LTLf Synthesis Under Environment Specifications[1]

Antonio Di Stasio

Sapienza Universtity of Rome

ICTCS 2022

ERC Advanced Grant
WhiteMech:
White-box Self Programming Mechanisms

SAPIENZA
UNIVERSITÀ DI ROMA

[1] Joint work with Giuseppe De Giacomo, Lucas Tabajara, Moshe Vardi and Shufang Zhu

Given a specification φ over inputs I and outputs O, expressed in:

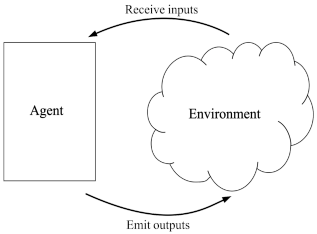
LTL (Pnueli 1977) or LTL_f (De Giacomo, Vardi 2013)

Syntax:

$$\varphi ::= a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi \mathcal{U} \varphi \mid \Diamond\varphi \mid \Box\varphi$$

Semantic:

A trace *trace* is an infinite (LTL) or finite (LTL_f) sequence over I and O. We write $trace \models \varphi$ to mean that τ satisfies φ.



Agent and Environment Strategies, and Traces

For an agent strategy $\sigma_{ag} : I^+ \to O$ and an environment strategy $\sigma_{env} : O^* \to I$

$$trace(\sigma_{ag}, \sigma_{env}) = (i_1 \cup o_1), (i_2 \cup o_2) \ldots \in 2^{I \cup O}$$

denotes the unique trace induced by both σ_{ag} and σ_{env}.

Synthesis Problem

Given an LTL/ LTL_f task *Goal* for the agent

Find agent strategy σ_{ag} such that $\forall \sigma_{env}.trace(\sigma_{ag}, \sigma_{env}) \models Goal$

SAPIENZA UNIVERSITÀ DI ROMA

Algorithm for LTL synthesis

Given LTL formula φ

1. Compute corresponding Nondeterministic Büchi Automaton (NBA) (exponential)
2. Determinize NBA into Deterministic Parity Automaton (DPA) (exp in states, poly in priorities (nested-fixpoints))
3. Synthesize winning strategy for Parity Game (poly in states, exp in priorities)

Complexity

LTL synthesis is 2EXPTIME-complete

Tools

- **Spot**[a]: a platform for LTL and ω-automata manipulation.
- **Strix**[b]: So far, the best tool for solving LTL synthesis.

[a] https://spot.lrde.epita.fr/
[b] https://strix.model.in.tum.de/

SAPIENZA UNIVERSITÀ DI ROMA

Algorithm for LTL_f synthesis

Given LTL_f formula φ

1. Compute corresponding Nondeterministic Finite Automaton (NFA) (exponential)
2. Determinize NFA to Deterministic Finite Automaton (DFA) (exponential)
3. Synthesize winning strategy for DFA game (linear)

Complexity

LTL_f synthesis is 2EXPTIME-complete

Tools

- **ltlf2dfa**[a]: a tool for traslating LTL_f into DFA.
- **Syft**, **Lysa**, **Lydia**,**Cynthia**[b]...

[a] http://ltlf2dfa.diag.uniroma1.it.
[b] https://github.com/whitemech

SAPIENZA UNIVERSITÀ DI ROMA

Domain

- Planning consider the agent acting in a **(nondeterministic) domain**
- The domain is a **model of how the world** (i.e. the environment) works
- That is, it is a **specification of the possible environment strategies**

$$[[Dom]] = \{\sigma_{env} | \sigma_{env} \text{ compliant with } Dom\}$$

Planning in nondeterministic domains

Given an task *Goal* for the agent, and a domain *Dom* modeling the environment

$$\text{Find agent behavior } \sigma_{ag} \text{ such that } \forall \sigma_{env} \in [[Dom]].trace(\sigma_{ag}\sigma_{env}) \models Goal$$

SAPIENZA UNIVERSITÀ DI ROMA

Agent Tasks terminate: LTL_f

- Because it is the agent that is planning/reasoning.
- If the task would not terminate, the agent would be stuck into doing the same task forever.
- We want to focus on autonomous intelligent agents that (1) get a task, (2) reason/plan autonomously to solve it, (3) execute the plan, (4) get another task, and so on.

