An algorithmic classification of open surfaces

Sylvain Maillot

January 8, 2013

Abstract

We propose a formulation for the homeomorphism problem for open n-dimensional manifolds and use the Kerekjarto classification theorem to solve this problem for $n = 2$.

1 Introduction

One of the most important problems in the topology of manifolds is the homeomorphism problem, which asks for a given dimension n whether there is an algorithm to decide if two compact n-dimensional PL manifolds are homeomorphic. It is easy to show that the answer is yes for $n = 1$. For $n = 2$, the answer is also yes, thanks to the classification of surfaces. For $n = 3$, the problem is much more difficult; a positive answer follows from Perelman's proof [3, 5, 4] of Thurston's geometrisation conjecture as well as earlier work (see e.g. [1, Section 1.4] and the references within.) When $n \geq 4$ the answer is negative [2].

In this paper we consider *open*, i.e. noncompact (but paracompact) manifolds. The homeomorphism problem as formulated above does not make sense, because in general open manifolds are not defined by finite data which one can feed into a computer. For compact manifolds, a consequence of a positive answer to the homeomorphism problem in dimension n is that one can make a list $M_0, M_1, \dots$ of compact n-manifolds, such that any compact n-manifold is homeomorphic to exactly one member of the list. When we consider open manifolds of dimension $n > 1$, such a list cannot exist, because there are uncountably many n-manifolds (see Appendix A.)

To deal with this issue, we restrict attention to manifolds which are generated by a simple recursive procedure. First we make the following definition.

Definition. Let $n \geq 1$ be a natural number. A *topological n-automaton* is a triple $\mathcal{X} = ((X_0, \dots, X_p), (C_1, \dots, C_p), (f_1, \dots, f_q))$ where

- p, q are nonnegative integers;
- for each k, X_k is a compact triangulated n-dimensional manifold-with-boundary, called a *building block*;
- for each $k \geq 1$, C_k is a connected component of ∂X_k, hereafter called the *incoming boundary component* of X_k. The other boundary components of the building blocks, including all boundary components of X_0, if any, are called the *outcoming boundary components*;
- for each i, there exist k, l with $k \leq l$ such that f_i is a simplicial homeomorphism from some outcoming boundary component of X_k onto C_l; f_i is called an *arrow*;

subject to the condition that every outcoming boundary component is the domain of exactly one arrow.

An example of topological automaton is given in Figure 1.

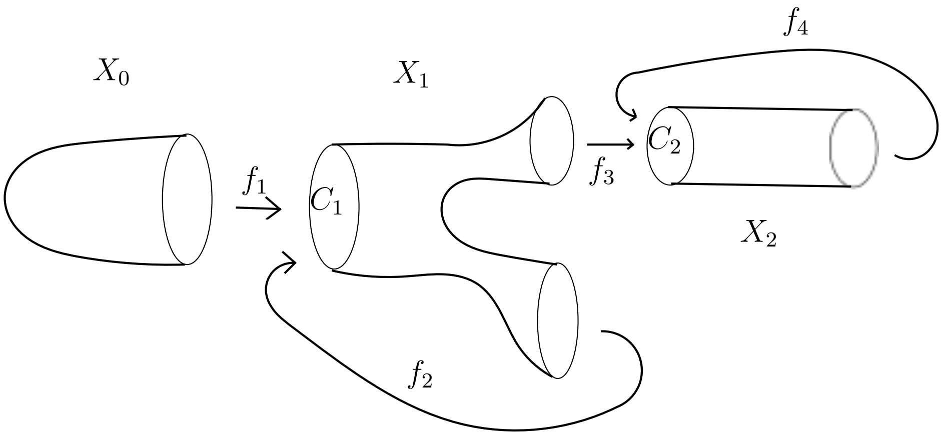

Figure 1: A topological 2-automaton $\mathcal{X}$

To any topological n-automaton $\mathcal{X}$ we associate an n-dimensional manifold $M(\mathcal{X})$ by starting with X_0 and 'following the arrows' to attach copies of the various building blocks. The idea is simple, but the formal definition is rather awkward, since we must specify an order in which the building blocks are attached, and keep track of the copies of the building blocks that have already been used. To this effect, note that by the last condition, the ordering of the f_i's induces a linear ordering L on the set of outcoming boundary components.

Let $\mathcal{X}$ be a topological n-automaton. We first define inductively a sequence of triples $\{(M_s, \phi_s, L_s)\}_{s \in \mathbf{N}}$ where for each s, M_s is a compact n-manifold-with-boundary, ϕ_s is a collection of bijections from the components of ∂M_s to outcoming boundary components of $\mathcal{X}$, and L_s is a linear ordering of the components of ∂M_s. For each $1 \leq k \leq p$ we generate a sequence of copies of X_k by setting $X_k^m := X_k \times \{m\}$ for each nonnegative integer m. By abuse of notation, we still denote by f_i the homeomorphisms between the various components of the ∂X_k^m's induced by f_i. We define M_0 to be X_0, ϕ_0 to be the set of identity maps of the components of ∂X_0, and L_0 is induced by L.

Assume that the triple $\{(M_s, \phi_s, L_s)\}$ has been defined. Take the first component C of ∂M_s, as given by L_s. Using ϕ_s, we can identify C to some outcoming boundary component of some X_k. Then by hypothesis there is a unique f_i with domain C. Let $C_{k'}$ be the range of f_i. Take the first copy of $X_{k'}$ which has not already been used in the construction, and attach it to M_s along f_i. Then repeat the operation for all components of ∂M_s, the order being determined by L_s. The resulting manifold is M_{s+1}. The set of homeomorphisms ϕ_{s+1} is determined by the identification of each added manifold with some building block. The ordering L_{s+1} is deduced from L_s and L in a lexicographical manner.

By construction, there is a natural inclusion map from M_s to M_{s+1}. Thus the M_s's form a direct system. We define $M(\mathcal{X})$ as the direct limit of this system. See Figure 2 for an example.

We can now state the main theorem of this paper, which can be thought of as a solution to the homeomorphism problem for open surfaces:

Theorem 1.1. *There is an algorithm which takes as input two topological* 2*-automata* $\mathcal{X}_1, \mathcal{X}_2$ *and decides whether* $M(\mathcal{X}_1)$ *and* $M(\mathcal{X}_2)$ *are homeomorphic.*

The proof of Theorem 1.1 relies essentially on the classification theorem for possibly noncompact surfaces, due to Kerekjarto and Richards [6]. We now recall the statement of this theorem.

