

Representing Directed Trees as Straight Skeletons

Oswin Aichholzer¹, Therese Biedl², Thomas Hackl¹, Martin Held³,
Stefan Huber⁴, Peter Palfrader³(✉), and Birgit Vogtenhuber¹

¹ Institut für Softwaretechnologie, Technische Universität Graz, 8010 Graz, Austria
`{oaich,thackl,bvogt}@ist.tugraz.at`

² Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON N2L 1A2, Canada
`biedl@uwaterloo.ca`

³ FB Computerwissenschaften, Universität Salzburg, 5020 Salzburg, Austria
`{held,palfrader}@cosy.sbg.ac.at`

⁴ Institute of Science and Technology Austria, 3400 Klosterneuburg, Austria
`stefan.huber@ist.ac.at`

Abstract. The straight skeleton of a polygon is the geometric graph obtained by tracing the vertices during a mitered offsetting process. It is known that the straight skeleton of a simple polygon is a tree, and one can naturally derive directions on the edges of the tree from the propagation of the shrinking process.

In this paper, we ask the reverse question: Given a tree with directed edges, can it be the straight skeleton of a polygon? And if so, can we find a suitable simple polygon? We answer these questions for all directed trees where the order of edges around each node is fixed.

1 Introduction

Many geometric structures on sets of points, line segments, or polygons, e.g. Delaunay triangulations, Voronoi diagrams, straight skeletons, and rectangle-of-influence graphs can be represented as graphs. The *graph representation* problem (for each of these geometric structures) asks which graphs can be represented in this way. That is, given a graph G , can we find a suitable input set S of points, segments, or polygons such that the geometric structure induced by S is equivalent to G ?

Graph representation has been studied for numerous geometric structures in the past. To name just a few examples: Every planar graph is the intersection graph of line segments [6], every wheel is a rectangle-of-influence graph [11], and all 4-connected planar graphs are Delaunay triangulations [8]. See also [7] for many results on proximity drawability of graphs.

Of particular interest to our paper are two results. First, Liotta and Meijer [12] studied when a tree can be represented as the Voronoi diagram of

OA and BV supported by Austrian Science Fund (FWF) I 648-N18; TB by NSERC; TH by FWF P23629-N18; MH and PP by FWF P25816-N15.

a set of points, and showed that this is always possible (and the points are in convex position). Secondly, Aichholzer et al. [4] studied when a tree can be represented as the straight skeleton of a polygon, and showed that this is always possible (and the polygon is convex).

1.1 Background

The *straight skeleton* $\mathcal{S}(P)$ of a simple polygon P is defined via a wavefront-propagation process: Each edge of P emits a wavefront edge moving in a self-parallel manner at unit speed towards the interior of the polygon.

Initially, at time $t = 0$, this wavefront is a single polygon that is identical to P . As the propagation process continues, however, the wavefront will change due to self-interaction: (i) In *edge events*, an edge of the wavefront shrinks to zero length and is removed from the wavefront. (ii) In *split events*, a vertex of the wavefront meets the interior of a previously non-incident wavefront edge. This split partitions the edge and the polygon into two parts that now propagate independently. (iii) If the input is not in general position, more complex interactions are possible. For example, entire portions of the wavefront collapse at once when parallel wavefronts that were emanated by parallel polygon edges meet, or new reflex wavefront vertices are created when multiple reflex vertices interact in a *vertex-event*. The propagation process ends once all components of the wavefront have collapsed. Therefore, the set of wavefront edges at any time t form one or more polygons, which we call the wavefront and denote by $\mathcal{W}(t)$.

The straight skeleton $\mathcal{S}(P)$, introduced by Aichholzer et al. [2], is then defined as the geometric graph whose edges are the traces of the vertices of $\mathcal{W}(t)$ over time; see Fig. 1. For simple polygons, $\mathcal{S}(P)$ always is a tree [2], with the leaves corresponding to vertices of P and interior vertices having degree 3 or more. Several algorithms are known to construct the straight skeleton [1, 9, 10].

We can distinguish between convex and reflex vertices of P or $\mathcal{W}(t)$. A vertex v is *reflex (convex)* if the interior angle at v is greater (less) than π . We call an arc of $\mathcal{S}(P)$ *reflex (convex)* if it was traced out by a reflex (convex) vertex of the wavefront. When discussing the wavefront propagation process, we will often interchangeably refer to wavefront vertices and straight skeleton arcs.

