f \in K[X]$. VP is the class of all p-families of polynomials where $L(f_n)$ is bounded polynomially in n. Let $q(n), r(n), s(n)$ be polynomially bounded functions. A p-family $(f_n) \in K[x_1, \dots, x_{q(n)}]$ is in VNP if there exists a family $(g_n) \in K[x_1, \dots, x_{r(n)}, y_1, \dots, y_{s(n)}]$ in VP such that

$$f(x_1, \dots, x_{q(n)}) = \sum_{\epsilon \in \{0,1\}^{s(n)}} g(x_1, \dots, x_{r(n)}, \epsilon_1, \dots, \epsilon_{s(n)}).$$

The classes VP and VNP are considered algebraic analogues to P and NP or more accurately #P. We can also define an algebraic version of AC_0, mentioned by Mahajan and Rao [19]. A p-family is in VAC_0 if there exists a family of arithmetic circuit of constant depth and polynomial size with unbounded fan-in that computes the family of polynomials.

The notion of a reduction in Valiant's model is given by p-projections. A p-family (f_n) is a p-projection of (g_n), written as $(f_n) \leq_p (g_n)$, if there exists a polynomially bounded function $q(n)$ such that for every n, $f(x_1, \ldots, x_{r(n)}) = g(a_1, \ldots, a_{q(n)})$ for some $a_i \in K \cup \{x_1, \ldots, x_{r(n)}\}$. Once we have a reduction, we get a notion of completeness in the usual way.

We will, however, use a different kind of reduction called a c-reduction. This is similar to a Turing reduction in the Boolean world. We define $L^g(f)$ as the number of gates for computing f where the arithmetic circuits are enhanced with an oracle gate for g. An oracle gate for the polynomial $g \in K[x_1, \ldots, x_{n'}]$ has as output $g(a_1, \ldots, a_{n'})$ where $a_1, \ldots, a_{n'}$ are the inputs to this gate. This allows us to evaluate g on $a_1, \ldots, a_{n'}$ in one step if we computed $a_1, \ldots, a_{n'}$ previously in our circuit.

We say (f_n) c-reduces to (g_n), written $(f_n) \leq_c (g_n)$, if there exists a polynomial p such that $L^{g_{p(n)}}(f)$ is bounded by some polynomial. This reduction, however, is only useful for VNP and not for VAC_0 and VP. In this paper we will exclusively deal with c-reductions for our VNP completeness results.

2.1 Complete Problems

We continue with the basic framework of graph properties. In the following K will be a field.

Definition 1 *Let X be a set of indeterminates. Let $\mathcal{E}$ be a graph property, that is, a class of graphs which contains with every graph also all of its isomorphic copies. Let $G = (V, E)$ be an edge weighted, undirected graph with a weight function $w : E \to K \cup X$. We extend the weight function by $w(E') := \prod_{e \in E'} w(e)$ to subsets $E' \subseteq E$.*

The generating function $\mathrm{GF}(G, \mathcal{E})$ *of the property $\mathcal{E}$ is defined as*

$$\mathrm{GF}(G, \mathcal{E}) := \sum_{E' \subseteq E} w(E')$$

where the sum is over all subsets E' such that the subgraph (V, E') of G has property $\mathcal{E}$.

The reader should notice that the subgraph still contains all vertices and just takes a subset of the edges.

In the following, let G be a graph and let $X = \{x_e \mid e \in E\}$. We label each edge e by the indeterminate x_e. We conclude by stating some basic VNP-complete problems. Proofs of these facts can be found in the textbook by Bürgisser [2].

Theorem 1 ([2]) $\mathrm{GF}(K_n, \mathcal{UHC}_n)$ *is VNP-complete where $\mathcal{UHC}_n$ is the set of all Hamiltonian cycles in K_n.*

Theorem 2 ([2]) *Let $\mathcal{CL}$ be the set of all cliques. Meaning, the set of all graphs, where one connected component is a complete graph and each of the remaining connected components consist of one vertex only. The family* $\mathrm{GF}(K_n, \mathcal{CL})$ *is VNP-complete.*

Theorem 3 **([2])** *Let $\mathcal{M}$ be the set of all graphs where all connected components have exactly two vertices. The family* $\mathrm{GF}(K_n, \mathcal{M})$ *is VNP-complete.*

This polynomial gives us all perfect matchings in a graph. It is well known that the original VNP-complete problem, the permanent, is equal to $\mathrm{GF}(K_{n,n}, \mathcal{M})$ for bipartite graphs which is a projection of $\mathrm{GF}(K_{n^2}, \mathcal{M})$.

2.2 The problem and related definitions

We now formulate our problem. Let G, H be undirected graphs. We will generally switch freely between having the variable indexed by either edges (x_e) or vertices ($x_{i,j}$ for $i, j \in V$). We let x_j correspond to the self-loop at vertex j.

A homomorphism from $G = (V, E)$ to $H = (V', E')$ is a mapping $f : V \to V'$ such that for all edges $\{u, v\} \in E$ there exist an edge $\{f(u), f(v)\} \in E'$. We can define the corresponding generating function as follows.

Definition 2 *Let $\mathcal{H}_H$ be the property of all connected graphs homomorphic to a fixed H. We denote by $\mathcal{F}^{H,n}$ the generating function* $\mathcal{F}^{H,n} := \mathrm{GF}(K_n, \mathcal{H}_H)$.

We can state now the first dichotomy theorem.

Theorem 4 **([8])** *If H has a loop or no edges, $\mathcal{F}^{H,n}$ is in VAC_0 and otherwise it is VNP-complete under c-reductions.*

Instead of looking at all graphs, we want to look at a restricted version. What happens if we do not want to find every graph homomorphic to a given H but every *cycle* homomorphic to a given H? We state our problem in the next definitions.

