The infinite possibilities that each day holds should stagger the mind. The sheer number of experiences I could have is uncountable, breathtaking, and I'm sitting here refreshing my inbox. We live trapped in loops, reliving a few days over and over, and we envision only a few paths laid out ahead of us. We see the same things each day, respond the same way, we think the same thoughts, every day a slight variation on the last, every moment smoothly following the gentle curves of societal norms. We act like if we just get through today, tomorrow our dreams will come back to us.

— Randall Munroe, "Dreams", http://xkcd.com/137/

7 Homotopy of Curves on Surfaces

In this lecture, we'll see efficient algorithms to determine whether a given loop ℓ on a given 2-manifold Σ is contractible. (We previously considered this problem for polygons with holes in the plane.) To make the problem concrete, we assume that Σ is represented by a polygonal schema Π with complexity n, and ℓ is a closed walk of length k in the induced embedded graph G or its dual G^*. After developing some necessary mathematical tools, I will describe a classical algorithm of Dehn [3] for the special case when G is a system of loops, which runs in $O(k)$ time. Dehn's algorithm and its later generalizations are the foundation of *geometric group theory*. Next I will describe a simple extension of Dehn's algorithm to more general schemata that runs in $O(gn + g^2k)$ time for surface of genus g. Finally, following a result of Dey and Guha [4], I will show how to improve the running time to $O(n + k)$.

Let Σ be a fixed, compact, connected 2-manifold. For most of the lecture, for reasons that will become clear, I will assume that $\chi(\Sigma) < 0$; thus, Σ is *not* the sphere, the projective plane, the torus, or the Klein bottle. I will briefly reconsider these surfaces at the end of the lecture.

7.1 The Fundamental Group

Recall the following definitions from Lecture 2. A ***path*** in Σ is a continuous function $\pi\colon [0,1] \to \Sigma$. The ***concatenation*** $\boldsymbol{\pi \cdot \sigma}$ of two paths π and σ with $\pi(1) = \sigma(0)$ is the path

$$(\pi \cdot \pi')(t) := \begin{cases} \pi(2t) & \text{if } t \le 1/2, \\ \sigma(2t-1) & \text{if } t \ge 1/2. \end{cases}$$

The ***reversal*** of a path π is the path $\overline{\pi}(t) := \pi(1-t)$. We easily observe that $\overline{\pi \cdot \sigma} = \overline{\sigma} \cdot \overline{\pi}$.

A ***loop*** is a path π whose endpoints coincide; this common endpoint is the loop's ***basepoint***. The concatenation of two loops is a loop, and the reversal of a loop is a loop.

A ***(path) homotopy*** between paths π and π' is a continuous function $h\colon [0,1] \times [0,1] \to \Sigma$ such that $h(0,t) = \pi(t)$ and $h(1,t) = \pi'(t)$ for all t, and $h(s,0) = \pi(0) = \pi'(0)$ and $h(s,1) = \pi(1) = \pi'(1)$ for all $s \in [0,1]$. Two paths π and π' are ***homotopic***, written $\boldsymbol{\pi \simeq \pi'}$, if there is a homotopy between them. A loop is ***contractible*** if it is homotopic to a constant path at its basepoint.

Homotopy is an equivalence relation; we write $[\boldsymbol{\pi}]$ to denote the ***homotopy class*** of a path π. In particular, we write $[x]$ to denote the homotopy class of contractible loops with basepoint x. We can extend concatenation and reversal to homotopy classes by defining $\overline{[\pi]} = [\overline{\pi}]$ and $[\pi] \cdot [\sigma] = [\pi \cdot \sigma]$. The following lemma implies that these operations are well-defined; the proof is an easy exercise.

Lemma 7.1. *The following invariants hold for all paths $\pi, \pi', \sigma, \sigma'$ in Σ:*

(a) If $\pi \simeq \sigma$, then $\overline{\pi} \simeq \overline{\sigma}$.

(b) If $\pi \simeq \pi'$ and $\sigma \simeq \sigma'$, then $\pi \cdot \sigma \simeq \pi' \cdot \sigma'$.

