Algorithmes pour la détection de similarités de code

Fabrice Huet
fabrice.huet@unice.fr

17 février 2017

Nombre d'étudiants souhaités : 1-3

Description du sujet

Détecter le plagiat est un un problème qui peut être compliqué tant la notion de plagiat est large. Dans une première approche, on peut considérer du copier-coller de texte et donc il s'agit simplement de rechercher dans un texte des sous-séquences d'un autre texte. Mais dans le cas où il y a ré-écriture pour ne garder que les idées ou l'enchainement logique, cette stratégie ne fonctionne pas. Afin de cadrer la problématique, nous allons considérer le problème du plagiat de code source en Java ou C. Dans ce contexte, le plagiat consiste en un copier-coller de morceaux de codes suivi, en général, de modifications rapides et semi-automatiques comme le renommage de variables afin de masquer la fraude. Le problème n'est donc plus de trouver des séquences strictement identiques mais similaires suivant une métrique donnée.

Il existe des logiciels propriétaires qui permettent une analyse de code mais ils sont souvent payants et ont rarement été évalués. D'un point de vue recherche, il y a eu de nombreux travaux mais encore une fois, peu d'évaluations croisées (`https://scholar.google.fr/scholar?q=code+similarity`).

Le but de ce TER est, dans un premier temps, de faire un état de l'art sur les différentes techniques existantes pour rechercher les similarités dans un ensemble de codes source. Dans un deuxième temps, plusieurs techniques seront implémentées et testées sur des projets écrits en C et Java afin de déterminer leur efficacité.

Mots clés : Arbre syntaxique abstrait, distance, Java, C

Lieu

Sessions de travail régulières au laboratoire I3S (Sophia Antipolis)

Prérequis

Connaissance de Java/C, notions de compilation et analyse de code.