Research project offer

Location : ISAE SUPAERO, Toulouse, France

Department : Department of Complex Systems Engineering (DISC)

Research group : CASC (Critical Systems Architecture and Design)

Supervisor : Christophe Garion

Email : christophe.garion@isae-supaero.fr

OFFER DESCRIPTION

Title : A Lustre to SPARK code generator with correctness annotations

Proposed duration and period : 4 to 6 months starting from March 2023

Context (max 10 lines)	ENAC and ISAE-SUPAERO are developing a platform for code generation and validation for Simulink/Stateflow models named CoCoSim. CoCoSim generates Lustre code form Simulink/Stateflow models and we are developing LustreC, a compiler from Lustre with several backends, e.g. to low level languages such as C or SPARK or Horn clauses for formal verification. The overall objective of CoCoSim is to formally prove properties such as contracts or invariants on models or code artifacts. In order to do that, the first step is of course to ensure the correctness of the generated code. Whereas huge steps have been done in compiler verification by writing compilers and proving their correctness through proof assistants like Coq (cf. for instance CompCert for the C programming language or Velus for Lustre), we adopt here a different approach, namely translation validation: we generate along with functional code a companion logical specification whose correctness guarantees that the generated code has the same semantics as the original Lustre code. For instance, when generating C code through the C backend, ACSL specifications are generated and can be proved using the Frama-C tool. We now want to generate SPARK code, to use the complete GNATprove toolset to prove the generated specification and to compare the proof process with the C/ACSL/Frama-C one.
Objectives and work (max 20 lines)	The internship will focus on the SPARK backend of LustreC compiler. The applicant should complete the following tasks: 1. discover the Lustrec compiler by completing the SPARK backend. For instance, arrays are available in Lustre and are currently lacking in the SPARK backend. The C backend should serve as a guide for this implementation. 2. automatically generate assertions in the SPARK code to enable code correctness verification by external tools. These assertions should represent the exact behavioral semantics of the Lustre code. Assertions in SPARK code should be proven by the GNATProve tool or may require the use of the Why3 language and platform. 3. verify the assertions generator on various use cases.

Possibility to continue with a PhD (Yes/No) : Yes	
REQUIRED APPLICANT PROFILE AND SKILLS	
Study level (tick possible choices)	☐Undergraduate students (3rd or 4th year) X Master students (1st or 2nd year) ☐ PhD students
Required profile and skills	The applicant should have a good background in Computer Science. Knowledge of compilation, mathematical logic, formal specification and experience in OCaml programming will be appreciated.
Other useful information	