An Efficient Algorithm for the Genus Problem with Explicit Construction of Forbidden Subgraphs

Hristo Djidjev+
Institute of Mathematics
Bulgarian Academy of Sciences
Sofia, Bulgaria
and
School of Computer Science
Carleton University
Ottawa, Canada K1S 5B6

and

John Reif*
Computer Science Dept
Duke University
Durham, NC 27706

0. Abstract

We give an algorithm for imbedding a graph G of n vertices onto an oriented surface of minimal genus g. If $g > 0$ then we also construct a forbidden subgraph of G which is homeomorphic to a graph of size $\exp(O(g)!)$ which cannot be imbedded on a surface of genus g-1. Our algorithm takes sequential time $\exp(O(g)!)n^{O(1)}$. Since $\exp(O(g)!) = \exp(\exp(O(g\log(g))))$, our algorithm is polynomial time for genus $g=O(\log\log(n)/\log\log\log(n))$. A simple parallel implementation of our algorithm takes parallel time $(\log n)^{O(1)}+O(g)!$ using $\exp(O(g)!)n^{O(1)}$ processors. We give also the smallest known upper bound, namely $\exp(O(g)!)$, on the number $F(g)$ of homeomorphic distinct forbidden subgraphs for graph imbeddings onto a surface of genus g.

The two best previous algorithms [Filotti, Miller, Reif,79] and [Robertson and Seymour,86] for graph imbedding onto a surface of genus g, required $n^{O(g)}$ and $f(g)n^2$ sequential time, respectively. The work of [Robertson and Seymour,86] also gave a finite bound for $F(g)$. However their proof spanned many papers and were highly nonconstructive; $f(g)$ and $F(g)$ were bounded by some (large) tower of exponents of g.

Our work provides a distinct constructive approach giving considerably improved bounds for $f(g)$ and $F(g)$ and vastly simplified proofs. In particular, we use a "bootstrap" technique that uses a discovered forbidden subgraph for given genus $g'<g$ to to aid us in determination of genus $g'+1$ imbeddings. It seems likely that our techniques can be extended to many other problems on graphs with bounded tree width.

+ Supported by Bulgarian Academy of Sciences grant for travel and visit to CS Dept, Duke University in Dec, 1988.

* Supported by Air Force Contract AFOSR-87-0386, DARPA/ISTO Contract N00014-88-K-0458, NASA/CESDIS Contract 550-63 NAS 5-30428 URSA.

1. Introduction

1.1. Topological Imbeddings

See Appendix A

1.2. Combinatorial Imbeddings

See Appendix A

1.3. The Complexity of Some Previous Algorithms for Graph Genus

The *genus* of graph G is the minimal $g \geq 0$ s.t. G can be imbedded onto a surface of genus g. Using purely combinatorial techniques, [Miller,85] has shown that the genus of a graph is the sum of the genus numbers of its biconnected components. He also showed that minimal genus imbeddings of any biconnected subgraphs can be easily combined in time $O(|G|)$, where $|G|$ denotes the number of vertices and edges of G, to get a minimal genus imbedding of G. Hereafter, we assume without loss of generality that the graph is biconnected.

The *genus problem* is: given a graph G determine the genus g of G. The genus problem is very difficult for g growing as a function of $|G|$. An enumerative algorithm of [Edmonds,60] gave a $|G|^{O(|G|)}$ algorithm for the genus of G. [Reif,78] first showed that the problem of extending a given graph imbedding is NP complete, and recently [Thomassen,89] showed that given a graph imbedding of genus g, the problem of testing if there is an imbedding of genus $< g$ is NP complete. This implies the problem of testing if a graph has genus g is NP complete, and therefore there does not exist a polynomial algorithm for finding the genus of the graph unless P=NP.

Nevertheless, the genus problem for imbedding graphs of unbounded size onto fixed surfaces of low genus g may be efficiently solved. Let a *PT algorithm* be a planarity testing algorithm taking sequential time $O(|G|)$, e.g. that of [Hopcroft and Tarjan,74]. [Klein and Reif,1987] gave the first efficient parallel planarity algorithm with $O(\log n)^2$ time and n processors, where n is the number of vertices of G. Recently, [Ramachandran and Reif,89] gave a $O(\log n)$ time parallel algorithm for graph planarity with work nearly $O(n)$. As another example, [Filloti,80] gave an $n^{O(1)}$ time algorithm for testing if a graph G has genus 1, i.e. can be imbedded onto a torus.

Graphs of bounded genus appear naturally in various applications - for example, in VLSI layout via bounded book thickness imbeddings. Many difficult graph problems can be solved in polynomial time in the case of graphs of bounded genus; for example [Miller,83] showed that for bounded genus graphs, the isomorphism problem can be solved in polynomial time. [Djidjev,85] gave a linear time algorithm for finding small separators of graphs of bounded genus.

[Fillotti, Miller, Reif,79] showed that given a graph G, its genus g and imbedding of G of genus g can be computed in time $|G|^{O(g)}$. This gave the first polynomial time bound for the genus problem with fixed genus g.

1.4. Forbidden Subgraphs

A key aspect of our algorithm, used to aid us in the construction of higher genus imbeddings, is the discovery of certain forbidden subgraphs of imbeddings of lower genus, as defined here.

A path in graph G will be called a *2-path*, if each of its (non-endpoint) vertices is incident with no more than two edges of G. A 2-path p of G will be called a *maximal 2-path*, if no other 2-path of G contains p. We will define the *branchsize* $[G]$ of G to be the number of maximal 2-paths of G. If $p=(v_1,...,v_k)$ is a maximal 2-path of F, then $(v_k,...,v_1)$ is also a maximal 2-path of F and will be denoted by p^R.

The *homeomorphic contraction* of G is gotten by substituting an edge for each maximal 2-path in G. **(See Figure 1.4)**. Note that the branchsize $[G]$ is the number of vertices of the homeomorphic contraction of G. Two graphs are *homeomorphic* if their homeomorphic contractions are isomorphic. **(See Figure 1.5)**.

We define graph FS to be a *forbidden$_g$ subgraph* of graph G if FS is a minimal subgraph with genus $> g$ (i.e., deletion of an edge or vertex of FS results in a graph of genus at most g). [Kuratowski,30] showed that the forbidden$_0$ subgraphs are all homeomorphic to K_5 or $K_{3,3}$. A number of researchers have independently observed that *PT* algorithms can be extended to find in time $O(|G|)$ a maximal (in specific context) planar subgraph of a nonplanar graph G, and also in time $O(|G|)$, given the maximal planar subgraph, find a forbidden$_0$ subgraph of G homeomorphic to K_5 or $K_{3,3}$. We will call such an extension a *PT-FS algorithm*. Recently [Khuller,Mitchell,Vazirani,89] gave an $O(\log^2 n)$ time and $O(n)$ processor parallel *PT-FS* algorithm, using the techniques of [Klein and Reif,88].

1.5. The Work of Robertson and Seymour

In a celebrated series of papers on graph minors, [Robertson and Seymour,I-VIII] proved that for each genus $g \geq 0$, there is a finite number $F(g)$ of homeomorphic distinct forbidden$_g$ subgraphs and furthermore, they showed that given a graph G of genus $> g$, in time $f(g)n^2$ there can be found a forbidden$_g$ subgraph of G. The upper bounds $f(g)$ and $F(g)$ were in their original work nonelementary functions of g (in fact $f(g)$ and $F(g)$ were originally not explicitly known and instead were computed by a procedure involving a sequence of towers of towers of iterated exponents). These results gave the polynomial time bound with fixed degree independent of g. More recent unpublished work of Robertson and Seymour has brought bounds on $f(g)$ and $F(g)$ to a bounded but large number of repeated exponents.

These results in graph theory are a major breakthrough - in our opinion the greatest breakthrough in graph algorithms in at least the last decade. However, the extreme dependence on the parameter g made their results difficult to apply in practice even for very small g (say $g > 3$). Also, because of the extreme complexity of their proof (encompassing many papers), their work have not been properly understood by the theoretical computer science community.

1.6. Our Contribution

The key contribution of our paper is to derive considerably improved bounds on *f(g)* and *F(g)* of $\exp(O(g)!) \leq \exp(\exp(O(g\log(g))))$. Our algorithm is polynomial time for genus $g < O(\log\log(n)/\log\log\log\log(n))$. Our results are proved essentially independently of the work and extensive papers of Robertson and Seymour. We use none of their results but do use minors extensively; we use a "bootstrap" technique that uses a discovered forbidden subgraph for given genus $g'<g$ to aid us in determination of genus $g'+1$ imbeddings. (We also use constraint graphs related to those of the parallel planarity paper of [Ramachandran and Reif,89]). We feel that aside from our improved results and more constructive approach, the main impact of our work is that we provide an independent verification and vastly simplified proofs of the basic results of Robertson and Seymour in a particular fundamental area of application, namely graphs of bounded genus. It seems likely that our techniques can be extended to many other problems on graphs with bounded tree width.

2. General Description of Our Imbedding Algorithm

2.1. Definition of Bridges

We will require a few more graph definitions concerning subgraphs. Given a graph *G* and a subgraph *H*, let *G-H* consist of the subgraph gotten by deleting from *G* all edges of *H* and deleting every vertex of *H* incident only to edges of *H*. Note that *G-H* may have vertices in common with *H*; these are called the *attachment vertices* of *G-H*. A *bridge B* of *H* in *G* is a subgraph of *G-H* induced from a maximal set of edges for which between any pair of edges there is a path in *G-H* avoiding any attachment vertices **(See Figure 2.1)**. *B* is a *bridge* of *F*, where *F* is a face of *I(H)*, if all attachment vertices of *B* are on *F*. The edges of bridges adjacent to attachment vertices are called *attachment edges*. If G_1 and G_2 are graphs, we define the graph $G_1+G_2 = (V(G_1)\square V(G_2)\ , E(G_1)\square E(G_2))$.

Fix a graph *G* and a subgraph *H* with a given imbedding *I(H)* of genus *g*. Define a *forbidden*$_g$ *subgraph* for *I(H)* in *G* to be a subgraph FS^* of *G-H* such that any extension of *I(H)* to $H+FS^*$ is of genus greater than *g*. Let a *forbidden*$_g$ *subgraph FS* for *H* in *G* be a minimal subgraph of *G* such that genus(*FS*+*H*) > genus(*H*) (see also Section 1.4).

2.2. Skeletal Imbeddings

Given an imbedding *I(G)* of genus *g* of a biconnected graph *G,* a *skeletal subimbedding I(H)* of *G* is an imbedding *I(H)* of genus *g* (consistent with imbedding *I(G)*) of a minimal biconnected subgraph *H* of *G*, such that if we delete any edge of *I(H)* then the resulting imbedded graph is either now a genus < *g* imbedding or no longer biconnected.

Let *T* be any spanning tree of *G*. A skeletal subimbedding *I(H)* of *I(G)* can be found by a 2 step process:

(1)repeatedly delete from G nontree edges (not necessarily preserving biconnectivity) until there is only one face. By Euler's formula there will be only $2g$-1 remaining nontree edges. The $2g$-1 basis cycles defined by these remaining nontree edges define a subimbedding of genus g with 1 face.

(2)Next we need add to this subimbedding at most $2g$-1 further basis cycles until the resulting imbedded subgraph *I(H)* is biconnected.

Thus finding a skeletal subimbedding *I(H)* of *I(G)* requires nondeterministic choice of at most $O(g)$ nontree edges, and consists of at most $O(g)$ basis cycles. The key point is that *I(H)* has at most $g^{O(g)}$ distinct imbeddings of genus g, whereas G may have $[G]!$ distinct imbeddings of genus g.

Given an (unimbedded) biconnected graph G and a spanning tree T of G, let $SI_g(G,T)$ be the set of all possible (with respect to T) skeletal subimbeddings *I(H)* for all possible imbeddings of *I(G)* of genus g. We can construct $SI_g(G,T)$ in deterministic time $([G]g)^{O(g)}$ (rather than inefficiently enumerating through all imbeddings of G of genus g) by simply enumerating all possible choices of the $O(g)$ basis cycles used to construct the skeletal subimbeddings and further enumerating through all possible imbeddings of these $O(g)$ basis cycles, and then verifying whether each resulting subimbedding is skeletal.

2.3 Outline of our Imbedding Algorithm

Note: "*guess*" and "*choose*" are to be executed by sequentially iterating over all possibilities.

Algorithm 2.1

Input biconnected graph G

Output the genus of G and an imbedding of G on a surface of minimum genus

```
1.      Call procedure PT on $G$. If $G$ is planar then  output "genus($G$) = 0" and the planar imbedding
        halt;
        else let $F_0$ be the  forbidden$_0$ subgraph of $G$ and let $g := 1$
2.      While $g \leq (n^2-n)/2$ do
Comment: $F_g$ denotes the current forbidden$_{g-1}$ subgraph of $G$.
Comment: $U_g$ denotes a set of subgraphs of $G$ used to augment $F_g$ to $F_{g+1}$.
Comment: Assume  as loop invariants: genus($F_g$) = $g$; $F_g$ is biconnected; $[F_g] \leq \exp(O(g)!)$.
2.1.    $U_g$ := empty set {};
2.2     Construct some spanning tree $T$  of $G$
2.3     Construct the set of skeletal subimbeddings $SI_g\ (F_g,T)$ by Algorithm of Section 2.2
2.4     For  each skeletal subimbedding $I(H_g)$ of $SI_g\ (F_g,T)$  do
Comment: there are at most   $[F_g]^{O(g)}$ skeletal subimbeddings of $F_g$
2.4.1   Construct all bridges $B_1,...,B_b$ of $G$-$H_g$.
2.4.2   If  for some $i$, $1 \leq i \leq b$, there is no planar imbedding of $B_i$ that can be imbedded onto $I(H_g)$
        then  let FS := subgraph of $H_g+B_i$ of branchsize $O(1)$ that cannot be added to $I(H_g)$;  goto 2.4.5
        ;  else do
```

```
2.4.2.  Find by the algorithms from Sections 3 and 4 an imbedding $I(H''_g)$ of a subgraph $H''_g$ of $G$,
        $H_g \le H''_g \le G$, such that $H''_g - H_g$ consists of $O(g)$ 2-paths and a minimal set $R$ of bridges of
        $G - H''_g$ that cannot be imbedded onto $I(H''_g)$ without increasing the genus. (We will let "$\le$"
        denote the subgraph relation.)
2.4.3.  If $R = \{\}$ then do
            Imbed the bridges of $G - H''_g$ onto $I(H_g)$ using the algorithm given in Appendix C for the 2-
            satisfiability problem
            output "genus($G$)=$g$" and the resulting genus $g$ imbedding $I(G)$ of $G$; halt fi
2.4.4.      Using the Algorithm of Appendix E, find a set of $O(1)$ paths belonging to bridges in $R$ that
            cannot be all imbedded onto $I(H''_g)$. Let $FS$ be the subgraph of $G$ consisting of those paths plus
            all 2-paths of $H''_g - H_g$
        od fi
2.4.5.  $U_g := U_g \cup \{FS\}$ od
2.5.    Using Algorithm of Appendix F, construct from $U_g$ and $F_g$ a biconnected subgraph $F_{g'}$ of $G$
        of branchsize $[F_g]^{O(g)}$ such that genus($F_{g'}$) = $g' > g$
2.6     $g := g'$
        od
```

2.4. Organization of the Paper

We have just given an outline of our imbedding algorithm. The implementation of Step 2.4.2 is decribed in Sections 3 and 4. Section 3 describes a reduction of the original problem to the weakly quasiplanar extension and 2-constraint imbedding problems. The *2-satisfiability problem* is to determine satisfiability of a boolean CNF formula with at most 2 literals per clause. Section 4 gives a reduction of the problem of finding a conflicting set of bridges to that of 2-satisfiability. Section 5 concludes the paper with bounds on the size of $[F_g]$ thus giving bounds on the total time of our imbedding algorithm as well as bounds on the size and number of forbidden$_g$ subgraphs.

