Internship Offer — SDS Team — INSA CVL

Design and Implementation of User-Friendly Concurrent Processes Description Language, and CTL* Model-Checker

Contact: Vincent Hugot — vincent.hugot@insa-cvl.fr

1 Practical Information:

Laboratory & Team:

Laboratoire d'Informatique Fondamentale d'Orléans (LIFO, EA 4022),
Systems and Data Security team,
INSA Centre Val de Loire,
88 boulevard Lahitolle
18022 Bourges

Supervisors:

The internship will be supervised by Vincent Hugot and Pascal Berthomé.

Duration and start:

5 to 6 months, at the candidate's earliest convenience.

The internship can potentially be followed by a Ph.D. thesis, for which funding is available.

Contact:

To apply or request additional information, send an email and a resume to `vincent.hugot@insa-cvl.fr`, preferably with an `[Internship]` prefix to the mail title.

Requirements:

- ◇ Being in the final year of a Master's Degree in computer science, engineering, mathematics, or equivalent.
- ◇ Proficiency in written English.

- Fluency in spoken English or French.
- Proficiency in Python3.
- Basic exposure to, and interest in, automata theory, formal grammars, and logic.

Useful Skills and Traits:

- Basic working knowledge of parser generators, such as lex+yacc/bison, ANTLR, Menhir, or any other. We'll likely use Lark (Python, EBNF, Earley+LALR(1)).
- Previous exposure to modal logics, CTL, LTL, CTL*, Büchi Automata, Model-Checking, etc.
- Though we shall work in Python, previous exposure to, and taste for, functional programming, in particular OCaml, Haskell, etc, would probably smooth things out.

2 Goals and Objectives

The internship has two distinct, though related and complementary, goals, which should be pursued concurrently.

- Short term: design and implement various improvements to an existing NFA framework, used for teaching and research (prototyping) purposes.

  Those improvements would be mostly on the teaching side of things, and related to process algebras and model-checking.
- Longer term: conduct a survey of the state of the art in process algebras and related formal languages, with an eye for simplicity wrt. a few examples and applications, for purposes of pedagogy and ease of integration with industrial processes involving non-specialists.

  The long term goal here is to facilitate the embedding of verified non-detrimental (i.e. that don't interfere with safety or functional goals) runtime security monitors in various industrial domains, e.g. automotive cybersecurity, following previous work with Valeo, workflow engines, following previous work with Qualnet, etc.

These goals are described in a bit more detail in the next sections:

2.1 Design and implementation of a simple process algebra and CTL* model-checking algorithm for pedagogical purposes

2.1.1 Implement CTL* model-checking algorithm

The CTL* task is the most straightforward. The first step will be to implement Büchi automata.

It is not expected that this task should take very long, as most of the data structures are already there and the algorithms are well-documented.

2.1.2 Design and Implement a Simple Concurrent Processes Specification Language

Given a corpus of target problems (Wolf & Goat & Cabbage Problems, Peterson's Algorithm and its generalization, the Dining Philosophers, and other classical concurrency problems) the intern will

- survey the literature on existing process algebras (e.g. CCS, CSP,. . .) and other formal languages (e.g. PROMELA) describing concurrent processes and their interactions with clear formal semantics
- design a specification language tailor-made for specifying the corpus problems, the primary objectives being simplicity (as few distinct concepts as possible) and clarity (straightforward application of those concepts)
- implement a compiler from this new language to transition systems, and to various target languages such as PROMELA, C, etc.

This task is expected to last for most if not all of the internship.

2.2 Survey Process Algebras and Verification Tools

Towards the end of the internship, the intern will undertake a more exhaustive survey of the literature on process algebras etc, this time with detailed comparisons of expressive power, user-friendliness, associated verification tooling, (both in terms of theory and of implemented programs), corpus of successful real-world applications if any, etc.

That survey may be fairly general or, likely, informed by specific industrial cybersecurity goals that will be given at the time. It will aim at compiling the necessary information for selecting (or developing) modelling languages and verification tools, depending on needs, and establishing a roadmap for future work.