The roof model [2] represents a convenient means to study the wavefront over the entire propagation period. It embeds the wavefront in three-space, where the z -axis represents time: $\mathcal{T}(P) := \bigcup_{t \geq 0} (\mathcal{W}(t) \times \{t\})$. The inner edges and vertices of this polytope correspond to arcs and nodes of the straight skeleton $\mathcal{S}(P)$, and the z -coordinate of each element corresponds to the time it was traced out by

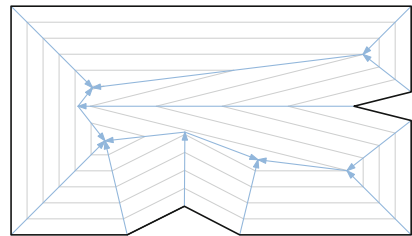


Fig. 1. The straight skeleton $\mathcal{S}(P)$ of an input polygon P (bold) is the union of the traces of wavefront vertices. Wavefront polygons at different times are shown in gray.

the wavefront-propagation process. Reflex arcs correspond to valleys and convex arcs to ridges.

If we exclude polygons where parallel polygon edges cause entire wavefront segments to collapse at one time, resulting in horizontal roof edges, then arcs of $\mathcal{S}(P)$ will have been traced out by the wavefront during its propagation process, and we can assign a natural direction to these arcs: make them point into their trace direction. This assignment gives rise to the *directed straight skeleton*, $\mathcal{S}_d(P)$.

A *directed tree* T is a directed graph whose underlying undirected graph is a tree, i.e., connected and acyclic. A *labeled tree* T_ℓ is a tree with assignments of labels to its arcs. For most of this paper, trees are *ordered*, i.e., for every node there is a fixed circular order in which the arcs appear around this node.

It is customary to refer to the edges and vertices of the straight skeleton as *arcs* and *nodes*, and to reserve *edges* and *vertices* for elements of input or wavefront polygons. We also use *arcs* and *nodes* to refer to elements of trees.

1.2 Our Results

Our paper was inspired by the work in [4], which studies undirected trees. However, the structure of the straight skeleton imposes directions on the arcs, except in degenerate cases. Hence, the natural question to ask is:

Problem 1 (Directed straight-skeleton realizability). Given a directed tree T , (i) is there a polygon P such that $\mathcal{S}_d(P)$ shares the structure of T (we denote this by $\mathcal{S}_d(P) \sim T$), and (ii) if yes, can we reconstruct such a polygon P from T ?

Having directions assigned to the arcs makes the straight-skeleton realizability problem significantly harder: For example, one easily sees that in a convex polygon the straight skeleton is a rooted tree (with exactly one sink), and so not all directed trees can be represented via convex polygons. Hence, the results from [4] do not transfer to directed trees.

The directed-straight-skeleton-realizability question can be asked for multiple meanings of “directed tree”: It could be a *geometric tree* (nodes are given with coordinates), an *ordered tree* (the clockwise order of arcs around each node is specified) or an *unordered tree* (we have the nodes and arcs but nothing else). For a geometric tree, the problem is trivial, since the locations of the leaves specify the vertices of the only polygon for which this could be the straight skeleton. (If leaves are not specified as points but only as “being on a ray”, then the geometric setting is non-trivial, but can be solved in polynomial time [5].)

In this paper, we consider the variant of the problem for ordered trees. In the case of polygons in general position, we give three obviously necessary conditions and show that these are also always sufficient. It turns out that the order of arcs around nodes is not important, so the algorithm also works for unordered directed trees. We then turn to polygons without restrictions on vertex-positions. In this case the directed straight skeletons can be significantly more complicated, and in particular, have arbitrary degrees. Testing whether a directed tree can be represented as straight skeleton requires deeper insight into the structure of

straight skeletons, and we can exploit these to develop such a testing algorithm and, in case of a positive answer, find a suitable polygon.

2 Trees from Polygons in General Position

In a first step we restrict the problem to polygons in general position. By general position we mean that no four edges have supporting lines which are tangent to a common circle. In particular this means that during the wavefront propagation process only standard edge and split events are observed, resulting in straight skeletons where all interior nodes are of degree exactly three.

