

ConfMaster@IJCAI 2017 printed for Sasha Rubin (sasharubin) at 2017-04-08 11:37:52

Discuss Paper

Paper# 3744: Bridging the Gap between LTL Synthesis and Automated Planning

Abstract

Linear Temporal Logic (LTL) synthesis can be understood as the problem of building a controller that defines a winning strategy, for a two-player game against the environment, where the objective is to satisfy a given LTL formula. It is an important problem with applications in software synthesis, including controller synthesis. Recent work has explored the close connection between automated planning and LTL synthesis but has not provided a full mapping between the two problems nor have its practical implications been explored. In this paper we establish the correspondence between LTL synthesis and fully observable non-deterministic (FOND) planning. We also provide the first explicit compilation that translates an LTL synthesis problem to a FOND problem. Experiments with state-of-the-art LTL FOND and synthesis solvers show automated planning to be a viable and effective tool for highly structured LTL synthesis problems.

Paper Type

Full Paper

Keywords

[Knowledge Representation, Reasoning, and Logic] Action, Change and Causality, [Planning and Scheduling] Planning Algorithms, [Agent-based and Multi-agent Systems] Formal verification, validation and synthesis, [Planning and Scheduling] Planning and Scheduling



Average Rating

3.60

Submission File





Assigned Area Chairs



 Hector Geffner  (#30160) (ICREA & Universitat Pompeu Fabra) assigned by Carles Sierra



Assigned Senior PC M...

 Giuseppe De Giacomo  (#27377) (Sapienza Università di Roma) assigned by Carles Sierra

Assigned PC Members

 AndreA Orlandini  (#20791) (National Research Council of Italy (CNR-ISTC)) assigned by Carles Sierra

 Jens Claßen  (#21006) (RWTH Aachen University) assigned by Carles Sierra

 Nicolás D'Ippolito  (#21701) (Universidad de Buenos Aires) assigned by Carles Sierra

Assigned Review Assi...



Sasha Rubin 🇮🇹 (#32466) (University of Naples, Federico II) assigned by Giuseppe De Giacomo

Reviews

★★★★★☆☆☆☆

Review from PC Member  AndreA Orlandini 🇮🇹 (#20791) (Created:



2017-04-04 15:41:58, Last modified: 3 days, 19 hours ago)

Originality

★★★★★☆☆☆☆

Technical Quality

★★★★★☆☆☆☆

Significance

★★★★★☆☆☆☆

Relevance

★★★★★☆☆☆☆

Quality of writing

★★★★★☆☆☆☆

Overall Score

★★★★★☆☆☆☆

Confidence on your assessment

★★★★★☆☆☆☆

Comments to Authors.

=== COMMENTS AFTER REBUTTAL ===

In their rebuttal, authors partially addresses my concerns. They provide a clarification and, though in extra pages, I'm confident they can provide proofs for theorems.

On the other hand (also seeing other reviews), I think the paper (as it is) requires some further additional work in order to enhance the overall quality of the paper.

I would encourage authors to keep working on this paper because I really believe they can generate an enhanced version to be considered for a future submission in any other major AI venue.

=== ORIGINAL REVIEW ===

The paper investigates the relationships between LTL Fond Planning and LTL

synthesis. A mapping from LTL synthesis problem and LTL FOND problem is proposed. The paper ends with a discussion on an assessment of solving different problem instances leveraging the encoding in order to evaluate different solving approaches.

The paper presents a nice contribution and slightly advances the state of the art. Nevertheless, several major issues affect the paper.

As stated also by the authors, this work takes inspiration from several related works and, indeed, the advancement proposed here is strongly leveraging their results. In this sense, the originality is rather low.

In general, the proposed result seem sound but the paper is missing to formally demonstrate theorems. At least a sketch of proof for Theo 1, Theo 2 and Theo 3 would be needed.

In my opinion, the significance of the proposed result seems rather low. First, the results proposed in this paper are (straightforward?) extension of already published results. Then, the evaluation analysis presents two major limitations: 1) the set of considered problems is limited and a more comprehensive analysis (more LTL synthesis problems/planning domains) should be considered in order to support significant statements. 2) the experimental results are rather obvious (both for FOND planning and LTL synthesis) and do not convey sharp results.

The paper is relevant for IJCAI even though, given the above points, the interest for the work may be minor.

The paper is generally well written. The section 4.1 is rather dense and difficult to follow. The explanation of details for the mapping is not easy and the reader may be lost. Using a running example would enhance the readability.

To summarize, the papers is technically sound and presents a nice result that strongly relies on past work and does not provide a sharp contribution. Readability could be enhanced and the evaluation of the results should be extended. For the reasons given above, I'm not in favor of the acceptance of the paper.

Rebuttal #582 (Created: 1 week, 3 days ago, Last modified: 1 week, 1 day ago)



Rebuttal



Thanks for your review. We will provide proof sketches, and enhance

readability using a running example as suggested, buying extra pages.

Significance: LTL synthesis is a classical problem that is critical to automated programming and controller synthesis, and has been a focus of recent invited talks/papers at IJCAI and ICAPS. This paper provides an important contribution because it proposes and demonstrates a promising computational approach. Ideas from planning made disruptive contributions to formal verification and we see this as a step towards similar potential impact for synthesis.

Not a Straightforward Extension: LTL synthesis requires computing infinite non-deterministic plans in unfair environments. No one has done it before and it required both a recognition of this relationship and an understanding of how to realize it. Efficiency was also a major challenge. This work developed techniques to exploit structure and relevance that were critical to exploiting FOND for synthesis. This is an important contribution of this work over the state of the art in LTL-FOND.

