HEEGAARD DOCUMENTATION

by John Berge[1]

1. INTRODUCTION

This file includes some detailed information about the mathematical underpinnings of the program and also takes a closer look at many of the program's routines.

The design decision to have the program use presentations of compact orientable 3-manifolds as its basic data, and the subsequent need for the program to be able to efficiently solve"realization" problems in order to recover Heegaard diagrams of the 3-manifolds involved, turns out to lead to a surprisingly complex program. In this file, we try to indicate what some of the problems are which the program faces, and to also indicate what the program's current strategies are for dealing with them.

Ultimately, the goal is to make understanding compact orientable 3-manifolds easier, and to make Heegaard diagrams as useful a tool in that understanding as possible. The program attempts to take a step in that direction, but it should be thought of as a provisional step, with the program itself, its approach and its routines all subject to revision and improvement. We wish to also mention that we do not believe that the road to understanding compact orientable 3-manifolds must lead through Heegaard diagrams. Rather, the idea is only to make such diagrams as useful as they can be. In this respect, it is probably worth stressing, that once the program has determined that a presentation is uniquely realizable, then the program"knows" the 3-manifold, and, in principle, additional routines could be added to the program to compute any of one's favorite 3-manifold invariants, or routines could be added to search for"interesting" features of the 3-manifold.

Finally, to facilitate making modifications, improvements, and additions to the program, we will make the source code available to those who wish to proceed further.

2. SOME USEFUL DEFINITIONS, NOTATION, AND GENERAL INFORMATION.

Def. Let P be a finite presentation on a set of g generators with n relators R_i, $1 \leq i \leq n$, such that R_i is freely and cyclically reduced for $1 \leq i \leq n$. We say P is "realizable" if there exists a handlebody H, of genus g, and a set C_i, $1 \leq i \leq n$, of n pairwise disjoint simple closed curves embedded in the boundary of H, together with a complete set of cutting disks D for H, such that when the members of C and D are appropriately labelled and oriented, C_i freely and cyclically reduces to a cyclic conjugate of R_i, for $1 \leq i \leq n$.

Remark. Note that, if D is a Heegaard diagram with genus greater than one, and D realizes a presentation P, then there are an infinite number of diagrams obtainable from D by performing Dehn twists along curves which bound disks in the underlying handlebody H, such that each of these diagrams also realizes P.

Remark. We will give an example later which shows why it is necessary to allow free and cyclic reductions in the definition of realizability unless P has minimal length.

Note: The program always assumes that the "realization" of a presentation is accomplished by

[1]Translated into LaTeX by Nathan Dunfield. Hopefully no errors were introduced in this process

adding 2-handles to the handlebody H along the set of curves C, and then any resulting 2-sphere boundary components are "capped-off" with 3-balls to yield a 3-manifold M.

Note: The program does not keep track of the orientations of Heegaard diagrams, and so does not keep track of the orientations of the corresponding 3-manifolds. In particular, homeomorphisms may be either orientation preserving or orientation reversing.

Remark. Many 3-manifolds have "inequivalent" Heegaard splittings. The program does not currently make any explicit attempt to search for such inequivalent splittings. They may arise naturally, however, as the program produces diagrams from an initial presentation by eliminating primitive relators in different ways. For example, this is essentially what occurs when the program finds distinct 1-relator presentations of the exteriors of tunnel- number-one knots, such as the $(-2,3,7)$ pretzel knot.

Remark. The program contains routines which will recognize the 3-sphere and lens spaces under certain conditions. These routines are not claimed to be algorithmic. On the other hand, we do not currently know of any examples for which these routines fail. It would, of course, be very interesting to have such examples.

Remark. The procedures which the program uses to generate new presentations are often not invertible, or at least not easily inverted. This is one of the major reasons that the program saves what eventually turn out to be superfluous presentations. These presentations are saved in case the program runs into a dead end and needs to backup to a previous stage in order to continue.

Def. Let P be a presentation, and let R be a relator which appears in P. The "algebraic length" of R is the total number of letters which appear in R after R is freely and cyclically reduced. The "algebraic length" $|P|$, of P is the sum of the algebraic lengths of all of the relators which appear in P.

Def. Let D be a Heegaard diagram. The "geometric length" $|D|$, of D is the total number of edges which appear in D.

Notation. We will often use upper-case and lower-case letters to denote generators and their inverses in presentations.

Def. A "defining relator" is a relator R, which is freely and cyclically reduced, and which has the property that there exists a generator say G, such that the total number of appearances of G and g in R is one.

Recall that this allows R to be used to express G in terms of the other generators appearing in R. Then the resulting expression can be used to eliminate all appearances of G and g from the remaining relators.

Def. A "primitive" is a relator which represents a free generator of the free group generated by the generators of the presentation.

By "deleting" or "eliminating" free generators, the program can reduce the rank of a free group and reduce the Heegaard genus of the associated manifold.

Def. A Heegaard diagram D, of a closed orientable 3-manifold M, is "pseudo-minimal" if there do not exist any automorphisms which reduce the length of either of the two dual presentations associated with D.

Def. Suppose G is a graph and V is a vertex of G. The "link" of V in G is the set of vertices of G which are joined to V by one or more edges.

Def. Suppose G is a graph. A "component" of G is a subset C, of the vertices of G, which is maximal with respect to the property that any two vertices in C can be joined by a path in G.

Def. Suppose V is a vertex of a graph G, and G has n components. If deleting V from G yields a graph G' which has more than n components, then we say V is a "cut vertex" of G.

Def. Let P be a presentation on generators $\{A,B,C,\ldots\}$, such that the relators of P are freely and cyclically reduced. (Assume P does not contain any empty relators.) If X is a generator, let x denote the inverse of X. The "Whitehead graph" of P is the graph with vertices $\{A,a,B,b,C,c,\ldots\}$, and an undirected edge joining vertex x to vertex Y for each occurrence of the pair of letters XY in the relators.

Note: It will be useful to think of the Whitehead graph of a presentation P on g generators and n freely and cyclically reduced relators, as the graph obtained in the following way:

(1) Let H be a handlebody of genus g with a complete set of cutting disks D.
(2) Orient and label the members of D so they correspond to the generators of P.
(3) Embed n pairwise disjoint simple closed curves in the interior of H so that they "represent" the freely and cyclically reduced relators of P, taking care that the total number of intersections of this set of curves with the members of D, equals the total length of the relators of P.
(4) Cut H open along the members of D to obtain a 3-ball B with $2g$ disks in its boundary.
(5) Identify the $2g$ disks in the boundary of B with the vertices of the Whitehead graph.
(6) When H is cut open along the members of D, the simple closed curves that represent the relators of P are cut into a set of arcs. Identify these arcs with the edges of the Whitehead graph.

Note that in general, there are multiple edges joining any given pair of vertices in the Whitehead graph. It will often be convenient to pass to a simpler graph which does not have any pair of its vertices joined by more than one edge.

Def. Let WG be the Whitehead graph of a presentation P. The "reduced Whitehead graph" of P, is the graph RWG obtained by replacing any sets of multiple edges, joining a pair of vertices of WG, with a single edge joining that pair of vertices.

3. THE BASIC THEORY WHICH UNDERLIES THE PROGRAM.

Most of basic theory which underlies the program is fairly simple, and has its roots in fundamental results about automorphisms of finitely generated free groups due to J.H.C. Whitehead in 1936, together with more recent results of Zieschang, which show that the algebraic results of Whitehead can be realized topologically in the case of Heegaard diagrams.

While the results of Whitehead and Zieschang form the mathematical basis for much of the program, the actual development of a program which effectively exploits their results requires more. In particular, it must be possible to find and perform certain simplifications of Heegaard diagrams, which Whitehead and Zieschang guarantee exist, in an efficient manner. This is important, because although efficiency is largely irrelevant from a mathematical point of view, it is a crucial consideration when it comes to the development of a practical program.

4. WHITEHEAD'S RESULTS ABOUT AUTOMORPHISMS OF FINITELY GENERATED FREE GROUPS.

Def. Let F be a finitely generated free group, and let G be a set of free generators for F. An automorphism ϕ of F is a "T-transformation" if ϕ permutes the generators in G, or if, for some fixed generator A in G, ϕ carries A to A, and ϕ carries each of the remaining generators X in G into one of X, AX, Xa or AXa.

Note: There are only a finite number of T-transformations in the automorphism group of F. Indeed it is not hard to see that the number of nontrivial T-transformations which are not permutations of the generators, in a free group of rank g, $g > 1$, is $g(4^{g-1} - 2)$ up to conjugacy. Note further, however, that although this number is finite, it grows exponentially in the rank of F.

Theorem. (Whitehead) Let P be a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Let F be the free group freely generated by the members of G. If there exists an automorphism π of F which reduces the length of P, then there exists a T-transformation ϕ of F which also reduces the length of P.

Def. A presentation P has "minimal length" if the relators of P are freely and cyclically reduced, and there do not exist any automorphisms of the free group F, freely generated by the generators of P, which reduce the length of P.

Def. Suppose P is a presentation which has minimal length. If ϕ is a T-transformation which transforms P into a presentation P', such that P and P' have the same length, then ϕ is a "level-T-transformation" or simply "level-transformation".

Theorem. (Whitehead) Let P and P' be finite presentations on a finite set G of g generators such that the relators of both P and P' are freely and cyclically reduced. Let F be the free group freely generated by the members of G. Suppose both P and P' have minimal length, and suppose there exists an automorphism π of F such that $\pi(P) = P'$. Then there exists a finite sequence of level-T-transformations which also transforms P into P'.

Cor. (Whitehead) Let P be a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Let F be the free group freely generated by the members of G. There is an effective procedure which finds a sequence of automorphisms of F that reduce P to minimal length.

Cor. (Whitehead) Let P be a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Suppose P has minimal length. Let F be the free group freely generated by the members of G. Then there exists an effective procedure to generate and enumerate the set of all minimal length presentations into which P can be transformed by automorphisms of F.

