TP 1

1 Introduction

Pour le premier TP, vous devez construire un logiciel en Haskell. Ce logiciel va calculer l'aire d'une forme géométrique complexe. Cette forme complexe est décrite par une liste de forme géométrique simple. Nous allons utiliser une méthode numérique pour faire le calcul. Cela va donner une approximation de l'aire.

Un code de départ est disponible sur Moodle. Il contient une fonction nommée traitement. Vous commencez votre code dans cette fonction. Bien sûr, vous devriez écrire d'autre fonction que vous allez utiliser.

La section suivante va décrire les entrées du logiciel. Ensuite, nous allons décrire la méthode numérique utilisée. La dernière section va décrire les modalités de remise du TP.

2 Description des entrées

Le logiciel va recevoir en entrées (sur la ligne de commande), dans l'ordre, le nom du fichier contenant la description de la forme complexe et une valeur entière représentant la précision (d) du calcul. Ce nombre sera une valeur entre 2 et 10 000. Un code de départ est disponible sur Moodle. Ce code lit les arguments et le contenu du fichier. Le contenu du fichier est placé dans une chaîne de caractères. Votre code devra extraire l'information de cette chaîne.

2.1.1 Contenu du fichier : description de la forme complexe

Le fichier contient la description de la forme complexe sous forme d'une liste de formes plus simples. Chaque ligne du fichier va contenir la description d'une forme simple (s_k). Voici la description de la syntaxe qu'utilise chaque ligne du fichier.

- carre c_x c_y t
  Représente un carré où
  - c_x est la position du centre sur l'axe des x.
  - c_y est la position du centre sur l'axe des y.
    - Remarque : ces coordonnées forment un vecteur $\hat{c} = (c_x, c_y)$.
  - t est la mesure d'un côté.
- rectangle c_x c_y b h
  Représente un rectangle où
  - c_x est la position du centre sur l'axe des x.
  - c_y est la position du centre sur l'axe des y.
    - Remarque : ces coordonnées forment un vecteur $\hat{c} = (c_x, c_y)$.
  - b est la base du rectangle.
  - h est la hauteur du rectangle.
- cercle c_x c_y r

Représente un cercle où

- c_x est la position du centre sur l'axe des x.
- c_y est la position du centre sur l'axe des y.
  - Remarque : ces coordonnées forment un vecteur $\hat{c} = (c_x, c_y)$.
- r est le raton du cercle.

- ellipse c_x c_y f_x f_y g

  Représente une ellipse où
  - c_x est la position du centre sur l'axe des x.
  - c_y est la position du centre sur l'axe des y.
    - Remarque : ces coordonnées forment un vecteur $\hat{c} = (c_x, c_y)$.
  - f_x est la distance focale sur l'axe des x.
  - f_y est la distance focale sur l'axe des y.
    - Remarque : ces coordonnées forment un vecteur $\hat{f} = (f_x, f_y)$.
  - g est la mesure du grand axe.
- polygone $p_{1,x}$ $p_{1,y}$ $p_{2,x}$ $p_{2,y}$ $\cdots$ $p_{n,x}$ $p_{n,y}$

  Représente un polygone où
  - $p_{w,x}$ $p_{w,y}$ est le wième sommet (points) du polygone ($w \in [1..n]$).
    - Remarque : chaque pair de coordonnées forme un vecteur $\widehat{p_w} = (p_{w,x}\ p_{w,y})$.
  - n est le nombre de sommets (points) que contient le polygone.
  - Les sommets sont décrits en ordre antihoraire, sinon les équations ne fonctionneront pas.

Le fichier ne contiendra pas d'erreur. Votre logiciel n'a pas à faire de vérification. Toutes les valeurs seront des `Doubles`. Pour le reste de cette description, nous allons désigner le nombre de formes simple par la lettre m. Donc $k \in [1..m]$.

3 TRAITEMENT

Pour calculer l'aire de la forme complexe, vous devez premièrement construire un rectangle englobant (section 3.1). Ensuite, ce rectangle sera divisé en plusieurs petits rectangles (selon la précision demandée). Votre logiciel va trouver la position du centre de chaque petit rectangle et vérifier si ce point est à l'intérieur d'au moins une des formes simples (section 3.2). Finalement, ces résultats sont utilisés pour trouver une approximation de l'aire de la forme complexe (section 3.3). Une dernière section contient un rappel sur des opérateurs de base d'algèbre vectorielle (section 3.4).

3.1 RECTANGLES ENGLOBANTS

L'objectif est de trouver le plus petit rectangle possible qui contient toutes les formes simples. Pour cela, il faut premièrement trouver un rectangle englobant pour chaque forme simple (section 3.1.1). Ensuite, nous pourrons les utiliser pour trouver le rectangle englobant de la forme complexe (section 3.1.2).

Un rectangle englobant est décrit pas deux points : le coin inférieur gauche (u_x, u_y) et le coin supérieur droit (v_x, v_y).

3.1.1 Forme simple

Voici les formules pour trouver le rectangle englobant pour une forme simple. Pour décrire le rectangle englobant de la kième forme simple dans ce document, nous utilisons la notation $(u_x^k, u_y^k)(v_x^k, v_y^k)$.