Definition 3 *Let $\mathcal{E}_n$ be a graph property. Then $\mathcal{F}^{H,n}_{\mathcal{E}_n}$ is the generating function for all graphs in $\mathcal{E}_n$ on n vertices homomorphic to a fixed graph H.*

Definition 4 *We define the following graph polynomials.*

- $\mathcal{F}^{H,n}_{cycle_n}$ *where $cycle_n$ is the property where one connected component is a cycle and the others are single vertices in a graph of size n.*

- $\mathcal{F}^{H,n}_{clique_n}$ *where $clique_n$ is the property where one connected component is a clique and the others are single vertices in a graph of size n.*

- $\mathcal{F}^{H,n}_{trees_n}$ *where $trees_n$ is the property where one connected component is a tree and the others are single vertices in a graph of size n.*

- $\mathcal{F}^{H,n}_{outerplanar_n}$ *where $outerplanar_n$ is the property where one connected component is a outerplanar graph and the others are single vertices in a graph of size n.*

- $\mathcal{F}^{H,n}_{planar_n}$ *where $planar_n$ is the property where one connected component is a planar graph and the others are single vertices in a graph of size n.*

- $\mathcal{F}^{H,n}_{genus(k),n}$ *where genus(k),n is the property where one connected component has orientable genus k and the others are single vertices in a graph of size* n.

We will use the notation $\mathcal{F}_{\text{cycle}}$, $\mathcal{F}_{\text{clique}}$, $\mathcal{F}_{\text{tree}}$, $\mathcal{F}_{\text{outerplanar}}$, $\mathcal{F}_{\text{planar}}$ and $\mathcal{F}_{\text{genus}(k)}$ as a shorthand.

Let us now introduce the degree of homogeneous components of a polynomial.

Definition 5 *Let* $\bar{x} = x_{i_1}, \ldots, x_{i_l}$ *be a subset of variables and* (f_n) *be a p-family where* $f_n \in K[x_1, \ldots, x_{r(n)}]$ *for a polynomially bounded function* $r(n)$. *We can write* f_n *as*

$$f_n = \sum_{\bar{i}} \alpha_{\bar{i}} \prod_{j=1}^{r(n)} x_j^{i_j}.$$

The homogeneous component of f_n *of degree* k *with variables* $\bar{x}$ *is*

$$\text{HOMC}_k^{\bar{x}}(f_n) = \sum_{\substack{i_1,\ldots,i_\ell \\ k=\sum_{j=1}^{l} i_j}} \alpha_{i_1,\ldots,i_\ell} x_{i_1}^{i_i} \ldots x_{i_\ell}^{i_\ell}.$$

Notice, that in the last equation the $\alpha_{i_1,\ldots,i_\ell} \in K[X]$ where

$$X = \{x_1, \ldots, x_{r(n)}\} \setminus \{x_{i_1}, \ldots, x_{i_l}\}$$

are polynomials in the variables X.

Finally, we need a last lemma in our proofs. This lemma was stated explicit by Rugy-Altherre [8] and can also be found in [2]. It will give us a way to extract all polynomials of homogeneous degree k in some set of variables in c-reductions.

Lemma 1 *Let* (k_n) *be a sequence of integers* (k_n) *and* (f_n) *be a family of polynomials over* $\mathbb{R}$. *Then there exists a c-reduction from the homogeneous component to the polynomial itself:*

$$\text{HOMC}_{k_n}^{\bar{x}}(f_n) \leq_c (f_n).$$

The circuit for the reduction has size in $\mathcal{O}(n\delta_n)$ *where* δ_n *is the degree of* f_n.

The reader should note that using this lemma will blow up our circuit polynomially in size and can hence be used only a constant number of times in succession. However, we can use this lemma on subsets of vertices. We replace every variable x_i in the subset by $x_i y$ for a new variable y and take the homogeneous components of y. We will use this technique to *enforce* edges to be taken. Notice that enforcing n edges to be taken only increases the circuit size by a factor of n. Additionally, we can set edge variables to zero to *deny* our polynomial using these edges.

We will in general ignore self-loops as if G has a self-loop it can only be homomorphic to H if H has a self-loop but then all graphs are homomorphic to H.

Next we will show one of the major hard problems we use to show completeness.

Lemma 2 *Let $\mathcal{HP}$ be the property where the single connected component is a Hamiltonian path from the vertex denoted by one to the vertex denoted by n where the Hamiltonian path does not form a cycle. Then $\mathrm{GF}(K_n, \mathcal{HP})$ is VNP hard.*

Proof: We will reduce this to Hamiltonian cycle on K_n. Let F be the polynomial $\mathrm{GF}(K_n, \mathcal{HP})$. We will compute $\sum_{1<i<j<n} \mathrm{HOMC}_1^{(1,i)}(\mathrm{HOMC}_1^{(n,j)}(F))$. Now we can replace every $x_{\{u,n\}}$ with $x_{\{u,1\}}$. This gives us now all Hamiltonian cycles in K_{n-1}. □

In general, we use the term hamiltonian paths to exclude cycles.

3 Dichotomies

We now assume all our circuits to be over the field of the real numbers $\mathbb{R}$. The reductions also work over $\mathbb{C}$.

3.1 Cycles

As a first graph class we look at cycles. The proof for the dichotomy will be relatively easy and gives us a nice example to get familiar with homomorphism polynomials and hardness proofs. In general our proofs work the following way. Given the polynomial of all graphs in some specific graph class homomorphic to H we will extract only a sum of some specific monomials with the help of homogeneous components. We can then evaluate some variables in this sum with constants to get the sum over all monomials for Hamiltonian Cycles, Matchings or Hamiltonian paths.

