Planning for LTL_f Goals in Presence of Uncertainty on the Initial State

Luciana Silo[1]

[1]Sapienza University of Rome
silo@diag.uniroma1.it

Joint work with Giuseppe De Giacomo

ERC Advanced Grant
WhiteMech:
White-box Self Programming Mechanisms
SAPIENZA
UNIVERSITÀ DI ROMA

Motivation

We are interested in **Planning for Temporally Extended Goals**.
This setting has been extensively studied in the literature, e.g.:

- use of Temporally Extended Goals on Infinite Traces [De Giacomo & Vardi 1999]
  - they handle uncertainty on initial state, but only for the infinite traces setting
- use of Temporally Extended Goals on Finite Traces
  - deterministic planning [Baier & McIlraith 2006] and non-deterministic planning [DeGiacomo & Rubin 2018]
  - they handle temporal goals on finite traces, but without uncertainty on initial state

Goal: *we extend the work of [De Giacomo, Vardi 1999] in the finite traces setting (with LTLf goals)*

Linear Temporal Logic on Finite Traces

LTL$_f$: the language

$$\varphi ::= A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2$$

- A: atomic propositions
- $\neg\varphi$, $\varphi_1 \wedge \varphi_2$: boolean connectives
- $\bigcirc\varphi$: "next step exists and at next step (of the trace) φ holds"
- $\varphi_1 \,\mathcal{U}\, \varphi_2$:"eventually φ_2 holds, and φ_1 holds until φ_2 does"
- $\bullet\varphi \doteq \neg\bigcirc\neg\varphi$: "if next step exists then at next step φ holds" *(weak next)*
- $\Diamond\varphi \doteq \mathtt{true}\,\mathcal{U}\,\varphi$: "$\varphi$ will eventually hold"
- $\Box\varphi \doteq \neg\Diamond\neg\varphi$: "from current till last instant φ will always hold"
- $Last \doteq \neg\bigcirc\mathtt{true}$: denotes last instant of trace.

Properties: expressibility (FOL over finite sequences or Star-free RE), reasoning (satisfiability, validity, entailment PSPACE-complete), model checking (linear on TS, PSPACE-complete on formula)

Non Deterministic Finite State Automata

An NFA is a tuple $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$:

- Σ is the alphabet;
- Q is a finite set of states;
- $Q_0 \subseteq Q$ is the set of initial states;
- $\delta = Q \times \Sigma \rightarrow 2^Q$ is the nondeterministic transition relation;
- $F \subseteq Q$ is the set of accepting states

Planning Domain

A planning domain is transition system $\mathcal{T} = (S, S_0, Act, R, Obs, \pi)$

- S is the finite set of possible states;
- $S_0 \subseteq S$ is the finite set of possible initial states;
- Act is the set of possible actions;
- $R : S \times Act \to S$ is the transition function that given a state and an action returns the next state;
- Obs is the finite set of possible observations
- $\pi : S \to Obs$ is the observability function

Planning for LTL_f goals: the idea

Planning Domain $\mathcal{T}$

X

NFA $\mathcal{A}_\varphi$

Single-exponential blow-up

Goal LTL_f formula φ

$A^{\varphi}_{\mathcal{T}}$

The product automaton recognizes all plans that achieves the goal in the domain

Standard Planning

Complete information on the initial state: initial state is a singleton $\{s_0\} \subseteq S$, set of observations coincides with the set of states $Obs = S$

To synthesize a plan compute NFA $\mathcal{A}_{\mathcal{T}}^{\varphi} = (Act, \mathcal{S}_T, \mathcal{S}_{T_0}, \delta_T, F_T)$:

- $\mathcal{S}_T = Q \times S$
- $\mathcal{S}_{T_0} = Q_0 \times \{s_0\}$
- $(q_j, s_j) \in \delta_T((q_i, s_i), a)$ iff $s_j = R(s_i, a)$ and $q_j \in \delta(q_i, \pi(s_i))$
- $F_T = F \times S$

