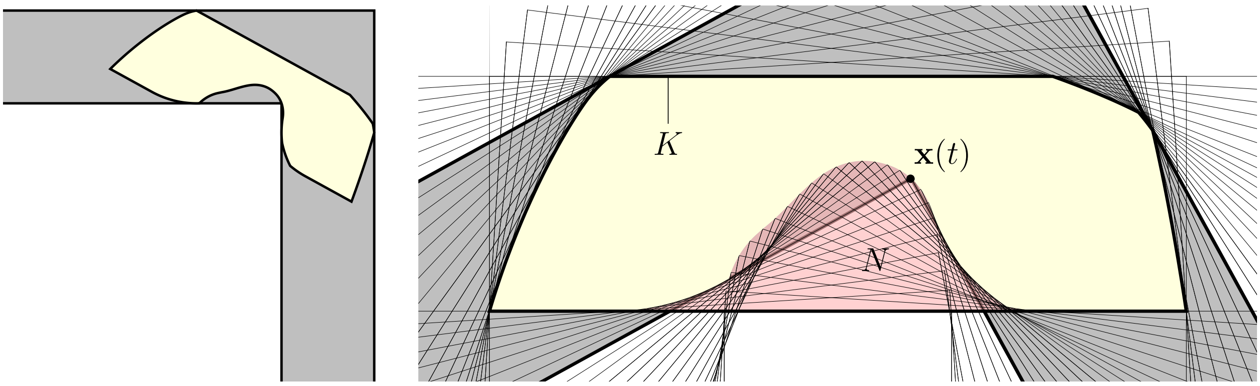


A sofa can be seen as the intersection of rotating hallways (see the figure above). So by taking a finite number of hallways and taking the polygonal intersection S, we get an approximation of the sofa with a larger area. Let Θ be a finite set of angles such that $\{0, \pi/2\} \subseteq \Theta \subseteq [0, \pi/2]$. That is, Θ is a partition

$$0 = \theta_0 < \theta_1 < \theta_2 < \cdots < \theta_n = \pi/2$$

of the interval $[0, \pi/2]$. `SofaDesigner` computes the set of all *monotone hallway intersections* S with the set of angles Θ and an area at least $2.2195\ldots$. Essentially, S is the polygonal intersection of the finite number of hallways of width 1 rotated by each angle of Θ, approximating a sofa from above (see the figure below).

`SofaDesigner` produces computer-assisted proofs of mathematical facts about an arbitrary intersection S. It generates proofs in computer files, so that the files representing the proofs can be verified separately by a verifier independent from `SofaDesigner`. The purpose of this is to ensure that the correctness of the proof do not depend on the correctness of the specific implementation of `SofaDesigner` (which consists of thousand lines of C++), and to enable cross-validation by multiple independent parties.

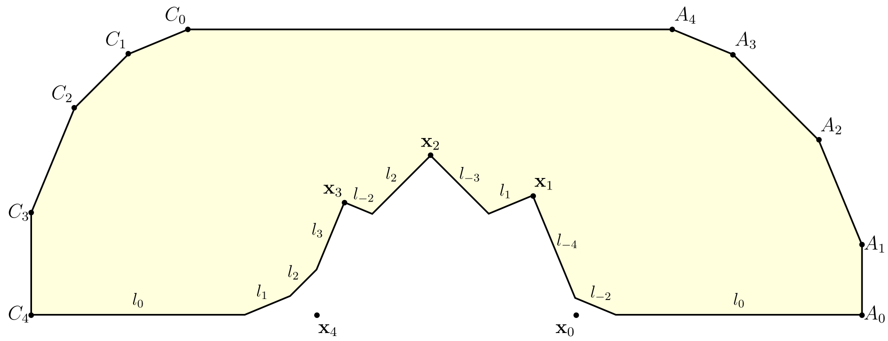
In particular, we have the following roles for `SofaDesigner` and its verifier.

- `SofaDesigner`
    1. Computes the set $\mathcal{S}$ of all monotone hallway intersections S with an area at least $2.2195\ldots$, and stores it as a tree structure $\mathcal{T}$ in a JSON format.

2. Given $\mathcal{T}$ and an inequality I on an arbitrary intersection S in $\mathcal{S}$, produces a proof in a custom JSON format.

