Discuss Paper

Paper# 3008: Specifying Non-Markovian Rewards in MDPs Using LDL on Finite Traces

Abstract

In Markov Decision Processes (MDPs), the reward obtained in a state depends on the properties of the last state and action. This state dependency makes it difficult to reward more interesting long-term behaviors, such as always closing a door after it has been opened, or providing coffee only following a request. Extending MDPs to handle such non-Markovian reward function was the subject of two previous lines of work, both using variants of LTL to specify the reward function and then compiling the new model back into a Markovian model. Building upon recent progress in the theories of temporal logics over finite traces, we adopt LDLf for specifying non-Markovian rewards and provide an elegant automata construction for building a Markovian model, which extends that of previous work and offers strong minimality and compositionality guarantees.

Paper Type Full Paper

Keywords [Knowledge Representation, Reasoning, and Logic] Action, Change and Causality, [Planning and Scheduling] Markov Decisions Processes, [Planning and Scheduling] Theoretical Foundations of Planning, [Uncertainty in AI] Markov Decision Processes

Average Rating 5.48

Submission File

Assigned Area Chairs Sheila McIlraith (#30960) (University of Toronto) assigned by Carles Sierra

Assigned Senior PC M... Dongmo Zhang (#31764) (Western Sydney University) assigned by Carles Sierra

Assigned PC Members Sasha Rubin (#21282) (University of Naples, Federico II) assigned by Carles Sierra

Catalin Dima (#21748) (LACL, Université Paris-Est Créteil) assigned by Carles Sierra

Florent Teichteil-Koenigsbuch (#25784) (Airbus Group Innovations) assigned by Carles Sierra

Assigned Review Assi... Benoît Barbot (#32486) (LACL, Université Paris Est Creteil) assigned by Catalin Dima

Reviews

★★★★★★★★☆☆☆

Review from PC Member Sasha Rubin (#21282) (Created: 2017-03-31 13:57:36, Last modified: 1 week ago)

Originality

★★★★★★★☆☆☆☆

Technical Quality

★★★★★★★★★☆☆

Significance

★★★★★★★★★☆☆

Relevance

★★★★★★★★★☆☆

Quality of writing

★★★★★★★★★☆☆

Overall Score

★★★★★★★★☆☆☆

Confidence on your assessment

★★★★★★★★☆☆☆

Comments to Authors.

This paper shows how to handle MDP by reward functions that depend on the history of a trace, not just the current state and last action.

The main idea is to use LDL_f as a specification language for rewards, i.e., the reward structure is given as a set of tuples (phi_i,r_i) of pairs of LDL_f formulas and rationals. The reward at a given history is the sum of the r_j's such that phi_j holds on the given history.

A similar reward structure appeared in [Bacchus et al., 1996] where the PLTL (past LTL) was used instead of LDL_f.

The advantages of using LDL_f over PLTL are that it is more expressive.

Originality
The main original idea is to use LDL_f rather than some other temporal logic to specify rewards. However, the stated justification for using LDL_f is that "Future logics are more commonly used in the model checking community, as they are considered more natural for expressing desirable properties." I am not wholly convinced this is a good

argument since, in the present framework, the reward at a given point in time only depends on the past. For instance, in program verification, "every request is eventually granted" looks to the future when applied to infinite strings, but in the current framework it is only applied to the prefixes of an infinite string, i.e., only to the past.

The paper borrows heavily from previous works, i.e., the idea of using the reward structure (phi,r) is from [Bacchus et al., 1996], the translations of formula to automata are from [De Giacomo and Vardi, 2013, 2015], and the idea of forming an extended MDP with these automata is from [Bacchus et al., 1996].

That said, the work is certainly a novel extension of existing ideas.

Technical Quality
The translations are sound. However, some of the statements need tightening, e.g.,
- In Section 5, what does it mean for an MDP to be minimal? In Theorem 5 it looks like only the DFA extension of the MDP is minimal. Also, the first sentence after the proof of Theorem 5 seems to contradict the first paragraph of the proof of Theorem 5.
- The notion of "blind minimality" is not clear. Is there a technical result associated with this notion? e.g., on-the-fly verification seems to correspond to "blind minimality", and in the case of model checking LTL one gets a PSPACE algorithm using on-the-fly algorithms, rather than EXPTIME.