We have carefully written this abstract to convey the crucial ideas and algorithmic techniques in the main portion of the paper, but various proofs have been placed in the Appendices. Appendix A gives some preliminary definitions of graph imbeddings. Appendix B gives a proof of the results of Section 3. While there are known linear time solutions to the 2-satisfiability problem, for completeness we give a particularly simple linear time algorithm in Appendix C. (The full paper also gives an NC algorithm for this problem). Appendix D gives a reduction from certain graph imbedding extension problems which are 2-constrained to 2-satisfiability. Appendix E gives a reduction of the problem of finding forbidden$_g$ subgraphs of imbedded graphs to finding forbidden$_0$ subgraphs. Appendix F gives an algorithm for defining a forbidden$_g$ subgraph for F_g of G of branchsize $[F_g]^{O(g)}$.

3. Reduction to Planarity Testing and 2-Constrained Imbedding Problems

3.1. Quasiplanar and Non-Splittable Extensions

First let us introduce some additional definitions and notations. Let H be a biconnected graph with imbedding $I(H)$ on a surface of genus g. If $g>0$ then some vertices may appear more than once on a single face. Let an *internal vertex* (*edge*) of $I(H)$ be a vertex(edge, respectively) of H which appears at least twice in a single face F of $I(H)$ **(See Figure 3.1)**. The *degree in F* of an internal vertex v of F is the number of times v appears on F. If a maximal 2-path contains one internal edge, then all its vertices are internal. Such path will be called *a maximal internal 2-path*. For any path p let $\hat{p}$ denote the set of all vertices of p except the endpoints. From the definitions we get

Observation 3.1. A maximal 2-path p is a maximal internal 2-path iff $\hat{p}$ belongs to exactly one face of $I(H)$.

Observation 3.2. Let p be a maximal internal 2-path and let v be an endpoint of p. Then the degree of v in the face containing p is at least 3.

Note that Observation 3.2 would have not been true if the assumption that H is biconnected were not made.

Definition 3.1. Let H be a subgraph of G and let $I(H)$ be an imbedding of H of genus g. $I(H')$ is an *imbedding extension* of $I(H)$ regarding G if $H \le H' \le G$, $I(H')$ is also an imbedding of H' of genus g and furthermore the orientation of edges around vertices of $I(H')$ is consistent with the orientation of those edges around vertices as given in $I(H)$.

If not mentioned otherwise, in this section H will denote a subgraph of G of genus g and $I(H)$ will denote a genus g imbedding of H. We will define an internal 2-path p to be *constrained* (with respect to a specified side *right* or *left of p*) if in any imbedding extension of $I(H)$, we require all attachment edges to p to be imbedded to the specified side of the ordered p (see Section 1.2 for definition of sides).

Definition 3.2. Let k be an integer, p be a 2-path of G with endpoints in H and g be the genus of $I(H)$. If there do not exist more than k extensions of $I(H)$ to $H+p$ of genus g, then p is called a *k-constrained path* with respect to $I(H)$. If B is a bridge of $G-H$ and there do not exist more than k extensions of genus g of $I(H)$ to $H+B$, then B is called a *k-constrained bridge* with respect to $I(H)$.

For example, if G is planar, then any 2-path of G with endpoints in H will be 2-constrained. As well known, planarity testing requires linear time [HT]. More generally, if all paths of G with endpoints in H are 2-constrained, Appendix C shows that finding an extension imbedding of $I(H)$ requires polynomial time.

Definition 3.3. $I(H')$ is a *weakly quasiplanar extension* (WQPE) of $I(H)$ regarding G if:

1) $I(H')$ is an extension of $I(H)$ regarding G and

2) Any bridge of $G-H'$ is 2-constrained.

For comparison, in [Filotti, Miller, Reif, 1979] the stronger notion of a *quasiplanar extension* (QPE) was used, where $I(H')$ is a QPE of $I(H)$ regarding G if no face of $I(H')$ has any internal vertex. We shall prove in this section that given an imbedding $I(H_g)$ of H_g of genus g, a possible WQPE of $I(H_g)$ regarding G can be found in $|G|[H_g]O((g^3)!)$ time.

The advantage of having a WQPE (or QPE) of $I(H_g)$ is that the corresponding extension problem can be reduced to solving a 2-satisfiability problem(see Section 4). Appendix C shows that this 2-satisfiability problem can be solved in polynomial time.

For each face F of $I(H)$ define $SF = \Sigma_{v \in M(F)}$ $(\deg_F(v)\text{-}2)$,
where $\deg_F(v)$ denotes the degree of v in F (see definition above) and $M(F)$ denotes the set of all nonconstrained internal vertices v of F such that $\deg_F(v) \geq 3$. Let $S(I(H))$ denote the sum of SF over all faces F of H. Then by Observation 3.2 it follows that we can obtain an upper bound on the number of maximal 2-paths if we can estimate $S(I(H))$. By the next lemma we provide an upper bound on $S(I(H))$.

Lemma 3.1. $S(I(H)) = O(g)$.

Proof: Makes simple use of Euler's formula; see Appendix B

Corollary 3.1. The number of all maximal internal 2-paths of $I(H_g)$ is $O(g)$.

Recently [Bender and Richmond,1990] have obtained a similar result as Corollary 3.1 giving the exact worst-case bound of $6g$-3 on the number of all maximal internal 2-paths of $I(H_g)$ $(g \geq 1)$.

Note that an imbedding $I(H^*)$ is a WQPE iff $S(I(H^*)) = 0$.

Let $F = (q_1,q,q_2,q^*)$ be a face of $I(H)$ where q and q^* are maximal directed internal 2-paths. Let p be a path in $G\text{-}H$ with endpoints on F. If an imbedding of p in F is such that the endpoints of p are on q_1 and q_2 respectively, then the imbedding will be called a *splitting imbedding* of p and p will be called a *splitting imbedded path.* **(See Figure 3.2.)** We define p as a *splitting path* for F, if any imbedding of p is a splitting imbedding.

Lemma 3.2. Any splitting imbedding of a path of $G\text{-}H$ onto $I(H)$ decreases the number $S(I(H))$ defined in Lemma 3.1 or splits a face F into 2 faces F_1 and F_2 such that $SF=SF_1+SF_2$ and $SF_1>0$, $SF_2>0$.

Proof. See [Filotti, Miller, Reif,79].

Definition 3.4. Let H be a subgraph of G and $I(H)$ be an imbedding of H. A *non-splittable extension* (NSE) of $I(H)$ regarding G is an imbedding $I(H')$ of a graph H', where $H \leq H' \leq G$ such that

1) $I(H')$ is an extension of $I(H)$;

2) No splitting path of $G\text{-}H'$ exists.

To find a WQPE of $I(H_g)$ regarding G, where H_g is a subgraph of G and $I(H_g)$ is an imbedding of H_g of genus g, we first construct an NSE $I(H'_g)$ of $I(H_g)$ regarding G and then we find a WQPE of $I(H'_g)$ regarding G.

Lemma 3.3. Let $I(H_g)$ be an imbedding of a graph H_g of genus g. Then a NSE $I(H'_g)$ of $I(H_g)$ regarding G can be constructed in $|G|[H_g]2^{O(g)}$ time.

Proof. We apply a simple procedure that repeatedly ($O(g)$ times) chooses any splitting path and guesses an imbedding of the path. See Algorithm 3.1 in Appendix B for details.

Define a bridge B of G- H'_g to be *internal* if B is incident to at least one internal vertex of $I(H'_g)$. Next in Appendix B we present an algorithm to extend an NSE $I(H'_g)$ of H'_g to an imbedding of G, if such an extension exists.

Algorithm 3.2

***Input*:** A non-splittable imbedding $I(H'_g)$ of H'_g in G

***Output*:** A subgraph H''_g of G, an imbedding $I(H''_g)$ of H''_g of genus g and a minimal set R of bridges of G-H''_g that cannot be all imbedded onto $I(H''_g)$ without increasing the genus.

The details of this Algorithm are given in Appendix B

This Algorithm 3.2 calls the algorithm from Appendix C for the 2-satisfiability problem. The total complexity is that of Algorithm 3.2.

Lemma 3.4. Given a non-splittable imbedding $I(H'_g)$ of H'_g in G we can find a subgraph H''_g of G and a minimal set R of bridges of G-H''_g that cannot be imbedded onto $I(H''_g)$ without increasing the genus in $2^{O(\mathrm{g})}n^{O(1)}$ time.

4. WQPE Imbeddings: The Reduction to 2-Satisfiability

Let H be a subgraph of G with a fixed imbedding $I(H)$ and let B be a bridge of G-H. We call B *incident* to face F of $I(H)$ if all attachment vertices of B (with respect to subgraph H) are in F. Note that B may be incident to more than one face. Since we are concerned with WQPE imbeddings, we can assume throughout this section that all bridges are 2-constrained. Let bridges B, B' *interlace in F* (with respect to given imbedding of B and B' in F) if both these bridges are incident to F and furthermore either (i) F contains distinct vertices ordered u, u', v, v' where u, v are in B and u', v' are in B' **(see Figure 4.1)** or (ii) both B and B' are attached to the same 3 distinct vertices of F. Recall the definition of inside and outside of an imbedded directed cycle given in Section 1.2.

Proposition 4.1 Let each of B, B' can be separately imbedded inside a face F. Then B, B' can be simultaneously imbedded inside F iff B, B' do not interlace.

Let $I(H)$ be a weakly quasiplanar imbedding regarding G. Suppose that any bridge of G-H can be added to $I(H)$. We shall investigate the problem of determining when *all* bridges of G-H can be added to $I(H)$. We are going to show that if all bridges cannot be added to $I(H)$, we can choose a suitable small subset of bridges, *a forbidden*$_g$ *set of bridges*, that still cannot be added to $I(H)$.

The *2-satisfiability problem* is to determine satisfiability of a boolean CNF formula with at most 2 literals per clause. While there are known linear time solutions to this problem, for completeness we give a particularly simple linear time algorithm in Appendix C.

First we define a 2-*constraint formula* $K = K(I(H)|G)$ as follows: For any bridge B of G-H construct literals $v_{B,F}$ and $\tilde{v}_{B,F}$, where B is incident to directed face F. Literal **Error!**. *If* $v_{B,F}$ *is fixed to a boolean value, then its complement* **Error!**$_{B,F}$ *is fixed to the complement* *boolean value. To complete the definition of K, for all interlacing bridges B and B' define a clause (* $\tilde{v}_{B,F}$ ***or*** $\tilde{v}_{B',F}$*) of K, where B and B' are incident to face F.*

Definition 4.1. *The 2-constraint formula K = K(I(H)|G) is said to be satisfiable, if there exists an assignment of* ***true*** *and* ***false*** *to the literals of K such that each clause of K is satisfied.*

From these definitions we easily obtain:

Lemma 4.1*. I(H) is extendible to G iff any bridge of G-H can be added to I(H) and the 2-constraint formula K = K(I(H)|G) is satisfiable.*

Proof. *see Appendix D*

Algorithm 4.1

Input *I(H)*, a weakly quasiplanar imbedding regarding *G*

Output value assignment on literals of a satisfiable constraint graph *K*=*K*(*I(H)*|*G*)

The Algorithm is quite simple ; see Appendix C for details

By the definition of *K* and Proposition 4.1, it follows that

Observation 4.1. The values assigned by Algorithm 4.1 satisfy the following properties:

1) All fixed literals receive their defined value;

2) For any literal $v_{B,F}$ in *K*, the value of $\tilde{v}_{B,F}$ is equal to the negation of the value of $v_{B,F}$.

Assume that *K* is satisfiable and the literals of *K* have values assigned by Algorithm 4.1. We imbed the bridges of *G-H* according to these values. This gives a simple procedure for extending our imbedding of H_g to include the bridges of $G\text{-}H_g$. In Appendix D we prove that the resulting imbedding extension still has genus g.

The results of this section can be used as a basis of the following algorithm for finding a forbidden_g set *C* of bridges for *K*(*H*|*G*), if *I(H)* is not extendible to *G*. Otherwise the algorithm extends *I(H)* to an imbedding of *G* of the same genus.

Algorithm 4.2

Input: *I(H)*, a weakly quasiplanar imbedding regarding *G*

Output: Imbedding of *G* of the same genus as *I(H)*, if *I(H)* is extendible to *G*, or a path *p* in *K* whose vertices correspond to a forbidden_g set *C* of bridges

1. Construct the graph *K* = *K*(*I(H)*|*G*).
2. Apply the algorithm of Appendix C

Step 1 can be implemented in $O(|V(G)|^2)$ time The time required by the Steps of Algorithm of Appendix C is $O(|K|)$, if breadth-first search is used.

5. Bounds on Total Time and the Size and Number of Forbidden$_g$ Subgraphs

Lemma 5.1 $[F_g] \leq \exp(O(g)!)$

Proof by induction. F_0 is known to have branchsize $O(1)$. For any $g \geq 0$, our inductive assumption is that $[F_g] \leq \exp((cg)!)$, for some constant $c > 1$. The bound $[F_g]^{O(g)}$ on the branchsize of F_{g+1} proved in Appendix F implies that F_{g+1} is of branchsize $[F_g]^{c'g}$, for some constant $c' > 1$. Thus $[F_{g+1}] \leq \exp((cg)!\ c'g) \leq \exp((c(g+1))!)$, for $c > c'$. **Q.E.D.**

Our Imbedding Algorithm 2.1 requires for each value of $g' = 0,1,...,g = \text{genus}(G)$, the construction of all possible skeletal subimbeddings $I(H_{g'})$ of $SI_{g'}\ (F_g,T)$, which can be computed by the Algorithm of Section 2.2 in deterministic time $(g[F_{g'}])^{O(g')} = \exp(O(g'!))$. For each possible skeletal subimbedding Algorithm 3.2 spends time (gotten by summing the times given in Lemmas 3.3, 3.4, and E.2) at most $|G|[H_{g'}]2^{O(g')}+2^{O(g')}n^{O(1)}+|G|^{O(1)}$. Then the time needed for each g' is

$$\exp(O(g'!))\ (\ |G|[H_{g'}]2^{O(g')}+2^{O(g')}n^{O(1)}+|G|^{O(1)}\ ) + [F_{g'}]^{O(g')},$$

where $[F_{g'}]^{O(g')}$ is the time given in Lemma 5.2. The sum of these enumeration times for $g' = 0,1,...,g=\text{genus}(G)$ gives the total deterministic time of our Imbedding Algorithm 1 as $\exp(O(g!))\ (\ |G|[H_g]2^{O(g)}+2^{O(g)}n^{O(1)}+|G|^{O(1)}\ ) = \exp(O(g!))\ |G|^{O(1)}$. This completes our analysis of the time complexity of the algorithm and implies our main result:

Theorem 5.1 Given an input graph G of genus g, our Imbedding Algorithm 2.1 takes time $\exp(O(g)!)\ |G|^{O(1)}$. Furthermore if $g > 0$, our algorithm yields a forbidden$_{g-1}$ subgraph F_g of branchsize $[F_g] \leq \exp(O(g!))$.

