Review of Prioritise the Best Variation

Rating: Reject
Check a revision?: Yes

Evaluation

I am summarizing in this section my main concerns that led me to recommend rejection. I am also making suggestions on what should be improved. Detailed comments, per page, are listed in the next section. The latter also reflect on my thought process as a reviewer, which should in turn help the authors in a revision.

Overall, I found this submission prepared in a haste: the writing is not careful (i.e., missing distillation of a compelling line of argumentation), nor polished, there are formatting mistakes, missing definitions, and I also found what I believe are mistakes in the formalization. I will comment on them in more detail below.

I also found it concerning that the authors fail to reveal their prior publication [3], from which, as far as I can tell, the present submission only differs by the addition of proofs and the example of Milner's cyclic scheduler. Moreover, it seems that the authors have accidentally omitted the translations of PCP into PGV, which are given in Section 3.2 in [3]. These definitions, however, would be necessary for understanding the technical development in Section 4.5. The authors should declare the dependence on prior work and detail how they contribute over it.

I did find smaller technical mistakes, which I'm commenting on below. My main concern with regard to the technical development, however, is my worry that the dynamics of PGV, as presented in Figure 1 is not correct. The dynamics seems to be stratified, in the sense that configuration reductions may rely on term reductions but not vice versa. The subject reduction proof then relies on this stratification, appealing to subject reduction of terms, for example, in the proof of subject reduction for configurations. However, I have been unsuccessfully trying to construct a derivation using the rules in Figure 1 for the examples presented in the paper.

For example, let's consider the example from page 21:

• $\mathbf{let}(x, x') = \mathbf{new\ in\ spawn}(\lambda().\mathbf{echo}_x); x'$

Here, it seems that I have to apply rule E-LiftM, which would require me to step **new** using a term reduction, but there is none.

Related, there also seems to be a mistake in the grammar at the bottom of page 6. Is $\mathcal{F}$ really needed, or if so, should it not be incorporated into $\mathcal{G}$ and/or E? It would be really helpful if the authors provided a derivation using the rules in Figure 1 for one of their examples.

If the dynamics cannot be stratified in this way, but instead is mutually inductive, then the subject reduction proof needs to be by simultaneous induction, also.

While I generally understand the use of priorities to rule out circular waiting dependency, I am unclear about what the invariants are that the type system maintains in terms of priorities. Also, I didn't get a clear understanding of the handling of priorities when it comes to closures. With regard to the former, for example, I am unclear about what the precise relationship is between the priority of the sequent and those in the context. There seems to be some relationship that is imposed by typing, but I was unable to distill the pattern, nor have the authors properly explained this. Closures are really tricky, and the authors should devote more time and space to convincingly argue that the enforced treatment is correct. Ideally,

there would be a discussion based on a number of examples, including one that would be ruled out by typing, so that the reader can get an understanding of what the key concerns are. The current treatment is short, with the only justification for the chosen treatment: "Closures suspend communication, so T-Lam stores the priority bounds of the function body on the function type, and T-App restores them."

Another question I was left wondering about is whether priorities are inferred by typing or whether the programmer has to provide them. The former seems to be the case, but this should be made explicit. Also, it would be helpful to provide a complete, unsuccessful typing derivation for Example 2.2. Two individual derivations, for each thread, are given on page 10, which show that the derivations impose contradicting orderings on the priorities. It would then also be helpful to show the rule that composes the two derivations and fails because of the contradiction. Which rule would that be?

Another technical comment I have is on the distinction between variables and channels (which the authors refer to as names). The latter are generated at runtime (via **new**), and are also gathered, in the environment. I believe that's also the reason why the usual theorems (preservation and progress) are stated relative to a non-empty context. However, the formalization should guarantee that the respective terms are closed. It would also be helpful if different meta variables were used for variables and names. Currently, x, y, z are used for both.

In terms of the writing, I think that the motivation for the need of a kind of multicut is very wishy-washy. For example, the claim that a cut-based approach lacks modularity is not substantiated. The avoidance of issues with regard to structural congruence (i.e., not type-preserving), is neither fundamental to a cut-based approach, i.e., the use of structural congruence in the first place is a choice, but not necessity.