Our main dichotomy for cycles is the following theorem.

Theorem 5 *If H has at least one edge or has a self-loop, then $\mathcal{F}_{cycle}$ is VNP-complete under c-reductions. Else it is in VAC_0.*

The next simple fact shows us which cycles are homomorphic to a given graph H. Let $\mathrm{ev}(n)$ be defined as n if n is even and $n-1$ if n is odd.

Fact 1 *Given H a graph with at least one edge, all cycles of length $\mathrm{ev}(n)$ are homomorphic to H.*

It is easy to see that by folding the graph in half we get one path which is trivially homomorphic to an edge. Our hardness proof will only be able to handle cycles of even length. Luckily this is enough to prove hardness.

Lemma 3 *Let* $\mathcal{UHC}_{\mathrm{ev}(n),even}$ *be the graph property of all cycles of length* $\mathrm{ev}(n)$*. Then* $\mathrm{GF}(K_{\mathrm{ev}(n)},\mathcal{UHC}_{\mathrm{ev}(n),even})$ *is VNP-hard under c-reductions.*

Proof: If n is even, we can immediately use the hardness of $\mathrm{GF}(K_n,\mathcal{UHC}_n)$ (cf. Theorem 1).

If n is odd, we need to contract one edge to get the polynomial $\mathrm{GF}(K_n,\mathcal{UHC}_n)$. We do this by evaluating $\mathrm{GF}(K_{n+1},\mathcal{UHC}_{\mathrm{ev}(n),\mathrm{even}})$. Notice how K_{n+1} now has $\mathrm{ev}(n+1) = n+1$ vertices. We contract an edge with the following argument. We enforce, via taking the homogeneous component of degree one of $x_{n+1,1}$, all cycles to use $x_{n+1,1}$. Additionally to this restriction, we will sum over all our given cycles where $x_{1,i}$ and $x_{j,n+1}$ are enforced to have degree one for $i < j$ (cf. the proof of Lemma 2). We then replace $x_{i,n}$ by $x_{i,1}$ for all i and set $x_{n+1,1}$ to one. This gives us all cycles of length n.

To see this let us look at the following argument. Let the edge $(n+1,1)$ be the edge we contract and let i,j be the points picked in the sum. If we connect i,j with a path through every point we can complete this into a cycle only one way. Notice, that every different choice of i,j will construct a different cycle if we contract 1 and $n+1$.

This concludes our reduction to $\mathrm{GF}(K_n,\mathcal{UHC}_n)$. □

Later proofs will also use the contracting idea from the previous lemma. A simple case distinction will give us the proof of the theorem.

Proof: [of Theorem 5]

If H has at least one edge, we know from Fact 1 that all even cycles are homomorphic to H and by this represented in our polynomial. If we take the homogeneous components of degree $\mathrm{ev}(n)$, we extract all even cycles of length $\mathrm{ev}(n)$. This is VNP-hard via the previous Lemma (3).

If H has a self-loop, we can map all cycles to the one vertex in H. We can then extract the Hamiltonian cycles of length n by using the homogeneous degree of n as all cycles are homogeneous to a self-loop.

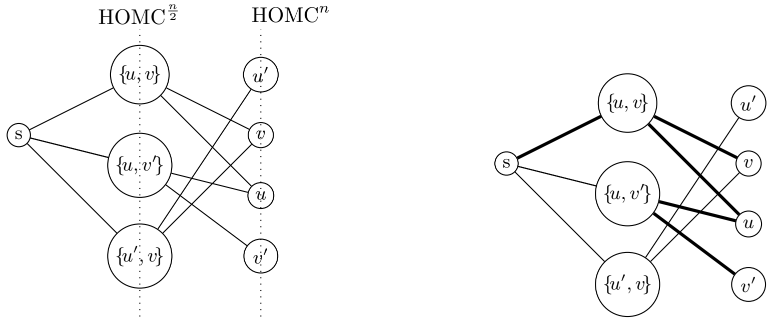
If H has no edge, our polynomial is the zero polynomial as we cannot map any graph G containing an edge to H.

Using Valiant's Criterion, we can prove membership of $\mathcal{F}_{\mathrm{cycle}}$ in VNP (cf.[2]). □

3.2 Cliques

Here, we will not use cycles in the hardness proof but work directly with the clique polynomial defined by Bürgisser [2]. The complete proof is an easy exercise. In contrast to the other results, computing $\mathcal{F}_{\mathrm{clique}}$ is easy for most choices of H.

Remark 3.1 *If* H *has a self-loop then* $\mathcal{F}_{clique}$ *is VNP-complete under c-reductions. Otherwise* $\mathcal{F}_{clique}$ *is in* VAC_0*.*



(a) Reduction from Trees to matching

(b) Invalid Construction

Figure 1

3.3 Trees

As the new characterization of VP had a specific tree structure we want to look at the generalized problem, namely for arbitrary trees with arbitrary weights. In previous sections our polynomial just contained the edges of the graph but for this section we need a slightly different model. If a monomial in our polynomial would select the edges E' we also select the vertices $\{u, v \mid \{u, v\} \in E'\}$ in our monomial. Hence, our polynomial will be over the following variable set $X = \{x_e \mid e \in E\} \cup \{x_v \mid v \in V\}$. It will be clear later why we need this special form.

