The Word problem : A geometric approach

Haijo Schipper*

Abstract

In this paper we will give an algorithm which solves the "word-problem" for fundamental groups of compact oriented surfaces. That is given a word – a product of generators or their inverses – determine whether that word is in the same equivalence class as identity. Or in geometrical terms given a surface and its generators and a curve following the generators, can that curve be contracted to a point? The algorithm will work in time $O(gn)$ where g is the number of generators and n the length of the word.

1 Introduction

In the last few years a new branch of computational geometry *computational topology* has developed. It studies complexity of and algorithms on topological structures just as computational geometry studies geometrical structures. Topology, sometimes informally called *rubber sheet geometry*, is closely related to geometry; however, it deals more with relations than with realizations. That is one is i.e more interested in incidence relations of triangles than in the actual coordinates of the vertices.

Related work in computational topology can be found in Mehlhorn and Yap [MeYa 88], Vegter [Ve 89], Vegter and Yap [VeYa 90], Hershberger and Snoeyink [HeSn 91a, HeSn 91b], and Schipper [Sc 92]. In Snoeyink [Sn 90] a series of topological algorithms can be found.

The problem we will solve is the following : given a group $\mathcal{G}$, with generators $a_1, b_1, a_2, b_2, \ldots, a_g, b_g$ and a set of relations $\mathcal{R} = \{a_i\overline{a}_i = 1, b_i\overline{b}_i = 1, a_1b_1\overline{a}_1\overline{b}_1a_2b_2\overline{a}_2\overline{b}_2 \cdots a_gb_g\overline{a}_g\overline{b}_g\}(i = 1,\ldots,g)$ ($\overline{a}_i$ is the inverse of a_i) and a word W which is a product of n generators or their inverses. Now we want to determine whether W equals 1 (in $\mathcal{G}$) using a finite number of steps. The problem was solved by Dehn [De 12a, De 12b, Gr 60]. The problem can be solved in an algebraic process, Dehn's first solution, but also in a topological way. In this case $\mathcal{G}$ is the fundamental group of a surface $\mathcal{S}$. The generators of the group are the generators of the surface. The word W corresponds to a curve W_c on $\mathcal{S}$ along the generators. Now W equals 1 if and only if W_c can be contracted to a point. (See Stillwell [St 80]). So we have to determine whether or not W_c is contractable or not. We will use the following theorem :

Theorem 1.1 *A curve c on a 2-manifold $\mathcal{M}$ can be contracted to a point if and only if the lifted curve $\tilde{c}$ in the universal covering space $\mathcal{U}(\mathcal{M})$ is closed.*

and construct a part of the universal covering space to determine if the lifted curve of W_c is closed or not.

The rest of the paper is organized as follows : In section 2 we introduce the universal covering space and in section 3 we will show the algorithm.

*Department of Computing Science, Groningen University, P. O. Box 800, 9700 AV GRONINGEN, The Netherlands, Email : haijo@cs.rug.nl

2 Preliminaries

2.1 The canonical polygon

It is well-known that surfaces can be represented by simple polygons with labelled directed edges, each label occurring twice. See for instance Henle [He 79] and Stillwell [St 80]. Each pair of edges with the same labels is identified according to the direction of the edges. Furthermore each simple polygon with labelled directed edges such that each label occurs twice, represents a closed surface. See figure 1.

An orientable surface of genus $g > 0$ is represented by a k-gon ($k \geq 4g$). If we have a surface of genus $g > 0$ in a polygonal schema we say it is in normal form if the consecutive edges are labelled : $a_1b_1\overline{a_1}\overline{b_1}a_2b_2\overline{a_2}\overline{b_2}\ldots a_gb_g\overline{a_g}\overline{b_g}$. This polygon is called the **canonical polygon**. The canonical polygon of the sphere is a 2-gon labelled $a\overline{a}$. If a surface is represented by its canonical polygon all the vertices of the polygon are identified. This fact follows from the identification of the pairs of edges.

Given a canonical polygon $\mathcal{P}_{\mathcal{M}}$ we can obtain $\mathcal{M}$ by glueing the edges with identical labels – in the correct direction – together. After doing that the labels of $\mathcal{P}$ give rise to a set of curves on $\mathcal{M}$ all intersecting at one vertex. We will call this set of curves **canonical set of generators**. This set of generators corresponds of course with the generators of the group $\mathcal{G}$.

