

From: **BEAT 2019** beat2019@easychair.org
Subject: BEAT 2019 notification for paper 6
Date: 27 November 2018 at 18:53
To: Vasco Thudichum Vasconcelos vmvasconcelos@fc.ul.pt

B

Dear Vasco,

We are happy to inform you that your talk proposal has been accepted. Find the reviews attached. Please take them in careful consideration when preparing your presentation. You will have 20m, including time for questions.

Notice that POPL early registration deadline is December 10.

Looking forward to seeing you in Cascais, warm regards,
António and Jorge

----- REVIEW 1 -----

PAPER: 6
TITLE: Checking the equivalence of context-free session types
AUTHORS: Andreia Mordido and Vasco Thudichum Vasconcelos

Overall evaluation: 2 (accept)

----- Overall evaluation -----

SUMMARY

The authors sketch an algorithm for deciding equivalence of context-free session types building on some prior work on BPA equivalence

PROS/CONS

- + Very interesting work addressing original problem
- + Potential for entertaining presentation
- Abstract assumes some knowledge on prior work, especially [5]. I think at least the notions of unnormed process and expansion list/tree should be recalled for those unfamiliar with [5].
- Given that the existing literature on context-free session types is quite limited, it is somewhat surprising that there is no mention of these two related works

[A] Jens Aagaard, Hans Hüttel, Mathias Jakobsen, Mikkel Kettunen:
Context-Free Session Types for Applied Pi-Calculus. EXPRESS/SOS
2018: 3-18

[B] Luca Padovani: Context-Free Session Type Inference. ESOP 2017:
804-830

In particular, [B] is very relevant as it presents a formulation of context-free session types that admits straightforward equivalence, subtyping and inference algorithms, at the cost of imposing similar structure between processes and session types. Thus that work doesn't really take into account the monoidal laws of sequential composition, which complicate matters substantially. Adding remarks along these lines would provide better context and more compelling motivation for the work.

- I found the choice of examples (page 3) somewhat unsatisfactory. The first one is positive, but it concerns regular session types with a finite state space (even though there is sequential composition, those types can be rewritten so that recursion is always in tail position). The second one addresses context-free session types that are not regular, but it is a negative one. In case there is no space/time to illustrate more than one example I would very much prefer seeing the algorithm at work on a single positive example with context-free session types that are not regular.

EVALUATION

I think the shortcomings listed above can be easily addressed, therefore I'm in favor of accepting this talk.

DETAILS

page 2: "noted that any process..." any BPA process?

page 3: "any true negative result", why "true"? are there negative results that have been dismissed for any reason?

page 3: "conviction"

----- REVIEW 2 -----

PAPER: 6

TITLE: Checking the equivalence of context-free session types

AUTHORS: Andreia Mordido and Vasco Thudichum Vasconcelos

Overall evaluation: 3 (strong accept)

----- Overall evaluation -----

SUMMARY

This talk proposal reports ongoing work on an algorithm for checking type equivalence for context-free session types, proposed by Thiemann and Vasconcelos. Here type equivalence between two types means bisimilarity between their respective LTSs.

The algorithm, implemented in Haskell, appears to rely heavily on known structures and techniques by Jancar and Moller [5]. The short abstract gives this algorithm, and illustrates it via examples of (non) equivalent session types.

ASSESSMENT

This is an interesting and well-written talk proposal, and so I am in favor of acceptance. The structure of the document gives me confidence that a presentation at the workshop would be interesting and engaging.

SUGGESTIONS

If accepted for presentation, I suggest the authors to clarify whether they needed to adapt/extend the (known) techniques in [5], which are very well known by now. Indeed, from the short abstract, their implementation seems to correspond to a specific (deterministic) case of the language in [5], but this is not completely clear.

(Just to clarify: even if this is indeed a not-so-surprising adaptation of the known techniques in [5], I find it very significant to have this kind of results formalized for session types, together with their soundness/completeness properties, and implemented in actual tools.)

I also wonder about the one hundred examples mentioned towards the end of the abstract - are these instances generated randomly? do they correspond to "useful" (non) bisimilar protocols?

----- REVIEW 3 -----

PAPER: 6

TITLE: Checking the equivalence of context-free session types

AUTHORS: Andreia Mordido and Vasco Thudichum Vasconcelos

Overall evaluation: 2 (accept)

----- Overall evaluation -----

The proposed talk is about an on-going work on checking type equivalence for context-free session types. The proposed solution follows from previous results on bisimilarity (un)decidability (Jancar and Moller, 1999). This abstract describes a Haskell implementation of the proposed solution, which haven't been proved correct yet.

There are a couple of aspects that I would like to see addressed in the presentation if the proposal is accepted. In particular, I think that this contribution needs to be presented in the context of the other few proposals in the literature dealing with context-free session types. Mainly,

- Padovani, L. Context-free session type inference. ESOP 2017, which describes a functional language with context-free session type featuring type inference. Moreover, its implementation relies on a notion of type-equivalence. Consequently, I wonder about the differences between the two approaches and whether the decision problems are different in these two settings. This should be made clear during the presentation.

- Aagaard, J., Hüttel, H., Jakobsen, M., & Kettunen, M. Context-Free Session Types for Applied Pi-Calculus. Express/SOS 2018. This paper addresses the problem of type equivalence and links it to checking bisimilarity on BPA. Besides, it is also mentioned that an algorithm could be obtained by re-using previous results on BPA, in particular, Yoram Hirshfeld & Faron Moller (1994): A Fast Algorithm for Deciding Bisimilarity of Normed Context- Free Processes. I found the above paper quite connected to this contribution and I think that a comparison will be also of help.

I found the topic well-suited for BEAT. I also think that the authors can easily address the previous suggestions in order to put in context this contribution wrt other related lines of works. For these reasons, I'm in favour of acceptance.