Let E be a set of edges then $V(E) = \{u, v \mid \{u, v\} \in E\}$. We define the polynomial

$$\mathcal{F}^{H,n}_{\text{v-tree}} = \sum_{E' \subseteq E} \prod_{v \in V(E')} x_v \prod_{\{u,v\} \in E'} x_{\{u,v\}}$$

where the sum is over all subsets of edges which are homomorphic to a given H and where one connected component is a tree and the other connected components are single vertices. We denote this polynomial by $\mathcal{F}_{\text{v-tree}}$ for short.

Theorem 6 *If H contains an edge, then $\mathcal{F}_{v\text{-}tree}$ is VNP-complete under c-reductions. Otherwise $\mathcal{F}_{v\text{-}tree}$ is in VAC_0.*

Proof: We use a reduction from trees to perfect matchings. It is obvious that a tree is always homomorphic to one edge.

We want to compute a matching on a graph given by (V, E). We can build a graph as in Figure 1a from a K_n by setting the weight of every edge not given to zero. Hence every monomial in our homomorphism polynomial which used edges not in the graph evaluates to zero. In detail, our graph has vertices $\{v \in V\} \cup \{v_e \mid e \in E\} \cup \{s\}$. We will denote by $v_{\{u,u'\}}$ the vertex corresponding to the edge between u and u' in this section. We add[1] the

[1] Remember that we are actually removing polynomials not containing these edges but it is easy to talk about our construction if we assume we build the graph.

edges $\{v_{(u,v)}, u\}, \{v_{(u,v)}, v\}$ and $\{s, v_e\}$ for every $e \in E$. Vertices of the form $\{v_e \mid e \in E\}$ will be called *edge-vertices* in this proof. Now as the vertices used are given by $\mathcal{F}_{\text{v-tree}}$ we can take the homogeneous components over vertices. We take the homogeneous components of degree $n/2$ over vertices $\{v_e \mid e \in E\}$ and of degree n of vertices $v \in V$. Our matching in the original graph is given by the edges (s, v_e).

Every matching in the original graph has obviously a tree in our graph. Left to prove is the other direction. Given a tree in our graph, we know that only $n/2$ edge-vertices are selected. As every vertex $v \in V$ has to be connected by an edge, edge-vertices have to go to pairwise different sets of $v \in V$. Hence we can compute a perfect matching which is as hard as computing the permanent.

Valiant's Criterion will again show the membership. □

We crucially need the fact that we get the adjacent vertices for free in our homomorphism polynomials. The reader might think restricting the edges out of s might suffice but this is not the case. Let us look at Figure 1b where we removed the restriction that a specific number of vertices have to be selected. The thick path then gives us a valid tree but an invalid matching. In this case our monomial would be

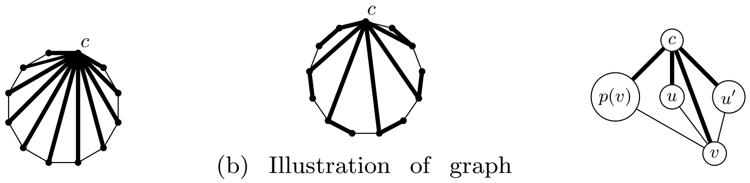
$$x_{s,\{u,v\}} x_{\{u,v\},v} x_{\{u,v\},u} x_{\{u,v'\},u} x_{\{u',v\},v'}$$

if we ignore the variables x_v for all v. We can see that replacing every edge $x_{a,b}$ with the variable $y_a y_b x_{a,b}$ for vertices a, b in our constructed graph and new variables y_a, y_b can be used for extracting homogeneous components. However, the monomial above has already degree 4 in the new y variables, namely $y_v y_u y_u y_{v'}$ is a factor of our monomial. Hence, this cannot be used to forbid the wrong instances.

It is unclear how to forbid this general behaviour without using our modified generating function. Splitting the vertices into two parts does not help but splitting a vertex into n many different vertices might. However, taking homogeneous components then would give us exponentially sized circuits. If we restrict the graph such that we would always select all edges outgoing from s we would prevent this case but the reconstruction of a matching is non trivial.

3.4 Outerplanar Graphs

Next we will show a dichotomy for outerplanar graphs. Remember that an outerplanar graph is defined by a graph that can be drawn on a plane such that every vertex is on the unbounded face of the drawing. Alternatively, it is outerplanar if it does not contain K_4 or the complete bipartite graph $K_{2,3}$ as a minor [9]. G' is a minor of G when G' can be constructed by deleting or contracting edges in G. Here contracting an edge (u, v) adds the neighbourhood of u to v and deletes v. Notice, that we also have the inverse operations, namely adding and subdividing edges; any graph G constructed this way from G' has G' as a minor.



(a) Triangle graph

(b) Illustration of graph with buddy vertices

(c)

Figure 2

We start with the case of a triangle homomorphic to H. We will use the case of triangles homomorphic to H as a simple stepping stone.

Lemma 4 *If a triangle is homomorphic to H then $\mathcal{F}_{outerplanar}$ is VNP hard under c-reductions.*

Proof: We will enforce a construction as in Figure 2a to occur exclusively in a polynomial p we compute with the help of homogeneous components from $\mathcal{F}_{\text{outerplanar}}$. For this, we pick an arbitrary vertex c and enforce all n outgoing edges from this vertex via homogeneous components. We further enforce the whole graph to have $n + n - 3$ edges. We call the resulting polynomial p. The graph given in Figure 2a is outerplanar as the outerplanar embedding (where all vertices belong to the unbounded face) is given in the figure. Let us call the set of graphs represented by the polynomial p now S. Remember that our polynomial, given from extracting homogeneous components is a summation over monomials where the monomials represent sets of edges and hence graphs. We still need to proof that all graphs in S are isomorphic to Figure 2a.