Let F be a possibly noncompact surface without boundary. The classical invariants which are used in the statement of the classification theorem are the genus, the orientability class, and the triple $(E(F), E'(F), E''(F))$ where $E(F)$ is the space of ends of F, $E'(F)$ is the closed subspace of nonplanar ends, and $E''(F)$ is the closed subspace of nonorientable ends. Here is the statement of the classification theorem:

Theorem 1.2 (Classification of surfaces, Kerekjarto,Richards [6])**.** *Let* F_1, F_2 *be two surfaces without boundary. Then* F_1, F_2 *are homeomorphic if and only if they have same genus and orientability class, and* $(E(F_1), E'(F_1), E''(F_1))$ *is homeomorphic to* $(E(F_2), E'(F_2), E''(F_2))$*.*

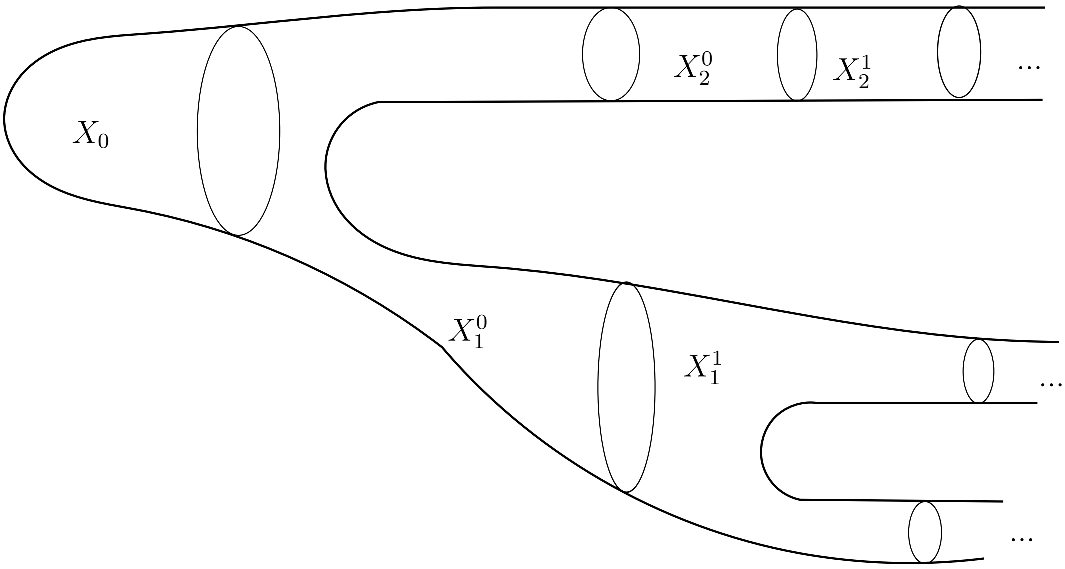

Figure 2: The surface associated to $\mathcal{X}$

Recall that a surface F is *planar* if every embedded circle in F separates. A planar surface F has genus 0, is orientable, and $E'(F) = E''(F) = \emptyset$. Thus planar surfaces are classified by their space of ends.

A topological 2-automaton is called *planar* if all of its building blocks are planar. This is equivalent to requiring that the associated surface should be planar.

The paper is structured as follows: in Sections 2 and 3, we deal with the planar case, i.e. we give an algorithm to decide whether the surfaces associated to two planar topological 2-automata are homeomorphic. In Section 2, we show how to associate to a planar topological 2-automaton $\mathcal{X}$ a combinatorial object $T(\mathcal{X})$, called an *admissible tree*, such that the structure of the space of ends of $M(\mathcal{X})$ can be read off $T(\mathcal{X})$. In Section 3, we introduce the notion of a *reduced tree*, which is a kind of normal form for $T(\mathcal{X})$, and prove Theorem 1.1 in the planar case. In Section 4, we explain how to generalize this construction in order to prove Theorem 1.1 in full generality. Finally, in Appendix A, we indicate a construction of uncountably many open surfaces which are pairwise nonhomeomorphic.

Acknowledgements The author would like to thank Gilbert Levitt, Eric Swenson, and Panos Papazoglu for useful conversations.

2 From automata to admissible trees

Let $\mathcal{X}$ be a planar topological 2-automaton. We start by forming the directed graph G' whose vertices are the building blocks and whose edges are the arrows. An arrow is called a *loop* if its domain and range lie in the same building block.

In order to describe the space of ends of the surface $M(\mathcal{X})$, take a finite alphabet A with as many letters as arrows in $\mathcal{X}$, and assign a letter to each arrow. We give the set $A^{\mathbf{N}}$ of infinite words in A the product topology. The space of ends of the surface $M(\mathcal{X})$ is homeomorphic to the subset $E \subset A^{\mathbf{N}}$ consisting of words which can be read by starting from X_0 and following the arrows.

To each finite word $w = a_1 \cdots a_n \in E$ we can associate the number k such that w leads to X_k, and the subset E_w of E of words whose prefix is w and whose other letters correspond to loops around X_k. There are three cases: if there is no such loop, then of course E_w is empty; if there is one, then E_w is a singleton; otherwise, E_w is a Cantor set, no matter how many loops around X_k there are. This follows from the well-known fact that the Cantor set is the only totally discontinuous compactum without isolated points. Hence we see that the number of loops around some building block is irrelevant as soon as it is greater than or equal to 2.

We are going to use this observation in order to associate to a planar topological 2-automaton $\mathcal{X}$ a simpler combinatorial structure from which the space of ends of $M(\mathcal{X})$ can be reconstructed. To this end, we introduce three formal symbols $*, O, \Theta$. To a planar topological automaton $\mathcal{X}$ we associate a decorated graph (G, f), where G is a finite directed graph endowed with a base vertex, called the *root*, and f is a map from the set of all vertices of G to the set $\{*, O, \Theta\}$, in such a way that the root has image $*$.

Vertices of G correspond to the building blocks X_k, the root being X_0. Edges of G correspond to arrows f_i which are not loops. The image of a vertex by f is $*$ (resp. O, resp. Θ) if there is no loop (resp. one loop, resp. two or more loops) around the corresponding building block. We say that a vertex is *of type* $*$, O, or Θ according to its image by f. A vertex different from the root is called an *ordinary vertex*.