Investigating the structure of such directed straight skeletons enables us to establish a number of necessary conditions for a directed tree to be a directed straight skeleton of a polygon in general position.

Necessary Conditions. Let P be a polygon in general position and let T be the directed tree such that $\mathcal{S}_d(P) \sim T$.

The leaves of T correspond to the vertices of P . In the roof model, these vertices have zero z -coordinate, while all other nodes have positive z -coordinates since they will have been created by an event at some time $t > 0$. Thus, any arc incident to a vertex v of P increases in elevation as it moves away from v . As such, all leaves of T must have in-degree 0 and out-degree 1.

The interior nodes all have degree 3 and are classified by their in- and out-degrees as follows:

in-degree 3: (*peak nodes*) A collapse of a wavefront component (of triangular shape) at the end of its propagation process is witnessed by a local maximum in the roof. These local maxima correspond to nodes with in-degree three.

in-degree 2: (*collapse nodes*) Edge events in the propagation process, i.e., collapsed wavefront edges, result in a node with two incoming arcs and one outgoing arc.

in-degree 1: (*split nodes*) Split events will cause a node that has only one incoming arc and two outgoing arcs.

in-degree 0: Since the roof model will have no local minima except at the edges of P [2], nodes with in-degree zero and out-degree three cannot exist.

Of these, the case of a split event requires some more attention since it imposes additional restrictions on the incoming arc. Recall that we can distinguish between reflex and convex vertices of P , and note that any vertex is either reflex or convex by the general position assumption. For a split event to occur, a reflex vertex of the wavefront must crash into a previously non-incident part of the wavefront.

In the absence of vertex events, which create skeleton-nodes of degree at least four and therefore do not happen when the polygon is in general position, no reflex vertex can ever be created by an event. Thus, any reflex vertex that is part of an event must have been emanating from a reflex vertex of the input polygon

itself. Accordingly, the incoming arc in a split event node must have a leaf at its other end.

We summarize these conditions in the following lemma.

Lemma 1. *Let P be a simple polygon in general position and let T be the directed tree such that $\mathcal{S}_d(P) \sim T$. Then in T the following hold:*

- (G1) *the incident arc of each leaf is outgoing,*
- (G2) *every interior node has degree 3 and at most two outgoing arcs,*
- (G3) *if an interior node has out-degree two, then the incoming arc connects directly from a leaf.*

Observations. Let T be a directed tree that satisfies conditions (G1–G3). Classify the interior nodes as split nodes, collapse nodes and peaks as above. The goal is to show that any such tree can indeed be realized as a straight skeleton. For this, we split the tree into multiple subtrees in a particular way (also illustrated in the example in Fig. 3). We have the following observations. Full proofs of the next four lemmas can be found in Appendix A in the arXiv-version of this work [3].

Lemma 2. *Let T be a directed tree that satisfies conditions (G1–G3). If T has no split nodes, then T has exactly one peak.*

Lemma 3. *Let T be a tree that satisfies conditions (G1–G3). Create a forest F as follows: At any split node s of T , remove s , remove the leaf incident to the incoming arc of s , and replace the two outgoing arcs of s by two new leaves that are connected to the other ends of these arcs. Then each component of F satisfies conditions (G1–G3) and has exactly one peak.*

Sufficient Conditions. It remains to be shown that the necessary conditions (G1–G3) from Lemma 1 are also sufficient. We show this by constructing a simple polygon P such that $\mathcal{S}_d(P) \sim T$, given a directed tree T that satisfies (G1–G3). We start by showing this for trees that have no split nodes.

Lemma 4. *For any directed tree T that satisfies (G1–G3) and has no split nodes, there is a convex polygon P such that $\mathcal{S}_d(P) \sim T$.*

Proof. We show this by constructive induction. Any triangle is a polygon such that its straight skeleton shares the structure of the peak node of T .

To construct a polygon P for a tree T with k interior nodes, we first construct a polygon P' for a tree T' with $k - 1$ interior nodes. We obtain T' by replacing a node of T and its two adjacent leaves with a single leaf ℓ . (There always is such a node.) To obtain P , we compute an exterior offset of P' and replace the vertex that corresponds to ℓ with a sufficiently small edge such that it collapses before the wavefront of P reaches P' .