A simple parallel implementation of our algorithm gives:

Theorem 5.2 Given an input graph G of genus g, our Imbedding Algorithm takes parallel time $O(g!) + (\log n)^{O(1)}$ using $\exp(O(g!))\ n^{O(1)}$ processors.

Note that there are at most b^b homeomorphic distinct graphs of branchsize $b \leq \exp(O(g!))$. This immediately implies that the number of homeomorphic distinct forbidden$_g$ subgraphs for any graph imbedding onto a surface of genus g, is at most a triple exponential function of g. However, we can get an even smaller bound.

Theorem 5.3 The number $F(g)$ of homeomorphic distinct forbidden$_g$ subgraphs for graph imbeddings onto a surface of genus g, is at most $\exp(O(g!))$.

Proof by induction. The number of homeomorphic distinct forbidden$_0$ subgraphs is $F(0) = 2$. For any $g \geq 0$, our inductive assumption is that

$$F(g) \leq \exp((dg)!), \text{ for some constant } d > 1.$$

The bound of Lemma 5.1 on the branchsize $[F_g] \leq \exp(O(g!))$ immediately implies that

$$F(g+1) \leq \max([F_g])F(g)^{d'g} \leq \exp((cg)!)\ F(g)^{d'g}, \text{ for some constants } c, d' > 1.$$

Thus $F(g+1) \leq \exp((cg)!)\ \exp((dg)!\ \ d'g) \leq \exp((d(g+1))!))$, for sufficiently large d. **Q.E.D.**

6. Conclusion

It would be of great interest to provide lower bounds on the number of forbidden subgraphs as a function of *g*.

References

Alpert, S.R. The genera of amalgamations of graphs. *Trans. Amer. Math. Soc.*, 178 (1973) 1-39.

Alpert, S.R., and Gross J. L. Graph Imbedding Problems, in *Research Problems* (Richard Guy, ed.).

Battle, J., Harary, F., Kodama, Y., and Youngs, J.W.T. "Additivity of the Genus of a Graph". *Bull. Amer. Math. Soc. 68* (1962), 565-568.

Bender, E.A., and Richmond, L.B. 3-Edge-Connected Embeddings Have Few Singular Edges, *J. Graph Theory*, 14(1990), 475-477.

Djidjev, H.N., A linear algorithm for partitioning graphs of fixed genus, Serdica Bulgaricae mathematicae publications, Vol 11, (1985), 369-387.

Edmonds, J. A combinatorial representation for polyhedral surfaces. *Not. Am. Math. Soc.* 7 (1960), 646.

Fellows, M. R., and Langston, M.A, Nonconstructive Advances in Polynomial-Time Complexity, *Info. Proc. Letters* 26, (1987), 157-162.

Fellows, M. R., and Langston, M.A, Nonconstructive Tools for Proving Polynomial-Time Decidability", J of the ACM 35, (1988) 727-739.

Fellows, M. R., and Langston, M.A, Nonconstructive Advances in Polynomial-Time Complexity, *Info. Proc. Letters* 26, (1987), 157-162.

Filotti, I.S. An algorithm for imbedding cubic graphs in the torus, *J. Comput. Syst. Sci.* 20 (1980), 255-276.

Filotti, I.S., Miller, G.L., and Reif, J.H. On determining the genus of a graph in $0(v^{0(g)})$ steps. In *Proceedings of the 11th ACM Symposium on Theory of Computing*, Atlanta, Ga., April, 1979, pp. 27-37.

Furst, Merrick L., Gross, Jonathan L., and McGeoch, Lyle A. Finding a Maximum-Genus Graph Imbedding, *JACM*, Vol 35, No. 3, July 1988, pp. 523-534

Gross, J.L., and Rosen, R.H. A linear time planarity algorithm for 2-complexes. *J. ACM* 26, 4 (Oct. 1979), 611-617.

Gross, J.L., and Tucker, T.W. *Topological Graph Theory.* Wiley Interscience, New York, 1987.

Hopcroft, J., and Tarjan, R. Efficient planarity testing. *J. ACM 21,* (Oct. 1974), 549-568.

Khuller, S., S.G. Mitchell, and V.V. Vazirani, "Processor efficient parallel algorithms for the two disjoint paths problem, and for finding a Kuratowski homeomorph", 30th Annual Symposium on Foundations of Computer Sceince", Durham, NC, Oct, 1989, p 300-305

Klein, P. and J.H. Reif, An Efficient parallel algorithm for Planarity", J. Comp. Sys. Sci., vol 37, 1988.

Kruskal, J. "Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture", *Trans. Amer. Math. Soc. 95* (1960), 210-225.

Kuratowski, "Sur le problem des courbes gauches en topologie", *Fund. Math. 15* (1930), 271-283.

Massey, W.S. *Algebraic Topology: An Introduction,* Harcourt, Brace, and World, New York, 1967.

Miller, G. L. "Isomorphism Testing of Graphs of Bounded Genus. *Proceedings of 12th Annual ACM Symposium on Theory of Computing*, ACM, Apr. 1980, 225-235.

ibid, *Lecture Notes in Computer Science*. Vol. 158: Testing and Canonical Forms for k-Contractible Graphs (A Generalization of Bounded Valence and Bounded Genus). In *Foundations of Computation Theory,* Springer-Verlag, N.Y., 1983, 310-327.

ibid, "Isomorphism of k-contractible Graphs (a Generalization of Bounded Valence and Bounded Genus)". *Information & Control 65,* 1-2 (Jan/Feb 1983), 1-20.

ibid, An Additivity Theorem for the Genus of a Graph, MIT, TR-85-342.

Ramachandran, V. and J.H. Reif, An optimal parallel algorithm for planarity, 30th Annual Symposium on Foundations of Computer Sceince", Durham, NC, Oct 1989.

Reif, J. On the Complexity of Extending Partial Imbeddings, Computer Science Dept, Univ Rochester, Technical Report TR33, Oct 1978.

ibid, Polynomial Time Recognition of Graphs of Fixed Genus, Computer Science Dept, Univ Rochester, Technical Report TR33, Oct 1978.

Robertson, N., and Seymour, P.D. "Graph minors. I. Excluding a forest", *J. Combinatorial Theory* (Ser. B), 39-61.
ibid, "Graph minors. II. Algorithmic aspects of tree-width", *J. Algorithms,* 7 (1986), 309-322.
ibid, "Graph minors. III. Planar tree-width", *J. Combinatorial Theory* (Ser. B), 49-64.
ibid, "Graph minors. IV. Tree-width and well-quasi-ordering". submitted.
ibid, "Graph minors. V.. Excluding a Planar Graph, *J. Combinatorial Theory* (Ser. B), (1986) 92-114.
ibid, "Graph Minors. VI. Disjoint paths across a disc", preprint, Sept 1986
ibid, "Graph Minors. XII. Disjoint paths on a surface", preprint, Sept 1986
ibid, "Graph minors. VIII. A Kuratowski theorem for general surfaces", preprint, Sept 1986
ibid, "Graph Minors. XIII. The Disjoint paths problem", preprint, Sept 1986
ibid, "Graph Minors. XVI. Wagner's Conjecture", preprint, to appear
ibid, "Graph Minors - a Survey", in *Surveys in Combinatorics* (I. Anderson, ed.), *Cambridge Univ. Press* (1985), 153-171.

Thomassen, "The graph genus problem is NP-complete", *J. of Algorithms*, 10(1989), 568-576.

Tutte, W.T. "Combinatorial Oriented Maps". *Can. J. Math. 31,* 5 (1979), 986-1004.

Wagner, K. "Uber Einer Eigenshaft der Ebener Complexe", *Math. Ann.*14 (1937), 570-590.

White, A.T. *Graphs, Groups, and Surfaces*, *North-Holland*, Amsterdam, 1973.

Youngs, J.W.T. Minimal imbeddings and the genus of a graph, *J. Math. Mech*., 12 (1963) 303-315. MR 26 #3043.

Appendix A Preliminary Definitions of Imbeddings

1.1. Topological Imbeddings

Throughout this paper, we only consider *surfaces* which are orientable 2-manifolds. For our purposes, such surfaces can be uniquely characterized by their *genus* g. Informally, a closed surface of genus g consists of a sphere with the addition of g handles. For example, the sphere has genus 0 and the torus has genus 1. The plane also has genus 0 but is not closed. A *topological imbedding* of an undirected graph $G = (V, E)$ is a mapping of G onto a surface S of genus g (this is also called a *2-cell imbedding*; see [White, 1973]), where each edge is associated with a simple segment on the surface S, where the vertices of the edge are at the two distinct endpoints of the segment, and where no two such edges intersect except at endpoints in the case of common vertices. The *faces* of the imbedding will be defined to be the boundaries of the connected regions gotten by deleting the imbedding of G from the surface. Euler's equation gives $n-m+f = 2c-2g$, where m, n, f, c, g are the numbers of edges, vertices, faces, connected components, and genus of the imbedding, respectively.

1.2. Combinatorial Imbeddings

The topological definition of graph imbedding given above presents difficulties to computer algorithms and their proofs. An alternative (but equivalent) definition will better serve our purpose. Given an undirected graph $G = (V,E)$, let n, m denote the number of vertices and edges, respectively and let its *size* $|G|= n+m$. We will represent an imbedding of graph G in a compact way of size $|G|$ by use of a combinatorial definition of graph imbeddings that is attributed to [Edmonds,60]. Let $D(G)$ be a directed graph derived from G by substituting in place of each undirected edge $\{u,v\}$, a pair of directed edges (u,v) and (v,u). A *combinatorial graph imbedding* $I(G)$ of (undirected connected) graph G is an assignment of a cyclic ordering to the set of these directed edges departing each vertex. **(See Figure 1.1)** The *faces* of this combinatorial imbedding will be defined to be the orbits of a certain permutation of directed edges; this permutation orders (w,v) immediately before (v,u) iff the combinatorial imbedding orders (v,u) immediately before (v,w) in the cyclic order around vertex v. **(See Figure 1.2)** The *genus* of a combinatorial imbedding is defined to be $g = (m-n-f)/2+c$ by the Euler formula using the numbers of (undirected) edges m, vertices n, faces f, and connected components c. Edmonds (see also [White,73]) showed that combinatorial imbeddings are equivalent to topological imbeddings. The advantage of combinatorial imbeddings is not just that they can be represented in size $|G|$ in a random access computer. An important additional advantage is that definitions and proofs about such imbeddings can be made entirely combinatorial. For example, given a directed simple cycle $C = (v_0,v_1,...,v_k= v_0)$ of G and an edge $\{v_i,x\}$ where x is not in C but v_i is in C, we define $\{v_i,x\}$ to be imbedded *inside* C (and otherwise *outside* C) if (in the cyclic order defined by $I(G)$ on the directed edges departing vertex v_i) directed edge (v_i,x) appears after directed edge (v_i,v_{i+1}) and before directed edge (v_i,v_{i-1}). **(see Figure 1.3)** As another example, given a directed simple path $p = (v_0,v_1,...,v_k)$ of G and an edge $\{v_i,x\}$ where x is not in p but v_i is in p, we define $\{v_i,x\}$ to be imbedded to the *right* of p (and otherwise *left* of p) if (in the cyclic order defined by $I(G)$ on directed edges departing vertex v_i) directed edge (v_i,x) appears after directed edge (v_i,v_{i+1}) and before directed edge (v_i,v_{i-1}). Hereafter, we will simply use the term *imbedding* to denote a combinatorial imbedding.

Appendix B Reduction to 2-Constrained Imbedding Extension Problems

B3.1. Quasiplanar and Non-Splittable Extensions

Lemma 3.1. *$S(I(H)) = O(g)$.*

Proof: The Euler characteristic of H is $m-n-f = 2g-1$ where m,n, f are the number of edges, vertices, and faces of $I(H)$. Build a spanning tree of H. Suppose that there exists a nontree edge e incident with two different faces of $I(H)$. Then e is a non-internal edge. Define a maximal 2-path p in H containing e. From Observation 3.1 $\hat{p}$ will contain no internal vertices. Remove $\hat{p}$ and merge the faces. Since the genus is the same, the Euler characteristic remains invariant. Repeat until no such edge e exists. The resulting graph H^* has 1 face and Euler characteristic $2g-2$. Therefore

(3.1) $\qquad m_1\text{-}n_1\text{-}1=2g\text{-}2,$

where n_1 and m_1 are the number of vertices and edges of H^*. As obviously the deletion of the vertices of $\hat{p}$ (which are not internal) from the corresponding graph does not decrease the current value of $S(I(H))$, then $S(I(H)) \leq S(I(H^*))$. We will show that the minimum vertex degree of H^* is at least 2 whence $S(I(H^*))=2(m_1\text{-}n_1)$ will follow.

As H is biconnected, then the degree of H is at least 2. Let w be an endpoint of p. Then w can not be incident to only one edge of H. Furthermore, since p is maximal, then w can not be incident to exactly two edges of H. Therefore w is incident to at least three edges of H. Then the removal of $\hat{p}$ from H does not create vertices of degree one or zero. Thus the degree of H' is at least 2, where $H'=H\text{-}p$. By induction, the minimum vertex degree of H^* is at least 2. By the definition of $S(I(H^*))$ and (3.1) we get

$S(I(H^*))=2(m_1\text{-}n_1)=4g\text{-}2.$ **Q.E.D.**

Lemma 3.3. Let $I(H_g)$ be an imbedding of a graph $H_g \leq G$ of genus g. Then a NSE $I(H'_g)$ of $I(H_g)$ regarding G can be constructed in $|G|[H_g]2^{O(g)}$ time.

Proof. Apply the following procedure to $I(H_g)$.

