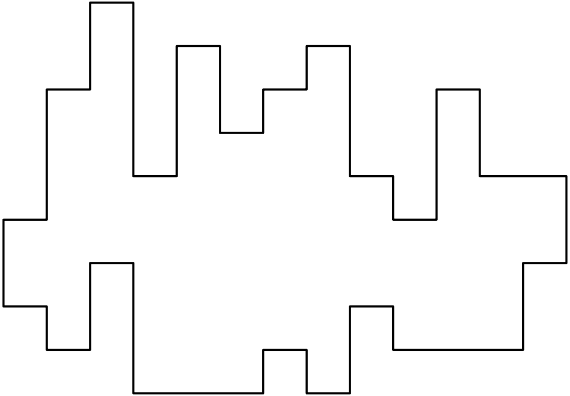
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

Computing the **Straight Skeleton** of an **Orthogonal Monotone Polygon** in **Linear Time**

Günther Eder, Martin Held, and Peter Palfrader

Preliminaries

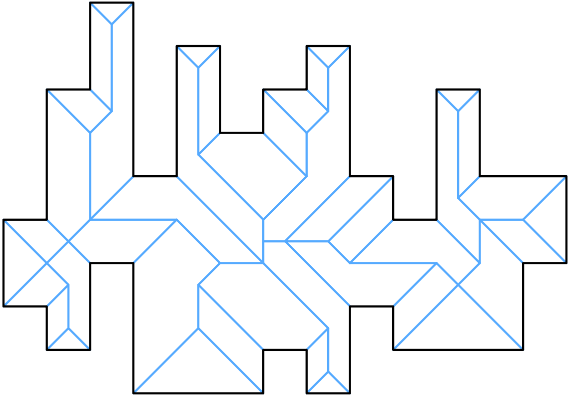
- P is an orthogonal x-monotone polygon with n vertices.



Preliminaries

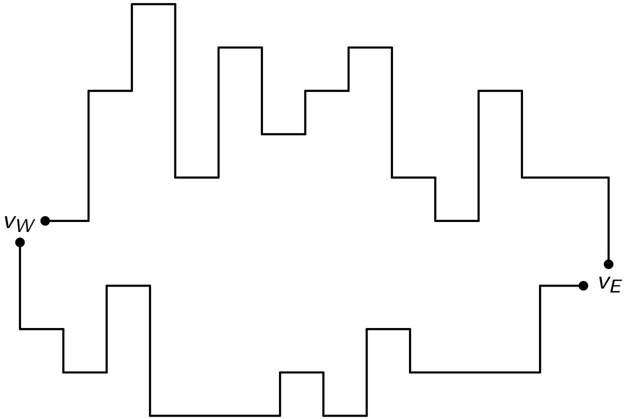
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

- P is an orthogonal x-monotone polygon with n vertices.
- $\mathcal{S}(P)$ denotes the straight skeleton of P.



Preliminaries

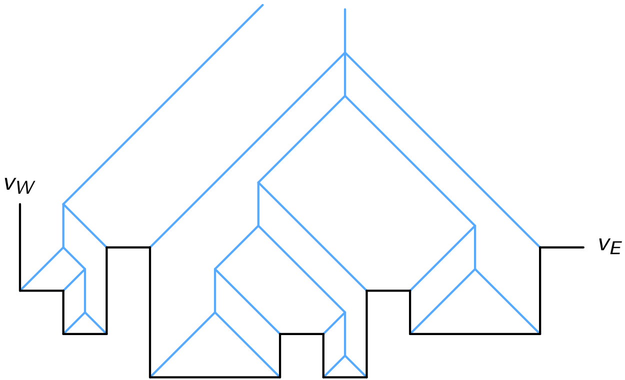
- P is an orthogonal x-monotone polygon with n vertices.
- $\mathcal{S}(P)$ denotes the straight skeleton of P.
- We split P into its upper and lower monotone chain.



Preliminaries

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

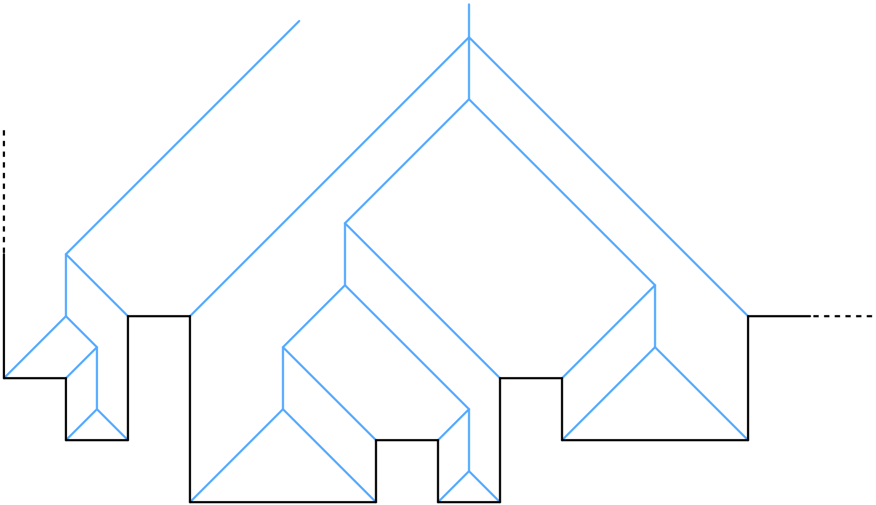
- P is an orthogonal x-monotone polygon with n vertices.
- $\mathcal{S}(P)$ denotes the straight skeleton of P.
- We split P into its upper and lower monotone chain.
- Looking at a single chain C, let $\mathcal{S}(C)$ denote its straight skeleton.



