Synthesis of Maximally Permissive Strategies for LTL$_f$ Specifications

Shufang Zhu, Giuseppe De Giacomo

IJCAI-ECAI 2022

Sapienza University of Rome

LTL_f Synthesis

- Autonomy, one of the grand objectives AI
  - Autonomous system, react to environment changes

- LTL_f synthesis [De Giacomo & Vardi, 2015]
  - Linear Temporal Logic on finite traces [De Giacomo & Vardi, 2013]
  - **Obtain** a winning strategy (system model) **From** a declarative specification

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec
  - Deterministic: $\sigma : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$, returns a single agent action

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec
  - Deterministic: $\sigma : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$, returns a single agent action
  - Nondeterministic: $\Pi : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec
    - Deterministic: $\sigma : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$, returns a single agent action
    - Nondeterministic: $\Pi : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically

- Every nondet. strategy Π captures a set of det. strategies σ

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec
    - Deterministic: $\sigma : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$, returns a single agent action
    - Nondeterministic: $\Pi : (2^{\mathcal{X} \cup \mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically
- Every nondet. strategy Π captures a set of det. strategies σ
    - preserves more autonomy

Strategy and Autonomy

- Strategy: prescribes for **history** + **env action**, makes **agn response**, generated traces satisfy spec
  - Deterministic: $\sigma : (2^{\mathcal{X}\cup\mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$, returns a single agent action
  - Nondeterministic: $\Pi : (2^{\mathcal{X}\cup\mathcal{Y}})^* \times 2^{\mathcal{X}} \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically
- Every nondet. strategy Π captures a set of det. strategies σ
  - preserves more autonomy
  - agent chooses det. strategy while in execution, without committing to any specific one beforehand

Maximally Permissive Strategy for LTL_f Specifications

- Maximally Permissive Strategy (MaxSet)
  - MaxSet = {all deterministic winning strategies}
  - Maximal autonomy

Maximally Permissive Strategy for LTL$_f$ Specifications

- Maximally Permissive Strategy (MaxSet)
  - MaxSet = {all deterministic winning strategies}
  - Maximal autonomy

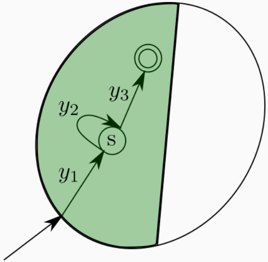
Theorem

LTL$_f$ **DOES NOT** admit a single nondeterministic strategy that captures MaxSet.

MaxSet of LTL$_f$ through Reachability Games

Observation

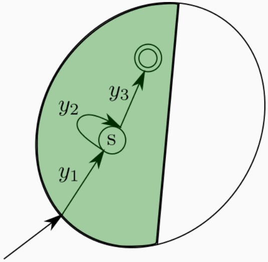
- Winning region maintains the ability to win
- "staying in the winning region" $\neq$ "win the game"
  - Taking y_2 at s forever does not lead to the goal



MaxSet of LTL$_f$ through Reachability Games

Key idea

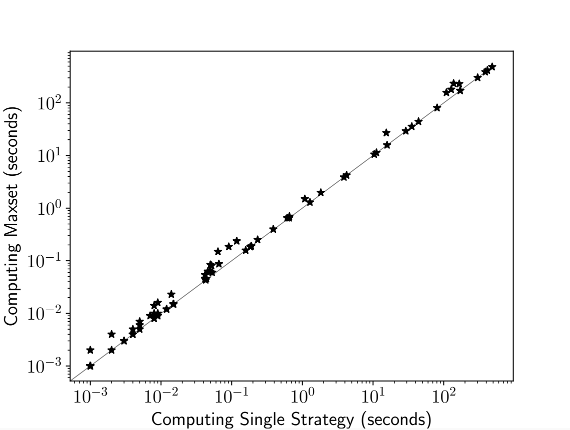
- Every winning strategy switches
  - from "deferring" (y_2 at s, staying in the winning region)
  - to "non-deferring" (y_3 at s, surely reaching the goal)
- No matter when to switch, the switching is mandatory



MaxSet of LTL$_f$ Specifications

- A single nondet. strategy Π_{df} captures all the deferring strategies (maintaining the ability to win)
- A single nondet. strategy Π_{ndf} captures all the non-deferring strategies (assuring to win)
- A declarative constraint forcing any choice function ch eventually switch from Π_{df} to Π_{ndf}

Experimental Evaluations



- Computing MaxSet only brings a **minor overhead** comparing to computing a single strategy

Conclusions and Future Research

- MaxSet of LTL_f specifications
  - Two nondeterministic strategies and a constraint
  - A minimal overhead

- Improve the performance of LTL_f synthesis
- MaxSet of other specification languages, e.g., LTL

Poster: Slot 201 at Row 6