- The verifier
  1. Verifies that the JSON-formatted tree $\mathcal{T}$ accurately represents the set $\mathcal{S}$ of all monotone hallway intersections S with an area at least $2.2195\ldots$.
  2. Verifies the proof that an inequality I on $\mathcal{S}$ from the JSON-formatted proof generated by `SofaDesigner`.

Definition of Monotone Hallway Intersection S

> **Figure [polygon-hallway-intersection].** An intersection S with five hallways ($n = 4$) and edge order $\mathbf{i}_K = [0, 1, 2, 3, -2, 2, -3, 1, -4, -2, 0]$.
>
> 
>

We first provide the precise definition and shape of a monotone hallway intersection S with angle set Θ. All the details in this section is important to ensure that the verification process is correct.

$H = \mathbb{R} \times [0, 1]$ is the horizontal strip of height 1. L is the right-angled hallway of width 1, with inner corner at $\mathbf{x} = (0, 0)$ and outer corner at $\mathbf{y} = (1, 1)$. Mathematically, $L = (-\infty, 1] \times [0, 1] \cup [0, 1] \times (-\infty, 1]$. Let a and b be the lines $x = 1$ and $y = 1$ passing through the outer corner $\mathbf{y}$ representing the outer walls of L. Let c and d be the half-lines $(-\infty, 0] \times \{0\}$ and $\{0\} \times (-\infty, 0]$ emanating from the inner corner $\mathbf{x}$ representing the inner walls of L.

For all $0 \leq i \leq n$, let $u_i = (\cos\theta_i, \sin\theta_i)$ and $v_i = (-\sin\theta_i, \cos\theta_i)$ so that the vectors u_i and v_i form a orthogonal basis. Also, let $v_{i+n} = -u_i$ for all $0 \leq i \leq n$ so that v_i is defined for all indices i from 0 to $2n$. For any $0 \leq i \leq n$, let L_i be a copy of the hallway L rotated counterclockwise by an angle of θ_i and translated so that the inner corner is positioned at $\mathbf{x}_i$. Here, the set S is the intersection of the horizontal strip H and the rotated hallways L_i. Let a_i, b_i, c_i and d_i be the sides of the hallway L_i corresponding to the sides $a, b, c,$ and d of L.

For the intersection $S = H \cap \bigcap_{i=0}^{n} L_i$ to be a monotone hallway intersection, we require the following conditions.

- (WLOG) Without loss of generality, by translating S horizontally we assume that $L_0 = L$ so that $\mathbf{x}_0 = (0, 0)$. Also, we assume that $\mathbf{x}_n$ has the y-coordinate equal to zero. Consequently, the sides a_n and c_0 are equal to the line $y = 1$.
- (Monotonicity) The sides $y = 0$ and $y = 1$ of H and the outer walls a_i and c_i of L_i for any $0 \leq i \leq n$ should determine all the sides of a convex polygon K called the *cap* of S. Although we allow some edges of K to degenerate and have zero length, all the lines $a_0, a_1, \dots, a_{n-1}, a_n = c_0, c_1, \dots, c_n$, and $y = 0$ should form the edges of K in counterclockwise order.

Consequently, the sides a_n and c_0 are equal to the line $y = 1$ and determines the upper horizontal edge of K. The sides $a_0, a_1, \dots, a_{n-1}$ forms the different edge of K on the right side in counterclockwise order, and the sides $c_1, c_2, \dots, c_n$ forms the different edges of K on the right side in counterclockwise order.

For any $0 < i \leq n$, let A_i be the intersection of a_{i-1} and a_i. Also let A_0 be the intersection of a_0 and the line $y = 0$. Likewise, for any $0 \leq i < n$, let C_i be the intersection of c_i and c_{i+1}. Also let C_n be the intersection of a_0 and the line $y = 0$. Then the points $A_0, A_1, \dots, A_n$ forms the vertices of K on the right side, and the points $C_0, C_1, \dots, C_n$ forms the vertices of K on the left side. For any $0 \leq i \leq n$, define $A_{n+1+i} = A_i$ so that the list $A_0, A_1, \dots, A_{2n+1}$ forms all the vertices of K in the counterclockwise order.