From the decorated graph (G, f) we can reconstruct the space of ends E of $M(\mathcal{X})$ as follows. Set $p := n_O + 2n_\Theta$, where n_O (resp. n_Θ) is the number of vertices of G of type O (resp. of type Θ). Then we consider the alphabet $A = \{a_1, \ldots, a_p\}$ and assign to each vertex v of T either a letter (if v is of type O), an unordered pair of letters (if v is of type Θ), or nothing. Let $L(G)$ be the set of infinite words w on A that can be obtained according to the following recipe: let $c = v_1 \cdots v_l$ be a finite injective path in G starting from the root. For each $k \in \{2, \ldots, l-1\}$, choose a finite (possibly empty) word in the letter(s) assigned to v_k; then choose an infinite word in the letter(s) assigned to v_l; then w is obtained by concatenating all those words. Finally, $L(G)$ is topologised as a subset of $A^{\mathbf{N}}$ endowed with the product topology.

We denote by V the set of vertices of G.

We shall change the graph structure of G without changing the space of ends. First we need a definition.

Definition. An *admissible tree* is a decorated graph, which is a tree, such that the only vertex of type $*$ is the root.

The next task is to turn the graph G into a tree. If G is disconnected, then we simply delete the connected components of G which do not contain the root. Clearly, this does not change the space E.

If G is not a tree, then there exist three distinct elements $x_1, x_2, y \in V$ such that there is an edge e_1 from x_1 to y and an edge e_2 from x_2 to y. Take y minimal with this property, i.e. farthest from the root as possible. Thus a similar situation does not happen in the subgraph Y consisting of the vertices accessible from y and the edges between them. Let $x_3, \ldots, x_p$ be the other elements of W such that there is an edge e_i going from x_i to y for each i (if any.) By minimality, the graph Y is a tree. We construct a new decorated, rooted, oriented graph by introducing for each $2 \leq i \leq p$ a copy (Y_i, y_i) of (Y, y), deleting e_i and replacing it by an edge e_i' connecting x_i to y_i. An example of this operation is represented in Figure 3.

By repeating this construction finitely many times, we obtain a decorated, rooted, oriented graph G'' which is a tree. It may not be admissible, because there might be vertices of type $*$ other than the root. If x is any such vertex, let e be the edge leading to x (by the previous construction, there must be exactly one), let x' be the initial vertex of e, and let $e_1, \ldots, e_p$ be the edges with initial vertex x (if any.) Modify G'' by deleting the vertex x and the edge e, and replacing each e_i by an edge e_i' with initial vertex x' and same terminal vertex as e_i. See Figure 4 for an example of this operation.

Repeating this procedure finitely many times, we obtain an admissible tree, which we call $T(\mathcal{X})$. The associated space $L(T(\mathcal{X}))$ is homeomorphic to $L(G)$, hence to the space of ends of $M(\mathcal{X})$.

Given a planar 2-automaton $\mathcal{X}$, the admissible tree $T(\mathcal{X})$ can be effectively constructed from $\mathcal{X}$. Furthermore, it follows from the construction that if $\mathcal{X}_1, \mathcal{X}_2$ are two planar 2-automata, then $M(\mathcal{X}_1)$ is homeomorphic to $M(\mathcal{X}_2)$ if and only if $L(T(\mathcal{X}_1))$ is homeomorphic to $L(T(\mathcal{X}_2))$.

Since the construction of $T(\mathcal{X})$ from $\mathcal{X}$ is effective, this reduces the planar case of Theorem 1.1 to the classification of admissible trees up to the

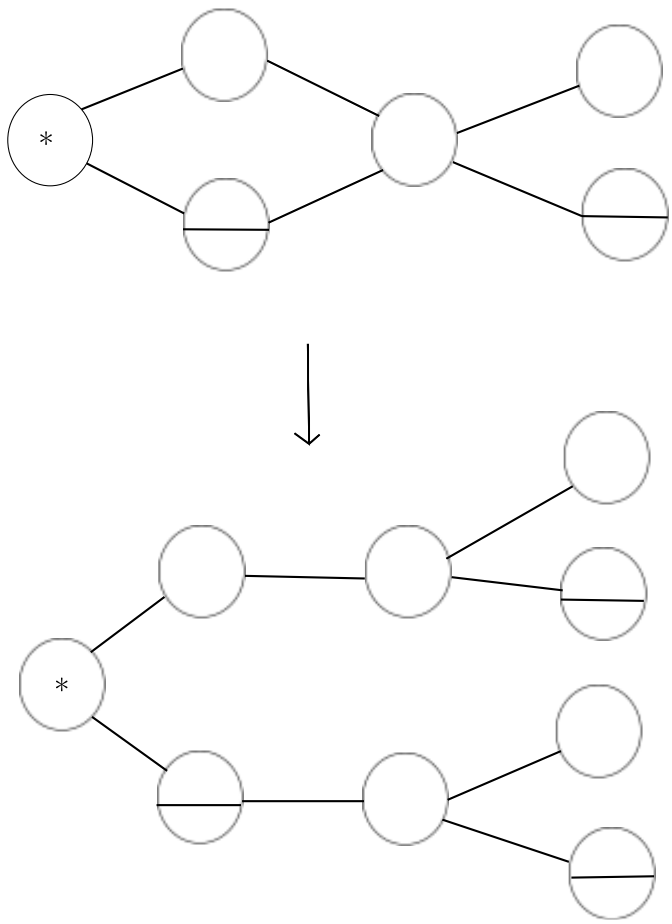

Figure 3: Making the graph G into a tree

equivalence relation of homeomorphism of the associated topological space. This classification is achieved in the next section.

3 The planar case

Let T_1, T_2 be two admissible trees. We say that they are *equivalent* if $L(T_1)$ and $L(T_2)$ are homeomorphic. We say that they are *isomorphic* if there is a type-preserving bijection between the set of vertices of T_1 and that of T_2 which respects the graph structure.

The goal of this section is to give an algorithm that determines whether two admissible trees are equivalent. We introduce three moves that can be used to simplify an admissible tree without changing its equivalence class.

