Abstract

In the thesis we study *architectural specifications* — a formalism in which the modular structure of the program development process may be expressed. The study is carried out in the context of the Common Algebraic Specification Language CASL.

We first define and analyze the notion of architectural specification. In particular, we consider a number of variants of the semantics of architectural specifications. We investigate the relationship between formalizing program development indirectly on models (algebras) and directly on programs. We introduce a notion of fully static correctness for specification languages with the amalgamation property; fully static correctness guarantees that a correct development process on models leads to a correct development process on programs.

The second part of the thesis is devoted to the *verification* of architectural specifications, that is, to proving that the program development process described by them is correct. We devise a sound and (relatively) complete proof calculus for verifying fully statically correct architectural specifications. Since CASL does not have amalgamation, we present two sound and complete verification methods for languages without amalgamation. Both rely on a representation of the concerned language in a language with amalgamation and further technical properties. In the first method, fully static correctness is replaced by so-called extended static correctness. When applying this to CASL, we use a representation in ENRICHEDCASL and prove that ENRICHED-CASL has the required properties; in the course of the proof we generalize results from semigroup amalgamation theory to operads and multicategories. In the second method no additional static assumptions are needed; when applying this to CASL, we use a representation in the multisorted version of CASL.

Keywords: Architectural specifications, algebraic specification, institutions, CASL.

ACM Classification: D.2.4, D.2.11, F.3.1