

# Une petite introduction à Plade

Michel Chilowicz

Email: [michel.chilowicz@univ-paris-est.fr](mailto:michel.chilowicz@univ-paris-est.fr)

Site web: <http://igm.univ-mlv.fr/~chilowi/>

Laboratoire d'Informatique Gaspard-Monge  
Université Paris-Est

Journée Logiciels du Laboratoire d'Informatique  
Gaspard-Monge

# La recherche de similarité dans du code source

## Applications

- ▶ Maintenance de projets logiciels (élimination de clones)
- ▶ Analyse de versions  
(reconstitution d'opérations de refactoring)
- ▶ Mise en évidence de copies illégitimes entre différents projets  
(avec potentielle obfuscation)

## Problématiques

- ▶ Repérer des zones de code source similaire
- ▶ Les représenter avec caractérisation  
(pertinence, inter-relations...)

## Plade en quelques mots...

- ▶ Objectif originel : PLAGiarism DEtector
- ▶ Outil expérimental d'évaluation d'approches de recherche de similarité
- ▶ Implanté en Java 1.5 (+ scripts en Python 2)

# Recherche de motifs similaires sur représentations

Composantes associables pour rechercher et caractériser des similarités

## Représentations

- ▶ Transformation du code en représentations exploitables (chaînes de lexèmes, AST...)
- ▶ Référentiels de stockage des représentations
- ▶ Support de transformations et abstractions multiples

## Recherche de motifs

- ▶ Algorithmes sur chaînes et arbres
- ▶ Association des approches pour filtrage rapide, recherche approfondie, consolidation

# Principales approches algorithmiques implantées

## Par indexation de suffixes (fact)

Factorisation du graphe d'appel de fonctions :  
fonctions synthétiques modélisant les similarités

## Par indexation en base d'empreintes (fingertree)

Recherche de germes et consolidation guidée par l'AST

—→ obtention de graphes de correspondances avec relations  
d'inclusion et de chevauchement

## Composantes principales

- ▶ *core* : référentiels, structures de base
- ▶ *parsers* : analyseurs lexicaux et syntaxiques (actuellement C(++) et Java)
- ▶ *fact* : factorisation de graphe d'appel (similarité fonctionnelle)
- ▶  *fingertree* : manipulation de sous-arbres indexés par hachage et consolidation de correspondances
- ▶ *gui* : interface graphique Swing d'exploration de correspondances

## Composantes connexes

- ▶ *algostruct* : sérialisation, classes utilitaires générales, manipulation de graphes
- ▶ *factools* : bibliothèque de construction d'objets
- ▶ *beech* : arbre  $k$ -aire équilibré pour indexation sur stockage de masse
- ▶ *fixarr* : algorithmes de manipulation de tables de suffixes, construction de graphes de farmax

## Perspectives

- ▶ Très prochainement : version publique Affero GPL v3 de Plade et de ses composantes connexes  
Affero GPL  $\sim$  GPL avec copyleft valable également pour une distribution en service réseau
- ▶ Mise en ligne d'un service web (interfaçable avec Subjects)
- ▶ Amélioration des approches existantes et proposition de nouvelles méthodes