Reply to reviewers
LMCS # 6784 Manuscript:
"Compositional equivalences based on Open pNets"

October 10, 2021

Dear Editor, dear reviewers,

Please find enclosed the revision of our submission to LMCS. We improved our article according to the comments made by the anonymous reviewers.

We would first like to thank the reviewers for their analysis of our paper, and the useful advices they gave us.

We detail below the major changes we made in this revision. After this, we will review in the next pages the comments of the reviewers and explain how we addressed them. *Comments that could be addressed trivially and exactly as suggested by the reviewers are omitted.*

In this new version, you will find the major improvements listed below:

-

Best regards,
Rabéa Ameur-Boulifa, Ludovic Henrio, Eric Madelaine.

As explained above, we quote below and reply to text requiring changes that we did not address in a straightforward manner. Major points of the reviews have not been omitted in any case.

Annex 1: response to first reviewers' comments

- The definition of pNets is cumbersome. I understand that this work extends prior work where the model has been motivated already, but I have my reservations on the utility of the formalisms. I found the running example (although useful) to be hard to understand. It would perhaps be useful to provide anecdotal evidence showing that comparable formalisms cannot handle certain features of the example (which would justify the complexity of the model used).

> The definition of pNets is indeed not simple but as it is placed at a very generic and semantic leve,l we do not believe there are good alternatives in the literature. Process algebras for example do not feature the hierarchical and parametric aspects of pNets and defining equivalence checking tools on these frameworks is difficult. Of course the examples can be expressed in any concurrent Turing-complete model but none of the existing works feature the same results as ours on the modelling of existing parallel operators, and compositional reasoning on equivalence.
>
> We added most of these arguments in the introduction, and the related work section discusses details on equivalence checking. We also discuss in more details the relation with process algebras in the paper.

- I was left unconvinced that using open automata as an intermediary structure to define the bisimualtion relations is the best approach to develop this material. The authors should give more convincing argument that such definitions could not be adequately defined directly on pNets (an already complex framework). It left me wondering: 1) How do you validate the translation of pNets to Open Automata? Shouldn't there be some result that states that the translation is semantic preserving?

> Indeed such a theorem could be stated and proven: the two semantics are behaviourally equivalent, this would be proven by exhibiting a complex bisimulation between the two models, restricted only on the closed non-parametric part. In practice a proof of equivalence would make the article much longer with a lot of technical developments; we believe it would make the article more difficult to read. However we added a discussion on equivalence between semantics at the end of Section 2.2.

2) Why not work directly in terms of Open Automata (and forget all about pNets)? A prima facie, they appear to be of a comparable complexity to pNets (if not simpler). There are few people who would be interested in using a framework that relies on the second rule in Def 9 as a modelling language...

> The reason for the coexistence of the two formalisms is that one is hierarchical and adapted to modelling, the other is flat and more based on logic, and adapted to semantic/equivalence reasoning. We added one subsection in the introduction that presents the two formalisms and the reason why both exist.

- The definition of bisimulation is slightly non-standard (eg Def 7). For instance, why does the transfer property require a "set" of matching transitions? Normally, the existence of one suffices. Moreover, given that a set is required, shouldn't there be an additional requirement that the matching set is necessarily non-empty?

> The following explanation has been added to the introduction:
>
> " Bisimulation over a symbolic and open model like open pNets or open automata is different from the classical notion of bisimulation because it cannot rely on the equality over a finite set of action labels. Classical bisimulations require to exhibit, for each

transition of one system, a transition of the other system that simulates it. Instead, bisimulation for open automata rely on the simulation of each open transition of one automaton by a set of open transitions of the other one, that should cover all the cases where the original transition can be triggered. ”

Concerning non-emptiness, the fact that the matching set of transitions completely cover the cases of the original ones by nature prevents the empty set from being a good match.

- In general, weak bisimulation is *not* a congruence for all types of interacting contexts. For instance, in the case of CCS, the classsical counter example is that whereas tau.a.0 bisim a.0 we do *not* have tau.a.0 + b.0 bisim a.0 + b.0 Given the generality of the pNets framework I found the claim stated in Theorem 7 to be suspicious (and hard to ascertain). The authors should perhaps provide more intuition why this is the case for their framework.