2.2 The universal covering space

Schwarz discovered in 1882 the universal covering space using the canonical polygon. The construction of the universal covering space is as follows : take infinitely many copies of $\mathcal{P}_{\mathcal{M}}$ and glue them together along the identified edges. What results is a – topological – regular tessellation of the plane with $4g$-gons. Note that the $4g$ corners of $\mathcal{P}_{\mathcal{M}}$ are identified. Since the edges of the canonical polygon represent a set of canonical generators of the surface, the labelling of edges around each vertex in the universal covering space is the same as the labelling of the generators around the basepoint on $\mathcal{M}$. See figure 1. Thus starting with a single copy of $\mathcal{P}_{\mathcal{M}}$, with its vertices on a circle C_0, there is a topologically unique way to complete the neighbourhood of each vertex with copies of $\mathcal{P}_{\mathcal{M}}$. Of course they are not congruent. All other vertices of those copies lie on a larger circle C_1. This process can be repeated ad infinitum. See figure 2.

The universal covering space of the sphere is the sphere itself and the universal covering space of a manifold of higher genus is homeomorphic to the plane.

We will denote the universal covering space of a manifold $\mathcal{M}$ by $\mathcal{U}(\mathcal{M})$.

Definitions 2.1 *We say that a vertex v in the universal covering space has* **distance** k *from the base polygon if it lies on the circle C_k. The distance of an edge from the base polygon is the minimum of the distances of its endpoints. A polygon has distance k from the base polygon if it is incident with a vertex at distance k, but not with a vertex at distance $k+1$. Edges of the universal covering space between vertices of different distances will be called* **spokes**, *edges between vertices at the same distance of the base polygon will be called* **arcs**. *Polygons at distance k incident with two vertices at distance $k-1$ will be called* **bridges**. *The* **fan** *of a vertex v at distance k consists of the polygons incident with v at distance $k+1$ which are not bridges.*

2.3 Lifting paths

Since $\mathcal{U}(\mathcal{M})$ is a covering space of $\mathcal{M}$, there is a map – the so called covering map – $\phi : \mathcal{U}(\mathcal{M}) \rightarrow \mathcal{M}$ from the simplexes of $\mathcal{U}(\mathcal{M})$ onto $\mathcal{M}$ with the following properties : (see also [St 80, 2.2.1])

1. ϕ preserves incidence relations. That is if x and y are incident in $\mathcal{U}(\mathcal{M})$ then $\phi(x)$ and $\phi(y)$ are incident in $\mathcal{M}$.

2. For each oriented edge e in $\mathcal{U}(\mathcal{M})$ we have $(\phi(e))^{-1} = \phi(e^{-1})$.

3. For each vertex v of $\mathcal{U}(\mathcal{M})$, ϕ maps the collection $\{e_i\}$ one-to-one to the collection of edges $\{e_i'\}$ where $e_1, e_2, \ldots$ are the oriented edges of $\mathcal{U}(\mathcal{M})$ with initial point v and $e_1', e_2', \ldots$ are the oriented edges of $\mathcal{M}$ with initial point $\phi(v)$. (This condition is called "local homeomorphism" – the neighbourhoods of corresponding vertices look alike.)

The latter condition implies that a path p in $\mathcal{M}$ is uniquely covered by a path $\tilde{p}$ in $\mathcal{U}(\mathcal{M})$ for a given start point in $\mathcal{U}(\mathcal{M})$. Let p be the sequence $v_0, e_1, v_1, e_2, v_2, \ldots$ and let $\tilde{v_0}$ be some point of $\mathcal{U}(\mathcal{M})$ which is mapped onto v_0 by ϕ. Then there is a unique edge $\tilde{e_1}$ incident with $\tilde{v_0}$ which is mapped onto e_1, giving us an unique point $\tilde{v_1}$ of $\mathcal{U}(\mathcal{M})$ which is mapped onto v_1. Continuing in this way we retrieve an unique path in $\mathcal{U}(\mathcal{M})$ which covers p once we have chosen a basepoint. This path $\tilde{p}$ is called the **lifted** path of p. Note that closed curves are not necessarily lifted into closed curves; in fact they are only lifted into a closed curve if they can be contracted to a point (theorem 1.1).

The proof of theorem 1.1 is quite simple and can be found in Stillwell [St 80, 6.1.1].