5. SOME RESULTS OF ZIESCHANG WHICH RELATE WHITEHEAD'S RESULTS TO HEEGAARD DIAGRAMS.

Def. Let D be a Heegaard diagram, and let H be the underlying handlebody. Suppose H has genus n and that $d_i, 1 \leq i \leq n$, is a complete set of cutting disks for H. We assume that the intersections of the curves of the diagram with the cutting disks are minimal up to isotopy in the boundary of H. Suppose d is a nonseparating disk in H such that d is disjoint from d_i for $1 \leq i \leq n$. If it is possible to replace one of the d_i(s) with d so as to obtain a complete set of cutting disks for H which has fewer intersections with the attaching curves of the 2-handles of D, then we say that a "geometric-T-transformation" has occurred.

Theorem. (Zieschang) Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. If there exists a T-transformation which reduces $|P|$, then there exists a geometric-T- transformation which reduces $|D|$.

Cor. (Zieschang) Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. Suppose that P' is any minimal length presentation obtained from P via a finite sequence of T-transformations, then there exists a finite sequence of geometric T-transformations which transform D into a Heegaard diagram D' such that $|D'| = |P'|$ and D' realizes P'.

Def. Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. Suppose P has minimal length, $|D| = |P|$, and Σ is a geometric-T-transformation which carries D into D' such that $|D'| = |D|$. Then we say Σ is a "level-geometric-T-transformation".

Cor. (Zieschang) Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. Suppose P has minimal length, $|D| = |P|$, and P' is a presentation obtained from P by a level-T-transformation. Then there exists a level-geometric-T-transformation which carries D into D' such that $|D'| = |P'|$, and D' realizes P'.

Remark. Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. Suppose that ϕ is a T-transformation which reduces the length of P. Zieschang's results guarantee that there is a geometric-T-transformation Σ which reduces the geometric length of D. Now Σ induces some T-transformation of P, but it is not necessarily the case that $\Sigma = \phi$. Indeed, there may be no geometric-T-transformation which induces ϕ. It is also not hard to produce examples where any geometric-T-transformation which reduces $|D|$, must actually increase $|P|$.

6. REFERENCES

(1) J.H.C. Whitehead, On equivalent sets of elements in a free group, Ann. of Math. (2)37(1936), 782-800.

(2) Zieschang, On simple systems of paths on complete pretzels. Amer.Math.Soc.Transl. (2) Vol.92,1970, 127-137

7. HOW THE PROGRAM INSURES THAT THE DIAGRAMS IT PRODUCES ARE ALL HEEGAARD DIAGRAMS OF THE SAME 3-MANIFOLD.

It is obviously important that the Heegaard diagrams which the program produces all be Heegaard diagrams of the same 3-manifold. In order to guarantee that this is the case, new presentations must be obtained from previous presentations only in certain ways. Here we describe how the program does this, and we indicate when it is possible to recover a Heegaard diagram of the original 3-manifold from the transformed presentation.

Suppose P is a realizable presentation of a 3-manifold M. The program is allowed to produce a new presentation P' from P in one of following four ways, provided the stated conditions hold.

(1) If P does not have minimal length, let P' be any presentation obtained from P by reducing P to minimal length.

(2) If P has minimal length, let P' be any presentation obtained from P by a sequence of level-transformations.

(3) If P is a minimal length balanced presentation, P has a unique realization as a Heegaard diagram D, where $|D| = |P|$, and D is a Heegaard diagram of a closed manifold, let $\hat{P}$ be the dual presentation corresponding to D. Then let P' be any presentation obtained from $\hat{P}$ by reducing $\hat{P}$ to minimal length.

(4) If P is a minimal length presentation, P has a unique realization as a Heegaard diagram D, where $|D| = |P|$, and R is a relator of P, let $\hat{P}$ be a presentation obtained from P by replacing R with a new relator R', where R' is a geometrically realizable bandsum of R and another relator R" of P. Then let P' be any presentation obtained from $\hat{P}$ by reducing $\hat{P}$ to minimal length.

The next lemma follows from the preceding results of Whitehead and Zieschang.

Lemma 1. Suppose P is a realizable presentation of a 3-manifold M. Suppose that P'' is a presentation obtained from P by reducing P to minimal length or P has minimal length and P'' is obtained from P via a sequence of level-transformations. Then M has a Heegaard diagram D'' such that D'' realizes P'' and $|D''| = |P''|$.

Lemma 1. Suppose P is a realizable presentation of a 3-manifold M. Suppose that P' is a presentation obtained from P by reducing P to minimal length or P has minimal length and P' is obtained from P via a sequence of level-transformations. Suppose furthermore, that there is a unique Heegaard diagram D', with $|D'| = |P'|$, which realizes P'. Then D' is a Heegaard diagram of M.

Proof. This is an immediate consequence of the existence provided by Lemma 1) and the uniqueness provided by Lemma 2).

Applying these lemmas to the four situations above yields the following theorem.

Theorem. Suppose P is a realizable presentation of a 3-manifold M. Suppose that P' is a minimal length presentation obtained from P in one of the four ways listed above, and suppose there is a unique Heegaard diagram D', with $|D'| = |P'|$, which realizes P'. Then D' is a Heegaard diagram of M.

Remark. The program's manipulations of presentations are based largely on this Theorem. In particular, this means that the program places a premium on obtaining minimal length presentations whose realizations by Heegaard diagrams, with geometric length equal to the algebraic length of the presentation, are unique.

Remark. Suppose P is a minimal length presentation and there are several distinct Heegaard diagrams, with geometric length equal to $|P|$, which realize P. Then one of the diagrams which realizes P is a diagram of M. However, determining which realization is "correct" requires additional information. At this point, we have not settled on an efficient scheme which the program can use to do this.

8. AN EFFICIENT METHOD FOR FINDING T-TRANSFORMATIONS THAT REDUCE THE LENGTH OF A PRESENTATION.

As has been alluded to, a practical program needs an efficient method for locating and performing T-transformations which reduce the length of a presentation. We describe such a procedure below. The fact that such an efficient procedure exists does not seem to be well-known. We also wish to point out that the procedure we will describe can be applied to any presentation, and so it may well have other applications elsewhere. For these reasons, we will give a fairly detailed and complete description of the procedure.

The key to finding T-transformations which will reduce the length of a given presentation P efficiently, is to look at the problem of finding such a T- transformation as one of finding a minimal cut set of edges in the Whitehead graph of P, where the cut set of edges separates the vertices corresponding to the "operating" generator and its inverse from each other. This reduces the problem of finding a T-transformation that will reduce the length of P to a well-understood problem in graph theory of finding a maximal flow in a network. There are extremely efficient methods known for solving this type of problem.

Here are some of the details. Suppose P is a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Let F be the free group freely generated by the members of G. Let WG be the Whitehead graph of P, and let RWG be the reduced Whitehead graph of P. Let A be a generator in G, and let a be its inverse. Suppose that ϕ is a

T-transformation which is not a permutation of the generators in G. Then if is X is a generator in G, ϕ acts on X, (up to conjugation by a), in one of the following four ways: $X \to AX$, $X \to Xa$, $X \to AXa$ or $X \to X$.

Lemma. There is a one-to-one correspondence between T-transformations in which the generator A acts as a multiplier on the remaining generators of G, (up to conjugacy) and partitions of the vertices of the Whitehead graph of G into two disjoint subsets, SA, Sa, where A lies in SA, and a lies in Sa.

Proof. Define a correspondence of the required type by the following rule:

(1) A lies in SA, and a lies in Sa.
(2) $\phi(X) = AX$ if and only if X lies in SA, and x lies in Sa.
(3) $\phi(X) = Xa$ if and only if X lies in Sa, and x lies in SA.
(4) $\phi(X) = AXa$ if and only if both X and x lie in SA.
(5) $\phi(X) = X$ if and only if both X and x lie in Sa.

Now, consider the Whitehead graph WG of P as being embedded in a 3-ball B with $2g$ disjoint disks on ∂B, which can be identified in pairs to yield a handlebody H of genus g, as described above. Let SA and Sa be the partition of the vertices of WG determined by ϕ using the rule given by the preceding lemma. Observe that we may consider B as being divided into two hemispheres N and S, by an equatorial disk D, such that all of the vertices of SA lie in N, while all of the vertices of Sa lie in S. We also suppose that the arcs which represent the edges of WG are embedded in B so that they intersect D minimally, up to homotopy in B, keeping their endpoints fixed.

Choose a point $b \in S$, to serve as the basepoint of the fundamental group of H. Then, for each pair of vertices $\{\mathrm{X}, x\}$ of WG, choose an oriented path which leaves b, goes to the disk on ∂B representing X, and returns to b from the disk on ∂B which represents x. Suppose these paths are chosen so that they intersect D minimally. Clearly, these paths form a set of generators for $\pi_1(H)$, which we may view as representing the current set of generators.

Next, observe that since the disk A lies in N, and the disk a lies in S, we can replace the cutting disk corresponding to the generator$A$, with the disk D. Suppose this is done. The result is a new set of cutting disks for H, with the change of cutting disks inducing an automorphism μ of $\pi_1(H)$, and we see that if D is relabelled as A, then $\mu = \phi$.

Observe that the number of times generator A and its inverse a appear in P, in this new basis for $\pi_1(H)$, is completely determined by the number of edges of WG which join vertices in SA to vertices in Sa, while the number of appearances of the remaining generators in P is unchanged. Observe also, that because WG corresponds to P, and P is freely and cyclically reduced, all intersections of the simple closed curves in H, which represent relators of P, are essential intersections with this new set of cutting disks for H.

The following definitions will now be useful.

Def. If X is a vertex of a graph G, let $V(X)$ denote the number of edges of G meeting X.

Def. Let G be a graph, X and Y vertices of G, and let C be a subset of the edges of G. Let G' be the graph obtained by deleting the edges in C from G. We say that "C is a cut-set of edges of G separating X and Y", if X and Y lie in different connected components of G'.

Def. Let C be a cut-set of edges of a graph G. Define the "capacity of C" to be the number of edges which C contains.

Def. Let G be a graph, X and Y vertices of G, and let C be a cut-set of edges of G separating X and Y. We say C is a "minimal cut-set of edges separating X and Y", if no subset D, of the edges of G, such that D contains fewer edges of G than C contains, separates X and Y.

Note that a cut-set of edges separating X and Y may be empty.

We can now prove the main theorem of this section.