Algorithm Setup

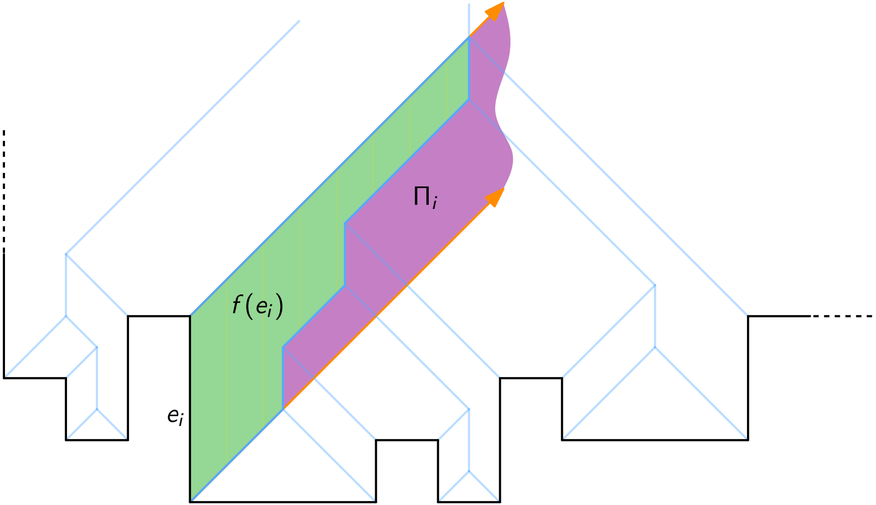
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

The arcs of $\mathcal{S}(C)$ have only three directions: $\binom{1}{1}$, $\binom{-1}{1}$, and $\binom{0}{1}$.



Algorithm Setup

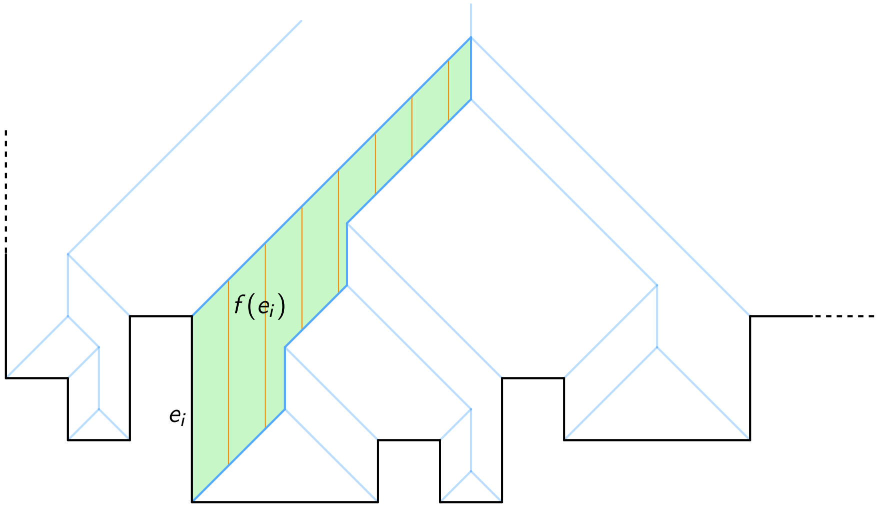
A face $f(e_i)$ of $\mathcal{S}(C)$ lies inside of the half-plane slab Π_i.



Algorithm Setup

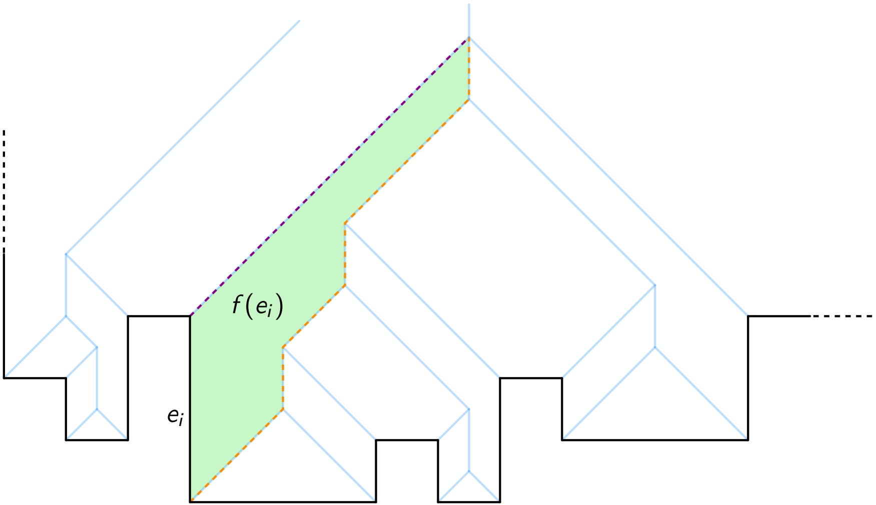
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

Also, $f(e_i)$ is monotone in respect its input edge as well as to a line perpendicular to it.



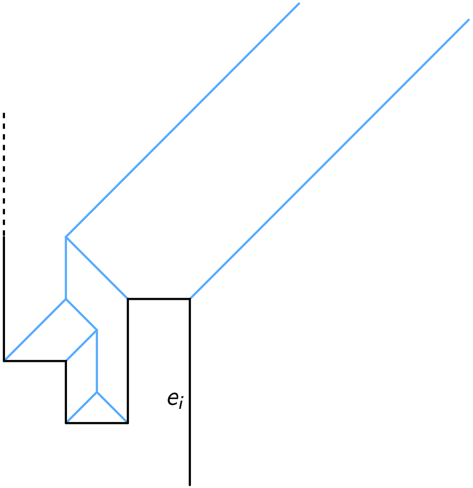
Algorithm Setup

Let us separate $f(e_i)$ into its left and right chain.