To run experiments we wrote a synthesis-to-FOND translator. For the other direction we wrote domain-specific translations (a limiting factor). Experiments show that planning technology is more effective than traditional methods for solving highly structured problems. This is an important message to communicate, given a recent trend towards solving FOND via synthesis (e.g. Sardina and D'Ippolito 2015).

Assessment from  Sasha Rubin  (#7291) (Created: 1 day, 14 hours ago, Last modified: 1 day, 14 hours ago)

Review assessment. Only visible to Area Chairs.

★★★★★☆☆☆☆

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

★★★★★☆☆☆☆

Review from PC Member  Jens Claßen  (#21006) (Created:

2017-04-05 13:29:45, Last modified: 2 days, 22 hours ago)

Originality

★★★★★☆☆☆☆

Technical Quality

★★★★★☆☆☆☆

Significance

★★★★★☆☆☆☆

Relevance

★★★★★★★☆☆

Quality of writing

★★★★★★★☆☆

Overall Score

★★★★★★☆☆☆☆

Confidence on your assessment

★★★★★★☆☆☆☆

Comments to Authors.

The paper investigates the relation between LTL synthesis (LTLS) on the one hand and fully-observable non-deterministic (FOND) planning on the other hand. This is done by first establishing a theoretical correspondence between LTLS realizability and the existence of strong plans for FOND problems, next by presenting a translation that allows solving LTLS by iteratively searching for strong-cyclic FOND solutions for increasing horizons, and finally by means of an experimental evaluation that shows the strengths and weaknesses of solvers for the different problem classes.

All in all, the paper is a good read. Larger parts about the different mappings and translations tend to be very technical, but this lies in the nature of the subject matter. Unfortunately, at certain points it becomes quite hard to follow the central construction of Section 4 as not all details are explained. In particular this holds for the roles of the superscripts S and T for variables $q, q^T, q^S, Q^{\{S, T\}}$, or the meaning of the i and j subscripts for $\text{trans}(T)$ actions.

Thus, especially as a non-expert in the area, I am not fully able to judge the correctness and significance of the paper's contribution. From what I can see, it appears to make a valuable contribution to establish a correspondence between the two involved areas, both from a theoretical and a practical point of view.

Minor comments:

In Section 2.3 you say "LTL executions are defined just like in FOND", does this include finite ones? LTL is only defined for infinite sequences, but finite ones can of course be extended by using "sink" states.

In Section 3, when constructing the synthesis problem given a FOND instance, you say "For space, we do not go into the details of how to encode $\xi_{\{a, e\}}$." A little more explanation than the sentence that follows would be nice, if not the full technical definition, then at least by means of an example. (A complete example for the full

construction of Section 4 would even be nicer.)

Table 1 contains cells labelled "time", which probably denotes time-out instances. This should be mentioned in the text somewhere, and it should be said what the time-out limit actually was.

Nitpicking:

p.2, column 1, l.14: "A policy π is a partial function [...]" -> remove dot after π

p.2, column 1, l.24: "[...] that is such that $s \models \phi$ " -> " $s \models \phi$ " should be " $s \models \phi$ "

p.2, last paragraph of Section 2.2: "it is possible to construct Non-deterministic Büchi Automaton" -> put "a" between "construct" and "Non-deterministic"

p.2, column 2, second paragraph of Section 2.4: Rephrase the sentence "Formally, a synthesis problem is a tuple [...]" (missing verb)

p.4, column 1, paragraph "Automaton Mode": "[...] the agent can decide which automaton transitions wants to perform [...]" -> insert "it" between "transitions" and "wants"

p.4, column 2, paragraph below Definition 1: "The non-determinism introduced by the action accept is such strong-cyclic plans that are solution to P' yield infinite plan executions [...]" -> insert "that" between "such" and "strong-cyclic"

Comments after rebuttal:

The authors' response does help clarify parts of the construction. The general criticism about Section 4 being hard to follow remains though. I second the advice by one of the other reviewers to reformulate the construction such that the formal mapping is separated from implementational details and optimizations. Also, including an illustrating example would improve understandability immensely.

Rebuttal #7151 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)



Rebuttal



Thanks for your review.

Meaning of q : A plan produces an automaton fluent q if it yields a run of the automaton finishing in q . Notably, multiple q fluents can be true in a single state, thereby capturing multiple runs of the automaton simultaneously.

Meaning of AT : The truth of fluents q^AT indicate, together with fluent q , that the runs that finish in q should be progressed into runs that visit an accepting state before the next 'accept' action is performed.

Meaning of AS : Fluents q^AS and q^AST are auxiliary copies of q and q^AT , respectively, used to progress multiple runs of the automaton via $Trans_T$ actions. $Trans_T$ for $T=(q1,q2)$ progresses $q1^AS$ into $q2$, $q1^AST$ into $q2^AT$ if $q2$ is not accepting, and sets the value of $guard(T)$ variables.

Correctness: When 'accept' is applied, one effect leads to the dummy goal. The other effect 'forgets' all runs finishing in q if q^AT holds true -- i.e. it deletes q and q^AT . Otherwise, fluents q^AT are added (see formula in the R#32466's response,, and an example in R#27377's response). These dynamics guarantee that *all* infinite executions that 'accept' infinitely often in a search tree yield automaton runs that visit accepting states infinitely often (otherwise, they will be eventually 'forgotten' by 'accept'). The agent can defer application of 'accept' as much as needed, until all necessary automaton runs visit accepting states, guaranteeing completeness. The $turn_k$ counter guarantees soundness in a (loopy) search graph.



Assessment from  Sasha Rubin  (#7291) (Created: 1 day, 14 hours ago, Last modified: 1 day, 14 hours ago)

Review assessment. Only visible to Area Chairs.

