LTL_f Synthesis as AND-OR Graph Search: Knowledge Compilation at Work

Giuseppe De Giacomo[1], Marco Favorito[1,2], Jianwen Li[3], Moshe Y. Vardi[4], Sheng-ping Xiao[3] and **Shufang Zhu**[1]

IJCAI-ECAI 2022

[1]Sapienza University of Rome
[2]Banca d'Italia
[3]East China Normal University
[4]Rice University

LTL_f Synthesis

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

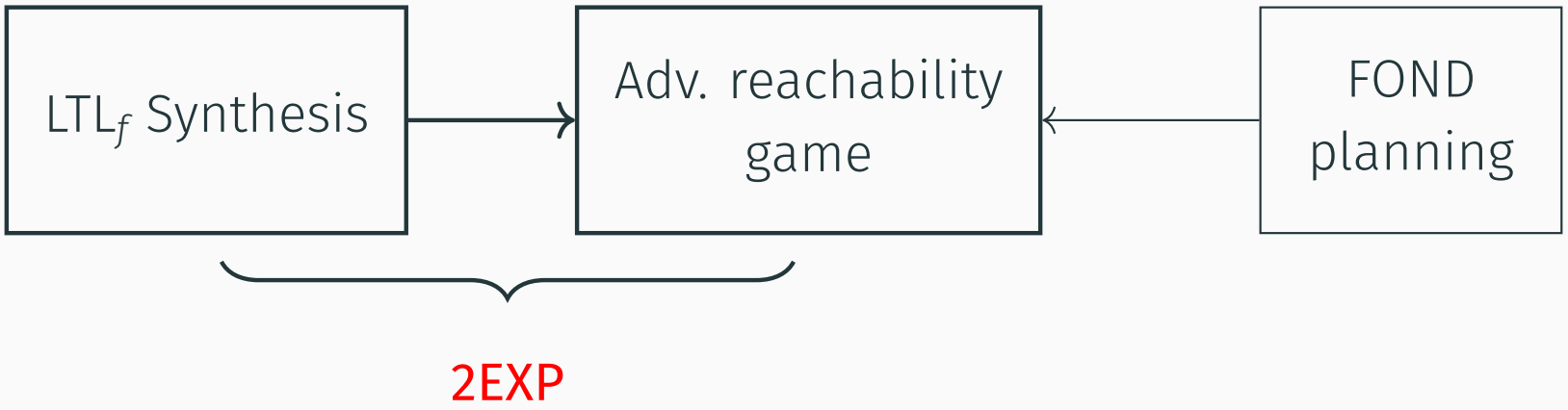
- LTL_f synthesis [De Giacomo & Vardi, 2015]
  - Linear Temporal Logic on finite traces [De Giacomo & Vardi, 2013]
  - **Obtain** a system **From** a declarative specification
  - System interacting with the environment, generated executions satisfy the specification