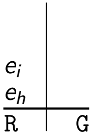
Algorithm Setup

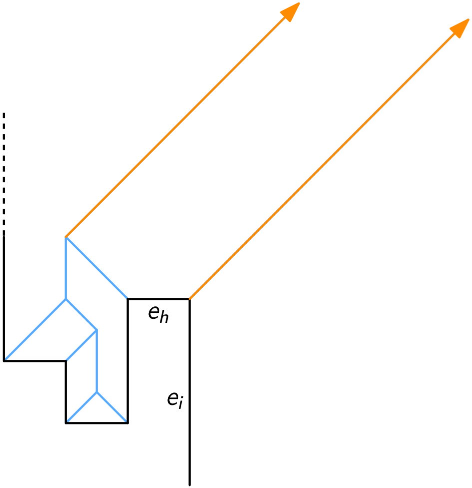
We maintain the partial straight skeleton $\mathcal{S}^*$ during our incremental construction.
It contains the left chains of all edges already inserted



Algorithm Setup

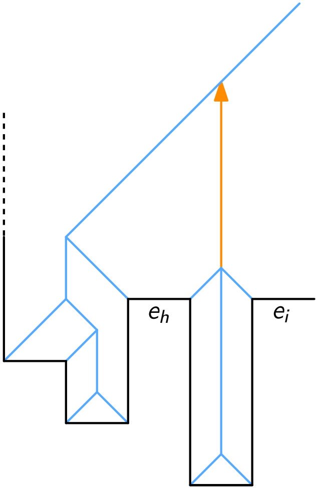
We maintain the partial straight skeleton $\mathcal{S}^*$ during our incremental construction.
It contains the left chains of all edges already inserted, as well as two stacks R





Algorithm Setup

We maintain the partial straight skeleton $\mathcal{S}^*$ during our incremental construction.
It contains the left chains of all edges already inserted, as well as two stacks R and G.



e_h	e_i
R	G

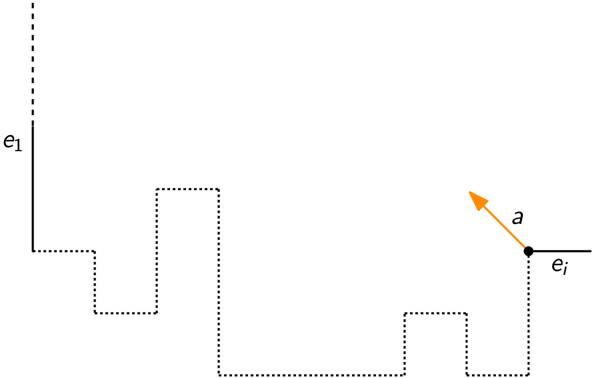
Constructing $\mathcal{S}(C)$

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

We start our incremental construction by adding e_1.

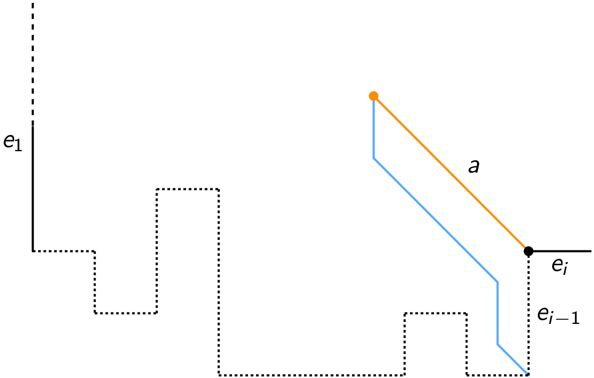
Constructing $\mathcal{S}(C)$

The first arc a of the left chain of $f(e_i)$ has $\binom{1}{1}$ or $\binom{-1}{1}$ direction.



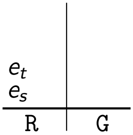
Constructing $\mathcal{S}(C)$

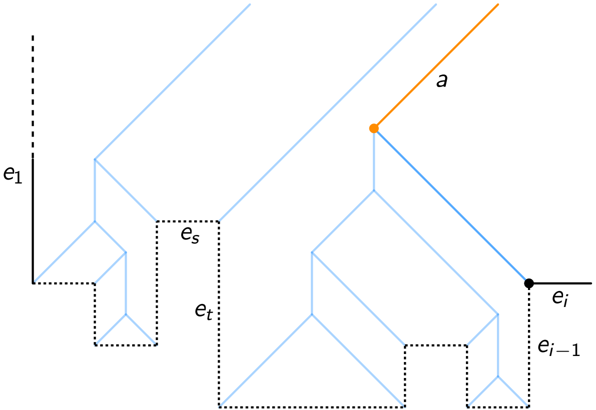
The first arc a of the left chain of $f(e_i)$ has $\binom{1}{1}$ or $\binom{-1}{1}$ direction.
It connects to the end of $f(e_{i-1})$'s left chain.



Constructing $\mathcal{S}(C)$

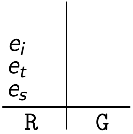
Subsequent arcs between e_i and the edge on top of R.

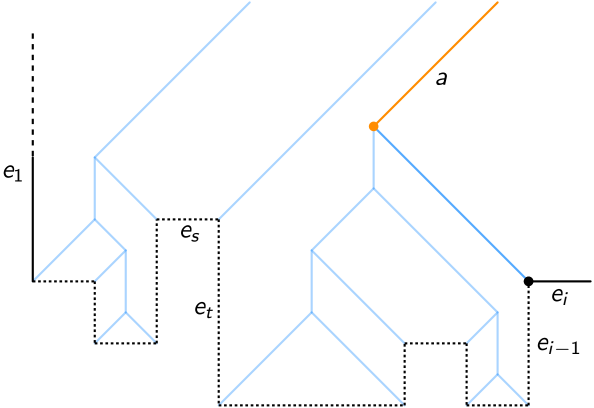




Constructing $\mathcal{S}(C)$

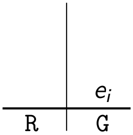
Subsequent arcs between e_i and the edge on top of R. The last arc of a chain ends in a ray,

