The list of response to PhD thesis reviewers.
Thesis Title: Formal Verification of systems with unbounded agents

Author: Tephilla Prince
Roll No: CS18DP001

1 Reviewer One Comments

1.1 Major Comments

1. In a few places, notions and notations are used before being introduced. It would be good to do a careful check throughout the thesis. For instance APS is used on page 18 as an example but is not defined till... 39 and even there it is rather unclear. It is only really defined in pg 79 (sec 6.1.1), but many properties about it are used from pg 18 onwards. This case-study should be introduced much earlier as it is referred to often.

   Answer: We have moved the case study to Sec. 2.2.3.

2. The prelims chapter needs a careful going over. There are many notational and other significant errors. For instance weighted Petri nets are used but only unweighted are formally defined. Buchi automata and acceptance are never defined but used in many places.

   Answer: Chapter 1: We give the definition of Buchi automata in Definition. 1.1.2. We describe omega-regular languages before defining Buchi automata.

   Chapter 2: We formally define the weighted Petri net along with Definition. 2.1.1.

3. In Chapter 4 algorithm 1 and 2 are not described except as pseudocode. Novelty in Algo 2 and notations must be made clear. Also some proof of correctness is needed. It may be good to possibly formulate a theorem saying that Algo2 is sound but not complete. Also, it seems that the Unfolding of a net for doing BMC is not defined formally.

(a) Answer: Suitable description for Algorithms has been provided in Sec. 4.1.1.

(b) It is well known that BMC is a technique that is incomplete therefore we believe that a separate proof would not add any novel contribution.

(c) Unfolding the net for BMC with respect to the various logics has been defined in Section 5.2. Note that the notion of unfolding here, is different from that of Ken McMillan's notion of unfolding which has been mentioned in the beginning of Sec.4.1.

4. In Chapter 5, in the experiments, I had two main questions: (i) why does your approach have false positives? I expected DCModelChecker to be sound but not complete, hence I am surprised to see 9% false positives which show lack of soundness. Did you examine these examples? Can you identify their source and fix it? (ii) From Expt 2/3 it seems DCModelChecker2.0 is not much better than DCModelChecker1.0... Especially in Table 5.3 it seems 1.0 is much better than 2.0. In Expt4 in sec 5.4.4, there is a statement saying "our experiments demonstrate 2.0 is better than 1.0" but these experiments don't seem to be presented in the thesis. Can you add these and comment on them, given this shows novelty and effectiveness of the 2D-BMC algorithm?

   (a) Answer: Firstly, after the time of submission of the thesis, we have rectified some of the false positives. The false positives which were rectified were due to mistakes in the handcrafted translation of the properties, for instance, the standard format for the properties from the benchmarks is in XML, which needed to be converted into a different format that DCModelChecker can use. Hence, these errors had crept in, and they were not due to an error in the algorithm. At the time of writing, there are 7 false positives in LTLFireability Experiment 1, which is 2%.

   (b) With respect to Experiment 2, we see from the plots that DCModelChecker2.0 obtains more counterexample traces than DCModelChecker1.0. We observe that DCModelChecker2.0 is better in the sense of falsification of properties (not in terms of execution times), which can be seen from the plots such as the CircadianClock-PT-000001 model, where in property 4, DCModelChecker1.0 (at bound 5 and 10) says unsat, whereas, DCModelChecker2.0 (at bound 10) gives a satisfying counterexample trace. This trend is observed in other models as well. We now explain this in detail in Sec. 5.4.2. I agree that in Table 5.3,DCModelChecker1.0 takes less time to evaluate than DCModelChecker2.0 and both of the tools have a consensus on the same answer.

5. In Chapter 6, the experimental results of DCModelChecker3.0 seem to be missing. I agree there is nothing to compare with, but it would be good to

present the results in a table/plot. I am a bit confused actually whether this tool is available or not since it is listed in Relatedwork section as well, but section 6.2 talks about tool and implementation. If it has been implemented some results should be presented, else it should be made clear that the contribution is the bounded semantics and the SMT encoding of the semantics but implementation is part of future work.

Answer: The experimental results have been added now (See Sec. 6.2.2). At the time of submission, while this table was not present in the thesis, our model checking tool was already available as an open source software in the public domain.