Algorithm 3.1

```
Input  biconnected graph $G$, a subgraph $H_g$ of $G$ and a genus $g$ imbedding $I(H_g)$
Output $I(H'_g)$, an NSE of $I(H_g)$ regarding $G$
Initially let $H'_g := H_g$ and $I(H'_g) := I(H_g)$.
repeat do
    1. Pick any splitting path $p$ in $G\text{-}H'_g$
    If no such splitting path $p$ exists, then output $I(H'_g)$; halt
    else
    2. $H'_g := H'_g+p$.
    3. Guess an imbedding of $p$ and add it to $I(H'_g)$.
    fi od
```

Denote $h=[H_g]$ and $s=S(I(H_g))$. To estimate the number of all possible ways to imbed p in Step 3 we notice that, since both endpoints of p are of degree 2 in H'_g, there are no more than 2 possible choices of imbedding each end edge of p. By Lemma 3.2, each Step 3 decreases the current value of $S(I(H'_g))$ by at least 1 or will not change $S(I(H'_g))$ but will increase the number of faces F with $SF>0$. All executions of Step 1 will require $O(|G|[H_g])$ time. Then, by the above arguments, the maximum time $T(h,s)$ required by Steps 2 and 3 of Algorithm 3.1 to make any imbedding with parameters h and s an NSE can be upper bounded by

$T(h,s) \leq 4T(h,s\text{-}1)+ O(|G|\, h)$

$\leq O(|G|\, h)2^{2s}.$

Thus the total sequential time required by Algorithm 3.1 is $|G|[H_g]2^{O(g)}$. **Q.E.D.**

Define a bridge B of $G\text{-}\, H'_g$ to be *internal* if B is incident to at least one internal vertex of $I(H'_g)$.

Next we present an algorithm to extend an NSE $I(H'_g)$ of H'_g to an imbedding of G, if such an extension exists, or otherwise to a WQPE. The algorithm will use the algorithm of Appendix C for the 2-satisfiability problem and the algorithm of Section B3.2 below.

Algorithm 3.2

Input: A non-splittable imbedding $I(H'_g)$ of H'_g in G

Output: A subgraph H''_g of G, an imbedding $I(H''_g)$ of H''_g of genus g and a minimal set R of bridges of $G\text{-}H''_g$ that cannot be all imbedded onto $I(H''_g)$ without increasing the genus

```
        Repeat do
1.          Pick a maximal sequence S=$p_1,...,p_k$ of maximal 2-paths such that there exists a face $F$ such that either $F$={S$_1$...$p^*$...S$_2$...$p^{**}$...}, or $F$={S$_1$...$p^*$...S$_3$...$p^{**}$...}, where S$_1$=$p_1,p'_1, ... , p_{k-1},p'_{k-1},p_k$ , S$_2$=$(p_1)^R,p"_1, ... ,(p_{k-1})^R p"_{k-1},(p_k)^R$ and S$_3$=$(p_k)^R,p"_{k-1},..., (p_2)^R,p"_1,(p_1)^R$, paths $p^*$, $p^{**}$ and $p_i$ are internal and paths $p'_i$ and $p"_i$ are non-internal. Set $H"_g:=H'_g$.
2.          Let $G'$ be the graph containing all bridges of $G$-$H"_g$ attached only to paths from $S_1$ and $S_2$ (respectively $S_3$). Apply the algorithm of Appendix C for the 2-satisfiability problem on $I(H"_g)$ and $G'$ to test whether there exists an extension imbedding, $I(G')$, of $G'$ (regarding $I(H"_g)$) such that any bridge of $G'$-$H"_g$ is attached either only to vertices on paths in $S_1$, or only to vertices on paths in $S_2$ (respectively $S_3$).
3.          If the extension from Step 2 does not exist then do
3.1.            Let $FS$ be the forbidden$_0$ subgraph of $O(1)$ branchsize given by the algorithm of Appendix E.
3.2.            Guess an imbedding of $FS$ and add it to $I(H"_g)$
3.3.            $H"_g:=H"_g+FS$
            od
        od
4. Comment: Treat each face $F$ individually for imbedding internal bridges with endpoints in $F$
4.1.        Let $M$ be the set of all internal bridges of $G$-$H"_g$. Find a suitable set $Q$ of $O(g)$ bridges of $G$-$H"_g$ and guess an imbedding of those bridges in $I(H"_g)$. Comment: As described below the resulting imbedding extension problem will be WQPE.
4.2.        Apply the algorithm from Appendix C to solve the arising 2-satisfiability problem in order to imbed the remaining bridges of $M$ onto $I(H"_g)$ or to find a minimal set $R$ of bridges of $G$-$H"_g$ that cannot be imbedded onto $I(H"_g)$ without increasing the genus.
```

Notice that as the original imbedding $I(H"_g)$ is non-splittable, it remains such after each subsequent update of $H"_g$ and $I(H"_g)$. Then if the extension $I(G')$ from Step 2 exists, $I(G')$ can be extended to an imbedding of G iff $I(H"_g)$ can be extended to an imbedding of G.

Provided that the set Q with the properties described in Step 4.1 exists, a similar analysis as for Algorithm 3.1 leads to the upper bound of $2^{O(g)}t_1(n)+t_2(g,n)$ on the time needed by Algorithm 3.2, where $t_1(n)$ and $t_2(g,n)$ are polynomials of n upper bounding the maximum time needed respectively by the algorithm from Appendix C (for the 2-satisfiablility problem) and by the algorithm for constructing Q. Below we show that Q exists and can be found in $2^{O(g)}n^{O(1)}$ time. Thus Algorithm 3.2 also takes $t_2(g,n)=2^{O(g)}n^{O(1)}$ time.

B3.2. Weakly Monotonic Sequences

Let $p=(v_0,v_1,...,v_j)$ and p' be maximal internal 2-paths. Let $F=(p,p_1,p',p_2,p^R,p_3,(p')^R,p_4)$ be a face of $I(H"_g)$ that is considered at some iteration in Step 4 of Algorithm 3.2, where p_i, i=1,...,4, are paths (**Figure 3.3**). W.l.o.g. we can assume that

(1) p_3 contains an internal 2-path (because of Step 2 of Algorithm 3.2), and

(2) p_2 also contains an internal 2-path (because otherwise F will be decomposable into sequences of paths each of the form $(z,z_1,z',z_2,z^R,z_3,(z')^R,z_4)$, where z and z' are maximal 2-paths and paths z_i, i=1,...,4, contain no internal edges; then each such sequence can be treated independently).

Notice that any imbedding of a bridge with one attachment vertex on p^R and another attachment vertex on p, p', or $(p')^R$ will be a splitting imbedding for F.

We will find a genus g extension of $I(H"_g)$ in which there will be no bridge of G-$H"_g$ attached both to p and p', if possible. Let originally there exist in F at least one bridge of G-$H"_g$ attached both to p and p'.

Remove from G-F all attachment edges to $\hat{p}_3$ and $\hat{p}_4$. For each integer $i \leq k$ define a graph G_i by deleting from the resulting graph all components whose attachment vertices are in $\{v_{i+1},...,v_k\}$. Let i^*, $1 \leq i^* \leq k+1$, be the maximum integer such that G_{i^*-1} can be imbedded onto F with p' constrained to left. The following three cases exist: $I(H''_g)$ can not be extended to imbed G_k; $I(H''_g)$ can be extended to imbed G_k with p' constrained to left; $I(H''_g)$ can be extended to imbed G_k, but in no extension of $I(H''_g)$ p' is constrained to left. Let us consider these cases in more detail.

Case 1. $I(H''_g)$ can not be extended to imbed G_k (without increasing the genus g). Then G_k contains a forbidden$_g$ subgraph for $I(H''_g)$ of $O(1)$ branchsize (see Appendix E).

Case 2. $I(H''_g)$ can be extended to imbed G_k and there exists an extension of $I(H''_g)$ in which p' is constrained to left.

We say for any two vertices z_i and z_j belonging to the same directed path $u=(z_1,z_2,...)$ that z_i *appears before* (respectively *after*) z_j on u, if $i \leq j$ (resp. $i \geq j$). Let □$=B_1,...,B_b$ be the sequence of the connected components of G_k and for each $i<i' \leq b$ any attachment vertex of B_i on p appears before any attachment vertex of $B_{i'}$ on p. (Note that B_i, $1 \leq i \leq b$, are bridges of G-H''_g). We call □ *weakly monotonically increasing* (resp. *decreasing*) sequence, if for each $1 \leq i<i' \leq b$ any attachment vertex of B_i on p' appears before (resp. after) any attachment vertex $B_{i'}$ on p' (**Figure 3.4**). □ is *weakly monotonic*, if it is either weakly monotonically increasing or weakly monotonically decreasing sequence.

Observation 3.3. If G_k can be imbedded onto F with p' constrained to left, then there is (in general non-unique) decomposition of G_k into two weakly monotonic sequences. Given the imbedding of G_k, such a decomposition can be found in $O(|G|)$ time.

Proof: Imbed G_k onto F with p' constrained to left. Let □$_1$ contain all bridges whose attachment edges to p are imbedded to left of p and let □$_2$ contain all bridges whose attachment edges to p are imbedded to right of p. If there is a bridge B attached to p both to left and to right of p, then insert B in (say) □$_1$. Then □$_1$ is a weakly increasing and □$_2$ is a weakly decreasing sequence. Defining □$_1$ and □$_2$ takes $O(|G|)$ time. **Q.E.D.**

We now address the problem of imbedding weakly monotonic sequences. Let $W=M_0,...,M_m$ be a weakly monotonic sequence of bridges of F (e.g. one of the weakly monotonic sequences forming the decomposition of the sequence □). Assume that the sequence is increasing. (Decreasing sequences are considered analogously.) Let us choose an imbedding of M_0 and M_m. Remember that if an imbedding of M_0 (respectively M_m) has one attachment vertex on p^R and another attachment vertex on p, p', or $(p')^R$, then M_0 (respectively M_m) will contain a splitting imbedded path for F. Let us consider all other remaining ways to imbed M_0 and M_m. The four possible cases are illustrated on **Figure 3.5**.

(i) M_0 and M_m are attached to p and p' (**Figure 3.5 (a)**). Then $M_1,...,M_{m-1}$ can be also attached to p and p' (such imbedding is possible as the sequence W is weakly monotonic).

(ii) M_0 is attached to p and p' and M_m is attached to p and $(p')^R$ (**Figure 3.5 (b)**). Then any bridge in W can be attached only to p' (and not to $(p')^R$) and therefore is 2-constrained. In case of multiple bridges with the same attachment vertex on p as M_0 or M_m we additionally choose M_0 and M_m to have maximum attachment vertices on the directed p'. And if there are multiple bridges with the same attachment vertices on p and p' as M_0 and M_m we imbed those bridges in the same way as M_0 and M_m.

(iii) M_0 is attached to p and $(p')^R$ and M_m is attached to p and p' (**Figure 3.5 (c)**). This will be possible only if M_0 and M_m have the same attachment vertex on p. Then all bridges in W will have the same attachment vertex on p. This case is similar to the previous Case (ii).

(iv) M_0 and Mm are attached to p and $(p')^R$ (**Figure 3.5 (d)**). Again it follows that all bridges in W will have the same attachment vertex on p. This case is similar to Case (i).

In all cases (i)-(iv) we add M_0 and M_m to Q, where Q is the set defined in Step 4.1 of Algorithm 3.2. This completes the analysis of Case 2.

Case 3. $I(H''_g)$ can be extended to imbed G_k, but in no extension of $I(H''_g)$ p' is constrained to left. Then a minimal subgraph, FS, of G_{i^*} of $O(1)$ branchsize can be found by the algorithm in Appendix E such that FS can not be imbedded onto $I(H''_g)$ with p' constrained to left. In any imbedding of FS onto F there will be at least one path of vertices imbedded to the right of p'. Let the attachment vertex on p of the above path be v_s, $1 \leq s \leq i^*$. Then all edges $(v_s,v_{s+1}),\ldots,(v_{j-1},v_j)$ belong to two faces of the resulting imbedding and therefore are non-internal. The case where there exist more than one bridge attached to v_s can be handled in the same way as Case 2, because the set of all such bridges determines a weakly monotonically decreasing sequence. Finally by the definition of i^* and since $s \leq i^*$ then G_{s-1} can be imbedded onto $I(H''_g)$ with p' constrained to left. Therefore we can apply to G_{s-1} the algorithm from Case□2.

Notice that in all cases we add $O(1)$ paths to the set Q and either decrease the invariant $S(I(H''g))$ which by Lemma 3.1 is $O(g)$, or make at least 2 maximal internal 2-paths, p and p', weakly quasiplanar. By Corollary 3.1 the number of all maximal internal 2-paths is $O(g)$. Thus we proved the following

Lemma 3.4. Given a non-splittable imbedding $I(H'_g)$ of H'_g in G we can find in $2^{O(g)}n^{O(1)}$ time a subgraph H''_g of G and a minimal set R of bridges of $G\text{-}H''_g$ that cannot be imbedded onto $I(H''_g)$ without increasing the genus.

474*Appendix C* The 2-Satisfiability Problem

Let F be a conjunction of 2-literal clauses each of form (L_{J_i} **or** L_{K_i}). Let G be a graph with vertex set {*true*, *false*, all literals and their complements}.

For each clause with just one literal L let there be edge from *true* to L and from (**not** L) to *false*. For each clause with two distinct literals (L_{J_i} ***or*** L_{K_i}) add an edge from (**not** L_{J_i}) to L_{K_i} and an edge from (**not** L_{K_i}) to L_{J_i} .Thus G has a directed edge for each logical implication resulting from a single clause of F.

Theorem C.1. F is unsatisfiable iff

(1) there is a path from *true* to *false* or

(2) there is a literal L such that there is a cycle containing L and (**not** L).

Proof: Clearly, there is a logical implication from a literal L to another L' iff there is a path from L to L'. Thus if either of the latter cases (1) and (2) hold, the formula is not satisfiable.

If neither of the latter cases (1), (2) hold then we claim that we can assign to the literals **true** and **false** so as to satisfy the formula.This is done as follows:

(Initialize) First compute the strongly connected components of G and collapse each strong component into a single node. All the literals in such a node have implications to and from each other, so they must have the same truth value on any satisfying assignment. We will now label nodes in such a way so as to satisfy the formula (i.e., we will not violate any clause). Let S be the node *true* if it exists and otherwise let S be the empty set.

(Loop)

(a) If S is not empty, choose any node r from S. We label all nodes w reached from r with **true**, and label **false** any remaining nodes which contain the complement of any lateral in each reachable node w.

(b) Choose if possible any node r' such that r' is any so far unlabeled node with entering edges (if any) only from nodes labelled **false**. Then we assign r' **false** and for each literal L in v, we add the node r containing (**not** L) to S and set r to **true**. Then we go back to the Loop.

(c) If there is no such r' to choose and S is empty, then we terminate.

Claim 1. All nodes are eventually labelled.

Proof: Since we have contracted all strongly connected components, we can not have a loop of so far unlabeled nodes. Thus the contracted graph is acyclic.Suppose that we complete the algorithm with an unlabeled node. Then let v be such an unlabeled node with no ancestors that are unlabeled. (Such a node always exists in an acyclic graph.) Clearly, v can have no ancestors labeled **true**, since then v would have been labeled **true**. Thus all entering edges (if any) to v are from nodes labelled **false**, so v would have been chosen to be r' and labelled **false**, a contradiction.

Claim 2. If the set S initially contains *true*, then we never reach node *false* in the first iteration, nor do we reach both a literal and its complement.

Proof: Suppose we reach the node *false* from a root node r = *true* which we labeled **true**. Thus there is a path from r to *false*.Then by construction of G, we would have also a path from *true* to a node containing the complement of a literal in r, a contradiction with the assumption that case (1) does not hold.