Especially for a journal submission, the treatment of related work lacks substance. Related work should be contrasting, i.e., summarize related work and point out differences with presented work. Also, it should be comprehensive and extensive. The following works should be included as well: [1, 2, 4].

Detailed Comments

- References: It would be nice if references included DOI links.
- Abstract: It seems rather unusual to include citations.
- P2, "there have been several attempts at developing Curry-Howard correspondences": Attempts, implies that it failed.
- P2, "they achieve by restricting structure of": the structure of
- P2, "processes and shared channels to trees": What is meant by shared channels? Please clarify.
- P2, "communicate via exactly one series of channels": What is meant by series of channels?
- P2, "can be tweaked to satisfy reflection": What is meant by reflection here?
- P2, "succinctness of process calculi": What is meant?
- P2, "Concurrent lambda-calculi maintain a clear separation between the program ... and the configurations": True also for some other session-typed languages.
- P3, "Priority GV offers a more fine-grained analysis of communication structures": Clarify?
- P3, "is typeable and guaranteed deadlock-free": Guaranteed to be
- P4: Why does top denote a lower bound and bottom an upper bound? Shouldn't it be the other way around?
- P5, "top-most connective": Left-most?
- P5: "to denote the minimum and maximum": ..., respectively. Also, that listing seems to be missing max!
- P5: The programs in Example 2.1 are not well-formed wrt abstract syntax of terms defined on page 5. For example, the term "recv x" cannot be parsed.

- P6: I am not sure how to parse the grammar of configurations. The two productions seem not clearly separated. I believe it should be phi ::= closed circle — open circle and then separately, C, D, E ::== ...?
- P6: again, the definitions for the flags lack space to separate them. Also, this seems not a proper inductive definition, i.e., base cases are missing. Moreover, what seems required is a left to right reading here, i.e., the right to left reading doesn't seem to make sense. I think the proper way to define this would be as an inductive definition over the configuration, in particular decomposing $C||D$. The introduction of "+" seems unnecessary, moreover the semantics of it is not defined. And, after having read the paper, these definitions are later on never used.
- P6, "overlap between flags and priorities": Could you clarify what the implications of this overlap are? Or does it just mean that the color indicates what "circle" means?
- P6, grammar, bottom of page: Why is K (defined earlier in the term syntax) a value? Shouldn't K be mentioned in rhs of E? Also, I do not understand the need for the thread evaluation context, nor do I understand how it fits with the remaining notions, E and G. F doesn't show up there. Could it be that F should appear in the rhs of G? Maybe instead of the hole?
- SC-ResComm: Why is the side condition necessary? Doesn't name restriction generate fresh names?
- P8, "allows restriction to applied": to be applied
- P8, "We no longer require that every child thread finishes by returning a terminated channel": Doesn't that have implications for linearity?
- P8, middle of page, in (b): not sure how to parse this arrow/implication?
- Fig2: rule T-Absurd would admit weakening, so is wrong.
- P8, "In PGV, we keep track of a lower and upper bound on the priorities of a term, i.e., while evaluating the term, when does it start communicating, and when does it finish": Nitpicking, but this is not really properly formulated. What does the "it start" refer to? Use of resp. would be helpful.
- P8, last sentence: At this point it's not clear to me how the lower bound is approximated.
- T-Pair: why should $p < \mathsf{min_{pr}}(\Delta)$?
- P9: At first, I didn't see how the typing rule for New ensures that both endpoint have same priority. Maybe remind the reader that S and $\overline{S}$ requires priorities to be the same.
- P10: Example lambda abstraction, I don't think that the metavariable A was ever introduced. Also, I am not familiar with type schemas, some explanation would be helpful.
- Fig2, runtime typing: Shouldn't these rules incorporate the earlier defined + operator to ensure that there is only one main thread?
- About top and bottom on lolli: aren't they used counter standard usage? I.e., shouldn't the greatest upper bound be top? After all the priorities on the sequent are joined.
- Figures 3-5: I have not checked them for mistakes. What is the significance of providing this syntactic sugar?
- Proof of Lemma 3.1 misses cases T-Var and T-Const.
- Lemma 3.2: Metavariable Theta has not been introduced earlier. For typing environments, only Gamma and Delta have been introduced. Also, shouldn't there be a relationship on p and q?
- Lemma 3.2 proof, 2nd case: $\mathsf{pr}()$ should be $\mathsf{min_{pr}}()$ And in general, it is not clear what is meant by right arrow with label V/x. What is the semantics of it? Note, I didn't check the remaining cases. But in general I find the proof not detailed enough. Several steps are omitted.
- The proof of Lemma 3.3 is sloppy. It omits the typing of evaluation contexts, which is all glossed over in the last case, dismissed as immediate. Also, it is stated for open terms. Presumably, Gamma can only contain channel names at this stage, but that is not made precise, nor guaranteed.
- Proof of Lemma 3.4: it does not seem to go by induction of C = C' because structural congruence is not an inductive definition. Instead, it seems typing derivations of the structurally equal configurations are provided and it is shown that the same type is derived.

