Chapter 2

Unfoldability of Trees

2.1 Introduction

We say that a rooted tree embedded in the plane is in *straight configuration* if it consists of a single vertex, or

- It lies completely in a cone with apex the root r, and
- All subtrees of the children of the root are straight, their cones are pairwise disjoint, and the ray from each child c extending the line segment $\overline{rc}$ lies completely within the cone belonging to c.

See Figure 2.1.

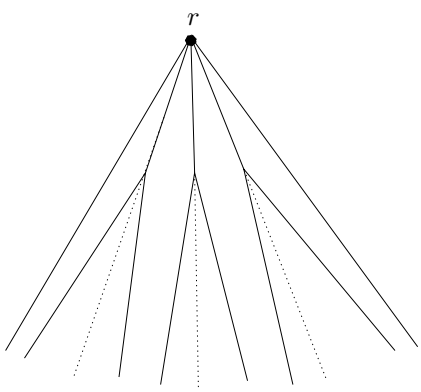

Figure 2.1: The structure of a tree in straight configuration.

We say that a tree is *unfoldable* if it can be moved into a straight configuration.

Alt, Knauer, Rote and Whitesides [2] showed that it is PSPACE-hard to decide whether two given configurations of a tree in the plane can reach each other.

There are many nice questions about unfoldability of trees. In general, one would like to know which conditions a tree must satisfy for being unfoldable.

This is a very wide question, and one could try to answer more particular questions about the unfoldability of trees of given characteristics. In this chapter we show that monotone trees are unfoldable. This result was already published in 2002 by Kusakari et al. [38] but we were not aware of it.

The algorithm given by Biedl et al. [8] for convexifying monotone polygons uses similar ideas as the algorithm described here.

2.2 Monotone Trees are Unfoldable

As Snoeyink and Stolfi in [50], given two segments with disjoint interiors in $\mathbb{R}^2$, we say that e_j is *below* e_i (we denote $e_j \preceq e_i$) if a vertical ray drawn downward from a point $p_i \in e_i$ encounters e_j at a point $q_j \in e_j$ (here both points are not necessarily in the interior of the segments, but $p_i \neq q_j$ must be satisfied). See Figure 2.2.

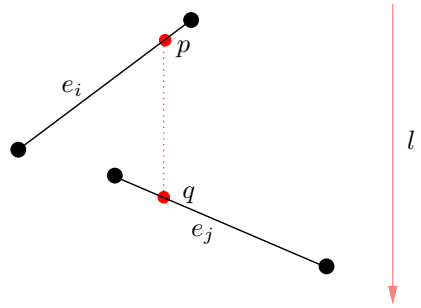

Figure 2.2: The segment e_j is below e_i.

This relation is acyclic, as shown in de Bruijn [10]. Hence, given a set of segments, there is an element without predecessor with respect to belowness, which we call $\preceq$*-minimal*.

A planar embedding of a tree in $\mathbb{R}^2$ is called *monotone* if there is a direction l such that the root is the topmost vertex and every path from the root to the leaves intersects at most once any line l' perpendicular to l; in other words, every path from the root to the leaves goes always downwards. Given a monotone planar embedding of a rooted tree, consider any orthogonal coordinate system such that l is the y-axis and the root is the topmost vertex.

We say that an edge of a tree is *final* if it is incident to a leaf. We define two operations:

- *Remove straight vertical parents of leaves.* When we have an arc uvw such that uv, vw are vertical, w is a leaf and v has degree 2, remove v and coalesce uvw into a single final vertical edge uw. See Figure 2.3.
- *Merge final vertical edges.* When two final edges e_1, e_2 incident to the same node are vertical, merge them in a single edge of length the maximum of the lengths of e_1, e_2. See Figure 2.4.

We say that a monotone tree is *reduced* after applying recursively these two operations until they can no longer be applied. Given a reduced tree T which is monotone with respect a direction l, we denote by T' the tree resulting from omitting the vertical final edges.