★★★★★☆☆☆☆

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

★★★★★☆☆☆☆

Review from PC Member  Nicolás D'Ippolito  (#21701) (Created: 2017-04-05 21:42:53, Last modified: 2 days, 13 hours ago)

Originality

★★★★★☆☆☆☆

Technical Quality

★★★☆☆☆☆☆☆

Significance

★★★☆☆☆☆☆☆

Relevance

★★★★★★★★★★

Quality of writing

★★★★☆ ☆ ☆ ☆ ☆ ☆

Overall Score

★★★★☆ ☆ ☆ ☆ ☆ ☆ ☆

Confidence on your assessment

★★★★★ ★ ★ ★ ☆

Comments to Authors.

The paper presents an equivalence between LTL reactive synthesis and FOND planning. The work presents a compilation from LTL reactive synthesis to FOND planning and prove the translation to be correct.

Authors evaluate their translation with state-of-the-art planners and synthesis tools.

The paper explain the approach a bit vaguely. The results, if correct, are indeed very interesting and relevant to both communities Control Synthesis and Planning. However, there is an issue with the treatment of controlled actions. The approach process nondeterministic controllable actions as monitored ones. This mistake makes the approach and Theorem 2 to be incorrect. In fact, one wonders how they have conducted their experiments successfully with such a bug in the translation. Authors have commented on this but due to the imprecise description and formalization of the translation it is very hard to assess the translation validity.

As authors point out, there has been many efforts exploring partial translations relating FOND Planning and Reactive Synthesis (RS). This is the first work that formally establishes the link between the two worlds. In particular, the RS to FOND direction, which is the more complex.

Another obscure point is how authors handle the fact that they require strong solutions but they seem to be computing strong-cyclic ones.

The translation itself is a clever variation of known tricks to force planners to produce plans that loop. Indeed, is not technically challenging, and more importantly, is incorrect.

Pros:

- The problem the paper considers is very relevant to the community.
- Evaluation seems adequate

Cons:

- The translation from LTL synthesis to FOND planning may have a bug. Authors' explanation helped but the translation remains quite obscure in some points. This reviewers could not assess the validity of the translation.

- Complexity analysis is missing from the paper. Authors' do commented on that in the rebuttal though.

-Many of the explanations, complete paragraphs even, are very similar to those of the cited paper [Camacho et al., 2017].

Minor comments:

- the explanation before Theorem 1 is not easy to follow, among other things, because it has some implicit assumptions. For instance, it is possible to reason in terms of the negation of realizability is possible only because two-player games used in reactive synthesis are determined.

Rebuttal #7161 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)



Rebuttal

Thanks for your review.

Complexity: The transformation of LTL into NBA is worst-case EXP in the size of the formula. FOND is EXPTIME-complete in the problem size. Therefore, our approach is 2EXPTIME, and “optimal” as it matches with the complexity of LTL synthesis (Vardi 1995, Kupferman and Vardi 1997).

Correctness: We believe that the concerns you raised about correctness may stem from a misunderstanding. You noted that “the approach processes nondeterministic controllable actions as monitored one.” This is not the case. The confusion may derive from the fact that in control synthesis, the first player is typically the agent, whereas in LTL synthesis the environment plays first. This is noted in Section 3, paragraph 1 “...the environment ‘plays first’”. Nevertheless we will emphasize this more strongly.

The uncontrollable non-deterministic actions in our translation are the `move_k` plays of the environment, which set the value of the uncontrollable variables (X). The agent has no control over them, but has full observability before deciding the assignment to controllable (Y) variables.

Alternatively, the confusion may have arisen from a misunderstanding that planning states capture multiple runs of the automaton. See R#27377's response for a walkthrough example, and R#21006's response for a discussion on correctness.

If this does not explain your concern then please elaborate in your final review.

Assessment from  Sasha Rubin  (#7291) (Created: 1 day, 14 hours



ago, Last modified: 1 day, 13 hours ago)

Review assessment. Only visible to Area Chairs.

★★★★★★☆☆

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

★★★★★★★★

Review from Senior PC Member  Giuseppe De Giacomo  (#27377)



(Created: 2017-04-07 16:11:33, Last modified: 19 hours ago)

Originality

★★★★★★☆☆

Technical Quality

★★★☆☆☆☆

Significance

★★★☆☆☆☆

Relevance

★★★★★★★★

Quality of writing

★★★★★☆☆

Overall Score

★★★☆☆☆☆

Confidence on your assessment

★★★★★★★★

Comments to Authors.

The authors propose a technique to reduce LTL synthesis to FOND strong cyclic planning for standard reachability goals. The result would be of maximal interest, since in doing this they avoid the notorious determinization step which has resisted good algorithms for more than 20 years.

Unfortunately the reduction is not thoroughly explained, and the authors appear to be unaware of the quantum leap that their reduction would bring to the scientific community, if correct. The construction is very badly described, with a lot of handwaving, which is unacceptable for such an important result.

In fact from the sketch in the paper the construction appears to be wrong, missing exactly the point of determinization. Indeed it seems to leave the nondeterministic choices of the NBA to the agent, which is incorrect. Let me explain the need for determinization with examples.

1. As a warmup let us consider a very simple two player game (much like a FOND domain):

Arena:

States: s1, s2, s3, s4, s5

Actions: a, b

Transitions:

s1--a->s2

s1--a->s3

s2--a->s4

s3--b->s5

Initial state: s1

Agent: controls the actions,

Env: controls the nondeterminism in the transitions.

Goal is given as the following NFA:

States n1, n2, n3, n4

Initial state n1

Final states: {n4}

Transitions:

n1--a->n2

n1--a->n3

n2--a->n4

n3--b->n4

(This NFA recognize the language {aa,ab}).

