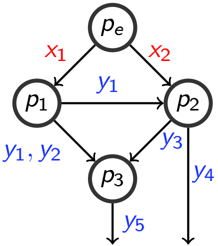
Distributed Synthesis for LTL Fragments

Krishnendu Chatterjee, Thomas A. Henzinger, **Jan Otop**,
Andreas Pavlogiannis

IST AUSTRIA
Institute of Science and Technology

21 October 2013

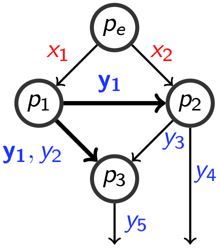
An **architecture** is a directed graph describing topology of the system.



- **Communication** is done through variables V.
- Communication is **instantaneous**.
- Process p has **I(p)**, **O(p)**, its **input** and **output** variables.
- Process p behaves according to its **local strategy** $\sigma_p : \left(2^{I(p)}\right)^* \to 2^{O(p)}$.
- p_e is the **environment**.

- Local strategies give **the collective strategy** $\sigma : \left(2^{O(p_e)}\right)^* \to 2^{V \setminus O(p_e)}$.
- Reactive system as a function: The **execution** of σ on $\pi = a_1 a_2 \ldots \in \left(2^{O(p_e)}\right)^\omega$ is $\Gamma^\sigma(\pi) = \sigma(a_1)\sigma(a_1 a_2) \ldots \in \left(2^{V \setminus O(p_e)}\right)^\omega$

An **architecture** is a directed graph describing topology of the system.



- **Communication** is done through variables V.
- Communication is **instantaneous**.
- Process p has **I(p)**, **O(p)**, its **input** and **output** variables.
- Process p behaves according to its **local strategy** $\sigma_p : \left(2^{I(p)}\right)^* \to 2^{O(p)}$.
- p_e is the **environment**.

- Local strategies give **the collective strategy** $\sigma : \left(2^{O(p_e)}\right)^* \to 2^{V \setminus O(p_e)}$.
- Reactive system as a function: The **execution** of σ on $\pi = a_1 a_2 \ldots \in \left(2^{O(p_e)}\right)^\omega$ is $\Gamma^\sigma(\pi) = \sigma(a_1)\sigma(a_1 a_2) \ldots \in \left(2^{V \setminus O(p_e)}\right)^\omega$

Realizability

- A **computation** of σ is the convolution of the environment output π and the execution of σ, i.e., for $\pi = a_1 a_2 \ldots$ and $\Gamma^\sigma(\pi) = b_1 b_2 \ldots$ the computation is: $\pi \otimes \Gamma^\sigma(\pi) = (a_1, b_2)(a_2, b_2) \ldots \in \left(2^V\right)^\omega$

Satisfaction

A collective strategy σ **satisfies** an LTL specification φ iff its every computation satisfies φ, i.e., for every $\pi \in \left(2^{O(p_e)}\right)^\omega$, $\pi \otimes \Gamma^\sigma(\pi) \models \varphi$.

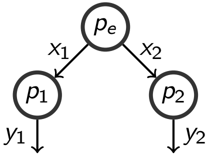
Realizability

Given an architecture $\mathcal{A}$ and an LTL specification φ, decide whether there exist local strategies σ_p for all processes p, that generate the collective strategy σ that satisfy φ.

- If so, synthesize them.

Example

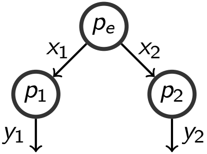
Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \land \Box(x_2 \implies \Diamond y_2) \land \Box\neg(y_1 \land y_2)$ in the architecture:

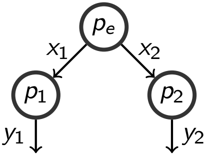


It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \land y_1)) \land (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \land y_2)) \land \Box\neg(y_1 \land y_2)$.

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

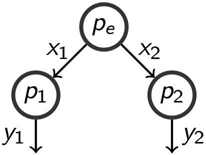
The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \wedge y_1)) \wedge (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \wedge y_2)) \wedge \Box\neg(y_1 \wedge y_2)$.

Suppose it is realizable.

x_1	1	0	1	0	1	0	1
y_1							
x_2	0	1	0	1	0	1	0
y_2							

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

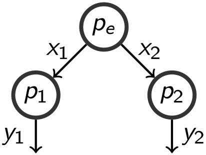
The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \wedge y_1)) \wedge (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \wedge y_2)) \wedge \Box\neg(y_1 \wedge y_2)$.

Suppose it is realizable.

x_1	1	0	1	0	1	0	1
y_1	1	0	0	0	1	0	1
x_2	0	1	0	1	0	1	0
y_2	0	1	0	1	0	1	0

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

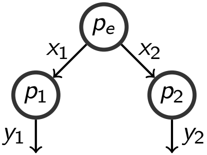
The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \wedge y_1)) \wedge (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \wedge y_2)) \wedge \Box\neg(y_1 \wedge y_2)$.

