Symbolic LTL_f Synthesis

Shufang Zhu [1], Lucas M. Tabajara [2] , Jianwen Li [2], Geguang Pu [1], Moshe Y. Vardi [2]

[1]East China Normal University, [2]Rice University

Introduction

- Linear Temporal Logic (LTL)
- Extensively used in AI and CS [BK, AAAI'96; CGV, KR'02; GV, ECP'99]
- LTL_f: LTL over *finite* traces
- Many AI problems of interest reduce to LTL_f synthesis [De Giacomo & Vardi, IJCAI'15]

LTL_f

LTL_f formulas

- a set $\mathcal{P}$ of propositional symbols
- closed under
  - boolean connectives, Negation($\neg$), And($\wedge$), Or($\vee$)
  - temporal operators, Next(X), Until(U), Eventually(F), Release(R), Globally(G)

Note: Same syntax, *different* semantics

- **LTL**: GFp means that p holds infinitely often
- **LTL_f**: GFp means that p holds at the last point

LTL_f Synthesis

Given:

LTL_f formula ϕ over $\mathcal{P}$

- Input variables: $\mathcal{X}$
- Output variables: $\mathcal{Y}$

Obtain:

Strategy $g : (2^{\mathcal{X}})^* \rightarrow 2^{\mathcal{Y}}$, a function from past history of inputs (in $2^{\mathcal{X}}$) to outputs

- At each instant, return an output (in $2^{\mathcal{Y}}$)
- **Goal**: trace [inputs & outputs] satisfy ϕ

Classical Solution to LTL_f Synthesis

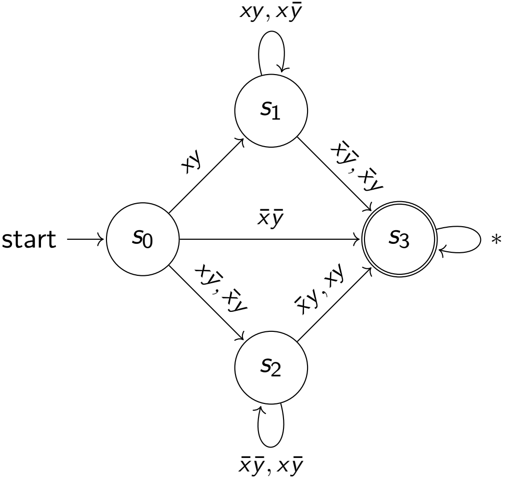
[De Giacomo & Vardi, IJCAI'15]: Reduction to DFA games

For an LTL_f formula ϕ over $\mathcal{P}$, traces(ϕ) $\subseteq (2^{\mathcal{P}})^*$.

- [De Giacomo & Vardi, IJCAI'13]: A conversion from an LTL_f formula ϕ to a DFA A_ϕ such that $L(A_\phi)$=traces(ϕ).
- [Kupferman & Vardi, 2001]: Conversion to DFA may incur a doubly exponential blow-up in the worst case.

Given an LTL_f formula ϕ over $\mathcal{P} = \mathcal{X} \cup \mathcal{Y}$, the corresponding DFA game is described as a tuple $\mathcal{G} = (2^{\mathcal{X} \cup \mathcal{Y}}, S, s_0, \delta, F)$.

DFA Games: Example



- Environment: $\mathcal{X}$
- Controller: $\mathcal{Y}$
- Controller wins: accepting state reached

Controller winning strategy
$g : S \mapsto 2^{\mathcal{Y}}$

Solving DFA Games

DFA Games:

- Controller: Chooses outputs in $2^{\mathcal{Y}}$.
- Environment: Chooses inputs in $2^{\mathcal{X}}$.
- Goal: Construct a word accepted by DFA.

DFA Game Solving

- **Realizability:** Does the controller have a winning strategy?
- **Synthesis:** Compute a winning strategy for the controller.

Our Contribution

- Reduction of LTL_f synthesis to LTL synthesis
  1. Translate LTL_f formula to an equi-realizable LTL formula
  2. Use LTL synthesis tool
- First direct tool for LTL_f synthesis
  1. A symbolic framework of LTL_f synthesis
  2. Leveraging Boolean-synthesis techniques from CAD
- Symbolic framework outperforms conversion to LTL synthesis

Symbolic LTL_f Synthesis

We introduce a symbolic framework for LTL_f synthesis:

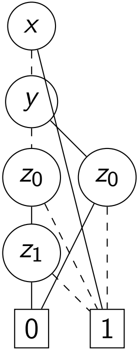
1. Construct DFA symbolically represented by BDDs using MONA [1] [Henriksen et al., TACAS'95]
2. Solve DFA game using BDD techniques
   - Compute fixpoint of winning states of the DFA
   - Synthesize winning strategy using a Boolean-synthesis procedure

[1] MONA: A tool for monadic second-order logic on words

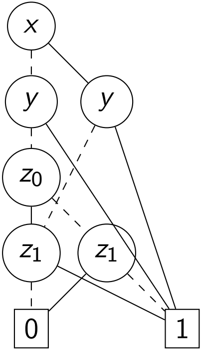
Symbolic DFA Representation

DFA can be symbolically encoded using BDDs.

