

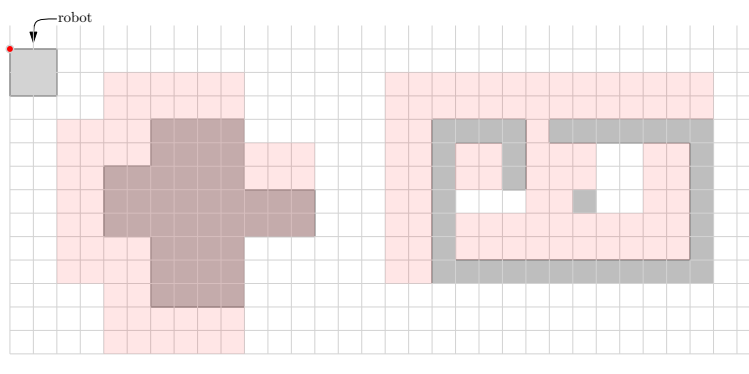
Exercice 1.

On considère un robot de forme cubique qui se déplace dans le plan par translation (il ne peut pas tourner sur lui-même).

1. Décrire l'espace des configurations de ce robot en explicitant le point chaud choisi.

Il suffit de décrire la position du robot au sol. Comme l'orientation du robot est fixée, on peut choisir comme espace de configurations le plan \mathbb{R}^2 et d'utiliser comme point chaud un coin du robot (par exemple le coin haut-gauche).

2. Tracer, l'espace libre du robot pour les obstacles suivants :



Exercice 2

Prouver que pour tous $P \subset \mathbb{R}^2$, $Q \subset \mathbb{R}^2$ et $\vec{t} \in \mathbb{R}^2$ on a

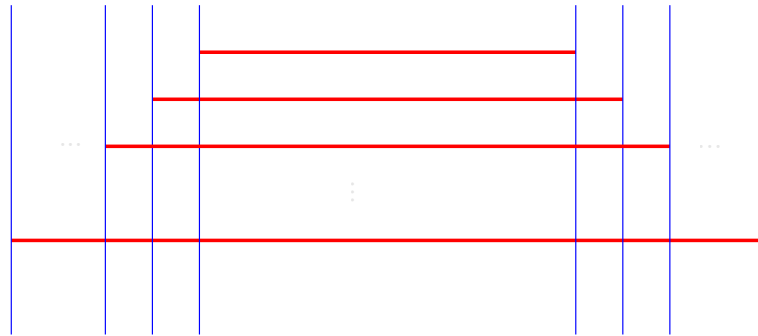
$$(P + \vec{t}) \cap Q \neq \emptyset \Leftrightarrow \vec{t} \in Q \oplus (-P)$$

Supposons qu'il existe un point $a \in (P + \vec{t}) \cap Q$. Alors $a \in Q$ et il existe un point $b \in P$ tel que $a = b + \vec{t}$. Ainsi, $\vec{t} = a + (-b) \in Q \oplus (-P)$.

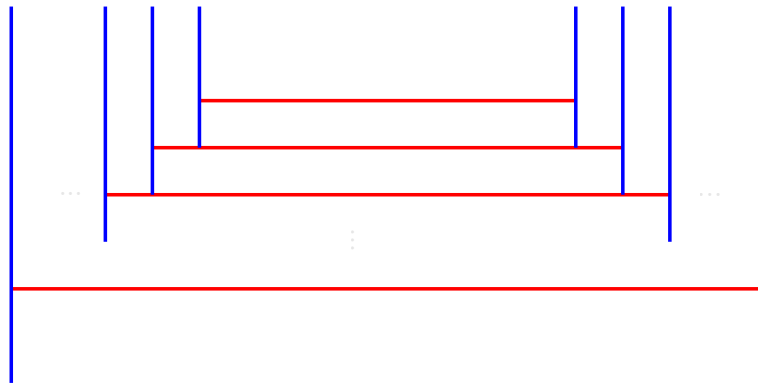
Réciproquement, supposons que $\vec{t} \in Q \oplus (-P)$. Alors il existe $a \in Q$ et $b \in P$ tels que $\vec{t} = a - b$. Par conséquent, $a = b + \vec{t}$ et $Q \cap (P + \vec{t})$ contient a , et est donc non vide.

Exercice 3

1. Proposer une famille de n segments du plan pour laquelle la complexité de la décomposition cellulaire par des droites est $\Omega(n^2)$.



2. Donner la décomposition trapezoïdale de votre exemple



Exercice 4

On souhaite décrire plus précisément l'algorithme de calcul de carte trapezoïdale par balayage.

1. Quelle est la sortie de l'algorithme ?

La sortie de l'algorithme est la liste des cellules de la décomposition cellulaire ainsi que le graphe d'adjacence de ces cellules. On peut représenter cela par une structure de donnée qui contient pour chaque cellule : (a) le numéro du segment supérieur s'il existe, (b) le numéro du segment inférieur s'il existe, (c) le x minimum de la cellule, (d) le x maximum de la cellule, (e) la ou les références des cellules voisines à gauche si $x_{\min} \neq -\infty$, (f) la ou les références des cellules voisines à droite si $x_{\min} \neq +\infty$

2. Que maintient l'algorithme au cours du balayage ?

Une liste L contenant la séquence ordonnée des bords d'obstacles coupés par la droite de balayage, et une liste C contenant les cellules coupées par la droite de balayage, et donc partiellement initialisées (il ne manque que le x -max).

3. Que fait la phase d'initialisation ?

Elle initialise :

- L à une liste vide,
- C à une liste comportant une seule cellule de $x_{\min} - \infty$, sans segment supérieur ni inférieur,
- une liste d'événements, chacun correspondant à un sommet d'un obstacle, ces événements étant triés par x croissants.

4. Quels sont les événements et comment sont-ils traités ?

Les événements sont les positions de la droite de balayage où elle passe par un sommet. Lors du traitement d'un événement, on commence par déterminer les extrémités des rayons verticaux issus de ce sommet en examinant les voisins de ce sommet dans L . Ensuite on clôt la ou les cellules dont le balayage est terminées, on crée la ou les cellules commençant à être balayées, et on renseigne les voisinages entre ces cellules.

5. Comment se décompose la complexité de cet algorithme ?

L'initialisation, dominée par le tri des événements, se fait en temps $O(n \log n)$. Ensuite, il y a $O(n)$ événements, chacun traité en temps $O(\log n)$. L'ensemble prends un temps $O(n \log n)$.