MSc Thesis Proposal

José Bacelar Almeida and Jorge Sousa Pinto
CCTC-UM

September 2009

1 Title

Implementation of a Verification Conditions Generator for a subset of the C language

2 Context

This work is proposed in the context of the research project *RESCUE, REliable and Safe Code execUtion for Embedded systems* funded by the Portuguese Foundation for the Science and Technology, FCT (*Fundação para a Ciência e a Tecnologia*). The DIUM/CCTC team in this project consists of José Bacelar Almeida, Manuel Barbosa, Maria João Frade, and Jorge Sousa Pinto.

The project kicked off in January 2008. In addition to team support, it will provide travel money, as well as fund other human resources (namely research assistant grants) to work in tasks closely associated with this PhD project.

The RESCUE project aims at providing innovative, efficient and expressive mechanisms for the secure implementation and execution of code, with an emphasis on problems posed by embedded systems. Innovative mechanisms are required to develop techniques that will allow embedded applications to be statically checked against safety policies, to self-adapt considering the availability of resources, and to perform software upgrades without human intervention.

3 Program Verification and VCGens

Program verification is the activity that certifies that a given program conforms to its specification, be it in terms of an input-output relation (e.g. the program implements a given sorting algorithm, or outputs the Fibonacci number of the input), or in terms of same safety properties (e.g. the program is free of segmentation faults) or security properties (e.g. there is no interference between high-security and low-security data).

The modern approach to program verification involves two components:

1. An automatic or interactive theorem prover;
2. An application called *Verification Conditions Generator* (VCGen)

The idea is that the VCGen runs on a given input program annotated with a specification and invariants, and generates a set of first-order proof obligations. The VCgen is *sound* in the sense that, if all the proof obligations are valid (which is proved by using the theorem prover) then the program is correct with respect to its specification.

Program verification systems based on VCGens include Caduceus (for the C language), Krakatoa, KeY, and ESC/Java (also for Java), and Boogie (for Spec#).

4 Goals

Unfortunately, all the existing VCGens are too complex to be used in a teaching and research environment, and their specifications are either not available or difficult to understand. The goal of the current project is to implement a VCGen for a subset of the C language.

The candidate will implement a VCGen step-by-step, starting from a toy language and progressively adding features to it and to the VCGen. One of the most interesting aspects of the VCGen will be the generation of safety conditions. The aim is to have a modular, well-designed VCGen that can be easily modified, adapted, and extended for various purposes in the context of graduate teaching or research tasks.

The VCGen will follow the design developed in a recent DI/CCTC technical report by Maria João Frade and Jorge Sousa Pinto, and will also serve as companion application to the forthcoming book "Rigorous Software Development" by José Bacelar Almeida, Jorge Sousa Pinto and Simão Melo de Sousa, to be published by Springer early in 2009.