We cannot require the environment specification to be in LTL_f: we must use LTL

The environment has to react to the agent's moves anyway, independently of whether the agent accomplishes its task (in a finite number of steps or not)

SAPIENZA UNIVERSITÀ DI ROMA

Which kinds of environment assumptions can the agent make?

- Nondeterministic planning domains;
- Forms of fairness ($\Box\Diamond\phi$) and stability ($\Diamond\Box\phi$)[ZhuDeGiacomoPuVardiAAAI2020];
- Safety properties [DeGiacomoDiStasioPerelliZhuKR2021];
- GR(1) formulas [DeGiacomoDiStasioTabajaraVardiZhuIJCAI2021];
- ..

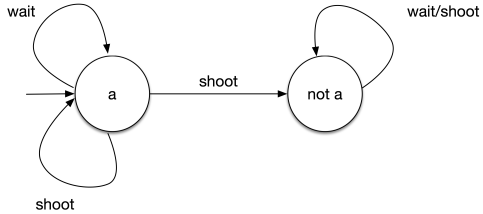
Environments Specifications as LTL formulas

A natural generalization is to consider general environment specifications expressed as arbitrary LTL formulas [DeGiacomoDiStasioVardiZhuKR2020].

For example let the environment specification be formed by $Env_1 \wedge Env_2$ where:

Env_1 is the LTL formula expressing the dynamics of the environment (as a planning domain):

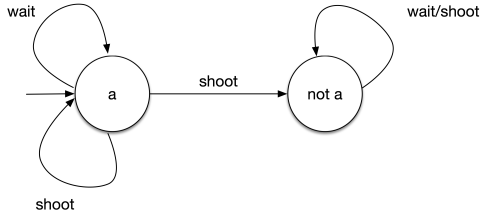
$$\Box(alive \rightarrow \bigcirc(wait \rightarrow alive))$$
$$\Box(alive \rightarrow \bigcirc(shoot \rightarrow (alive \vee \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(wait \rightarrow \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(shoot \rightarrow \neg alive))$$
$$\Box((wait \wedge shoot) \wedge (wait \rightarrow \neg shoot))$$



For example let the environment specification be formed by $Env_1 \wedge Env_2$ where:

Env_1 is the LTL formula expressing the dynamics of the environment (as a planning domain):

$$\Box(alive \rightarrow \bigcirc(wait \rightarrow alive))$$
$$\Box(alive \rightarrow \bigcirc(shoot \rightarrow (alive \vee \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(wait \rightarrow \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(shoot \rightarrow \neg alive))$$
$$\Box((wait \wedge shoot) \wedge (wait \rightarrow \neg shoot))$$



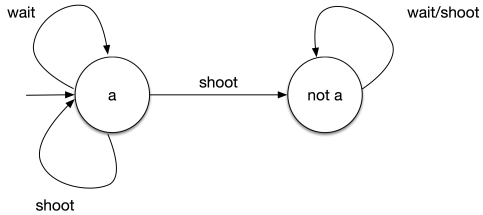
Env_2 is the LTL formula expressing some fairness over the nondeterministic effect a, e.g.,

$$\Box\Diamond shoot \rightarrow \Diamond\neg a$$

For example let the environment specification be formed by $Env_1 \wedge Env_2$ where:

Env_1 is the LTL formula expressing the dynamics of the environment (as a planning domain):

$$\Box(alive \rightarrow \bigcirc(wait \rightarrow alive))$$
$$\Box(alive \rightarrow \bigcirc(shoot \rightarrow (alive \vee \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(wait \rightarrow \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(shoot \rightarrow \neg alive))$$
$$\Box((wait \wedge shoot) \wedge (wait \rightarrow \neg shoot))$$



Env_2 is the LTL formula expressing some fairness over the nondeterministic effect a, e.g.,

$$\Box\Diamond shoot \rightarrow \Diamond\neg a$$

Let $Goal$ be an LTL_f formula which expresses an agent task, e.g.,

$$\Diamond\neg a$$

SAPIENZA UNIVERSITÀ DI ROMA

Environment specifications in LTL/LTL_f

Let *Env* be an LTL/LTL_f formula over I and O.

$$[[Env]] = \{\sigma_{env} | \forall \sigma_{ag}.trace(\sigma_{ag}, \sigma_{env}) \models Env\}$$

i.e *Env* denotes all environment strategies that play according to the specification whatever is the agent strategy.