This polygon P will then satisfy $\mathcal{S}_d(P) \sim T$. See Fig. 2 for an illustration. \square

Furthermore, it is possible to add a constraint on one interior angle of the polygon:

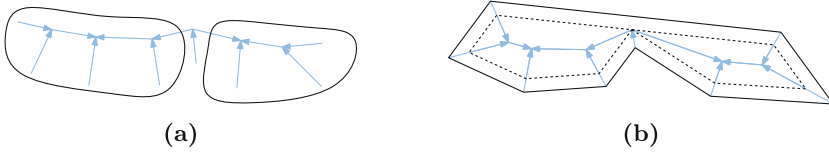


Fig. 3. (a) Given a directed tree we split it into a forest F of subtrees without split nodes. (b) Recursing on the structure of F , we can create convex polygons for each element (dotted) and then merge them into ever larger polygons.

Lemma 5. *Let T be a directed tree without split nodes and let ℓ be a leaf of T . Further, let α be an arbitrary angle with $0 < \alpha < \pi$. Then there exists a convex polygon P such that $\mathcal{S}_d(P) \sim T$ and such that the interior angle at the vertex that corresponds to ℓ is α .*

Now we are ready to consider trees with split nodes.

Lemma 6. *Let T be a directed tree that satisfies conditions (G1–G3). Then there exists a polygon P such that $\mathcal{S}_d(P) \sim T$.*

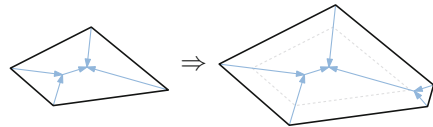


Fig. 2. Creating the polygon for a tree with $k + 1$ interior nodes from a polygon for a tree with one less.

Proof. As in Lemma 3, we split T at split nodes, also dropping the incoming reflex arc and its incident leaf. We obtain a forest $F = \{T_1, T_2, \dots, T_n\}$ where each T_i is a tree without split nodes.

This forest can in turn be considered an undirected tree, where each T_i gives rise to a node and nodes are connected if and only if the corresponding trees originally had a split node in common. We pick an arbitrary root for F , say T_1 , and construct a convex polygon P_1 such that $\mathcal{S}_d(P_1) \sim T_1$.

This root T_1 is connected to one or more children in F via split nodes. Let T_2 be such a child, and let $\ell_1 \in T_1$ and $\ell_2 \in T_2$ be the two leaves obtained when splitting at the split node common to T_1 and T_2 . Let v_1 be the vertex in P_1 corresponding to ℓ_1 , and assume it has angle $\alpha_1 < \pi$.

We construct a convex polygon P_2 such that $\mathcal{S}_d(P_2) \sim T_2$ and the vertex v_2 corresponding to ℓ_2 has angle $\alpha < \pi - \alpha_1$. This enables us to merge P_1 and P_2 in the following way: We place P_2 in the plane such that v_1 of P_1 and v_2 of P_2 occupy the same locus. We rotate P_2 such that the angle between a pair of edges of P_1 and P_2 is exactly π . Which pair of edges is chosen depends on where in the cyclic order the incident reflex leaf at the split node in T lies. The layout of P_1 and P_2 then corresponds to the layout of the wavefront at the split-event time. If we now compute a small outer offset and designate this to be P , then the directed straight skeleton of P has the same structure as the subtree of T that is made up by T_1, T_2 , the split node, and its incident leaf; see Fig. 3.

We then repeat this process for another child of T_1 or T_2 . Note that it may be necessary to scale the polygon that we add to a sufficiently small size so that

it does not conflict with other parts of the polygon already constructed. This is always possible since for each vertex of a polygon there exists a disk that intersects the polygon only in the wedge defined by the vertex.

Once all elements of the forest have been processed, we obtain a polygon P whose straight skeleton has the same structure as T , i.e., $\mathcal{S}_d(P) \sim T$. \square

Notice that conditions (G1–G3) do not depend on the order of arcs around nodes; we can construct a polygon for any such order. So in particular if T is an unordered tree that satisfies (G1–G3), then we can pick an arbitrary order and the lemma holds. Hence, we get the following theorem:

Theorem 1. *An (ordered or unordered) directed tree T is the directed straight skeleton of a simple polygon P in general position if and only if T satisfies conditions (G1–G3).*

3 Realizing Trees with Labeled Arcs

Recall that an arc of the straight skeleton is called *reflex* (*convex*) if it was traced out by a reflex (convex) vertex of the wavefront. One can easily see that the construction in Lemma 6 creates a polygon where all arcs of the straight skeleton are convex, with the exception of arcs from leaves to split nodes.