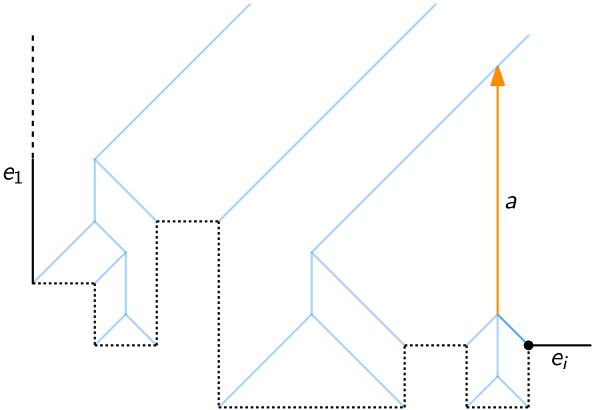




Constructing $\mathcal{S}(C)$

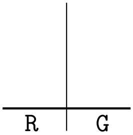
Subsequent arcs between e_i and the edge on top of R. The last arc of a chain ends in a ray, unfinished ghost arc,

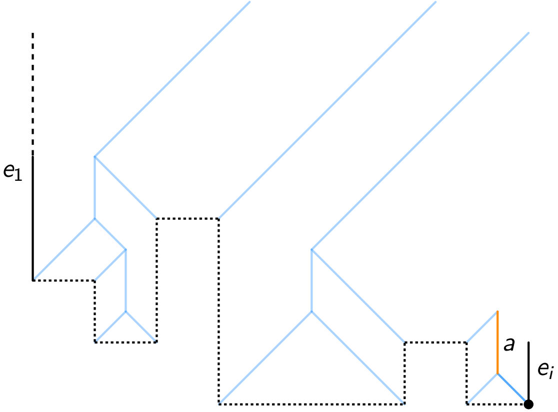




Constructing $\mathcal{S}(C)$

Subsequent arcs between e_i and the edge on top of R. The last arc of a chain ends in a ray, unfinished ghost arc, or bounded vertical arc.

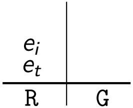


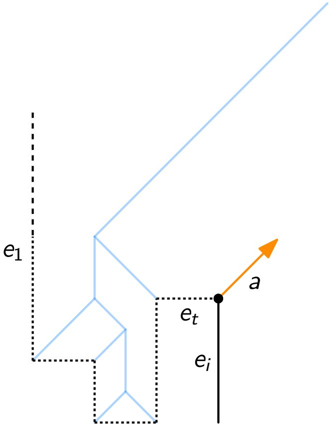


Arc a has $\binom{1}{1}$ Direction

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

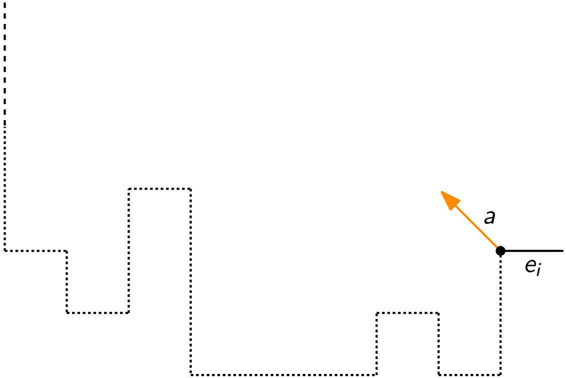
We follow with a case distinction for the next arc a added in the left chain of e_i.
Arc a is a ray and we push e_i onto R.





Arc a has $\binom{-1}{1}$ Direction

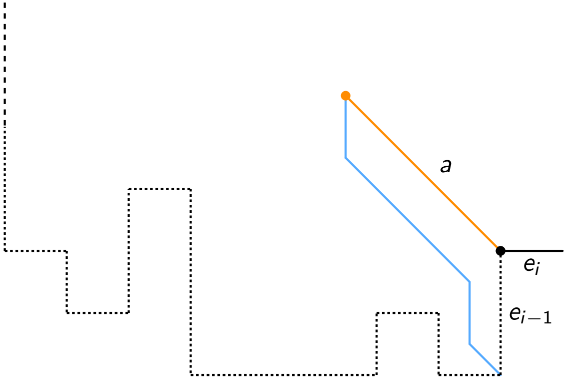
Arc a is either a bounded arc or a ray.



Arc a has $\binom{-1}{1}$ Direction

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

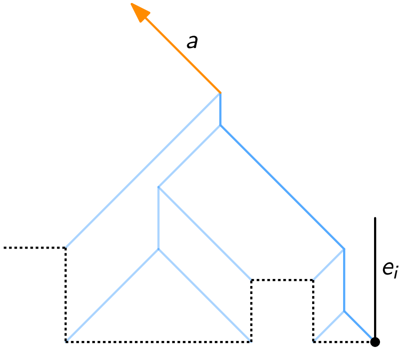
If the left chain of e_{i-1} terminates in a bounded arc, and a is the first arc on the left chain of e_i, it ends where the left chain of e_{i-1} ends.



Arc a has $\binom{-1}{1}$ Direction

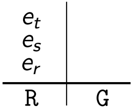
Otherwise, we look at e_t at the top of R. If e_t does not terminate in a $\binom{1}{1}$ ray, a is a $\binom{-1}{1}$ ray, e_i is pushed onto R, and the chain is completed.

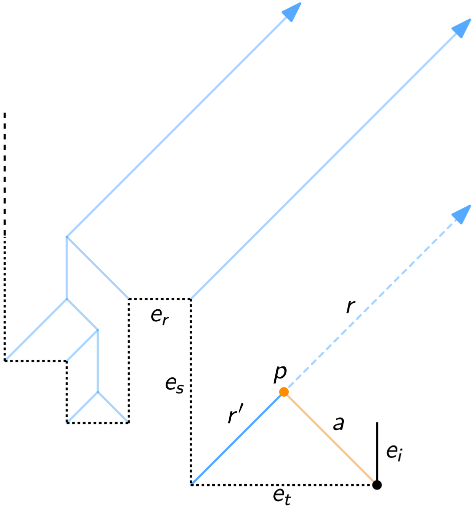
R	G
e_i	



Arc a has $\binom{-1}{1}$ Direction

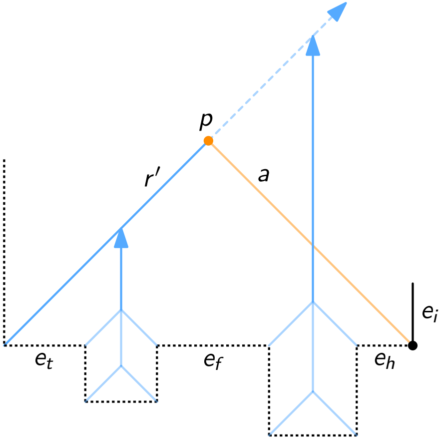
Otherwise, the left chain of e_t terminates in a $\binom{1}{1}$ ray r. At p arc a intersects ray r. In $f(e_{i-1})$ we modify r into a bounded arc r' that ends at p, where a ends as well.