- State variables
  $\mathcal{Z} = \{z_0, \ldots, z_k\}$, $k = \log n$
-
- k BDDs for transition function



BDD for z_0



BDD for z_1

Solving a DFA Game Symbolically

Perform the fixpoint computation using BDDs:

- Set of winning states represented by BDD W over $\mathcal{Z}$
- Set of pairs (winning state, winning output) represented by BDD T over $\mathcal{Z} \cup \mathcal{Y}$

Realizability:

- Check if $W(Z_0) = 1$

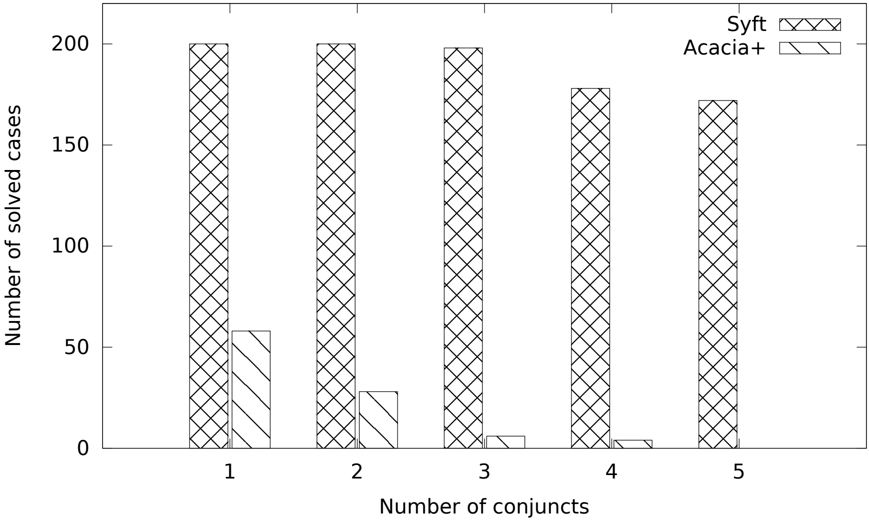
Synthesis:

- Construct strategy $\tau : 2^{\mathcal{Z}} \rightarrow 2^{\mathcal{Y}}$ such that if $W(Z) = 1$ then $T(Z, \tau(Z)) = 1$ (Boolean synthesis) [Fried, Tabajara & Vardi, CAV'16]

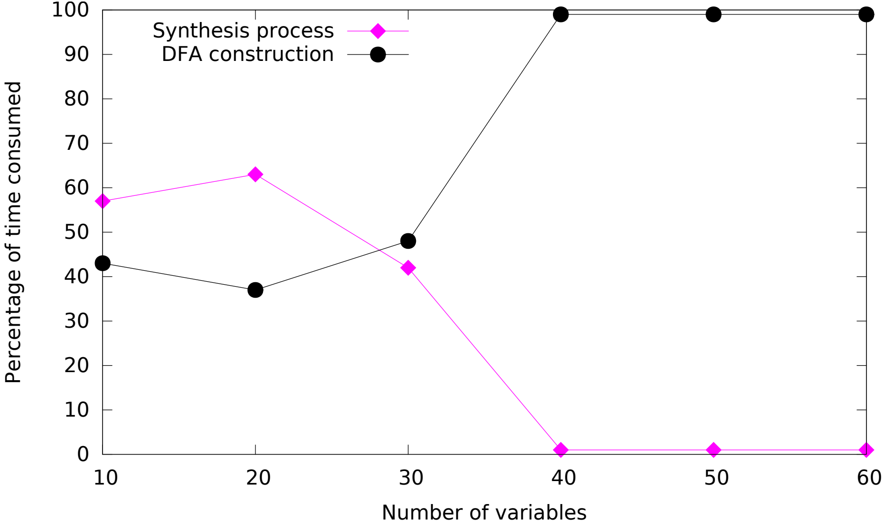
Experimental Evaluation

- Large synthetic dataset:
  - Random conjunctions from 100 base cases (industrial patterns)
  - Number of conjuncts: l
  - Number of variables: n
  - 200 LTL_f formulas for each (l, n)
- The symbolic approach is implemented in a tool **Syft**.
- LTL synthesis tool **Acacia+** [Bohy et al., CAV'12].

Symbolic Framework vs. Conversion to LTL synthesis



DFA Construction dominates Synthesis



Conclusions

- First implementation of a synthesis framework for LTL_f synthesis
- Experiments demonstrate advantages of symbolic approach
- Potential for further improvement in DFA construction