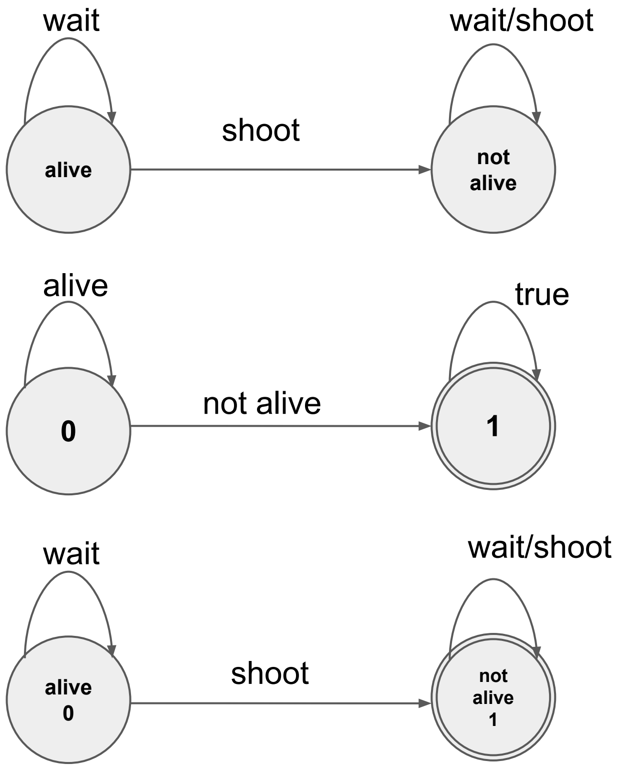
Theorem: A plan p for $\mathcal{T}$ realizing the specification $\mathcal{A}_{\varphi}$ exists iff $L(\mathcal{A}_{\mathcal{T}}^{\varphi}) \neq \emptyset$.

Standard Planning-Yale Shooting Example

Yale Shooting Domain $\mathcal{T}$

$\mathcal{A}_{\Diamond \neg alive}$

$\mathcal{A}_{\mathcal{T}}^{\Diamond \neg alive}$



Conformant Planning

Incomplete information on the initial state and no observability on the states: $S_0 = \{s_{00}, \ldots, s_{0k-1}\}$ for $k > 1$

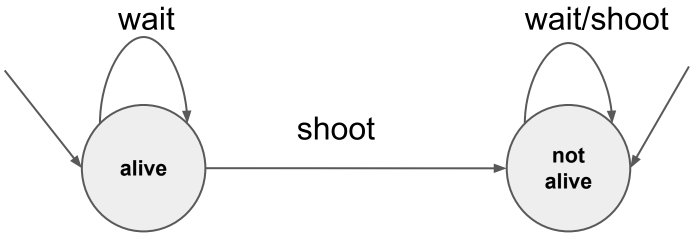
To synthesize a plan compute NFA $\mathcal{A}_{\mathcal{T}}^{\varphi} = (Act, \mathcal{S}_T, \mathcal{S}_{T_0}, \delta_T, F_T)$:

- $\mathcal{S}_T = Q^k \times S^k$
- $\mathcal{S}_{T0} = Q_0^k \times \{(s_{00}, \ldots, s_{0k-1})\}$
- $(\overrightarrow{q_j}, \overrightarrow{s_j}) \in \delta_T((\overrightarrow{q_i}, \overrightarrow{s_i}), a)$ iff $s_{jh} = R(s_{ih}, a)$ and $q_{jh} \in \delta(q_{ih}, \pi(s_{ih}))$ for $h = 0, \ldots, k{-}1$
- $F_T = F^k \times S^k$

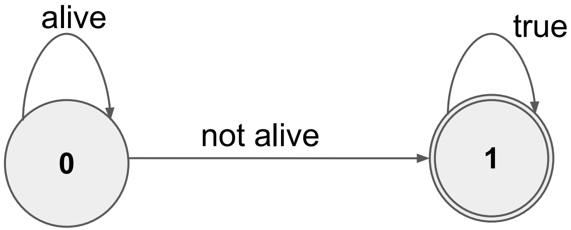
Theorem: A plan p for $\mathcal{T}$ realizing the specification $\mathcal{A}_{\varphi}$ exists iff $L(\mathcal{A}_{\mathcal{T}}^{\varphi}) \neq \emptyset$.