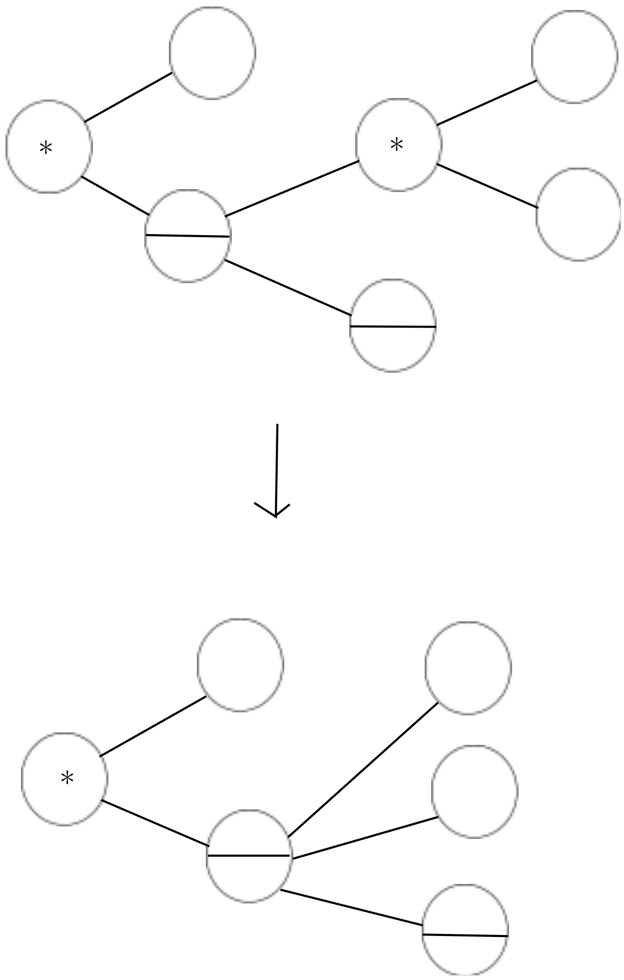

Figure 4: Making the tree admissible

Let T be an admissible tree. If v is a vertex of T, we denote by $T(v)$ the subtree consisting of v and its descendants.

Move 1 Let v be a vertex of T of type Θ. Let v' be the father of v. Assume that v' is not the root, and that v is the only son of v'. Remove the edge between v and v' and replace v' by v.

Move 2 Let v_1 be a vertex of T. Let v_2, v_3 be two descendants of v_1 such that v_2 is a son of v_1, and v_3 is not. Assume that the subtrees $T(v_2)$ and $T(v_3)$ are isomorphic. Remove $T(v_2)$.

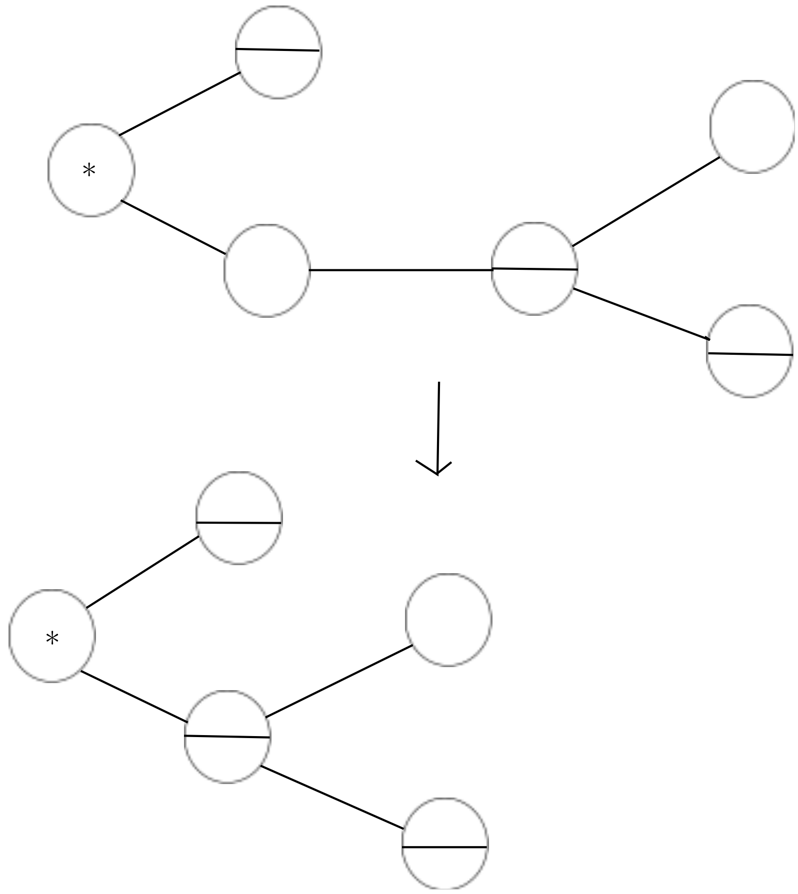

Figure 5: Move 1

Move 3 Let v_1 be a vertex of T. Let v_2, v_3 be two distinct sons of v_1. Assume that v_1 is not the root, or that both v_2 and v_3 are of type Θ. Further assume that the subtrees $T(v_2)$ and $T(v_3)$ are isomorphic. Remove $T(v_3)$.

The three moves are represented in Figures 5, 6, and 7 respectively.

Definition. An admissible tree is *reduced* if none of Moves 1–3 can be performed. For ease of reference we say that an admissible tree has property (R_i) if Move i is not possible.

We leave to the reader the tedious but elementary task of checking that Moves 1–3 do not change the associated space $L(T)$ up to equivalence. We then have the following proposition:

Proposition 3.1. *There is an algorithm which to any admissible tree T associates a reduced admissible tree equivalent to T'.*