Suppose it is realizable.

x_1	1	0	1	0	1	0	1
y_1	1	0	0	0	1	0	1
x_2							
y_2							

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \land \Box(x_2 \implies \Diamond y_2) \land \Box\neg(y_1 \land y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

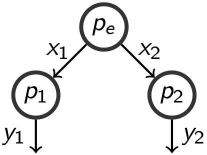
The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \land y_1)) \land (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \land y_2)) \land \Box\neg(y_1 \land y_2)$.

Suppose it is realizable.

x_1	1	0	1	0	1	0	1
y_1	1	0	0	0	1	0	1
	↓	↓	↓	↓	↓	↓	↓
x_2							
y_2							

Example

Consider a specification
$\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

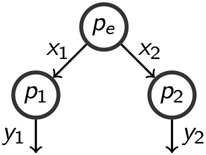
The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \wedge y_1)) \wedge (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \wedge y_2)) \wedge \Box\neg(y_1 \wedge y_2)$.

Suppose it is realizable.

x_1	1	0	1	0	1	0	1
y_1	1	0	0	0	1	0	1
	↓	↓	↓	↓	↓	↓	↓
x_2	1	0	0	0	1	0	1
y_2							

Example

Consider a specification $\varphi_1 \equiv \Box(x_1 \implies \Diamond y_1) \wedge \Box(x_2 \implies \Diamond y_2) \wedge \Box\neg(y_1 \wedge y_2)$ in the architecture:



It is realized by σ_1, σ_2 such that:
$\sigma_1(w) = \{y_1\}$ if $|w|$ is even and $\emptyset$ otherwise, and
$\sigma_2(w) = \{y_2\}$ if $|w|$ is odd and $\emptyset$ otherwise.

The following specification is not realizable
$\varphi_2 \equiv (\Box\Diamond x_1 \implies \Box\Diamond(x_1 \wedge y_1)) \wedge (\Box\Diamond x_2 \implies \Box\Diamond(x_2 \wedge y_2)) \wedge \Box\neg(y_1 \wedge y_2)$.

Suppose it is realizable.

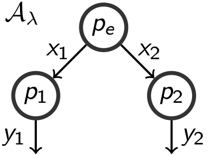
x_1	1	0	1	0	1	0	1
y_1	1	0	0	0	1	0	1
x_2	1	0	0	0	1	0	1
y_2							

x_2 holds infinitely often, but only when y_1 holds!

Undecidability

Theorem (Pnueli, Rosner)

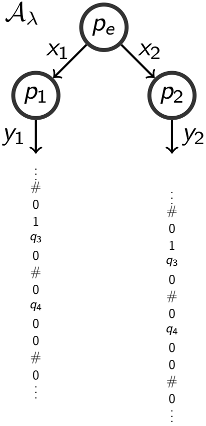
Realizability of LTL specifications on the following architecture $\mathcal{A}_\lambda$ is undecidable.



For every Turing Machine M, there is a specification τ_M, that forces p_1, p_2 to output the sequence of consecutive configurations of $M(\epsilon)$ terminated by the final configuration.

Theorem (Pnueli, Rosner)

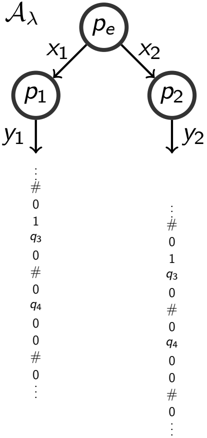
Realizability of LTL specifications on the following architecture $\mathcal{A}_\lambda$ is undecidable.



For every Turing Machine M, there is a specification τ_M, that forces p_1, p_2 to output the sequence of consecutive configurations of $M(\epsilon)$ terminated by the final configuration.

Theorem (Pnueli, Rosner)

Realizability of LTL specifications on the following architecture $\mathcal{A}_\lambda$ is undecidable.

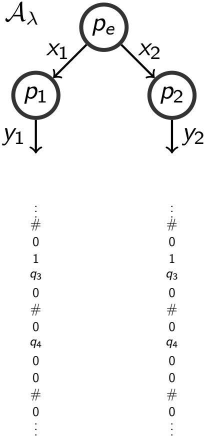


For every Turing Machine M, there is a specification τ_M, that forces p_1, p_2 to output the sequence of consecutive configurations of $M(\epsilon)$ terminated by the final configuration.

Undecidability

Theorem (Pnueli, Rosner)

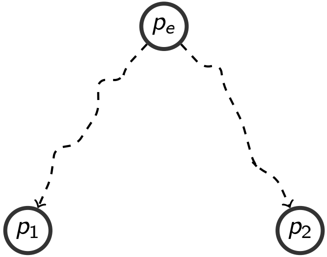
Realizability of LTL specifications on the following architecture $\mathcal{A}_\lambda$ is undecidable.



For every Turing Machine M, there is a specification τ_M, that forces p_1, p_2 to output the sequence of consecutive configurations of $M(\epsilon)$ terminated by the final configuration.

Parametric on the Architecture

- For which classes of architectures is realizability decidable?
- Complete characterization base on the *information fork* criterion.
- Processes p_1, p_2 form an information fork in architecture $\mathcal{A}$ if there exist paths $p_e \rightsquigarrow p_i$ in $\mathcal{A}$ such that do not traverse edges in $I(p_{-i})$.