Synthesis under environment specifications in LTL/LTL_f

Given an LTL_f task *Goal* for the agent, and an LTL/LTL_f environment specification *Env*:

Find agent strategy σ_{ag} such that $\forall \sigma_{env} \in [[Env]].trace(\sigma_{ag}, \sigma_{env}) \models Goal$

SAPIENZA UNIVERSITÀ DI ROMA

Consistent environment specifications

Is any $\mathrm{LTL}/\mathrm{LTL}_f$ formula a valid environment specification? No, *Env* needs to be "consistent"!:

$$[[Env]] \neq \emptyset \qquad \text{i.e. } \exists \sigma_e . \forall \sigma_{ag} . trace(\sigma_{ag}, \sigma_e) \models Env$$

SAPIENZA UNIVERSITÀ DI ROMA

Environment Specifications

Let Env be an LTL/LTL$_f$ formula over $I \cup O$.

$$[[Env]] = \{\sigma_{env} | \sigma_{env} \text{ satisfies } Env \text{ whatever is the agent strategy}\}$$

Synthesis under environment specifications in LTL/LTL$_f$

Given an LTL/ LTL$_f$ task $Goal$ for the agent, and an LTL/LTL$_f$ environment specification Env:

Find agent strategy σ_{ag} such that $\forall \sigma_{env} \in [[Env]].trace(\sigma_{ag}, \sigma_{env}) \models Goal$

Theorem [AminofDeGiacomoMuranoRubinICAPS2019]

To find agent strategy realizing $Goal$ under the environment specification Env, we can use standard LTL/LTL$_f$ synthesis for

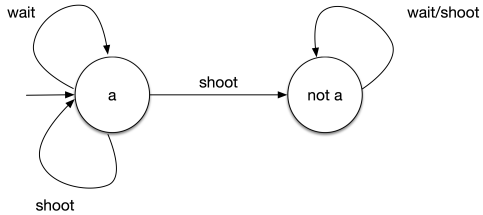
$$Env \rightarrow Goal$$

LTL_f Synthesis Under LTL Environment Specifications

For example let the assumption be formed by $Env_1 \wedge Env_2$ where:

Env_1 is the LTL formula expressing the dynamics of the environment (as a planning domain):

$$\Box(alive \rightarrow \bigcirc(wait \rightarrow alive))$$
$$\Box(alive \rightarrow \bigcirc(shoot \rightarrow (alive \vee \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(wait \rightarrow \neg alive))$$
$$\Box(\neg alive \rightarrow \bigcirc(shoot \rightarrow \neg alive))$$
$$\Box((wait \wedge shoot) \wedge (wait \rightarrow \neg shoot))$$



Env_2 is the LTL formula expressing some fairness over nondeterministic effects, e.g.,

$$\Box\Diamond shoot \rightarrow \Diamond\neg a$$

Let $Goal$ be an LTL_f formula which expresses an agent task, e.g.,

$$\Diamond\neg a$$

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

... but we can exploit the simplicity of dealing with LTL_f given:

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

... but we can exploit the simplicity of dealing with LTL_f given:

- Env_1: LTL

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

... but we can exploit the simplicity of dealing with LTL_f given:

- Env_1: ~~LTL~~ $\rightarrow$ LTL_f

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

... but we can exploit the simplicity of dealing with LTL_f given:

- Env_1: ~~LTL~~ $\rightarrow$ LTL_f
- Env_2: LTL

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$Env_1 \wedge Env_2 \rightarrow Goal$$

Naive Solution

Translate to LTL and then do standard LTL synthesis for $Env_1 \wedge Env_2 \rightarrow Goal$.