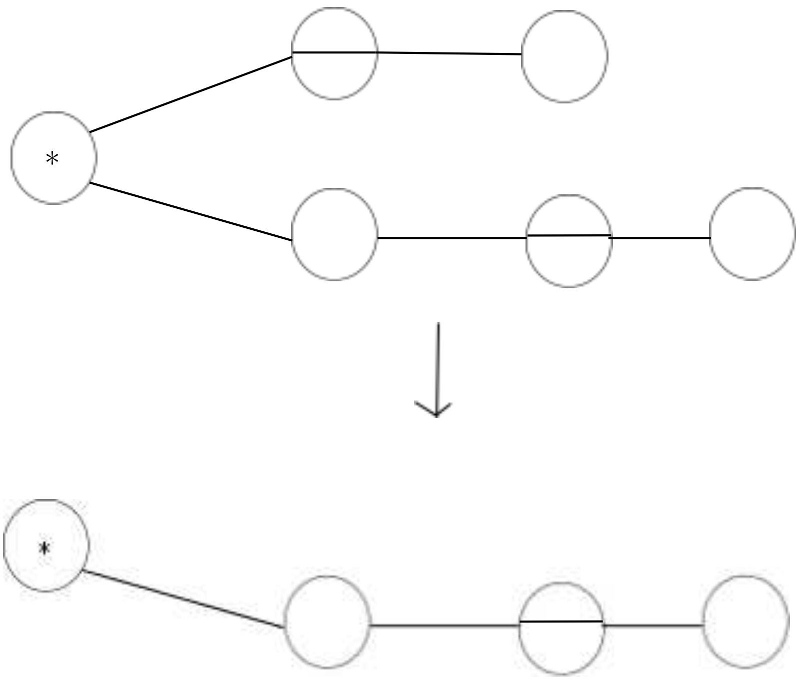

Figure 6: Move 2

Proof. Starting with T, apply Move 1 as much as possible, then Move 2 as much as possible, then Move 3 as much as possible, then Move 1 again etc. Each time a move is applied, the number of vertices of the tree goes down, hence the process eventually stops. □

The main result of this section is

Theorem 3.2. *Two reduced admissible trees T_1, T_2 are equivalent if and only if they are isomorphic.*

The 'if' part is clear, so we only have to prove the 'only if' part. First we make some definitions. Let L be a topological space. We define inductively a sequence of subsets $L_{-1}, L_0, L_1, \dots$ of L, each L_k being endowed with an equivalence relation, in the following way: first set $L_{-1} := \emptyset$. Assume that for some p all L_k have been defined for $k < p$, together with their equivalence relations. Then L_p is defined as the set of elements $x \in L$ such that $x \notin \bigcup_{k<p} L_k$, and there exists a closed subset K of L containing x, homeomorphic to a point or a Cantor set, such that the following two properties are satisfied:

i. For every $y \in K$, if (y_n) is a sequence converging to y, then y_n eventually belongs to $K \cup \bigcup_{k<p} L_k$.

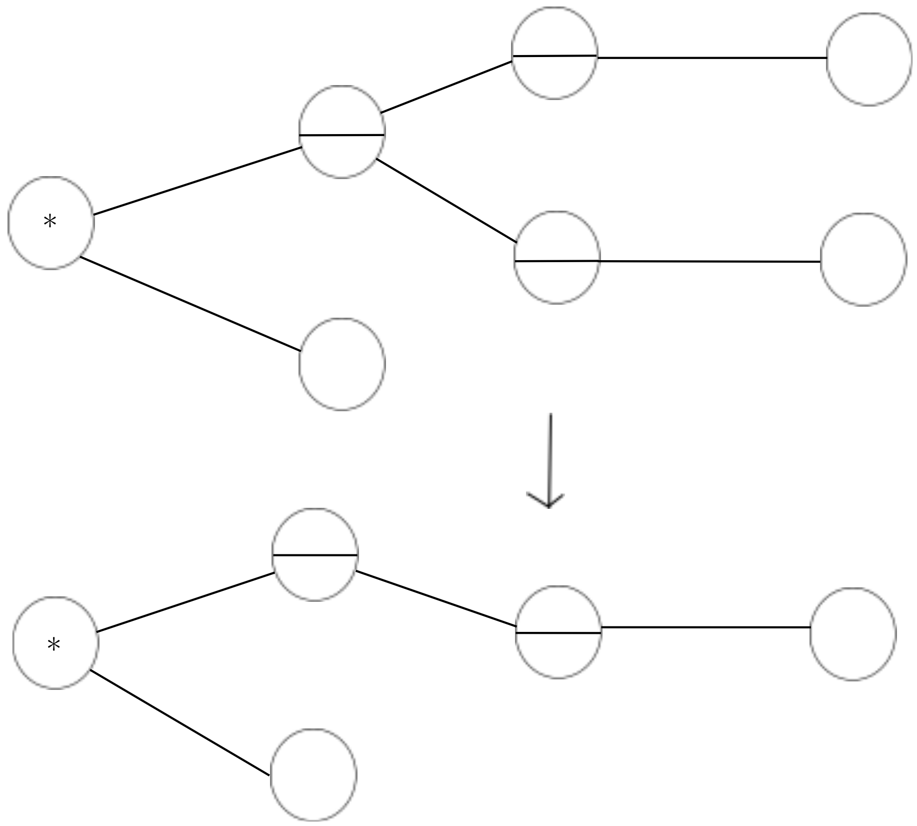

Figure 7: Move 3

ii. For every $y, z \in K$, for each sequence (y_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, if y_n tends to y, then there is a sequence (z_n) tending to z, such that for each n, z_n belongs to that same equivalence class.

The equivalence relation on L_p is defined by saying that two points $x, x' \in L_p$ are *equivalent* if the following properties are satisfied:

i. x is isolated in L_p if and only if x' is;

ii. for each sequence (x_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, converging to x, there is a sequence (x'_n) converging to x', such that for every n, x'_n is topologically equivalent to x_n;

iii. for every sequence (x'_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, converging to x', there is a sequence (x_n) converging to x, such that for every n, x_n is topologically equivalent to x'_n.

This process may go on forever, or stop when $L_p = \emptyset$ for some p. In general, there might be points of L which do not belong to any L_p. If $x \in L_p$ we say that x has *depth* p. Otherwise x has *infinite depth*. Also observe that points of depth 0 are either isolated, or belong to some clopen Cantor subset of L. In the latter case, they are *condensation points* of L, i.e. they do not have any countable neighbourhood.

Next we extend the definition of topological equivalence to points which belong to possibly different topological spaces.

Definition. Let (L, x) and (L', x) be pointed topological spaces. We say that x is *topologically equivalent to* x' if they have same (finite) depth p, and *same topological type*, the latter notion being defined inductively on p in the following way: if $p = 0$, then we say that x have same topological type if they are both isolated or both condensation points; if $p \geq 1$, assuming the equivalence relation has been defined for points of depth $< p$, we say that x and x' have same topological type if the following conditions are satisfied:

i. x is isolated among points of depth p if and only if x' is;

ii. for every sequence (x_n) of points of depth $< p$, belonging to a single equivalence class, converging to x, there is a sequence (x'_n) converging to x', such that for every n, x'_n is topologically equivalent to x_n;

iii. for every sequence (x'_n) of points of depth $< p$, belonging to a single equivalence class, converging to x', there is a sequence (x_n) converging to x, such that for every n, x_n is topologically equivalent to x'_n.