6. In Chapter 7, the results are nice but some of the proofs of theorems are referred to a conference paper. These should be perhaps included here. Also in the proofs it would be good to highlight where conservativeness is used to obtain decidability. Currently this seems a bit mysterious.

   Answer: The proofs for the theorems (namely Theorem 1,2 and 3) are already present in the thesis. These theorems cite the PNSE 2024 paper, to show their source, since they are a repetition of the proofs and theorems from the conference paper. With respect to the proofs for Theorems 1, 2 and 3, the style of writing is such that the proofs follow from the previous corollaries. For instance, we say in Chapter 7 Sec 7.4, in the paragraph following Corollary 2, "since the $(\leq_f, 0)$-coverability for conservative EOS is decidable, we obtain the following Theorem 1" and refer to Theorem 1.

1.2 Minor Comments

1. glossary is empty

   Answer: This has been rectified.

   Chapter 1

2. pg3: Buchi automata are not defined, accept states or condition not present in def 1.1.1

   Answer: A new definition Definition. 1.1.2 has been added to resolve this.

3. pg4: fig1.1 equivalent BA to what? Omega words not defined

   Answer: The caption has been fixed. We describe ω-word now.

4. pg6,7- contributions are out of order? 5 should come earlier than 2-4?

   Answer: As suggested, the contributions have been reordered.

   Chapter 2

5. pg11: where do you use conservativeness in entire thesis? If not needed it can be removed.

Answer: The notion of conservativeness is used in Chapter 7.

6. Pg12: fig 2.3: the picture does not match the caption in particular the marking.

   Answer: This has been rectified.

7. Pg13: fig 2.4: same issues

   Answer: This has been rectified.

8. pg14: SMT solvers are used without any introduction.

   Answer: This has been addressed in the beginning of the Sec. 2.2.

9. Pg14: you seem to be using weighted petri nets in many places throughout the thesis (even in sec2.2) but definition in 2.1 and semantics also are not weighted.

   Answer: We have addressed this in Definition. 2.1.1.

10. Pg17: unbounded Petri nets are used but boundedness and unbounded was not clearly defined earlier?

    Answer: Unbounded Petri Net is defined in Definition. 2.1.8.

11. Pg17: formal definition of unfolding is not present. This is also slightly confusing since in petri nets unfolding has a slightly different meaning from unfoldings for BMC...

    Answer: We discuss the Petri net semantics in Sec. 2.1.1 which shows how the Petri net is unfolded. Additionally, in Sec. 2.2, we now disambiguate the notion of unfolding. In this thesis, by unfolding, we refer to the process of obtaining the subsequent configurations from the initial marking of the net. This is not to be confused with the notion of unfolding described by Ken McMillan in his seminal work on partial orders.

Chapter 3

12. pg 29: Buchi automata are not defined.

    Answer: Definition. 1.1.2 has been added.

13. Pg30: Kripke structure should be defined and also an examples of syntax and semantics will help.

    Answer: Definition. 3.1.1 for Kripke structures is now added.

14. Pg34: what is a live agent? The fact that agents and clients are same needs to be written earlier.

    Answer: This has been addressed in Sec.3.2 under subsection Unbounded semantics.

15. Pg 38-41: since APS is not defined fully yet, many of these parts don't make clear sense.

    Answer: This has been rectified by introducing the case study earlier as suggested. See Sec. 2.2.3.

    Chapter 4

16. pg 47: model cannot be a subpoint of verify? Maybe write both?

    Answer: This has been rectified by posing two separate questions.

17. Pg 50: unbounded net not defined in prelim?

    Answer: This has been rectified. See Definition. 2.1.8.

18. Pg 52: state space explosion defined after its used.

    Answer: State space explosion has now been explained in the Introduction Section of Chapter 1.

    Chapter 5

19. pg58: 2D-BMC encoding needs to be defined clearly and explained better. Maybe an example will help?

    Answer: The $2D$-BMC encoding is already a part of Sec. 5.2. We have suitably changed the title to reflect this. The $2D$-BMC encoding of nets and LTL_{LIA} properties are difficult to read, hence the example of this is available in the ReadMe file of the associated tool.

20. Pg63: DCModelchecker1.0 properties should perhaps be written earlier and emphasized. PNML, ANTLR etc are used before they are defined.