Significance
The paper shows that the choice of LDL_f has a number of good properties:
- LDL_f is very expressive (much more than LTL_f or PLTL),
- there is an easy compilation of an MDP with LDL_f rewards into an extended MDP.
- this translation is compositional in the sense that adding (or indeed removing) a reward formula does not require redoing the whole translation, and one can use on-the-fly methods and minimisation techniques from automata to build canonical and minimal extensions of the MDP.

In comparison with previous work, either the logic is more expressive [Bacchus et. al., 1996], or the logic is more natural [Thiebaux et. al., 2006].

Relevance
This will be of interest to the planning and formal-methods communities at IJCAI. My feeling is that the simplicity of the techniques and expressiveness of the logic will open up new avenues of research, as suggested in the conclusion.

The related work is very clearly discussed.

Quality of writing
Very well written with only a few issues:
- The discussion on pg 1/2 about minimality needs tightening. e.g., in "Unlike classical dynamic programming methods that require the entire state space a- priori, these algorithms generate reachable states only.", what do the classical techniques do?
- After theorem 2: "and then possibly"... why possibly?
- Section 3: you should specify the set of atoms used in the formulas of the reward function, i.e., S.
- Definition 1: "feasible trajectory" is not defined
- Theorem 5: "being A_s a DFA, such two states"-->"A_s, being a DFA, two such states"
- paragraph about "blind minimality" on pg 5: M vs M'; also, the last paragraph before section 6 starts by saying "we can progress the extended MDP by ..." and ends by giving *another* way to "progress the extended MDP"?
- Section 7: "In addition, since the automata construction algo- rithm is based on progression, unlike [Bacchus et al., 1996], we can use information about the initial state to prevent the generation of unreachable states." I do not understand this.
- pg 6, col 1, par -1: gets a rewards --> gets a reward

Comments after rebuttal

I am satisfied with the responses. I suggest the notions of minimality and consequences for this work should be carefully and explicitly discussed.

Rebuttal #4774 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)

Rebuttal

Our main motivation in adopting LDLf is its increased expressiveness wrt both LTLf and PLTL. Wrt preferring future over past in the specification of rewards, nothing conclusive can be said. We just observe that often, one conceives the property by looking at the future. The example of the reviewer itself is stated in English by looking at the future.
A notable case in which looking at the future is important is when we give procedural constraints expressed as regular expressions. Indeed programs are intrinsically expressed in the future direction.

The minimality in Theorem 5 concerns the extended MDP. It claims essentially minimality of the COMBINATION of the DFA's and the original domain (S,A,Tr). We will clarify this better.

There is no contradiction in the sentence after Theorem 5. There, we simply stress that while, IN GENERAL, the synchronous product of minimal DFA's may not be minimal, in our case, because final states must be kept distinct,

the product is indeed minimal.

The formal definition of blind minimality is in Thiebaux et al. As the reviewer observes, this is related to the possibility of computing the cartesian product of the DFA and the domain on-the-fly. We will discuss this link in more details.

We construct the automaton using progression, taking into account the initial state, and therefore, even when imitating Bacchus et al., we generate fewer states. Obviously, the minimization eventually generates the same automaton.

★★★★★★☆☆☆☆

Review from PC Member  Florent Teichteil-Koenigsbuch (#25784)

(Created: 2017-04-05 21:14:28, Last modified: 2 days, 14 hours ago)

Originality

★★★★★★★★☆☆

Technical Quality

★★★★★★☆☆☆☆

Significance

★★★★★★★★☆☆

Relevance

★★★★★★★★★☆

Quality of writing

★★★★★★☆☆☆☆

Overall Score

★★★★★★☆☆☆☆

Confidence on your assessment

★★★★★★☆☆☆☆

Comments to Authors.

--- COMMENTS AFTER REBUTTAL ---

I thank the authors for their insightful rebuttals and clarifications.

In the rebuttal the authors defend the lack of experimental evaluations of their framework by arguing that past works on Non-Markovian Rewards Decision Processes are also purely theoretical. I don't find this argument very convincing: it's not because others failed to demonstrate the practical feasibility of their approach that the authors of the paper under review should also do so.