Arc a has $\binom{-1}{1}$ Direction

Finally we have to process the elements of G below r' and a.

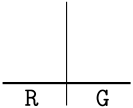


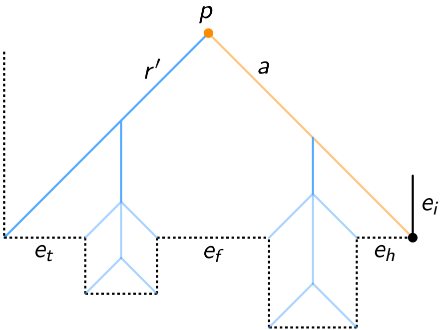
R	G
e_t	e_h
	e_f

Arc a has $\binom{-1}{1}$ Direction

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

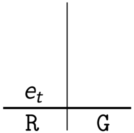
Finally we have to process the elements of G below r' and a.

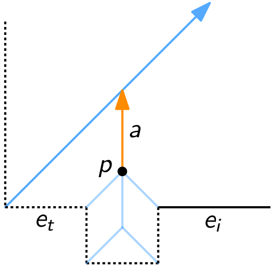




Arc a has $\binom{0}{1}$ Direction

Arc a is either a ghost arc or bounded vertical arc, starting at a point p.

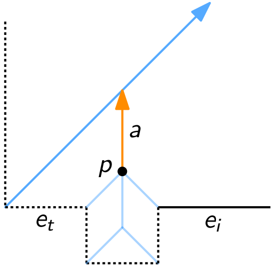




Arc a has $\binom{0}{1}$ Direction

Arc a is either a ghost arc or bounded vertical arc, starting at a point p. In case a is a ghost arc we push e_i onto `G`.

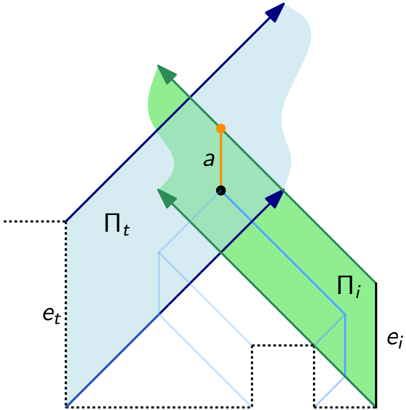
e_t	e_i
`R`	`G`



Arc a has $\binom{0}{1}$ Direction

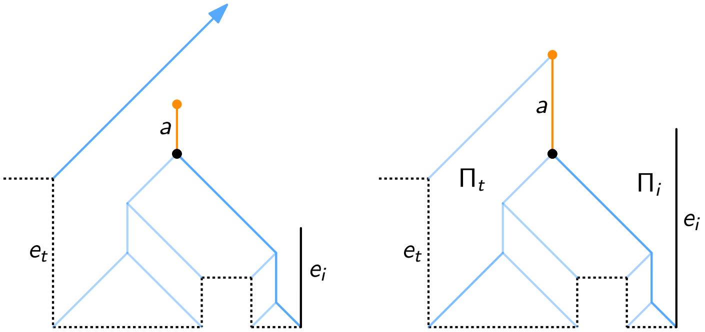
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

Otherwise, a is the line segment from p that is contained in both Π_t and Π_i.



Arc a has $\binom{0}{1}$ Direction

Otherwise, a is the line segment from p that is contained in both Π_t and Π_i.



Finalizing $\mathcal{S}(C)$

- We process the elements that remain on `G`.

Finalizing $\mathcal{S}(C)$

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

- We process the elements that remain on `G`.
- All arcs inserted intersect only rays or ghost arcs.