For any $0 \leq i \leq n$, define the support function values $p_i = A_i \cdot u_i$ and $q_i = C_i \cdot v_i$ of the cap K. Note that we also have $p_i = \mathbf{x}_i \cdot u_i + 1$ and $q_i = \mathbf{x}_i \cdot v_i + 1$. Note that as the sides a_n and c_0 are equal to the line $y = 1$, we have $p_n = q_0 = 1$. $0 \leq i \leq n$, define $p_{n+i} = q_i$ so that the list $p_0, p_1, \dots, p_{2n}$ is a concatenation of $p_0, p_1, \ldots, p_n$ and $q_0, q_1, \ldots, q_n$. For every $0 \le i \le 2n$, define the side lengths $s_i = (A_{i+1} - A_i) \cdot v_i$ of K.

Coordinate Systems of S

A monotone hallway intersection S is completely determined by its cap K. So the state of S can be described by any coordinate system representing the cap K. There are three coordinate systems we can use for K.

- For every $0 \le i \le n$, the inner corners $\mathbf{x}_i = (x_i, y_i) \in \mathbb{R}^2$ of the hallways with $\mathbf{x}_0 = (x_0, y_0) = (0, 0)$ and the y-coordinate y_n of $\mathbf{x}_n$ equal to zero. With the constraints we have a total of $2n - 1$ free coordinates $x_1, \ldots, x_n, y_1, \ldots, y_{n-1}$ to determine.
- The support function values $p_0, p_1, \ldots, p_n$ and $q_0, q_1, \ldots, q_n$ of the right and left part of the cap, with constraints $p_0 = p_n = q_0 = 1$. In this coordinate, we have $2n - 1$ free values $p_1, \ldots, p_{n-1}$ and $q_1, \ldots, q_n$ to determine.
- The edge lengths $s_0, s_1, \ldots, s_{2n}$ of cap K, with constraints $\sum_{i=0}^{n} \cos(\theta_i) s_i = 1$ and $\sum_{i=0}^{n} \sin(\theta_i) s_{n+i} = 1$ to ensure that the height of K is equal to 1. Using the two equalities, we can eliminate the variables $s_0 = 1 - \sum_{i=1}^{n} \cos(\theta_i) s_i$ and $s_{2n} = 1 - \sum_{i=0}^{n-1} \sin(\theta_i) s_{n+i}$ and have $2n - 1$ free variables $s_1, s_2, \ldots, s_{2n-1}$.

All three coordinate systems are convertible to each other. For example, the coordinates of $\mathbf{x}_i$ determine the values $p_i = \mathbf{x}_i \cdot u_i + 1$ and so the lines $a_i = \left\{ z \in \mathbb{R}^2 : z \cdot u_i = p_i \right\}$ and the intersections $A_i = a_{i-1} \cap a_i$. The side lengths are then recoverable from the formula $s_i = (A_{i+1} - A_i) \cdot v_i$. In this way, we can express the edge lengths $s_0, s_1, \ldots, s_{2n}$ from the coordinates of $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$. We can also work the other way and get the coordinates of $\mathbf{x}_i$ from the edge lengths $s_0, s_1, \ldots, s_{2n}$. Start with $A_0 = (0, 0)$, and use the formula $A_{i+1} = A_i + s_i v_i$ to retrieve all the vertices $A_0, \ldots, A_{2n}$ of K, the support function values $p_i = A_i \cdot u_i$ and $q_i = A_{i+n} \cdot v_n$, and then $\mathbf{x}_i = (p_i - 1) u_i + (q_i - 1) v_i$. Note that all the conversions are affine-linear. So any of the three coordinate systems determine a cap K.

`SofaDesigner` chooses the system of support function values $p_1, \ldots, p_{n-1}$ and $q_1, \ldots, q_n$. Let $\mathbf{p} = (p_1, \ldots, p_{n-1}, q_1, \ldots, q_n)$ be the list of free variables in the given order. We also require each side length s_i to be nonnegative in order for K to be a proper cap. For any $0 \leq i \leq 2n$, let E_i be the inequality $s_i \geq 0$ where s_i is written in terms of $\mathbf{p}$. Define $\mathcal{K}_\Theta$ as the domain of $2n-1$ free real variables $\mathbf{p}$ in support function coordinates with the linear constraints E_i for all $0 \leq i \leq n$. Then $\mathcal{K}_\Theta$ is the space of all caps of monotone hallway intersections with angle set Θ. The goal of `SofaDesigner` is to understand the subset of $\mathcal{K}_\Theta$ where the cap gives rise to the monotone hallway intersection $S = H \cap \bigcap_{i=0}^n L_i$ of area at least 2.2195.