Personally I cannot fully support a contribution that claims to improve decision makers' life (by providing a more expressive framework) without experimentally verifying that it is reasonably practical.

On the other hand I like the proposed approach, for both the more expressive grammar and the equivalent MDP construction which I find quite appealing.

Moreover, the authors could be interested in reading papers about Path-Constrained Markov Decision Processes which consist in maximizing expected cumulated rewards under a set of PCTL constraints that can be viewed as temporally extended goals. This line of research is not equivalent to the approach studied in the paper under review, but still connected. Paper to read/cite: *"Saturated Path-Constrained MDP: Planning under Uncertainty and Deterministic Model-Checking Constraints"* (AAAI 2014) by Sprauel, Kolobov and Teichteil.

This paper proposes a new fresh view on Non-Markov Reward Decision Processes based on an expressive dynamic temporal logic on finite traces named LDL_f. The idea of expressing non-Markovian rewards with temporal logics is not new but using modern intuitive logics like LDL_f, which allows problem designers to use regular expression-like formulas for instance, definitely helps modeling complex NMRDP problems more easily. On the technical side, the paper is quite involved and honestly not always easy to follow for non experts of latest temporal logic research (more on this below), but it succeeds in convincing the reader about three important computational merits of the approach: 1/ the generation of the extended MDP (product of the original MDP with the DFAs) is incremental in the sense that DFA variables are just additional variables of the original MDP's state variables; 2/ the extended MDP is proven to be minimal, i.e. it does not contain equivalent (redundant) states; 3/ the product MDP can be generated on-the-fly from the initial product state, thus exploring only reachable states like Thiebaux et al's approach. The paper makes a also good job at clearly positioning the proposition relatively to existing research on the topic.

My main concerns are about the perhaps too high technicality of some parts of the paper (lack of pedagogy because of cumulated effects of paper length constraint and complexity level of the topics addressed in the paper), and about the lack of experiments that could have help evaluate the practical feasibility of the approach in addition to providing real pedagogical examples.

In the background section, the explanations on the transformation from a LDL_f formula to a NFA is quite indigestible, perhaps too heavy for being summarized in a single column. On the other hand, it is a known result from recent literature. Thus I wonder why the authors don't simply refer to De Giacomo et al's paper on this transformation instead of polluting the present paper with very technical material which is nearly non understandable by non experts of LDL_f.

No experimental results are provided. The authors say that their approach can enjoy an on-the-fly construction of reachable states of the product MDP (as Thiebaux et al) but do not demonstrate how effective it is to do so in their framework. Moreover, they claim that in practice, because minimal DFAs are usually small, entirely generating minimal DFAs first is as efficient as directly generating the reachable product MDP embedding DFA states on-the-fly, but no results (neither theoretical nor experimental) are provided to support this claim.

Other comments:

- Page 2, section 1: "The aim of this paper is to bring to bear developments" -> remove "to bring"?

- Page 3, column 1: the third example properties uses multiple semicolon-separated formulas inside mathcal(L) whereas it was only provided a definition for 2 such semicolon-separated formulas.

- Page 3, column 2: what is a quoted subformula?

- Page 4, column 2, below definition 1: in the definition of the minimal DFA, Q should be replaced with Q_i (definition of delta_i)

- Page 4, column 2, below definition 1: you may want to say that the minimal DFA is constructed using De Giacomo et al's algorithm (without necessarily detailing nor explaining this algorithm...)

Rebuttal #4784 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)

Rebuttal

This is a theoretical paper, on the border between KR and planning. It suggests a way of expressing non-Markovian rewards and of solving it. The expressive power is stronger than that of past work, and the extended MDP is characterized much more concisely and in a principled manner. We note that both cited conference papers were also strictly theoretical.

LDLf NFA construction. We certainly could drop it, and refer to [De Giacomo and Vardi, 2015], if the reviewers ask so. It felt to us as if the paper is incomplete without this key part of the transformation algorithm.