3 The algorithm

3.1 Overview of the algorithm

The general idea of the algorithm is as follows. Starting with a single copy of the canonical polygon we maintain a structure representing a part of the universal covering space. We pick a starting point and construct the lifted curve. This means we have to add other copies of the polygon to the structure. The adding is done such that the structure is a disc, the lifted curve lies inside the structure (not on the boundary) and there won't be a pair of polygons in the structure representing the same part of the covering space. If the lifted curve is completed we only have to test whether we the lifted curve is closed or not, that is did we finish in the starting point?

3.2 The structure

The structure we will maintain represents a part of the universal covering space (see section 2.2) and consists of several polygons linked together. Each polygon appearing in our structure is either a $4g$-gon (as in the universal covering space) or a polygon under construction. This will be a triangle which we will call an unfinished polygon. A edge is called a **boundary edge** if it bounds the structure. A polygon in the structure will have a pointer for each of its edges. If the edge is not a boundary edge, the pointer will point to the other polygon which is incident with the edge. In this way given a polygon and a edge (which corresponds to a certain generator) we can find in constant time the corresponding adjacent polygon (if it is in the structure). Each edge in the structure is given one of the following colours :

- white, if it is a edge between two $4g$-gons.
- black, if it is a edge between two polygons which are not both finished. It will turn out that arcs are never black.
- red, if it is both a spoke and a boundary edge.
- blue, if it is an arc and a boundary edge of a $4g$-gon.
- green, if it a boundary edge of an unfinished polygon and not a spoke. It will turn out that green edges correspond to a set of consecutive arcs in the universal covering space. One might call these edges "super-edges".

Vertices can have two colours. A vertex with distance k is purple if it is connected by a single spoke with a vertex at distance $k-1$ and yellow otherwise. See figure 2.

With each polygon its distance is associated. We also need a pointer pointing to the current polygon.

Our structure initially consists of one $4g$-gon which we will call our **base polygon**, with distance 0. All its edges will be blue.

3.3 Adding to the structure

There are just three ways to enlarge the structure :

1. Complete an unfished polygon.
2. Adding a bridge to the structure.
3. Adding a fan of unfinished polygons. A fan consists of $4g-3$ or $4g-4$ triangles separated by black edges.

Obviously each of the cases takes $O(g)$ time, and enlarges the structure with size $O(g)$. We also define the following procedure :

Complete neighbourhood of vertex v. v is already in the structure having distance k. Also if this procedure is called than the bridge(s) with distance k v is incident with are also in the structure. v has to incident with two bridges with distance $k+1$ so those are added to the structure, if they were not already. Finally the fan of v is added. Of course, given a pointer to v this procedure can be performed in $O(g)$ time.

During the algorithm we will maintain some invariants or which we will mention :

1. The structure represents a part of the universal covering space and is homeomorphic to a disc.
2. The neighbourhood of the current vertex is in the structure.
3. All bridges in the structure are finished.
4. All edges travelled are white.

3.4 The algorithm

Let s be the string of which we want to determine whether it reduces to 1 or not. If $s=\epsilon$ it does, so we are ready.

Otherwise we initialize the structure with a single $4g$-gon, all of its edges blue. Pick any vertex v and complete its neighbourhood. v will be the first current vertex.

While $s <> \epsilon$ we will do the following :

Let $s=\alpha s'$. Let e be the edge incident with v corresponding to α. Let v' be the other endpoint of e. Depending on the colours of e and v' we will do the following :

1. e is black. Then v' has to be purple. First we will finish those polygons incident with e which are not finished. This will be at least one polygon and (of course) not more than two. Then the neighbourhood of v' is completed.
2. e is white and v' is yellow. Then the neighbourhood of v' is finished. See figure 3.
3. e is white and v' is purple. Then v' is the endpoint of a spoke e' whose other endpoint w has lesser distance than v. Depending on the colour of e' we will do :
   (a) e' is white. In this case we complete the neighbourhood of v'.
   (b) e' is black. Then e' is incident with one unfinished polygon. These polygon will be finished after which the neighbourhood of v' will be completed. See figure 4.
   (c) e' is red. In this case e' is the last edge of a chain of red edges. (Possibly a chain of one edge.). Let the chain be $e_1, e_2, \ldots, e_k = e'$, separated by the vertices $v_0, v_1, \ldots, v_k = v'$. v_0 has smallest distance and is starting point of the chain. Every next vertex has a distance of one more. For $v_i = v_0, \ldots, v_k$ we will complete the neighbourhood of v_i. See figure 5.