For any point $x \in \Sigma$, the set of homotopy classes of loops based at x define a *group*, with concatenation as the group operation, $[x]$ as the identity element, and reversal as the inverse operation. This group is called the ***fundamental group of Σ*** (based at x) and denoted $\boldsymbol{\pi_1(\Sigma, x)}$. To verify the group structure, we must check that concatenation is associative and that inverses behave correctly; again, the proof is an easy exercise.

Lemma 7.2. *The following invariants hold for all loops ℓ, ℓ', and ℓ'' in Σ with basepoint x.*
(a) $[\ell] \cdot [\bar{\ell}] = [\bar{\ell}] \cdot [\ell] = [x]$
(b) $[x] \cdot [\ell] = [\ell] \cdot [x] = [\ell]$
(c) $([\ell] \cdot [\ell']) \cdot [\ell''] = [\ell] \cdot ([\ell'] \cdot [\ell''])$

The fundamental group is *not* necessarily abelian; $\ell \cdot \ell'$ and $\ell' \cdot \ell$ may not be homotopic.

7.2 Group Presentations

We can derive an explicit description of the fundamental group of any surface from any system of loops for that surface. The description is an example of a standard ***presentation*** of a group; before considering the fundamental group, we need to define group presentations in general. A presentation is a pair $\langle S \mid R \rangle$ composed of two sets S and R, with the following properties:

The set S is a set of group elements, called ***generators***, with the property that every element of the group can be written as the product of elements of S and their inverses. A ***word*** in S is a string over the alphabet $S \cup \bar{S}$, where $\bar{S}$ is the set of inverses of generators in S. Each word represents an element of the group; specifically, the empty string represents the group's identity element, and concatenation represents the group operation. The inverse $\bar{w}$ of a word w is defined by writing W in reverse and then inverting every character; thus, the inverse of $\mathtt{abc\bar{a}}$ is $\mathtt{a\bar{c}\bar{b}\bar{a}}$.

The set R is a set of words in S, called ***relators***, each of which is defined to be equivalent to the empty word. To satisfy the group axioms, any words of the form $x\bar{x}$ or $\bar{x}x$ are *defined* to be equivalent to the empty word. In general, two words w and x are considered equivalent if the word $w \cdot \bar{x}$ is equivalent to the empty word. Finally, $\langle S \mid R \rangle$ represents the group of equivalence classes of words, with concatenation as the group operation.[1]

For the special case of a ***free*** group, where the relator set is empty, we normally use the simpler notation $\langle S \rangle$. When the generator and relator sets are given explicitly, we normally write their elements without the set-braces.

Here are a few examples of group presentations:

- $\langle x \rangle$ is the group of integers $\mathbb{Z}$. The generator x could represent either 1 or -1.
- $\langle x \mid xx \rangle$ is the two-element group $\mathbb{Z}_2$. If we describe $\mathbb{Z}_2$ as the set $\{0, 1\}$ under mod-2 addition, the generator x represents the integer 1. Alternatively, if we describe $\mathbb{Z}_2$ as the set $\{\text{TRUE}, \text{FALSE}\}$ under exclusive-or, the generator x represents the element TRUE.
- $\langle x, y \mid xy\bar{x}\bar{y} \rangle$ is the planar integer lattice $\mathbb{Z}^2$. The relator $xy\bar{x}\bar{y}$ implies that $xy = yx$; thus, the group is abelian. To see the correspondence with the usual description of $\mathbb{Z}^2$, think of x and y as the vectors $(1, 0)$ and $(0, 1)$.
- Similarly, $\langle x, y, z \mid xy\bar{x}\bar{y}, xz\bar{x}\bar{z}, yz\bar{y}\bar{z} \rangle$ is the three-dimensional integer lattice $\mathbb{Z}^3$.

[1]The group $\langle S \mid R \rangle$ is usually defined as $\langle S \rangle / N$, where N is the largest normal subgroup of $\langle S \rangle$ containing the elements of R. The two definitions are equivalent.