The empirical claim on the small size of minimal DFAs comes from model checking: while the worst-case complexity of automata generation is doubly

exponential, it is often the case that the automata required for a formula are much smaller. We certainly cannot make any empirical or theoretical claim regarding the relative efficiency of incremental vs. complete construction. If this is implied by our text, we will clarify. What we hypothesize is that given the above observation, the added advantage of on-the-fly generation, may not be significant. But in any case, the choice of on-the-fly vs. complete construction of the extended MDP is closely linked to the solution method you want to use. Our point is that with our specification language and techniques work with both methods. Note also that the construction of extended MDP for the minimal DFA's and the domain can be done on-the-fly in any case.

Assessment from Sasha Rubin (#7291) (Created: 1 day, 15 hours ago, Last modified: 1 day, 15 hours ago)

Review assessment. Only visible to Area Chairs.

★ ★ ★ ★ ★ ★ ☆ ☆ ☆ ☆

[Optional] Assessment comments. Only visible to Area Chairs.

★ ★ ★ ★ ★ ☆ ☆ ☆ ☆ ☆

Review from Senior PC Member Dongmo Zhang (#31764)

(Created: 2017-03-27 00:43:39, Last modified: 1 week, 5 days ago)

Originality

★ ★ ★ ★ ★ ☆ ☆ ☆ ☆ ☆

Technical Quality

★ ★ ★ ★ ★ ★ ☆ ☆ ☆ ☆

Significance

★ ★ ★ ★ ☆ ☆ ☆ ☆ ☆ ☆

Relevance

★ ★ ★ ★ ★ ★ ★ ★ ★ ☆

Quality of writing

★ ★ ★ ★ ★ ★ ☆ ☆ ☆ ☆

Overall Score

★ ★ ★ ★ ★ ☆ ☆ ☆ ☆ ☆

Confidence on your assessment

★★★★★★★★☆☆☆

Comments to Authors.

This paper aims to use linear dynamic logic (LDL) on finite traces as a formal description language for specifying non-Markovian rewards in MDPs. A non-Markovian reward requires referring to states other than the current state therefore temporal modalities can help. Early studies have used LTL or its variations. This paper aims to use the more expressive temporal language LDL to specify non-Markovian rewards in MDPs. The paper is well organized and quite readable.

I have a few questions regarding this work:

1. It is always a tradeoff between expressive power and computational efficiency. By using LDL, you gain more expressive power but lose computational efficiency.

2. It is unclear to me whether the extra expressive power of LDL would help us to address the typical issues of non-Markovian rewards. Do we have to use LDL instead of LTL to express something like “closing a door after it has been opened”. Note that we should not view the whole statement as a reward condition. The condition to reward action “closing the door” is “it has been opened”, which can bee easily described in LTL.

3. You said that “We generate a minimal equivalent extended MDP, exploiting existing techniques for constructing automata that track the satisfiability of an LDLf formula. This construction is relatively simple and compositional.” Does “simple” here mean more intuitive or mean simpler than the case in LTL? I don’t think the latter can happen but if it is the former, it seems to me that constructing an automaton tracking the satisfiability of an LTLf formula should be at a similar level of “simplicity” as the one for LDLf formulas if not simpler.

4. Section 6 is too informal comparing to the other part. I expect to see more complexity results here to support the proposed idea as I mentioned in question 1.

In general, the paper is well organized and the idea is well explained. Although most of the results are extensions from existing work, the proposed idea is new to me.

Rebuttal #4815 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)

Rebuttal

1. In terms of worst-case complexity, LTLf and LDLf are identical. Of course, it could be that in practice, LDLf generated automata are more difficult to

minimize, but we don't know.

2. As an example, suppose you want to reward a trace if two non-mutually-exclusive properties, say a and b (eg with b=true), alternate. This is easily expressed in LDLf as: <(ab)*>End. Instead, there is no translation in LTLf. Eg, the LTLf formula Box((a -> Next b) and (b -> Next a)) fails in capturing such a property. Consider the trace {a,b}{b}, this is satisfied by the LDLf formula but not by the LTLf one, as the first occurrence of b is not followed by an a. More generally in addition to LTLf properties, LDLf formulas can capture procedural constraints, expressed as regular expression (which are not expressible in LTLf), see the discussion in [De Giacomo and Vardi, 2015].

3. We don't claim that the LDLf automata construction is simpler than LTLf's. In fact, they are very similar algorithmically. So the "simple" refers to the conceptual difficulty of the algorithm, both for LTLf and LDLf.