We call the implied order of the graph, the order of the outer circle of vertices starting from the star and ending at it again without any edges crossing. As there are two such orderings let us fix an arbitrary one for every graph. We want to look at graphs in S that do not have the implied order of the outer vertices. These graphs are outerplanar by definition of $\mathcal{F}_{\text{outerplanar}}$ and not isomorphic to the graph in Figure 2a. We will prove that this is impossible by contradiction.

Having not the implied order on the outer vertices gives us two cases. Either there exists a vertex $v \neq c$ which has degree greater or equal to 4 or there exists at least a vertex $v \neq c$ of degree less than 3 and all other vertices $u \neq c$ have degree at most 3.

Let us look at the first case. Let v be the first such vertex and let $p(v)$ be the predecessor of v in the partial order. Notice that by enforcing all n instead of just $n - 2$ edges starting at the center, a predecessor $p(v) \neq c$ has to exist. Let u, u' denote the other vertices adjacent to v different than $p(v)$ and c. As we enforced edges from c to every vertex, we can easily see the $K_{2,3}$ with v, c on the one side and $u, u', p(v)$ on the other side when we delete the edge $\{v, c\}$ (see Figure 2c). Hence the graph cannot be outerplanar and hence we have the contradiction. This implies that every vertex except c and the two neighbouring vertices have degree at most 3. Enforcing the overall number of edges gives us that every vertex has at least degree 3 and hence implies equality. Hence every graph in S is isomorphic to the graph in Figure 2a as every vertex has a unique predecessor and successor where the first predecessor is the vertex left of c and the last successor the vertex right of c.

Now our homomorphism polynomial restricted to these edges sums over all monomials corresponding to the graph in Figure 2a. The outer path gives us almost all Hamiltonian paths. In fact, it gives us all permutation of $n-2$ vertices. We need to remove the center of the star by evaluating the edges with one as well as evaluate the other enforced edges with one. This polynomial is now VNP hard by Lemma 2. Taking the homogeneous components as described only increases the circuit by a factor of n. □

Theorem 7 *If H has an edge then $\mathcal{F}_{outerplanar}$ is VNP-complete under c-reductions and otherwise trivial.*

We want to use a similar construction as in Lemma 4 but have the graph constructed be homomorphic to a single edge so that the constructed graph can be homomorphic to any H with a single edge.

Proof: For every vertex v, except c, we choose a *buddy vertex v'*. We enforce the edge between every vertex and his buddy vertex and set the edge between a buddy vertex and c to zero. Additionally, we set all edges from v to any other non buddy vertex to zero and all edges from a buddy vertex to a different buddy vertex to be zero. In essence this splits every vertex into a left and right part (see Figure 2b). Similar to the proof of Lemma 4 we enforce edges from c to v for every non buddy vertex and to take exactly $n + 2n - 3$ edges. Let us call the set of all graphs extracted with this construction from $\mathcal{F}_{\text{outerplanar}}$ S' and the set of all graphs given from the restriction in Lemma 4 of $\mathcal{F}_{\text{outerplanar}}$ where we assume a triangle is homomorphic to H, S

Let us now prove that there exists a bijection between S and S'. Take an outerplanar graph G' from S'. We can then contract the edges between every vertex and its buddy vertex. This gives us a graph G in S. If the combined degree of a vertex and its buddy vertex (disregarding the connecting edge between v and v') would be greater than 3 G' would be not outerplanar. The reason for this is again the contraction. As we can contract it to a graph with a vertex of degree 4 of the form in S the proof of Lemma 4 would tell us that this graph would be not outerplanar. As G is a minor of the graph in S', G' would not be outerplanar. In essence, our graph with combined degree greater than 3 can be constructed from a G by adding and subdividing edges which can be constructed from $K_{2,3}$ if G was not outerplanar.

For the other direction a similar proof holds. If we have given a graph as in Figure 2a we can subdivide the edges as stated earlier and have a graph of the form as in Figure 2b.

Now that we know that the sets of graphs S and S' are in a bijective relation, we need to transfer the hardness. This is easy as we can do the following variable replacement. Let us look at the variable $x_{u',v}$ corresponding to the edge (u', v), where u' is the buddy vertex of u and v is a non buddy vertex. We can evaluate

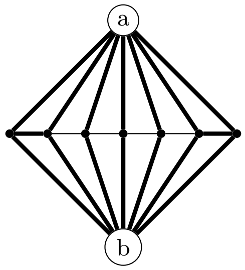


Figure 3: Planar Gadget

this with $x_{u,v}$. Similarly, for the edge (v', u) we can evaluate $x_{v',u}$ with $x_{v,u}$. Finally, we evaluate $x_{v,v'}$ with 1. With this and the bijection between S and S' we constructed the polynomial p as in Lemma 4 even if H is homomorphic to an edge.

Taking the homogeneous components increases the circuit size by a factor of n.

We know by [21] that checking if a graph is outerplanar is possible in linear time. With this we can use Valiant's Criterion to show the membership. □

3.5 Planar Graphs

Lemma 5 *All planar graphs with $n + 2 + 2(n + 2)$ edges and the edges required as in Figure 3 are in a bijective relation to all Hamiltonian paths on a set of vertices we can denote by $v'_1, \dots, v'_n$.*

Proof: Let us again call the set of graphs represented by the polynomial $\mathcal{F}_{\text{planar}}$ where we restrict all graphs to have $n + 2 + 2(n + 2)$ edges overall and the edges as in Figure 3 S. Let us call the set of all graphs that are isomorphic to Figure 3 S'. We first want to show that these sets are the same.