Lemma 2.1. *Let T' be a reduced monotone tree without vertical final edges. Then we can always find in T' a $\preceq$-minimal final edge.*

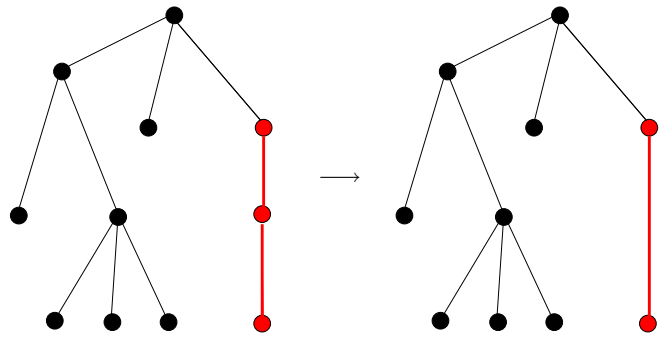

Figure 2.3: Removing straight vertical parents of leaves.

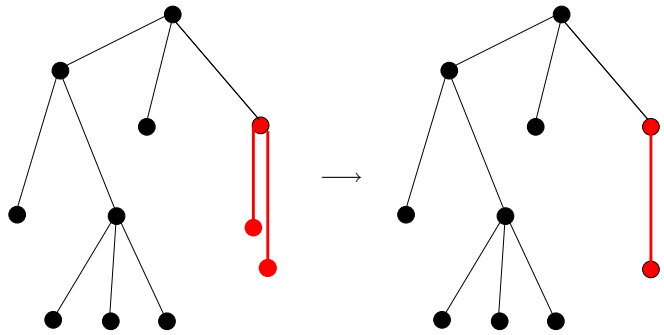

Figure 2.4: Merging final vertical edges.

De Bruijn's acyclicity result [10] proves that we can always find in T' a $\preceq$-minimal edge, but this edge may not be a final edge. To prove that a $\preceq$-minimal final edge can always be found, we need the following lemma.

Lemma 2.2. *Given a simple polygon C, consider the walk from a bottommost vertex b to a topmost vertex t along the right (or the left) boundary chain. Let x be a rightmost vertex (leftmost vertex) of C whose ingoing edge is not vertical. Let e_x^- and e_x^+ be the ingoing edge and the outgoing edge of x respectively. Then the extremal vertex x has the property that its outgoing edge e_x^+ lies in the upper half-plane defined by e_x^-.*

Proof. The vertices b, t and x belong to the convex hull of C. Let $\mathcal{R}$ be the closed polygonal region bounded by the path along C from b to x and the lower hull of C between b and x. The top vertex t lies outside $\mathcal{R}$; thus, the path along C from x to t must be totally exterior to $\mathcal{R}$ since C is simple. Hence e_x^+ must lie in the upper half-plane defined by e_x^-, otherwise it would get into $\mathcal{R}$. See Figure 2.5. In the particular cases where x coincides with the bottommost or the topmost vertex of C, this is also fulfilled. □

Proof of Lemma 2.1. We use this easy algorithm:

Algorithm FindMinimalFinalEdge (T')
Input. A reduced monotone tree $T' = (V, E)$ without vertical final edges
Output. A $\preceq$-minimal final edge
1. Take a final edge e_i

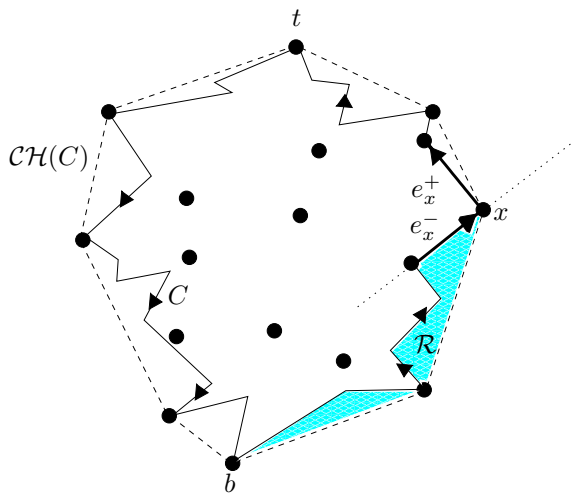

Figure 2.5: A simple directed polygon C.