The second case follows from the assumption of condition 2.

Claim 3. We never reach from a root r chosen from S a node w already labeled **false** (even within a given iteration).

Proof by contradiction:. Suppose not, so then w must have been previously labeled **false**. By the construction of G, there must be a node w' previously labeled **true** containing the complement of a literal in w.Thus by definition of G there is also a path from w' to a node r' containing a complement of a literal in r. But since node w' has been previously labeled **true** thus node r' would have previously labeled **true**, and so node r would have been previously labeled **false**, a contradiction with the assumption that r is chosen to be an unlabeled node.
The above three claims imply that we satisfy the formula. **Q.E.D.**

Corollary C.1. Given a 2-satifiability problem with formula F we can determine in $O(|F|)$ time if F is satisfiable. If F is satisfiable, we can find a satisfying assignment. If F is not satisfiable, then we can find in $O(|F|)$ time a sequence of implications satisfying cases (1) or (2) of Theorem C.1.

Appendix D The Reduction of 2-Constrained Imbedding Problems to 2-Satisfiability

Lemma 4.1. *I(H)* is extendable to G iff any bridge of *G-H* can be added to *I(H)* and the 2-constraint formula $K = K(I(H)|G)$ is satisfiable.

Proof. => Assume that K is not satisfiable. Then one of the Conditions (1) and 2) from Theorem C.1 is not satisfied. Suppose that Condition (1) is not satisfied. Then for some similarily fixed literals $v_{B,F}$ and $v_{B',F'}$ there exists an implication path p in K between literals $v_{B,F}$ and $\tilde{v}_{B',F'}$. W.l.o.g. we assume that the literals $v_{B,F}$ and $v_{B',F'}$ are fixed to value **true** and p does not contain other fixed literals or their complements. Let p=($v_{B,F}=v_{B_0,F_0},v_{B_1,F_1},...,v_{B_{j-1},F_{j-1}}$, $\tilde{v}_{B',F'}$). Suppose that we try to construct an extension of *H* by imbedding the bridges B_0,B_1,...,B_j. As $v_{B,F}$ is fixed, then *B* is incident only to *F* and should be imbedded on side-1 of *F* in any extension of *I(H)* in *G*. Then if *I(H)* is extendable to *G*, then in any such extension B_0 is imbedded on side-1 of F_0, B_1 is imbedded on side-1 of F_1,..., B_{j-1} is imbedded on side-1 of F_{j-1}, *B*' is imbedded side-2 of *F*'. But the latter is impossible as $v_{B',F'}$ is fixed to value **true** and therefore *B*' can be imbedded only on side-1 of *F*'.

Suppose Condition (1) of Theorem C.1 is not satisfied. Then there exists an implication path *p*=(v_{B_0,F_0}, v_{B_1,F_1},..., v_{B_j,F_j} = $\tilde{v}_{B_0,F_0}$) in *K*. Suppose that we try to construct an extension of *H* by imbedding the bridges B_0,B_1,...,B_j. Then B_0 has to be imbedded either on side-1 or side-2 of F_0. Assume that B_0 is imbedded on side-1 of F_0. Then, by the definitions of *K* and *p*, B_1 has to be imbedded on side-1 of F_1, B_2 has to be imbedded on side-1 of F_2, ..., and hence B_j has to be imbedded on side-1 of F_j. So B_0 has to be imbedded on side-2 of F_0. Then *I(H)* is not extendable to *G*, a contradiction. Assume that B_0 is imbedded on side-2 of F_0. Then B_j has to be imbedded on side-1 of F_j, B_{j-1} has to be imbedded on side-1 of F_{j-1}, ... , B_0 has to be imbedded on side-1 of F_0, a contradiction with the assumption.

Thus in all cases the assumption that *K* is not satisfiable leads to a contradiction that *I(H)* is extendable to *G*.

<= Assume that *K* is satisfiable. Assign **true** and **false** values to the literals of *K* by the algorithm of Appendix C. Intuitively, a **true** (**false**, respectively) value to $v_{B,F}$ will correspond to an imbedding of *B* on side-1 (on side-2, respectively) of *F*. The algorithm constructs a value assignment of a satisfiable 2-constraint formula *K* = *K*(*I(H)*|*G*) in *O*(|*G*|) time. This gives the body of the following algorithm:

Algorithm 4.1
Input *I(H)*, a weakly quasiplanar imbedding regarding *G*
Output value assignment on literals of a satisfiable 2-constraint formula *K*=*K*(*I(H)*|*G*)

We have assumed above that *K* is satisfiable and the literals of *K* have values assigned by the algorithm of Appendix C. We imbed the bridges of *G* according to these values: if $v_{B,F}$ is **true** (**false**, respectively) then imbed *B* on side-1 (side-2) of *F* respectively. This gives a simple procedure for extending our imbedding of H_g to include the bridges of G-H_g. We show that the resulting imbedding extension still has genus *g*. By Observation 4.1, 1) any bridge incident to only one side of a face should be embedded on that side. Then the value assignment obtained from Algorithm 4.1 allows each bridge to be embedded into its corresponding face provided that there is no conflict with the other bridges. For the sake of contradiction, suppose that bridges *B* and *B*' are assigned to be imbedded on side-1 of the same face *F*, but cannot in fact be simultaneously imbedded on side-1 of *F*. Then $v_{B,F}$ and $v_{B',F}$ have a **true** value and *B* and *B*' interlace. From our construction of the 2-constraint formula, there exists a clause in *K* of form ($\tilde{v}_{B,F}$ **or** $\tilde{v}_{B',F}$). This clause is not satisfied as both $\tilde{v}_{B,F}$ and $\tilde{v}_{B',F}$ have a **false** value. This contradiction shows that we have defined a valid imbedding of *G* extending H_g. **Q.E.D.**

Appendix E Finding Forbidden$_g$ Subgraphs using a Planarity Algorithm

For a given WQPE *I*(H_g) we would like to find some subgraph *FS* of *G* with branchsize [*FS*]=*O*(1) that cannot be added to *I*(H_g), if the genus of H_g is less than the genus of *G*. For this end we could apply a known planarity testing algorithm [Hopcroft and Tarjan, 74] so to test in linear time whether *I*(H_g) can be extended to an imbedding of the entire graph. When the extension is not possible, a modification of the algorithm will find a forbidden$_g$ subgraph of branchsize *O*(1). In order to skip the lengthy description of that modified algorithm, we shall show here (using a somewhat less efficient, but polynomial time construction) that a forbidden$_g$ subgraph of $I(H_g)$ of branchsize $O(1)$ in $H_g + B_1+...+B_j$ (where $B_1,...,B_j$ are the bridges) can be found as well by applying the forbidden$_0$ subgraph algorithm *PT-FS* as described below on an appropriate modification of $H_g + B_1+...+B_j$. Let $p = (v_{B_0,F_0}, v_{B_1,F_1},..., v_{B_j,F_j})$ be an implication path in $K(H_g|G)$ found by Algorithm 4.2. Notice that bridges B_{i-1} and B_i are interlacing for $1 \leq i \leq j$.

Lemma E.1. There exists a face F of $I(H_g)$ and a minimal subpath $p' = (v_{B_{j'},F_{j'}}, ..., v_{B_{j''},F_{j''}})$ of p such that
(i) bridges B_{i-1} and B_i, $j´+1<i<j''$, are interlacing in F, and
(ii) bridges $B_{j'},...,B_{j''}$ cannot all be added to $I(H_g)$.

Proof: Consider the following cases (a) and (b).

(a) p is not a cycle. Suppose that p' is a subpath of p of minimum length that satisfies Condition (ii) of the lemma. We shall prove that p' contains no fixed literals except for its endpoints. As p is not a cycle and p is an implication path then either v_{B_0,F_0} and $\tilde{v}_{B_j,F_j}$ or $\tilde{v}_{B_0,F_0}$ and v_{B_j,F_j} are literals fixed to **true**. Let v_{B_0,F_0} and $\tilde{v}_{B_j,F_j}$ be the literals fixed to **true**. Suppose that p' contains a literal v_{B_i,F_i} fixed to **true** different from v_{B_0,F_0}. Then the path $(v_{B_i,F_i},..., \tilde{v}_{B_j,F_j})$ satisfies Condition (ii) of the lemma and is shorter than p'. A similar contradiction will arise in the case where p' contains a complement of a literal fixed to true different from $\tilde{v}_{B_0,F_0}$. Then p' contains no fixed literals except for its endpoints. Therefore there exists a face F such that any bridge B_i, $j'<i<j''$ is attached to F and thus bridges B_{i-1} and B_i, $j´+1<i<j''$, are interlacing in F.
(b) p is a cycle. Suppose that p contains a fixed (say fixed to **true**) literal $\mathrm{v}_{\mathrm{B_i,F_i}}$ (otherwise the lemma follows directly). Then we can transform p into an implication path p^* of K, $p^* =(v_{B_i,F_i}, v_{B_{i+1},F_{i+1}},..., v_{B_j,F_j} = \tilde{v}_{B_0,F_0}, \tilde{v}_{B_1,F_1},..., \tilde{v}_{B_i,F_i})$. As the endpoints of p^* are fixed literals we can apply the proof from Case (a) to p^*. **Q.E.D.**

According to Lemma E.1 there exists an implication path p' and a face F of $I(H_g)$ such that the bridge corresponding to each literal from p' is attached to F. We shall apply the *PT-FS* algorithm on the subgraph J of H_g to find a forbidden$_g$ subgraph *FS* for $I(H_g)$ of branchsize $O(1)$, where J consists of F plus all bridges of p'.
Case 1. Let p' be a cycle. Then bridges $B_{j'}$ and $B_{j''}$ also interlace in F. Thus J is nonplanar and the *PT-FS* algorithm yields a forbidden subgraph of $O(1)$ size.
Case 2. Let p' be not a cycle. Then its endpoints $v_{B_{j'},F_{j'}}$ and $v_{B_{j''},F_{j''}}$ will be fixed vertices. Define an edge e (corresponding to H_g) that interlaces with both $v_{B_{j'},F_{j'}}$ and $v_{B_{j''},F_{j''}}$. Define a graph $J'=J+e$ to which the proof of Case 1) can be applied. Remove e from the resulting forbidden$_g$ subgraph for $I(H_g)$ of branchsize $O(1)$.

To summarize the results of this section we state the following:

Lemma E.2.. If the genus of G is greater than g, then a forbidden$_g$ subgraph of branchsize $O(1)$ for extending $I(H_g)$ in G can be found in $|G|^{O(1)}$ time.

Appendix F. Construction of a branchsize $[F_g]^{O(g)}$ forbidden$_g$ subgraph for *G*

Let $I_1, I_2, ..., I_s$ be the finite list of all possible distinct skeletal subimbeddings $I(H_g)$ in $SI_g(F_g,T)$. In the previous sections we showed how for each $I_j=I(H_g)$ to construct a branchsize $O(g)$ forbidden$_g$ subgraph of G for extending $I(H_g)$. Define U_g to be the union of these forbidden$_g$ graphs for I_j, j = 1, ...,s. Then any possible imbedding of U_g will clearly contain a forbidden$_g$ subgraph of G. The branchsize of U_g defined in this way could be in the worst case of the same order as the number of possible skeletal subimbeddings $I(H_g)$ in $S\ I_g\ (F\ _g,T)$. By the results of Section 2.2, this number is $([F_g\ ]g)^{O(g)} \le [F_g\ ]^{O(g)}$ since $[F_g]$ is at least g. This suffices for our theoretical results. It is possible in practice, however, to significantly reduce the maximum possible branchsize of a forbidden$_g$ subgraph of G by use of an equivalence relationship on elements of U_g, as shown below.

By the results of Appendix E, the forbidden$_g$ subgraph for each I_j has branchsize $O(1)$ and so joins $O(1)$ vertices. Two subgraphs FS and FS' from U_g are *equivalent* if they satisfy the conditions:

(i) FS and FS' are homeomorphic, and

(ii) $(FS+F_g)$ and $(FS'+F_g)$ are homeomorphic.

Lemma F.1 If FS, FS' are distinct but equivalent elements of U_g, then U_g-FS' is still a forbidden subgraph of G of genus g.

Proof If FS is a forbidden$_g$ subgraph for I_j and FS' is a forbidden$_g$ subgraph for $I_{j'}$ and they are equivalent, then it is easy to show that FS is a forbidden$_g$ subgraph for $I_{j'}$. **Q.E.D.**

The number of subgraphs in U_g will be reduced to just one distinct representative for each set of equivalent subgraphs so that the resulting subgraph is still a forbidden$_g$ subgraph of G for H_g. This may be very useful in practice but does not help the theoretical bound; in the worst case the number of subgraphs in U_g remains $[F_g]^{O(g)}$. We describe the following algorithm for reducing the branchsize of U_g. Note that to acheive the theoretical bounds the call to [Fillotti, Miller, Reif,79] need not be made; the subgraph resulting at step 6 always has genus at least g+1, so we can just let g' = g+1 in line 5.

Algorithm F.1

```
Input the union $U_g$ of the forbidden$_g$ subgraphs for all possible skeletal subimbeddings $H_g$ in $SI_g\ (F_g\ ,T)$ .
1.  Initial value: $U'_g := U_g$.
2.  Pick a graph $FS$ from $U'_g$.
3.  Find the graph $FS^*$ without degree-2 vertices to which $FS$ is homeomorphic.
4.  If there exists a graph $FS'$ in $U'_g$ homeomorphic to $FS^*$ and with the same attachment vertices
        then $U'_g := U'_g$-$FS$.
Comment $U'_g\ + F_g$ is a forbidden$_g$ subgraph of $G$ with branchsize $[F_g]^{O(g)}$ and genus > $g$
5.  (optional) Using [Fillotti, Miller, Reif,79] compute the genus $g'$ of $U'_g\ + F_g$
6.  $F_{g'} := U'_g\ + F_g$
Output $F_{g'}$ , which is a biconnected subgraph of $G$ with branchsize $[F_g]^{O(g)}$ and genus $g'$ greater than $g$ .
```

Note: Since F_g is assumed to be biconnected, so is $F_{g'}$.

The time complexity of Algorithm F.1 is easily seen to be $[F_g]^{O(g)}$, if standard adjacency list data structures for the graph and its dual are used.

Lemma F.2 Given the union U_g of the forbidden$_g$ subgraphs for all possible WQPE imbeddings $I_j(H_g)$ of H_g, for all possible skeletal subimbeddings H_g in $SI_g\,(F_g,T)$, then in time $[F_g]^{O(g)}$ we can construct a genus $g' > g$ biconnected subgraph $F_{g'}$ of G of branchsize $[F_g]^{O(g)}$.

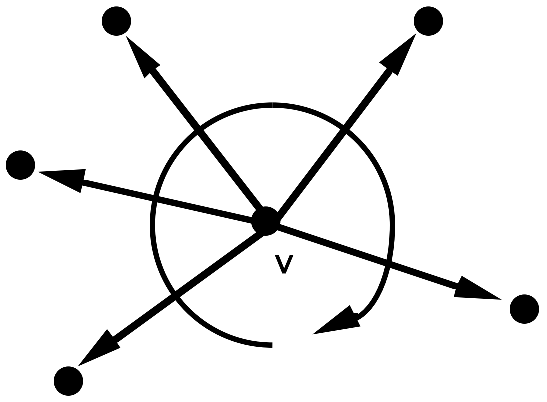

Figure 1.1

The cyclic order of directed edges around a vertex.