Every group has infinitely many presentations. For example, $\langle z \mid zzzzz \rangle$ and $\langle x, y \mid xxy, x\bar{y}\bar{y} \rangle$ are both presentations of the group $\mathbb{Z}_5$. (Do you see why?) A group is ***finitely generated*** if it has a presentation with a finite number of generators, and ***finitely presented*** if it has a presentation with a finite number of generators and relators.

7.3 Presentations for the Fundamental Group

Let L be a system of loops for Σ, and let x be its only vertex. We give each edge in L an arbitrary orientation, so it really is a loop.

Lemma 7.3. *Let L be a system of loops in Σ with basepoint x. Any loop in Σ with basepoint x is homotopic to a concatenation of loops in L and their reversals.*

Thus, in every homotopy class in $\pi_1(\Sigma, x)$, there is a loop whose image lies entirely in L. The homotopy class of such a loop ℓ can be described by listing the labels and orientations of the edges in L that ℓ traverses. Whenever we traverse a loop along its orientation, we record its label; when we traverse an loop against its orientation, we record its label with a bar over it.

Two loops with the same traversal sequence are clearly homotopic, but homotopic loops can have different traversal sequences. Specifically, any ***spur*** of the form $\mathsf{a}\bar{\mathsf{a}}$ or $\bar{\mathsf{a}}\mathsf{a}$ is obviously contractible, and any loop that completely traverses the boundary of the unique face of L is contractible. Thus, any two traversal sequences that differ only by inserting or deleting spurs and facial walks are homotopic.

The converse implication is true as well—the traversal sequences of two homotopic paths differ only by inserting or deleting spurs and facial walks. (The proof is another easy exercise. No, really!)

Lemma 7.4. *Let L be a system of loops with basepoint x in a 2-manifold Σ, and let r be the traversal sequence of the single face of L. Then $\langle L \mid r \rangle$ is a presentation of the fundamental group $\pi_1(\Sigma, x)$.*

For example, if the polygonal schema of L has signature $(\mathsf{ab}\bar{\mathsf{a}}\bar{\mathsf{b}}\mathsf{cd}\bar{\mathsf{c}}\bar{\mathsf{d}})$, we obtain the presentation $\pi_1(\Sigma, x) = \langle \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d} \mid \mathsf{ab}\bar{\mathsf{a}}\bar{\mathsf{b}}\mathsf{cd}\bar{\mathsf{c}}\bar{\mathsf{d}} \rangle$.

7.4 Universal Cover of a System of Loops

Recall that the ***universal covering space*** $\tilde{\Sigma}$ of a surface Σ is the unique simply-connected covering space of Σ. Our fixed basepoint $x \in \Sigma$ is the projection of one or more points in $\tilde{\Sigma}$, called ***lifts*** of x. Let $\tilde{x}$ denote an arbitrary lift of x. Any path $\pi\colon [0,1] \to \Sigma$ with $\pi(0) = x$ is the projection of a unique path $\tilde{\pi}\colon [0,1] \to \tilde{\Sigma}$ with $\tilde{\pi}(0) = \tilde{x}$. The following lemma follows from tedious definition-chasing:[2]

Lemma 7.5. *A loop ℓ is contractible in Σ if and only if its lift $\tilde{\ell}$ is a loop in $\tilde{\Sigma}$.*

Let L denote a fixed system of loops in Σ. Let $\bar{g} = 2 - \chi(\Sigma)$ denote the number of loops in L. (The number $\bar{g}$ is called the ***Euler genus*** of Σ; it is twice the standard genus when Σ is orientable, and equal to the standard genus when Σ is not orientable.) Euler's formula implies that $\bar{g} \geq 3$. The system of loops L lifts to an infinite graph $\tilde{L}$ in the universal covering space $\tilde{\Sigma}$. Each face of $\tilde{L}$ is a lift of the unique face of L, and each vertex of $\tilde{L}$ is a lift of the unique vertex of L. Thus, $\tilde{L}$ is a tiling of $\tilde{\Sigma}$ by $2\bar{g}$-gons meeting at vertices of degree $2\bar{g}$. The only regular ***geometric*** structure that fits $\tilde{L}$ is a regular tiling of the ***hyperbolic plane*** by regular r-gons with interior angles $\pi/\bar{g}$.