4. Section 6 observes how to handle the case of rewards for complete traces only. This is an important semantic change, but requires little to handle. So it is not clear to us what complexity results the reviewer would like to see. In any case, we don't have better complexity bounds for complete traces. Also the transformation works for both LTLf and LDLf.

Assessment from Sasha Rubin (#7291) (Created: 1 day, 15 hours ago, Last modified: 1 day, 15 hours ago)

Review assessment. Only visible to Area Chairs.

★ ★ ★ ☆ ☆ ☆ ☆ ☆ ☆ ☆

[Optional] Assessment comments. Only visible to Area Chairs.

★ ★ ★ ★ ☆ ☆ ☆ ☆ ☆ ☆

Review from Review Assistant Benoît Barbot (#32486) (Created: 2017-03-25 13:36:35, Last modified: 1 week, 6 days ago)

Originality

★ ★ ★ ☆ ☆ ☆ ☆ ☆ ☆ ☆

Technical Quality

★ ★ ★ ★ ★ ★ ★ ★ ☆ ☆

Significance

★ ★ ★ ☆ ☆ ☆ ☆ ☆ ☆ ☆

Relevance

★★★★★★☆☆☆☆

Quality of writing

★★★★★★★☆☆☆

Overall Score

★★★★☆☆☆☆☆☆

Confidence on your assessment

★★★★★★★☆☆☆

Comments to Authors.

This paper proposed a construction to evaluate reward in LDL on an MDP.

Given an MDP and as set of LDL formula the authors build a set of finite state automaton using algorithm LDLf 2NFA from [De Giacomo and Vardi, 2015].

Then an MDP is constructed by taking synchronised product over action of the initial MDP and all the automaton.

Computing rewards on the resulting MDP is proven to be equivalent to computing rewards specify in LDL on the initial MDP.

The paper is easy to read and is technically sound. However the contribution

is very slim. Model checking a model with a formula by taking the synchronised product with an automaton representing the formula is the base of most model-checking tool, this paper apply this pattern to MDP and LDL formula and uses published techniques to make it work.

Rebuttal #4816 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)

Rebuttal

This paper is using a combination of existing techniques (on which it is quite clear) to address (what we consider to be) an important problem for MDPs. Its contribution is not in the set of model-checking tools it uses, but in providing (as we believe, and we hope you agree) a better way of addressing this problem than the two previous methods. As such, we believe the two relevant evaluation questions are: 1. Is the problem important enough? 2. Does the solution improve upon existing approaches?

Assessment from Sasha Rubin (#7291) (Created: 1 day, 15 hours ago, Last modified: 1 day, 15 hours ago)

Review assessment. Only visible to Area Chairs.

★★☆☆☆☆☆☆☆☆

[Optional] Assessment comments. Only visible to Area Chairs.

Comments

 Dongmo Zhang (#31764) wrote 2 days, 12 hours ago:

Dear all,

Many thanks for all your insight inputs. Unfortunately I am not an expert in MDPs and it would be too late to get another expert involved in the review. However, the discussions inside of the review panel was very fruitful. We've got plenty of valuable feedback to the authors. It seems to me that our discussion is now focusing on the significance of the work, especially if the higher expressive power of LDLf plays a significant role in NMRDPs. We still have one or two days for further discussions before we start voting.

Cheers,

Dongmo

 Florent Teichteil-Koenigsbuch (#25784) wrote 2 days, 14 hours ago:

Dear all,

I've slightly lowered my relevance score and increased my score regarding the paper's quality of writing.

I've also added post-rebuttal comments to my review.

Best,

Florent

 Florent Teichteil-Koenigsbuch (#25784) wrote 2 days, 23 hours ago:

Dear sasha, dear all,

Q: "Regarding significance, the question, it seems to me, is to what extent MDPs with rewards are suitable modeling tools as compared with MDPs with temporally extended

preferences/goals"

A: MDPs with rewards are not incompatible with MDPs with temporally extended goals. I did not mention it in my review but I've personally work on MDPs with both rewards and temporally extended goals, we named this framework "Path-Constrained Markov Decision Processes". We rather viewed temporally extended goals as a set of PCTL constraints that every reward-maximizing policy must satisfy. Useful references are:

- *"Path-Constrained Markov Decision Processes: bridging the gap between probabilistic model-checking and decision-theoretic planning"* (ECAI 2012)
- *"Saturated Path-Constrained MDP: Planning under Uncertainty and Deterministic Model-Checking Constraints"* (AAAI 2014, joint work with Jonathan Sprauel and Andrey Kolobov)

But I think that this line of work is quite different from the approach proposed in the paper under review where temporal logic constraints control "rewards trajectories" but not only "state trajectories". It is connected but there are some subtleties that make the algorithms very different in the end.