Indeed, the plus operator of CCS does not verify the requirements “non-observability of tau transitions” for the weak bisimulation to be a congruence. We can define the weak bisimulation for CCS but it does not verify the congruence property. **[TODO:define the bisimulation, or encode the nondeterministic choice operator ?]**

This is now illustrated by Example 4 and 5 in the article.

l -6: why not use an asymmetric symbol instead of $\otimes$? A

$\otimes$ is a form of composition: we have changed for something similar to $\circ$, renamed it into $\odot$

l 26: ”Variables in V1 and V2 verify the predicate $P_{s,t}$” I am not sure what you mean by this.

This has been rephrased:

the states are related when the value of the variables in s and t verify the predicate $Pred_{s,t}$. Note that this implies that the free variables of $Pred_{s,t}$ belong to V_1 and V_2.

l 27..28: why should V1 and V2 be disjoint?

Variables used in $Pred_{s,t}$ should refer to variables of the automata, but there must be no ambiguity on which automaton it refers to. If the variable x should have the same value in both automata, we add x=x’ to Pred. Otherwise hidden information would be confusing and the soundness of the theory would be difficult to ensure.

We explained this point better in the text

pg 14: why is the term ”Specification pNet” capitalised?

Indeed, we renamed in a consistent manner all our examples

pg 17: l -4: I found this condition hard to map to the intuition given on pg 18 line 1..2

We explicitly added the following explanation:

statement (2) states that all the open transitions where a hole does a τ must be of the shape given in statement (1)

Annex 2: response to second reviewers’ comments

SYNOPSIS

The paper proposes a generalisation of transition system specification (well known in the literature on structural operational semantics) called an open automaton, that generalises the traditional transition systems in two dimensions: hierarchical composition and conditional transition that may update the variables once the transition is executed (akin to extended finite state machines). The paper mainly uses open automata for semantic purposes: first, to define the semantics of parametrised networks (a graph-theoretic specification language); second, to define the (weak) bisimulation relations on the states of open automata (in turn between the states of a parametrised network). Lastly, congruence results are established both for strong and weak bisimulation relations w.r.t. the hierarchical composition operator.

EVALUATION

The paper is fairly well written and clearly falls in the scope of the journal LMCS; however, it comes short in reaching the quality transpired by an LMCS journal. The introduction needs an improvement especially in terms of motivation; this seems to be the case with the formal definitions which in some cases need to be properly defined (see the detailed comments). For e.g., why were pNets introduced?

Since pNets have been introduced for over 15 years, why are we studying (weak) bisimulation and establishing the usual basic facts about it now? Were there any case studies in your experience that suggest the need for such results? What are the reasons to pick bisimulation not simulation relations for the purpose of conformance between a specification and its implementation? It would be great to support these with some evidence.

> pNets had been used to do model-checking based on the translation (of closed pNets with parameters instantiated) to other formalisms, we now study them directly. We decided to first design an equivalence relation rather than preorders that reveal to be more complex (simulation is discussed in our new conclusion). In our example it happens that we have weak equivalence which is stronger than weak simulation in any case.

Secondly, the choices made behind the mathematical notations and definitions are at a low level of abstraction; unfortunately, which makes the proofs in the appendix as unreadable. This is disappointing since most of the theorem/lemma statements are those that one would expect to be true.

In addition, purely from the semantic point of view, the theoretical development is rather straightforward and most of the definitions/results already known in the literature on process algebras are engineered to the setting of open automata.

> We believe that this argument is rather subjective. From an objective point of view, the parameterised aspects of our model are from our point of view quite new, as well as the conditions for congruence of weak bisimulation. We added arguments highlighting the novelty in the introduction.

As a result, unfortunately, I do not endorse the publication of this manuscript. Nevertheless, I do believe that the paper may be more fitting in a journal which endorses both software engineering and formal methods.

DETAILED COMMENTS

> In the following, we simply removed from the list the comments for typos that we have directly corrected.

P1 Abstract L10 *... that includes parameters, and ...* parameters for what? Please elaborate more on this in the introduction.

> This has been reworded.

P2 L24 *... provides a cleaner version...* I am afraid such a sentence does not add any value if there is no explanation discussing it in detail somewhere later in the paper.

> We change this for a more precise statement.

P3 L-11 What is an action algebra? I believe it is an undefined term in the paper.

We have improved the paragraph defining term algebras to identify more precisely the action algebra.

P4 L-6 *We distinguish two kinds. . . .* please rewrite this sentence

Done

P4 L20 A minor point: Technically, $a^{i\in 3}$ is not allowed by your syntax since indices need to also appear as a subscript.

[TODO:on a oublie celui-la... ?]

