Tzoref-Brill, Rachel; Maoz, Shahar

Lattice-based semantics for combinatorial model evolution. (English) Zbl 06527558

Finkbeiner, Bernd (ed.) et al., Automated technology for verification and analysis. 13th international symposium, ATVA 2015, Shanghai, China, October 12–15, 2015. Proceedings. Cham: Springer. Lect. Notes Comput. Sci. 9364, 276-292 (2015)

Summary: Combinatorial test design (CTD) is an effective test design technique, considered to be a testing best practice. CTD provides automatic test plan generation, but it requires a manual definition of the test space in the form of a combinatorial model. As the system under test evolves, e.g., due to iterative development processes and bug fixing, so does the test space, and thus, in the context of CTD, evolution translates into frequent manual model definition updates.

In this work, we show that the Boolean semantics currently in use by CTD tools to interpret the model is inadequate for combinatorial model evolution, and propose to replace it with a new lattice-based semantics that (1) provides a consistent interpretation of atomic changes to the model via Galois connections, and (2) exposes which additional parts of the model must change following an atomic change, in order to restore validity. We further use the new lattice-based semantics to define new higher-level atomic operations for combinatorial model evolution. Finally, we identify recurring abstraction and refinement patterns in the evolution of 42 real-world industrial models, and use the new lattice-based semantics to define new higher-level atomic constructs that encapsulate these patterns.

The proposed lattice-based semantics and related new modeling constructs advance the state-of-the-art in CTD with a new foundation for model evolution and with better tools for change comprehension and management.

For the entire collection see [Zbl 1325.68017].

MSC:

68Q55 Semantics in the theory of computing
06A15 Galois correspondences, closure operators (in relation to ordered sets)
06B23 Complete lattices, completions
68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)

Full Text: DOI