... but we can exploit the simplicity of dealing with LTL_f given:

- Env_1: ~~LTL~~ $\rightarrow$ LTL_f
- Env_2: LTL
- $Goal$: LTL_f

Separating LTL_f environment specificaitons

$$(Env_1 \wedge Env_2 \rightarrow Goal) \iff (Env_2 \rightarrow Env_1 \rightarrow Goal) \iff (Env_2 \rightarrow \neg Env_1 \vee Goal)$$

where $Goal' = \neg Env_1 \vee Goal$ is expressed in LTL_f and Env_2 in LTL.

Separating LTL_f environment specificaitons

$$(Env_1 \wedge Env_2 \rightarrow Goal) \iff (Env_2 \rightarrow Env_1 \rightarrow Goal) \iff (Env_2 \rightarrow \neg Env_1 \vee Goal)$$

where $Goal' = \neg Env_1 \vee Goal$ is expressed in LTL_f and Env_2 in LTL.

Problem

Solve the synthesis problem for

$$Env_2 \rightarrow Goal'$$

SAPIENZA UNIVERSITÀ DI ROMA

Separating LTL_f environment specificaitons

$$(Env_1 \wedge Env_2 \rightarrow Goal) \iff (Env_2 \rightarrow Env_1 \rightarrow Goal) \iff (Env_2 \rightarrow \neg Env_1 \vee Goal)$$

where $Goal' = \neg Env_1 \vee Goal$ is expressed in LTL_f and Env_2 in LTL.

Problem

Solve the synthesis problem for

$$Env_2 \rightarrow Goal'$$

How can we exploit that $Goal'$ is LTL_f?

SAPIENZA UNIVERSITÀ DI ROMA

Separating LTL_f environment specificaitons

$$(Env_1 \wedge Env_2 \rightarrow Goal) \iff (Env_2 \rightarrow Env_1 \rightarrow Goal) \iff (Env_2 \rightarrow \neg Env_1 \vee Goal)$$

where $Goal' = \neg Env_1 \vee Goal$ is expressed in LTL_f and Env_2 in LTL.

Problem

Solve the synthesis problem for

$$Env_2 \rightarrow Goal'$$

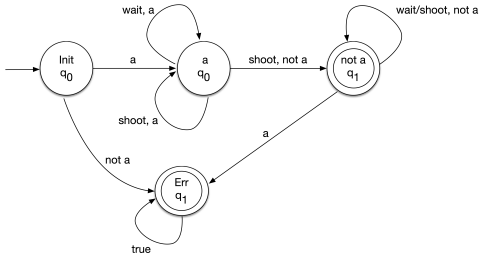
How can we exploit that $Goal'$ is LTL_f?

Two-stage technique!

SAPIENZA UNIVERSITÀ DI ROMA

1° Stage

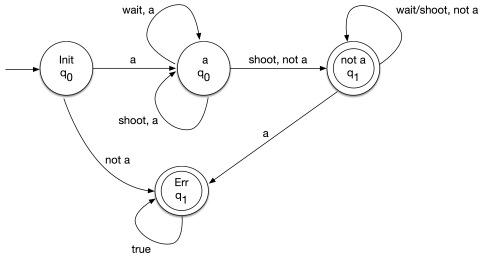
- Compute the corresponding DFA $\mathcal{A}$ of $Goal' = \neg Env_1 \vee Goal$.



SAPIENZA UNIVERSITÀ DI ROMA

1° Stage

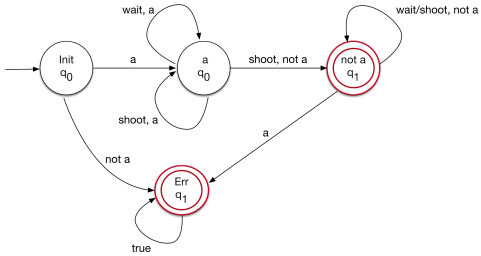
- Compute the corresponding DFA $\mathcal{A}$ of $Goal' = \neg Env_1 \vee Goal$.
- Solve the reachability game for the agent over $\mathcal{A}$.