Let $G \in S$ with a given embedding in the plane similar to Figure 3. Left to show is that we will always have an ordering of the vertices in the middle. Then the graph in S' would be in S' and $S = S'$ as all graphs from S' fulfill the criteria for membership in S.

Let now such an ordering not be given, meaning there exists a unique left most vertex v with two right successors u, u' and a predecessor $p(v)$. By construction the predecessor always exists. We denote the top and bottom vertex by a and b in our graph. We can now build a $K_{3,3}$ minor in the following way. $S_1 = \{v, a, b\}$ and $S_2 = \{u, u', p'\}$. As a and b are connected to every vertex we only need to check that u is connected to u, u' and p which is by assumption. This proves that via edge deletion our graph would have a $K_{3,3}$ minor if the vertices would not give us a permutation. Hence $S \subseteq S'$.

It is now easy to see that every vertex has a unique predecessor and successor and hence the graph gives a permutation of the vertices. By slightly adjusting this to exclude the first and last two vertices and the vertices denoted by a and b we get a path for these vertices. We call these vertices $v'_1, \dots, v'_n$. □

Theorem 8 *If H has an edge then $\mathcal{F}_{planar}$ is VNP-complete under c-reductions. Otherwise $\mathcal{F}_{planar}$ is in VAC_0.*

Proof: We again get all paths by enforcing our polynomial to exclusively have the graph from Figure 3 by taking appropriate homogeneous components and setting variables to zero. By Lemma 5 this gives us all path which is VNP-hard by Lemma 2.

However, this graph is not yet homomorphic to a single edge. To accomplish this, we will use a graph of size $2n$. We, as in the outerplanar case, enforce every vertex, except a and b, to have a buddy vertex u_v. Then we subdivide the edge (a, v) and (b, v) for every original, meaning none buddy, vertex v with a new vertex v'_a, v'_b respectively. This will give us for every part a square consisting of the vertices a, v, v'_a, u_v and the square b, v, v'_b, u_v. We again set the edges from a buddy vertex to another buddy vertex to zero.

Now it is easy to see that we can fold a to b which leaves us with a grid of height one. A grid can be easily folded to one edge. The size of the circuit is increased by a factor of at most $2n$.

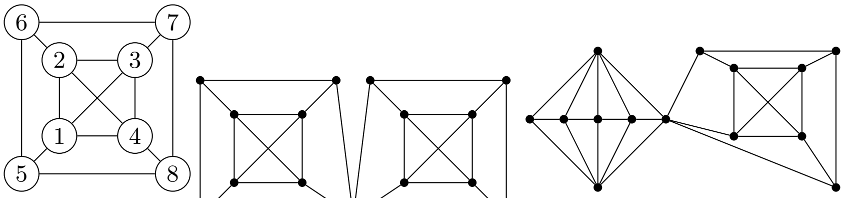
Left to show is the correctness. As in the outerplanar case, we will first show that the sets of graphs for the graphs described here and in Lemma 5 are in a bijective relation to each other. Given a graph from the construction in this proof, we can contract edges between v and the buddy vertex v' for every v. Additionally, we contract the edges (v'_a, a) and (v'_b, b). This gives us a graph as in the construction in Lemma 5. From a graph in Lemma 5, we can give the construction as described in the first part of this proof. Hence the two sets of graphs have a bijection between them.

Finally, we evaluate the edges (v'_a, a), (v'_b, b) and (v, v') with one and as in the outerplanar case the edges (u', v) with $x_{u,v}$ and (v', u) with $x_{v,u}$. This gives us then the correct polynomial.

As testing planarity is easy, we can use Valiant's Criterion to show membership. □

3.6 Genus k graphs

Graph embeddings are one of the major relaxations of planarity. For this we find a surface of a specific type such that a graph can be embedded in this surface without any crossing edges. If we want to increase the *orientable genus* of a surface by one, we can glue a handle onto it which edges can use without crossing other edges. To increase the *non-orientable genus* we can glue a crosscap to it. We call a graph a *orientable genus k graph* if there exists a surface of orientable genus k such that G can be embedded in this surface and k is minimal. Similarly, if there exists a surface of non-orientable genus where we can embedd a graph, we call this a *non-orientable genus k graph.* Notice, that a genus 0 graph is planar, no matter if we look at orientable or unorientable genus. In the following, if we talk about genus, the statement will hold for orientable and non-orientable genus. While the topic of graph genus is vast, we will mostly use theorems as a



(a) orientable Gadget (b) Two orientable Gadgets (c) Orientable Gadget with planar gadget

Figure 4

blackbox and only reason about graphs of genus zero and one. For a detailed coverage of the topic, the reader is referred to [9].

With the planar result in place we can use the simple proof strategy. Enforce a genus k graph where we append the planar construction. If we now enforce all our graphs in the homomorphism polynomial to be isomorphic to the genus k graph which is appended to the planar gadget our genus bound will ensure that our planar gadget gives us all Hamiltonian paths on the set of vertices. Of course this holds only if the combined graph of the genus k graph and the planar construction does not reduce the genus.

Lemma 6 *The graph in Figure 4a has orientable genus one.*

Proof: We can use the given embedding with one handle for the crossing in the middle to show an upper bound of one.

We again construct a $K_{3,3}$ with the sets $S_1 = \{2, 1, 6'\}, S_2 = \{3, 4, 7'\}$ where $6'$ is the vertex constructed from contracting the edge $(5, 6)$ and $7'$ from the edge $(7, 8)$. And hence the graph is not planar and has a lower bound for the orientable genus of one. □

The next theorem shows how we can glue graphs together to increase the orientable genus in a predictable way.