I am not at all sufficiently expert in LTL/LDL and their variants to fully appreciate the relevance of the paper on this aspect. But as an MDP researcher I was quite interested in their new approach to non-Markovian rewards modeling, although I am disappointed by the lack of experimental evidence that their approach can work in practice.

Best,

Florent

 Sasha Rubin (#21282) wrote 3 days, 1 hour ago:

Dear Sheila,

Thanks for the citations and the detailed descriptions of the body of work connecting LDL_f and related formalisms to planning. I've taken a look at the papers, and have a few remarks (A. and B.) and then a comment regarding the effect the missing citations have on novelty and significance.

A. Missing citations to work on non-deterministic planning, i.e., AAAI06, ICAPS06, AIJ09, ICAPS07, KR08, AAAI17, AAAI11, ICAPS15.

The authors should have discussed your and others' work on LTL_f, LTL-RE, FO-LTL, con-GOLOG and explained why they are proposing LDL_f (rather than these other logics) as a language for expressing non-Markovian rewards in MDPs. They only do this for LTL_f.

The authors could mention that their technique works, of course, for LTL-RE since this is just LDL_f with syntactic sugar. I'm not sure much can be said about FO-LTL and con-GOLOG since the paper under review is about a propositional setting, not a

first-order setting.

B. Missing citations to work on MDPs, i.e., AAAI15.

As you wrote, this paper is about MDPs with preferences of goals (rather than simply using a single temporally extended goal). This is related to the idea of objective-LTL, see http://dblp.org/rec/journals/amai/KupfermanPV16. The authors could discuss the differences/similarities between using objectives/preferences to using reward structures on MDPs.

To summarise, although relevant, I do not believe these omissions effect the novelty because the paper under review (as well as the papers already cited [Bacchus et al., 1996; Thi´ebaux et al., 2006]) are about MDPs with rewards. Regarding significance, the question, it seems to me, is to what extent MDPs with rewards are suitable modeling tools as compared with MDPs with temporally extended preferences/goals. Perhaps someone with a deeper knowledge than mine on MDPs can comment on this?

Sasha

 Sheila McIlraith (#30960) wrote 3 days, 6 hours ago:

Hi All,

I don't wish to argue for or against this work, but I did want to note a body of related work that you may or may not be aware of since the authors neglect to cite it, that may speak to the novelty or perhaps significance of some (not all) of the contributions. Unfortunately, most of the work was done by my students and me, which I wish were not the case in this situation.

There has been extensive work in automated planning using a finite version of LTL and doing so by translating finite LTL (the term LTL_f was only coined recently, but finite LTL has been around for a while) to automata -- NFAs and also AAs. (See Baier the Baier papers for citations of other work.) There has also been work on translating a Golog-like language with regular expressions to NFAs. LDL_f can be thought of as LTL_f extended with regular expressions, so translations for finite LTL and a regular expression language to NFAs for the purposes of planning have been published. Further, a lot of engineering went into the implementations of particularly the finite LTL translation to NFAs over and above the theory, which avoided the worst case exponential blow up of such a translation which just kills the planner. Note that all these works are implemented systems.

1) Finite LTL -> NFAs for deterministic planning where the LTL was used to represent a goal or control knowledge

These papers, together with Jorge Baier's 2009 thesis discuss the translation of finite

LTL and finite first-order LTL to NFAs, their implementation and experimentation in automated planning. (Note that others e.g., Edelkamp, Cresswell and Coddington, ... also did similar work.