For later constructions (for trees with higher degrees), it will be important that we test not only whether an ordered tree can be realized, but additionally we want to impose onto each arc whether it is reflex or convex in the straight skeleton. We study this question here first for trees with maximum degree 3.

So assume we have a directed tree T that satisfies (G1–G3). Additionally we now label each arc of T with either “reflex” or “convex”, and we ask whether there exists a polygon P that realizes this labeled directed tree in the sense that $\mathcal{S}_d(P) \sim T$ and the type of skeleton-arc in $\mathcal{S}_d(P)$ matches the label of the arc in T . We denote this by $\mathcal{S}_d(P) \sim_\ell T$.

We observe that a peak node is created when a wavefront of three edges, a triangle, collapses. Therefore, all incident arcs at a peak node are convex.

For collapse nodes, we know that the outgoing arc is convex. (Recall that reflex arcs in a straight skeleton are only created in vertex events, which cannot exist when all interior nodes have degree three.) At least one of the incoming arcs needs to be convex, as two reflex wavefront vertices meeting in an event will result in a node of degree at least four.

We have already established that the incoming arc at a split node needs to be reflex. Furthermore, it is easy to see that the two outgoing arcs of a split node need to be convex. We summarize the necessary conditions for a labeled directed tree to correspond to a straight skeleton in the following lemma:

Lemma 7. *Let P be a simple polygon in general position and let T_ℓ be the labeled directed tree such that $\mathcal{S}_d(P) \sim_\ell T_\ell$. Then*

- (L1) *for peak nodes, all incoming arcs are convex;*
- (L2) *for collapse nodes, at least one incoming arc and the outgoing arc are convex;*
- (L3) *for split nodes, the incoming arc is reflex and both outgoing arcs are convex.*

We will now show that (L1–L3) are also sufficient:

Lemma 8. *Any labeled directed tree T_ℓ that meets conditions (G1–G3) from Lemma 1 and (L1–L3) from Lemma 7 is realizable by a simple polygon P .*

Proof. Since reflex arcs in T_ℓ only originate at leaves and terminate at collapse or split nodes (interior nodes never have outgoing reflex arcs, and peak nodes have no incoming reflex arcs), we can create a tree T'_ℓ by replacing each collapse node that has an incoming reflex arc with a leaf and dropping the two incident incoming arcs and their leaves. This resulting tree T'_ℓ will have no reflex arcs except for those leading from a leaf to split nodes by (L1), (L3), and (G3). Thus, we can create a polygon P' such that $\mathcal{S}_d(P') \sim_\ell T'_\ell$ by the process described in the proof of Lemma 6, respecting all labels.

We now obtain P by offsetting P' slightly to the outside. Then, we modify P at each vertex v that corresponds to a leaf in T'_ℓ that was the result of replacing a collapse node of T_ℓ . Note that each such vertex v is convex since the outgoing arc of a collapse node is convex by (L2). We insert a small edge in place of v , replacing it with v_1 , the edge, and v_2 . We choose the angles at v_1 and v_2 such that one of them is reflex and the other is convex, in order to match the labeling of T_ℓ . Figure 4 illustrates this operation.

By making the new edge sufficiently small, we can ensure that these events happen before the wavefront becomes identical to P' , and thus before all remaining events of the wavefront propagation. □

4 Arbitrary Node Degrees

Once we allow for straight skeletons where interior nodes can have degrees larger than three, a number of previous constraints no longer hold. Most importantly, during the wavefront propagation vertex events can happen, resulting in new reflex vertices in the wavefront after the event. Consequently, for instance, split nodes no longer need to be adjacent to leaves. Larger node degrees also result in more complex variants of split, collapse, and peak nodes. Note that we continue to restrict polygons from having parallel edges as those might cause skeleton arcs which have no direction (when they get created as a result of two wavefront edges

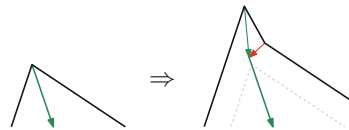


Fig. 4. Extending a convex vertex of P such that a leaf in its tree is replaced by a collapse node where one incoming arc is reflex and one is convex.

crashing into each other) or straight skeleton arcs that are neither reflex nor convex (when two parallel wavefronts moving in the same direction become incident at an event).