Conformant Planning-Yale Shooting Example

Yale Shooting Domain $\mathcal{T}$



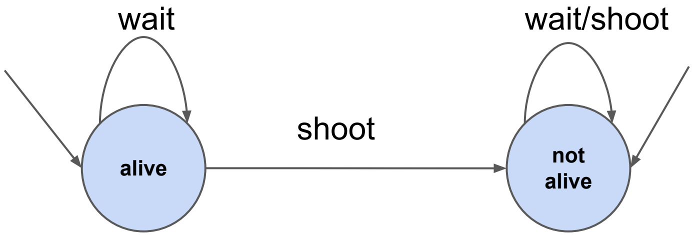
$\mathcal{A}_{\Diamond\neg alive}$



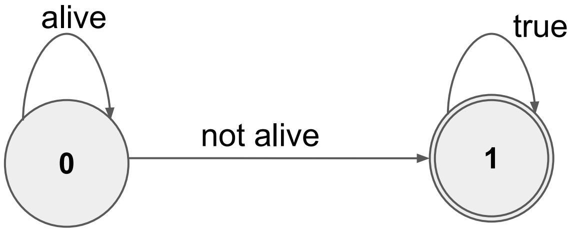
Conformant Planning-Yale Shooting Example

Possible initial states

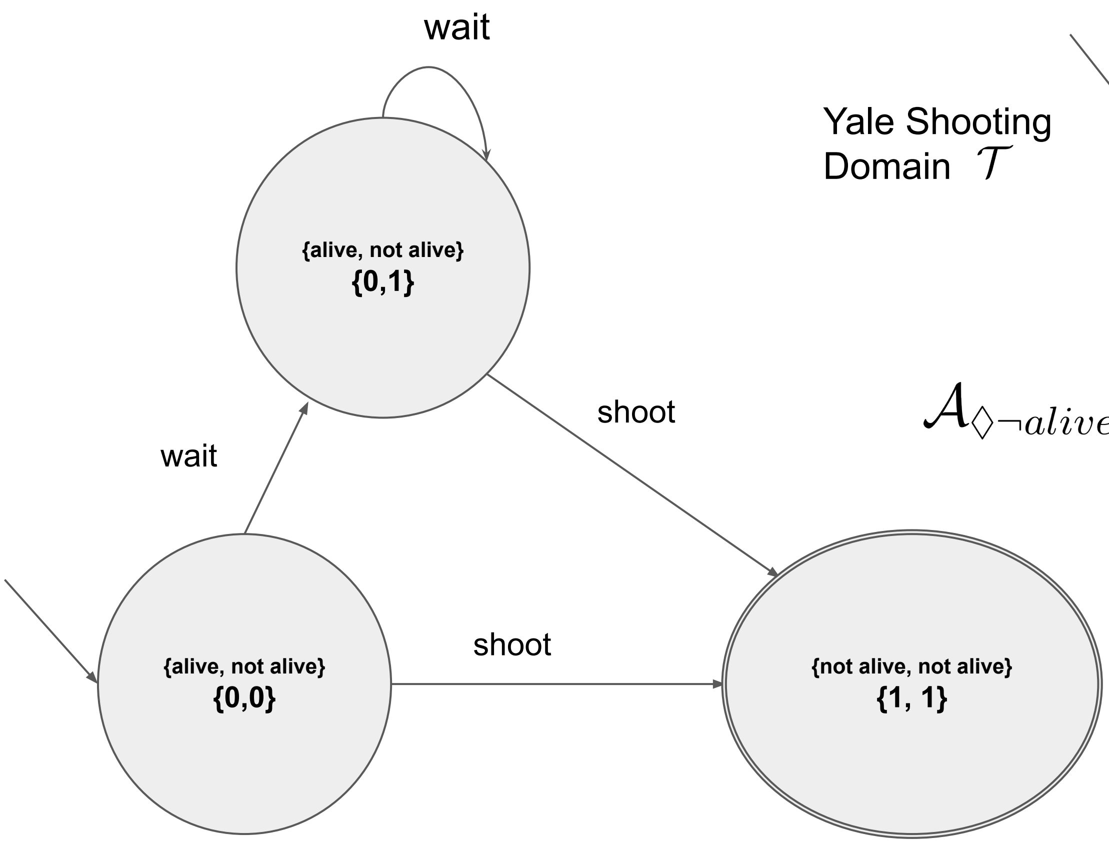
Yale Shooting Domain $\mathcal{T}$



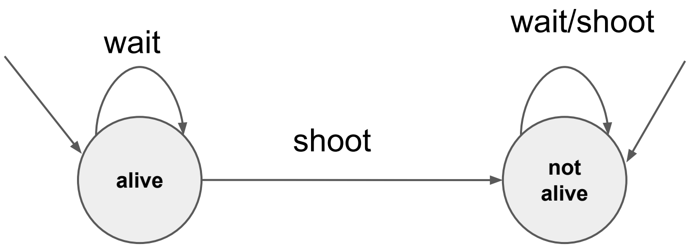
$\mathcal{A}_{\Diamond\neg alive}$



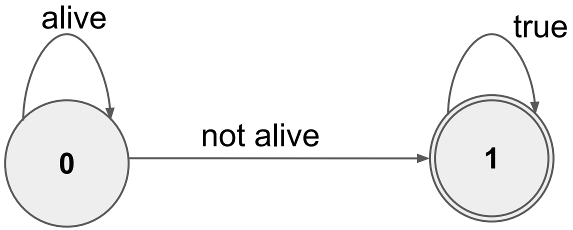
Conformant Planning-Yale Shooting Example



Yale Shooting Domain $\mathcal{T}$



$\mathcal{A}_{\Diamond \neg alive}$



$\mathcal{A}_{\mathcal{T}}^{\Diamond \neg alive}$

Contingent Planning

Incomplete information on the initial state and observability on the states

To synthesize a plan compute NFA $A_{\mathcal{T}}^{\varphi} = (Act^k, \mathcal{S}_T, \mathcal{S}_{T0}, \delta_T, F_T)$:

- $\mathcal{S}_T = Q^k \times S^k \times \mathcal{E}_k$
- $\mathcal{S}_{T0} = Q_0^k \times \{(s_{00}, \ldots, s_{0k-1})\} \times \equiv_0$ where $i \equiv_0 j$ iff $s_{0i} = s_{0j}$
- $(\vec{q_j}, \vec{s_j}, \equiv') \in \delta_T((\vec{q_i}, \vec{s_i}), \vec{a}, \equiv)$ iff $s_{jh} = R(s_{ih}, a_h)$ and

  if $l \equiv m$ then $a_l = a_m, l \equiv' m$ iff $l \equiv m$ and $\pi(s_{jl}) = \pi(s_{jm})$
- $F_T = F^k \times S^k \times \mathcal{E}_k$

Theorem: A plan p for $\mathcal{T}$ realizing the specification $\mathcal{A}_{\varphi}$ exists iff $L(\mathcal{A}_{\mathcal{T}}^{\varphi}) \neq \emptyset$.

Conclusion

Contributions:

- Extension of the work of [De Giacomo, Vardi 1999] in the finite traces setting with LTLf goals
- The techniques introduced here can be extended to work for a wide range of transition-based formalisms
- The approach works for any temporal specification language that can be translated into automaton (e.g. LDL_f, PPLTL, PPLDL etc.)

Future works:

- Investigate the complexity of the planning problems
- Provide implementations of the presented techniques
- Find interesting use-cases and applications