LTL\textsubscript{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

<table>
<thead>
<tr>
<th></th>
<th>Reactive Synthesis</th>
<th>Best-Effect Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy Existence</strong></td>
<td>Realizable tasks</td>
<td>Always</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>Adversarial</td>
<td>Any</td>
</tr>
<tr>
<td><strong>Goal Completion</strong></td>
<td>Realizable tasks</td>
<td>Best-effort&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Goal Complexity</strong></td>
<td>$2\text{EXPTIME-complete}$</td>
<td>$2\text{EXPTIME-complete}$</td>
</tr>
<tr>
<td><strong>Domain Complexity</strong></td>
<td>$\text{EXPTIME-complete}$</td>
<td>$\text{EXPTIME-complete}$</td>
</tr>
</tbody>
</table>

<sup>1</sup>Ensures goal for realizable tasks; else, ensures goal for a maximal set of environment behaviors.
Comparing Synthesis Approaches to Planning

<table>
<thead>
<tr>
<th></th>
<th>Reactive Synthesis</th>
<th>Best-Effort Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy Existence</strong></td>
<td>Realizable tasks</td>
<td>Always</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>Adversarial</td>
<td>Any</td>
</tr>
<tr>
<td><strong>Goal Completion</strong></td>
<td>Realizable tasks</td>
<td>Best-effort(^1)</td>
</tr>
<tr>
<td><strong>Goal Complexity</strong></td>
<td>2\text{EXPTIME}\text{-complete}</td>
<td>2\text{EXPTIME}\text{-complete}</td>
</tr>
<tr>
<td><strong>Domain Complexity</strong></td>
<td>\text{EXPTIME}\text{-complete}</td>
<td>\text{EXPTIME}\text{-complete}</td>
</tr>
</tbody>
</table>

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

\(^1\)Ensures goal for realizable tasks; else, ensures goal for a maximal set of environment behaviors.
A planning domain is defined as \( \mathcal{D} = (2^F, s_0, Act, React, \alpha, \beta, \delta) \).
Framework

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

▶ **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 \( D = (2^F, s_0, Act, React, \alpha, \beta, \delta) \)
  - Fluents: \( F \); State Space: \( 2^F \); Size of \( D \): \( |F| \)
  - Initial state: \( s_0 \in 2^F \)
  - Agent actions: \( Act \)
  - Environment Reactions: \( React \)
Framework

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

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

- Fluents: $\mathcal{F}$; State Space: $2^\mathcal{F}$; Size of $\mathcal{D}$: $|\mathcal{F}|$
- Initial state: $s_0 \in 2^\mathcal{F}$
- Agent actions: $\text{Act}$
- Environment Reactions: $\text{React}$
- Agent actions preconditions: $\alpha : 2^\mathcal{F} \rightarrow 2^{\text{Act}}$
- Environment reaction preconditions: $\beta : 2^\mathcal{F} \times \text{Act} \rightarrow 2^{\text{React}}$
- Transition function: $\delta : 2^\mathcal{F} \times \text{Act} \times \text{React} \rightarrow 2^\mathcal{F}$; $\delta(s, a, r)$ defined only if $a \in \alpha(s)$ and $r \in \beta(s, a)$
Best-Effort Synthesis in Nondeterministic Planning Domains

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

---

[2] Dominance is defined only for strategies always satisfying action (resp. reaction) preconditions.
Best-Effort Synthesis in Nondeterministic Planning Domains

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

**Dominance**

Agent strategy \( \sigma_1 \) **dominates** \( \sigma_2 \) for goal \( \varphi \) in domain \( D \), written \( \sigma_1 \geq_{\varphi|D} \sigma_2 \), if for every environment strategy \( \gamma \), \( \text{Play}(\sigma_2, \gamma) \models_D \varphi \) implies \( \text{Play}(\sigma_1, \gamma) \models_D \varphi \).

Agent strategy \( \sigma_1 \) **strictly dominates** \( \sigma_2 \), written \( \sigma_1 >_{\varphi|D} \sigma_2 \), if \( \sigma_1 \geq_{\varphi|D} \sigma_2 \) and \( \sigma_2 \not\geq_{\varphi|D} \sigma_1 \).

