Abstract

Assume-guarantee verification method has been recognized as a promising approach to verify component-based software (CBS) with model checking. The method is not only fitted to component-based software but also has a potential to solve the *state space explosion problem* in model checking. This method allows us to decompose a verification target into components so that we can model check each of them separately. Model-based verification methods in general and the assume-guarantee verification method in particular of a system are performed with respect to its model which exactly describes the behavior of the system. Thus, we have to obtain the accurate model of the system before applying the verification techniques. However, these methods generally assume that the ways to obtain the model and its correctness are available. This means that the model-based verification methods assume the availability and correctness of the model which describes the behavior of the system under study. Nonetheless, this assumption may not always hold in practice due to the modelling errors, bug fixing, etc. In addition, evolving of the existing components of CBS is a daily and unavoidable activity during the software life cycle. Therefore, even if the assumptions hold, the model could be invalidated when the software is evolved by adding or removing some behaviors. Unfortunately, the consequence of these tasks is the whole evolved software must be rechecked. The purpose of this research is to provide an effective approach for modular verification of evolving component-based software systematically in the context of the component evolution. When a component is evolved after adapting some refinements, the proposed framework focuses on this component and its model in order to update the model and to recheck the whole evolved system. The framework also reuses the previous verification results and the previous models of the evolved components to reduce the number of steps required in the model update and modular verification processes.

This dissertation has three main contributions. The first contribution of the research is to propose a method for generating minimal assumptions for the assume-guarantee verification of component-based software. The proposed method is an improvement of the L*-based assumption generation method. The key idea of this method is finding the minimal assumptions in the search spaces of the candidate assumptions. These assumptions are seen as the environments needed for the components to satisfy a property and for the rest of the system to be satisfied. The minimal assumptions generated by the proposed method can be used to recheck the whole system at much lower computational cost.

The second contribution is to propose an effective framework for assume-guarantee verification of component-based software in the context of the component evolution at design level. In this framework, if the model of a component is evolved, the whole component-based software of many models of the existing components and the evolved model of the evolved component is not required to be rechecked. The framework only checks whether the evolved model satisfies the assumption of the system before evolving. If it does, the evolved component-based software still satisfies the property. Otherwise, if the assumption is too strong to be satisfied by the evolved model, a new assumption is regenerated. We propose two methods for new assumption regeneration: assumption regeneration and minimized assumption regeneration. The methods reuses the current assumption as the previous verification result to regenerate the new assumption at much lower computational cost.

The third contribution of the research is to propose a framework for modular conformance testing and assume-guarantee verification of component-based software in the context of component evolution at source code level. This framework includes two stages: modular conformance testing for updating inaccurate models of the evolved components and assume-guarantee verification for evolving component-based software. When a component is evolved, the proposed framework focuses on this component and its model in order to update the model and to recheck the whole evolved system. The framework also reuses the previous verification results and the previous models of the evolved components to reduce the number of steps required in the model update and assume-guarantee verification processes.

Key Words: **verification, model checking, assume-guarantee reasoning, assume-guarantee verification, modular verification, component evolution, conformance testing, learning algorithm, assumption, component-based software.**