Finally v' will be the new current vertex and $s := s'$.

If we processed the whole string s we have to check whether we ended in the starting point or not to determine if the lifted curve is closed or not.

3.5 Complexity

Theorem 3.1 *Given a group $\mathcal{G}$ generated by the set $\{a_1, b_1, a_2, b_2, \ldots, a_g, b_g\}$ and the set of relations $\mathcal{R} = \{a_1\overline{a}_1 = b_1\overline{b}_1 = a_2\overline{a}_2 = b_2\overline{b}_2 = \cdots = a_g\overline{a}_g = b_g\overline{b}_g = a_1b_1\overline{a}_1\overline{b}_1a_2b_2\overline{a}_2\overline{b}_2\ldots a_gb_g\overline{a}_g\overline{b}_g = 1\}$ and an expression $s = \alpha_1\alpha_2\alpha_3\ldots\alpha_n$ in $\mathcal{G}$. The question whether s equals 1 can be answered in time $O(gn)$ using $O(gn)$ storage.*

Proof We will use the algorithm of section 3.4. Its correctness can easily be proven using the invariants. For details we refer to the full paper.

In the cases 1), 2), 3a) and 3b) it is easy to see the algorithm needs $O(g)$ time and adds $O(g)$ storage. This leaves us one difficult case, case 3c. Suppose we have a chain of k red edges. Then we have to complete $k+1$ neighbourhoods, taking $O(kg)$ time. Notice however that each red edge is a spoke and a boundary of a bridge. Also each bridge can have at most one red edge. Also to reach such a red edge $\Omega(g)$ blue edges of that bridge have to be made white. Also on each completion of a neighbourhood at most one blue edge per polygon is made blue. So we can charge the costs per red edge over $\Omega(g)$ other updates. Since non-red edges never become red again, each blue edge is charged once, giving an amortized update time in case 3c) of $O(1)$.

q.e.d.

4 Conclusions

In this paper we gave an algorithm to determine whether a word in a fundamental group of a 2-manifold equals 1 or not. Of course the algorithm can be used to determine if two words W_1 and W_2 belong to the same equivalence class. If so then $W_1(W_2)^{-1} = 1$.

The use of the universal covering space is interesting and can be used for related problems such as the *contractibility problem* (see Schipper [Sc 92]) or the determination of shortest curve homotopic to a given curve. This can be done by constructing a sufficient large part of the universal covering space and using Dijkstra's algorithm. However, it is not clear if it can be done in polynomial time.

References

[De 12a] Dehn, M. : "Über unendliche diskontinuierliche Gruppen." *Math. Ann. Vol. 71*, 1912, pp. 116 - 114.

[De 12b] Dehn, M. : "Transformation der Kurven auf zweiseitige Flächen." *Math. Ann. Vol. 72*, 1912, pp. 413 - 421.

[Gr 60] Greenlinger, M. : "Dehn's algorithm for the word problem." *Communications on Pure and Applied Mathematics, Vol. XIII*, 1960, pp. 67 - 83.

[He 79] Henle, M. : *A combinatorial introduction to topology*. San Francisco : W. H. Freeman and Company, 1979.

[HeSn 91a] Hersberger, J. and Snoeyink, J. : "Around and around :computing the shortest loop." *Proceedings of the third Canadian Conference on Computational Geometry*. 1991, pp. 157 - 161.

[HeSn 91b] Hersberger, J. and Snoeyink, J. : "Computing minimum length paths of a given homotopy class." *"Proc. of the 2nd Workshop on Algorithms and Data Structures (WADS '91). August 1991."* In : *Lecture Notes in Computer Science, Vol. 519*. Berlin : Springer-Verlag, 1991, pp. 331 - 342.

[MeYa 88] Mehlhorn, K. and Yap, C. K. : "Constructuve hopf's theorem : or how to untangle closed planar curves." *Proceedings ICALP, Springer Lecture Notes in Computer Science, Vol. 317* 1988, pp. 410 - 423.

[Sn 90] Snoeyink, J. S. : "Topological Approaches in Computational Geometry." Stanford University : *Report No. STAN-CS-90-1331.*, 1990.

[Sc 92] Schipper, H. : "Determining contractibility of curves." *Proceedings of the 8th ACM Symposium on Computational Geometry*, Berlin, 1992. To appear.