Next we apply this to $L = L(T)$ for some admissible tree T and make a few elementary remarks.

Let T be an admissible tree. If v is a vertex of T, we denote by L_v the set of words in $L(T)$ which have infinitely many letters corresponding to v. For every ordinary vertex v, any two elements of L_v have the same depth and are equivalent. Thus it makes sense to talk about the depth of a vertex, and we have an equivalence relation on vertices.

If $x \in L_v$ for some leaf v of T, then x is isolated or belongs to some Cantor clopen subset of L. Thus v has depth 0. If v is an ordinary vertex of T which is not a leaf, and all descendants of v have finite depth, then v has finite depth, equal to M or $M+1$, where M is the maximum of the depths of the descendants of v. The former possibility is possible only if v is equivalent to one of its sons. This implies that every ordinary vertex has finite depth, and depth is nonincreasing along paths in the tree starting from the root and going to the leafs. It turns out that for *reduced* trees, it is strictly decreasing. This will be a key point in the proof.

Proposition 3.3. *For every integer $p \geq 0$, the following assertions hold:*

i. Let T be a reduced tree and v be an ordinary vertex of T. If v has depth p, then every descendant of v has depth strictly less than p.

ii. Let T_1, T_2 be reduced trees. For $i = 1, 2$ let v_i be an ordinary vertex of T_i. If v_1 is topologically equivalent to v_2 and has depth p, then $T_1(v_1)$ is isomorphic to $T_2(v_2)$.

Proof. The proof is by induction on p. First we deal with the case $p = 0$. If Assertion (i) does not hold, let (T, v) be a counterexample with v *minimal*, i.e. as close to the leaves as possible. Then v is not a leaf, and has only leaves as descendants. If some son of v has type O, then every element of L_v is an accumulation point of distinct isolated points, contradicting the assumption that v has depth 0. Hence every son of v is of type Θ. By (R_1), v has at least two sons, and by (R_3) it has at most one. This is a contradiction.

This proves in particular that the ordinary vertices of depth 0 of a reduced tree are exactly the leaves. From this, Assertion (ii) in the case $p = 0$ is immediate.

Let $p \geq 1$. Assume that the Proposition holds for all $k < p$. We prove Assertion (i) by contradiction, taking again a minimal counterexample (T, v), with v of depth p. By monotonicity, there must be a son v' of v which also has depth p. As remarked earlier, v' must be equivalent to v. More precisely, letting x be a point of L_v and (y_n) be a sequence of points of $L_{v'}$ converging to x, there must exist a Cantor set $K \subset L$ containing x and almost all the y_n, such that all points of K are equivalent.

Claim. The vertex v' has type Θ.

We prove the claim by contradiction. If v' has type O, then the only way to construct a Cantor set containing infinitely many points of $L_{v'}$ is to use a descendant of v' of type Θ. By minimality of v, every descendant of v' has depth $< p$, so a Cantor set K with the above properties cannot exist. This proves the claim.

We proceed with the proof of Assertion (i). By Property (R_1), v' has at least one sibling v''. Suppose that v'' has depth $< p$. Since v'' can be used to produce an injective sequence of points converging to x, and v is equivalent to v', there must exist a descendant v''' of v' equivalent to v''. By Assertion (ii) of the induction hypothesis, $T(v'')$ is isomorphic to $T(v''')$. This contradicts (R_2). Hence v'' has depth p.

Claim. The trees $T(v')$ and $T(v'')$ are isomorphic.

Once the claim is proved, there is an immediate contradiction with (R_3). Let us prove the claim.

First note that v'' and v' are both equivalent to v. Hence v' is equivalent to v''. In particular v'' also has type Θ. We shall show that for each son w' of v' there is a son w'' of v'' such that $T(w')$ is isomorphic to $T(w'')$. Then by symmetry, the same will be true for v' and v'' exchanged, and using the fact (implied by (R_3)) that in a reduced tree the isomorphism type of a $T(v')$ is determined by the type of v' and the isomorphism type of the trees generated by its sons, the claim will follow.

Let w' be a son of v'. By minimality of v, w' has depth $< p$. Since v' is equivalent to v'', there is a descendant w'' of v'' which is equivalent to w'. By Assertion (ii) of the induction hypothesis, $T(w')$ and $T(w'')$ are isomorphic. There remains to show that w'' is a *son* of v''.

If not, we construct sequences w'_i, w''_i of vertices of T in the following way: set $w'_0 := w'$ and $w''_0 := w''$. Let w''_1 be the son of v'' who is an ancestor of w''. Arguing as above, we get a descendant w'_1 of v' such that $T(w'_1)$ is isomorphic to $T(w''_1)$. After finitely many steps, this construction stops, with some w'_k being a son of v', or some w''_k being a son of v''. At the last stage we obtain a contradiction with (R_2). This completes the proof of the claim, hence that of Assertion (i).

The proof of hereditarity of Assertion (ii) is based on similar arguments: let (T_1, v_1) and (T_2, v_2) satisfy the hypothesis. By Assertion (i), we know that all descendants of v_1 and v_2 have depth $< p$. Reasoning as above, we show that v_1 has type O if and only if v_2 has, and for every son v'_1 of v_1 there is a son v'_2 of v_2 such that $T_1(v'_1)$ is isomorphic to $T_2(v'_2)$. By rule (R_3) this is enough to guarantee that $T_1(v_1)$ is isomorphic to $T_2(v_2)$. □

Proof of Theorem 3.2. Let T_1, T_2 be reduced admissible trees whose associated topological spaces are equivalent. An argument similar to that used in the proof of Part (ii) of Proposition 3.3 shows the following Lemma:

Lemma 3.4. *The isomorphism types of the trees generated by the sons of the roots of* T_1, T_2 *are the same.*

By rule (R_3), each type of subtree with ancestor of type Θ can occur at most once. This need not be true for types of subtrees with ancestor of type O. However, each son of the root of type O contributes a single point, so the number of such points is invariant under homeomorphism. This concludes the proof of Theorem 3.2. □

4 The general case

In this section, we indicate how to adapt the proof from the planar case to the general case.