Forme s_k	u_x^k	u_y^k	v_x^k	v_y^k
carre	$c_x - \frac{t}{2}$	$c_y - \frac{t}{2}$	$c_x + \frac{t}{2}$	$c_y + \frac{t}{2}$
rectangle	$c_x - \frac{b}{2}$	$c_y - \frac{h}{2}$	$c_x + \frac{b}{2}$	$c_y + \frac{h}{2}$
cercle	$c_x - r$	$c_y - r$	$c_x + r$	$c_y + r$
ellipse	$c_x - \frac{g}{2}$	$c_y - \frac{g}{2}$	$c_x + \frac{g}{2}$	$c_y + \frac{g}{2}$
polygone	$\min_{w \in [1..n]} p_{w,x}$	$\min_{w \in [1..n]} p_{w,y}$	$\max_{w \in [1..n]} p_{w,x}$	$\max_{w \in [1..n]} p_{w,y}$

3.1.2 Forme complexe

Pour calculer le rectangle englobant de la forme complexe $(u_x^C, u_y^C)(v_x^C, v_y^C)$, il suffit de prendre les extremums des rectangles englobants des formes simples.

$$u_x^C = \min_{k \in [1..m]} u_x^k$$

$$u_y^C = \min_{k \in [1..m]} u_y^k$$

$$v_x^C = \max_{k \in [1..m]} v_x^k$$

$$v_y^C = \max_{k \in [1..m]} v_y^k$$

3.2 MÉTHODE NUMÉRIQUE

Maintenant que vous avez le rectangle englobant, vous pour diviser ce rectangle en petit rectangle ($a_{i,j}$) et trouver le centre de chaque petit rectangle. La taille du petit rectangle est calculée comme suit.

$$a_{base} = \frac{(v_x^C - u_x^C)}{d}$$

$$a_{hauteur} = \frac{(v_y^C - u_y^C)}{d}$$

Les coordonnées du centre d'un petit rectangle seront $\widehat{a_{i,j}} = (x_i, y_j)$

$$x_i = u_x^C + a_{base}(i + 0.5)$$

$$y_j = u_y^C + a_{hauteur}(j + 0.5)$$

Nous voulons vérifier si le centre d'un petit rectangle est à l'intérieur d'une forme simple. Voici le prédicat (`estIntérieur(`$\widehat{a_{i,j}}, s_k$`)`) déterminant si un point $\widehat{a_{i,j}}$ est à l'intérieur d'une forme s_k. Certaines de ces équations utilisent l'algèbre vectorielle, voir plus loin pour un rappel sur ces opérateurs.

Forme s_k	`estIntérieur(`$\widehat{a_{i,j}}, s_k$`)`
carre	$\left(c_x - \frac{t}{2} \leq x_i \leq c_x + \frac{t}{2}\right) \wedge \left(c_y - \frac{t}{2} \leq y_i \leq c_y + \frac{t}{2}\right)$
rectangle	$\left(c_x - \frac{b}{2} \leq x_i \leq c_x + \frac{b}{2}\right) \wedge \left(c_y - \frac{h}{2} \leq y_i \leq c_y + \frac{h}{2}\right)$
cercle	$\left\|\hat{c} - \widehat{a_{i,j}}\right\| \leq r$
ellipse	$\left\|(\hat{c} + \hat{f}) - \widehat{a_{i,j}}\right\| + \left\|(\hat{c} - \hat{f}) - \widehat{a_{i,j}}\right\| \leq g$
Polygone(*)	$\forall w \in [1..n], \left(\widehat{a_{i,j}} - \widehat{p_w}\right) \cdot \left({}^{\perp}(\widehat{p_w} - \widehat{p_z})\right) \leq 0$

(*) : si $w = n$, alors $z = 1$, sinon $z = i + 1$.

En utilisant ce prédicat, pour chaque petit rectangle, vous devez vérifier s'il existe une forme simple pour laquelle le centre du petit rectangle est à l'intérieur.

`vérifier(`$\widehat{a_{i,j}}$`)` $= \exists k \in [1..m]$, `estIntérieur(`$\widehat{a_{i,j}}, s_k$`)`

3.3 Calcul de l'aire

Pour calculer l'approximation de l'aire de la forme simple, vous devez compter combien de petits rectangles donne `Vrai` pour le prédicat `vérifier`. Désignons le nombre de `Vrai` par la lettre q.

`Aire`$= a_{base} \times a_{hauteur} \times q$

Votre fonction va simplement retourner cette valeur.

3.4 Rappel d'algèbre vectorielle

Soit les vecteurs $\hat{u} = (u_x, u_y)$ et $\hat{v} = (v_x, v_y)$.

- Addition : $\hat{u} + \hat{v} = (u_x + v_x, u_y + v_y)$
- Soustraction : $\hat{u} - \hat{v} = (u_x - v_x, u_y - v_y)$
- Produit scalaire : $\hat{u} \cdot \hat{v} = u_x \times v_x + u_y \times v_y$
- Norme : $|\hat{u}| = \sqrt{u_x^2 + u_y^2}$
- Vecteur perpendiculaire : ${}^{\perp}\hat{u} = (u_y, -u_x)$

4 Modalités

4.1 Directive

1. Le TP est à faire seul ou en équipe de deux (maximum).
2. Écrivez les noms des auteurs au début du fichier (en commentaire).
3. Tout le code doit être dans un seul fichier.
4. Vous **ne pouvez pas** inclure (importer) d'autre librairie.
5. Commentaire :