LTL_f Best-Effort Synthesis in Nondeterministic Planning Domains

Gianmarco Parretti
`parretti@diag.uniroma1.it`

(joint work with Giuseppe De Giacomo and Shufang Zhu)

Sapienza University of Rome

March 2023

Background

- **Game-theoretic** approach to **planning** for goals in **Linear Temporal Logic on Finite Traces (LTL$_f$)**
- ***Reactive synthesis*** is a **general form of planning** that finds a strategy to realize a temporal goal, i.e., a ***winning strategy***
- **Best-effort synthesis** is an **extension** of **reactive synthesis** that finds a strategy that ensures that the agent does its best to achieve the goal, i.e., a ***best-effort strategy***

Comparing Synthesis Approaches to Planning

	Reactive Synthesis	Best-Effort Synthesis
Strategy Existence	**Realizable tasks**	**Always**
Environment	**Adversarial**	**Any**
Goal Completion	**Realizable tasks**	**Best-effort**[1]
Goal Complexity	2EXPTIME-**complete**	2EXPTIME-**complete**
Domain Complexity	EXPTIME-**complete**	EXPTIME-**complete**

[1]Ensures goal for realizable tasks; else, ensures goal for a maximal set of environment behaviors.

Comparing Synthesis Approaches to Planning

	Reactive Synthesis	Best-Effort Synthesis
Strategy Existence	**Realizable tasks**	**Always**
Environment	**Adversarial**	**Any**
Goal Completion	**Realizable tasks**	**Best-effort**[1]
Goal Complexity	2EXPTIME-**complete**	2EXPTIME-**complete**
Domain Complexity	EXPTIME-**complete**	EXPTIME-**complete**

- **Best-effort synthesis**, suitable in **non-strictly adversarial dmains**, e.g., **FOND**

[1]Ensures goal for realizable tasks; else, ensures goal for a maximal set of environment behaviors.

Framework

- A **planning domain** is defined as $\mathcal{D} = (2^{\mathcal{F}}, s_0, Act, React, \alpha, \beta, \delta)$

Framework

- A **planning domain** is defined as $\mathcal{D} = (2^{\mathcal{F}}, s_0, Act, React, \alpha, \beta, \delta)$
  - **Fluents:** $\mathcal{F}$; **State Space:** $2^{\mathcal{F}}$; **Size of** $\mathcal{D}$**:** $|\mathcal{F}|$
  - **Initial state:** $s_0 \in 2^{\mathcal{F}}$

Framework

- A **planning domain** is defined as $\mathcal{D} = (2^{\mathcal{F}}, s_0, Act, React, \alpha, \beta, \delta)$
  - **Fluents:** $\mathcal{F}$; **State Space:** $2^{\mathcal{F}}$; **Size of** $\mathcal{D}$**:** $|\mathcal{F}|$
  - **Initial state:** $s_0 \in 2^{\mathcal{F}}$
  - **Agent actions:** Act
  - **Environment Reactions:** $React$

Framework

- A **planning domain** is defined as $\mathcal{D} = (2^{\mathcal{F}}, s_0, Act, React, \alpha, \beta, \delta)$
  - **Fluents:** $\mathcal{F}$; **State Space:** $2^{\mathcal{F}}$; **Size of** $\mathcal{D}$**:** $|\mathcal{F}|$
  - **Initial state:** $s_0 \in 2^{\mathcal{F}}$
  - **Agent actions:** Act
  - **Environment Reactions:** $React$
  - **Agent actions preconditions:** $\alpha : 2^{\mathcal{F}} \to 2^{Act}$
  - **Environment reaction preconditions:** $\beta : 2^{\mathcal{F}} \times Act \to 2^{React}$