The equivalent minimal DFA is:

States d1, d2, d3

Initial state d1

Final states: {d3}

Transitions:

d1--a->d2

d2--a->d3

d2--b->n3

Now if we take the synchronous product of the Arena and the DFA and play the reachability game, where Agent chooses actions and Env chooses transitions, we get winning strategy for Agent:

$\pi(s1d1) = a$

$\pi(s2d2) = a$

$\pi(s3d2) = b$

If instead we take the synchronous product of the Arena and the NFA and play the reachability game assuming that Agent controls the transitions of the NFA we DON'T GET ANY WINNING STRATEGY for Agent.

In particular if Agent tries the following:

$\pi(s1n1) = (a, n2)$, Env can play $s3$ where action a is not available. Similarly if Agent tries the following:

$\pi(s1n1) = (a, n3)$, Env can play $s2$ where action b is not available.

What is happening is that Agent takes responsibility for the NFA TRANSITIONS, these are SEEN BY ENV, which can look for a countermove. Instead the nondeterminism should be kept free to satisfy the goal all along independently of the choices of Agent and Env. (Sometimes we say that the automaton must have perfect foresight, while strategies have none.)

The result is that the solutions of this "wrong use" of NFA nondeterminism are sound, but incomplete in general.

2. This problem with nondeterminism does show in LTL synthesis, making Lemma 2 incorrect. Here is an example:

Consider only two propositions:

X controlled by Env

Y controlled by Agent

and the NBA:

States: $n1, n2, n3, n4$

Initial state $n1$

Accepting states $\{n4\}$

Transitions:

n1 -true-> n2

n1 -true-> n3

n2 -!X&!Y-> n4

n3 -X&Y-> n4

n4 -true-> n4

This NBA recognizes $\text{next}(X=Y)$ (which by the way admits a DBA as well, for the matter). Obviously there is a winning strategy: at the second step Agent copies on Y the value of X (and continues arbitrarily).

Now lets Agent resolve nondeterminism. Then initially Agent need to choose between n2 and n3. Supposes Agent chooses n2, then Env plays X, thus blocking the execution of the NBA. Similarly if Agent chooses n3, then Env plays !X, thus blocking again the execution of the NBA. Hence Agent CANNOT FIND A WINNING STRATEGY!

I hope that these examples clarify the issue of nondeterminism, and can help in later versions of the paper, either to clarify the construction, or to aim for variants of synthesis for which this ad-hoc handling of nondeterminism is indeed complete.

Minor comments:

Page 2, col 2, first paragraph. Clarify that for LTL-FOND you are considering policies that depend on the histories not only on states.

Page 2, col 2. "play first" and "play second" in LTL is a minor technicality. (By the way, if you think the Env to set the propositions and the Agent to set the actions, the turn taking is exactly as in FOND, including the initial state as the first --fully constrained-- choice of the environment.) In any case one can easily reformulate problems to delay the response of either players at will with some book-keeping (this is often done)

Page 3, col 1. $f:(2^X)^* \rightarrow 2^Y$.

Page 3, col 1, Theorem 1. This result can be suitably reformulated to link realizability and existence (vs non-existence) of strong plan. (See comments on inessentiality of "play first" and "play second".)

Page 3, col 2, Section 4. This section needs to be written very precisely. It combines many elements some of which are essential and some of which are optimizations. It should be rewritten by first giving a pristine construction that focus on all the important aspects, such as handling nondeterminism, iterative horizon setting, going from strong to strong cyclic solutions, etc. Later, once correctness is established, optimization for performa can be considered as well.

Comment after rebuttal:

From the rebuttal I now understand, that at least for NFA (the NBA proposed for the LTL formula was indeed just NFA with the last state repeated forever) that the authors do a sort of subset construction for determinization on the fly. But as Safra and Safraless constructions show the subset construction is not sufficient to handle the infinite case. So more explanation is needed. So while I cannot say that the construction does not work, I cannot say that it works either. This is a very important result, if true, and it should be presented in a formally crisp way, possibly also relating it to the difficulties treated in literature and how they are handle by the proposed construction.

Rebuttal #7167 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)



Rebuttal

Thanks for this detailed review. We respectfully note that examples 1 and 2 are not counterexamples. This is because a single planning state can correspond to more than one automaton state, thereby capturing multiple runs of the automaton.

Example1 simulates a FOND problem with actions {a,b}. Note that a “complete” formulation would have NBA transitions labeled with preconditions (e.g, $n2 \rightarrow (s1 \vee s2) \wedge a \rightarrow n4$). We go over the original example and show a plan is found.

For readability, we obviate the turn_k fluents, summarize the env move_k actions into a single play, and start in a state S0 that simulates the initial state suggested by the reviewer.

Env vars. $X = \{s1, s2, s3, s4, s5\}$

Agent vars. $Y = \{a, b\}$

$S0 = (s1, n1 \wedge S, \text{aut_mode})$

$\text{Trans_}(n1 \rightarrow a \rightarrow n2); \text{state } S1 = (n1 \wedge S, n2, s1, a, \text{aut_mode})$

$\text{Trans_}(n1 \rightarrow a \rightarrow n3); S2 = (n1 \wedge S, n2, n3, s1, a, \text{aut_mode})$

$\text{switch2env}; S3 = (n2, n3, \text{env_mode})$

Case 1: env plays s2; $S4 = (n2, n3, s2, \text{env_mode})$

$\text{switch2aut}; S5 = (n2 \wedge S, n3 \wedge S, s2, \text{aut_mode})$

$\text{Trans_}(n2 \rightarrow a \rightarrow n4); S6 = (n2 \wedge S, n3 \wedge S, n4, s2, a, \text{aut_mode}, \text{can_accept})$
accept, non-deterministically leading to the dummy goal



Case 2: environment plays s3; $S4 = (n2, n3, s3, \text{env_mode})$

$\text{switch2aut}; S5 = (n2 \wedge S, n3 \wedge S, s3, \text{aut_mode})$

$\text{Trans_}(n3 \rightarrow b \rightarrow n4); S6 = (n2 \wedge S, n3 \wedge S, n4, s2, b, \text{aut_mode}, \text{can_accept})$
accept, non-deterministically leading to the dummy goal

In either case, a weak plan is found, which visits accepting state n4. This is possible because state S3 captures all runs of the automaton (n2, n3).