We now have three different descriptions of the contractibility problem, with increasing generality:

[2] Recall that the universal cover can be *defined* as the set of homotopy classes of paths in Σ that start at x. In light of this definition, we can take $\tilde{x}$ to be the homotopy class of the constant path at x, and define $\tilde{\pi}$ by setting $\tilde{\pi}(t) = [\pi|_{[0,t]}]$ for all t. In particular, $\tilde{\pi}(1) = [\pi]$. Thus, π is a contractible loop if and only if $\tilde{\pi}(1) = \tilde{x}$.

- Given a loop ℓ in a system of loops L, is ℓ contractible?
- Given a path $\tilde{\ell}$ in a regular hyperbolic tiling $\tilde{L}$, is $\tilde{\ell}$ a loop?
- Given a word $X(\ell)$ in the generators of $\pi_1(\Sigma)$, is $X(\ell)$ equivalent to the identity element?

The third formulation is called the ***word problem*** for finitely presented groups. Dehn [1, 2] proposed the word problem as a target or algorithmic study, along with two other problems called the ***conjugacy problem*** (Given two words x and y, is there a group element z such that $xz = zy$?) and the ***isomorphism problem*** (Given two group presentations, do they describe isomorphic groups?). Soon after he proposed these problems, Dehn [3] described algorithms for the word and conjugacy problems for surface fundamental groups, using *combinatorial* properties of the associated hyperbolic tilings. Dehn's algorithms are the seeds of combinatorial and geometric group theory.

Dehn actually conjectured that "Solving the word problem for all groups may be as impossible as solving all mathematical problems." This was a remarkable prediction; it would be another 40 years before Turing provided the formalism to even state this conjecture crisply. All three of Dehn's problems are undecidable in general.

7.5 Dehn's Algorithm

Dehn solved the word problem for surface fundamental groups by recasting it as a problem in hyperbolic geometry. Dehn's key insight is the following combinatorial lemma:

Lemma 7.6. *Let $\tilde{L}$ be a tiling of the plane by $2\bar{g}$-gons meeting at vertices of degree n, for some integer $\bar{g} \geq 2$. Let $\tilde{\ell}$ be a nonconstant simple loop in $\tilde{L}$. Then $\tilde{\ell}$ contains $2\bar{g} - 2$ consecutive edges of some face of $\tilde{L}$.*

Proof: We construct a nested series of disks $D_1 \subset D_2 \subset D_3 \subset \cdots$ in the plane, each of which is the union of faces of $\tilde{L}$, as follows. The innermost disk D_1 is an single face in the interior of $\tilde{\ell}$, and for all $i \geq 2$, the disk D_i is the union of all faces that have at least one vertex in D_{i-1}. For each $i \geq 1$, let C_i denote the boundary of D_i. Finally, let $A_1 = C_1$, and for all $i \geq 2$, let $A_i = D_i \setminus D_{i-1}$.

Suppose inductively that every vertex of C_{i-1} has degree at most 3 in D_{i-1}; this claim is trivial when $i = 2$. Then each vertex of C_{i-1} is adjacent to at least $2\bar{g} - 2$ faces in A_i. Thus, each face of A_i contains at most one edge of C_{i-1} (because $2\bar{g} \geq 4$). It follows that every face of A_i shares edges with at most three other faces in D_i; moreover, these edges are consecutive on the boundary of each face. In other words, each face of A_i has at least $2\bar{g} - 3$ consecutive edges on C_i. We conclude easily that every vertex of C_i has degree at most 3 in D_i, so our inductive hypothesis is true for all i.

In particular, for all i, every face of A_i has at least $2\bar{g} - 3$ consecutive edges on C_i.

