Müller, Peter

Modular specification and verification of object-oriented programs. (English) Zbl 0974.68035
Hagen: FernUniv. Hagen, Fachbereich Informatik, vi, 261 p. (2001).

Summary: The paradigm shift from procedural to object-oriented programming promoted modular software development. Especially reuse of prefabricated software modules increases the demand for precise specifications and quality certification, and thus for modular specification and verification techniques. Such techniques must be capable of handling object-oriented language features such as subtyping, inheritance, and dynamic method binding, and have to support modular development of specifications and proofs. In particular, they should enable specifications and proofs to be reused along with implementations.

This thesis presents modular specification and verification techniques for the functional behavior, frame properties, and type invariants of 00-programs. The key idea underlying this work is the formal integration of state-of-the-art specification and verification techniques with a type system for alias control.

We present the universe type system that can be used to control aliasing statically. It combines strong type constraints for read-write references with the flexibility of read-only references. This combination guarantees an invariant that enables modular verification while retaining enough flexibility to handle most common implementation patterns, especially patterns such as binary methods and iterators that are not supported by related approaches.

The declarative interface specification technique presented in this thesis provides pre-post-specifications, abstract fields with explicit dependencies, modifies-clauses, and type invariants. Functional method behavior can be covered by pre-post-specifications. Abstract fields are used to map object structures to values of an abstract domain. The dependencies of an abstract field on the concrete fields that represent it are explicitly declared. Together with modifies-clauses, these declarations are used to express frame properties. Frame properties are particularly difficult to verify in a modular way since they require one to prove that certain abstractions are not modified by a method even if these abstractions are declared in other modules. To cope with this problem, we exploit the invariant guaranteed by the universe type system to define a novel semantics for modifies-clauses and to restrict the permissible dependencies of abstract fields in a way that makes modular verification of frame properties possible. Regarding type invariants as special abstract fields allows us to apply the specification and verification technique for frame properties to invariants.

For verification, we use a Hoare-style programming logic that is capable of handling OO-features and modularity. In particular, it ensures that only those properties of a module can be proved that hold in all well-formed contexts in which the module might be reused. That is, the logic guarantees modular soundness of our verification technique.

Our techniques are presented for a programming language similar to sequential Java, but can be adapted to procedural and other object-oriented languages as well.

MSC:

68N19 Other programming paradigms (object-oriented, sequential, concurrent, automatic, etc.)

Keywords:

object-oriented programming; modular software