Similarly in Example2.

Assessment from  Sasha Rubin  (#7291) (Created: 1 day, 14 hours ago, Last modified: 1 day, 14 hours ago)

Review assessment. Only visible to Area Chairs.

★★★★★★★☆☆

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

★★★★☆☆☆☆

Review from Review Assistant  Sasha Rubin  (#32466) (Created: 2017-04-08 08:55:49, Last modified: 2 hours ago)

Originality

★★★★★★★☆☆

Technical Quality

★★★★☆☆☆☆

Significance

★★★★☆☆☆☆

Relevance

★★★★★★★☆☆

Quality of writing

★★★★☆☆☆☆

Overall Score

★★★★☆☆☆☆

Confidence on your assessment

★★★★★★★☆☆

Comments to Authors.

This paper is about using FOND_P planners to solve LTL synthesis.

LTL realisability (sometimes called LTL synthesis) is the problem of deciding if there exists a finite-state strategy that satisfies a given LTL property no matter how the environment behaves [Pnueli and Rosner 1989]. The idea is that the set of propositions of the formula are partitioned into those controlled by the agent and those controlled by the environment, and that each "player" can set the values of their variables (players take turns, with the environment going first). The realisability problem is known to be

2EXPTIME-compelte.

This paper suggests a translation of LTL synthesis to strong-cyclic FOND planning (Theorem 3) and compares the translation with state-of-the-art synthesis tools.

Originality

Although there are translations of synthesis into two-player graph games (which themselves are very similar to the state-based representations of FOND with temporally extended goals), this is the first translation into a compact representation of FOND with reachability goals.

Technical Quality

I have some minor concerns, and a major concern.

Minor concerns:

a) In the abstract and elsewhere it says that this paper is the first to supply a full mapping between LTL synthesis and FOND planning. The meaning of "full mapping" (also "correspondence") is not explained. A good answer would be optimal translations in both directions. However, only a translation from LTL synthesis to FOND planning is given; and no optimality is claimed. By "optimal" I mean that, e.g., synthesising Φ by reducing to FOND planning is 2EXPTIME.

b) The authors see a problem with the "inverted turns", where I see no problem. In LTL synthesis and in the game-theoretic view of FOND, it is the environment that moves first. Said another way, there is a simpler version of theorem 1 which says that " (X, Y, ϕ) is realisable iff M_ϕ has a strong plan".

My main concern is that the translation (section 4.1) is not understandable. This is for two reasons: i) the construction is not given clearly or formally enough for me to verify it is correct, and ii) there is not enough intuition to explain the construction. Moreover, from what I can gather, I do not believe the construction is correct.

Although I don't have a formal counter-argument or counter-example to the translation (the translation is not stated clearly or formally enough for me to establish this), the following fundamental issue is not properly dealt with: In order to solve LTL synthesis one typically applies a combinatorially complex construction from automata theory. The translation given in this paper has no such combinatorics, nor is it explained how it circumvents these constructions.

Here are two approaches to solving LTL synthesis. In each, I point out the "complex" step (shorthands: N = nondeterministic, B = Buchi, P = Parity, C = Co-Buchi, W = Word, T = Tree):

a) LTL --> NBW --> DPW --> DPT --> emptiness

The second step, determinisation, is "complex", e.g., use Safra's construction.

b) LTL --> UCW --> UCT --> emptiness

The third step, emptiness check, is "complex", e.g., use Muller and Schupp's simulation lemma.

Question to the authors: how does your construction deal with, or avoid, these complex combinatorial constructions?

Significance

The abstract states "In this paper we establish the correspondence between LTL synthesis and fully observable non-deterministic (FOND) planning." This is a laudible and important objective. However, the present paper falls short of meeting this objective.

Relevance

The topic will be of interest to the planning community and the formal methods communities in IJCAI.

Regarding related work:

- Little to no discussion is given on the classic ways of solving LTL synthesis, and its relationship between the submitted work.

Question to authors: what is the relationship between previous/classic ways of solving LTL synthesis and the proposed algorithm?

- pg 2: The citation [Camacho et. al. 2017] is given for FOND with LTL goals. Much more can be said, including references to FOND with LTL goals in which the domain is given explicitly
<http://dblp.org/rec/conf/ecp/GiacomoV99>.

Quality of writing The paper is poorly written. Here are some typos and vague notions:

- abstract: "two player game against the environment" --> "game with one player against the environment"
- introduction, par 2: N^3 -time algorithm ... N is undefined.
- pg 2, par 3: dfn of "s" is a result of applying a in s" should be bracketed.
- pg 2, par 3: "finite executions" should be italicised when defined

(since "finite execution" might simply mean any execution that is of finite length).