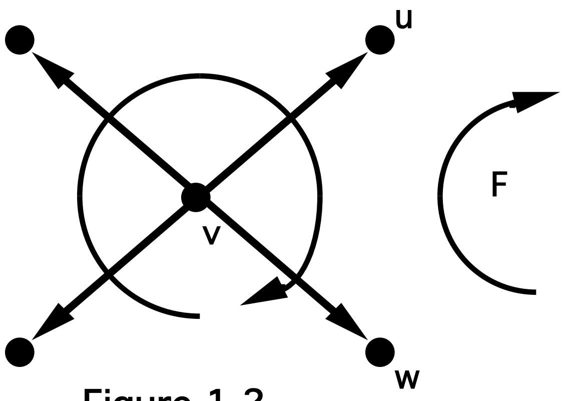

Figure 1.2

Definition of the permutation of directed edges used in defining a face.

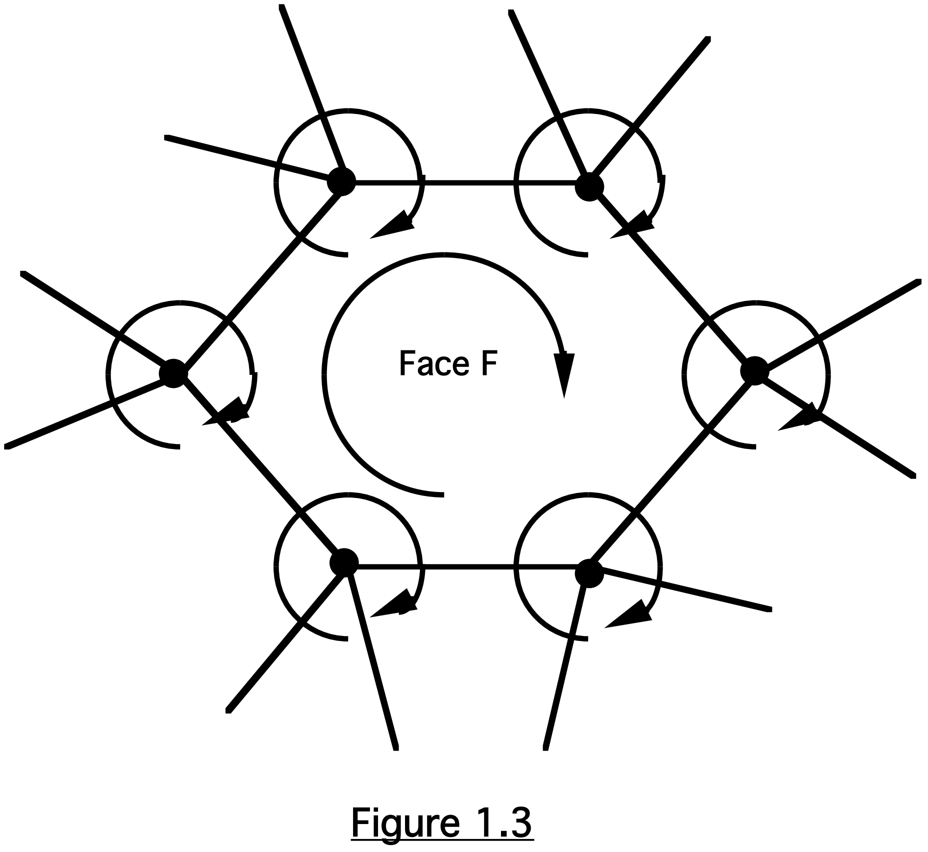

Figure 1.3

A face F defined by an orbit of the permutation of directed edges.

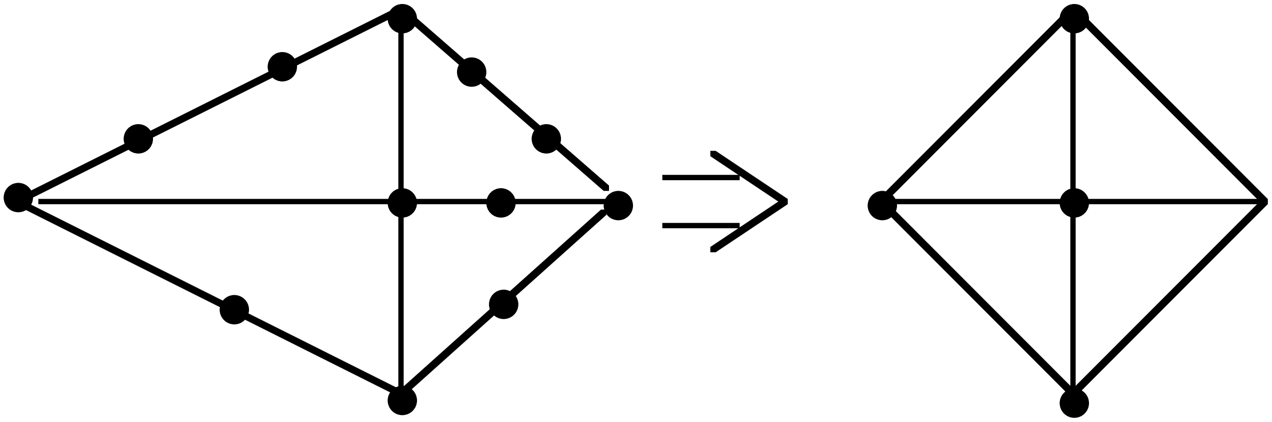

Figure 1.4

Homeomorphic contraction of a graph.

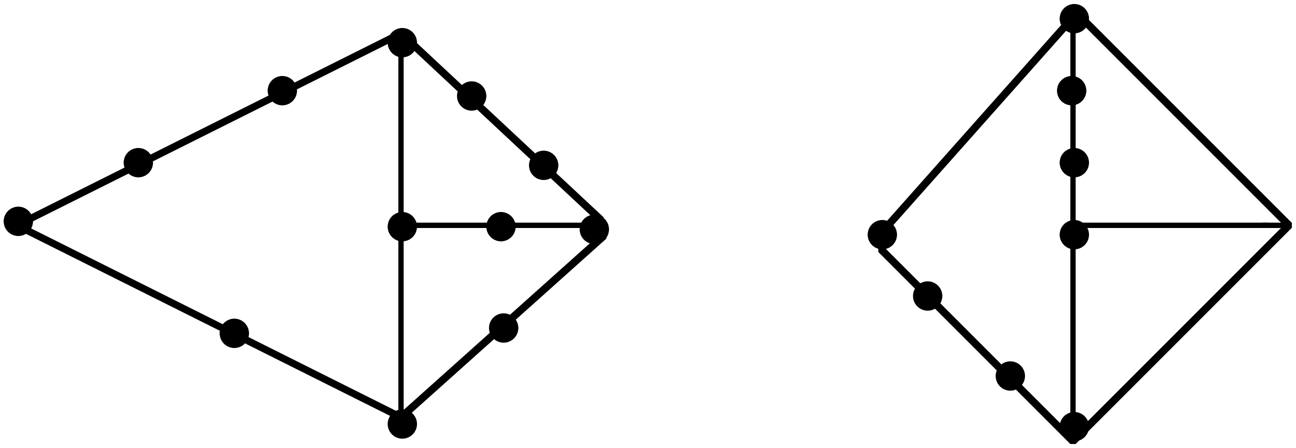

Figure 1.5

Homeomorphic graphs.

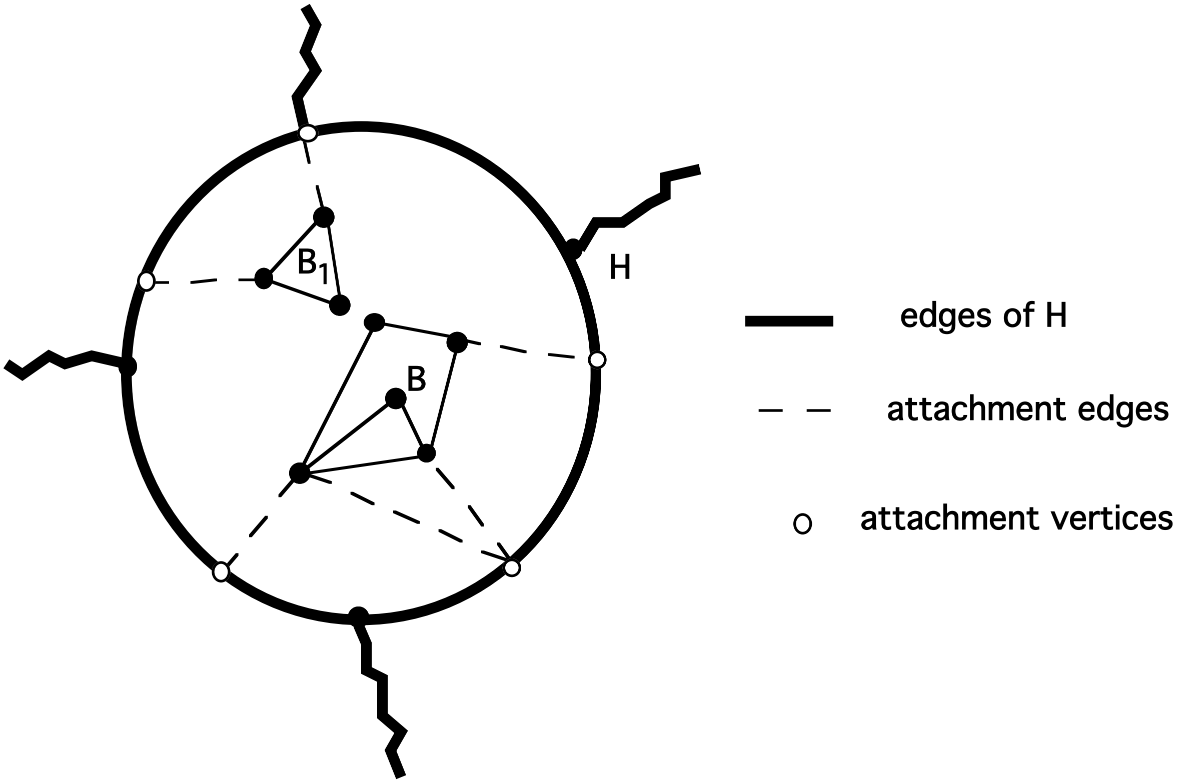

Figure 2.1. Bridges B and B_1

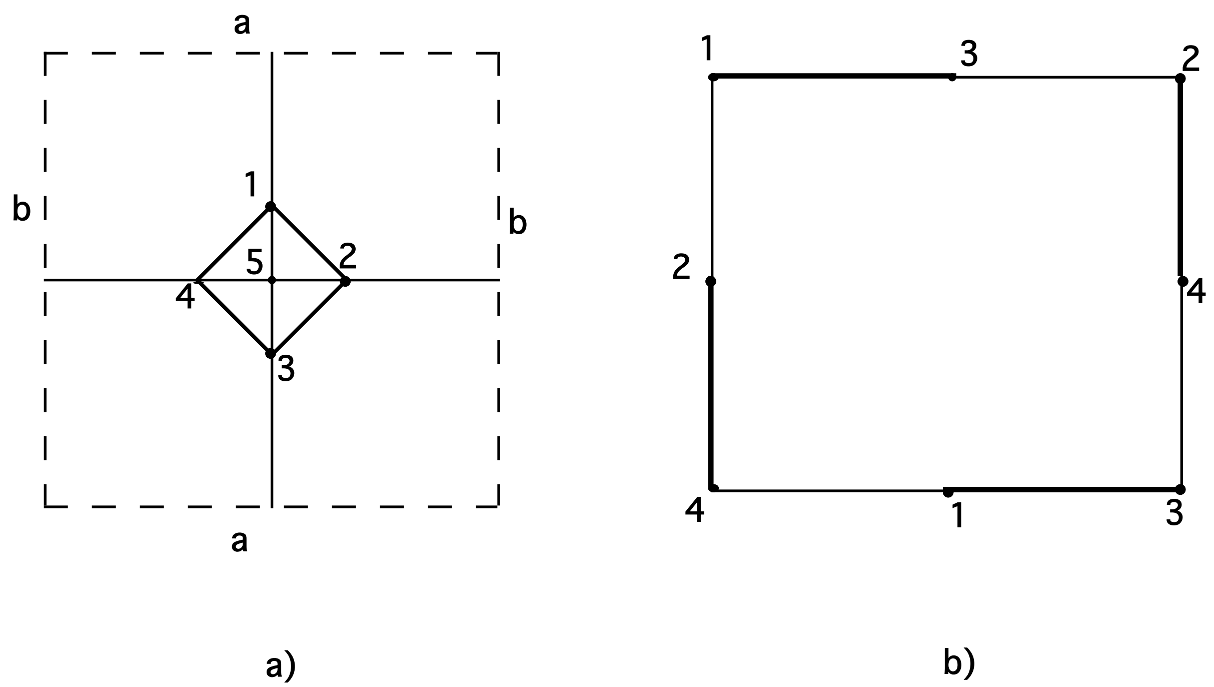

Figure 3.1. Internal vertices and edges

a) An imbedding if K_5 on the torus

b) The outer face of the imbedding in a). Bold edges and all vertices are internal.