Framework

- A **planning domain** is defined as $\mathcal{D} = (2^{\mathcal{F}}, s_0, Act, React, \alpha, \beta, \delta)$
  - **Fluents:** $\mathcal{F}$; **State Space:** $2^{\mathcal{F}}$; **Size of** $\mathcal{D}$**:** $|\mathcal{F}|$
  - **Initial state:** $s_0 \in 2^{\mathcal{F}}$
  - **Agent actions:** Act
  - **Environment Reactions:** $React$
  - **Agent actions preconditions:** $\alpha : 2^{\mathcal{F}} \to 2^{Act}$
  - **Environment reaction preconditions:** $\beta : 2^{\mathcal{F}} \times Act \to 2^{React}$
  - **Transition function:** $\delta : 2^{\mathcal{F}} \times Act \times React \mapsto 2^{\mathcal{F}}$;
    - $\delta(s, a, r)$ defined only if $a \in \alpha(s)$ and $r \in \beta(s, a)$

Best-Effort Synthesis in Nondeterministic Planning Domains

- **Agent strategy:** $\sigma : (2^{\mathcal{F}})^+ \rightarrow \mathit{Act}$
- **Environment strategy:** $\gamma : \mathit{Act}^+ \rightarrow \mathit{React}$

[2]Dominance is defined only for strategies always satisfying action (resp. reaction) preconditions

Best-Effort Synthesis in Nondeterministic Planning Domains

- **Agent strategy:** $\sigma : (2^{\mathcal{F}})^+ \to \mathit{Act}$
- **Environment strategy:** $\gamma : \mathit{Act}^+ \to \mathit{React}$

Dominance[2]

Agent strategy σ_1 **dominates** σ_2 for goal φ in domain $\mathcal{D}$, written $\sigma_1 \geq_{\varphi|\mathcal{D}} \sigma_2$, if for every environment strategy γ, $\mathtt{Play}(\sigma_2, \gamma) \models_{\mathcal{D}} \varphi$ implies $\mathtt{Play}(\sigma_1, \gamma) \models_{\mathcal{D}} \varphi$. σ_1 **strictly dominates** σ_2, written $\sigma_1 >_{\varphi|\mathcal{D}} \sigma_2$, if $\sigma_1 \geq_{\varphi|\mathcal{D}} \sigma_2$ and $\sigma_2 \not\geq_{\varphi|\mathcal{D}} \sigma_1$

[2]Dominance is defined only for strategies always satisfying action (resp. reaction) preconditions

Best-Effort Synthesis in Nondeterministic Planning Domains

- **Agent strategy:** $\sigma : (2^{\mathcal{F}})^+ \rightarrow \mathit{Act}$
- **Environment strategy:** $\gamma : \mathit{Act}^+ \rightarrow \mathit{React}$

Dominance[2]

Agent strategy σ_1 **dominates** σ_2 for goal φ in domain $\mathcal{D}$, written $\sigma_1 \geq_{\varphi|\mathcal{D}} \sigma_2$, if for every environment strategy γ, $\texttt{Play}(\sigma_2, \gamma) \models_{\mathcal{D}} \varphi$ implies $\texttt{Play}(\sigma_1, \gamma) \models_{\mathcal{D}} \varphi$. σ_1 **strictly dominates** σ_2, written $\sigma_1 >_{\varphi|\mathcal{D}} \sigma_2$, if $\sigma_1 \geq_{\varphi|\mathcal{D}} \sigma_2$ and $\sigma_2 \not\geq_{\varphi|\mathcal{D}} \sigma_1$

LTL$_f$ Best-Effort Synthesis in Nondeterministic Planning Domains

Given: Planning domain $\mathcal{D}$ and LTL$_f$ agent goal φ over $\mathcal{F}$

Obtain: A **best-effort strategy**, i.e., σ for which there is no σ' s.t. $\sigma' >_{\varphi|\mathcal{D}} \sigma$

[2]Dominance is defined only for strategies always satisfying action (resp. reaction) preconditions

Synthesis Technique [1]

- **Synthesis technique** based on solving simultaneously suitable DFA **games**

Synthesis Technique [1]