- pg 2: "satisfies the goal" --> "achieves the goal"
- pg 2: $\Box \phi \equiv \neg \Diamond * \neg \phi$
- pg 2: construct Non-deterministic --> "construct a Non-deterministic"
- pg 2, sec 2.4: the domain of f is better written $(2^X)^+$
- pg 2, sec 2.4: "Intuitively, no matter what the choice of the environment is, which is given by the sequence $X_1 X_2 \dots$, the controller has a strategy, given by f , that ensures formula ϕ is satisfied in the resulting game." --> "Intuitively, the environment has a strategy such that, no matter what sequence $X_1 X_2 \dots$ " In other words, the order of the quantifiers matters.
- pg 2, sec 3: "Rather, in FOND, the play sequence is inverted since the environment decides the outcome of an action, which is in turn defined by the agent (controller)." I disagree. The natural way to view a FOND as a 2-player game is for the environment to move first by setting all fluents in I to be true and fluents not in I to be false. that, starting from a "dummy initial" state, the environment moves to the state I .
- The phrase "2 player games" is not used carefully enough. Sometimes it seems to mean some abstract notion of game, other times it refers to games played on graphs ("game structures" pg 2).
- pg 3, par 2: $(2^X) \rightarrow (2^X)^+$
- section 4:
 - the construction is muddled with optimisations and details of implementation (e.g., the use of "spot"; pre-processing ϕ looks like an optimisation; the inclusion of "Regularize" is stated to be an optimization for action "switch2env" but not for action "accept"). This makes it hard to understand the core ideas that make the transformation correct.
 - what is the domain and range of Δ ? cf. " $T \in \Delta$ " and later " $T = \Delta(q, q')$ "
 - the terms " $\text{Pre_switch2aut}(h, h')$ " does not parse, i.e., Pre does can't take arguments (by dfn on pg 2), and switch2aut does not take parameters (by dfn on pg 3).
 - " $\text{Pre_trans}(T)$ " and " $\text{Eff_trans}(T)$ " mention q_i and q_j . I assume $T = (q_i, q_j)$.
 - why does one need v_I and $v_{\neg I}$? isn't having one of them enough? i.e., to simulate that I is true set v_I and to simulate I is false unset v_I .
 - The exact meaning of q^S , q^T and $q^{\{ST\}}$ are unclear. In particular, q^S and $q^{\{ST\}}$ are not explained.

Questions to the authors: what is the intended meaning of q^S , q^T , and $q^{\{ST\}}$?

- No good intuition about the meaning of h_{\max} is given. Perhaps this is similar to the idea that if one has a winning strategy in

a finite-duration variation of a Buchi game (for sufficiently large number of steps), then one has a winning strategy in the original Buchi game. See, e.g., journals/ijfcs/Fearnley012.

- The paragraph after definition 1: what does it mean to "reach, *recognizably*, an accepting state"?; "is such strong cyclic plans"; what is the "dummy goal"?
- I am confused by the mechanics of the translation. When I try to follow the progression of the fluents I find that they can get stuck in a way that does not match the given synthesis problem. Given an NBW for the formula, consider an infinite accepting trace $q_0 q_1 \dots$. Follow the sequence of fluents representing states of the NBW in the FOND problem. The initial set of such fluents (let's call them state-fluents) is $\{q_0\}$. After the environment sets its fluents, the action `switch2aut` replaces q_0 by q_0^S . Then a transition fires (say from q_0 to q_1) and the state-fluents are $\{q_0^S, q_1\}$. Then `switch2env` fires and the state-fluents are $\{q_1\}$. Then suppose `accept` fires and the second effect is enabled: then the state-fluents are $\{q_1^T\}$. Then after the environment sets its fluents the action `switch2aut` results in $\{q_1^{\{ST\}}\}$. At this point, no transition can fire because every transition requires a precondition of the form q^S . This results in a finite (and thus fair) execution of the FOND which does not reach the goal, while the corresponding trace in the NBW was accepting.

Question to the authors: Is this sequence of evaluations of the fluents intended?

- pg 6: the "switches" domain is discussed in the body but does not appear in Table 1.

Overall: The topic is interesting and relevant for the IJCAI community. Using planning technology to solve synthesis problems would be a fantastic addition to automatic synthesis tools (<http://www.syntcomp.org/>). However, this submission has too many typos, too little formalisation, and too little clear intuition to be sure what the authors had in mind. Moreover, my best understanding of the paper suggests that the main translation is incorrect.

Response after submission

Thank you for the response. I am very interested to see this work corrected, clearly explained in light of existing work, and published in the future. I hope you will find my comments helpful.

After your response, I gather that your translation is doing a basic

subset construction of some kind. I highly recommend that you explain precisely how one can solve LTL synthesis with such a basic subset construction. That is, do you have a new conceptual approach to solving LTL synthesis? or does your translation "implement" an existing approach? (Of course the former would be a breakthrough, although the latter would still be *very* interesting).

As I wrote in the review, I know of two broad approaches to solving LTL synthesis:

using determinisation: $LTL \rightarrow NBW \rightarrow DPW \rightarrow DPT \rightarrow \text{emptiness}$

and

using emptiness of tree automata: $LTL \rightarrow UCW \rightarrow UCT \rightarrow \text{emptiness}$

Here are some pointers to the literature that you might find useful.

DETERMINISATION

An excellent exposition of Safra's determinisation construction is in Safra's PhD thesis:

<http://www.math.tau.ac.il/~safra/PapersAndTalks/PhDthesis.ps>

Another good exposition is Lecture 26 of Theory of Computation, Kozen, 2006, Springer.

EMPTINESS OF ALTERNATING TREE AUTOMATA

Emptiness of APTs can be reduced to solving a parity game over tiles, see Chapter 9, Section 9.6 of "Automata, Logics, and Infinite Games: A guide to current research", Gradel, Thomas, Wilke (Eds), 2002, LNCS 2500.