Suppose $\tilde{\ell}$ intersects C_i (and thus A_i) but not C_{i+1}. Any maximal subpath in A_i must consist of an edge from C_{i-1} to C_i, a subpath of C_i, and an edge back to C_{i-1}. This subpath contains at least $2\bar{g} - 2$ consecutive edges of at least one face of A_i. □

We can restate this lemma in terms of systems of loops or group presentations as follows:

Lemma 7.6. *Let L be a system of loops on a surface with Euler genus $\bar{g} \geq 2$. Let R be the traversal sequence of the boundary of the unique face of L, and let ℓ be a non-constant contractible loop in L. Then the traversal sequence $X(\ell)$ contains a subword of length $2\bar{g} - 2$ that is also a subword of a cyclic shift of R or $\overline{R}$.*

Lemma 7.6. *Let $\langle L \mid R\rangle$ be a one-relator presentation of $\pi_1(\Sigma)$, for some surface Σ with Euler genus $\bar{g} \geq 2$. Let X be an nonempty trivial word in $(L \cup \bar{L})^*$. Then X contains a subword of length $2\bar{g}-2$ that is also a subword of a cyclic shift of R or $\overline{R}$.*

To simplify the presentation, let's use the term ***subrelator*** to mean a subword of a cyclic shift of either R or $\bar{R}$. We say that a subrelator is ***large*** if its length is at least $2\bar{g}-2$.

Dehn used this lemma as the basis for the following algorithm. Scan through the traversal sequence one character at a time. At each step, check whether the last two characters define a spur, or if the last $2\bar{g}-2$ characters are a subrelator. In the former case, remove the last two characters; in the latter case, replace a maximal subrelator with its complement in $O(\bar{g})$ time. The input loop is contractible if and only if this algorithm transforms its traversal sequence into the empty word.

Dehn did not analyze the running time of his algorithm—remember, he did this in 1912—but a naïve $O(\bar{g}^2 x)$ time bound is straightforward, where x is the length of ℓ (or its traversal sequence). It takes at most $O(\bar{g}^2)$ time to check whether the current prefix ends with a large subrelator, plus $O(\bar{g})$ time to reduce the word if necessary. More careful analysis reduces the running time to $O(\bar{g}x)$. There is at most one cyclic shift of the relator (or its inverse) that ends with any pair of characters; otherwise, the system of loops would have more than one vertex. Thus, we really only need $O(\bar{g})$ time to check if a reduction is possible.

In fact, the last factor of r can also be removed. There are several ways to improve the search time. One approach, reminiscent of the Knuth-Morris-Pratt string-matching algorithm, is to construct a finite-state machine that accepts a string if and only if it has a large subrelator as a suffix. A suitable finite-state machine with $O(\bar{g}^2)$ states can be constructed in $O(\bar{g}^2)$ time.[3] I'll describe another, arguably simpler method later in this lecture.

Finally, we can amortize away the reduction time, by charging each reduction to the resulting decrease in length. Canceling a spur takes $O(1)$ time and decreases the length of the word by $\Omega(1)$, and teplacing most of a relator with its complement takes $O(\bar{g})$ time and decreases the length by at least $2\bar{g}-4 = \Omega(\bar{g})$. Since the total decrease in length over the entire algorithm is trivially at most x, the total reduction time is also at most $O(x)$.

Theorem 7.7. *Let L be a system of loops on a surface Σ with Euler genus $\bar{g} \geq 3$. There is an algorithm to determine whether a given closed walk ℓ in L of length x is contractible in Σ in $O(\bar{g}+x)$ time.*

7.6 . . . in More General Graphs

So far we have considered only closed walks in a system of loops; we need to do a bit more work to apply Dehn's algorithm to loops in more complicated embedded graphs. Let G be an arbitrary cellularly embedded graph in Σ, and let ℓ be a closed walk in G. We want to determine whether ℓ is contractible in Σ. Without loss of generality, we will assume that every face of G is a triangle.

Lemma 7.8. *Let G be cellularly embedded graph in a surface Σ, and let x be a vertex of G. Any loop in Σ with basepoint x is homotopic to a closed walk in G.*

As in the proof of the surface classification theorem, we begin by constructing a tree-cotree decomposition (T, C, X). Recall that in such a decomposition, T is a spanning tree; C is spanning cotree (dual to a spanning tree of $G^* \setminus T^*$); T and C are disjoint; and $X = G \setminus (T \cup C)$. Contracting every edge in T and deleting every edge in C transforms X into a system of loops; thus, X contains exactly $\bar{g}$ edges.