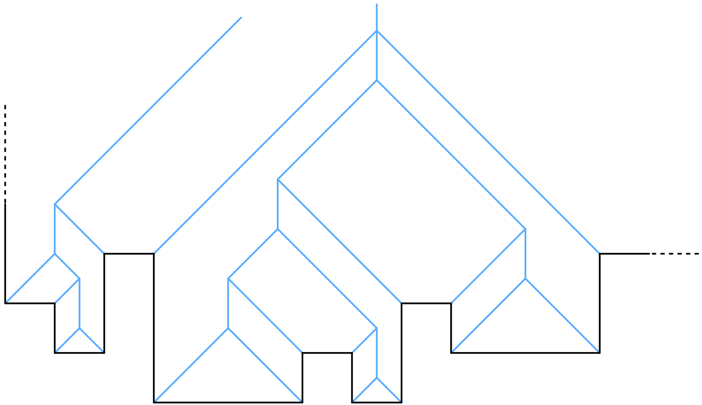
Finalizing $\mathcal{S}(C)$

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab

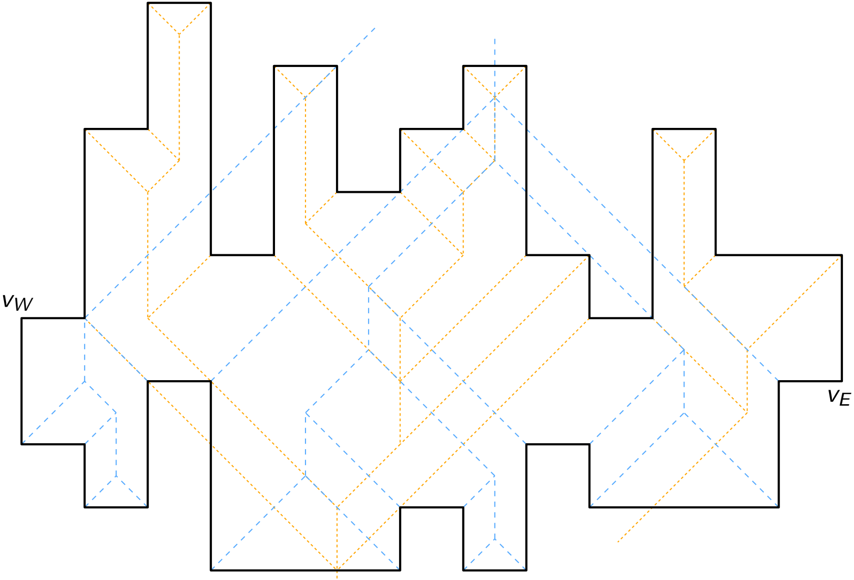
- We process the elements that remain on `G`.
- All arcs inserted intersect only rays or ghost arcs.

Theorem

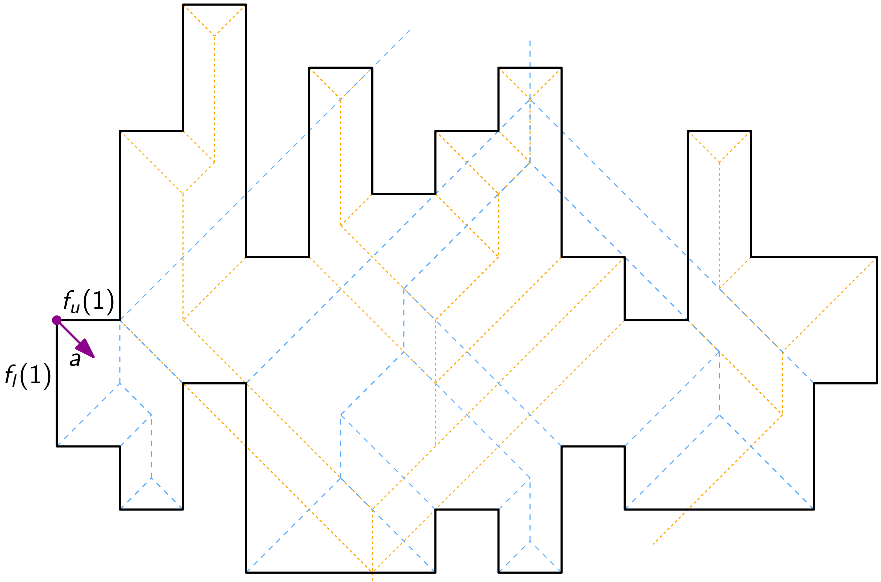
Our incremental construction approach creates $\mathcal{S}(C)$ in $\mathcal{O}(n)$ time.