    - DCModelChecker 1.0 and DCModelChecker 2.0 perform $2D$-BMC of Petri nets using the logic LTL_{LIA}. The key difference between the two tools is in the semantics used for the nets. DCModelChecker 1.0 can verify properties with interleaving semantics. We incrementally extended DCModelChecker 1.0 by adding support for true concurrent semantics to the tool, to obtain DCModelChecker 2.0 which supports both.
    - We describe PNML and ANTLR in the beginning of Sec. 5.3.1 where we give their expansion. Later on describe their usage in Sec. 5.3.3.

21. Pg77: what tapaal used earlier?

    Answer: The reasoning for choosing the tools for comparison are as follows. In Experiment 1 (Sec. 5.4.1) where we compute the tool confidence as well as verify LTLFireability properties, we compare against only ITS-Tools since (at the time of experimentation) ITS-Tools was the Model Checking

contest winner in the LTL Fireability category. Since the purpose of this experiment was to evaluate the tool confidence, it would not make any difference to compare against more than one tool, apart from the winning tool. In Experiment 3 (Sec. 5.4.3) where we verify unbounded Petri nets which is not a category in the contest, we compared our tools against both ITS-Tools and Tapaal, since Tapaal is faster than ITS-Tools and we give a comparative perspective of our tool's performance. This discussion is now added to the Conclusion Sec. 5.4.5.

Chapter 6

22. pg 79: example of APS is coming too late?

    Answer: The case study has been introduced to Sec. 2.2.3.

23. Pg 86: only few points in the semantics are described. If the rest are routine/similar and hence left out it should be made clear. Else they should be added.

    Answer: We now give the complete semantics and selectively explain the various statements.

24. Pg 87: what is UCSTL? Not defined earlier. What is MFOTL? If it is important it should be defined earlier, and not in related work. Proof of Lemma 1 details should be written as this is a thesis; not just a sketch.

    - UCSTL has been removed. It is called FOTL_1 in the rest of the thesis.
    - We describe MFOTL in Sec. 3.3.
    - We give the unbounded semantics for $FOTL_1$. We refer to Lemma 1 in Armin Biere's 2003 paper titled Bounded model checking which says that if a formula is satisfied in a bounded run, it is satisfied in an unbounded run. And we state the Lemma for $FOTL_1$.

Chapter 7

25. pg91: standard Pns are defined earlier... if you are not changing the notation, can just refer to that.

    Answer: From Chapter 1- 6, we use the standard Petri nets Definition. 2.1.1. However, in Chapter 7, we use the markings as multisets, which is clarified in Sec. 7.2.3.

26. Pg95: can recall conservativeness and explain where it is needed maybe? Or how it is used in [65]

    - We describe conservativeness of Elementary Object Systems in Sec. 7.2.4.
    - We use the property of conservativity of EOSs to obtain the decidability results in this chapter. This has been added in the end of Sec. 7.2.4.

27. pg 114: why switch from SMT to ASP? May add some justification.

    Answer: In this thesis, we mention two different encodings to PN, using SMT as well as ASP. In Sec. 7.11.3, we mention that the ASP encoding in Telingo is elegant and natural to specify PN behaviour.

2 Reviewer Two Comments

Clarifications/Amendments (if any)

The writing needs considerable improvement in my honest opinion. The following are chapterwise suggestions for amendments:

Chapter-1:

1. Page 3, Para 1: Mention the other ways of addressing the state-space explosion problem such as state abstraction.

   Answer: In chapter 1, the introduction has now been modified to include various other techniques such as abstraction, partial order reduction, cegar, compositional verification, on the fly verification with relevant references.

2. Fig 1.3: Try to place the figure close to the text where it is referred.

   Answer: Fig 1.3 is present in the next section. The figure 1.3 refers to bounded model checking, it is placed close to Example 1.1.5 where it is referenced.

3. Page 3, Section 1.1: A definition of Buchi automaton is required before Defn 1.1.2.

   Answer: Definition 1.1.3. has been added in Chapter 1.

4. Why do you introduce the BMC problem on Buchi Automaton as the model? Why not introduce Petri nets right away?

   Answer: We follow the style of Armin Biere et al in the seminal 2003 paper titled "Bounded model checking" from which we have drawn inspiration of our encodings. Therefore, the explanation using the similar style seems helpful for our understanding to draw parallels between the existing literature and ours.