[St 80] Stillwell, J. : "Classical Topology and Combinatorial Group Theory." In : Halmos, P. R., Gehring, F. W. and Moore, C. C. (Eds.) : *Graduate Texts in Mathematics, Vol. 72*. New York, Heidelberg, Berlin : Springer-Verlag, 1980.

[Ve 89] Vegter, G. : "Kink-free deformations of polygons" *Proceedings of the 5th ACM Symposium on Computational Geometry*, 1989, pp. 61 - 68.

[VeYa 90] Vegter, G. and Yap, C. K. : "Computational complexity of combinatorial surfaces." *Proceedings of the 6th ACM Symposium on Computational Geometry*, 1990, pp. 102 - 111.

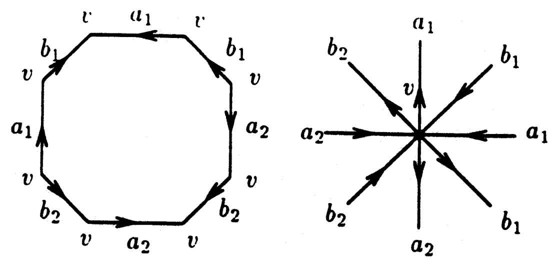

Figure 1: The canonical polygon of the double torus and the neighbourhood of a vertex in the universal covering space.

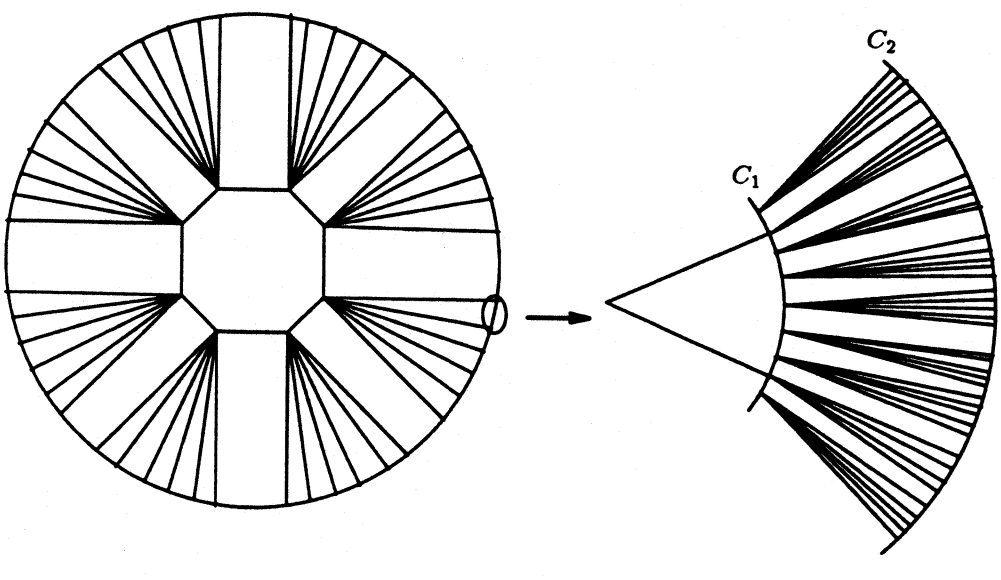

Figure 2: The universal covering space of the double torus.

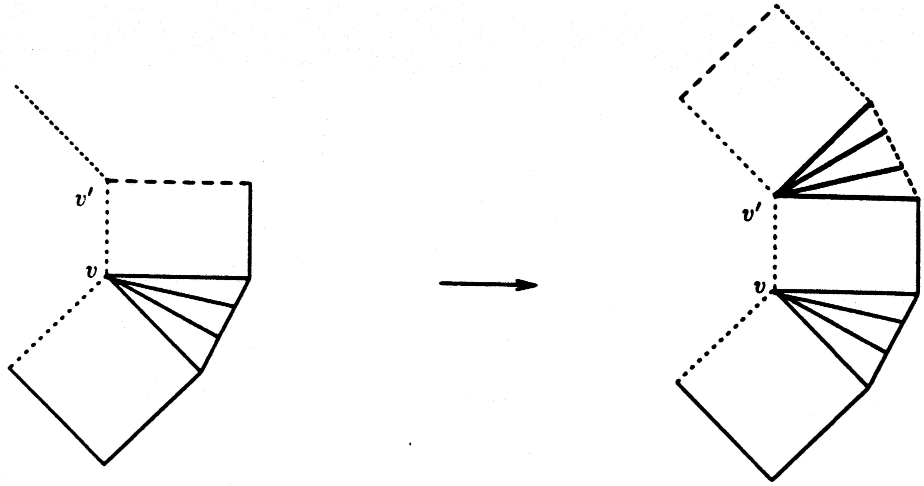

Figure 3: Travelling a white edge, towards a yellow vertex.