Definition 6 ([20]) *G is a* vertex amalgam *of H_1, H_2 if G is obtained from disjoint graphs H_1 and H_2 where we identify one vertex form H_1 with one vertex from H_2.*

With this we restate a theorem from Miller [20] to compute the orientable genus of a given graph.

Theorem 9 ([20]) *Let $\gamma(G)$ be the orientable genus of a graph G. Let G be constructed from vertex amalgams of graphs $G_1, \ldots, G_n$. Then $\gamma(G) = \sum_{i=1}^{n} \gamma(G_i)$.*

This now gives us immediately the result that a graph constructed as in Figure 4b with k gadgets has genus k.

Theorem 10 *If H has an edge then $\mathcal{F}_{genus(k)}$ is VNP-complete under c-reductions for any k. Otherwise $\mathcal{F}_{genus(k)}$ is in VAC_0.*

Proof: With Theorem 9 and the construction in Figure 4 we are almost done. We enforce a genus k graph to occur in our polynomial by enforcing k connected blocks as in Figure 4 where one block is connected to a planar gadget. Additionally, we enforce for our planar gadget the same edges as in Theorem 8. Hence all graphs that are homomorphic to the planar gadget have to have genus zero and hence are planar. We can then use the proof as in Lemma 5 to extract all Hamiltonian paths.

The only thing left to do is to modify our graphs such that they are homomorphic to an edge without violating the properties. It is clear that we can fold our genus one gadgets together. If we then subdivide the edge $(1,3)$ and $(2,4)$ (which keeps our block property) we can first fold 7 to 5 and 3 to 1. Folding then again 6 to 8 and 2 to 4 we get a square with two dangling edges. The dangling edges can be folded onto the square and the square is homomorphic to one edge. This construction increases the size of the circuit at most by a factor of $14k + 2n$. As testing for a fixed genus is in NP, we can use Valiant's Criterion to show membership. □

We can now look at the Euler genus. The Euler genus of a graph is defined as the minimal number n such that G can be drawn on a surface with n crosscaps or $n/2$ handles. A similar amalgam theorem exists for this type of genus. We again have an additivity theorem.

Theorem 11 ([20]) *Let $\gamma'(G)$ be the Euler genus of a graph G. Let G be constructed from vertex amalgams of graphs $G_1, \dots, G_n$. Then $\gamma'(G) = \sum_{i=1}^{n} \gamma'(G_i)$.*

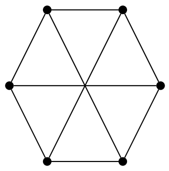


Figure 5: Non-orientable genus 1 graph

Lemma 7 *The graph in Figure 5 has non-orientable genus one.*

Proof: It is easy to see that this graph is not planar and hence has non-orientable genus greater than zero. The graph is embeddable into a surface of non-orientable genus 1 if we set the crosscap into the middle. □

Now we can combine this gadget with the edge amalgams to give the following theorem.

Theorem 12 *Let $\mathcal{F}^{H,n}_{\textit{Euler-genus(k)},n}$ where Euler-genus(k),n is the property where one connected component has Euler genus k and the others are single vertices in a graph of size n. If H has an edge then $\mathcal{F}^{H,n}_{\textit{Euler-genus(k)},n}$ is VNP-complete under c-reductions for any k. Otherwise $\mathcal{F}^{H,n}_{\textit{Euler-genus(k)},n}$ is in VAC_0.*

Similarly, we can look at the non-orientable genus. However, there exists no additivity theorem for the non-orientable genus like Theorem 9. However, Mohar [22] showed that we can construct a graph with a fixed non-orientable genus.

Theorem 13 *Let $K^{(k)}_{n,m}$ be the graph obtained from $K_{n,m}$ by deleting an arbitrary set of k edges. Then the non-orientable genus of $K^{(k)}_{n,m}$ is given by*

$$\hat{\gamma}(K^{(k)}_{n,m}) = \max\{\lceil \frac{(m-2)(n-2)-k}{2} \rceil, 1\}$$

unless $n = k, m = k+1$ or $n = k+1, m = k$ and k even or $n = m = k$ for every k.

With this we can use a similar argument as in Theorem 10. We restrict our graphs from the polynomial to have a non-orientable genus k graph as in Theorem 13 and restrict a planar gadget glued to it.

Theorem 14 *Let $\mathcal{F}^{H,n}_{\hat{\gamma}(k),n}$ where $\hat{\gamma}(k), n$ is the property where one connected component has non-orientable genus k and the others are single vertices in a graph of size n. If H has an edge then $\mathcal{F}^{H,n}_{\hat{\gamma}(k),n}$ is VNP-complete under c-reductions for any k. Otherwise $\mathcal{F}^{H,n}_{\hat{\gamma}(k),n}$ is in VAC_0.*

4 Conclusion

We have shown many dichotomy results for different graph classes but some classes are still open. We want to especially mention the case of our graph class being the class of trees. It is known that we can use Kirchoff's Theorem to find all spanning trees of a given graph. This, however, does not include monomials of total degree less than $n-1$ which our polynomials include. It is unclear how to decrease the size of the trees without disconnecting them into forests. From the algebraic view, the knowledge ends here. In the counting view, where we solve the task of counting all trees in a graph, a bit more is known. Goldberg and Jerrum [14] showed that counting the number of subtrees that are distinct up to isomorphism is #P-complete. This, combined with our dichotomy for trees including the vertices, gives us a strong indication that the similar problem is VNP-hard in the algebraic world.