5. Page 4: Para 1, Line 1: G set is undefined here.

   Answer: This is fixed by a new formal definition of Buchi automata that we have now written (See Definition 1.1.2.). Earlier, we had an informal definition.

6. Page 5, para 1: Honestly, I did not understand the example. What is the "bound" here? Because you mention "üsing the principle of bounded model checking".

   Answer: We have addressed the notion of the bound, in Sec.4.1.1. The same is repeated below: In the BMC approach, the number of steps upto which the system is verified, say k, indirectly places a bound on the number of tokens which are present in the net, thereby restricting the number of

clients that are represented in the unbounded client-server system. For instance, when unrolling the system for k steps in the BMC approach, it indirectly places a cap that there can be atmost k newly generated tokens at a place (taking into consideration the weight of the arc and the initial marking). It is to be noted that the bound in BMC is not fixed apriori, which differentiates it from the parametric verification approach where the parameter (the number of clients) is fixed apriori.

7. Page 6, Para 1, Line 5: Generally, requests are made by the clients and responded by the server. It is written the opposite.

   Answer: This has been fixed. The client requests are responded to by the server.

Chapter 2

8. Page 10, last line: The PN of Figure 2.1 doesn't show the transition weights. The PN after firing t_1 is the same PN of Fig 2.1? Please check.

   Answer: The figures have been redrawn. On firing transition t_1, in the PN in Fig. 2.1, we obtain the PN in Fig. 2.2.

9. Page 12, Fig 2.3: Please specify the F(p,t) and F(t,p) values, without which it is hard to follow the example. P_3 should not have a token. Either the marking in the caption is incorrect or the figure.

   Answer: The marking in the caption of the figure has been rectified. Place P_3 does not have a token.

10. Fig 2.4: Need a better/clear figure here. The markings in P_3 is not clear.

    Answer: This corresponds to Fig. 2.4. We say that "in concurrent semantics, the following sequence is also additionally possible in a single step, resulting in Fig. 2.4".

11. Figure 2.5: Can transitions that share tokens like t_1, t_2, fire concurrently? What if firing of one disables the other? Please clarify.

    Answer: We address this in the discussion following Definition. 2.1.12. The condition $\sum_{t \in \tau'} F(p,t) - M(p) \geq 0$ takes care of the sufficiency of tokens at each of the pre-places of the transitions in τ' such that the transitions can be fired. This is a particularly elegant rule in the context of representing concurrent firing of transitions in a Petri net in a SMT solver. This automatically addresses scenarios where the firing of one transition might disable another, the resultant set of transitions that are subsequently fired are all valid transitions that are enabled and whose concurrent firing is allowed by the Petri net semantics.

12. Page 14, Sect 2.2 Para 1:

(a) An introduction to "ünfolding a Petri-net" is crucial here. An example to demonstrate unfolding with bounded depth would help the reader. This is shown much later in the thesis, but is required right here.

Answer: In Sec. 2.2, we now disambiguate the notion of unfolding. In this thesis, by unfolding, we refer to the process of obtaining the subsequent configurations from the initial marking of the net (Petri net semantics are discussed in Sec. 2.1.1). This is not to be confused with the notion of unfolding described by Ken McMillan in his seminal work on partial orders.

(b) Unfolding a PN with true concurrency will need smaller bound to cover markings. What is the relation between bound K of analysis in interleaving semantics vs the bound of analysis in concurrent semantics? A discussion is necessary.

Answer: We do not differentiate between the bound of interleaving semantics versus concurrent semantics. Now, we discuss the difference in the executions using these two semantics, in a remark in Sec. 2.1.1. Given a PN where transitions $t_1, t_2, \cdots t_n$ are enabled and there are sufficient tokens in the pre places of $t_1, t_2, \cdots t_n$ such that the transitions may be fired, in the context of true concurrent semantics, we wish to fire maximal number of transitions, wherever possible. Consequently, this means that if there are intermediate markings, which are reachable by an interleaved firing of a subset of the transitions $t_1, t_2, \cdots t_n$, then those markings become unreachable when executing in a truly concurrent manner.

13. Page 18: What is the definition of an unbounded PN?

    Answer: This has been newly added. See Definition. 2.1.8.

