Ion Petre and Andrei Paun

PC co-chairs of CMSB 2022

June 12, 2022

Dear Ion Petre and Andrei Paun,

Subject: Submission of revised paper 13

Thank you for your email dated June 10, 2022 enclosing the reviewers’ comments. We have carefully reviewed the comments and have revised the manuscript accordingly. Our responses are given in a point-to-point manner below.

We hope the revised version is now suitable for publication and look forward to hearing from you.

Sincerely,

Van-Giang Trinh and Sylvain Soliman

**Response to Reviewer 2:**

We would like to thank the reviewer for careful and thorough reading of this manuscript and for the thoughtful comments and constructive suggestions, which help to improve the quality of this manuscript. Our responses are given below (the reviewer’s comments are in ***italics***).

**Major comments:**

*1) The revision must at least improve:*

*- the comparison with the state of the art, especially with respect to the characterization of minimal trap spaces of https://doi.org/10.1038/s41467-020-18112-5 in terms of complexity, and with ASP encoding used by mpbn based on the DNF representation of the activation condition only for locally-monotone BNs (https://doi.org/10.1109/ICTAI.2019.00014): your encoding adds the converse part which allows addressing non-monotone BNs.*

*- include mpbn in the benchmark*

**Reply:**

*2) In substance, the presented result seems more related to the fact that it is not necessary to have prime implications/minimal DNF representation of activation and inactivation conditions for identifying trap spaces, but that non-minimal DNF representations are sufficient. The link with Petri nets/concurrency is thus not apparent.*

**Reply:**

*3) It would have been more insightful to establish a link between siphons of (safe?) Petri nets in general and their attractors, following the parallel between trap spaces and attractors of BNs. In the case of safe Petri nets, would place complements allow generalizing the result beyond the specific encoding?*

**Reply:**

*4) Another line would be to bring insight from a complexity point of view, but the complexity of deciding the siphon property is not provided. Moreover, it seems that one of the main costs is actually in the translation of a BN into Petri net.*

**Reply:**

**Minor comments:**

*1) the definition of Boolean network updates with vi(t+1) is not clear/formal enough - in its current form it resembles the parallel update mode as there is no distinction between the state of M and the possible updates.*

**Reply:**

*2) trap spaces are defined over the Boolean function f, and do not depend on the update mode. It would have been much simpler (and more accurate) to define them without the notion of reachability/transition (trap spaces are subspaces closed by f).*

**Reply:**

*3) page 3, last paragraph: there is no need to capitalize the names of the logical operators*

**Reply:**

**Response to Reviewer 3:**

We would like to thank the reviewer for careful and thorough reading of this manuscript and for the thoughtful comments and constructive suggestions, which help to improve the quality of this manuscript. Our responses are given below (the reviewer’s comments are in ***italics***).

**Minor comments:**

*1) p5, ’a set of places that once empty remains empty.’*

*-> ‘empty place’ hasn’t been introduced and might be mistaken (by an uninitiated reader) with an empty set;*

**Reply:**

Indeed, ‘empty place’ has not been introduced. We have changed ‘empty’ by the equivalent term ‘unmarked’ because the term ‘marked’ has been already introduced in Section 2.2.

*2) Peterson, 1981 did not discuss siphons (nor traps), but already included (bio)chemical reactions into his modelling examples, see Chapter 3. The modelling of chemical reactions by Petri nets is also shown in Murata, 1989, meaning quite some time before Reddy, 1993.*

**Reply:**

*3) For a broader perspective on the application of trap/siphon analysis in the given context of system biology (in the wider sense), I suggest to check out:*

*+ Zevedei-Oancea, Schuster, Topological Analysis of Metabolic Networks Based on Petri Net Theory, In Silico Biology 3 (2003) 323–345 323 IOS Press*

*+ Blätke et al., BioModel Engineering with Petri Nets, https://doi.org/10.1016/B978-0-12-801213-0.00007-1*

**Reply:**

*4) The efficient computation of siphons/traps has a rather long tradition in the Petri net community. An approach exploiting the power of SAT solvers had been proposed and realised in:*

*Oanea et al, New Algorithms for Deciding the Siphon-Trap Property, LNCS 6128, Springer 2010.*

*It would be interesting to read a brief comparison with your ASP approach, also whether you could extend your method to deal with siphons/traps or the related siphon/trap property. I realise that this might be (sort of) included in reference 32; if yes, please indicate so.*

**Reply:**

*5) Please be more explicit about the relation between Boolean models and logical models. You speak about Boolean models in Section 2.1, and then about Petri net encoding of Logical models in Section 2.2; a reader might be puzzled by this sudden change.*

**Reply:**

As suggested by the reviewer, we have adjusted Section 2.2. Specifically, we only speak about the Petri encoding of Boolean models because our study is focused on Boolean models.

*6) The tool name Trappist should be introduced in the text, and the caption for Table 1/2 adjusted correspondingly.*

**Reply:**

As suggested by the reviewer, we have introduced the tool name Trappist in the first paragraph of the Evaluation section. We have also adjusted the captions of Tables 1 and 2 correspondingly.

*7) I suggest to add row numbers in Table 1/2, which then would also provide the precise figures for the number of models considered.*

**Reply:**

As suggested by the reviewer, we have added row numbers in Tables 1 and 2.

*8) 'However, the current method already appears to already perform very well'*

*-> delete one 'already'*

**Reply:**

The correction has been made.

*9) Thanks to the performance improvement coming with Trappist, we now do know for sure that logical models may come with an overwhelming number of trap spaces, often exceeding 1000, which leaves the intriguing question how to evaluate these 1000 or more (even with Trappist so far not always computable) attractors/phenotypes?*

**Reply:**