Theorem. Suppose P is a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Let F be the free group freely generated by the members of G. Let WG be the Whitehead graph of P. Let A be a generator in G, and let a be the inverse of A. There exists a T-transformation which reduces the number of appearances of A and its inverse in P, if and only if there exists a cut-set C of edges of the Whitehead graph WG, such that C separates A and its inverse a, and the capacity of C is less than $V(A)$.

Proof. The preceding discussion shows that if there is a T-transformation which reduces the number of appearances of the generator A together with its inverse in P, then there is a cut-set of edges C of WG such that C separates vertex A from vertex a, and the capacity of C is less than $V(A)$.

Conversely, suppose that C is a cut-set of edges of WG which separates vertex A from vertex a in WG, and suppose the capacity of C is less than $V(A)$. Let G' be the graph obtained when the edges of C are deleted from WG. Let SA be the set of vertices of the component of G' containing vertex A, and let Sa be the remaining vertices of G'. Define a T-transformation ϕ, in which A acts on the remaining generators in G, by using the rule given in the preceding lemma above. Now consider WG to be embedded in a 3-ball B, as in the preceding discussion. We see that ϕ reduces the number of appearances of A and its inverse, in P.

For the next theorem, we will assume that the Whitehead graph of P is connected. We do this only to simplify the statement of the theorem. The appropriate statement, in the case where the Whitehead graph of P is not connected, is not hard to find.

Theorem. Suppose P is a finite presentation on a finite set G of g generators such that the relators of P are freely and cyclically reduced. Let F be the free group freely generated by the members of G. Suppose that P has minimal length. And suppose the Whitehead graph WG of P is connected. Let A be a generator in G, and let a be the inverse of A. There exists a non-trivial level-T-transformation in which A acts on the remaining generators of G, if and only if there is a cut-set C of edges of the Whitehead graph WG such that C separates A and its inverse a, the capacity of C equals $V(A)$, and neither component, of the graph obtained by deleting the edges of C from WG, consists of only a single vertex.

Proof. Observe that the graph G' obtained by deleting the edges of C from WG, has only two components. Let SA be the set of vertices of the component of G' containing vertex A, and let Sa be the set of remaining vertices of G'. Use SA and Sa to define a T-transformation, and proceed as before.

Up to this point, we have only shown that there is a strong connection between T-transformations, and cut-sets of edges. In particular, we have seen that if we wish to reduce the number of appearances of say, generator A and its inverse in a presentation P, then we should look for a cut-set C, of edges of the Whitehead graph WG of P, such that C separates vertex A from vertex a in WG, and C has capacity less than $V(A)$. But, the number of appearances of generator A and its inverse in the transformed presentation is equal to the capacity of C. This suggests we should be "greedy" and look for minimal cut-sets of edges separating vertex A from vertex a. This turns out to be the right thing to do, because there are very efficient ways to find such minimal cut-sets.

9. FINDING T-TRANSFORMATIONS EFFICIENTLY BY FINDING MAXIMAL FLOWS IN NETWORKS.

The efficient way to find minimal cut-sets and hence T-transformations, is to dualize the problem and to look for a "maximal flow" in the network given by the Whitehead graph of P. This problem has been intensively studied in many areas, including computer science, graph theory, and operations research, so it should not be hard to find numerous references. Since this is the case, we will only include a sketch of the basic ideas.

Suppose we wish to find a minimal cut-set of edges separating vertex A from vertex a in the Whitehead graph of some presentation P. Consider the vertex A to be the "source" vertex and vertex a to be the "sink" vertex of a network given by WG. Now the idea is to find a maximal collection of directed paths in WG, such that each path runs from the source to the sink, and such that no two paths have any edges in common and no path traverses a given edge more than once. Let F be such a maximal collection. Then the duality Theorem of Ford and Fulkerson asserts that the number of paths in F equals the capacity of a minimal cut-set of edges separating source and sink.

Suppose that F is given, then there is no directed path, which joins source to sink, and which does not have directed edges, with identical directions, in common with paths in F. Let SA be the set of vertices of WG, which are accessible from the source vertex A by directed paths which share no directed edges, with identical directions, with members of F. Let Sa be the set of remaining vertices of WG. Then the edges which join vertices in SA to Sa form a minimal cut-set of edges of WG separating vertex A from vertex a.

As far as efficiency goes, we mention only that there are ways to find "maximal flows" and, by duality, minimal cut-sets in a graph with V vertices and E edges in time which is at worst $O(VE(\log V))$.

Remark. Note that these same ideas can be applied, essentially without change, to find T-transformations of sets of "true" words in a free group. In this case, the Whitehead graph has an additional vertex b, which serves as a basepoint for all of the paths involved, but aside from this, there is no basic difference.

10. A PROOF OF ZIESCHANG'S THEOREM

At this point, we can give a proof of Zieschang's Theorem which uses the idea of looking for minimal cut-sets of the Whitehead graph of a presentation. The proof will show that, under most circumstances, the T-transformations, which the program uses to simplify presentations, are actually geometric.

Theorem. (Zieschang) Suppose D is a Heegaard diagram with underlying handlebody H and P is the presentation of the fundamental group associated with the handlebody H. If there exists a T-transformation which reduces the length of P, then there exists a geometric- T-transformation which reduces the geometric length of D.

Proof. We assume P is freely and cyclically reduced. There are two cases: 1) $|D| > |P|$, 2) $|D| = |P|$. Suppose $|D| > |P|$. Then there exists an edge E of D and a vertex X of D such that both ends of E lie on ∂X. Let N be a regular neighborhood in D of X union E. Then ∂N consists of two simple closed curves, one of which, say C, separates X from its inverse x, in D. Then replacing the cutting disk X with a disk in H, whose boundary is C, is a geometric-T-transformation which reduces $|D|$. So we may suppose that $|D| = |P|$. Then no edge of D has both of its endpoints on the same vertex of D. Thus D is essentially an embedding of the Whitehead graph WG of P, in the plane.

Now suppose that ϕ is a T-transformation in which generator A acts on the remaining generators of P, $|\phi(P)| \leq |P|$, and that no other T-transformation, in which A acts on the remaining generators of P, reduces $|P|$ more than ϕ does. Let SA, Sa be the partition of the vertices of WG induced by ϕ using the correspondence described in the lemma above. Let C be the cut-set of edges of WG consisting of those edges of WG which join vertices in SA to vertices in Sa. Then C is a minimal cut-set of edges of WG separating vertex A and vertex a.

Now consider cutting the set of edges of D which correspond to the edges in C. Note that the minimality of C implies that the number of components of D increases by at most one. Let CA be the component of $D - C$, which contains vertex A. Let N be a regular neighborhood in D of CA. Then N is a connected planar surface, and the closure of each complementary domain of N is a disk. Let Da be the disk which contains vertex a in its interior. Observe that minimality of C implies that the only edges of WG which intersect $\partial N \cap \partial Da$. Replace ∂A with ∂Da. Note this is a geometric-T-transformation.

Then we have the following corollary.

Cor. Suppose D is a Heegaard diagram with underlying handlebody H, P is the freely and cyclically reduced presentation of the fundamental group associated with the handlebody H, and $|D| = |P|$. Suppose that ϕ is a T-transformation in which generator A acts on the remaining generators of P, $|\phi(P)| \leq |P|$, and that no other T-transformation, in which A acts on the remaining generators of P, reduces $|P|$ more than ϕ does. If D is connected, there exists a geometric-T-transformation Σ which induces ϕ. If D is not connected, there exists a geometric-T-transformation Σ, such that Σ(P) = ϕ(P).

Proof. If D is connected, all of the vertices of WG except those in CA, lie in the interior of Da. Thus Σ induces the same partition of the vertices of WG as does ϕ, and so Σ induces ϕ. If D is not connected, there may be vertices of WG which lie in other components of the complement of N. However, the action of a T-transformation on P is completely determined by the set C of edges which join the vertices in the set SA to vertices in the set Sa. This set is identical in both cases. Hence, even though we need not have Σ = ϕ, it is still the case that Σ(P) = ϕ(P).

So we see that, in most cases of interest, e.g. when $|D| = |P|$ and the Whitehead graph of P is connected, the T-transformations, which the program uses to modify presentations, are geometric.

11. ATTEMPTING TO RECOVER A HEEGAARD DIAGRAM FROM A MINIMAL LENGTH PRESENTATION.

Suppose P is a minimal length presentation which is realizable and the program wishes to produce a Heegaard diagram D which realizes P. Producing D involves several steps. The first step is motivated by the following result which follows from the results of Whitehead and Zieschang.

Lemma. Suppose P is a minimal length presentation which is realizable and WG is the Whitehead graph of P. Then WG is planar.

Proof. By the results of Whitehead and Zieschang above, there exists a Heegaard diagram D which realizes P such that $|D| = |P|$. If we consider D as a graph by identifying the disks of D to points, then the resulting graph is an embedding of the Whitehead graph WG of P in the plane.

Remark. This lemma is false without the stipulation that P has minimal length. That is, there exist realizable presentations which do not have minimal length, and whose Whitehead graphs are not planar. We will give an example later.

Now the program is essentially in the position of attempting to recover D from P, so the program's first step is to obtain the Whitehead graph WG of P, and then to pass to the reduced Whitehead graph RWG of P. It follows from the preceding lemma that WG and hence RWG must by planar. The next step is to find an embedding of RWG in the plane. Suppose for the moment that RWG has a unique embedding in the plane. Then the embedding of RWG can be extended to an embedding of WG in the plane. This extension of the embedding of RWG to an embedding of WG is unique. (See the section titled THE DEFINITION OF "PAIRS OF SEPARATING VERTICES" for an explanation as to why this is the case.) If we then replace the vertices of WG with small disks, we have recovered D except for information about how pairs of disks corresponding to pairs of inverse vertices of WG should be identified. The last step is to determine how these pairs of inverse disks should be identified. The basic procedure, which the program uses to do this, is described a following section.

We now briefly consider the situation where RWG does not have a unique embedding in the plane. Suppose P' is a presentation obtained from P via a sequence of level- transformations. Then P' is realizable, and there exists a Heegaard diagram D' which realizes P' with $|D'| = |P'|$. If the RWG of P' embeds uniquely in the plane, the program will try and recover the diagram D' instead of D. Generally, D' serves as well as D, so not having D itself is no great loss. It is a fortunate situation from the program's point of view, that if the reduced Whitehead graph of a presentation does not embed uniquely in the plane, then there exist level-transformations. It is also a fortunate situation from the program's point of view, that when this situation occurs, there is generally a sequence of level-transformations which leads to a presentation whose reduced Whitehead graph embeds uniquely in the plane. Since level-transformations seem to be fairly effective in this situation, the program has made the use of level- transformations a major part of the tactics which it uses in order to find a diagram that realizes a given presentation.