14. Fig 2.6: What is APS? It is said much later in the thesis. Say it here.

    Answer: As suggested by the reviewers, we move the APS case study to Sec. 2.2.3.

15. Page 19, Defn 2.2.1: What is v?

    Answer: Now, in the Sec. 2.2.1, before Definition. 2.2.1, we say the following: "To handle the movement of tokens inside the ν-net, we employ a finite set of variables, Var using which we label the arcs of the net. There is a special variable $\nu \in Var$, which introduces tokens into the place, which is described later.". Additionally, in the text following Definition. 2.2.1, we explicitly say the following: "For instance, in Fig. 2.6, $Var = \{\nu, s, c\}$ ".

16. Page 20, Para 1, last line: I do not see the mode of the transition in Fig 2.6, as claimed!

Answer: As already given in Definition. 2.2.3, "A mode of a transition t is a mapping $\sigma : Var(t) \rightarrow Id$, instantiating every variable in the adjacent arcs of t to some identifier. " For instance, in Fig. 2.6, the arc from place p_{PR} to t_{rej} is labeled by the variable c, hence only client tokens can move along that arc. The arc from place p_{SR} to t_{acc} is labeled by the variable s hence only the server token, i.e., 0 can move along that arc. There is exactly one arc labeled with ν from transiton t_src to place p_{PR} which introduces new tokens.

Chapter 3:

17. Page 33, Sect 3.2: The state names do not match with the states of Fig 2.6.

    Answer: Now, a new Figure. 3.5 has been given to address this issue.

Chapter 4:

18. Sect 4.1: Say what is the distinction between Algo1 and Algo2.

    Answer: In Sec. 4.1.1, a discussion on the BMC algorithms has now been added.

Chapter 5:

19. Sect 5.4: What is MCC?

    Answer: Model Checking Contest. This is now present in the list of abbreviations at the beginning of the thesis.

20. Page 71: Is ITS a BMC tool? What might be the cause of the discrepancy between ITS tools and DCModelchecker? Why is the tool confidence not 1? Need more discussion.

    Answer: ITS-Tools is not a BMC tool.

    ITS-Tools is a symbolic model checker containing the tools its-reach, its-ctl and its-ltl. Among these, the its-reach tool can perform bounded depth exploration of the state space. However, ITS-Tools does not perform BMC of LTL properties (See Sec 2.2 in the TACAS 2015 paper titled Symbolic Model-Checking Using ITS-Tools authored by Yann Thierry-Mieg and private correspondence with the author at the end of this document).

    For some of the properties, we have identified that the false positives were due to errors in translating the handcrafted properties. This is one of the limitations of our tools and we aim to improve it in future work.

21. Page 72, Para 3: "The execution times of DCMC 2.0 and DCMC 1.0 are comparable". Why so? Given that DCM 1.0 one works with true concurrent semantics, should it not be faster? Need further discussion here.

Answer: DCModelChecker1.0 works with interleaving semantics. DCModelChecker2.0 also supports concurrent semantics whereve possible. We currently do not have an optimal encoding. We suspect that DCModelChecker2.0 takes more time to construct the huge number of constraints with each step of the unfolding. However, we need more detailed experiements to be able to say this. As part of future work, we would like to have an optimal encoding of the PN with true concurrent semantics, such that the two can be compared more clearly. This has been written as part of future work (See Sec. 5.4.5).

With respect to Experiment 2, we see from the plots that DCModelChecker2.0 obtains more counterexample traces than DCModelChecker1.0. And they both have similar execution times. We observe that DCModelChecker2.0 is better in the sense of falsification of properties (not in terms of execution times), which can be seen from the plots such as the CircadianClock-PT-000001 model, where in property 4, DCModelChecker1.0 (at bound 5 and 10) says unsat, whereas, DCModelChecker2.0 (at bound 10) gives a satisfying counterexample trace. This trend is observed in other models as well. We now explain this in detail in Sec. 5.4.2.

22. Page 77: DCMC1.0 and DCMC2.0 are BMC Tools. How does evaluation of unbounded PNs in BMC tools going to be meaningful/significant? Verification of unbounded client server systems with bounded model checking reads like an oxymoron. Please clarify.

