Implementing an Object-Oriented theory in HOL

Kenro Yatake
School of Infomation Science,
Japan Advanced Institute of Science and Technology

January 10, 2006

Abstract

The Object-Oriented developing method has become the mainstream of the software development. In the upstream phase of the development, analysis models are constructed with a modeling language such as UML. The problem is that these models tend to include inconsistencies since they are based on informal diagrams. In order to discover inconsistencies in the early stage of development, it is important to define formal semantics for the models and conduct verification on them. Our verification targent is the invariant properties about object attributes, or properties about datatypes. To verify these properties, it is relevant to applying theorem provers.

As a semantics of analysis model verification, we implemented an Object-Oriented theory in the HOL system. In general, analysis models are abstracted using high abstract types and domain-specific types. In order to cope with this characteristic of analysis models, we enabled objects to have attributes of arbitrary types. The embedding problem is that it is difficult to express an object, which has an arbitrary number of arbitrary types of attributes, as a general type in the simple first-order type system of HOL. To cope with this problem, we take the approach of automatically constructing the theory depending on the type information of each system.

The soundness of the theory is guaranteed by *definitional extension*. This is a standard method to construct sound theories in HOL, where new theories are derived from existing sound theories by only allowing introduction of definition and derivation by sound inference rules. Specifically, the theory is derived from the operational semantics of a heap memory model which is defined by primitive theories such as lists and pairs.

As for the invariant verification, we propose a collaboration-based method. In this method, collaborations are defined as functions on the system state using primitive operators given in OO theory and invariants are proved by induction on the state transition sequence from the initial state of the system. Compared to statechart-based verification method, this method has an advantage on verifying invariants involving multiple objects.

Keywords : Object-Oriented, theorem proving, HOL, invariant, collaboration