In order to understand what combinations of reflex and convex incoming and outgoing arcs may exist at a node in a directed straight skeleton, we study the different shapes that a wavefront may have at an event at locus p and time t . At a time $t - \delta$ immediately prior to the event, the wavefront will consist of a combination of reflex and convex vertices, tracing out reflex and convex arcs, all of which will meet at p at time t . We choose δ sufficiently small such that no event will happen in the interval $[t - \delta, t)$.

Consider the wavefront around a locus p at an event, and consider the *wedges* that have been already swept by the wavefront. With wedge we mean the area near p swept over by a continuous portion of the wavefront polygon until time t .

If a single wedge covers the entire area around p , we call it a *full wedge*. The interior angles of other wedges might be less than π , greater than π , or exactly π as illustrated in Fig. 5. We call the first type of wedge *reflex* and the second type of wedge *convex*, after their corresponding wavefront vertices in the simple case. The third type of wedge we simply call π -wedge.

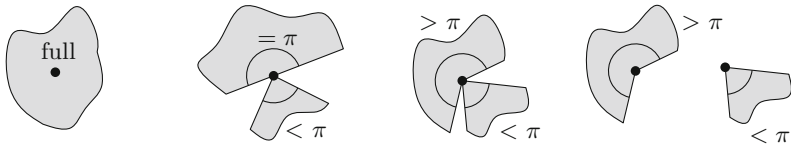


Fig. 5. Wavefront wedges at event times are either full wedges or can be classified by their interior angle. The wedges (already-swept areas) are gray, the remaining white sectors are covered by the wavefront polygon(s).

A single wedge may have been traced out by just one wavefront edge if a wedge at an event has an interior angle of exactly π , but in all other cases it is the area covered by two or more wavefront edges and their incident wavefront vertices, which have traced out one or more incoming arcs at p . Note that all but the two outermost edges of this part of the wavefront collapse at time t .

We will establish *arc-patterns* to describe combinations of arcs at a node. Such a pattern is a string consisting of the types of arcs: r for an incoming reflex arc, c for an incoming convex arc, and \hat{r} and \hat{c} for their outgoing counterparts. We will use operators known from language theory, such as parentheses to group blocks, the asterisk (*) to indicate the preceding character or group may occur zero or more times, the plus sign (+) to indicate the preceding block may occur one or more times, and the question mark (?) to indicate it may exist zero times or once. When defining patterns we give them variable names in capitals.

We now investigate which combination of arcs may trace out which types of wedges. For this purpose we first characterize single wedges and provide their

describing arc-patterns. Then we examine all possible single wedge combinations and provide allowed arc-patterns for interior nodes.

Due to lack of space we give here only an overview on the arc-patterns of single wedges and the arc-patterns for possible combinations of these wedges at interior nodes. For a detailed analysis please see Appendix B (see footnote 1).

As mentioned, single wedges can be reflex wedges, convex wedges, π -wedges, and so called *full wedges*, the latter being traced out by a wavefront that collapses completely around a locus p . The arc-pattern for a reflex wedge is $R := r (cr)^*$, i.e., one reflex vertex, followed by zero or more (convex, reflex) pairs of vertices and thus arcs. If a reflex wedge is created by a single reflex wavefront vertex (and its incident edges) then we call it *trivial*. Otherwise, we call it *non-trivial*, with $R_+ = r (cr)^+$ specifying the arc-pattern of a non-trivial reflex wedge.

The arc-pattern for a convex wedge is $C := r^2 c (r^2 c)^* r^2$. Like for reflex wedges we distinguish trivial (traced out by a single convex vertex) and non-trivial convex wedges, the latter (any pattern matching C and having length at least two) being denoted by C_+ .

The case of π -wedges is related to the case of reflex wedges. A π -wedge is either traced out by exactly one wavefront edge (no incoming arcs), or it has the same pattern as for a non-trivial reflex wedge. Hence, we have as arc-pattern $P := \emptyset | R_+$. Note that since we have explicitly excluded parallel polygon edges for this problem setting, only trivial π -wedges can exist. Therefore, we will set $P := \emptyset$ here.

Last, the arc-pattern of a full wedge is $F := c (r^2 c)^* c (r^2 c)^* c (r^2 c)^*$.

