sujet de stage de Master 2 recherche (2024)

Semantics and types for synchronous programming with state machines in a multi-periodic setting

Advisors: Patrick Baillot, Julien Forget and Sylvain Salvati (prenom.nom@univ-lille.fr)
Website: `https://pro.univ-lille.fr/patrick-baillot/`

Research lab: CRIStAL, Université de Lille, Équipe *SyCoMoRES*
Lab website: `https://www.cristal.univ-lille.fr/?lang=fr`

Duration: 4-6 months
Keywords: type systems, semantics, synchronous languages, OCaml

Context: In the area of critical real time systems it is crucial to have development methods which can ensure safety guarantees. Synchronous programming languages such as LUSTRE or SIGNAL [BCE$^+$03] have been introduced to provide a high level of abstraction for programming real-time systems. They are based on solid, elegant and yet simple mathematical foundations, which make it possible to handle the compilation and the verification of a program in a formal way. Their idea is to abstract away from real time and reason instead on a logical-time scale, defined as a succession of instants, where each instant corresponds to a reaction of the system. The program thus performs computations on flows, and some static analyses ensure that those are well-behaved. In particular, the *clock calculus*, typically presented as a type system, checks that computations that operate according to different clocks are combined in a correct manner.

However there is a need for more expressive languages. First, complex embedded systems, such as e.g. aircraft flight control systems, are generally *multi-periodic*, because different devices of the system have different physical characteristics. The PRELUDE language [FBLP10] has been designed precisely to program such systems by handling explicitly various real-time constraints (e.g. periodicity, deadline).

A second characteristic of control systems is that they are often *multi-mode*, where each mode implements a different behaviour; for the aircraft control system example one can think of take-off, cruise and landing modes. An intuitive way of thinking about multi-mode systems is by using a state machine representation, where each state corresponds to a mode [CPP05]. However defining the semantics of such a system is not straightforward and various choices are possible. Moreover the clock calculus, should be defined according to the semantics adopted.

In [For22, FF22] the PRELUDE language has been extended to a multi-mode setting. However this was done with a particular choice of semantics, which in some cases leads to large periods for the flows. Moreover the clock type system designed for this semantics makes it difficult to carry out correctness proofs.

Objectives: In this internship we propose to explore a general semantics for a multi-periodic and multi-mode synchronous language and to investigate the corresponding static analysis, in particular a dedicated clock type system. We will use for that the PRELUDE synchronous language of [For22] and propose to generalize its semantics so as to account for various ways of synchronizing the flows. In a second step, a clock type system will be defined which will reflect the choices made in the semantics. Finally an automatizable type inference procedure will be defined. If time permits, it will also be interesting to investigate if the type system can be extended with a notion of polymorphic clock.

Expected background: Ideally, the candidate should have some familiarity with functional programming, denotational semantics and type systems. Some basic experience of programming in OCaml or another functional language would be appreciated.

References

[BCE+03] A. Benveniste, P. Caspi, S.A. Edwards, N. Halbwachs, P. Le Guernic, and R. de Simone. The synchronous languages 12 years later. *Proceedings of the IEEE*, 91(1):64–83, 2003.

[CPP05] Jean-Louis Colaço, Bruno Pagano, and Marc Pouzet. A conservative extension of synchronous data-flow with state machines. In Wayne H. Wolf, editor, *EMSOFT 2005, September 18-22, 2005, Jersey City, NJ, USA, 5th ACM International Conference On Embedded Software, Proceedings*, pages 173–182. ACM, 2005.

[FBLP10] Julien Forget, Frédéric Boniol, David Lesens, and Claire Pagetti. A real-time architecture design language for multi-rate embedded control systems. In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung, editors, *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010*, pages 527–534. ACM, 2010.

[FF22] Frédéric Fort and Julien Forget. Synchronous semantics of multi-mode multi-periodic systems. In Jiman Hong, Miroslav Bures, Juw Won Park, and Tomás Cerný, editors, *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, April 25 - 29, 2022*, pages 1248–1257. ACM, 2022.

[For22] Frédéric Fort. *Programming adaptative real-time systems*. PhD thesis, Université de Lille, 2022.