First, instead of the simple set $\{*, O, \Theta\}$, we introduce the infinite set $S := \{*_i, *_i^c, *_\infty, *_\infty^c, O, O^h, O^c, \Theta, \Theta^h, \Theta^c\}$, where i takes all nonnegative integer values in $*_i$, and all positive integer values in $*_i^c$. If some vertex of a decorated graph has type $*_i, *_i^c, *_\infty$ or $*_\infty^c$, then we say that this vertex has *starred type*.

We form the decorated graph (G, f) from the directed graph underlying the automaton by assigning to each building block X_k a symbol in S, in the following way: if there is no loop around X_k, then we put $*_i$ if X_k is orientable of genus i, and $*_i^c$ if X_k is nonorientable with i cross-caps; if there is one loop around X_k, then we put O if X_k is planar, O^h if it is orientable, but not planar, and O^c if it is nonorientable; if there are two or more loops, we put Θ if X_k is planar, Θ^h if it is orientable but not planar, and Θ^c if it is nonorientable.

Then we modify the decoration to take into account the fact that the subsets of nonplanar ends and nonorientable ends are closed in the set of ends: if some vertex of type $*_i$ has a descendant of type O^h or Θ^h, but no descendant of type O^c or Θ^c, then we change its label to $*_\infty$. If some vertex of type $*_i$ or $*_i^c$ has a descendant of type O^c or Θ^c, then we change its label to x_∞^c. If some vertex of type O (resp. Θ) has a descendant of type $O^h, \Theta^h, *_i$ or $*_\infty$, but no descendant of type $*_i^c$, O^c, or Θ^c, then we change its type to O^h (resp. Θ^h). Lastly, if some vertex of type O or O^h (resp. Θ or Θ^h) has a descendant of type $O^c, \Theta^c, *_i^c$, or $*_\infty^c$, then we change its type to O^c (resp. Θ^c).

Next we modify the decorated graph in order to make it a tree, by duplicating some of its parts. This is done exactly as in the planar case.

The next task is to make the tree admissible, which in this context means that only the root has starred type. This involves collapsing some edges of the tree G and merging some vertices. More precisely, we apply recursively the procedure of merging the starred vertices with their fathers. This is done according to two rules: first, when a vertex of starred type is merged with a vertex of type T that is not starred, then the resulting vertex has type T. Otherwise let T_1 be the type of the father, T_2 be the type of the son, and T_3 be the type of the resulting vertex. If $T_1 = *_i$ and $T_2 = *_j$, then $T_3 = *_{i+j}$; if $T_1 = *_i^c$, and ($T_2 = *_j$ or $T_2 = *_j^c$), then $T_3 = *_{i+j}^c$; if $T_1 = *_\infty$, then $T_3 = *_\infty$; if $T_1 = *_\infty^c$, then $T_3 = *_\infty^c$.

We recover the topological invariants of $M(\mathcal{X})$ from the admissible tree $T(\mathcal{X})$ in much the same way as in the planar case. First we read the genus (finite or infinite) and orientability class of $M(\mathcal{X})$ on the image of the root. Next, if T is an admissible tree, we make the following definitions. The space of ends $L(T)$ is defined as in the planar case by treating every vertex of type O^h or O^c as if it were O, and every vertex of type Θ^h or Θ^c as if it were Θ. The subspace of nonplanar ends $L'(T)$ is defined similarly, using only the subtree consisting of the root and the vertices of type O^h, Θ^h, O^c and Θ^c. The subspace of nonorientable ends $L''(T)$ is again defined similarly using only the root and the vertices of type O^c and Θ^c.

We say that two admissible trees T_1, T_2 are *equivalent* if they have same genus and orientability class, and the triple $(L(T_1), L'(T_1), L''(T_1))$ is homeomorphic to $(L(T_2), L'(T_2), L''(T_2))$.

Next we define reduced trees. This is done using three moves which extend the moves used in the planar case. Moves 2 and 3 are exactly the same as in the planar case. Only Move 1 needs the following adjustment:

Move 1' Let v be a vertex of T of type Θ (resp. Θ^h, resp. Θ^c). Let v' be the father of v. Assume that v' is not the root, and that v is the only son of v'. Further assume that v' has type O or Θ (resp. O^h or Θ^h, resp. O^c or Θ^c.) Remove the edge between v and v' and replace v' by v.

Let T be an admissible tree. If none of Moves 1', 2, 3 can be performed, we say that T is *reduced*. Then we have the following extension of Proposition 3.1, which is proved in the same way:

Proposition 4.1. *There is an algorithm which to any admissible tree T associates a reduced admissible tree equivalent to T'.*

Theorem 3.2 extends to the following result:

Theorem 4.2. *Two reduced admissible trees T_1, T_2 are equivalent if and only if they are isomorphic.*

The proof of Theorem 4.2 is similar to that of Theorem 3.2. From the construction, it is clear that two isomorphic trees are equivalent. For the converse, we first need to adapt the notion of depth and equivalence from topological spaces to triples of topological spaces.

Let (L, L', L'') be a triple of topological spaces, where $L'' \subset L' \subset L$ and L', L'' are closed. We define inductively a sequence of subsets $L_{-1}, L_0, L_1, \ldots$ of L, each L_k being endowed with an equivalence relation, in the following way: first set $L_{-1} := \emptyset$. Assume that for some p all L_k have been defined for $k < p$, together with their equivalence relations. Then L_p is defined as the set of elements $x \in L$ such that $x \notin \bigcup_{k<p} L_k$, and there exists a closed subset K of L containing x, homeomorphic to a point or a Cantor set, such that the following properties are satisfied:

i. Either all points of K belong to L' or none of them do;

ii. Either all points of K belong to L'' or none of them do;

iii. For every $y \in K$, if (y_n) is a sequence converging to y, then y_n eventually belongs to $K \cup \bigcup_{k<p} L_k$.

iv. For every $y, z \in K$, for each sequence (y_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, if y_n tends to y, then there is a sequence (z_n) tending to z, such that for each n, z_n belongs to that same equivalence class.

The equivalence relation on L_p is defined by saying that two points $x, x' \in L_p$ are *equivalent* if

i. x is isolated in L_p if and only if x' is;

ii. x belongs to L' if and only if x' does;

iii. x belongs to L'' if and only if x'' does;

iv. for each sequence (x_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, converging to x, there is a sequence (x'_n) converging to x', such that for every n, x'_n is equivalent to x_n, and

v. for every sequence (x'_n) of points of $\bigcup_{k<p} L_k$ belonging to a single equivalence class, converging to x', there is a sequence (x_n) converging to x, such that for every n, x_n is equivalent to x'_n.