However, the exact set of circumstances under which it is possible to pass via level-transformations from a presentation whose reduced Whitehead graph does not embed uniquely in the plane, to a presentation whose reduced Whitehead graph does embed uniquely, are not completely clear. This question is discussed in more detail below.

12. EXAMPLE: A PRESENTATION P, WITH THE PROPERTY THAT ANY REALIZATION OF P HAS GEOMETRIC LENGTH GREATER THAN $|P|$.

Here is the example promised above. We like this example because it helps to justify the effort the program makes to reduce presentations to minimal length, before it attempts to determine whether they are realizable.

Let P be the following presentation whose relators are freely and cyclically reduced.

R1: AAAAABBAACCC
R2: AAABBBCCCCC
R3: AABDCCBD

If you obtain the reduced Whitehead graph RWG, of P, you will see that RWG is not planar. This shows that if there exists a Heegaard diagram D, which realizes P, then $|D| > |P|$.

If P had minimal length, the fact that the reduced Whitehead graph of P is not planar would imply that P is not realizable. However, P does not have minimal length, since the automorphism $D \rightarrow bD$ reduces the length of P by 2. Applying this automorphism yields the following presentation P':

R1: AAAAABBAACCC
R2: AAABBBCCCCC
R3: AADCCD

Now you can easily verify that P' has minimal length and P' is realizable, which, of course, implies that P is realizable.

13. THE STRATEGY WHICH THE PROGRAM USES, WHEN IT ATTEMPTS TO DETERMINE WHETHER A PRESENTATION IS REALIZABLE.

Before discussing the procedure which the program follows when it attempts to determine whether a presentation is realizable, it is important to point out that determining whether a presentation is realizable and finding all possible realizations if they exist, is essentially solvable. This is another of Zieschang results. Some care is necessary here because of the fact that one can obtain many different diagrams that realize a presentation by performing Dehn twists about curves that bound disks in the underlying handlebody. To avoid this problem, we observe that if P is a presentation which we wish to check for realizability, and P' is any presentation obtained from P by reducing P to minimal length, then P is realizable if and only if P' is realizable.

Theorem. (ZIESCHANG) Suppose P is a finite presentation which is freely and cyclically reduced. Suppose that P' is obtained from P by reducing P to minimal length. Then there is a constructive procedure to determine whether P' is realizable, and to find all possible Heegaard diagrams D such that D realizes P' and $|D| = |P'|$.

Proof. Let WG be the Whitehead graph of P'. There are only a finite number of distinct embeddings of WG in the plane up to homeomorphism. Then for each such embedding, and each pair of vertices of WG which are inverses say X,x, there are only a finite number of distinct ways to identify the ends of the edges of WG which meet X with the ends of the edges of WG which meet x.

While this gives a constructive procedure for solving the realization problem, and a constructive procedure for finding all possible realizations of a minimal length presentation, its brute force approach is much too inefficient to be used in any but the simplest situations. Hence we seek more efficient procedures. We have not completely solved this problem. The procedures that the program currently uses seem to be quite effective in practice, but we do not know to what extent they provide an algorithm for determining realizability. We will discuss this more below after giving an outline of the procedures the program currently uses.

When the program tries to determine whether the original presentation passed to it is realizable, it first attempts to do so without reducing the number of generators involved. Then, if the program is not initially successful in determining whether the original presentation is realizable, the program will start looking for primitive relators which it can use to reduce the genus of the associated Heegaard diagram. However, some care is required here for the following reasons.

Note that if P is a realizable presentation, and P' is any presentation obtained from P by deleting a sequence of primitive relators, then P' is also realizable and P' determines the same 3-manifold as P does. So, if P' turns out to be not realizable, then neither is P. On the other hand, if P' is realizable, then one cannot conclude, in general, that P is realizable. Under certain circumstances, however, one can infer the realizability of P from the realizability of P'. This is the case, when P' was obtained from P by deleting what the program calls "trivial generators". Here is the definition of a "trivial generator".

Def. Let P be a presentation, a generator G appearing in P, is a "trivial generator" if:

(1) The total number of appearances of G, together with its inverse g, in the relators of P is less than or equal to two.
(2) There is exactly one relator in P of length one which has the form G or g.

It is not hard to see that P is realizable if and only if P' is realizable, provided P' is obtained from P by deleting a sequence of "trivial generators". So when the program looks for primitive relators, which it can use to reduce the number of generators appearing in the original presentation, it looks first for "trivial generators", and if it obtains a realizable presentation P' from P by deleting a sequence of "trivial generators", then the program asserts that the original presentation is realizable.

If the program is not successful in determining whether P is realizable by deleting a sequence of "trivial generators", then the program looks for other primitive relators which it can use to eliminate generators. In this case, if the program finds a presentation P' which is realizable, the program will assert that P' is realizable, but the program will not assert that the original presentation P is realizable.

Finally, here is an outline of the procedure the program uses to determine realizability.
`Check_Initial_Presentation_For_Realizability(){`

1. Reduce P to minimal length.
2. Obtain the Whitehead graph WG, of P.
3. Obtain the reduced Whitehead graph RWG, of P.
4. Check whether RWG is connected.
   A. If RWG is not connected, the associated 3-manifold is a connected-sum. Split the relators of P into two disjoint sets which correspond to presentations of the summands, and return.
   B. If RWG is connected, continue.
4. Check RWG for pairs of separating vertices.

A. If *RWG* has pairs of separating vertices, call the program's routine `Level_Transformations()`, and have it look for new presentations by sliding the components of separations around.

   a. If `Level_Transformations()` finds an annulus, report the existence of the annulus and return.

   b. If `Level_Transformations()` finds a presentation which has no pairs of separating vertices in its Whitehead graph, go to 5).

   c. If `Level_Transformations()` finds a presentation whose reduced Whitehead graph is nonplanar, report that the initial presentation is not realizable, and return.

   d. Otherwise, see if the current presentation has any "trivial generators" which can be eliminated.

      1. If such "trivial generators" exist, eliminate some, and go to 1).
      2. Otherwise, look for other relators which represent "primitives".

         A. If such "primitives" exist, use some of them to reduce the rank of the presentation, and go to 1).

         B. Otherwise, declare that the program cannot determine whether the presentation is realizable, and return.

B. If *RWG* has no pairs of separating vertices, continue.

5. Check *RWG* for planarity.
   A. If *RWG* is not planar, declare *P* not realizable and quit.
   B. Otherwise, *RWG* has a unique embedding in the plane. Find this embedding, extend the embedding to an embedding of *WG*, and continue.

      Note: Because of way the program defines pairs of separating vertices, the embedding of *WG* is completely determined by the embedding of *RG*.
6. Try to determine how ends of edges at pairs of inverse vertices in the Heegaard diagram must be identified.
   A. If there are generators which appear in the presentation with only one exponent, up to absolute value, and this exponent is large enough to cause nonuniqueness, report this fact and call `NonUnique_Initial_Diagram()`.
   B. If there is a "valence-two-annulus" present, report this fact and call `NonUnique_Initial_Diagram()`.
   C. Otherwise, the identifications of the edges at pairs of inverse vertices are uniquely determined; compute what these identifications are, and continue.
7. Check whether the diagram just determined realizes the current presentation.
   A. If the diagram does not realize the current presentation, declare the initial presentation not realizable and return.
   B. If the current presentation was obtained by eliminating "primitives" which were not "trivial generators", declare that the current presentation is realizable.
   C. Otherwise, the current presentation was obtained from the original presentation by "level-transformations", and eliminating only "trivial generators". Declare the original initial presentation realizable.

}

Question. It would be interesting to have an example of a presentation P with the following properties.

(1) P is not realizable.
(2) The program cannot detect that P is not realizable.
(3) There is a set S_1 of relators of P, each of which is a primitive, such that the presentation P_1 obtained by forming the quotient of P by the members of S_1, is realizable, and P_1 uniquely determines a 3-manifold $M1$.
(4) There is a set S_2 of relators of P, each of which is a primitive, such that the presentation P_2 obtained by forming the quotient of P by the members of S_2, is realizable, and P_2 uniquely determines a 3-manifold M_2.
(5) M_1 and M_2 are not homeomorphic.

14. THE DEFINITION OF "PAIRS OF SEPARATING VERTICES".

Suppose P is a presentation which has minimal length, RWG is the "reduced Whitehead graph" of P, and RWG is connected. The definition of "pairs of separating vertices" has been chosen so that if RWG has no pairs of separating vertices, then both RWG and the Whitehead graph WG of P have unique embeddings in the plane up to homeomorphism. Uniqueness of the embedding of WG follows from uniqueness of the embedding of RWG, when no pairs of separating vertices exist, because all of the edges of WG, which join a given pair of vertices in WG, must be embedded so that they are parallel to each other, thus forming a "band" of parallel edges.

Note: Because RWG corresponds to a presentation which has minimal length, RWG does not have any "cut vertices".

Before giving the definition of "pairs of separating vertices", we need the following preliminary definitions.

Def. Let V be a vertex of RWG, the "valence" of V is the number of distinct vertices joined to V by edges of RWG.

Def. A vertex V in RWG is a "major vertex" of RWG if V has valence greater than two in RWG.

Now suppose that $V1$ and $V2$ are two major vertices of RWG. Let RWG' be the graph obtained by deleting $V1$ and $V2$ from RWG and suppose that RWG' has more than one component. We say that the pair $\{V1,V2\}$ form a "pair of separating vertices" of RWG provided one of the following holds:

1) RWG has only two major vertices, namely the pair {V1,V2}, and:
   a) $V1$ and hence $V2$, has valence greater than 3 or,
   b) There is more than one edge joining $V1$ and $V2$ in the Whitehead graph WG of P.
2) RWG' has more than two components, or RWG' has two components, $V1$ and $V2$ are joined by an edge and at least one component of RWG' contains a major vertex of RWG.
3) RWG' has two components, $V1$ and $V2$ are not joined by an edge and each component of RWG' contains a major vertex of RWG.