SAPIENZA UNIVERSITÀ DI ROMA

1° Stage

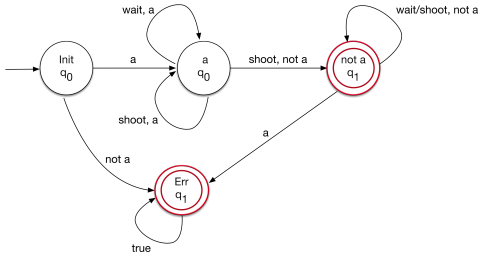
- Compute the corresponding DFA $\mathcal{A}$ of $Goal' = \neg Env_1 \vee Goal$.
- Solve the reachability game for the agent over $\mathcal{A}$.
- Check whether the initial state is winning for the agent.



SAPIENZA UNIVERSITÀ DI ROMA

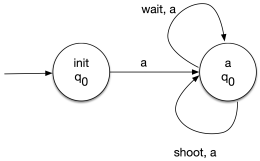
1° Stage

- Compute the corresponding DFA $\mathcal{A}$ of $Goal' = \neg Env_1 \vee Goal$.
- Solve the reachability game for the agent over $\mathcal{A}$.
- Check whether the initial state is winning for the agent.
- If the initial state is not winning go to Stage 2, otherwise return the agent winning strategy.



2° Stage

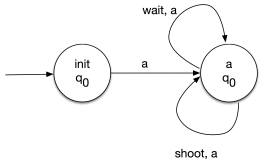
- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.

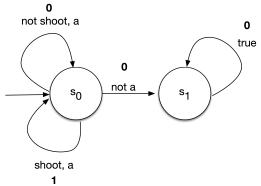


SAPIENZA UNIVERSITÀ DI ROMA

2° Stage

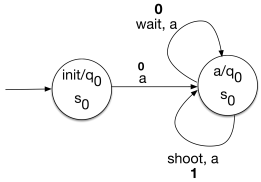
- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.
- Compute the corresponding DPA $\mathcal{B}$ of Env_2.





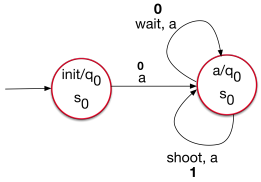
2° Stage

- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.
- Compute the corresponding DPA $\mathcal{B}$ of Env_2.
- Do the cartesian product between $\mathcal{A}'$ and $\mathcal{B}$.



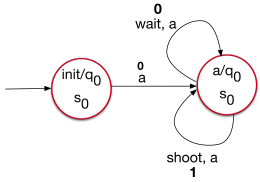
2° Stage

- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.
- Compute the corresponding DPA $\mathcal{B}$ of Env_2.
- Do the cartesian product between $\mathcal{A}'$ and $\mathcal{B}$.
- Solve the parity game for the environment over $\mathcal{A}' \times \mathcal{B}$.



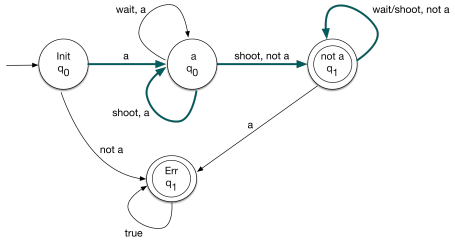
2° Stage

- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.
- Compute the corresponding DPA $\mathcal{B}$ of Env_2.
- Do the cartesian product between $\mathcal{A}'$ and $\mathcal{B}$.
- Solve the parity game for the environment over $\mathcal{A}' \times \mathcal{B}$.
- Check if the initial state is winning for the agent; if not return "Unrealizable".



2° Stage

- Remove from $\mathcal{A}$ the agent winning set of Stage 1, say $\mathcal{A}'$.
- Compute the corresponding DPA $\mathcal{B}$ of Env_2.
- Do the cartesian product between $\mathcal{A}'$ and $\mathcal{B}$.
- Solve the parity game for the environment over $\mathcal{A}' \times \mathcal{B}$.
- Check if the initial state is winning for the agent; if not return "Unrealizable".
- Return the agent winning strategy by combing the agent winning strategies in Stage 1 and 2.



