Verifying UML Models Annotated with OCL Strings

Ankit Jha

Department of Computer Science Maynooth University, Ireland Supervisor: Hao Wu & Rosemary Monahan Ankit.Jha.2021@mumail.ie

Abstract

The Unified Modeling Language (UML) is widely used in building different types of software models. It is the central element for Model Driven Engineering (MDE) and software development methodology. The graphical notations of UML not only allow software engineers to model different aspects of software but also enhances the collaboration among team members. Meanwhile, the Object Constraint Language (OCL) is a formal language that helps to set specific rules or constraints in a precise way, which is text representation of formal rules. When enriched with OCL, a powerful constraint language, UML models attain a heightened level of precision and expressiveness, facilitating detailed specification of system properties and behavior. Using OCL string expressions, we can specify constraints which is associated with string which are widely encountered in our daily life situations such as username or password, which should follow a specific pattern, etc. OCL string is extremely helpful to specify these type of constraints which include strings. Some efforts were made by using an automated approach grounded in Constraint Programming to verify UML models annotate with OCL constraints including string. Despite of advancements in verification techniques, still there are challenges, such as handling complex string-based constraints, handling Quantifiers and the need for efficient tool support. However, the rise of SMT solvers has introduced a new era in formal verification, offering automated tools capable of reasoning about convoluted models. These solvers are designed to determine the satisfiability of logical formulas. Their ability to manage a diverse range of data types and complex constraints makes them particularly suitable for this task. Nevertheless, the performance and efficiency of SMT solvers exhibit variance depending on the characteristics of the problem at hand. Therefore, it becomes necessary to compare SMT solvers experimentally based on different parameters such as time taken to get the result. We have tried to find best SMT solver in order to verify UML models that have been annotated with OCL strings.