- **Synthesis technique** based on solving simultaneously suitable DFA **games**
- **1a.** Transform **planning domain** $\mathcal{D}$ into **transition system**

$$\mathcal{D}^{+} = (Act \times React, 2^{\mathcal{F}} \cup \{s^{ag}_{err}, s^{env}_{err}\}, s_0, \delta')$$

  with:
  - Agent and environment error states s^{ag}_{err} and s^{env}_{err}.
  - **Transition function** δ' s.t. $\delta'(s, a, r) = s$ if $s \in \{s^{ag}_{err}, s^{env}_{err}\}$, and

$$\delta'(s, a, r) = \begin{cases} \delta(s, a, r) & \text{if } a \in \alpha(s) \text{ and } r \in \beta(s, a) \\ s^{ag}_{err} & \text{if } a \notin \alpha(s) \\ s^{env}_{err} & \text{if } a \in \alpha(s) \text{ and } r \notin \beta(s, a) \end{cases}$$

Synthesis Technique [1]

- **Synthesis technique** based on solving simultaneously suitable DFA **games**
- **1a.** Transform **planning domain** $\mathcal{D}$ into **transition system**

$$\mathcal{D}^+ = (Act \times React, 2^{\mathcal{F}} \cup \{s_{err}^{ag}, s_{err}^{env}\}, s_0, \delta')$$

  with:
  - Agent and environment error states s_{err}^{ag} and s_{err}^{env}.
  - **Transition function** δ' s.t. $\delta'(s, a, r) = s$ if $s \in \{s_{err}^{ag}, s_{err}^{env}\}$, and

$$\delta'(s, a, r) = \begin{cases} \delta(s, a, r) & \text{if } a \in \alpha(s) \text{ and } r \in \beta(s, a) \\ s_{err}^{ag} & \text{if } a \notin \alpha(s) \\ s_{err}^{env} & \text{if } a \in \alpha(s) \text{ and } r \notin \beta(s, a) \end{cases}$$

- **1b.** Transform **agent goal** φ into DFA $\mathcal{A}_\varphi = (\mathcal{T}_\varphi, Reach(R_\varphi))$ with $\mathcal{T}_\varphi = (2^{\mathcal{F}}, Q, q_0, \varrho)$ and $R_\varphi \subseteq Q$. Note $\varrho : Q \times 2^{\mathcal{F}} \to Q$

Synthesis Technique [2]

- **2. Compose** $\mathcal{D}^+$ and $\mathcal{T}_\varphi$ into
  $$\mathcal{G} = (Act \times React, (2^{\mathcal{F}} \cup \{s_{err}^{ag}, s_{err}^{env}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)$$
  with **transition function** ∂:

  $$\partial((s, q), a, r) = \begin{cases} (s', \varrho(q, s')) & \text{if } s' \notin \{s_{err}^{ag}, s_{err}^{env}\} \\ (s_{err}^{ag}, q) & \text{if } s' = s_{err}^{ag} \\ (s_{err}^{env}, q) & \text{if } s' = s_{err}^{env} \end{cases}$$

  where $s' = \delta'(s, a, r)$

Synthesis Technique [2]

- **2. Compose** $\mathcal{D}^+$ and $\mathcal{T}_\varphi$ into
$$\mathcal{G} = (Act \times React, (2^{\mathcal{F}} \cup \{s_{err}^{ag}, s_{err}^{env}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)$$
with **transition function** ∂:
$$\partial((s,q),a,r) = \begin{cases} (s', \varrho(q, s')) & \text{if } s' \notin \{s_{err}^{ag}, s_{err}^{env}\} \\ (s_{err}^{ag}, q) & \text{if } s' = s_{err}^{ag} \\ (s_{err}^{env}, q) & \text{if } s' = s_{err}^{env} \end{cases}$$
where $s' = \delta'(s,a,r)$
- **3.** Compute a **positional winning strategy** in game κ_{adv} in game $(\mathcal{G}, Reach(\neg S_{err}^{ag} \cap (S_{err}^{env} \cup R'_\varphi)))$. Let W_{adv} be the **winning region**

Synthesis Technique [2]