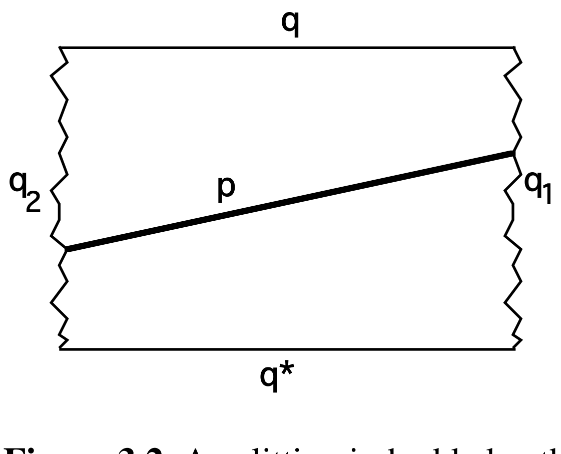

Figure 3.2. A splitting imbedded path p

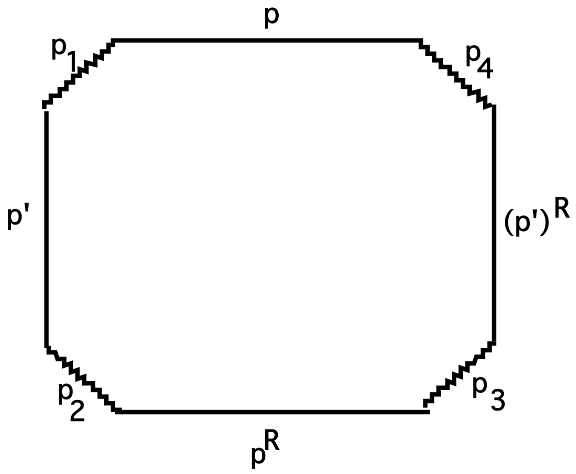

Figure 3.3. The face *F*

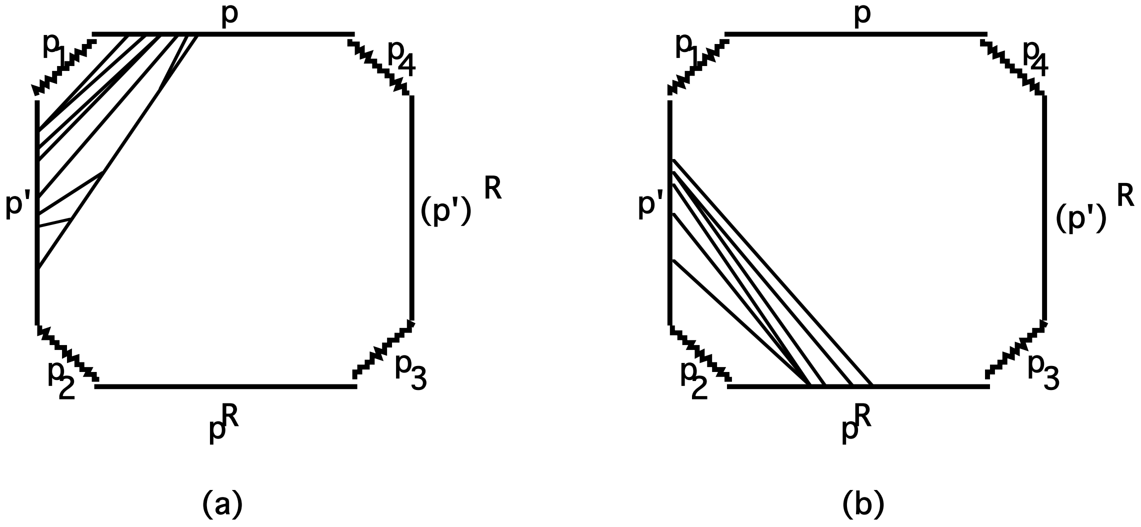

Figure 3.4. Weakly monotonic sequences: **(a)** increasing, **(b)** decreasing

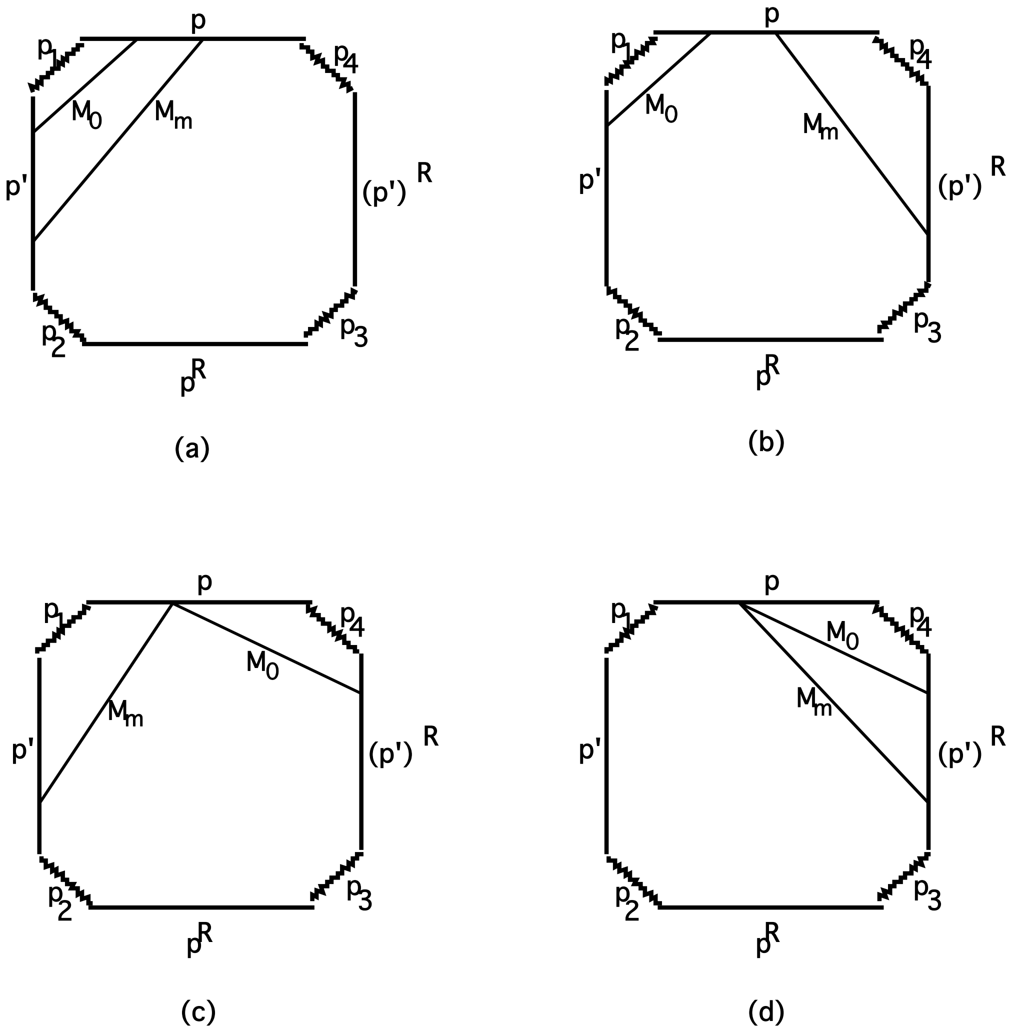

Figure 3.5. Possible imbeddings of M_0 and M_m

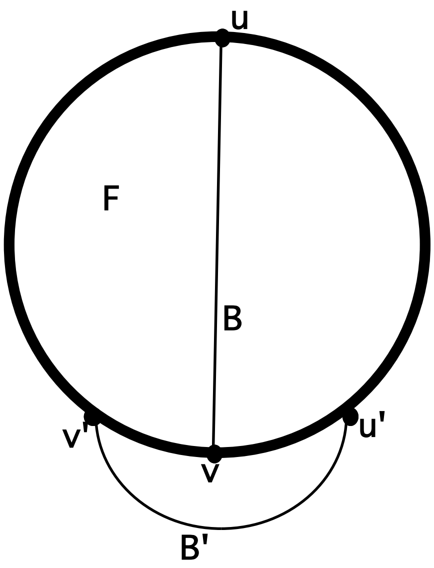

Figure 4.1. Interlacing bridges B and B'

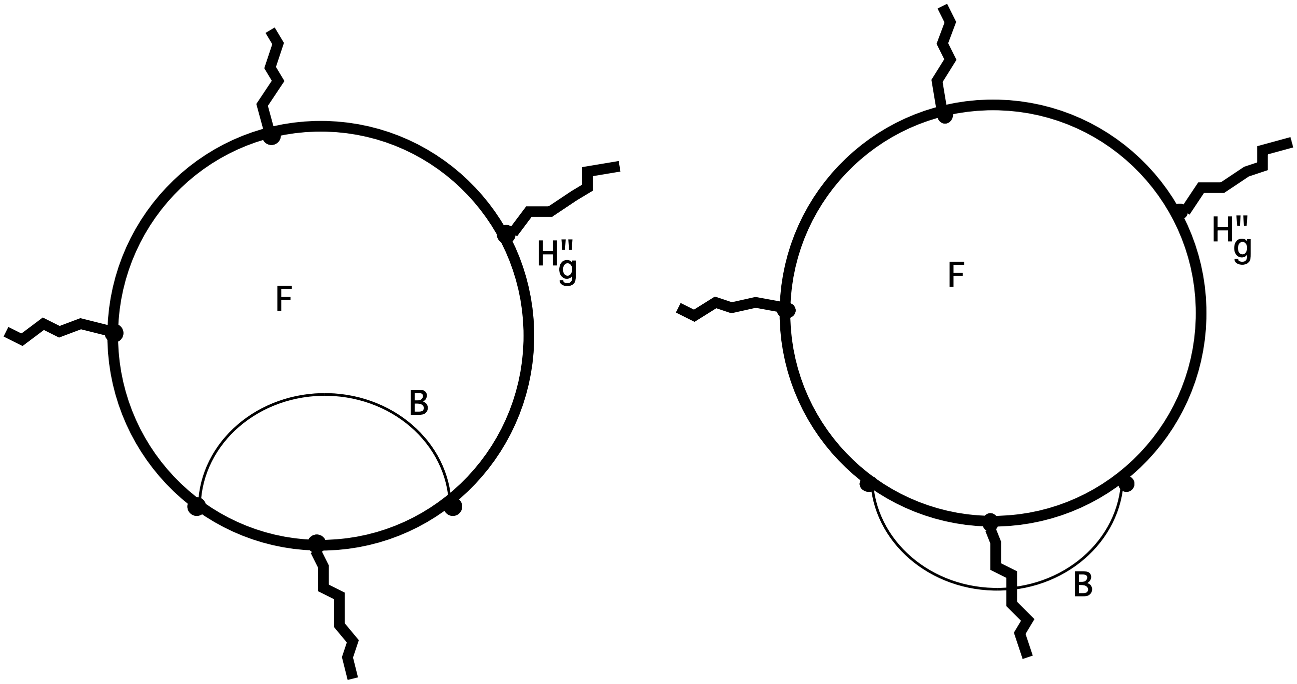

Figure 4.2. $v_{B,F}$ is fixed a) to **true**; b) to **false**

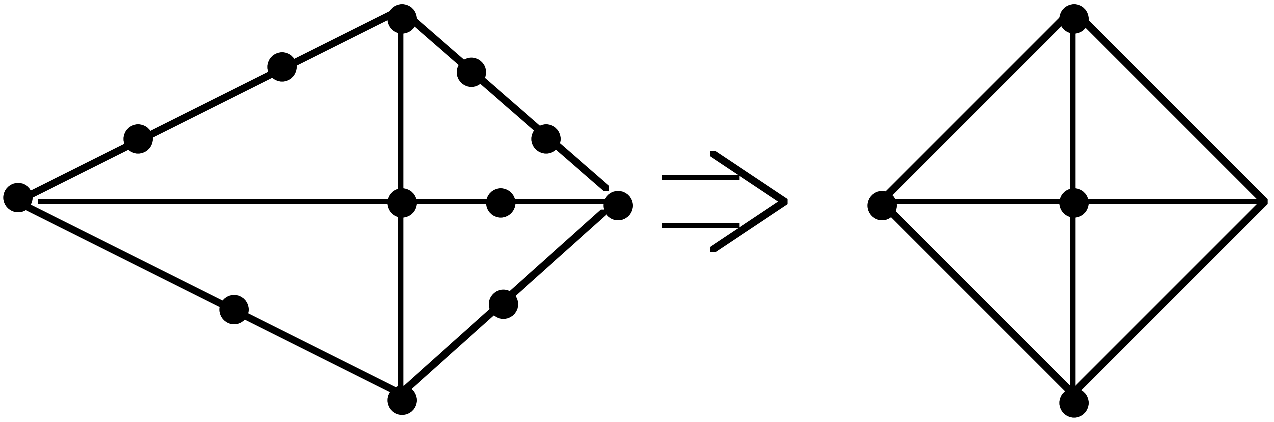

Figure 1.4

Homeomorphic contraction of a graph.

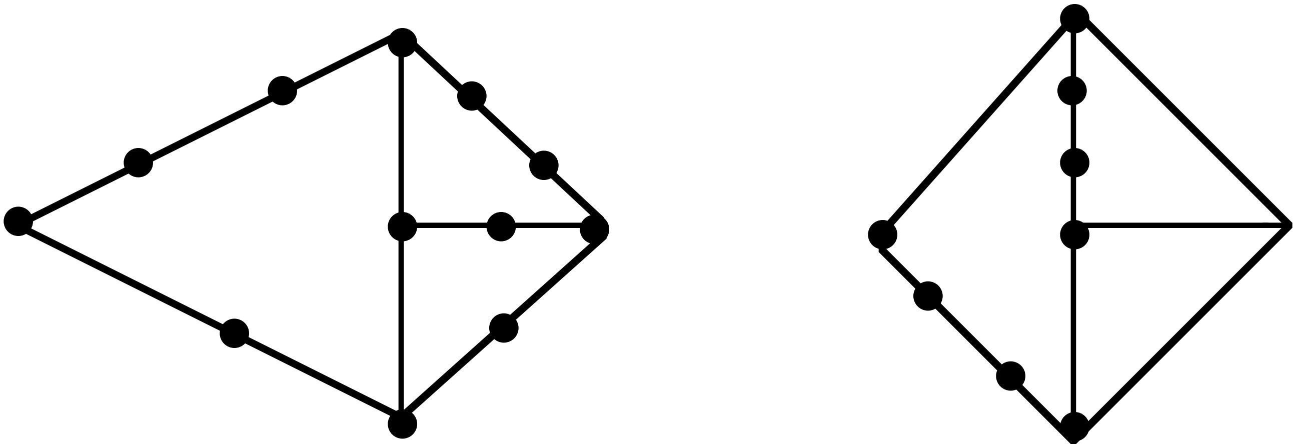

Figure 1.5

Homeomorphic graphs.