Planning with First-Order Temporally Extended Goals Using Heuristic Search. Baier, J.; and McIlraith, S. AAAI-06

Planning with Temporally Extended Goals Using Heuristic Search. Baier, J.; and McIlraith, S. ICAPS 2006

Jorge Baier's thesis provides further details
Effective search techniques for non-classical planning via reformulation. Baier, J. A. Ph.D. Thesis, University of Toronto, 2010.
2) finite LTL *preferences* to NFAs for preference-based planning with deterministic actions. LTL used to represent temporally extended preferences (Baier, Bacchus, McIlraith, AIJ2009)
Preferences are different from rewards, but related in the sense that they need to be optimized. The following work used an LTL like language (no automata) for MDPs with preferences
Meilun Li, Zhikun She, Andrea Turrini, Lijun Zhang:
Preference Planning for Markov Decision Processes. AAAI 2015

4) Regular expressions -> NFAs for deterministic planning where the RE was used to represent a temporally extended goal or control knowledge
Exploiting Procedural Domain Control Knowledge in State-of-the-Art Planners. Baier, J. A.; Fritz, C.; and McIlraith, S. A. ICAPS 2007

There was a KR2008 paper by Fritz et al, (Con Golog Sin Trans) that discusses some more advanced issues in greater detail.
The above 2 translations are compositional as noted in a AAAI2008 Nectar paper (Baier et al)

4) finite and infinte LTL -> NFAs for *non*deterministic planningThis paper uses our finite LTL->NFA translation from earlier work as well as *in*finite LTL-NFA translation for nondeterministic planning. Everything is implemented with extensive experiments. This is relevant over and above the previous work on implemented and tested translations of LTL and RE to NFA since it is planning in an environment with nondeterministic actions and MDPs are of course nondeterministic. Note that the authors cite this paper in a list of many papers, at the very beginning, but no discussion of the overlap with this and previous work.

Alberto Camacho, Eleni Triantafillou, Christian J. Muise, Jorge A. Baier, Sheila A. McIlraith:
Non-Deterministic Planning with Temporally Extended Goals: LTL over Finite and Infinite Traces. AAAI 2017
An earlier version appeared at an IJCAI 2016 workshop
Alberto Camacho, Eleni Triantafillou, Christian J. Muise, Jorge A. Baier, Sheila A. McIlraith:
Non-Deterministic Planning with Temporally Extended Goals: Completing the Story for

Finite and Infinite LTL (Amended Version). KnowProS@IJCAI 2016
5) A translation of PLTL to NFAs for a planning like application was exploited in Sohrabi, Baier, M, AAAI-11
There is also work (not by us) at AAAI 2015 on Prefernce-based MDPs which uses LTL (but not regular expession
=====
We also had a workshop paper where we translated LDL_f and LTL_f combined (as a language we called LTL-RE) to AA's (Alternating Automata) -- not NFAs -- for planning). We did experiments with the LTL_f fragment but not with the LDL_f because we found the syntax of LDL_f awkward and realized that we really needed the syntactic sugar of a procedural programming language like Golog (if-then-else, while, sequence) to write anything meaningful and felt this was just taking us back to our previous 2007 work.

A Unifying Framework for Planning with LTL and Regular Expressions, Triantafillou, Baier, McIrlaith MOCHAP workshp at ICAPS 2015.

Dongmo Zhang (#31764) wrote 3 days, 12 hours ago:

Many thanks, Sasha. In this case, the better way would be to propose restrictions on the languages rather than the other way around. Any thoughts from other reviewers?

Cheers,

Dongmo

Sasha Rubin (#21282) wrote 3 days, 20 hours ago:

Dear Dongmo,

Satisfiability is PSPACE-complete for both LTLf and LDLf.

See http://dblp.org/rec/conf/ijcai/GiacomoV13

Sasha

Dongmo Zhang (#31764) wrote 3 days, 23 hours ago:

Hi Sasha,

I meant reasoning or satisfiability.

Cheers,

Dongmo

 Sasha Rubin (#21282) wrote 3 days, 23 hours ago:

Dear Dongmo:

You wrote:

I am not convinced that the computational complexity of LTLf and LDLf are identical, given that LDLf is much more expressive than LTLf. Does any one know the result?

Do you mean the cost of the translation from LTLf/LDLf formulas to NFAs?

