

**CHENNAI – PONDICHERRY**

**Identifying and Summarizing Systematic Code Changes via Rule Inference**

**Abstract**

Programmers often need to reason about how a program evolved between two or more program versions. Reasoning about program changes is challenging as there is a significant gap between how programmers think about changes and how existing program differencing tools represent such changes. For example, even though modification of a locking protocol is conceptually simple and systematic at a code level, diff extracts scattered text additions and deletions per file. To enable programmers to reason about program differences at a high level, this paper proposes a rule-based program differencing approach that automatically discovers and represents systematic changes as logic rules. To demonstrate the viability of this approach, we instantiated this approach at two different abstraction levels in Java: first at the level of application programming interface (API) names and signatures, and second at the level of code elements (e.g., types, methods, and fields) and structural dependences (e.g., method-calls, field-accesses, and subtyping relationships). The benefit of this approach is demonstrated through its application to several open source projects as well as a focus group study with professional software engineers from a large e-commerce company.