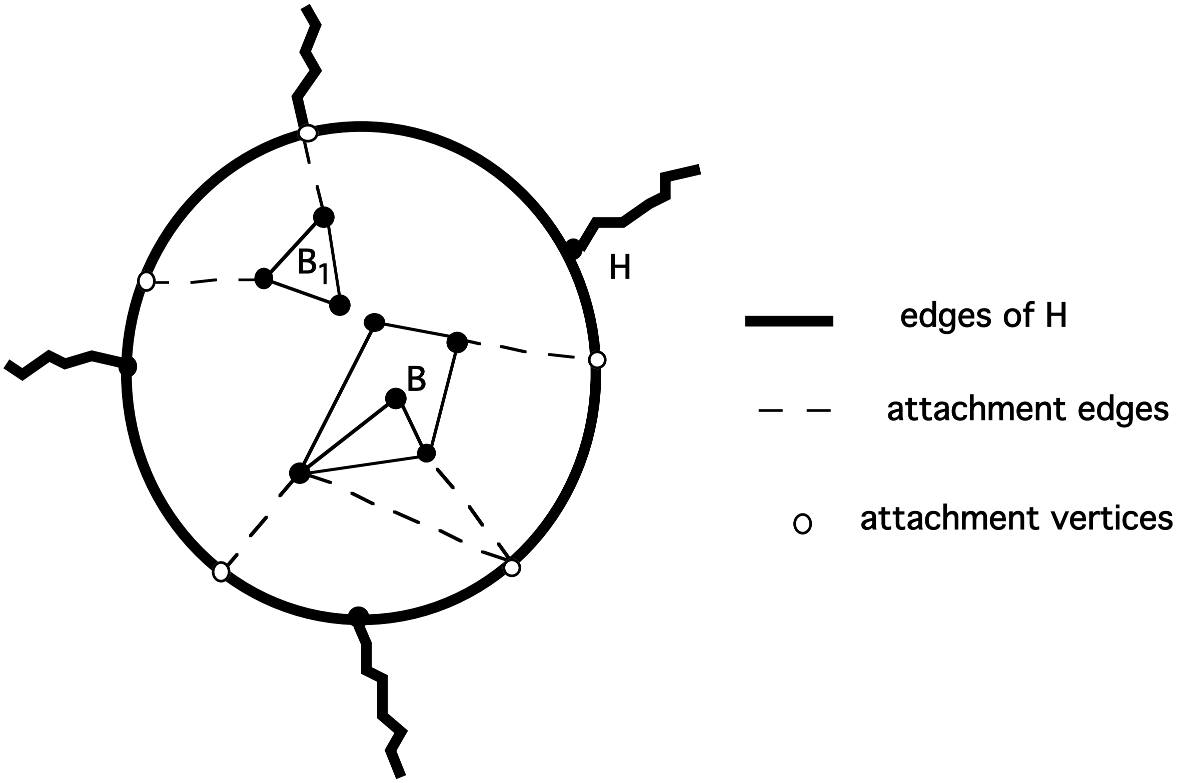

***Figure 2.1*. Bridges B and B_1**

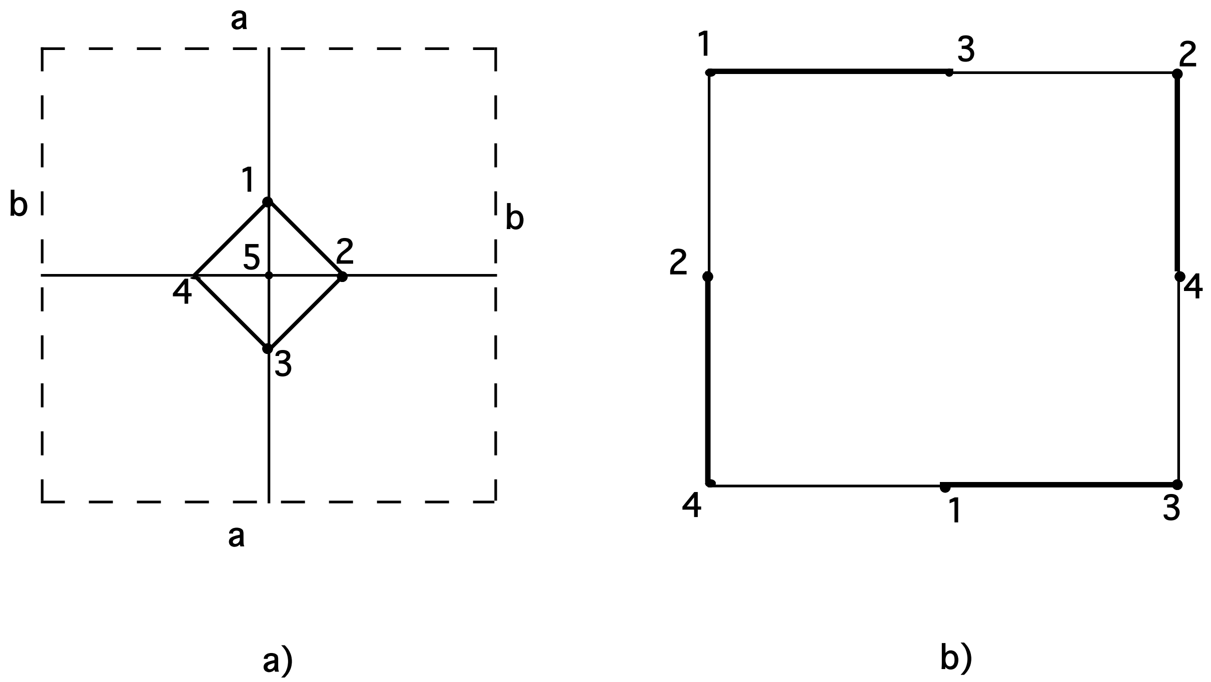

***Figure 3.1*. Internal vertices and edges**

a) An imbedding if K_5 on the torus

b) The outer face of the imbedding in a). Bold edges and all vertices are internal.

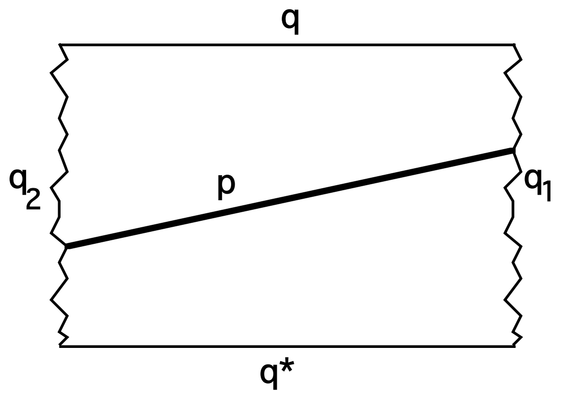

***Figure 3.2*. A splitting imbedded path p**

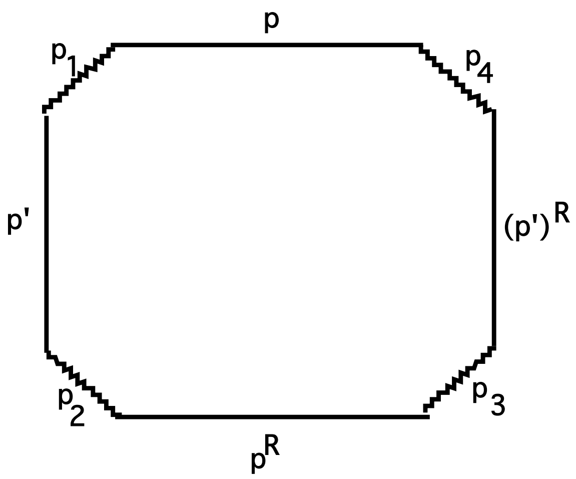

***Figure 3.3*. The face F**

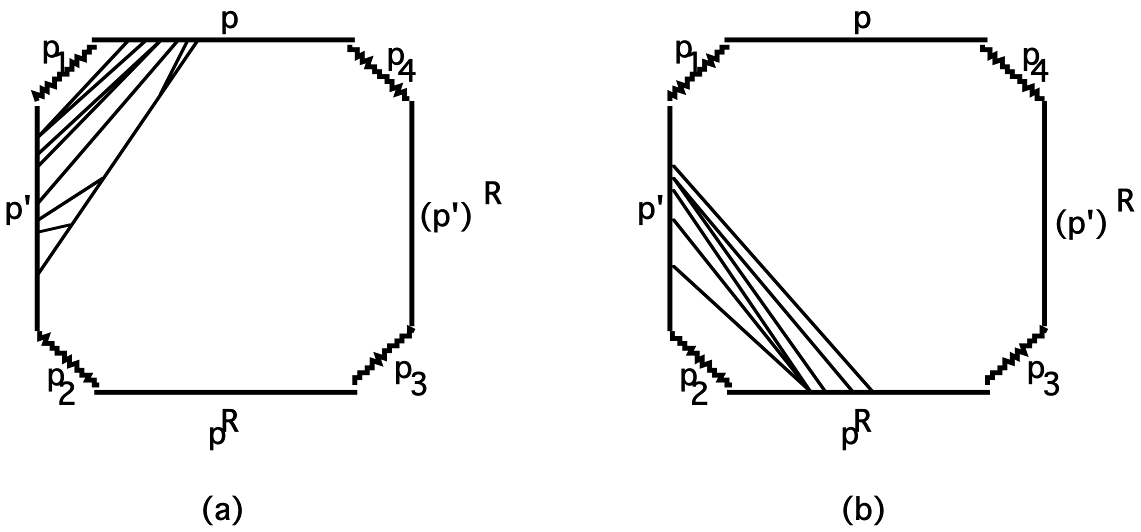

***Figure 3.4*. Weakly monotonic sequences: (a) increasing, (b) decreasing**

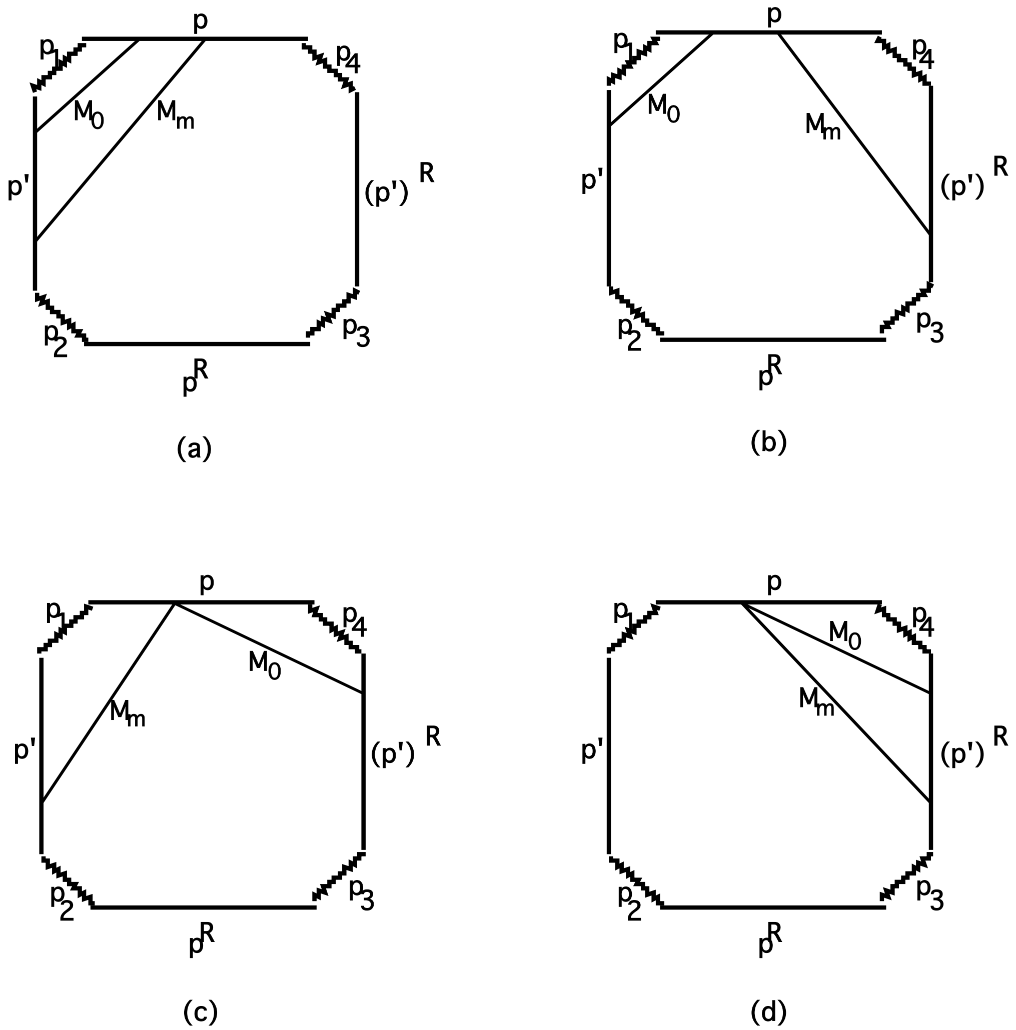

***Figure 3.5*. Possible imbeddings of M_0 and M_m**

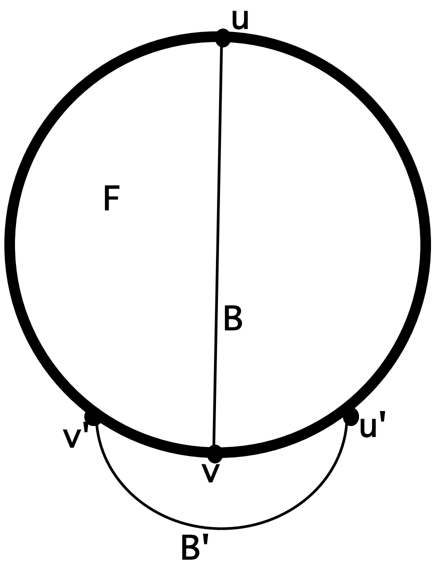

***Figure 4.1*. Interlacing bridges B and B'**

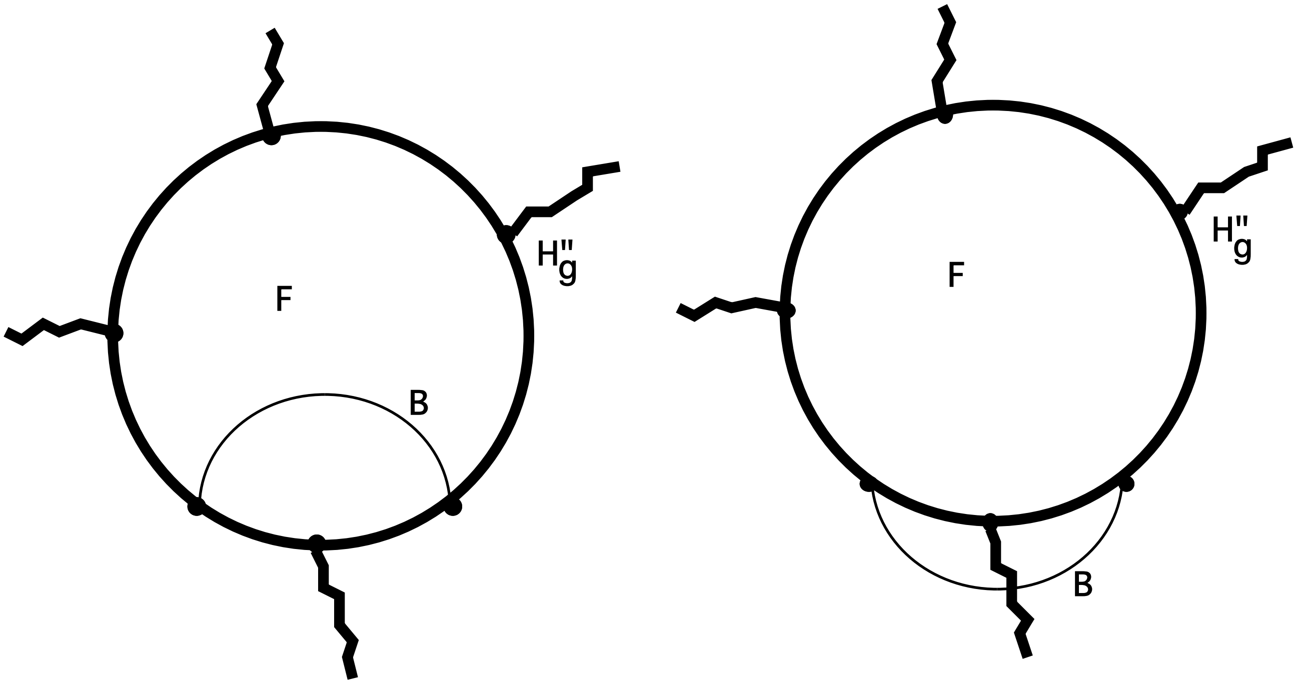

***Figure 4.2*. $v_{B,F}$ is fixed a) to true; b) to false**