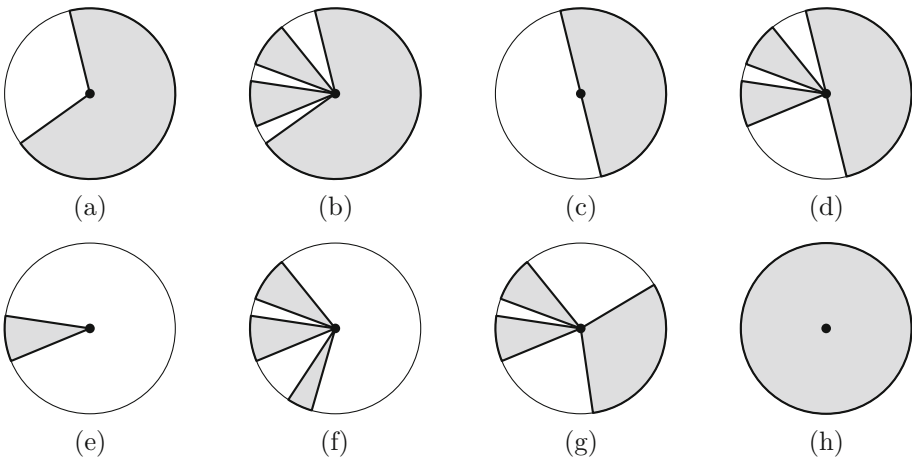


Fig. 6. All wedge combinations possible at a node.

Analyzing possible combinations of single wedges at interior nodes we get seven different allowed arc-patterns (see Fig. 6(a-h)): $N_a := C_+ \hat{c}$, $N_b := C \hat{c} (R \hat{c})^+$, $N_d := P \hat{c} (R \hat{c})^+$, $N_e := R_+ \hat{r}$, $N_f := (R \hat{c})^+ R \hat{r}$, $N_g := (R \hat{c})^+ R \hat{c}$, and

$N_h := F$. (Note that the pattern from Fig. 6(c) will not result in an event without parallel polygon edges, which we have excluded.) A simple split node is matched via N_d , a simple collapse node is handled by N_a (with C_+ being either cc , rc , or cr), and a simple peak node is matched by N_h .

Since these are all possible wavefront/wedge combinations, any interior node of the straight skeleton will have to match $N := N_a|N_b|N_d|N_e|N_f|N_g|N_h$. We can state the following lemma:

Lemma 9. *Let P be a simple polygon and let T_ℓ be the labeled directed tree such that $\mathcal{S}_d(P) \sim_\ell T_\ell$. Then the cyclic order of arcs of any interior node of T_ℓ needs to match the pattern specification N defined above.*

After the necessity of the discussed conditions we now prove their sufficiency.

Lemma 10. *Let T_ℓ be any labeled directed tree for which (i) the cyclic order of arcs of each interior node matches the pattern specification N defined above and (ii) each leaf has out-degree one. Then T_ℓ is realizable by some simple polygon P , that is, $\mathcal{S}_d(P) \sim_\ell T_\ell$.*

Proof. We will construct P in a manner similar to the one described in Lemma 6. We start by identifying maximally connected components C_1, C_2, \dots, C_n of T_ℓ containing nodes with out-degree either zero or one. These components take the place of the subtrees of our forest from Lemma 6, and they are connected in T_ℓ via split-nodes, i.e., nodes with out-degree ≥ 2 .

We pick an arbitrary component C_1 and create a polygon P_1 such that $\mathcal{S}_d(P_1) \sim_\ell C_1$ as follows. If there are no outgoing arcs from C_1 , we start at its unique peak node. Otherwise, there is exactly one outgoing arc, and we begin at the node it is incident to. Constructing a polygon for this node is straightforward by applying the concepts learned from considering convex, reflex, and full wavefront wedges. We proceed by extending this initial polygon step by step like in Lemma 4, treating each reflex or convex vertex of the polygon as a wavefront wedge to be constructed, until we have a polygon for the entire component.

Next, we pick one of the split nodes connected to C_1 . That split node, we call it n , will have one of the forms from Fig. 6 that have at least two white sectors. The polygon we just created will cover one of these white sectors. (Depending on the type of arc connecting C_1 to n , the polygon will either have a reflex or convex vertex for this arc.) We continue by constructing polygons for all remaining white sectors in the same fashion we used for constructing P_1 . Note that this process allows us to force at least one angle, and therefore we can construct polygons that fit into the white sectors for n .