LTL_f Synthesis and FOND Planning

LTL_f Synthesis and FOND Planning

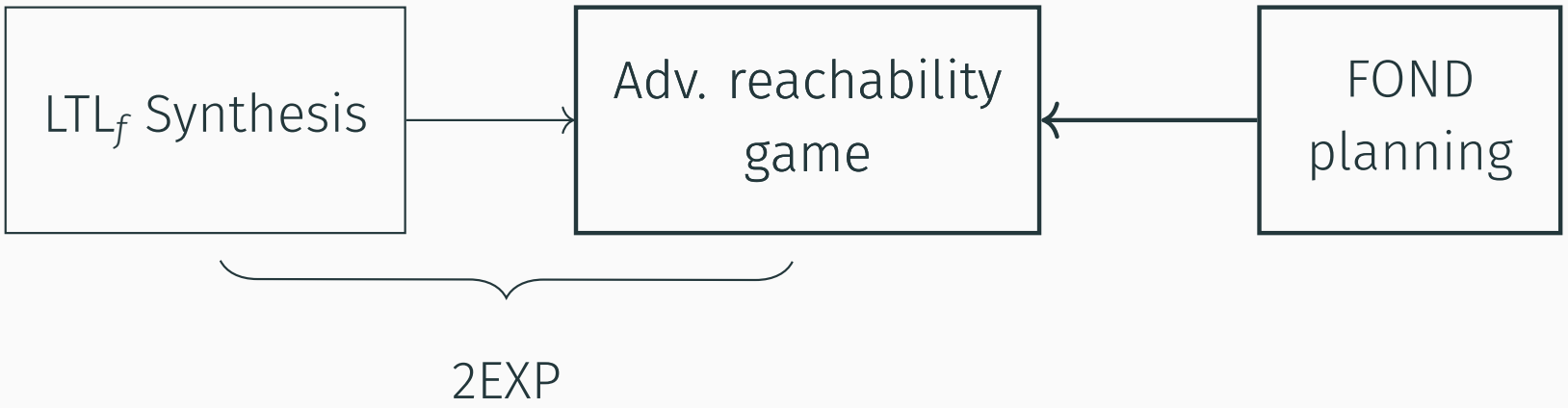
- Fixpoint computation on constructed DFA, e.g., Model Checking

LTL_f Synthesis and FOND Planning



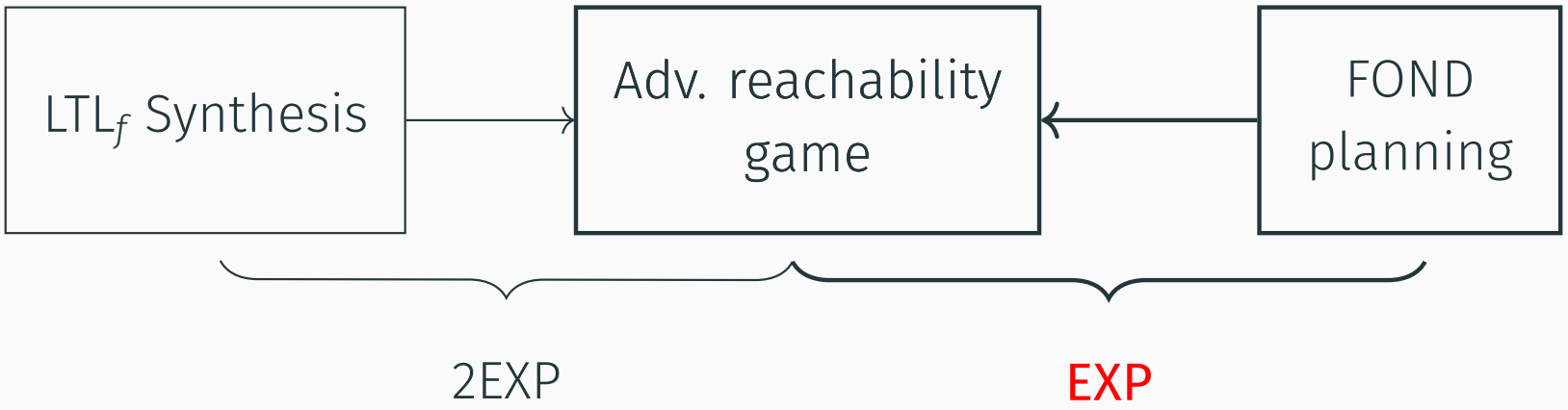
- Fixpoint computation on constructed DFA, e.g., Model Checking
- Problem: DFA can be double-exponential

LTL_f Synthesis and FOND Planning



- Forward search while expanding the space, e.g., FOND planning

LTL_f Synthesis and FOND Planning



- Forward search while expanding the space, e.g., FOND planning
- However, State space in FOND, only single-exponential

LTL_f Synthesis as AND-OR Graph Search

- Forward LTL_f synthesis approach
  - Adopting AND-OR graph search, as in FOND planning
  - Over double-exponential search space

LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction

LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction
   - Construct search space on-the-fly via formula progression

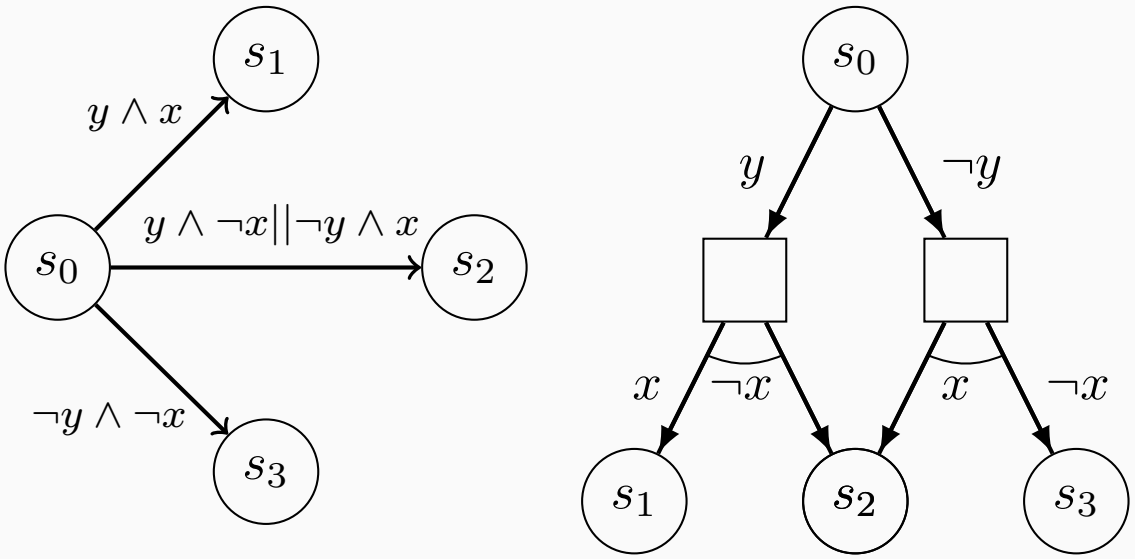
LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction
   - Construct search space on-the-fly via formula progression
   - LTL_f formula φ, as a DFA state, what happens **now** (label), what should happen **next** accordingly (successor state)

LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction
   - Construct search space on-the-fly via formula progression
   - LTL_f formula φ, as a DFA state, what happens **now** (label), what should happen **next** accordingly (successor state)

2. Adversarial reachability game on DFA as AND-OR graph search



LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction
   - Construct search space on-the-fly via formula progression
   - LTL_f formula φ, as a DFA state, what happens **now** (label), what should happen **next** accordingly (successor state)

2. Adversarial reachability game on DFA as AND-OR graph search

3. Sentential Decision Diagrams (SDDs) [Darwiche et al., 2011]

LTL_f Synthesis as AND-OR Graph Search

1. Direct LTL_f-to-DFA construction
   - Construct search space on-the-fly via formula progression
   - LTL_f formula φ, as a DFA state, what happens **now** (label), what should happen **next** accordingly (successor state)