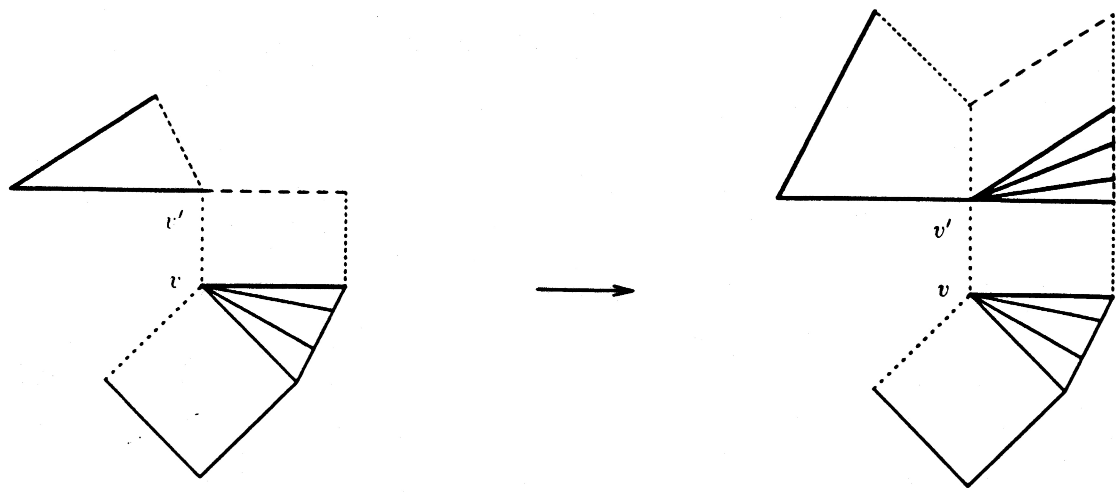

Figure 4: Travelling a white edge, towards a purple vertex I.

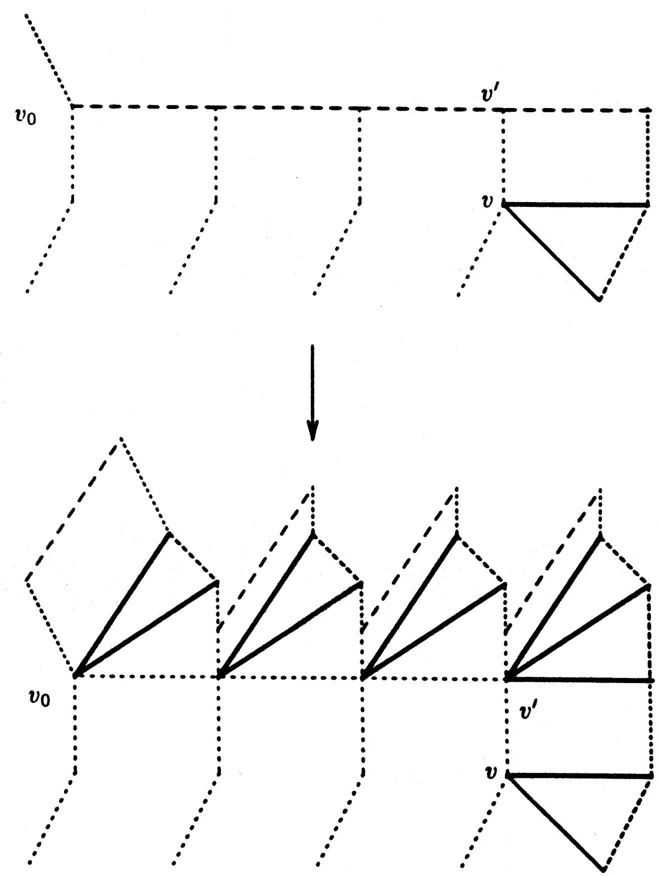

Figure 5: Travelling a white edge, towards a purple vertex II.