Reviewer #1

I suggest to describe the overall concept of the learning approach first (see above for my hopefully correct summary) and to give the technical details later, rather than describing the technical details of compilation only without providing the overall picture. Going top down would make reading much easier. The second reason for revision is the comparison to ARMS. I suggest to compare with LOUGA that is better than ARMS in all aspects (runtime and quality of obtained models) so ARMS is no more state-of-the-art.

Page 3, Section 2.1: The definition of effects is strange. The authors define positive and negative effects as literals. What if a negated predicate (which is a literal) is among positive effects? Does it mean the same as having the positive version of the predicate among negative effects? That is very confusing.

Also, there are additional assumptions about the model, namely positive effects cannot be among preconditions while negative effects must be among preconditions. It would be useful to discuss these assumptions more as there are domain models that do not satisfy these assumptions. Do we really need them?

Page 4: There is a nice discussion about partial traces, but I have found it incomplete because the relation between actions and state partial observability is not fully clear. For example, if we have FO for actions, does it mean FO or PO* for states? Also PO* for states looks like a special case of PO based on the definition. I would suggest to include some discussion there.

Page 5, Table 2: I suggest including LOUGA there too.

Page 10, Section 4.2: on(v1,v1) looks a bit strange. Do we really need such predicate with repeated variables?

Page 11, Section 4.3: Maybe this section fits better in the section Background. Do you use the assumption about having negative effects among preconditions and not having positive effects among preconditions like before? This is not clear from the definition of triggered.

Page 11, Section 4.4. This section is hard to understand, at least at the beginning. One needs to read it completely before understanding the details. I suggest to describe the concept at high level first and going to details then.

Page 11, Section 4.4. This section is hard to understand, at least at the beginning. One needs to read it completely before understanding the details. I suggest to describe the concept at high level first and going to details then.

Page 12: As I understand, the programming actions are building the model by adding preconditions and effects to the model. This immediately brings the question about symmetries as these programming actions can be used in any order (any permutation of them is fine). It would be useful to discuss this now. There are a few words about using SAT-based planner later, where parallel actions can be used. I think that this is critical for efficiency of the approach. Note also that the proposed method looks like a generate-and-test approach - we first generate a candidate model and then we verify that the model complies with plan traces. This does not seem particularly efficient.

Page 13: There is implication used in the precondition. It was was never mentioned before that a general formula can be used.

My major problem here is that it is not clear how the actions validating states interleave with actions applying the learned actions. This must be clarified as the states are generated by actions.

Page 20: In the graphs, it is not clear how the values are obtained. Are these average values between the domains or mean values or is it just for one selected domain (which?). The graph with runtimes show what I was afraid about - the efficiency degrades fast with larger example plans. This raises the question if longer plans can be used if needed for some domains.

Reviewer #2

Reviewer #3