Area of S by its Niche $\mathcal{N}(K)$

Let $\mathcal{A}(K)$ be the area of the monotone hallway intersection S with cap $K \in \mathcal{K}_\Theta$. The goal of `SofaDesigner` is to understand the area functional $\mathcal{A} : \mathcal{K}_\Theta \to \mathbb{R}$ completely. Define the *niche* $\mathcal{N}(K)$ of the monotone hallway intersection S as the region bounded from below by the line $y = 0$, and bounded from above by the upper envelope of the line $y = 0$ and the half-lines c_i and d_i for all $0 \leq i \leq n$. For any set X, denote its area as $|X|$. Then the area $|S|$ of S is equal to $\mathcal{A}(K) = |K| - \mathcal{N}(K)$ because S is the cap K subtracted by the union of inner corners $\mathcal{N}(K)$. Note that the area $|K|$ of cap is a quadratic polynomial in terms of the coordinates of $\mathcal{K}_\Theta$ (one can for example use the shoelace formula). However, the area of the niche $\mathcal{N}(K)$ is much harder to understand.

If the order of edges appearing in the niche $\mathcal{N}(K)$ is determined, we can understand the area $|\mathcal{N}(K)|$ as a quadratic polynomial of the coordinates of K. For every $0 \leq i \leq n$, define the line l_i as the line extending the half-line d_i, and the line l_{-i} as the line extending the half-line b_{n-i} (note that l_0 is equal to the line $y = 0$ in both cases). Let $\mathbf{i}_K$ be the list of all the indices i of the lines l_i appearing in the niche $\mathcal{N}(K)$ from left to right. So $\mathbf{i}_K$ is either a list of length one with only one zero, or a list with length at least two starting and ending with zero. If $-n \leq i, j \leq n$ are different indices, define the point $p_{i,j}$ as the intersection of the lines l_i and l_j. Then the coordinates of points $p_{i,j}$ are affine-linear with respect to the coordinates $\mathbf{p}$ of $\mathcal{K}_\Theta$. If $\mathbf{i}_K$ is the list $i_1, i_2, \dots, i_m$, then the vertices of the niche are exactly $p_{i_1,i_2}, p_{i_2,i_3}, \dots, p_{i_{m-1},i_m}$ from left to right with the first and last vertex on the line $y = 0$. So by using the shoelace formula, we can express the area of the niche $\mathcal{N}(K)$ in terms of the quadratic polynomials of the coordinates of $p_{i,j}$'s, and thus the coordinates $\mathbf{p}$ of $\mathcal{K}_\Theta$.

Branch-and-bound Tree $\mathcal{T}$

The difficulty with the niche $\mathcal{N}(K)$ is that the order $\mathbf{i}_K$ of the lines appearing in the niche can change drastically depending on the cap K. We need extra restrictions on the coordinates $\mathbf{p}$ of K to ensure a niche with a fixed order $\mathbf{i}_K$ of lines. To this end, `SofaDesigner` builds a custom branch-and-bound tree $\mathcal{T}$ and stores it as a JSON-formatted file.

The tree $\mathcal{T}$ is a complete binary tree representing the subset of $\mathcal{K}_\Theta$ with $\mathcal{A} \geq 2.2195$. Each node N_i is equipped with a finite set I_i of affine-linear combinations of the coordinates $\mathbf{p}$. Each $f \in I_i$ represents the inequality $f \geq 0$, so that the node N_i represents the subset $\mathcal{K}_i \subseteq \mathcal{K}_\Theta$ where the value of f is non-negative for every $f \in I_i$. The root node N_1 of the tree $\mathcal{T}$ represents the whole $\mathcal{K}_\Theta$ with the constraints $I_i = s_0, s_1, \dots, s_{2n}$. Every non-leaf node N_i is split into two child nodes N_j and N_k by the inequalities $g_i \geq 0$ and $-g_i \geq 0$ for some affine-linear combination g_i of the coordinates $\mathbf{p}$. So, the constraints I_j of node N_j is $I_j = I_i \cup \{g_i\}$, and the constraints I_k of node N_k is $I_k = I_i \cup \{-g_i\}$. This ensures that the union of the sets $\mathcal{K}_i$ represented by the leaf nodes N_i of $\mathcal{T}$ is equal to the whole space $\mathcal{K}_\Theta$.

The first task of a verifier is to verify that the file representing $\mathcal{T}$ indeed adheres to the aforementioned property above.

TODO: under construction from here