Gmail

Luke Ong <c.h.luke.ong@gmail.com>

LICS2021 review response (submission 207)

2 messages

LICS2021 <lics2021@easychair.org> 11 March 2021 at 09:25
To: Luke Ong <Luke.Ong@cs.ox.ac.uk>

Dear Luke,

Thank you for your submission to LICS2021. The LICS2021 rebuttal period will be between 10 March and 14 March. Please note that all LICS deadlines are AoE.

During this time, you will have access to the current state of your reviews and have the opportunity to submit a response of up to 600 words. Please keep in mind the following during this process:

* The response must focus on any factual errors in the reviews and any questions posed by the reviewers. It must not provide new research results or reformulate the presentation. Try to be as concise and to the point as possible.

* The rebuttal period is an opportunity to react to the reviews, but not a requirement to do so. Thus, if you feel the reviews are accurate and the reviewers have not asked any questions, then you do not have to respond.

* The reviews are as submitted by the PC members, without any coordination between them. Thus, there may be inconsistencies. Furthermore, these are not the final versions of the reviews. The reviews can later be updated to take into account the discussions at the program committee meeting, and we may find it necessary to solicit other outside reviews after the rebuttal period.

* The program committee will read your responses carefully and take this information into account during the discussions. On the other hand, the program committee will not directly respond to your responses, either before the program committee meeting or in the final versions of the reviews.

* Your response will be seen by all PC members who have access to the discussion of your paper, so please try to be polite and constructive.

* We have received almost all reviews before the rebuttal, 98.5% of them, to be precise. Still, a very small number of papers have a review missing. If yours is one of those, and the missing review arrives during the rebuttal period, I will notify you.

The reviews on your paper are attached to this letter. To submit your response you should log on the EasyChair Web page for LICS2021 and select your submission on the menu.

----------------------- REVIEW 1 ---------------------
SUBMISSION: 207
TITLE: Supermartingales, Ranking Functions and Probabilistic Lambda Calculus
AUTHORS: Andrew Kenyon-Roberts and Luke Ong

----------- Overall evaluation -----------
The paper develops a method of establishing that a higher-order probabilistic program with continuous distributions almost surely terminates in the context of lambda calculus. There are a number of new ideas in the paper, including the use of

ranking supermartingales, sparse tracking functions, antitone ranking functions, and restricted sets of reduction strategies that ensure the Church-Rosser property. Although the method doesn't look like a completed project, the advances are impressive (at least, to a non-specialist).

For a non-expert in the area, the paper is a challenging read, with tough notation, but it contains a lot of examples (also further developed in the appendix). All in all, I think this is a very solid piece of work and recommend acceptance.

- What is \mathsf R in the definition of types?
- What is r.v. before Definition III.2?
- What is a.s. in Definition III.2?

----------------------- REVIEW 2 ---------------------
SUBMISSION: 207
TITLE: Supermartingales, Ranking Functions and Probabilistic Lambda Calculus
AUTHORS: Andrew Kenyon-Roberts and Luke Ong

----------- Overall evaluation -----------
Summary

This paper considers how to prove almost-sure termination (AST) for probabilistic higher-order programs. AST is the natural generalization of termination to probabilistic programs, and there has been plenty of work in recent years on verifying this property. Existing work largely considers first-order, imperative programs, developing the method of ranking supermartingales (and generalizations, such as antitone ranking supermartingales) to verify AST. This paper adapts these ideas to higher-order probabilistic PCF, and also with continuous sampling. The authors introduce sparse ranking supermartingales to make it possible to avoid defining the ranking function for "uninteresting" steps, and antitone ranking supermartingales to prove AST when the expected termination time is not finite. The last contribution in the paper considers when AST under the "standard reduction" (CBV) can be shown by proving AST under an alternative reduction strategy, which may be significantly easier for some programs.

General comments

This paper tackles an important and interesting problem. To date most work on termination of probabilistic programs has considered first-order programs (though there are certainly exceptions, as discussed in related work), and this work takes many ideas from the first-order literature and transports it to the higher-order setting. The idea of a sparse ranking supermartingale is nice, and addresses a particular issue in the higher-order setting---it may be inconvenient to define the ranking function for all reduction steps, and we may want to focus on just specific, "interesting" reduction steps. Likewise, antitone ranking supermartingales have been considered in the imperative setting, but this paper shows that they can also prove AST for higher-order programs.

