
HyperAST: Analyser efficacement de grands historiques de code

Quentin Le Dilavrec*^{†1}, Djamel Eddine Khelladi*^{‡1}, Arnaud Blouin*^{§1}, and Jean-Marc Jézéquel*^{¶1}

¹Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) – Université de Rennes, Institut National des Sciences Appliquées - Rennes, Université de Bretagne Sud, École normale supérieure - Rennes, Institut National de Recherche en Informatique et en Automatique, CentraleSupélec, Centre National de la Recherche Scientifique, IMT Atlantique – Avenue du général Leclerc Campus de Beaulieu 35042 RENNES CEDEX, France

Résumé

Cette démonstration présente l’HyperAST (1), une approche d’analyse d’historiques de code performante, se fondant sur la redondance du code à travers le temps et l’espace, ainsi que sur les possibilités d’analyses partielles de code. Actuellement, l’analyse des historiques de code se fait en mode ”batch”: chaque version (ou commit) est traitée (en grande partie) indépendamment les unes des autres pour calculer un ensemble de métriques ; à la fin de l’analyse ces métriques sont utilisées pour observer l’évolution de la base de code au cours du temps. Notre approche propose de traiter plus finement l’historique de code, au niveau de l’AST, et de partager les éléments identiques dans et entre chaque version. Cette démonstration vise à expliquer ce principe au travers de trois scénarios. Groupes de travail : VL et CLAP

Mots-Clés: Mining Software Repositories, Temporal Code Analysis, Code Quality

*Intervenant

[†]Auteur correspondant: quentin.le-dilavrec@irisa.fr

[‡]Auteur correspondant: djamel-eddine.khelladi@irisa.fr

[§]Auteur correspondant: arnaud.blouin@irisa.fr

[¶]Auteur correspondant: jezequel@irisa.fr