Now we have the wavefront polygon as it should be when the event at n happens. We compute a small exterior offset, joining all polygons into one larger polygon. The new reflex or convex vertices at this point are then further subdivided as required by the incoming arcs for n .

We repeat this process until we have covered all split nodes and thus all components and have thereby created a polygon whose structure matches T_ℓ . \square

Please see Appendix C in the arXiv-version of this work [3].

Combining Lemmas 9 and 10, we obtain the following theorem:

Theorem 2. *An ordered labeled directed tree T_ℓ is the directed straight skeleton of a simple polygon P without parallel edges if and only if (i) the cyclic order of arcs of each interior node of T_ℓ matches the pattern specification N defined above and (ii) each leaf has out-degree one.*

5 Conclusion

In this work we developed a complete characterization of necessary and also sufficient conditions such that a given directed and labeled ordered tree can be represented as the straight skeleton of a simple polygon. This extends previous work on representing trees via related geometric structures [4, 12].

We leave the algorithmic question – how efficient suitability of a given input tree can be tested and, in case of an affirmative answer, a corresponding simple polygon can be computed – for future research. We conjecture that both is possible in time linear in the size of the given tree.

References

1. Aichholzer, O., Aurenhammer, F.: Straight skeletons for general polygonal figures in the plane. In: Samoilenko, A. (ed.) *Voronoi's Impact on Modern Sciences II*, vol. 21, pp. 7–21. Institute of Mathematics of the National Academy of Sciences of Ukraine, Kiev, Ukraine (1998)
2. Aichholzer, O., Aurenhammer, F., Alberts, D., Gärtner, B.: A novel type of skeleton for polygons. *J. Univ. Comput. Sci.* **1**(12), 752–761 (1995)
3. Aichholzer, O., Biedl, T., Hackl, T., Held, M., Huber, S., Palfrader, P., Vogtenhuber, B.: Representing directed trees as straight skeletons [cs.CG] (2015). <http://arxiv.org/abs/1508.01076>
4. Aichholzer, O., Cheng, H., Devadoss, S.L., Hackl, T., Huber, S., Li, B., Risteski, A.: What makes a tree a straight skeleton? In: *Proceedings of the 24th Canadian Conference on Computational Geometry, (CCCG 2012)*, pp. 253–258. Charlottetown, PE, Canada (2012)
5. Biedl, T., Held, M., Huber, S.: Recognizing straight skeletons and Voronoi diagrams and reconstructing their input. In: Gavrilova, M., Vyatkina, K. (eds.) *Proceedings of the 10th International Symposium on Voronoi Diagrams in Science & Engineering (ISVD 2013)*, pp. 37–46. IEEE Computer Society, Saint Petersburg, Russia (2013)
6. Chalopin, J., Gonçalves, D.: Every planar graph is the intersection graph of segments in the plane (Extended Abstract). In: *Proceedings of 41st Annual ACM Symposium Theory Computing (STOC 2009)*, pp. 631–638. ACM, Bethesda, MD, USA (2009)
7. Di Battista, G., Lenhart, W., Liotta, G.: Proximity drawability: a survey. In: Tamassia, R., Tollis, I.G. (eds.) *GD '94. LNCS*, vol. 894, pp. 328–339. Springer, Princeton, NJ, USA (1995)
8. Dillencourt, M.B., Smith, W.D.: Graph-theoretical conditions for inscribability and delaunay realizability. *Discrete Math.* **161**(1–3), 63–77 (1996)

9. Eppstein, D., Erickson, J.: Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. *Discrete Comput. Geom.* **22**(4), 569–592 (1999)
10. Huber, S., Held, M.: A fast straight-skeleton algorithm based on generalized motorcycle graphs. *Int. J. Comput. Geom. Appl.* **22**(5), 471–498 (2012)
11. Liotta, G., Lubiw, A., Meijer, H., Whitesides, S.: The rectangle of influence drawability problem. *Comput. Geom.* **10**(1), 1–22 (1998)
12. Liotta, G., Meijer, H.: Voronoi drawings of trees. *Comput. Geom.* **24**(3), 147–178 (2003)