A different expansion of these results would be the case of bounded treewidth. As mentioned earlier, in the counting version the case of bounded treewidth is indeed the most general form and completely characterizes the easy and hard instances of counting graph homomorphisms. Additionally, recent advancements showed that graph homomorphisms of a specific type characterize VP. Can homomorphism from graph classes parameterized by treewidth, similar to the counting case, be used for a complete characterization of VP and VNP? This, however, seems unlikely as we believe that the case for trees is already hard even if we do not allow vertex variables as in this paper. A result on the hardness for the graph class being trees would clarify this. As a side note, we should mention the related result by Courcelle et al. [5]. They showed that for any graph property $\mathcal{E}$ which is definable in monadic second-order logic $\mathrm{GF}(G_n, \mathcal{E})$ is in VP if G_n is a family of graphs of bounded treewidth. However, in our case, we assume G_n to be K_n and hence not have bounded treewidth. It seems unlikely that we can transfer this algorithm as we cannot decompose K_n into a nice tree decompositon and work on the bags given by it.

An interesting research direction would be the case of disconnected graph properties. Rugy-Altherre looked at the property that any graph is homomorphic to a given graph H. This includes disconnected graphs with connected components larger than one vertex. We instead only looked at restricted homomorphisms where one major connected component exists. It is unclear to the author if our proofs could be adapted to this case.

Acknowledgments I want to thank my doctoral advisor M. Bläser for his guidance. I additionally want to thank R. Curticapean for many discussions on the counting versions on problems and B. V. Raghavendra Rao for introducing me to this topic. Additionally, I like to thank S. Tavenas for his helpful comments.

References

[1] A. A. Bulatov and M. Grohe. The complexity of partition functions. *Theor. Comput. Sci.*, 348(2-3):148–186, 2005. doi:10.1016/j.tcs.2005.09.011.

[2] P. Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7. Springer, 2000.

[3] J. Cai, X. Chen, and P. Lu. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM J. Comput.*, 42(3):924–1029, 2013. doi: 10.1137/110840194.

[4] C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theor. Comput. Sci.*, 239(2):211–229, 2000. doi:10.1016/S0304-3975(99) 00220-0.

[5] B. Courcelle, J. A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001. doi: 10.1016/S0166-218X(00)00221-3.

[6] V. Dalmau and P. Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

[7] V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In P. V. Hentenryck, editor, *8th International Conference on Principles and Practice of Constraint Programming - CP 2002*, volume 2470 of *Lecture Notes in Computer Science*, pages 310–326. Springer, 2002.

[8] N. de Rugy-Altherre. A dichotomy theorem for homomorphism polynomials. In B. Rovan, V. Sassone, and P. Widmayer, editors, *37th International Symposium on Mathematical Foundations of Computer Science 2012 - MFCS 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2012. doi:10.1007/978-3-642-32589-2_29.

[9] R. Diestel. *Graph Theory.* Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2000.

[10] A. Durand, M. Mahajan, G. Malod, N. de Rugy-Altherre, and N. Saurabh. Homomorphism polynomials complete for VP. In V. Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014*, volume 29 of *LIPIcs*, pages 493–504. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPIcs.FSTTCS.2014.493.

[11] M. E. Dyer and C. S. Greenhill. The complexity of counting graph homomorphisms. *Random Struct. Algorithms*, 17(3-4):260–289, 2000. doi: 10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W.

[12] C. Engels. Dichotomy theorems for homomorphism polynomials of graph classes. In M. S. Rahman and E. Tomita, editors, *9th International Workshop on Algorithms and Computation - WALCOM 2015*, volume 8973 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2015. `doi:10.1007/978-3-319-15612-5_25`.

[13] E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In H. E. Shrobe, T. G. Dietterich, and W. R. Swartout, editors, *8th National Conference on Artificial Intelligence.*, pages 4–9. AAAI Press / The MIT Press, 1990. URL: `http://www.aaai.org/Library/AAAI/1990/aaai90-001.php`.

[14] L. A. Goldberg, M. Grohe, M. Jerrum, and M. Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM J. Comput.*, 39(7):3336–3402, 2010. `doi:10.1137/090757496`.

[15] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1), 2007. `doi:10.1145/1206035.1206036`.

[16] M. Grohe and M. Thurley. Counting homomorphisms and partition functions. *CoRR*, abs/1104.0185, 2011. URL: `http://arxiv.org/abs/1104.0185`.

[17] P. Hell and J. Nesetril. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. `doi:10.1016/0095-8956(90)90132-J`.

[18] P. Hell and J. Nešetřil. *Graphs and homomorphisms*, volume 28. Oxford University Press Oxford, 2004.

[19] M. Mahajan and B. V. R. Rao. Small space analogues of valiant's classes and the limitations of skew formulas. *Computational Complexity*, 22(1):1–38, 2013. `doi:10.1007/s00037-011-0024-2`.

[20] G. L. Miller. An additivity theorem for the genus of a graph. *Journal of Combinatorial Theory, Series B*, 43(1):25 – 47, 1987. `doi:10.1016/0095-8956(87)90028-1`.

[21] S. L. Mitchell. Linear algorithms to recognize outerplanar and maximal outerplanar graphs. *Inf. Process. Lett.*, 9(5):229–232, 1979. `doi:10.1016/0020-0190(79)90075-9`.

[22] B. Mohar. Nonorientable genus of nearly complete bipartite graphs. *Discrete & Computational Geometry*, 3:137–146, 1988. `doi:10.1007/BF02187903`.

[23] L. G. Valiant. Completeness classes in algebra. In M. J. Fischer, R. A. DeMillo, N. A. Lynch, W. A. Burkhard, and A. V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing - STOC 1979*, pages 249–261. ACM, 1979. `doi:10.1145/800135.804419`.