We propose to study and advance reasoning by induction within an SMT context, starting from the work of Andrews et al. [4].

Currently, SMT solvers essentially handle first-order logic, and have very limited capabilities to reason by induction. In [1,8], the SMT solvers are used as black-boxes to discharge proof obligations of induction-based provers. Later on in [4], Reynolds and Kuncak proposed as solution the integration of explicit induction reasoning inside SMT solvers. Explicit induction schemas are used to perform the inductive strengthening of (the negation and skolemisation of) some universally quantified formula given as input. Their approach is automatic to automatically discover subgoals that can be then proved by induction. A disadvantage of the method is that it requires the definition of induction schemas before the proof starts, so it may happen that crucial induction hypotheses are missing or the provided induction hypotheses are unusable.

On the other hand, proofs by induction can also be automatically performed using implicit induction by the means of reductive inference systems [2, 3, 5, 6, 7], as shown when proving properties about conditional specifications. The lazy use of induction hypotheses, i.e., only when they are needed, and the easy development of mutual induction proofs are some of the advantages of this approach.
The proposed methods will be implemented as a prototype within the veriT SMT solver (http://www.verit-solver.org/).

**Principales activités**

The work can be organized as follows

- examine libraries of formal proofs to identify proof obligations requiring reasoning by induction, and build a library of benchmarks;
- develop an integration schema between induction and SMT:
  - as a first step, the ideas from reference [4] above will be implemented in the veriT solver;
  - then, the goal is to replace explicit by implicit induction reasoning;
- implement this integration schema;
- test the integration schema against the set of benchmarks.

**Compétences**

It is necessary to be acquainted with satisfiability modulo theories techniques. Knowledge of interactive theorem proving, reasoners by induction, or formal reasoning is a plus.

The development of prototypes will be done using the C language. Being familiar with a C-like language is thus required.

**Avantages sociaux**

- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- French courses

**Rémunération**

Salary: 2653€ gross/month