[3] Even this $O(\bar{g}^2)$ term can be reduced to $O(\bar{g})$ by representing the finite state machine implicitly. Details are left as an entertaining exercise for the reader.

Now let ℓ be an arbitrary loop in G. We can determine whether ℓ is contractible by first transforming it into a homotopic loop ℓ'' in the system of loops $L = G \,/\, T \setminus C$, and then applying Dehn's algorithm to ℓ''. In principle, we can compute ℓ'' as follows.

The subgraph $G \setminus C = T \cup L$ is a cellularly embedded graph with a single face, and thus can be represented by a polygonal schema with one polygon P; the edges of C define an internal triangulation of this polygon. We first define an intermediate loop ℓ' by replacing each edge $e \in C$ with a walk π_e around the boundary of P between the same corners. (It doesn't actually matter which walk we choose, but for concreteness, think of π_e as a shortest path on the boundary of P.) Finally, we set $\ell'' = \ell' \,/\, T$.

Unfortunately, replacing a single edge of C might require a path of length $\Theta(n)$, so explicitly constructing ℓ'' might require $\Theta(nk)$ time and space in the worst case. So instead, we compute the traversal sequence of ℓ'' directly from ℓ, after the following further preprocessing of the graph.

Choose an arbitrary face as the root of C^*, and orient all edges of C^* away from this root. By orienting C^*, we assign a parent edge and two child edges to every face of G except the root face. Moreover, each edge $e \in C$ is a parent of exactly one face; call the other two edges of that face *left*(e) and *right*(e), so that $e \simeq \mathit{left}(e) \cdot \mathit{right}(e)$.

We now recursively define the signature $X(e)$ of an edge e as follows: (1) If $e \in T$, then $X(e)$ is the empty string. (2) If $e \in L$, then $X(e)$ is the label of e. (3) If $e \in C$, then $X(e) = X(\mathit{left}(e)) \cdot X(\mathit{right}(e))$. We can compute the signature of every edge of C in $O(\bar{g}n)$ time by depth-first search.

Given any loop ℓ, we can compute its signature (which we called $X(\ell'')$ earlier) in $O(\bar{g}k)$ time by concatenating the signatures of its edges. A direct implementation of Dehn's algorithm now checks whether this signature represents a contractible loop in $O(\bar{g}^2 + \bar{g}k)$ time. The total running time is $O(\bar{g}n + \bar{g}k)$. (The $\bar{g}^2$ term vanishes because $\bar{g} \le n$.)

Theorem 7.9. *Let G be a cellular graph of complexity n on a surface Σ with Euler genus $\bar{g} \ge 3$. There is an algorithm to determine whether a given closed walk in G of length k is contractible in Σ in $O(\bar{g}n + \bar{g}k)$ time.*

7.7 ...in Linear Time

If $\bar{g} = O(1)$, the algorithm we just described already runs in $O(n+k)$ time, but if $\bar{g}$ is large compared to n or k, more work is required to obtain a linear-time algorithm. The final improvement, suggested by Dey and Guha [4], is to use a *compressed* representation of traversal sequences that is still amenable to Dehn's approach. Dey and Guha actually describe an algorithm for the following harder problem: Given two cycles, are they freely homotopic? Specializing their approach to test only contractibility allows us to simplify their algorithm considerably.

We will represent subrelators by ordered pairs of integer (i, j), where $0 \le i, j < 2\bar{g}$, as follows:

$$(i,j) := \begin{cases} R[i+1\,..\,j] & \text{if } i \le j,\\ R[i+1\,..\,2\bar{g}] \cdot R[0\,..\,j] & \text{if } i \ge j. \end{cases}$$

For example, if $R = \mathtt{abc\bar{a}d\bar{b}\bar{c}\bar{d}}$, then $(0,4) = \mathtt{abc\bar{a}}$ and $(7,2) = \mathtt{\bar{d}ab}$. Observe that $(i, j) \cdot (j, i)$ is always a cyclic shift of the relator R, and thus equal to the trivial group element. Thus, each pair (i, j) actually represents *two* subrelators, but only a single group element. More generally, we have $(i, j) \cdot (j, k) = (i, k)$ for any indices i, j, and k.

