Abstract

PROGRAM VERIFICATION FOR MICROCONTROLLER-SPECIFIC CODING RULES

Degree: Doctor of Science (Information Science)

Student Name: Nguyen Thi Thuy

Student Id: S2020013

Microcontrollers are small computers specially designed for embedded systems. Since they were introduced, microcontrollers have been used in a wide range of applications, including mission-critical applications such as satellite rockets, automotive systems, and medical devices. Hence, it is essential to ensure the reliability of microcontroller-based applications. When developing these applications, hardware-dependent features must be considered in addition to standard C language features. For hardware-dependent features, a hardware manual is provided for each microcontroller. In this manual, information that requires special attention as coding rules are emphasized, along with the explanation of the features of the hardware. Currently, the process of verifying these coding rules is performed manually, as no single tool can directly handle this task. The reason is that the coding rules are non-standard while existing verification tools often support standard coding rules only. Verifying the coding rules is time-consuming and laborious as both the volume of the embedded source code (e.g., thousands of lines of code) and the number of coding rules are often large (e.g., the hardware manual of a popular microcontroller contains 2415 pages and 492 fragments that describe notes or cautions).

Several researchers tried to handle this task. In general, they tended to create new tools or extend existing tools. Implementing new tools or extensions is a heavy and inflexible solution. Additionally, new hardware models are frequently introduced. It is impractical to introduce one verification tool for each model.

This research aims to automate these coding rules' verification processes by proposing a flexible approach. Specifically, we proposed a verification framework that utilizes advanced techniques in program analysis and model-driven engineering. Firstly, the program analysis techniques (i.e., pattern matching, abstract interpretation-based static program analysis, bounded model checking, and counterexample-guided abstract refinement) are combined to analyze C programs effectively. Secondly, heuristic-based natural language processing techniques are employed to analyze the hardware manual and extract hardware knowledge. Thirdly, model-driven engineering techniques are employed for comprehensively modeling

the hardware, compiler, and source code of microcontroller-based systems. Finally, model querying techniques enable flexibly verifying the system against the target coding rules.

The approach was evaluated by applying to handle a benchmark source code and an industrial source code. The experiment with the benchmark source code showed that the approach is feasible in verifying microcontroller-based systems against the register-access coding rules. Although the benchmark was a small-size source code only, the source code represented different ways for violations to occur. The approach analyzed the source code successfully and detected all expected violations of the target register-access coding rules. We even find violations that senior developers miss. The experiment with the industrial source code showed that our verification framework was applicable to a real product. The precision in this experiment was 0.8, as two false warnings were detected. The recall was 1, as all expected violations were found.

This research contributes a practical solution for an important problem in the industry. Formal verification theory is highly established through a long history of development. Many methods proposed in academics could work well with small benchmark source code, but not many are applicable to industrial applications. The industrial setting is much more complex in comparison with laboratory environments. Among needs in the industry, verifying systems against microcontroller-specific coding rules is a heavy yet critical task. There is an emergency call for automated solutions. In response, our work is expected to be a practical and effective solution to judge the conformance of the coding rules and a tool for other tasks like source code understanding. The proposed verification framework promises to reduce the huge number of manual tasks in the current verification process in practice.

Keywords: Microcontroller-specific coding rules, C programs, Program analysis, Knowledge modeling, Program verification