- **2. Compose** $\mathcal{D}^+$ and $\mathcal{T}_\varphi$ into
  $$\mathcal{G} = (Act \times React, (2^{\mathcal{F}} \cup \{s_{err}^{ag}, s_{err}^{env}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)$$
  with **transition function** ∂:
  $$\partial((s,q),a,r) = \begin{cases} (s', \varrho(q, s')) & \text{if } s' \notin \{s_{err}^{ag}, s_{err}^{env}\} \\ (s_{err}^{ag}, q) & \text{if } s' = s_{err}^{ag} \\ (s_{err}^{env}, q) & \text{if } s' = s_{err}^{env} \end{cases}$$
  where $s' = \delta'(s, a, r)$
- **3.** Compute a **positional winning strategy** in game κ_{adv} in game $(\mathcal{G}, Reach(\neg S_{err}^{ag} \cap (S_{err}^{env} \cup R'_\varphi)))$. Let W_{adv} be the **winning region**
- **4.** Compute a **positional cooperatively winning strategy** κ_{coop} in game $(\mathcal{G}, Reach(\neg S_{err}^{ag} \cap \neg S_{err}^{env} \cap R'_\varphi))$. Let W_{coop} be the **winning region**

Synthesis Technique [3]

- **5. Return** the agent strategy σ **induced** by κ constructed as follows:

$$\kappa(s,q) = \begin{cases} \kappa_{adv}(s,q) & \text{if } (s,q) \in W_{adv} \\ \kappa_{coop}(s,q) & \text{if } (s,q) \in W_{coop} / W_{adv} \\ \text{any } a \in \alpha(s) & \text{if } (s,q) \notin W_{coop} \cup W_{adv} \end{cases}$$

Synthesis Technique [3]

- **5. Return** the agent strategy σ **induced** by κ constructed as follows:

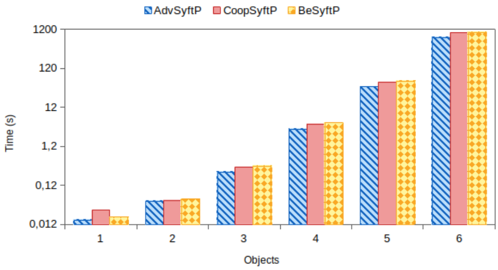
$$\kappa(s,q) = \begin{cases} \kappa_{adv}(s,q) & \text{if } (s,q) \in W_{adv} \\ \kappa_{coop}(s,q) & \text{if } (s,q) \in W_{coop}/W_{adv} \\ \text{any } a \in \alpha(s) & \text{if } (s,q) \notin W_{coop} \cup W_{adv} \end{cases}$$

Correctness

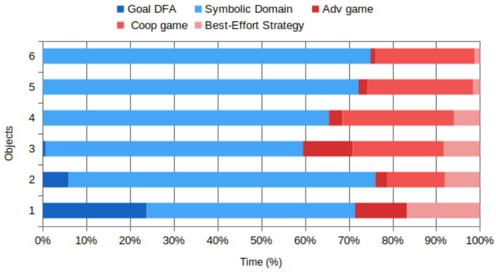
- **Thm 1**. There exists a **strong plan** for φ in $\mathcal{D}$ iff there exists a **winning strategy** for $(\mathcal{G}, Reach(\neg S^{ag}_{err} \cap (S^{env}_{err} \cup R'_{\varphi}))$
- **Thm 2**. There exists a **cooperative plan** for φ in $\mathcal{D}$ iff there exists a **cooperatively winning strategy** for $(\mathcal{G}, Reach(\neg S^{ag}_{err} \cap \neg S^{env}_{err} \cap R'_{\varphi}))$

Implementation and Empirical Evaluation

- **Symbolic implementations:** *BeSyftP*, *AdvSyftP*, *CoopSyftP*
- **Empirical evaluation** on scalable pick-and-place benchmarks



(a) Comparison of *BeSyftP*, *AdvSyftP* and *CoopSyftP*



(b) Relative time cost of major operations in *BeSyftP*

Conclusion and Future Works

- **Conclusion**
  - **Best-effort synthesis** is suitable to **address unrealizability** of agent tasks
  - Brings only a minimal overhead wrt computing winning strategies
- **Future Works**
  - Empirical validation
  - **Extension** to multiple planning domains and agent goal specifications