Another way to solve emptiness of alternating tree automata is to reduce it to solving a two player game of *imperfect information* (after which one would apply a belief-space/subset construction), see <http://dblp.org/rec/conf/fsttcs/FijalkowPS13>. In the context of solving synthesis this means that the agent picks its variables, the environment picks its variables and chooses the transition of the UCT and the direction in the tree; however, the agent is not allowed to see the current state of the automaton, otherwise it is given more power than it could have (i.e., it could win the game even though the formula is not realisable).

UNIVERSALITY

As a sanity check you should make sure that your translation works for LTL universality. Indeed, universality is a special case of synthesis: just assume the agent does not control any variables.

See <http://dblp.org/rec/conf/stoc/Reif79> for the point that one needs

imperfect-information games to solve universality of automata.

REDUCING TO FINITE-DURATION GAMES

Your translation, it seems, implicitly reduces (perfect information) buchi games to a (perfect information) reachability game. This step should also be clearly explained. One way I know how to do this is as follows: play until a state has been repeated, and the agent wins if a buchi state is seen on the cycle (correctness of the reduction follows from the fact that Buchi games are memoryless determined). More sophisticated techniques (for more sophisticated winning conditions) can be found here (and in the references of this paper):

<http://dblp.org/rec/journals/ijfcs/Fearnley012>

Your translation evokes the incremental bounded-synthesis procedure of <http://dblp.org/rec/journals/sttt/FinkbeinerS13> which itself is based on the safraless procedure for emptiness of UCT <http://dblp.org/rec/conf/focs/KupfermanV05>

The idea there is, roughly, to reduce emptiness of UCT to emptiness of the UCT with the extra condition that the automaton can't visit a co-Buchi state more than k times (where k is the integer parameter).

Finally, there is very recent work that considers various encodings of bounded synthesis, http://link.springer.com/chapter/10.1007%2F978-3-662-54577-5_20

FROM STRONG TO STRONG-CYCLIC

Your translation somehow reduces strong planning to strong-cyclic planning (as suggested by the text at the start of Section 4.1). This step should be carefully and clearly explained, i.e., what is the mechanism by which one reduces 2 player games with an adversarial opponent to 2 player games with a fair opponent?

Confidential Comments (Not visible to the authors)

I updated my "comments after response" (improved accuracy, added text and references on bounded synthesis).

Rebuttal #7198 (Created: 1 week, 1 day ago, Last modified: 1 week, 1 day ago)



Rebuttal

Thanks for the detailed review. Notation is explained in R#21006's response, and complexity in R#21701's.

Compilations from FOND to synthesis were discussed in e.g., (De Giacomo & Vardi 2013). We examine the opposite direction.

Response to main concern: We will elaborate upon the construction and

underlying intuitions with extra pages. Our compilation avoids determinization of the NBA by capturing multiple -- up to all, if the agent decides -- runs of the automaton in a single planning state. More precisely, and abusing notation, fluent q is true in state s if there exists a run of the automaton finishing in q . See walkthrough example in Response to #27377, and correctness discussion in R#21006's response.

We found some minor errors (that explain R#32466's last example issues) in the translation right after submission, corrected them, reran the experiments and received improved results from which we draw the same conclusion. The issue you found is a manifestation of the error we fixed. The details:

The second effect of "accept" must update the automaton fluents as follows: $\{q \wedge \sim q^T \rightarrow q^T \text{ s.t. } q \in Q\} \cup \{q \wedge q^T \rightarrow \{\sim q^T, \sim q\} \text{ s.t. } q \notin Q_{Fin}\}$, and reestablish $turn_1$.

Finally, the part of Regularize that is optional is deleting v_l fluents.

In the dynamics of the example, the effect of "accept" does not delete q_1 , and the resulting planning state is $\{q_1, q_1^T\}$. After the environment moves, the action `switch2aut` results in $\{q_1^S, q_1^ST\}$, and so on.

Comments



Giuseppe De Giacomo 🇮🇹 (#27377) wrote 3 days, 2 hours ago:

Dear friends,

After the discussion I will propose rejection for the paper (I'll write in the meta review that the paper proposes important and ambitious techniques, but in its current form it is by far too messy to check the correctness and accept it at IJCAI) . Please revise your reviews by today, adding a new section at the end of your reviews: "Comments after rebuttal:" to explain to the authors the impact of their rebuttal on your view of the paper.

Thanks!

-Giuseppe



AndreA Orlandini 🇮🇹 (#20791) wrote 3 days, 20 hours ago:

Dear all,

In their rebuttal, authors partially addresses my concerns. I am going to revise (increasing) a little bit my scores but I think the paper (as it is) is rather weak for IJCAI.

Nevertheless, as also stated by Giuseppe, this work sounds promising. I believe that with some additional efforts it could lead to a more interesting paper.

Best,
Andrea



Nicolás D'Ippolito 🇦🇷 (#21701) wrote 3 days, 23 hours ago:

Hi all, I agree they seem to be doing the subset construction. I still have doubts about how they actually treat controlled actions, but again, the translation is not thoroughly explained.

I agree with Giuseppe the paper is not ready but it's certainly interesting. I'll introduce comments to improve the paper in my review.

best

N



Giuseppe De Giacomo 🇮🇹 (#27377) wrote 6 days, 1 hour ago:

Dear friends,

The rebuttals are in. Please have a look at them and at each other reviews and let's start the discussion. The discussion phase ends on April 5th 23:59 (UTC-12). We need to revise our reviews by then.

