FOND Planning for LTL_f and PLTL Goals

Francesco Fuggitti

YORK UNIVERSITÉ UNIVERSITY

Actions Workshop @ICAPS19
Berkeley, July 12, 2019

Joint work with Prof. Giuseppe De Giacomo

Introduction

- Classical planning with temporally extended goals starting from Bacchus and Kabanza (1998)
  - capture a richer class of finite plans
- Recently, Fully Observable Non-Deterministic planning for LTL/LTL_f with Camacho et. al (2017/18), De Giacomo et. al (2018)
- MSc Thesis: LTL and Past LTL on Finite Traces for Planning and Declarative Process Mining

What's new in this work?

- $\text{LTL}_f\text{2DFA}$: a tool for translating temporal formulas to automata
- Extended goals represented also with Past LTL (PLTL)
  - possible computational advantage wrt LTL_f (single vs double exponential time translation to automata (Chandra, Kozen, and Stockmeyer 1981))

Example

PLTL Goal: $\varphi = vehicleAt(l22) \wedge \diamondminus(vehicleAt(l31))$

- New automata compilation technique in PDDL

PLTL and LTL_f

- Linear Temporal Logic on finite traces: LTL_f
  - next: $\bigcirc happy$
  - until: $reply\,\mathcal{U}\,acknowledge$
  - eventually: $\Diamond rich$
  - always: $\Box safe$
- Past Linear Temporal Logic: PLTL
  - yesterday: $\circleddash happy$
  - since: $reply\,\mathcal{S}\,acknowledge$
  - once: $\Diamond\!\!\!-\, rich$
  - hystorically: $\boxminus safe$

Reasoning in LTL_f/PLTL

- transform a formula φ into a DFA $\mathcal{A}_\varphi$
- for every trace π, an LTL_f/PLTL formula φ is such that:

$$\pi \models \varphi \iff \pi \in \mathcal{L}(\mathcal{A}_\varphi)$$

- Currently, we don't allow mixing LTL_f and PLTL (left for future work)

YORK UNIVERSITÉ UNIVERSITY

LTL_f2DFA: from LTL_f and PLTL formulas to DFA

Translation procedure:

1. starting from an LTL_f/PLTL formula φ, we translate it to FOL on finite sequences (De Giacomo and Vardi 2013; Zhu et al. 2018)
2. apply MONA able to transform Monadic Second Order Logic (and hence FOL as well) on finite strings to **minimum** DFA automata

Example: $\varphi = \Diamond G$

1. FOL translation: $fol(\varphi, x) = \exists y. x \leq y \leq last \wedge G(y)$, where $[x/0]$
2. MONA program: `m2l-str; var2 G; ex1 y:0<=y & y<=max($) & y in G`

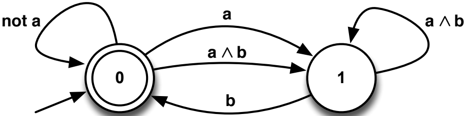
LTL_f2DFA implementation

Python package supporting:

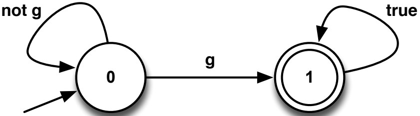
- parsing of LTL_f/PLTL formulas
- translation to FOL, DFA
- option for DECLARE assumption (De Giacomo et al. 2014)
- available as a service online at http://ltlf2dfa.diag.uniroma1.it

Examples:

$$\varphi_1 = \Box(a \Rightarrow \bigcirc b)$$



$$\varphi_2 = \Diamond g$$



FOND Planning for Extended Temporal Goals

- A *fully observable non-deterministic* (FOND) domain with initial state is a tuple $\mathcal{D} = \langle 2^{\mathcal{F}}, A, s_0, \varrho, \alpha \rangle$ where:
  - $\mathcal{F}$ is a set of *fluents* (atomic propositions);
  - A is a set of *actions* (atomic symbols);
  - $2^{\mathcal{F}}$ is the set of states;
  - s_0 is the initial state (initial assignment to fluents);
  - $\alpha(s) \subseteq A$ represents *action preconditions*;
  - $(s, a, s') \in \varrho$ with $a \in \alpha(s)$ represents *action effects* (including frame)