2. **while** $\exists\, e_j \in E$ such that exist points $p_i \in e_i, q_j \in e_j$, $p_i \neq q_j$, where the directed segment $p_i q_j$ points downwards in the direction l
3. Walk from p_i to q_j
4. **if** e_j is not a final edge
5. Walk arbitrarily downwards along T' from q_j until you get a final edge e_i and stop walking at the point q_i where e_i is reached
6. **else** $e_i := e_j$
7. **return** e_i

The algorithm describes a directed walk, a simple polygonal chain in $\mathbb{R}^2$. See Figure 2.6. Let P be this simple polygonal chain.

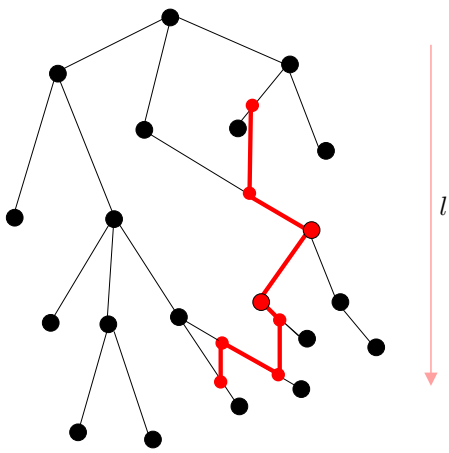

Figure 2.6: The obtained polygonal chain P (in thick lines).

Walking along P, suppose we reach a rightmost (leftmost) vertex p_i after a non-vertical edge of the path, which belongs to an edge e_i. This vertex p_i does not satisfy Lemma 2.2:

Once p_i is reached, then we go vertically downwards (if e_i is a final edge, from p_i to $q_j \in e_j$ such that $e_j \preceq e_i$), or downwards along T' (if e_i is not a final edge). Then P is not a cycle. □

Then we can prove the following theorem:

Theorem 2.1. *Every monotone tree is unfoldable.*

Proof. A monotone tree can be unfolded as follows:

Algorithm UnfoldMonotoneTree (T)
Input. A reduced tree T monotone with respect a direction l
Output. An unfolding of T
1. **while** T is not unfolded
2. Find a $\preceq$-minimal final edge $e = uv$ in T' (v is a leaf of T')
3. Rotate e in T until the directed segment uv is vertical pointing downwards
4. **if** the flexible joint v is adjacent to a final vertical edge e_v in T
5. Keep e_v also vertical during the rotation
6. Reduce T
7. **return** T

Step 4 is represented in Figure 2.7.

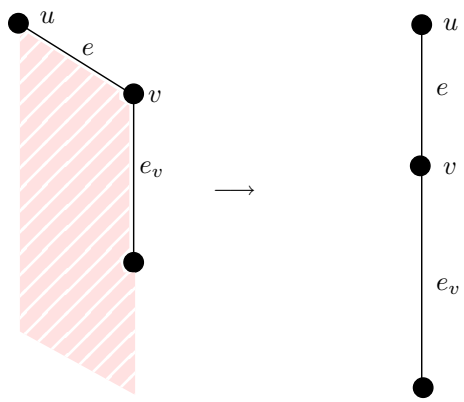

Figure 2.7: Rotate e in T until it points down.

Note that obviously we do not have problems in the 4th step when e is not a final edge in T. Note also that if e is $\preceq$-minimal in T', then it is also $\preceq$-minimal in T (leaving out e_v if it exists).

□

2.3 Open Problem

An *arbitrarily flattened tree* is a tree which is embedded within a strip of width ε, for ε arbitrarily small. Another related question that one could study is the following.

Question 2.1. *Is every arbitrarily flattened tree unfoldable?*

Note that if we could prove that every arbitrarily flattened tree can be flattened to a self-touching configuration, then we would have a positive answer to Question 2.1. This is true since all flat configurations can reach each other: one can show that if a tree linkage configuration can be straightened, then it can be straightened using any vertex as the root [6].