15. HOW THE PROGRAM DETERMINES THE IDENTIFICATIONS OF THE EDGES MEETING A VERTEX V WITH THE EDGES MEETING THE INVERSE OF V WHEN THE PROGRAM IS RECOVERING A HEEGAARD DIAGRAM FROM AN EMBEDDING OF THE WHITEHEAD GRAPH OF A PRESENTATION IN THE PLANE.

Suppose P is a minimal length presentation with Whitehead graph WG, reduced Whitehead graph RWG, RWG is connected, RWG has no pairs of separating vertices, and P is realizable. Then RWG has a unique embedding in the plane, and this unique embedding can be extended to a unique embedding of WG. Suppose this has been done. In order to recover the Heegaard diagram determined by P, the program needs to be able to determine, for each pair of inverse vertices, how the edges which meet one vertex are to be identified with the edges meeting its inverse. The problem falls naturally into two cases:

1) The corresponding generator appears in P only with exponents less than or equal to two in absolute value.
2) The corresponding generator appears in P with an exponent which is greater than two in absolute value.

We treat case 1) first. Suppose that $\{Y,y\}$ is such a pair of vertices. Assume for the moment that there are no relators in P which are powers of Y. Then among the bands of parallel edges that join vertex Y or vertex y to other vertices of WG, let M be a band which contains a maximal number of edges. Let m be the number of edges of WG in M. Let E be the total number of edges of WG meeting Y (which is of course equal to the total number of edges of WG meeting y.) Note that if one of $\{\mathrm{Y},y\}$ has valence two in RWG and the other has valence greater than two in RWG, then we may choose the band M such that it meets the vertex of valence two. Next, note that it is a consequence of the minimal length of P that $2m \leq E$.

Suppose, for the purposes of illustration, that the band M joins vertices x and Y. The program then reads through the relators and their inverses, looking for appearances of subwords of length 3 which start with XY. Suppose it finds n_1 appearances of XYF, n_2 appearances of XYg and say n_3 appearances of XYh. Then $m = n_1 + n_2 + n_3$, and the m edges meeting Y must be identified with m consecutive edges meeting y so that n_1 of these edges go to vertex F, n_2 of these edges goto vertex g and n_3 of these edges go to vertex h. If either vertex Y or vertex y has valence greater than two in RWG, there is only one way to make such an identification. If both vertex Y and vertex y have valence two in RWG, then there are no more than two possible ways to make such an identification. (When there are two possible ways to make this identification, the program will later call its subroutine `Valence_Two()`which will attempt to resolve the ambiguity.)

This takes care of case 1) except when there are relators which are powers of Y. However, it is quite easy to make the appropriate modifications to handle this situation, so we will omit the details.

Finally, we treat case 2). So suppose that $\{Y,y\}$ is a pair of vertices such that the corresponding generator appears in P with an exponent which is greater than two in absolute value, and suppose, for the moment, that there are no relators in P which are powers of Y. Since Y appears in P with an exponent which is greater then two in absolute value, there are edges in WG joining Y and y. Furthermore, all such edges are parallel to each other. This implies that Y appears in P with exponents which take at most three distinct absolute values, and these values are of the form p,q and $p+q$, with p and q relatively prime. (See the lemma below for a proof of this.)

In this case, the program counts the total number of appearances of Y^p, Y^q, and Y^{p+q}, in the relators and their inverses. Suppose the number of times Y appears with each of these exponents is respectively d, e and f and at least two of these numbers are nonzero. Then proceeding counter-clockwise around vertex Y, starting with the first edge that joins Y to a vertex distinct from y, one must meet either d consecutive edges which represent Y^p followed by f consecutive edges which represent Y^{p+q} and finally e consecutive edges which represent Y^q or one must meet e consecutive edges which represent Y^q, followed by f consecutive edges which represent Y^{p+q} and finally d consecutive edges which represent Y^p.

If one of $\{Y,y\}$ has valence greater than two in RWG, the program can easily determine which of these two alternatives holds. Suppose, for example, that Y has valence greater than two in RWG. And suppose that the first vertex following y in counter-clockwise order about Y is x and that the vertex preceding y in counter- clockwise order about Y is z. Then x and z are distinct. The program again reads through the relators, this time looking for subwords of the form XY^e and ZY^e where e is one of $p,q,p+q$. The values of e which appear give the program enough information to determine which of the two alternatives above is correct. It is then an easy matter, involving a simple formula, to determine explicitly what the appropriate identifications of the edges at Y and y must be.

As in case 1), if both vertex Y and vertex y have valence two in RWG, then there are at most two possible ways to identify the edges at Y with the edges at y. (As before, if this ambiguity arises, the program will later call its subroutine `Valence_Two()` which will attempt to resolve the ambiguity.)

This takes care of case 2) except when there are relators which are powers of Y, or Y appears in the relators with only one exponent. Note that if there are relators which are powers of Y, then Y must appear with only one exponent. And if Y appears with with only one exponent, there is a genuine ambiguity.

The next lemma extends a result due to Osborne for Heegaard diagrams of genus two to minimal length diagrams of arbitrary genus. It also underscores how obtaining diagrams without pairs of separating vertices simplifies the situation.

Lemma. Suppose P is a minimal length realizable presentation, X is a generator of P, x is the inverse of X, the pair of vertices $\{X,x\}$ do not form a pair of separating vertices of the reduced Whitehead graph RWG of P, and X appears in P with an exponent e, where $|e|>1$. Then, up to absolute value, X can appear in P with at most 3 distinct exponents, which must be of the form: p,q and $p+q$, where p and q are relatively prime.

Proof. Let D be a Heegaard diagram which realizes P, with $|D|=|P|$. Let H be the under- lying handlebody. Since X appears in P with exponent e, where $|e|>1$, there exist edges of D joining vertex X of D to vertex x of D. Since $\{X,x\}$ do not form a pair of separating vertices of RWG, all of the edges of D which join X to x must be parallel to each other. Let N be a regular neighborhood in D of X,x and an edge of D joining X to x. Note that up to isotopy in D, we may assume that all of the edges of D which join X to x, lie in the interior of N. Note ∂N cuts off a punctured torus from the Heegaard surface ∂H. The attaching curves of the 2-handles which realize P, are either disjoint from ∂N, or they are cut into sets of arcs by ∂N. It is well-known and not hard to show, that there can be at most 3 distinct sets of properly embedded arcs on a punctured torus, such that no two arcs in distinct sets are parallel. This implies that if a subarc of an attaching curve of a 2-handle lies in N, then it intersects ∂X in one of at most 3 ways. But successive subarcs of an attaching curve, which lie in N, are separated by intersections of the attaching curve with cutting disks of H, from other handles, distinct from X. Thus the claim holds.

16. SOME REMARKS ABOUT OBSTRUCTIONS TO REALIZING A PRESENTATION UNIQUELY.

Suppose that P is a minimal length presentation. There are essentially three types of obstructions to uniqueness which arise as the program attempts to determine whether P is realizable, each of which is fairly easy to understand. The most obvious obstruction occurs when the underlying handlebody H contains an essential disk X, such that ∂X is disjoint in ∂H from the attaching curves of the 2- handles of a diagram. If X separates H, the minimal length of P implies that the associated 3-manifold M is a connected sum. However, the program will often not be able to determine whether the sum is along a disk represented by X, or along a 2-sphere formed by X and a disk lying in a 3-ball used to cap-off a boundary component of M. If X does not separate H, then again the associated 3-manifold is a connected sum, but the situation can be somewhat more complicated than the previous situation. There are essentially 3 possibilities.

1) ∂X is an essential separating curve in ∂M, so that X represents a 1-handle which joins two distinct boundary components of $M - N(X)$.

2) ∂X is an essential nonseparating curve in ∂M, so that X essentially represents a disk connected sum of $M - N(X)$ and a solid torus.

3) ∂X is capped off by a disk in a 3-ball used to cap-off a boundary component of M. In this case, X essentially represents an $S^1 \times S^2$ connected summand of M.

Often, the program will not be able to determine which of these possibilities is correct, and it should admit that it does not know. If the program does know how M is constructed, then it should describe the boundary structure of M, and list the number of summands of the form $I \times D^2, S^1 \times S^2, S^1 \times D^2$ that are involved. The program does this in a somewhat cryptic way, however, which we will try and explain. What the program does which may be a little confusing, is it lumps all of these "handles" together. For example, suppose M has 3 summands, M_1, M_2, M_3. And suppose M_1 is closed, M_2 is a handlebody of genus 2, and M_3 is a handlebody of genus 3. The program will lump the handles of M_2 and M_3 together and tell you that there are 5 $S^1 \times D^2$(s) present. However, it should also tell you that M_1 is closed, and that M has 2 boundary components, one of genus 2, and one of genus 3, this allows you to deduce that the 5 $S^1 \times D^2$(s) are partitioned into a handlebody of genus 2 and a handlebody of genus 3, and thus you have enough information to determine how M is constructed.

17. PAIRS OF SEPARATING VERTICES IN THE REDUCED WHITEHEAD GRAPH OF P.

The second type of obstruction, and the one that seems to arise most frequently in practice, is the existence of pairs of separating vertices in the reduced Whitehead graph of P. This topic is discussed in several other places, so we will not say any more about it here, and instead pass on to the third type of obstruction.

18. "ANNULI" AS OBSTRUCTIONS TO REALIZING A PRESENTATION UNIQUELY.

The third type of obstruction, is perhaps the most interesting from a topological point of view, because, when it arises, it tends to give some topological information about the structure of the

3-manifold M, which might not otherwise be easily detected. This obstruction arises when there is an essential annulus A in the underlying handlebody H such that ∂A is disjoint in ∂H from the attaching curves of the 2-handles of a diagram which realizes P. We note that if such an annulus exists, then A may or may not represent an essential annulus, disk or 2-sphere in M depending upon whether A is incompressible in M and whether one or both boundary components of A bound disks in M. At the present time, the program alerts you to the fact that it has found such an annulus in the underlying handlebody, but it then leaves further disposition of the situation up to you.

As far as what characterizes uniqueness of realizability in general, we have considered the question, and believe we can show that the following result is true.

