Flavio Leonardo Cavalcanti de Moura <contato@flaviomoura.mat.br>

ICTAC 2018 notification for paper 14

ICTAC 2018 <ictac2018@easychair.org> Tue, Jul 10, 2018 at 10:34 AM
To: "Flavio L. C. De Moura" <flaviomoura@unb.br>

Dear Flavio L. C. De Moura,

We regret to inform you that your submission 14

A constructive formalisation of the Modular Strong Normalisation Theorem

to ICTAC 2018 has been rejected.

Please find the reviews on your submission below.

The conference attracted 59 full submissions, many of them of high quality. Hence the competition was strong and the selection hard to make. In the end, we decided to accept 22 submissions and 4 more submissions conditionally.

Thank you once again for submitting to ICTAC. We regret to have disappointed you, but hope that the reviewer feedback will help you strengthen your paper for re-submission elsewhere.

Kind regards,

Bernd and Tarmo

----------------------- REVIEW 1 ---------------------
PAPER: 14
TITLE: A constructive formalisation of the Modular Strong Normalisation Theorem
AUTHORS: Flavio L. C. De Moura, Daniel Ventura, Raphael S. Ramos and Fabrício S. Paranhos

Overall evaluation: -1 (weak reject)

----------- Overall evaluation -----------
This paper proves the modular strong normalization, which is a property of abstract term rewriting systems, in Coq. The modular strong normalization gives a sufficient condition for a union of two terminating reduction relations to be terminating. The paper mostly consists of Coq scripts annotated with prose. The proof is done without resorting to classical axioms.

Points for:

+ A mechanized constructive proof of modular strong normalization (MSN).

Points against:

- Few insights on the technical results are given.

- The prose description of the mechnized proof is very easy to follow.

First of all, I appreciate the authors' effort spent on mechanizing the proof of the theorem. Proofs in proofs assistants require every single detail to be written down and so it's usually much more time-consuming than hand-written proofs.

As one who is not an expert on term rewriting systems, however, I fail to appreciate motivation for the work and significance of the result. I don't see why the authors pay attention to this particular theorem. Are the known proofs unsatisfactory in some sense (apart from the fact that no one has mechnized)? The authors seem to emphasize the fact that their proofs are constructive, which I agree (in general) is a good thing, but do not really discuss its implication. Is the algorithmic content of _this proof_ interesting?

A related point is that I'm not sure it is really the case that "the constructive approach is not the standard way to prove termination of a reduction relation." I think many proofs about SN can be conducted constructively even under the standard definition of SN (using negation).

Finally, I find the attempt at annotating every step of a formal proof with prose interesting but I'm not sure if it's working well. IMHO, it's not easy to follow. I don't really know why but maybe because Coq proofs are mostly backward whereas hand-written proofs mix forward and backward styles.

----------------------- REVIEW 2 ---------------------
PAPER: 14
TITLE: A constructive formalisation of the Modular Strong Normalisation Theorem
AUTHORS: Flavio L. C. De Moura, Daniel Ventura, Raphael S. Ramos and Fabrício S. Paranhos

Overall evaluation: -2 (reject)

----------- Overall evaluation -----------
The given paper describes a constructive Coq-proof for the modular strong normalization theorem.

The paper provides many fully explained Coq-proofs, so where for instance for every line in the following example proof-script its effect and its motivation is described.

```
Lemma SNinclUnion {A} {redA red'A: Red A}: (forall b, SN' redA b ->
                               forall c, red'A b c -> SN' redA c) ->
                        (forall a, (SN' ((refltrans redA) # red'A) a) ->
                      (SN' redA a) -> (SN' (redA !_! red'A) a)).
Proof.
  intros Hstable a HSNcomp.   (* comment *)
  induction HSNcomp.          (* comment *)
  intros HSN.                 (* comment *)
  apply inclUnion.                  .
  - assumption.                     .
  - intros b Hcomp.                 .
    apply H0.
    + assumption.
    + inversion Hcomp; subst. clear Hcomp.
      assert(H': SN' redA b0).
      {
        apply stabComp with a; assumption.
      }
      apply Hstable with b0; assumption.
Qed.
```

When reading the paper and at the same time having a running Coq session for seeing the intermediate proof obligations, one can easily understand how the formal proofs are performed.

So, the paper is definitely readable, it fits the topic of ICTAC, and the proofs are Coq checked (though there is one mistake in an informal proof).

Still, I do not recommend acceptance, because of the low significance of the contribution: the modular strong normalization theorem is a not too difficult result on abstract rewriting which only requires 10 lines of informal reasoning.

Moreover, the whole formalization is just 420 lines long, including even the basic definition for relations, (refl) transitive closures, relation union / composition, strong normalization, etc.

So, for someone who wants to learn Coq, this paper might serve as a nice tutorial

that explains many basic things. But the paper is a not a research paper
which presents a significant formalization task.

Details:

Please insert page numbers.

p2: should hold for all predicate"s".

p3: simulation technique [15]. Please insert a ~, so that there is no line break
between "technique" and "[15]."

p4: add labels "strongly sim." and "weakly sim." below the graphs, and perhaps
use less big/bold font for A and B

p4: Your informal proof is wrong:
The property
SN (->2^*#->1) inter SN(->2) = SN(->1 u ->2)
is not in general satisfied: Consider the relations
x ->1 y
y ->2 y
Then x not in SN(->1 u ->2), but x in SN (->2^*#->1) and x in SN(->2).

p5: compose: forall "a" "b" c. ... (alphabetical order)

p5: Please mention earlier that the initial part of Section 3 is mainly copied
from [13].

p5,6,11: I strongly wonder, why these definitions actually occur in
your theory at all. Shouldn't Coq itself have suitable definitions of
relations, relation composition, transitive closure, inverse, union, etc.?
Can some other Coq user reuse your development, if you define your own
type for relation, union, etc.?

p5: Do not present the natural number example: Do you want to write a Coq
tutorial or a research paper? If you really want to explain an inductive
definition, then take one from your development, e.g., SN or SN'.

p7: "since library ssreflect is not used": Here it would be interesting to
know
- why you deviate, and not just include ssreflect
- to which extend you had to adjust the theory of [13]

p7: be"l"ow, not bellow

p8: Let P be a patriarchal predicate. (no predicate"d")

p15: The proof is constructive. Be less repetitive, this is
mentioned too often (Chapters 1, 3, 4).

p15: "Constructive proofs ... are preferred in the context of Computer
Science".
This is quite a bold statement and I'm actually not convinced.
There are several large and relevant formal classical proofs
about computer science (operating systems, security of protocols,
programming languages, ...)
in HOL-based theorem provers.

----------------------- REVIEW 3 ---------------------
PAPER: 14
TITLE: A constructive formalisation of the Modular Strong Normalisation Theorem
AUTHORS: Flavio L. C. De Moura, Daniel Ventura, Raphael S. Ramos and Fabrício S. Paranhos

Overall evaluation: -1 (weak reject)

----------- Overall evaluation -----------
The content of the paper is precisely captured by the title, that is, it is a detailed step-by-step description of the proof in Coq of a theorem about term-rewriting systems.

The paper is written very clearly and I had no problem understanding the content.

I am not in favour of this paper, because I don't think it offers anything more than the formalisation itself. That is, the proof technique is given elswhere (which the authors admit) and the proof itself does not involve overcoming any Coq's caveats (which would be interesting for people struggling with similar problems). In fact, it seems quite short and straightforward. The authors leave extracting the computational content of the theorem as future work.

To sum up, my opinion is: good work, but the contribution not interesting enough.