---

[2] Dominance is defined only for strategies always satisfying action (resp. reaction) preconditions.
Agent strategy: $\sigma : (2^F)^+ \rightarrow Act$

Environment strategy: $\gamma : Act^+ \rightarrow React$

Dominance\[^2\]

Agent strategy $\sigma_1$ dominates $\sigma_2$ for goal $\varphi$ in domain $D$, written $\sigma_1 \geq_{\varphi|D} \sigma_2$, if for every environment strategy $\gamma$, $\text{Play}(\sigma_2, \gamma) \models_D \varphi$ implies $\text{Play}(\sigma_1, \gamma) \models_D \varphi$.

Agent strategy $\sigma_1$ strictly dominates $\sigma_2$, written $\sigma_1 >_{\varphi|D} \sigma_2$, if $\sigma_1 \geq_{\varphi|D} \sigma_2$ and $\sigma_2 \not\geq_{\varphi|D} \sigma_1$.

LTL\(_f\) Best-Effort Synthesis in Nondeterministic Planning Domains

Given: Planning domain $D$ and LTL\(_f\) agent goal $\varphi$ over $F$

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

\[^2\]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

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

with:

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

$$\delta'(s, a, r) = \begin{cases} 
\delta(s, a, r) & \text{if } a \in \alpha(s) \text{ and } r \in \beta(s, a) \\
\delta s_{err}^{ag} & \text{if } a \notin \alpha(s) \\
\delta s_{err}^{env} & \text{if } a \in \alpha(s) \text{ and } r \notin \beta(s, a) 
\end{cases}$$
Synthesis Technique [1]

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

- **1a.** Transform **planning domain** $\mathcal{D}$ into **transition system**

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

with:

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

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

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

$$
G = (\text{Act} \times \text{React}, (2^F \cup \{s^{ag}_{err}, s^{env}_{err}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)
$$

with transition function $\partial$:

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

where $s' = \delta'(s, a, r)$
2. Compose $\mathcal{D}^+$ and $\mathcal{T}_\varphi$ into

$$\mathcal{G} = (\text{Act} \times \text{React}, (2^F \cup \{s_{err}^{ag}, s_{err}^{env}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)$$

with transition function $\partial$:

$$\partial((s, q), a, r) = \begin{cases} 
(s', \varrho(q, s')) & \text{if } s' \not\in \{s_{err}^{ag}, s_{err}^{env}\} \\
(s_{err}^{ag}, q) & \text{if } s' = s_{err}^{ag} \\
(s_{err}^{env}, q) & \text{if } s' = s_{err}^{env}
\end{cases}$$

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

3. Compute a positional winning strategy in game $\kappa_{adv}$ in game $(\mathcal{G}, \text{Reach}(\neg S_{err}^{ag} \cap (S_{err}^{env} \cup R'_\varphi)))$. Let $W_{adv}$ be the winning region
Synthesis Technique [2]

2. Compose $D^+$ and $T_\varphi$ into

$$G = (Act \times React, (2^F \cup \{s_{err}^{ag}, s_{err}^{env}\}) \times Q, (s_0, \varrho(q_0, s_0)), \partial)$$

with transition function $\partial$:

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

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

3. Compute a **positional winning strategy** in game $\kappa_{adv}$ in game

$$(G, Reach(\neg S_{err}^{ag} \cap (S_{err}^{env} \cup R'_\varphi))))$$. Let $W_{adv}$ be the **winning region**

4. Compute a **positional cooperatively winning strategy** $\kappa_{coop}$ in game

$$(G, Reach(\neg S_{err}^{ag} \cap \neg S_{err}^{env} \cap R'_\varphi)))$$. Let $W_{coop}$ be the **winning region**
5. **Return** the agent strategy \( \sigma \) **induced** by \( \kappa \) constructed as follows:

\[
\kappa(s, q) = \begin{cases} 
\kappa_{\text{adv}}(s, q) & \text{if } (s, q) \in W_{\text{adv}} \\
\kappa_{\text{coop}}(s, q) & \text{if } (s, q) \in W_{\text{coop}} / W_{\text{adv}} \\
\text{any } a \in \alpha(s) & \text{if } (s, q) \not\in W_{\text{coop}} \cup W_{\text{adv}}
\end{cases}
\]
Synthesis Technique [3]

5. Return the agent strategy $\sigma$ induced by $\kappa$ constructed as follows:

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

Correctness

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

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