Theorem. Suppose P is a minimal length presentation on g generators, $g > 1$, such that the reduced Whitehead graph of P is connected, and suppose that D and D' are distinct Heegaard diagrams of genus g with $|D| = |D'| = |P|$, such that both D and D' realize P. Let H be the underlying handlebody of D. Then there is an essential annulus A in H, with ∂A disjoint in ∂H from the attaching curves of the 2-handles of D. Similarly, there exists such an annulus in the underlying handlebody H' of D'.

The proof is too long to give here, but we can indicate the general ideas. Suppose that P' is a minimal length presentation obtained from P via a sequence of level-transformations. If the reduced Whitehead graph of P' embeds uniquely in the plane, then either there exists a unique diagram D', with $|D'| = |P'|$, which realizes P', or a Valence-Two-Annulus exists, or there is a generator X of P' which appears with only one exponent, which is greater than two in absolute value, and again an annulus exists. Hence every presentation obtainable from P via level-transformations, has a reduced Whitehead graph which has pairs of separating vertices. In particular the reduced Whitehead graph of P has pairs of separating vertices. Next, we use, what are essentially covering space techniques, to show that there exists an essential singular annulus A, in the underlying handlebody H, with ∂A disjoint from the attaching curves of the 2-handles of D. One can then do surgery on A in order to obtain the desired essential annulus.

19. HOW THE PROGRAM HANDLES PRESENTATIONS WHICH CONTAIN DUPLICATED RELATORS.

The program will accept and handle presentations which contain relators which are identical up to taking cyclic conjugates or inverses, but it is not very tolerant of them. The reason for this is not hard to see, and comes from the fact that the program places a heavy reliance on the uniqueness of the realization of a minimal length presentation in order to insure that it is preserving the homeomorphism type of the associated 3-manifold. Suppose, for example, that P is a presentation which contains two relators R_1 and R_2, which are identical up to taking cyclic conjugates and inverses. Let H be the underlying handlebody and suppose D is a realization of P with $|D| = |P|$, and suppose that R_1 and R_2 are realized by simple closed curves C_1 and C_2 in ∂H, such that C_1 and C_2 are not parallel in ∂H. Then D is clearly not unique, since P obviously has another realization in which C_1 and C_2 are embedded in ∂H such that they are parallel. Thus the situation is the following: Either P does not have a unique realization, in which case, the program will make no further use of P, or P has a unique realization, in which case, copies of duplicated relators are redundant, and the program can and will delete the extraneous copies.

20. AN EXCEPTION TO THE RULE THAT PRESENTATIONS IN WHICH A GENERATOR APPEARS WITH ONLY ONE EXPONENT, WHICH IS GREATER THAN TWO IN ABSOLUTE VALUE, ARE NOT UNIQUELY REALIZABLE.

Generally, if a generator appears in the relators with only one exponent, and that exponent is greater than two in absolute value, then the Heegaard diagram is not unique. There are exceptions to this however. These exceptions occur when the "reduced" Whitehead graph is homeomorphic to a circle and there are not too many edges in the Heegaard diagram. The program contains a routine which is called to check whether uniqueness holds in this special set of circumstances. Without this special check, the program would declare that the diagram corresponding to the relator *AAABB* was not unique because generator *A* appears only with exponent 3. However this diagram is unique because there is an involution of the diagram which exchanges the two major faces of the diagram. Here is an outline of this test for uniqueness.

`Test_Uniqueness()`{

(1) If more than one generator appears with maximal exponent greater than two in absolute value, declare the diagram not unique and return.
(2) If the generator which appears with maximal exponent greater than two in absolute value appears with exponent *e* and *e* is not 3,4 or 6, declare the diagram not unique and return. Note that 3,4 and 6 are the only integers greater than two which have the property that at most two smaller integers are relatively prime to them.
(3) If there is a vertex of the reduced Whitehead graph which is joined to more than two other vertices, declare the diagram not unique and return.
(4) If *i* and j are vertices of the Whitehead graph, which are not inverses, and *i* and j are joined by more than one edge in the Whitehead graph, declare the diagram not unique and return.
(5) Declare the diagram unique and return.

}

This routine has been involved when the program produces messages like the following:

(1) "The diagram is not unique because there is a generator which appears with only one exponent and that exponent is greater than 6."
(2) "The diagram is not unique because there is a generator which appears only with exponent 5."
(3) "The diagram is not unique because there is a generator which appears only with exponent 3 or only with exponent 4 and this exponent occurs more than once."
(4) "The diagram is not unique because there is a generator which appears with only one exponent, either 3,4 or 6, and a needed symmetry does not exist."

21. HOW THE PROGRAM ATTEMPTS TO FIND PRESENTATIONS, WHICH HAVE NO PAIRS OF SEPARATING VERTICES IN THEIR REDUCED WHITEHEAD GRAPHS, BY PERFORMING LEVEL-TRANSFORMATIONS.

The program puts a premium on obtaining minimal length presentations which have no pairs of separating vertices in their reduced Whitehead graphs. It does this since such graphs have a unique embedding in the plane, and this unique embedding can be extended to a unique embedding of the Whitehead graph of the presentation in the plane. And this, in turn, makes finding the Heegaard diagram which realizes the presentation fairly easy.

Suppose P is a minimal length presentation whose reduced Whitehead graph G is connected and $\{X,Y\}$ is a pair of separating vertices of G. Let x denote the inverse of X, and let y denote the inverse of Y. We need to distinguish between two types of separating vertices, and so we make the following definition:

Def. Let G' be the graph obtained by deleting X and Y from G. Say $\{X,Y\}$ is a "Type 1" pair if Y = x, or if there exists a component C of G' which contains both x and y. If no such component C exists, say that $\{X,Y\}$ is a "Type 2" pair.

Now, because P has minimal length, the following lemmas hold.

Lemma. Let the notation be as above. Suppose that $\{X,Y\}$ is a Type 1 pair of separating vertices of G. Let C be a component of G' such that neither x or y lies in C. Let m be the number of edges of the Whitehead graph WG of P which join vertices of C to vertex X. Similarly, let n be the number of edges of WG which join vertices of C to vertex Y. Then $m = n$.

Lemma. Let the notation be as above. Suppose that $\{X,Y\}$ is a Type 2 pair of separating vertices of G. Then G' has only two components, and there is no edge joining vertex X to vertex Y in G.

Lemma. Let the notation be as above. Suppose that $\{X,Y\}$ is a Type 2 pair of separating vertices of G. Let C_x be the component of G' which contains vertex x, and let C_y be the component of G' which contains vertex y. Let m be the number of edges of the Whitehead graph WG of P which join vertices of C_y to vertex X. Similarly, let n be the number of edges of WG which join vertices of C_x to vertex Y. Then $m = n$.

These lemmas imply that if P has minimal length, and the reduced Whitehead graph G of P has a pair of separating vertices, then there always exist level-transformations. In particular, there always exists level-transformations which can be viewed as "sliding" around a subgraph of G, which is attached to the rest of G at only two vertices. However, as indicated by the lemmas, the level-transformations involved differ according to whether the pair of separating vertices is a Type 1 or Type 2 pair. The program includes routines designed to deal with each of these two cases. The program's routine `Level_Transformations()` deals with Type 1 pairs, and the program's routine `Level_Transformations_2()` deals with Type 2 pairs.

Here is a brief description of how these routines work. `Level_Transformations()` and `Level_Transformations_2()` call each other recursively, with `Level_Transformations()` being called first. Each invocation of `Level_Transformations()` "owns" a presentation. On entry, the routine first checks that the presentation being passed to it is distinct from those passed to all previous invocations. The routine also checks at this point whether the previous invocation of `Level_Transformations()` has managed to get rid of all pairs of separating vertices in the graph. If so, the routine saves this presentation and returns. Otherwise this presentation "belongs" to this invocation, and the routine saves a copy of the presentation.

The routine then looks for a pair of separating vertices. If it finds a pair, say $\{X,Y\}$, then it looks for a component "TheComp", of the separation produced by deleting $\{X,Y\}$, with the property that neither X inverse or Y inverse are in TheComp.

Then, the objective is to "slide" the component TheComp around the Whitehead graph, sliding first to the "left" and then (if necessary) to the "right" until either TheComp is attached to the main body of the graph at more than two distinct vertices, or we arrive at a vertex of attachment

V of TheComp with the property that V inverse is a member of TheComp, or we manage to slide TheComp completely around the graph along some path, and the presentation has returned to the original presentation. If the last case occurs, then there is an "annulus" present. Otherwise we call `Level_Transformations()` recursively and pass it the new presentation. (See below for more precise details about what it means to "slide" a component around.)

Finally, when `Level_Transformations()` has done all it can with the presentation passed to it, `Level_Transformations()` calls `Level_Transformations_2()`, which deals with any Type 2 pairs. Except for differences in the sets of vertices which it tries to slide around, `Level_Transformations_2()` is similar to `Level_Transformations()`. Outlines of these two routines are given below. In particular, the outline of `Level_Transformations_2()` indicates how the sets of vertices which it tries to slide around differ from the sets of vertices that `Level_Transformations()` slides around.

`Level_Transformations(P)`{

1) Check whether presentation P "belongs" to a previous invocation of `Level_Transformations()`. If it does, return.
2) Check whether P has any pairs of separating vertices.
   A) If P has no pairs of separating vertices, see if P is "new"
      a) If P is "new", save a copy of P and return.
      b) If the program already has a copy of P on file, return.
   B) P has pairs of separating vertices, so continue.
3) Save a copy of P. P now "belongs" to this invocation of `Level_Transformations()`.
4) Find the components produced by deleting the current set of separating vertices from the reduced Whitehead graph of P.
5) If $\{X,Y\}$ are the current separating vertices, look for a component "TheComp" of the separation, with the property that neither x or y lie in TheComp.
   A) Slide TheComp over as many handles as possible; starting with handle X. Let P' be the presentation obtained from P, by this slide of TheComp.
   B) If there is an annulus present, return.
   C) Call `Level_Transformations(`P'`)`.
   D) Restore P.
   E) Slide TheComp over as many handles as possible; starting with handle Y. Let P'' be the presentation obtained from P, by this slide of TheComp.
   F) If there is an annulus present, return.
   G) Call `Level_Transformations(`P'').
   H) Restore P.
   I) If there are other components of the separation which we can slide around, go to 5).
6) Look for another pair of separating vertices.
   A) If there are other pairs of separating vertices, go to 4).
   B) All pairs of separating vertices have been checked for Type 1 separations. Call `Level_Transformations_2(`P).
   C) Return.