Now from the rebuttal of the authors, it looks like they ARE doing a subset construction (the one for determinization) to avoid the environment to know too much in reacting to the agent strategy. Though, a lot remains obscure to me:

1. How this subset construction extend it to the Safra or Safraless construction to handle LTL on (infinite traces)?
2. How to they go *strong* planning to *strong cyclic* planning, thus neutralizing the fairness assumption?
3. Is the role of the incremental construction essential or not. If it is essential why is so?
4. Also, why the authors made a big deal about planning for NOT PHI in the previous section and then they plan for PHI in their main section?

Finally the complexity should be better analyzed, considering it wrt the size of the NBA in particular.

I believe that the paper is too confused to be accepted now.

But, I also believe that they might be after something good and original. Indeed many of parts of their proposal resonate with some constructions that have been separately looked up in formal methods.

I think we should try to help them in clarify their ideas and express them suitable in next the iteration of the paper.

To do so it is important that they put their construction in relation to the crucial problem/techniques for handling LTL synthesis devised in recent years by the formal methods community. So we should give them a list of references to which compare their solution. Sasha I think you can help us a lot with this.

Looking forwards for your comments!

Cheers,

-Giuseppe



Sasha Rubin 🇮🇹 (#32466) wrote 1 week ago:

Dear all:

The response did not adequately address how the translation is supposed to overcome or avoid the usual determinisation steps. Thus, I do not understand the intended translation well enough to have much confidence that the idea is correct.

Sasha



Jens Claßen 🇩🇪 (#21006) wrote 1 week, 4 days ago:

After reading the other reviews and given the recent discussion, I further adapted my overall score and confidence. I am curious about the authors' response.



Nicolás D'Ippolito 🇦🇷 (#21701) wrote 1 week, 4 days ago:

Hi, by determinising the controllable actions I meant BEFORE the "the agent chooses and execute one action $\text{trans}(T)$ ". I'm now much more convinced the translation has problems.

I'm now lowering my score and updating my review.

best

ps. I've got a comment on the paper pointing out that the nondet on controlled events is a problem, but somehow later I've managed to convince myself otherwise. Me bad.



Nicolás D'Ippolito 🇦🇷 (#21701) wrote 1 week, 4 days ago:

Hi, by determinising the controllable actions I meant BEFORE the "the agent chooses and execute one action $\text{trans}(T)$ ". I'm now much more convinced the translation has problems.

I'm now lowering my score and updating my review.

best

ps. I've got a comment on the paper pointing out that the nondet on controlled events is a problem, but somehow later I've managed to convince myself otherwise. Me bad.



Giuseppe De Giacomo 🇮🇹 (#27377) wrote 1 week, 4 days ago:

Dear Nicolas,

My first example is only a warmup. But the second example does apply! It is a standard LTL realizability problem: in the NBA (as in LTL) you don't see the micro-states where only X' is set. (if you don't think so please explain because we need to be on the same page wrt this.)

What do you mean with "are controllable transitions being determinised?". Look at page 4, col 1 Automaton Mode, the agent chooses and execute **one** action " $\text{trans}(T)$ " among the possible ones, and in doing so the agent chooses both Y (completing the guard(T)) and next NBA state (this is the being error!!!).

Cheers,

-Giuseppe

PS. Nicolas, Jens, you may want to lower the score of your review anyway: the fact that we have doubts on how the construction works is per se unacceptable for an IJCAI paper.

PS2. We have only few hours before the reviews are given to the authors, if we need to modify them, we should act fast.



Nicolás D'Ippolito 🇦🇷 (#21701) wrote 1 week, 4 days ago:

Hi all, in general I agree that if the nondeterminism is consequence of controlled actions then the approach might not handle it correctly. I also agree that the explanations on the translation are poor, I though it was related to my lack of expertise in PDDL, though,

I'd note that in the paper they consider the Pnueli's setting where the states are only labelled with variables. Then, the game proceeds as follows: from a state XY the environment chooses which variables X' to be set true/false (leading to a "micro" state), only then the system does updates output variables Y' and the game reaches a state $X'Y'$. In such context, the games Giuseppe proposes are not valid. However, a small variation of the example may still expose a problem.

I have a question to everyone: are controllable transitions being determinised? I'm know uncertain if that's the case. If they are not determinised, I'd agree that a strong cyclic solution that assumes fairness would not satisfy the LTL when translated back to P.



Jens Claßen 🇩🇪 (#21006) wrote 1 week, 4 days ago:

Dear all,


I cannot answer Sasha's questions, and because of the given doubts in the paper's correctness, I toned down my review (and confidence score). It seems I was overly enthusiastic with my assessment of this paper after reviewing some really bad ones.

If there is actually a flaw in the construction, then this is obviously a dealbreaker. I wonder whether and how that would show in their experimental evaluation (i.e. did they verify the solutions the different solvers returned?). If it is sound, but incomplete, then of course all solutions would still be correct.

Cheers,

Jens



Giuseppe De Giacomo  (#27377) wrote 1 week, 4 days ago:


Sasha,

Can you put your questions in your review as question for the authors as well?

Thanks,

-Giuseppe




Sasha Rubin  (#32466) wrote 1 week, 5 days ago:

Dear all,

Does anyone understand if there is an implicit subset construction in the given translation? e.g., how many Q fluents (i.e., elements of Q , $Q^S = \{q^s : q \in Q\}$, Q^T , Q^{ST}) can be true at any one time? Also, what is the meaning of Q^S , Q^T , Q^{ST} ?

Sasha



Giuseppe De Giacomo  (#27377) wrote 1 week, 5 days ago:

Dear friends,

There is the strong suspect that the translation proposed is incorrect (possibly sound but not complete), since it sidesteps one of the key ingredients that makes reactive synthesis hard, namely determinization of the specs.

Please have a look at each others review. If the result i correct is an important one. If not, we should stop the paper to avoid future embarrassment.

Cheers,

-Giuseppe