P4 L22 I am afraid we do not use abusive vocabulary in mathematics, rather we only abuse notation. Please rewrite.

Done

P4 L26 *
uplus is the disjoint union of sets* Please rewrite this sentence; it is considered to be a bad practice to start the sentence with a symbol (cf. Donald Knuth on mathematical writing available from $https : //jmlr.csail.mit.edu/reviewing - papers/knuth_mathematical_writing.pdf$).

Done

P4 L-9 *We denote y ¡- e a substitution* What do you mean by this? Do you mean that y¡-e is a partial function from the set of variables to terms that is only defined for the variable y as the term e?

This paragraph has been fully rewritten, giving a more formal definition of substitutions.

This includes answer to this comment and the following three.

P4 L-9 *The application of a substitution. . . * Its strange to denote an application of a substitution without any denotation for its argument.

P4 L-7 The notion of substitutions on indexed sets is more subtle; they may not be a function unlike a substitution in the traditional setting. For e.g., $x_1 < -e, x_2 < -e' with x_1 = x_2$.

P4 L-6 *
otimes is the composition operator. . . * Is this operation totally defined? This even makes it necessary that substitutions are defined properly.

P5 Def 1 Is vars(s) defined earlier?

Yes, in the Notations section, term algebra paragraph.

P5 Def 2 How is the symbol l quantified in the clause pertaining to synchronisation vectors.

This comes from the definition of indexed sets in section 2.1: the superscript $l \in I_k \uplus J_k$ means l as ranging through the set $I_k \uplus J_k$

P6 Def 3 In the first clause defining Sort, the notation *?x ¡- x* is not defined since, technically, a substitution substitutes a variable not a term like ?x.

Refering to the Action Algabra definition in section 2.1, the token "?x" is indeed a variable (an input variable).

P6 L-15 It would be nice to have a sentence motivating synchronisation vectors that they are essentially transitions; here the example from 2.3 was helpful.

We have added 2 sentences and a reference, it should help understanding.

P6 Def 4 When is a pNet of the right sort?

> The sorts of pNets and Holes are sets of action signatures, so compatibility is simply set inclusion. We have modified Definition 3.

P9 Def 5 Here, is $Sort_j$ again a set of parameterised actions?

> No, Sorts are sets of action signatures. There was a bug here, we have corrected the sort compatibility as "Sort(β_j) $\in Sort_j$"

P9 Def 5 L6 *all variables in the different terms β_j and α* What do you mean by this phrase?

> We rewrote this in a more formal way.

P9 Def 5 L7 By *assignments* do you mean indexed substitutions?

> We rewrote this in a more formal way.

P9 Def 6 L-4 Please explain *simple logic* and *paper rules* in more detail?

> We rephrased quite a lot this part in particular we write now:
>
> It is important to understand the difference between the red dotted rule and a normal inference rule. They correspond to two different logical levels. On one side, classical (black) inference rules use an expressive logic (like any other computer science article). On the other side, open transition rules (with dotted lines) are logical implications, but using a logic with a specific syntax and that can be mechanized (this logic includes the boolean expressions $\mathbb{B}$, boolean operators, and term equality).

P9 Def 6 L8 *We take in this article ...* Doesn't add anything and it isn't clear what the writer is trying to convey here? I guess that the intent is to stay that the set of transition is closed under the implication given below in Page 9.

Also, it would be good to add some accompanying texts that explain this implication. My understanding is that this implication says that the set of open transitions is closed under the refinement of predicates, which is a strong assumption from the modelling point of view. This is because modeller is forced to add new behaviour which (s)he is not interested to capture.

> We slightly rewrote this part and we now explicitly say that in practice, in the tools, we have a finite representation of the transitions and thus we only interact with the user based on this finite representation (see the paragraph between Theorem 1 and Theorem 2). In any case, the written closure, in general, has no finite representation.
>
> From the point of view of the article, proofs are simpler to write if we suppose that the set of open transitions is closed under the refinement of predicates.

Lastly, I was anticipating a more formal way to derive transitions of an open automaton just like how transitions are derived by a witnessing proof in the context of transition system specifications. For latter, a reference is as follows: J.F. Groote. Transition system specifications with negative premises. TCS 118, 1993 $https://www.sciencedirect.com/science/article/pii/$0304397593901116 Also, I believe that the induced transitions in T are all meaningful (in the sense of Glabbeek, see below) because there are no negative transitions in the premise of an open transition. R.J. van Glabbeek. The meaning of negative premises in transition system specifications II. JLAP, 2004 $https://www.sciencedirect.com/science/article/pii/S$1567832604000281 Such a discussion with a possible related work is missing and should be provided at least in a separate remark.

