that are provided by the script execution of a package, that thus may be assumed by the scripts of
needed to guarantee execution without errors. It should also automatically compute post-conditions
the preconditions on the file system (such as assumed existence of some files or directories) that are
approach will allow us to automatically infer for each script what are
constraint language, in the same vein that formally verified abstract interpreters were recently
in this post-doc position is expected to be somehow generic, i.e. modular, with respect to the
constraint language to describe symbolic states, in that case it will
respect to the concrete semantics is expected. Also, a symbolic interpreter is tightly connected with a
formalized concrete interpreter. A formal proof of the correctness of the symbolic execution with
to the design of this symbolic interpreter, in a formal way using Why3, building up on the already
in a fully automated way is symbolic execution. The first goal of this post-doc position is to participate
in analyzing scripts of the Debian GNU-Linux distribution.
A first step of CoLiS was the definition of shell-like dedicated language (itself named \`CoLiS\") to
describe the actions operated by installation scripts. Unlike the POSIX shell, the syntax and the
semantics of the CoLiS language were defined using a formal approach [1], within the Why3
environment (http://why3.lri.fr, [2,3]). Why3 is an
environment for the formal specification and the proof of programs, developed in the Toccata team
where the post-doc position is offered. A reference interpreter of the CoLiS language was developed
in Why3 and formally proved correct with respect to the semantics [1].
Another initial step of the CoLiS project is the implementation of a parser of the POSIX shell
language, in order to analyze the real installation scripts. This syntactic analyzer already allowed us to
calculate statistical data on the full Debian distribution to figure out which constructs were widely
and which were not. A translation tool of these scripts into the CoLiS language is in progress.

Mission confiée

The initial analysis of the needs lead us to decide that a useful method for analyzing scripts semantics
in a fully automated way is symbolic execution. The first goal of this post-doc position is to participate
to the design of this symbolic interpreter, in a formal way using Why3, building up on the already
formalized concrete interpreter. A formal proof of the correctness of the symbolic execution with
respect to the concrete semantics is expected. Also, a symbolic interpreter is tightly connected with a
constraint language to describe symbolic states, in that case it will
be a constraint language of the file system tree. The constraint language we need is still to be
determined, in cooperation with the partners of the CoLiS project. The symbolic interpreter to design
in this post-doc position is expected to be somehow generic, i.e. modular, with respect to the
constraint language, in the same vein that formally verified abstract interpreters were recently
approach will allow us to automatically infer for each script what are
the preconditions on the file system (such as assumed existence of some files or directories) that are
needed to guarantee execution without errors. It should also automatically compute post-conditions
that are provided by the script execution of a package, that thus may be assumed by the scripts of
packages that depend on the former package. The second goal of the job is to contribute to the set-up of an infrastructure for automatic execution of the symbolic interpreter on all Debian installation scripts, in an adequate order with respect to packages dependencies. This approach should be also used for proving properties about composition of scripts, such as showing that installation followed by de-installation is more or less the identity.


**Principales activités**

Main activities (5 maximum):

- Design a symbolic execution engine for CoLiS programs
- Formalize this engine in Why3 and prove it correct
- Design an experimental platform for analysis of installation scripts
- Write reports and submit research articles

All these activities will be conducted in collaboration with other CoLiS project members. Regular working groups are organized at IRIF laboratory in Paris.

**Compétences**

The candidate should have a PhD thesis in the domain of formal methods for software engineering. A basic knowledge of an interactive (such as Coq, Isabelle, PVS) or autoactive verification system (such as Why3, Dafny, KeY) is expected. Some taste in implementing using functional programming (OCaml, Haskell, etc.) would be a plus.

**Avantages sociaux**

- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Paid leave
- Flexible working hours
- Sports facilities

**Rémunération**

Monthly gross salary : 2.653 euros