    Answer: Now, in Sec. 4.1.1, we disambiguate the notion of bounds. In the BMC approach, the number of steps upto which the system is verified, say k, indirectly places a bound on the number of tokens which are present in the net, thereby restricting the number of clients that are represented in the unbounded client-server system. For instance, when unrolling the system for k steps in the BMC approach, it indirectly places a cap that there can be atmost k newly generated tokens at a place (taking into consideration the weight of the arc and the initial marking). It is to be noted that the bound in BMC is not fixed apriori, which differentiates it from the parametric verification approach where the parameter (the number of clients) is fixed apriori.

23. In Table 5.2 experiments, what was the bound used?

    Answer: The bound was 5. This is mentioned in Sec. 5.4.3.

Chapter 6

24. Page 80, Fig 2.6: The figure number should be corrected.

    Answer: The APS case study Fig. 2.6 is originally introduced in Chapter 2. Since it is being *recalled* in Chapter 6, the figure number is that of the first occurrence of the figure. To avoid confusion, we now we modify

the caption of the recalled figure in Chapter 6, as follows: "Recall the restricted ν-net modeling an unbounded client-server system".

25. Page 81, Para 2: These states are not shown in the v-net of FIgure 2.6. Similarly, t_{cp} transition is missing. This is disappointing given that the example of this figure serves as the running example.

    Answer: This has now been rectified. The case study is now present in Sec. 2.2.3.

26. Page 82: The examples are already described in Page 38. Please avoid repetition.

    Answer: This has now been removed.

27. Page 88: There is no discussion on the experimental results of the tool on running MC instances with $FOTL_1$ properties. This must be added!!

    Answer: Now, we have newly introduced Sec. 6.2.2, where we have a table of experimental data. To the best of our knowledge, there are no known benchmarks for $FOTL_1$ and ν-nets, hence we create our own ν-net model and properties in $FOTL_1$ for our experiments. Here, we refer to our open source tool, which was publicly available at the time of thesis submission also.

28. I suggest that please add a limitations section to clarify the shortcomings of the tool, if any.

    Answer: We now add a discussion of the shortcomings of the tool in the newly introduced Sec. 6.2.3.

Tephilla Prince <183061002@iitdh.ac.in>

[Research] BMC+LTL+SMT Tool

Tephilla Prince <183061002@iitdh.ac.in> Thu, Apr 7, 2022 at 3:42 PM
To: Ramchandra Phawade <prb@iitdh.ac.in>, Sheerazuddin S <sheeraz@nitc.ac.in>

Dear Sir

Please find the forwarded email from Prof.Yann Thierry-Mieg,LIP6 (highlighted).

Based on searching online and Prof.Yann's reply, it seems like there are no BMC tools that have preceded ours.

Thanks
Tephilla

---------- Forwarded message ---------
From: Yann Thierry-Mieg <yann.thierry-mieg@lip6.fr>
Date: Thu, Apr 7, 2022 at 3:21 PM

To: Tephilla Prince <183061002@iitdh.ac.in>

Hi,

>tools to perform bounded model checking for petri nets using linear temporal logic, queried using SMT/SAT.

I'm not sure what you mean,

* LTL tools yes,

* BMC tools usually are for reachability properties rather than LTL, I guess one could try to encode using BMC the problem of reachability inside the product with an automaton, but I don't know of a tool that attempts that

* queried with SAT/SMT is usually the case when talking about BMC, that is what most ppl have in mind.

Currently ITS-tools does use a layer of SMT when solving LTL, we try to prove simpler assertions we call "knowledge" and combine them to simplify or even prove the property. But I would not call that BMC.

Best regards

ytm

==
= Yann Thierry-Mieg
= Maître de conférences, HDR
= Labo. d'Informatique de Paris 6 (LIP6)
= Equipe MoVe: Modelling and Verification
= Sorbonne Université
= 4 Place Jussieu
= 25/26 Bureau 210
= tel: +33-(0)1-44-27-71-04
==

Le 2022-04-07 07:48, Tephilla Prince a écrit :
> Dear Prof.Yann
>
> I am a student researcher from IIITDh, India.
> We had interacted earlier regarding ITS-Tools.
>
> I would like to know, if you are aware of any tools to perform bounded model checking for petri nets using linear temporal logic, queried using SMT/SAT.

>
> Thank you very much for your time.
>
> Regards
> Tephilla