> We explain now the relation with GSOS and negative premisses in the related works

P10 Def 7 It was frustrating to not find out what is the type of an FH-bisimulation R, which I think is necessary if you want to compute such an R. Firstly, it should be mentioned that $Pred_{s,t}$ is some predicate over $V_1 \cup V_2$. Second, please give the formal type of conditional relation R before defining the transfer property of an FH-bisimulation. Here, by a conditional relation R on the sets X and Y I mean a function of type $X \times Y - -- > L$, where L is some lattice modelling the values that the relation can take. For instance, traditional relation on X and Y can be seen as a function $X \times Y - -- > 2$, where 2=0,1 is the obvious Boolean algebra of two point set; so $xRy \iff R(x, y) = 1$. Similarly, in your case, the lattice L should be the Boolean algebra P(Pred), where P(X) is the powerset of X, for any set X. This raises the issue whether FH-bisimulation is an instance of conditional bisimulation (when the lattice L is P(Pred)) as defined in the following paper: H.Beohar, B. König, S. Küpper, A. Silva. Conditional transition systems with upgrades. $https : //www.sciencedirect.com/science/article/abs/pii/S0167642319301169$

> We added the formal definition of R before the definition of FH-bisimulation. Basically, in our case L are the boolean expressions, denoted B, and corresponding to the set of all possible predicates. We believe the powerset is bigger that what we do, and we do not allow a pair of states to be matched to several Predicates. In one word the relation is now typed (with side constraints), see the details in the text.
>
> We also briefly compare to the cited article in the related work section.

P11 L1 What is X in the open transition $OT_x^{x \in X}$?

> In the context of this article, we consider that an indexed set $OT_x^{x \in X}$ defines both an indexation set X and a set of elements OT_x for $x \in$. We additionally clarified this in the notation section. Additionally to the previous text that stated: "$t_i^{i \in I}$ defines first I the set over which the family is indexed, and then a_i the elements of the family" we added: "Note that, sentences of the form "there exists $t_i^{i \in I}$" means there exists I and a function that maps each element of I to a term t_i."

P11 L2 Typo: jx as the subscript of
beta. You could use β_{j_x}?

> We should have standardly written $\beta_{j,\,x}$ but the multiplicity of commas would make the article difficult to read. We add the following footnote: "In this article, we denote β_{jx} a double indexed set, instead of the classical $\beta_{j,\,x}$. Indeed the standard notation would be too heavy in our case."

P11 L-8 What do you mean by a finite predicate?

> This was a mistake steaming from an earlier version. We removed "the predicate" from this sentence.

P12 L6 Could you expand on what is bisimulation theory for open pNets?

> This was quite a loose formulation. We have rewritten this as "the characterization and properties of a bisimulation relation".

P13 Def 9 Please exemplify *I_1 for the others*?

> We made this more explicit: I1 refers to the set of sub-nets that are not pLTS.

P13 Def 9 In the first line of the premise of Tr2, shouldn't the occurrences of α' in SV_k just be α?

> No, we have the constraint $\alpha = \alpha'$, but as actions are parameterised it corresponds to some instantiation and unification. For example in the simple protocol we would have $\alpha = in(m)$ and $\alpha' = in(MSG)$ with MSG a real message object, obtained by the r-send transition of the receiver automaton.

P13 Def 9 Tr2 by $*fresh(\alpha'_m, \alpha', \beta_j, \alpha)*$ should be understand that the variables occurring in $\alpha'_m, \alpha', \beta_j$ (for all j), and α are fresh?

It is written just above the definition

P16 L-4 You claim Theorem 5 is quite useful in practice; however, I wonder here about FH-simulation. First, whether a similar result holds for simulation? Second, FH-simulation would be more relevant if we disallow that the set of transitions in an open automaton are closed under the refinement of predicates (Def. 6).

See above: simulation is more complex and is discussed in the conclusion.

P17 Def 11 Isn't (1) a special case of (2)?

No because $T = \emptyset$ verifies (2) but not (1). We added an explanatory sentence below.

P18 Def 12 Other than the symbol used for weak transition, is there any semantic difference in Def 12 between a weak open transition and an open transition defined earlier?

> "γ ranges over sequences of action terms while "β ranges over action terms" This is now recalled before the definition. This has a huge semantic impact as a single weak transition is the composition of several transitions of the holes.