Any subrelator of length at least 2 is represented by a *unique* pair (i, j). For individual generators, however, our representation is ambiguous; any generator is represented by exactly two pairs of the form $(i, i \pm 1)$. For example, if $R = \mathtt{abc\bar{a}d\bar{b}\bar{c}\bar{d}}$, the pairs $(0,1)$ and $(4,3)$ both represent the generator $\mathtt{a}$. To avoid this ambiguity, we represent individual generators explicitly. Finally, the empty subrelator is represented by all $2\bar{g}$ pairs of the form (i,i); to avoid this ambiguity, we represent the empty subrelator with a special symbol $\varnothing$.

For every edge in C, the associated traversal sequence is a subrelator, and thus can be represented by a pair, a single generator, or the identity symbol $\varnothing$. Moreover, if the concatenation of two subrelators is a new subrelator, we can compute that new subrelator in $O(1)$ time. Thus, we can compute the subrelator associated with each edge in C in $O(n)$ time with a straightforward depth-first search of C^*.

Finally, we compute the ***signature*** of any loop ℓ by concatenating the subrelators associated with the edges traversed by ℓ, and then reduce the signature as much as possible. Our target representation for an arbitrary word is a *minimal* sequence of subrelators, where no adjacent pair of subrelators can be merged. The loop ℓ is contractible if and only if its reduced signature is empty. In general, the reduced signature of a path is *not* unique. For example, if $R = \mathtt{abc\overline{a}d\overline{bcd}}$, then the word $\mathtt{abcb\overline{d}a}$ has four signatures:

$$\begin{aligned}
(0,2)\cdot(7,5)\cdot(7,1) &= \mathtt{ab}\cdot\mathtt{cb}\cdot\mathtt{\overline{d}a}\\
=(0,2)\cdot(7,4)\cdot\mathtt{a} &= \mathtt{ab}\cdot\mathtt{cb\overline{d}}\cdot\mathtt{a}\\
=(0,3)\cdot(6,4)\cdot\mathtt{a} &= \mathtt{abc}\cdot\mathtt{b\overline{d}}\cdot\mathtt{a}\\
=(0,3)\cdot\mathtt{b}\cdot(7,1) &= \mathtt{abc}\cdot\mathtt{b}\cdot\mathtt{\overline{d}a}
\end{aligned}$$

Despite this ambiguity, we can reduce any signature of length k in $O(k)$ time, as follows. We scan through the subrelators one at a time. At each step, we repeatedly check whether the last subrelator is trivial, the last two subrelators define a spur, or the last two subrelators can be concatenated, and simplify the signature in each case. We continue to the next subrelator only when the current prefix cannot be further simplified.

The figure below shows an example of the reduction algorithm in action for the relator $\mathtt{abc\overline{a}d\overline{bcd}}$; thus, the underlying surface is the double-torus. The input is the signature $(6,1)\cdot(6,2)\cdot\mathtt{\overline{b}}\cdot(1,6)\cdot(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$ of the trivial traversal sequence $\mathtt{\overline{cd}a}\cdot\mathtt{\overline{cd}ab}\cdot\mathtt{\overline{b}}\cdot\mathtt{bc\overline{a}d\overline{b}}\cdot\mathtt{\overline{cb}}\cdot\mathtt{\overline{cd}abc\overline{a}}\cdot\mathtt{dcb}\cdot\mathtt{c\overline{a}d\overline{b}}$.