}

`Level_Transformations_2(P)`{

1) Find a pair of separating vertices of the reduced Whitehead graph G of P.
2) If the current pair of separating vertices $\{X,Y\}$, is a Type 2 pair, do the following:

A) Let G' be the graph obtained by deleting X and Y from G. Let C_x be the component of G' which contains vertex x, and let C_y be the component of G' which contains vertex y. Let CXx be C_x union vertex X. Let CYy be C_y union vertex Y.
B) Slide CXx over as many handles as possible; starting with handle Y. Let P' be the presentation obtained from P, by this slide of CXx.
C) If there is an annulus present, return.
D) Call `Level_Transformations(`P'`)` .
E) Restore P.
F) Slide CYy over as many handles as possible; starting with handle X. Let P" be the presentation obtained from P, by this slide of CYy.
G) If there is an annulus present, return.
H) Call `Level_Transformations(`P''`)`.
I) Restore P.

3) Look for another pair of separating vertices in the reduced Whitehead graph of P.
A) If there are other pairs of separating vertices, go to 2).
B) If all pairs of separating vertices have been checked, return.

}

22. A MORE DETAILED EXPLANATION OF WHAT IT MEANS TO "SLIDE" A COMPONENT OF A SEPARATION AROUND.

Suppose that $\{X, Y\}$ are a pair of separating vertices of RWG, and "TheComp" is a component of the separation. Here are is a more precise description of what the program does when it "slides" TheComp around. Suppose that P has been realized by a Heegaard diagram D, with $|D| = |P|$. Let $n > 1$, be the smallest integer such that there exists a set $SX$of oriented paths in D, which have the following properties:

1) Each path in $SX$is a union of oriented edges of D with the head end of edge i identified in D with the tail end of edge $i+1$ for $1 \leq i < n$.
2) The first edge of each path in SXhas its tail end at a vertex in TheComp and its head end at vertex X.
3) If 1 ¡ i ¡ n, the ith edges of all paths in SX join the same pair of vertices of RWG.
4) If i = n, either a), b) or c) below holds:
   a) The nth edges of two distinct paths in SX have their head ends at distinct vertices of RWG and c) has not occurred.
   b) The head end of the nth edge of each path in SX lies at a vertex V of RWG, such that the vertex of RWG which represents the inverse of V, lies in TheComp.
   c) The tail end of the nth edge of each path in SX lies at vertex Y, and the head end of the nth edge of each path in SX lies at a vertex in TheComp.

Now there is a sequence of $n-1$ level-transformations, which topologically amounts to sliding TheComp over handle X and continuing until:

1) If a) holds TheComp is now attached to rest of RWG at more than two distinct vertices.
2) If b) holds TheComp is still attached to the rest of RWG at only two distinct vertices, but the vertices of attachment form a Type 2 pair of separating vertices.
3) If c) holds the underlying handlebody H contains an essential annulus A, with ∂A disjoint from the attaching curves of the 2-handles which realize P.

Remark. We mention that these routines, while they seem to work quite well in practice, are only a means by which to search for a presentation which has no pairs of separating vertices in its reduced Whitehead graph, or failing that, to locate an essential annulus in the underlying handlebody. In particular, they do not provide an algorithm to either find such a presentation or to find an annulus. Thus answers to the following questions would be helpful in either placing these routines on a firmer footing, or in suggesting alternatives.

QUESTION 1) What characterizes the set of minimal length, realizable, presentations which have the property that some presentation obtainable from them via level-transformations, has a reduced Whitehead graph without any pairs of separating vertices ? In particular, is the answer to Question 2) yes ?

QUESTION 2) Suppose P is realizable, P has minimal length, and there is no essential annulus A, in the underlying handlebody H, such that ∂A is disjoint from the attaching curves of the 2-handles of P. Suppose the reduced Whitehead graph of P has a pair of separating vertices, does there exist a presentation P', obtainable from P via a sequence of level- transformations, such that the reduced Whitehead graph of P' has no pairs of separating vertices ?

QUESTION 3) Suppose P is a realizable minimal length presentation such that some presentation P', obtainable from P via level-transformations, has a reduced Whitehead graph without pairs of separating vertices. What is the most efficient way to find P' ?

23. LEVEL TRANSFORMATIONS OF PRESENTATIONS (GENERAL CASE)

Let P be a minimal length presentation whose Whitehead graph WG, is connected. Here is a short description of an efficient procedure we have devised, which uses network flows to find all of the nontrivial level-transformations of P. We apologize for being fairly sketchy about many of the details, and for leaving most of the proofs to the reader.

Suppose that A is a vertex of WG, and a is the vertex of WG which represents the inverse of A. We consider the problem of finding all nontrivial level-transformations, in which generator A acts on the remaining generators of P. By the preceding results, this is equivalent to finding all of the minimal cut-sets of edges of WG, which separate vertex A, from vertex a. Let C be such a minimal cut-set. We can look for C in following way. Consider vertex A as the source vertex and vertex a as the sink vertex of a flow in WG, and then find a maximal flow F in the network WG from vertex A to vertex a. Interpret F as a maximal collection of directed paths from vertex A to vertex a, such that no two paths in F have any edges in common.

Let G be a network all of whose edges are undirected. Note that if F is a flow on G, then we may view those edges of G, which are traversed by paths in F, as assuming an induced direction. Note this is well defined, since an edge of G is traversed by at most one directed path in F.

Def. Suppose that X and Y are a pair of vertices of G, which are joined by n edges of G. Say that the set of edges joining X and Y is "saturated" if all n of these edges have the same induced direction, and otherwise say the set of edges joining X and Y is "unsaturated".

The objective is to produce a sequence of new networks by identifying vertices in WG, and new maximal flows on these networks, until finally, we obtain a network G and a maximal flow F on G, such that F is acyclic and each set of edges, which joins a pair of vertices of G, is saturated. The point is that these changes can all be made without affecting the set of minimal cut-sets that separate source and sink. And, the set of minimal cut-sets can be easily enumerated, when the flow F is acyclic, and all bands of edges, of the network, are saturated.

Let G be a network, and let F be a flow on G. Observe that if X and Y are vertices of G and some of the edges of joining X and Y are directed from X to Y and some of the edges joining X and Y, are directed from Y to X, then we can do "surgery" on the paths in F to obtain another flow F' such that any two directed edges joining X and Y have the same induced direction, and F' has the same number of paths joining source and sink as F does.

Suppose this has been done for all pairs of vertices of G. Thus, we may assume that if X and Y are vertices of G, which are joined by edges of G, then any two directed edges joining X and Y have the same induced direction.

The following lemma is an immediate consequence of minimality and the fact that a maximal flow contains $V(A)$ independent edge-wise disjoint paths.

Lemma. Suppose X and Y are vertices of G, which are joined by an unsaturated set of edges in G, then X and Y lie on the same side of any minimal cut-set which separates source from sink.

Suppose X and Y are joined by an unsaturated set of edges, then we can form a new graph G', from G, by identifying X and Y and deleting any edges which joined X to Y. We do this operation repeatedly, and if necessary, perform surgery on the flow F, so that all sets of edges joining pairs of vertices of G' are saturated.

Next, suppose that G is a network, F is a maximal flow on G, from source to sink, and F has a cycle. Then we can do surgery on F and obtain a new maximal flow on G from source to sink, such that F' has the property that the sets of edges in the cycle are unsaturated. This allows us to identify all of the vertices of the cycle to a single vertex, to delete all of the edges of G, which join vertices in the cycle to each other, and to further contract the graph. It is also not hard to find another maximal flow on the contracted graph.

Def. Suppose G is an acyclic directed graph. Let X and Y be vertices of G. Say Y is a "descendant" of X, if there is a directed path in G from X to Y.

Def. Let S be a subset of the vertices of G. S is "hereditarily closed" if whenever a vertex X lies in S, then each descendant of X also lies in S.

Each hereditarily closed subset S of the vertices of G partitions the vertices of G into two subsets, S and its complement. This partition in turn induces a partition of the vertices of WG into two subsets, and the edges of WG which join vertices in one subset to vertices in the other subset, form a minimal cut-set of WG. These minimal cut-sets give us the desired set of level-transformations.

24. THE STRATEGY THE PROGRAM USES TO OBTAIN NEW PRESENTATIONS BY FORMING BANDSUMS.

Suppose P is a minimal length presentation whose Whitehead graph is connected, and D is a Heegaard diagram with $|D| = |P|$ which realizes P. Let H be the underlying handlebody. The relators in D, intersect the boundaries of the cutting disks of H in points, which cut the boundaries of these cutting disks into arcs. When the program forms bandsums, it uses subarcs of the boundaries of the cutting disks as the "band" along which the sum is formed. In a typical diagram, many of these arcs are "parallel" and hence lead to the same bandsum. The program therefore restricts its attention to bandsums formed along those subarcs, of the boundaries of cutting disks of H, which correspond to "corners" of the faces of the reduced Whitehead graph of P.

When the program forms bandsums, it is in one of two modes. In the GoingDown mode, the program is attempting to reduce the length of a diagram by performing automorphisms and bandsums. So, in this mode, the program checks bands corresponding to all of the corners of the faces of the reduced Whitehead graph of P, and it looks for a bandsum which reduces the length of P as much as possible.

Otherwise, the program is in its GoingUp mode, and it looks for bandsums which increase the length of P. This mode is an exploratory mode, and so the program needs to be able to generate a spectrum of presentations. Here is a brief description of how it currently does this.

Each presentation, which the program saves, can be used by the program as the root of a new branch of the program's search tree. The program will try to grow a new branch of the tree from a given presentation up to $2g$ times, provided the presentation is a presentation on g generators. Suppose n is an integer with $0 leqn < 2g$, and this is the nth attempt by the program to grow a new branch of the program's search tree from presentation P.

The program selects, at random, $3(2g-n)$ edges, as above, which will serve as trial bands. Then the program finds, in this set of trial bands, a band which yields a bandsum that increases the length of P by the smallest nonnegative amount, and performs that bandsum, yielding a new presentation P'. The program then obtains the diagram of P', and returns to form another bandsum, again selecting a subset of $3(2g-n)$ random trial bands. This process is continued until one of the following occurs.

