Dániel Darvas

Practice-Oriented Formal Methods to Support the Software Development of Industrial Control Systems

Summary

Formal specification and verification methods provide ways to describe requirements precisely and to check whether the requirements are satisfied by the design or the implementation. In other words, they can prevent development faults and therefore improve the quality of the developed systems. These methods are part of the state-of-the-practice in application domains with high criticality, such as avionics, railway or nuclear industry.

The situation is different in the industrial control systems domain. As the criticality of the systems is much lower, formal methods are rarely used. The two main obstacles to using formal methods in systems with low- or medium-criticality are *performance* and *usability*. Overcoming these obstacles often needs deep knowledge and high effort. Model checking, one of the main formal verification techniques, is computationally difficult, therefore the analysis of non-trivial systems requires special considerations. Furthermore, the mainly academic tools implementing different model checking algorithms are not suitable for users who are not experts in formal methods. The situation is similar with formal specification methods: they are typically too abstract or theoretical to be used by non-specialists, with reasonably long training period.

This work provides various solutions to both the challenges of performance and usability, and centred around the formal verification of industrial control systems. The aim is to provide more efficient verification algorithms and easy-to-use, practice-oriented formal (verification and specification) methods that can be applied to PLC (programmable logic controller) software used in industrial control systems, where the use of heavyweight, low-level methods is not necessary or not feasible. The proposed methods take the particularities of the target domain into account, making formal methods accessible without excessive effort needed.

First, this dissertation provides B-I-Sat, a new algorithm that improves the performance of the saturation-based model checking techniques by combining it with bounded model checking techniques. Saturation-based model checking already provides good performance for many different models. By combining it with bounded techniques, this performance can be further improved in certain cases.

Second, a verification workflow and its implementation are presented that allow the industrial practitioners to use the model checking for PLC-based control software. This is achieved by hiding all formal details and adapting the inputs and outputs of the verification workflow to the specific needs of the domain and the available knowledge. The contributions include a model checker-independent representation of the programs to be verified and property-preserving reduction algorithms to make the formal analysis feasible. Special attention is paid to the verification of safety-critical PLC programs, where development restrictions impose additional needs for the verification workflow.

Third, a formal specification language is proposed that is specifically targeting the behaviour description of program modules used in the PLC-based industrial control software. The language itself is heavily adapted to the domain and its needs. Furthermore, it is complemented by static analysis, code generation and conformance checking methods. For the conformance checking, special relations were introduced, responding to the real needs observed in the domain.

All these contributions are demonstrated and evaluated on real, industrial examples.