$\underline{(6,1)}$	$\mathtt{\overline{cd}a}$	$(6,2)\cdot\mathtt{\overline{b}}\cdot(1,6)\cdot(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*get next*
$\underline{(6,1)\cdot(6,2)}$	$\mathtt{\overline{cd}a}\cdot\mathtt{\overline{cd}ab}$	$\mathtt{\overline{b}}\cdot(1,6)\cdot(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*get next*
$(6,1)\cdot\underline{(6,2)\cdot\mathtt{\overline{b}}}$	$\mathtt{\overline{cd}ab}\cdot\mathtt{\overline{b}}$	$(1,6)\cdot(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*cancel spur*
$\underline{(6,1)\cdot(6,1)}$	$\mathtt{\overline{cd}a}\cdot\mathtt{\overline{cd}a}$	$(1,6)\cdot(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*get next*
$(6,1)\cdot\underline{(6,1)\cdot(1,6)}$	$\mathtt{\overline{cd}a}\cdot\mathtt{bc\overline{a}d\overline{b}}$	$(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*concatenate*
$(6,1)\cdot\underline{(6,6)}$	$\mathtt{\overline{cd}abc\overline{a}d\overline{b}}$	$(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*delete*
$\underline{(6,1)}$	$\mathtt{\overline{cd}a}$	$(3,1)\cdot(6,4)\cdot(0,5)\cdot(2,6)$	*get next*
$\underline{(6,1)\cdot(3,1)}$	$\mathtt{\overline{cd}a}\cdot\mathtt{\overline{cb}}$	$(6,4)\cdot(0,5)\cdot(2,6)$	*get next*
$(6,1)\cdot\underline{(3,1)\cdot(6,4)}$	$\mathtt{\overline{cb}}\cdot\mathtt{b\overline{d}}$	$(0,5)\cdot(2,6)$	*cancel spur*
$(6,1)\cdot\underline{\mathtt{\overline{c}}\cdot\mathtt{\overline{d}}}$	$\mathtt{\overline{c}}\cdot\mathtt{\overline{d}}$	$(0,5)\cdot(2,6)$	*concatenate*
$\underline{(6,1)\cdot(6,0)}$	$\mathtt{\overline{cd}a}\cdot\mathtt{\overline{cd}}$	$(0,5)\cdot(2,6)$	*get next*
$(6,1)\cdot\underline{(6,0)\cdot(0,5)}$	$\mathtt{\overline{cd}}\cdot\mathtt{dcb}$	$(2,6)$	*concatenate*
$\underline{(6,1)\cdot\mathtt{b}}$	$\mathtt{\overline{cd}a}\cdot\mathtt{b}$	$(2,6)$	*concatenate*
$\underline{(6,2)}$	$\mathtt{\overline{cd}ab}$	$(2,6)$	*get next*
$\underline{(6,2)\cdot(2,6)}$	$\mathtt{\overline{cd}ab}\cdot\mathtt{c\overline{a}d\overline{b}}$		*concatenate*
$\underline{(6,6)}$	$\mathtt{\overline{cd}abc\overline{a}d\overline{b}}$		*delete*
	1		*done!*

Reducing the signature of a trivial word in the surface group $\langle \mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d} \mid \mathtt{abc\overline{a}d\overline{bcd}}\rangle$.
The middle column shows the expansion of the underlined subrelators.

One important difference between this algorithm and the algorithm described in the previous section (Theorem 7.9) is that we do *not* need to make any assumptions about the Euler characteristic of the surface. Exactly the same algorithm works without modification for paths on the torus, the Klein bottle, the projective plane, and even (duh) the sphere!

Theorem 7.10. *Let G be a cellular graph of complexity n on a surface Σ. There is an algorithm to determine whether a given closed walk in G of length k is contractible in Σ in $O(n+k)$ time.*

It is tempting to make an even stronger claim: The algorithm does not even require that the underlying group is the fundamental group of the surface. The only assumption the algorithm requires is that the group has a finite presentation with one relator, in which each generator appears at most twice; these are called. But in fact, this claim is *not* stronger, because *every* group with that property is the fundamental group of a 2-manifold (possibly with boundary); the relator is the traversal sequence of a polygonal schema!

References

[1] M. Dehn. Über die Topologie des dreidimensionalen Raumes. *Math. Ann.* 69(1):137–168, 1910.

[2] M. Dehn. Über unendliche diskontinuierliche Gruppen. *Math. Ann.* 71(1):16–144, 1911.

[3] M. Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Math. Ann.* 72(3):413–421, 1912.

[4] T. K. Dey and S. Guha. Transforming curves on surfaces. *J. Comput. System Sci.* 58:297–325, 1999.