1) There exists an automorphism which reduces the length of the new presentation P'.
2) The lengths, of the new presentations produced, are more than double the length of the first presentation, on which the program formed the original bandsum.
3) The number of bandsums formed, without anything interesting happening, is greater than $4g$.
4) The program is unable to find the diagram corresponding to P'.

If case 1) occurs, the program reduces P' to minimal length, saves a copy of P' as the "`Top_Of_Chain`", and switches to GoingDown mode. After returning from GoingDown mode, the program replaces the original presentation P with the copy of P', saved as `Top_Of_Chain`, and the program resumes forming bandsums, starting from this presentation. This process is continued until terminated by cases 2), 3) or 4) occurring, or the program has generated and explored a sequence of more than 5 `Top_Of_Chain`presentations.

Finally, when the program exits this loop, n is incremented, so the next search will be over a smaller set of trial bands.

25. HOW THE PROGRAM ATTEMPTS TO DETERMINE THE IDENTIFICATIONS OF EDGES OF THE HEEGAARD DIAGRAM AT PAIRS OF VERTICES, EACH OF WHICH HAVE VALENCE TWO IN THE REDUCED WHITEHEAD GRAPH.

Here is a brief description of the routine which attempts to resolve ambiguities about how edges should be identified, in the Heegaard diagram, for a pair of vertices both of which have valence two in the reduced Whitehead graph of a presentation.

Suppose P is a minimal length presentation, with Whitehead graph WG, and reduced Whitehead graph RWG. Suppose that RWG has no pairs of separating vertices, so that RWG and WG have unique embeddings in the plane. Suppose that both vertex X of RWG and the the inverse vertex x

of X in RWG, have valence two in RWG. And, suppose that the program has determined that there are two possible ways to identify the edges of WG, which meet X with the edges of WG, which meet x. Let B be one of the bands of parallel edges in WG, which meet vertex X, and suppose the edges in B do not join vertex X to vertex x. We note that, since there is an ambiguity about how these identifications should be made, B contains at least two edges. The program has computed the two possible ways that the edges meeting vertex X can be identified with the edges meeting vertex x, and has stored these possibilities in two arrays, which we will call id1[] and id2[]. Next, the program assumes that the identification in id1[] is correct. Using the information in id1[], the program can trace the left-hand edge of B, and the right-hand edge of B, backward over vertex X, and it can follow the paths taken by the left-hand and right-hand edges of B until they leave vertex x, and split apart, with each path going to a distinct vertex of WG. (If this "splitting" did not occur, then there would not be any ambiguity about how edges should be identified at X and x.) Suppose the splitting takes place so that one path leaves vertex x and goes to vertex a of WG, while the other path leaves vertex x and goes to vertex b of WG. The program now reverses direction, and traces these paths in the opposite direction, starting from vertex a in one case, and starting from vertex b, in the other. The program continues to trace these two paths through the diagram, using known information about identifications at each new vertex reached, until one of the following occurs:

1) The two paths traced out have each become longer than the longest relator in P. In this case, P is not realizable, and the program reports that fact.
2) The two paths diverge, and proceed to distinct vertices of WG.
3) The two paths have reached a vertex say Y, which is one of a pair of vertices of valence two in RWG, and the identification of the edges of WG meeting Y with the edges of WG meeting y is still ambiguous.

Suppose 2) occurs, and suppose that the path which originated at vertex a goes to vertex C of WG, while the path that originated at vertex b goes to vertex D of WG. Thus, if P is realizable, and the identification in id1[] is correct for the pair $\{X,x\}$, then subwords of the form AWC, and BWD must occur in the relators of P, where W is the "word" traced out by the two paths up to the point where they diverged. While, if the identification in id2[] is correct for the pair $\{X,x\}$, then subwords of the form BWC, and AWD must occur in the relators of P, where W is the "word" traced out by the two paths up to the point where they diverged. Now it is not hard to show that, under these circumstances, if P is realizable, then at least two and at most 3 of the the 4 subwords $\{AWC,BWD,AWD,BWC\}$ can occur in the relators of P. Thus, by scanning the relators in P, the program can determine which of these possibilities occurs, and then it can determine which identification, of the edges meeting X with the edges meeting x, is correct. This reduces the number of pairs of vertices with ambiguous identifications, and the program can now use this information to try to resolve ambiguous identifications of other pairs of vertices.

26. VALENCE-TWO-ANNULI IN DIAGRAMS

If case 3) occurs, the program marks the band B as having been traced, and looks for another band of edges which it has not traced. If the program cannot find any band of edges, which it has not traced, and which leads to case 1), or case 2), then the diagram contains a cycle of valence two vertices, which form a "Valence-Two-Annulus". In this situation, there is an annulus A in the underlying handlebody H, such that: ∂A swallows an ambiguous vertex of valence two, then ∂A follows the left and right hand edges of the band of parallel edges, which leave this vertex, to another ambiguous vertex of valence two, swallows that vertex, and continues, in this manner, until it returns to the original vertex of valence two. If a Valence-Two-Annulus exists, then the program can arbitrarily choose either of the two possible ways to identify the edges meeting one pair of valence-two vertices, provided at least one vertex of the pair lies in the cycle which constitutes the annulus, and then this choice determines the identifications of all other pairs of ambiguous valence-two vertices, for which at least one vertex of the pair lies in the cycle. By repeating this step, the program can eventually determine the identifications of all ambiguous pairs of valence-two vertices. There are a couple of degenerate situations which can occur. One possibility is that all of the vertices in RWG are of valence-two, and the annulus in question has swallowed everything. In this case, we can ignore the annulus and proceed. Another possibility is that the annulus in question has swallowed all but two vertices of RWG. In this case, the omitted vertices form a pair of inverse vertices, and again the annulus does not cause a problem, so we can ignore it and proceed. (Observe that in this case, ∂A represents a power of a free generator of P.)

27. SOME RESULTS ABOUT REALIZING WORDS BY "PARALLEL PATHS" AND THE EXISTENCE OF "ANNULI".

Suppose P is a presentation which has minimal length, and W is a word such that W makes n pairwise disjoint appearances in the relators of P and their inverses. Let D be a Heegaard diagram, with $|D| = |P|$, which realizes P. It will be useful to have some results which indicate under what conditions the pairwise disjoint subarcs which "realize" the n appearances of W in the relators, must all be "parallel" to each other in D. We start with the following definition of "parallel edges":

Def. Suppose that X and Y are two vertices of a Heegaard diagram D, and there are n edges joining vertex X to vertex Y in D, where n is greater than one. We will say that these n edges are "parallel edges" of D if there exists a pair $E1$ and $E2$ of these edges, such that one of the two complementary regions of the 2-gon, formed by $E1$ and $E2$ together with X and Y, contains the other $n-2$ edges joining X and Y and this complementary region also contains no vertices of D.

If W is a word of length $K+1$, and w is an oriented arc that "realizes" W in D, then w is cut into a set of K successive oriented edges by the vertices of D. If W appears n times in the relators, where n is greater than one, and we want to say that the n arcs which "realize" W are "parallel", then we certainly want each of the K sets of edges, into which the n arcs that realize W are cut by the vertices, to be sets of parallel edges. However, it is not enough for each of these K sets of n edges each to be sets of parallel edges, we also want successive sets of parallel edges to "match up" properly. This motivates the following definition:

Def. Suppose that S is a set of n pairwise disjoint oriented arcs each of which is a union of K coherently oriented edges of a Heegaard diagram D, and suppose each arc in S "represents" the same word W. We will say that the members of S are "parallel" if there exist two members of S, L and R such that for $1 \leq i \leq K$, there exists a complementary region N_i of the 2-gon formed by the edges L_i and R_i and the vertices joined by L_i and R_i, such that the interior of Ni lies to the "right" of the oriented edge L_i, N_i contains no vertices of D, and the ith edge of each member of S lies in the closure of N_i.

The next lemma essentially shows that if the members of a set of edges are parallel in a realization of a minimal length presentation, then the members of the "next" set of edges are also parallel and the successive sets of parallel edges "match up".

Lemma. Suppose that S is a set of n pairwise disjoint oriented arcs each of which is a union of two coherently oriented edges of a Heegaard diagram D, each arc in S "represents" the same word W, D "realizes" a minimal length presentation P, and the first edges of the arcs in S are all parallel. Then the members of S are parallel.

Proof. Consider the word W, W has one of four forms: XYZ, XYY, XXY or XXX, where X, Y and Z are distinct letters. By drawing representative diagrams for each of these cases, and making use of the fact that a Heegaard diagram D, with $|D| = |P|$, which realizes a minimal length presentation P, has no cut vertices, one can easily show that the desired conclusion holds.

The next corollary follows immediately by induction on K.

Cor. Suppose that S is a set of n pairwise disjoint oriented arcs each of which is a union of K coherently oriented edges of a Heegaard diagram D, each arc in S "represents" the same word W, D "realizes" a minimal length presentation P, $|D| = |P|$, and the first edges of the arcs in S are all parallel. Then the members of S are parallel.

28. ANNULI

Under certain conditions, the program will assert that an "annulus" exists in a diagram. We can now use the preceding results to explain and justify the program's claims.

Suppose that P is a minimal length presentation, D is a Heegaard diagram which realizes P, $|D| = |P|$, the pair $\{X,Y\}$ is a pair of separating vertices of the reduced Whitehead graph G of P, and C is a component of the separation obtained by deleting vertex X and vertex Y from G. Suppose that each "path" which leaves component C via vertex X returns to C via vertex Y and in addition each such "path" represents the same word W. Then the program will assert that an annulus A exists in D such that A "swallows" C and otherwise A follows the paths that leave C via X and return to C via Y.

We can apply the preceding results, by observing that since the paths in question only have their endpoints in C, we can collapse C to a point, and treat it as a "vertex" of D. Then consider these paths as tracing out realizations of the word cWC, and apply the above results. We see that these paths must all be parallel. Hence the required annulus exists.

Note that this yields an essential annulus A, in the underlying handlebody H, since there exists a vertex V of D, distinct from X and Y, such that V does not lie in C. Thus A is not parallel into the boundary of H, and A is, of course, incompressible in H.