LTL_f Synthesis Under GR(1) Environment Specifications

- GR(1) formula, generalization of fairness
  - Powerful expressiveness on environment assumption
- LTL$_f$ formula, natural for finite-horizon task
  - Strong agent task specification
- Simple solution
  - No detour to LTL synthesis
  - Games on finite-word automata

- GR(1) game $\mathcal{G} = (\mathcal{A}, \psi)$
  - Game arena $\mathcal{A}$ encoding moves
  - Winning condition GR(1) formula ψ, powerful notion of fairness

$$\psi = \bigwedge_{i=1}^{m} \Box\Diamond \mathcal{J}_i \rightarrow \bigwedge_{j=1}^{n} \Box\Diamond \mathcal{K}_j$$

- **Complexity**: simple game solving, quadratic time

SAPIENZA UNIVERSITÀ DI ROMA

Problem

Solve the synthesis problem for

$$\varphi^e_{GR(1)} \rightarrow \varphi^a_{task}$$

Key Idea

- Agent goal: $\neg\varphi^e_{GR(1)} \vee \varphi^a_{task}$ — Environment goal $\varphi^e_{GR(1)} \wedge \neg\varphi^a_{task}$.
- Build the corresponding DFA $\mathcal{A}_{\varphi^a_{task}}$ of φ^a_{task}, and take its complement $\overline{\mathcal{A}_{\varphi^a_{task}}}$.
- Define a $\mathrm{GR}(1)$ game whose game arena is $\overline{\mathcal{A}_{\varphi^a_{task}}}$ and winning condition $\varphi^e_{GR(1)}$.
- Solve the $\mathrm{GR}(1)$ game for the agent, i.e., solve the dual of the $\mathrm{GR}(1)$ game.

LTL_f Synthesis Under Safety Environment Specifications

SAPIENZA UNIVERSITÀ DI ROMA

Definition

A safety property is a property which specifies that some (bad) behavior will never occur.

Examples:

- "always at most one process is in its critical section"
- "money can only be withdrawn once a correct PIN has been provided"

Important property

Any infinite trace violating the property has a finite prefix that is "bad";

- ... two processes are in the critical section ...

Usually: $\Box\neg\ldots$

SAPIENZA UNIVERSITÀ DI ROMA

Nonderministic Stretegy

A *non-deterministic* strategy for the agent is defined as a function $\Pi : (2^{\mathcal{X}})^* \to 2^{2^{\mathcal{Y}}}$.
The set of deterministic strategies induced by a non-deterministic strategy Π is the set

$$[[\Pi]] = \{\sigma : (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}} | \sigma(h) \in \Pi(h), \text{ for } h \in (2^{\mathcal{X}})^*\}.$$

Maximally Permissive Strategy

A non-deterministic strategy Π is at least as permissive as Π' if $[[\Pi']] \subseteq [[\Pi]]$. A non-deterministic strategy Π is a maximally permissive strategy if $[[\Pi']] \subseteq [[\Pi]]$, for every non-deterministic strategy Π'.

Theorem

Safety specifications admit a nondeterministic strategy Π that captures the maximally permissive strategy.[2]

[2] Bernet et al.: Permissive strategies: from parity games to safety games

Problem

Solve the synthesis problem for

$$\varphi^e_{\mathcal{S}} \rightarrow \varphi^a_{task}$$

Key Idea

1. Compute the *deterministic safety automaton* $\mathcal{D}$ of $\varphi_{\mathcal{S}}$ (**no Büchi determinization!**)
2. Solve the safety game for the environment over $\mathcal{D}$;
3. Construct the *maximally permissive strategy* $\mathcal{T}$ (transducer);
4. Build the corresponding DFA $\mathcal{A}_{\varphi^a_{task}}$ of φ^a_{task};
5. Do the product of $\mathcal{T}$ and $\mathcal{A}_{\varphi^a_{task}}$;
6. Solve the reachability game for the agent over it, and return a strategy, if exists.

Almost all the techniques are based on the following reduction

Theorem [Aminof et al. ICAPS 2019]

To find agent strategy realizing *Goal* under the environment specification *Env*, we can use standard synthesis for

$$Env \rightarrow Goal$$

Possible directions

- In case of safety environment specifications we can directly solve the problem without reduction to the implication.
- What about the other environment specifications? Fairness, $\mathrm{GR}(1)$, LTL, ...