Goals, planning and plans

- Goal: an LTL_f or a PLTL formula φ
- Planning: a **game** between the Agent and the Environment
- Plan: **strategy** to **win** the game

Our approach:

Idea: reduce the problem to classical FOND planning

1. Transform the LTL_f/PLTL goal φ into the corresponding minimum DFA $\mathcal{A}_\varphi$ through $\text{LTL}_f\text{2DFA}$
2. Construct a PDDL domain and problem that include $\mathcal{A}_\varphi$
3. Solve the new planning problem with any off-the-shelf planner
4. Extract the policy from the solution

How to encode $\mathcal{A}_\varphi$ in PDDL?

In the Domain:

- Action "`trans`": representing the transition function of $\mathcal{A}_\varphi$, parametric on objects of interest
- Predicate "`turnDomain`": to alternate between domain actions and "`trans`"

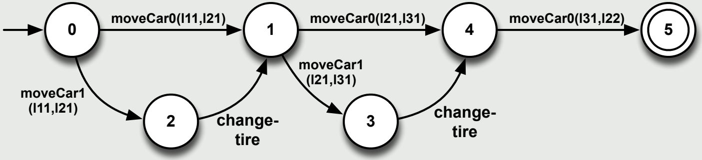
In the Problem:

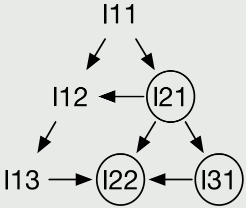
- New initial state including the initial state of the automaton
- New goal state with the final state(s) of the automaton evaluated on objects of interest

Implementation and Results

Example: the Triangle Tireworld domain

Objective: Drive from one location to another. A tire may be going flat. If there is a spare tire in the location of the car, then the car can use it to fix the flat tire.
Goal: $\varphi = vehicleAt(l22) \wedge \diamondminus(vehicleAt(l31))$
Plan *(Strong)*: any path from state 0 to state 5 achieves to the goal





Conclusions

Results:

- Provided the $\text{LTL}_f\text{2DFA}$ tool which implements the translation procedure from LTL_f/PLTL to DFA
- Proposed and implemented our approach in compiling LTL_f/PLTL goals along with the original planning domain, specified in PDDL
- Started investigating planning for PLTL goals

Future work

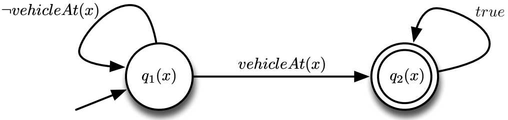
- Investigate the potential computational advantage of PLTL goals
- Study to what extent temporally extended goals can be crafted (or reformulated) to exploit PLTL and gain computational efficiency
- Investigate the LTLp_f logic (i.e. LTL_f and PLTL merged) for dealing directly with mixed formulas
- Employ different planners benchmarking major encoding techniques

Acknowledgments

- Professor Giuseppe De Giacomo
- Professor Yves Lespérance

Thanks ! Questions ?

Appendix



$$\varphi = \Diamond(vehicleAt(l13))$$

```
(:action trans
:parameters (?x - location)
:precondition (not (turnDomain))
:effect (and
(when (and (q1 ?x) (not (vehicle-at ?x))) (and (q1 ?x) (not (q2 ?x))
(turnDomain))
(when (or (and (q1 ?x) (vehicle-at ?x)) (q2 ?x)) (and (q2 ?x) (not (q1
?x)) (turnDomain))))
```

- Initial state: `and (... (q1 l13) (turnDomain))`
- Goal state: `(and (q2 l13) (turnDomain))`