- P22: I am surprised to read that progress seems to allow ready terms. Shouldn't a stronger progress statement be possible due to linearity?
- Lemma 3.7: here the remark is made that Gamma contains only session types. Reminding of my earlier remark regarding closed terms, I think this remark should also be made for subject reduction?
- P22, "we opt to move all nu-binders to the top": Where is that done?
- P22, explanations below Lemma 3.7: This seems important, but it should be elaborated, I wasn't able to understand the details. Also, I don't remember that the term canonical forms has been defined. Is a term in canonical form if it is ready? (Reading on, I see that the term canonical form is now defined. So, the sequencing here is probably not ideal.)
- P24, $\mathsf{pr}()$: here pr is used, earlier $\mathsf{min}_{\mathsf{pr}}$. Not sure whether that is intended, but it also caused mistakes in an earlier proof (see earlier remark). Actually, this definition uses again $\mathsf{min}_{\mathsf{pr}}$. Shouldn't it be, for $\mathsf{min}_{\mathsf{pr}}(\Gamma, x : A) = \mathsf{min}_{\mathsf{pr}}(\Gamma) \cap \mathsf{pr}(A)$?
- P26, "Finally, since our reduction relation is a strict subset of the reduction relation in the original [DG18b], we defer to their proofs.": I am not convinced by this argument, i.e., why should the proof that was carried out for a bigger system hold for a subset?
- P28, "Let this process be let this be P_i...": grammar
- Proof of Thm 4.4: this is not a rigorous argument. I would like to understand better the architecture of the proof. It seems to be going by cases, so I want to be convinced that they are exhaustive. Also, various inferences are not properly justified.
- Lemma 4.5: this lemma seems to rely on a translation (notation (.)), which has not been introduced yet, as far as I can tell! See also my remark in the previous section.
- Fig.10: What is the semantics of this arrow/implication?
- P36: It would be helpful to already alert the reader to the figure on the next page when discussing the setup here.

References

[1] M. Coppo, M. Dezani-Ciancaglini, L. Padovani, and N. Yoshida. Inference of global progress properties for dynamically interleaved multiparty sessions. In *COORDINATION*, 2013. doi: 10.1007/978-3-642-38493-6_4.

[2] M. Coppo, M. Dezani-Ciancaglini, N. Yoshida, and L. Padovani. Global progress for dynamically interleaved multiparty sessions. *MSCS*, 26(2):238–302, 2016. doi: 10.1017/S0960129514000188. URL `https://doi.org/10.1017/S0960129514000188`.

[3] W. Kokke and O. Dardha. Prioritise the best variation. In *FORTE*, volume 12719 of *LNCS*, pages 100–119. Springer, 2021. doi: 10.1007/978-3-030-78089-0_6. URL `https://doi.org/10.1007/978-3-030-78089-0\_6`.

[4] B. Toninho, L. Caires, and F. Pfenning. Higher-order processes, functions, and sessions: A monadic integration. In *ESOP*, volume 7792 of *LNCS*, pages 350–369, 2013. doi: 10.1007/978-3-642-37036-6_20. URL `https://doi.org/10.1007/978-3-642-37036-6\_20`.