Sasha

 Dongmo Zhang (#31764) wrote 4 days ago:

I am not convinced that the computational complexity of LTLf and LDLf are identical, given that LDLf is much more expressive than LTLf. Does any one know the result? The authors gave an example to show LDLf is more expressive than LTLf, which is known but did not show why LDLf is a better language for NMMDPs. Therefore, I am a bit doubtful of the significance of this work.

Cheers,

Dongmo

 Sasha Rubin (#21282) wrote 4 days, 11 hours ago:

Dear Florent,

I completely agree that all author claims should be backed up, and that telling us that the cited papers were theoretical is no justification for making unjustified claims about the practicality. However, as their response states, they did not intend to make empirical claims. We could simply insist that they remove unjustified claims or clearly state them as conjectures.

Moreover, IJCAI has a history of publishing theoretical papers if they are significant, and this is something I would like to maintain.

I think the main question is whether the paper is **significant** enough? This is where we have the widest spread: 8, 8, 4, 3.

Dongmo, as Senior PC, perhaps you could weigh in here?

Sasha

 Florent Teichteil-Koenigsbuch (#25784) wrote 4 days, 17 hours ago:

Dear all,

In the rebuttal the authors defend the lack of experimental evaluations of their framework by arguing that past works on Non-Markovian Rewards Decision Processes are also purely theoretical. I don't find this argument very convincing: it's not because others failed to demonstrate the practical feasibility of their approach that they should also do so.

Personally I cannot support a contribution that claims to improve decision makers' life (by providing a more expressive framework) without experimentally verifying that it is reasonably practical.

On the other hand I like the proposed approach, for both the more expressive grammar and the equivalent MDP construction. If all of you want to accept the paper, I can slightly increase my overall score, but not significantly.

Best,

Florent

 Benoît Barbot (#32486) wrote 4 days, 18 hours ago:

Dear Sasha,

My only concern was that the paper is too incremental as you have mentioned.

As other reviewers found the contribution sufficient, I agree to accept this paper.

--

Benoit

Sasha Rubin (#21282) wrote 4 days, 19 hours ago:

Dear Dongmo, Benoit:

Have your concerns been adequately addressed? Would you like to discuss them?

Sasha

Sasha Rubin (#21282) wrote 6 days, 21 hours ago:

Dear all:

What are your thoughts on the author rebuttals?

Dongmo: As demonstrated, there is no tradeoff (in computational complexity)

for the higher expressive power for using LDLf.

Benoit: I agree with you that the technical contribution is shallow.

Florent: I agree that the authors made claims about the expected behaviour of their algorithms in practice, and did not offer experiments to back this up. My feeling is that even without these claims, the paper is still worthwhile.

I would say this work is very incremental but does advance the state of the art, tackles an important problem in knowledge representation and synthesis, and is reasonably well written.

Sasha

 Sasha Rubin (#21282) wrote 1 week ago:

Dear all:

I read all the responses. I am satisfied and think this paper should be accepted.

Sasha

 Sasha Rubin (#21282) wrote 1 week, 5 days ago:

Dear all:

1. I disagree that one has to trade the added expressive power for computational efficiency. Indeed, using LDL_f in, e.g., Theorem 1, has the same computational complexity as using PLTL.

2. Reasons to use LDL_f instead of PLTL [Bacchus et. al., 1996] or $FLTL [Thi´ebaux et al., 2006] are:

a) LDL_f is more expressive than PLTL (e.g., one can use it to express "the door is only opened on even time steps"); and LDL_f is almost certainly at least as expressive as $FLTL since LDL_f can express fixpoints (since it is equivalent to MSO) which is the main ingredient that complicates $FLTL [Definition 2, Thiebaux et al., 2006]),

b) LDL_f is clearly more natural than the logic $FLTL in [Section 3.2, Thi´ebaux et al., 2006] which seems less useful since its semantics are indeed hard to understand.

Sasha

 Dongmo Zhang (#31764) wrote 1 week, 5 days ago:

Dear all,

All reviews are here now. We have a diversity of opinions on this work. Similar to Florent, my concern on this paper is also its "practical feasibility". Although a number of reasons were given for using LDLf instead of LTLf, only the reason for higher expressive

power is acceptable but it has to be traded off with computational efficiency. Any further discussions are welcome.

Cheers,

Dongmo