My main criticism of the paper is in the last contribution, which has the potential to be the most interesting---a choice of reduction strategies is something that is inherently a feature of higher-order programs, and indeed it would be interesting to consider adapting the reduction strategy in order to simply reasoning about AST. The high level goal is clear to me. However, I found the section where this contribution is described (Section 6) to be totally impenetrable. The section starts by defining positions, subterms, alternative reduction relation, reduction relation on skeletons, and goes on, but it was never clear to me why the authors are doing all of this. Even a small roadmap would really help me understand why we are covering these things and why we are defining these concepts, and how they fit in to the overall goal. Currently, Section 6 is a series of highly intricate technical definitions with somewhat cryptic examples, but there is little intuition and no sense of how the pieces fit together, and what the goal is.

It seems to me that Corollary VI.12 is the main result of the section. But I was also very confused here, and before in the definition of reduction strategy. The example of reduction strategy shows CBV, but gives no further examples. In particular, is CBN a valid reduction strategy under this definition? If not, why not? And if so, VI.12 seems strange---if M is AST wrt to CBN, is it also AST wrt to CBV? That seems highly suspicious...

Specific comments

* p2: Church-Rosserness -> the Church-Rosser property
* p2: "We introduce a novel addressing scheme for the possible random choices in a program's execution, which ensures that the same random choices are taken at corresponding positions in alternative reduction sequences, so that the same eventual result can be reached." :: this description sounds like a probabilistic coupling is being constructed (?) is there any connection?
* p2: free in s -> free in M (?)
* p3: "Y lambda f n :" this syntax doesn't match the grammar (throughout)
* p3: "The score construct is irrelevant to termination except that it fails if its argument is negative, thus allowing computations to fail after finitely many steps." :: arguably, it is not clear whether termination is an interesting/meaningful concept in the presence of score/conditioning
* p3: for all N2. -> for all N.2 (all footnotes)
* p3: "Let T be a r.v." :: maybe give the type of T?
* p3: int P(d omega) Yn(w) :: maybe swap for consistency? int Yn(w) P(d omega)
* p4: extra space before footnote
* p4: "we have that (Yn)n≥0 is a 1-ranking supermartingale" :: this needs more justification. for instance, why is not an eps-ranking supermartingale, where does the 1 come from?
* p4: Equation (1): M_{T_n}(s) -> M_{T_n(s)}(s)?
* p5: Section 4 said that "A finite number of expected Y-reduction steps does not necessarily imply a finite number of expected total reduction steps." But under lemma III.8, it is said that T_M^Y < ∞ a.s. iff M is AST. More discussions about the connection between Y-PAST and AST/PAST would be helpful and why Example IV.1 is not rankable (by mapping to Y reductions needed) will be appreciated.
* p5: contractum :: what is this?
* p5: "to define M ⊕p N" :: bad linebreak here
* p5: "Y lambda f n." grammar
* p5: On the bottom of page 5, why the map described is a sparse ranking function? (I thought the expected Y reduction from i (+)_0.5 (Y \lambda fn. n (+)_0.5 f(n+1)) (i +1) is 0.5 * 1 + 0.5 * 0.5 * 2 + ... + 0.5^c * c + ... > 1)
* p6: maybe cite prior work on antitone ranking functions?
* p6: what are these examples supposed to show?
* p6: Example V.1, M1, 3/2 -> 2/3 ?
* p6: actual values -> realized values
* p6: Example V.5 is a nice example, but it seems like a curiosity rather than a central point. the space might be better used for something else, e.g., to explain the next section.
* p6: "(Yk) k \in omega" :: if omega is supposed to be the natural numbers, then this clashes with the previous omega on this line.
* p6: Plainly -> It can be seen that
* p7: "Therefore by setting epsilon2(x) = ..." :: where does e come from? this needs more explanation.
* p7: "Even with this restriction" :: what restriction?
* p7: a roadmap here would be nice. you define positions, then subterms, then a more general reduction relation, then a reduction relation on skeletons, ... but where are we going and why are we doing all of this?
* p8: missing space: M1 M2
* p8: Definition VI.1 to be clear, is this relation is non-deterministic? that is, is it for any position alpha? also, this definition is hard to parse---consider putting "then" after the commas.
* p8: "Labelling the pre-chosen samples by the positions" :: where is this labelling? example?
* p8: footnote 6: why not just use a different arrow?
* p8: "with one more caveat introduced later" :: hard to follow. where will this

