Master Thesis Software Engineering
Diplom-/Masterarbeit ETH Zürich

Student
Fabian Yves Bannwart **Stud-ID** **01-908-755**
E-mail fabianba@student.ethz.ch

Supervisor
Peter Müller

Retaining Semantic Equivalence During Refactoring

The benefit of refactoring in software engineering for software maintenance is beyond any reasonable doubt. Refactoring is applying transformations that are aiming at making code more readable and easier to change while retaining equivalent functionality and external behavior. Refactoring as a discipline is often embedded in the software development cycle and extensive testing ought to guarantee correctness of the applied transformations. The fact that refactorings do actually retain the equivalence they advertise if applied correctly has not been established to date. It is what this Master Thesis is trying to achieve.

In existing textbooks, refactorings equivalence criteria are given informally. The thesis ought to formalize these criteria for a set of representative refactorings and find whether they are necessary and sufficient for correctness. The criteria shall be compared to semantic conditions based on an operational semantics for an object kernel language that abstracts the features of Java. As far as traditional, previously only informally stated conditions prove insufficient for correctness or overly restrictive, new syntactic conditions shall be established that are provably correct.
One of the key problems to be solved by this thesis is to define an appropriate notion of semantic equivalence.

A further topic of this thesis will be refactorings that are not intended to retain strict equivalence but do only change the semantics of programs that are outside the realm of their intended functionality or usage. After they have been identified, the thesis shall focus on how to determine intended functionality and how to make sure even these refactorings are applied safely.