2. Adversarial reachability game on DFA as AND-OR graph search

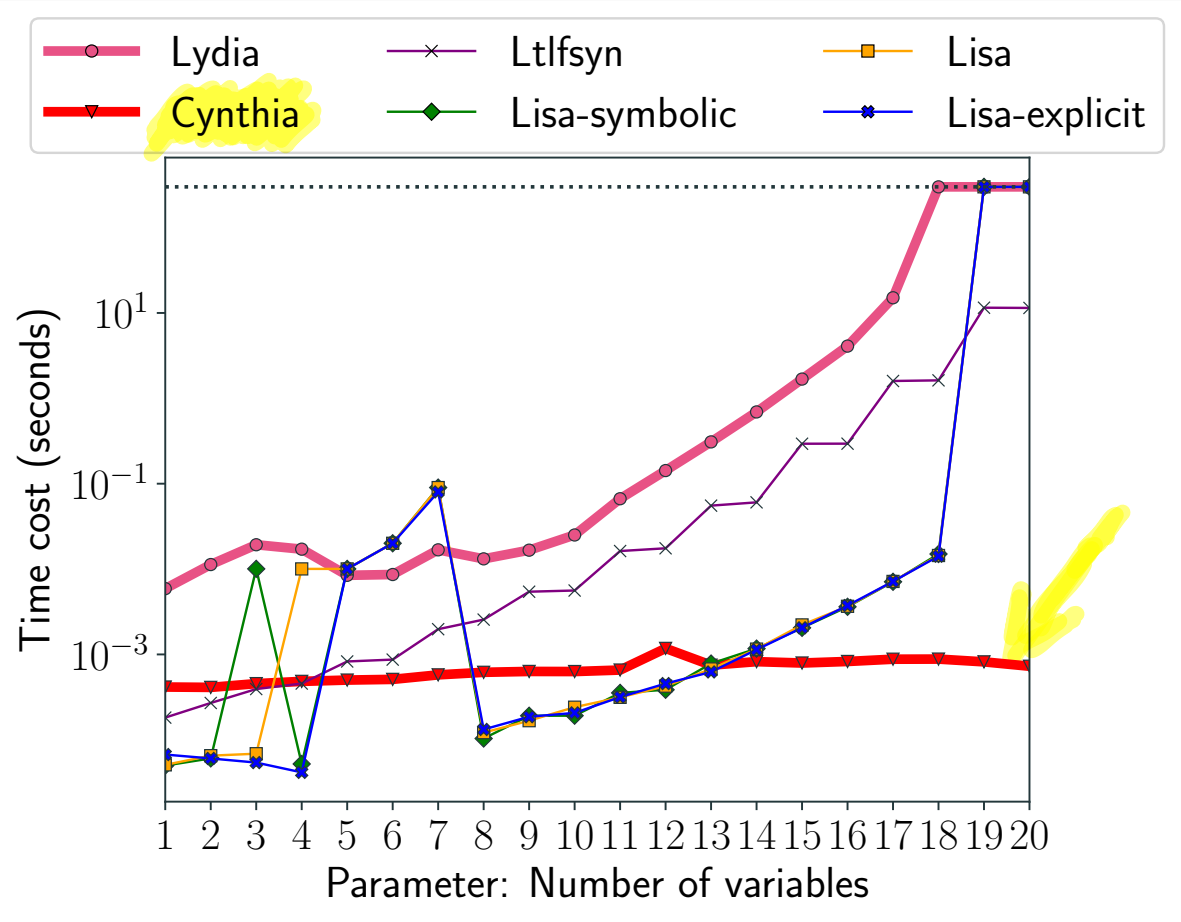
3. Sentential Decision Diagrams (SDDs) [Darwiche et al., 2011]
   - Compress labels leading to the same nodes, reduce the branching factor

Empirical Evaluation

- LTL_f Synthesis as AND-OR Graph Search, Cynthia
  - Uninformed search

- Baseline tools
  - Lisa [Bansal et al., 2020], Lydia [De Giacomo & Favorito., 2021], and Ltlfsyn [Xiao et al., 2021]

Experimental Results on Until-Patterns



$$U(n) = p_1 U(p_2 U(\ldots p_{n-1} U p_n))$$

Conclusions and Future Research

- Forward LTL_f synthesis adopting AND-OR graph search
  - Direct LTL_f-to-DFA construction
  - Expand the search space while performing AND-OR graph search
  - SDD to reduce branching factor
- Uninformed search, promising synthesis performance

- Move from uninformed search to informed search exploiting heuristics

Poster: Slot 200 at Row 6