caveat be introduced?
* p8: Example VI.2: should A[x] be A[X], since X is a hole?
* I see that this defines the semantics of A[r], where A is a skeleton, and r is a vector of real numbers in Sec 2. But here, I'm not sure if A is a skeleton, or a program, and what does A[x] and A[X] mean?
* Is "I" in A[x] a function? You used I for intervals too.
* "r occurs twice in the conditions on s: once as the value a sample must take, and once in the location of a sample". I have difficulties to see where r appears as the location of a sample.
* p8: "this set is unmeasurable" :: is this obvious? why?
* p9: "This is still not sufficient" :: for what?
* p9: "in which case" :: what case is this referring to?
* p9: I could not understand the purpose of Definition VI.5.
* p10: is CBN a reduction strategy, under your definition? if not, why not? if so, VI.12 looks suspicious. Another example of a reduction strategy would help.
* p11: Applications: this section is highly speculative, since it does not appear that any of the claims have been worked out in detail. maybe this is better under future work, or discussion?

More on Section 6:
* At the bottom of page 7, what kind of restrictions are needed to exclude the problem caused by the reduction strategy making copies of samples to be identical v.s.. independent?
* In Definition VI.I, could you give more intuition why λ should not occur after Y in α?
Is the example (lambda x. x 0 + x 0) (lambda y. sample) the kind of programs where making copies identical vs. independent causes problem?
Is that example meant to illustrate why λ should not occur after @2 in α?
Also, it will be helpful to highlight restriction here has to do with restriction on reduction strategies that satisfy confluence.
* Probably some high-level intuition for complicated cases in Definition VI.5, besides the discussions in examples VI.4 and example VI.6
* In Definition VI.8, what is s \circ i(M→N) ? Is circ composition? The type doesn't seem to match.
* Theorem VI.11 was confusing to me at first. Discussing how all the previous definitions imply the restrictions on reduction strategies would help. (Maybe moving some lines from the paragraph under Theorem VI.15)

----------------------- REVIEW 3 ---------------------
SUBMISSION: 207
TITLE: Supermartingales, Ranking Functions and Probabilistic Lambda Calculus
AUTHORS: Andrew Kenyon-Roberts and Luke Ong

----------- Overall evaluation -----------
The paper develops new techniques for proving almost-sure termination of probabilistic programs written in a prototypical higher-order typed functional language. This adds to a large body of work on formal methods for probabilistic programming language. The paper improves on previous results by dealing with higher-order programs and by considering a random choice operator (called sample) with a continuous distribution. The method underlying termination proofs is based on (sparse) antitone ranking functions.

I am not really an expert in the field but it does not seem that these ingredients are very original. However they are nicely combined, the paper is very well written, with a good set of illustrative example programs. The technical work is presented in a clear way, with the result that it does not seem overly difficult.

It is regrettable that the author(s) do not comment more on their ability to prove completeness of the method [for imperative programs and ranking functions in a well-founded partial ordering, completeness is easy to show]. Is this just lack of time or is there a well-identified stumbling work that could be mentioned? If a program is AST every reachable term has a stopping time and these stopping times can be compared when we're talking about terms reachable from one another. Now we have to turn this into a ranking function. Is the problem caused by the existence of loops between reachable terms (even when guaranteed to terminate almost surely)? Is the problem about showing that a complex set of recursive inequations indeed admit solutions?

Best wishes,
Leonid
(LICS'21 PC chair)

Luke Ong <Luke.Ong@cs.ox.ac.uk>
Draft To: LICS2021 <lics2021@easychair.org>
11 March 2021 at 09:34

Dear Leonid,

Many thanks for forwarding the reviews.

[Quoted text hidden]