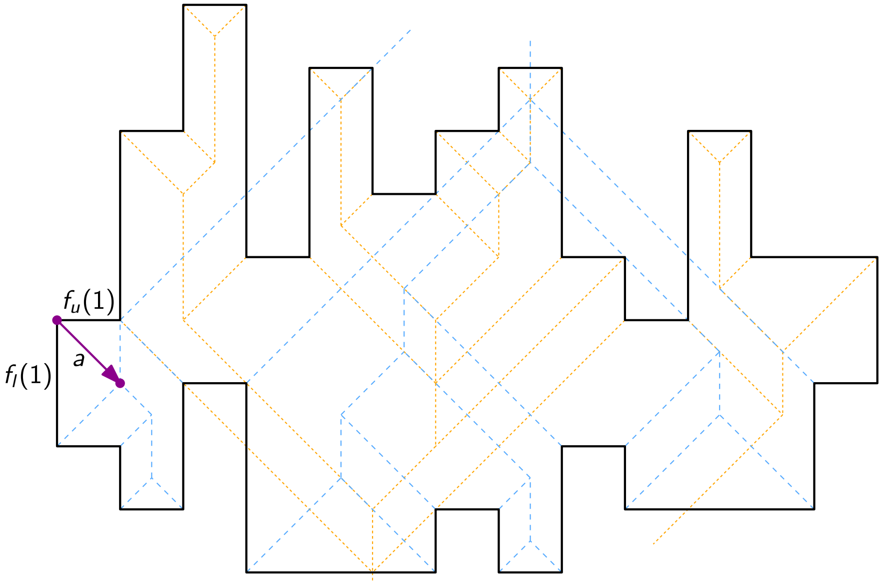
Skeleton Merging



Skeleton Merging

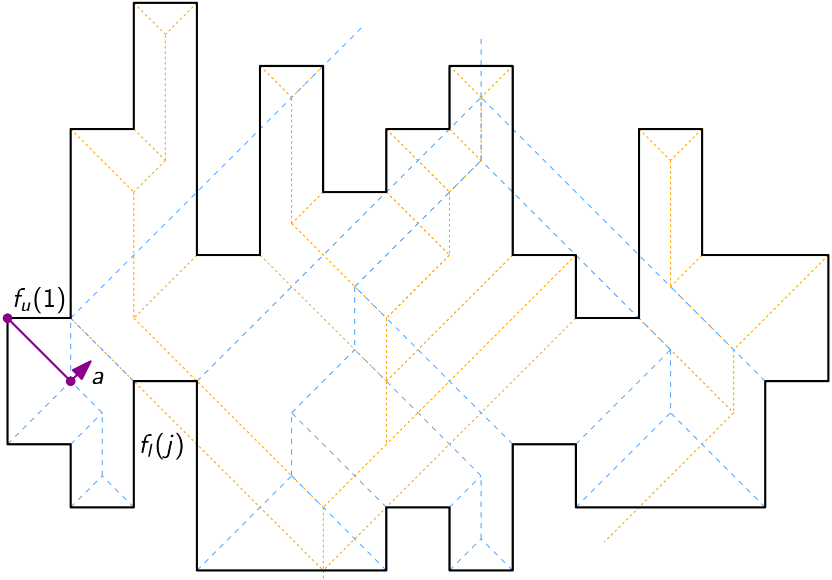


Skeleton Merging

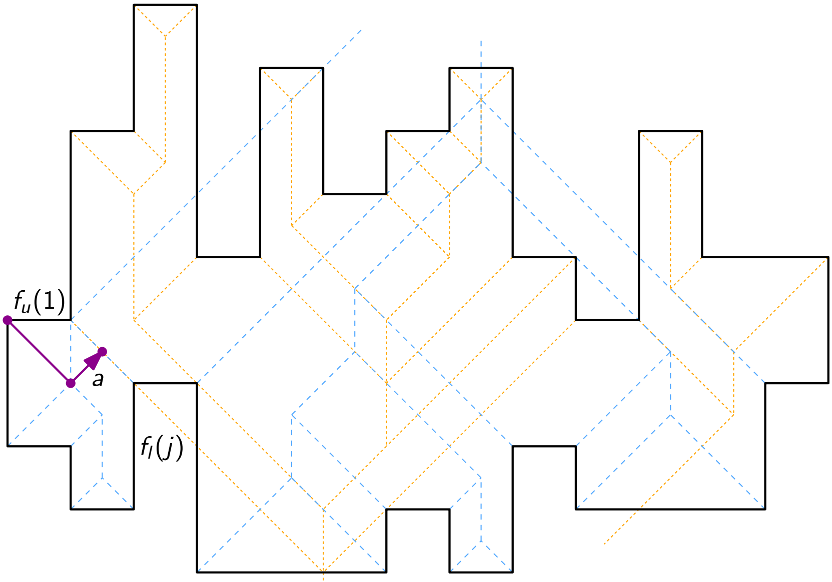


Skeleton Merging

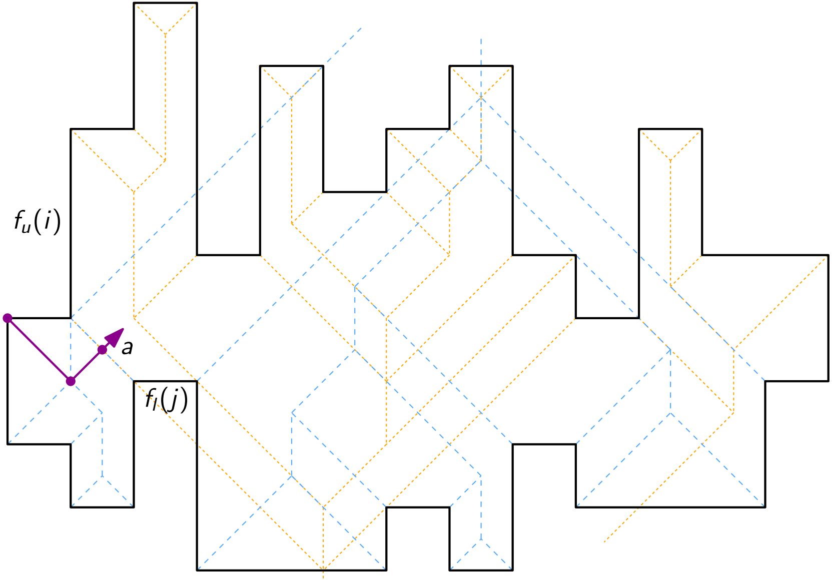
UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab



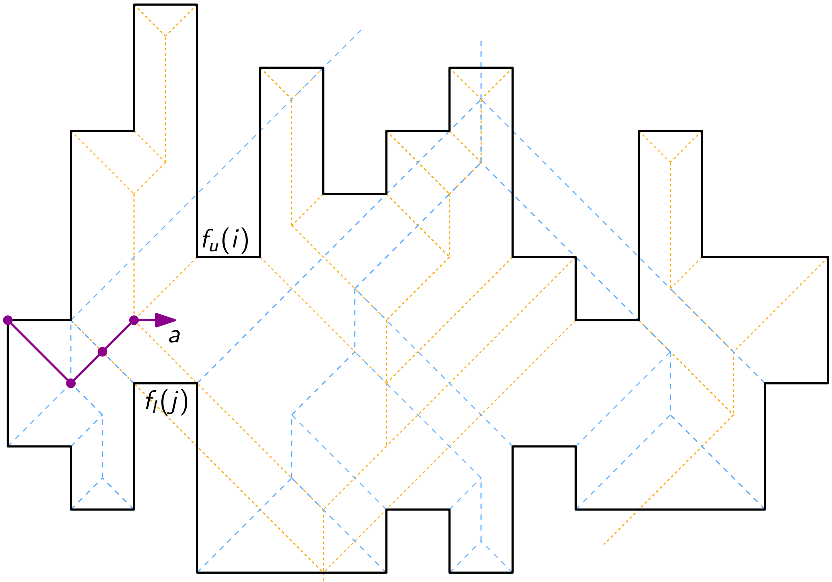
Skeleton Merging



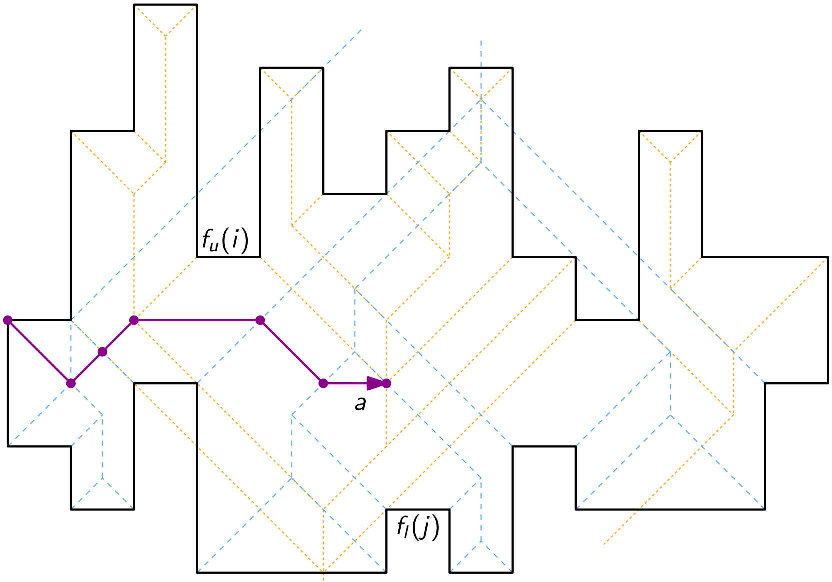
Skeleton Merging



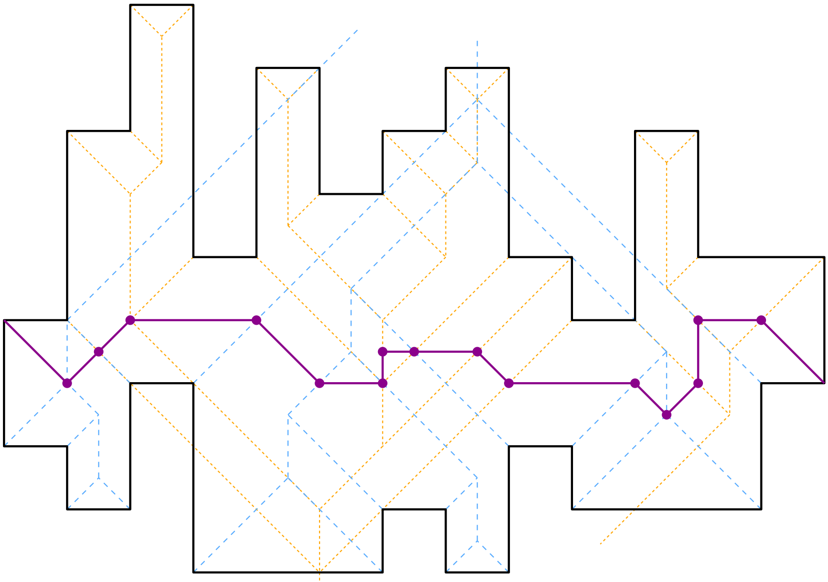
Skeleton Merging



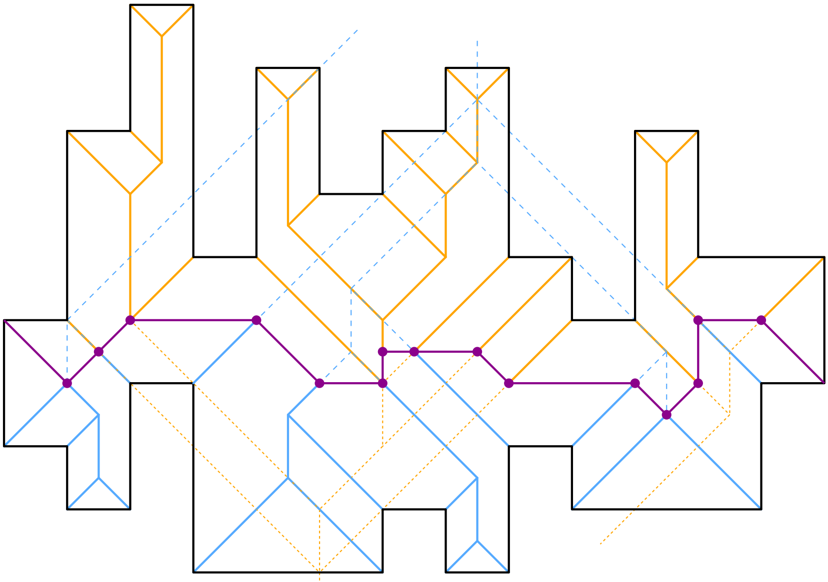
Skeleton Merging



Skeleton Merging

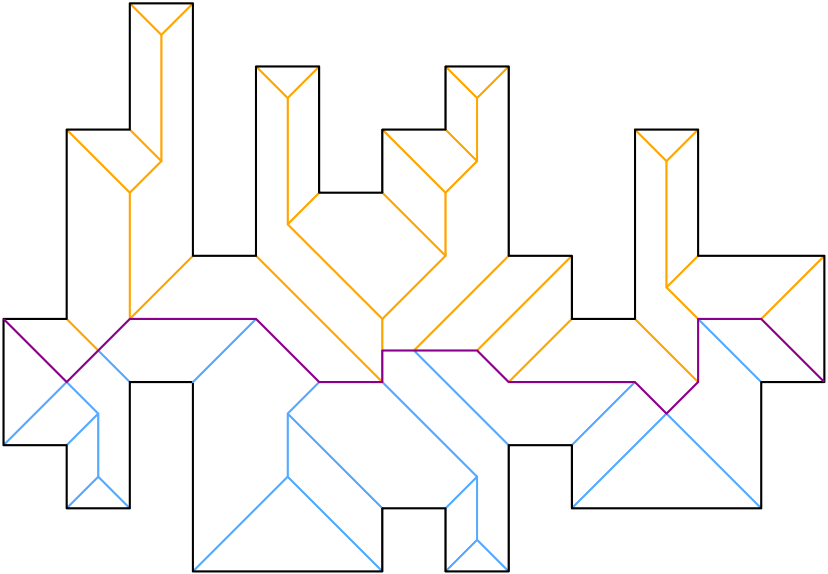


Skeleton Merging



Skeleton Merging

UNIVERSITY OF SALZBURG
Computational Geometry and Applications Lab



Summary

- Incremental construction of $\mathcal{S}(C)$ in linear time.
- Merge of both straight skeletons in linear time.

Questions?