The definition of depth is the same as in the case of spaces, i.e. a point of L has depth p if it belongs to L_p. Next we extend the definition of topological equivalence to points which belong to possibly different topological spaces.

Definition. Let (L_1, L'_1, L''_1) and (L_2, L'_2, L''_2) be triples of topological spaces as above. Let x be a point of L_1 and y be a point of L_2. We say that x is *topologically equivalent to* y if they have same (finite) depth p, and *same topological type*, the latter notion being defined inductively on p in the following way: if $p = 0$, then we say that x, y have same topological type if they are both isolated or both condensation points, x belongs to L'_1 if and only if y belongs to L'_2, and x belongs to L''_1 if and only if y belongs to L''_2. If $p \geq 1$, assuming the equivalence relation has been defined for points of depth $< p$, we say that x and y have same topological type if the following conditions are satisfied:

i. x is isolated among points of depth p if and only if y is;

ii. x belongs to L'_1 if and only if y belongs to L'_2;

iii. x belongs to L''_1 if and only if y belongs to L''_2;

iv. for every sequence (x_n) of points of depth $< p$, belonging to a single equivalence class, converging to x, there is a sequence (x'_n) converging to x', such that for every n, x'_n is topologically equivalent to x_n;

v. for every sequence (x'_n) of points of depth $< p$, belonging to a single equivalence class, converging to x', there is a sequence (x_n) converging to x, such that for every n, x_n is topologically equivalent to x'_n.

The same elementary remarks as in the planar case hold. In particular, it makes sense to talk about the topological type of a vertex of an admissible tree. As in the planar case, the key proposition is the following:

Proposition 4.3. *For every integer $p \geq 0$, the following assertions hold:*

i. Let T be a reduced tree and v be an ordinary vertex of T. If v has depth p, then every descendant of v has depth strictly less than p.

ii. Let T_1, T_2 be reduced trees. For $i = 1, 2$ let v_i be an ordinary vertex of T_i. If v_1 is topologically equivalent to v_2 and has depth p, then $T_1(v_1)$ is isomorphic to $T_2(v_2)$.

We omit the proof of this proposition, which is almost identical to that of Proposition 3.3. Then the proof of Theorem 4.2 is very close to the proof of Theorem 3.2.

5 Concluding remarks

Our definition of a topological automaton is rather restrictive. It is possible to broaden it, for instance by removing the restriction that $k \leq l$ in the last condition. This creates technical problems, but does not seem to enlarge the class of manifolds significantly.

At one extreme, one may think of associating a manifold to a Turing machine. One simple-minded way to do this, say in dimension 2, is to start with a disk; each time the Turing machine does something, add an annulus; if the machine stops, glue in a disk. Then the resulting surface is compact if and only if the Turing machine stops. Since the halting problem for Turing machines is undecidable, the homeomorphism problem for surfaces arising from this construction is undecidable, for a reason which has nothing to do with the topological complexity of the objects involved.

A Uncountably many surfaces

Proposition A.1. *There exist uncountably many planar surfaces up to homeomorphism.*

Proof. We use a construction which was shown to us by Gilbert Levitt. To any sequence $\mathcal{A} = (a_n)_{n\in\mathbf{N}}$ in $\{0,1\}^{\mathbf{N}}$ we associate a compact totally discontinuous topological space $A_{\mathcal{A}}$ in the following way: we start with a Cantor set C and an arbitrary sequence $\bar{x}_n$ of pairwise distinct points of C. We define inductively spaces X_n by letting X_0 be a single point x_0 and X_{n+1} be the space consisting of a point x_{n+1} and a sequence of copies of X_n converging to it. Then we set

$$A_{\mathcal{A}} := C \cup_{x_n=\bar{x}_n} \bigsqcup_{n\in\mathbf{N},a_n=1} X_n,$$

i.e. for each integer n such that $a_n = 1$, we attach X_n to C along the points $x_n, \bar{x}_n$.

We claim that $A_{\mathcal{A}}$ and $A_{\mathcal{A}'}$ are homeomorphic only if $\mathcal{A} = \mathcal{A}'$. To see this, observe that the set of condensation points of $A_{\mathcal{A}}$ (i.e. the set of points all of whose neighbourhoods are uncountable) is exactly C. Thus if there is a homeomorphism $f : A_{\mathcal{A}} \to A_{\mathcal{A}'}$, then f must map C to C. The other points have finite Cantor-Bendixson rank, and the rank is preserved by f, i.e. f sends isolated points to isolated points, limits of sequences of isolated points to similar points etc. Thus if $a_n = 1$ for some integer n, then $a'_n = 1$ and f sends $\bar{x}_n \in A_{\mathcal{A}}$ to $\bar{x}_n \in A_{\mathcal{A}'}$. Conversely, if $a'_n = 1$ for some n, then $a_n = 1$. Thus $\mathcal{A} = \mathcal{A}'$. □

References

[1] L. Bessières, G. Besson, S. Maillot, M. Boileau, and J. Porti. *Geometrisation of 3-manifolds*, volume 13 of *EMS Tracts in Mathematics*. European Mathematical Society (EMS), Zürich, 2010.

[2] W. W. Boone, W. Haken, and V. Poenaru. On recursively unsolvable problems in topology and their classification. In *Contributions to Math. Logic (Colloquium, Hannover, 1966)*, pages 37–74. North-Holland, Amsterdam, 1968.

[3] G. Perelman. The entropy formula for the Ricci flow and its geometric applications. ArXiv : math.DG/0211159, Nov 2002.

[4] G. Perelman. Finite extinction time for the solutions to the Ricci flow on certain three-manifolds. ArXiv : math.DG/0307245, Jul 2003.

[5] G. Perelman. Ricci flow with surgery on three-manifolds. ArXiv : math.DG/0303109, Mar 2003.

[6] I. Richards. On the classification of noncompact surfaces. *Trans. Amer. Math. Soc.*, 106:259–269, 1963.

Institut de Mathématiques et de Modélisation de Montpellier, CNRS - Universit Montpellier 2, Case Courrier 051, Place Eugène Bataillon, 34095 MONTPELLIER Cedex, France
sylvain.maillot@math.univ-montp2.fr