

Analyse statique en 2D

Contact: [Pierre-Evariste Dagand](#)

Introduction générale

Dans leur livre [Introduction to Static Analysis, An Abstract Interpretation Perspective](#), Xavier Rival et Kwangkeun Yi introduisent un petit langage de programmation visant à décrire des déplacements sur le plan. Inspiré de [Logo](#), ce langage offre les primitives suivantes :

```
p ::= init(R)
      (place le curseur aléatoirement dans la région R)
  | translation(u, v)
      (translation par le vecteur (u, v))
  | rotation(x, y, theta)
      (rotation centrée en (x, y) et d'angle theta)
  | p ; p
      (séquence d'opérations)
  | { p } or { p }
      (choix aléatoire)
  | iter { p }
      (boucle infinie)
```

où $x, y, u, v \in \mathbb{R}$ décrivent des coordonnées sur le repère orthonormé et $R \in (\mathbb{R}^2)^*$ décrit une région du plan par l'intermédiaire des coordonnées (dans le sens horaire) d'un polygone.

Par exemple, le programme suivant

```
init((0,0); (0, 1); (1, 1); (1, 0));
iter {
  {
    translation(1, 0)
  } or {
    rotation(0, 0, 90)
  }
}
```

initialise un pointeur dans la région $x \in [0, 1], y \in [0, 1]$ puis effectue répétitivement soit une translation sur l'axe des abscisses, soit une rotation d'angle droit centrée à l'origine du plan.

Ce projet long s'intéresse à la question suivante : **étant donné un programme p et une région R_{danger} , peut-on garantir que aucune des exécutions possibles de p ne traverse la région R_{danger} ?**

Cette problématique s'inscrit dans le cadre de l'*analyse statique*, qui est le sujet du module Analyse Statique (IFGCY130) en M2. Ce projet long propose donc

une introduction au domaine.

Objectifs

L'objectif du projet est d'introduire le cadre conceptuel de l'analyse statique par interprétation abstraite pour un langage graphique. Il s'agira de

- s'approprier les concepts mathématiques du domaine,
- implémenter ces concepts, et
- supporter la visualisation des exécutions et des analyses statiques

Ce troisième point est particulièrement important et ne saurait être négligé : une attention particulière devra être portée à la correction et à la qualité des visualisations. Ce projet contient donc une part non négligeable de traitements graphiques et géométriques que l'on retrouve traditionnellement dans les jeux 2D.

Calendrier

Ce projet suivra le découpage suivant :

- choix d'un langage de programmation à partir d'exemples de visualisations
- modélisation de la syntaxe des commandes graphiques dans le langage de programmation choisi
- visualisation d'une exécution aléatoire d'un programme
- visualisation des régions interdites $R_{interdit}$
- détection, à l'exécution, d'une violation de région interdite
- implémentation de l'abstraction de signe pour le fragment **sans itération**
- implémentation de l'abstraction d'intervalle pour le fragment **sans itération**
- traitement naïf de l'itération, par itération instable
- traitement correct de l'itération, par élargissement
- implémentation de l'abstraction polyédrale convexe

On s'attellera à réaliser une tâche par semaine, sauf pour les deux dernières tâches qui pourront être traitées plus ou moins complètement selon le temps disponible en fin de projet.

Références

- [Introduction to Static Analysis, An Abstract Interpretation Perspective \(Rival, Yi\)](#) (Chapitre 2 **uniquement**)
- [Abstraction \(Cousot\)](#)
- [HTML Canvas \(W3C\)](#)
- [vg \(OCaml\)](#)