Theorem(Finkbeiner,Schewe)

Every architecture either:

- Has an information fork (undecidable).
- Can be reduced to a pipeline (decidable).

Our approach

- LTL formulae that appear in the undecidability proof are complicated.

Question

What are the LTL fragments for which the realizability problem is decidable?

- That question can be approached from two directions:
  - Prove that realizability is undecidable in weak LTL fragments.
  - Find LTL fragments for which the realizability problem is decidable.

Reachability specifications $\mathsf{LTL}_\Diamond$

$\mathsf{LTL}_\Diamond$

- $\psi \in \mathsf{LTL}_1$ iff it is a Boolean combination of P and $\mathcal{X}P$, where P is propositional. (only non-nested $\mathcal{X}$)
- $\varphi \in \mathsf{LTL}_\Diamond$ iff $\varphi \equiv Q \to \Diamond\psi$, where $\psi \in \mathsf{LTL}_1$ and Q is propositional.

Theorem

The realizability of specifications from $\mathsf{LTL}_\Diamond$ in architectures containing information fork is undecidable.

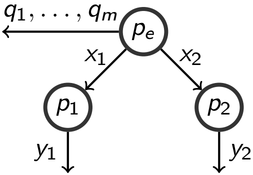
Reachability specifications $\text{LTL}_{\Diamond}$

$\text{LTL}_{\Diamond}$

- $\psi \in \text{LTL}_1$ iff it is a Boolean combination of P and $\mathcal{X}P$, where P is propositional. (only non-nested $\mathcal{X}$)
- $\varphi \in \text{LTL}_{\Diamond}$ iff $\varphi \equiv Q \rightarrow \Diamond\psi$, where $\psi \in \text{LTL}_1$ and Q is propositional.

Theorem

The realizability of specifications from $\text{LTL}_{\Diamond}$ in architectures containing information fork is undecidable.



- τ_M is a (variant of) formula that forces p_1, p_2 to output a computation of a TM M.
- A safety automaton A_{safe} recognizes $\mathcal{L}_{\tau_M}$.
- Specification $\gamma \in \text{LTL}_{\Diamond}$ states that eventually
  - p_e (does not) simulate A_{safe} with $q_1, \ldots, q_k$,
  - p_1 outputs the final configuration.

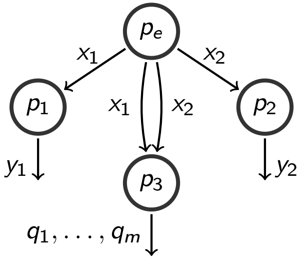
Safety specifications $\mathsf{LTL}_\Box$ over Overlapping Inputs

$\mathsf{LTL}_\Box$

- $\psi \in \mathsf{LTL}_1$ iff it is a Boolean combination of P and $\mathcal{X}P$, where P is propositional. (only non-nested $\mathcal{X}$)
- $\varphi \in \mathsf{LTL}_\Box$ iff $\varphi \equiv Q \wedge \Box\psi$, where $\psi \in \mathsf{LTL}_1$ and Q is propositional.

Theorem

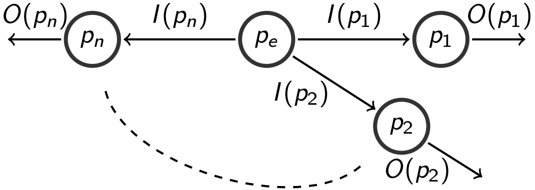
The realizability of specifications from $\mathsf{LTL}_\Box$ in an architecture $\mathcal{A}$ containing an information fork-meet is undecidable.



- The proof is as for $\mathsf{LTL}_\Diamond$, but p_3 simulates A_{safe} instead of p_e, i.e.:
- A safety automaton A_{safe} recognizes $\mathcal{L}_{\tau_{\boldsymbol{M}}}$.
- Specification $\gamma \in \mathrm{LTL}_\Box$ ensures that p_3 simulates A_{safe}.

Safety specifications over Disjoint Inputs

Consider a class of **star architectures with disjoint inputs:**



Lemma

A formula $\phi = Q \wedge \Box\psi$ is realizable iff it is realizable by strategies with double exponential memory.

Sufficiently long plays can be repeated.

Theorem

Realizability of $\mathrm{LTL}_{\Box}$ specifications on star architectures with disjoint inputs is in EXPSPACE.

Fragments of LTL without $\mathcal{X}$

LTL_{AG}

$\varphi \in \mathsf{LTL}_{AG}$ if for propositional formulae P, Q, R_i, F_i, φ is of the form

$$\varphi = \Box P \rightarrow \Box Q \wedge \bigwedge_i \Box\Diamond R_i \wedge \bigwedge_i \Diamond F_i$$

Theorem

Realizability of LTL_{AG} specifications is NEXPTIME-complete.

Fragments of LTL without $\mathcal{X}$

LTL_{AG}

$\varphi \in \text{LTL}_{AG}$ if for propositional formulae P, Q, R_i, F_i, φ is of the form

$$\varphi = \Box P \to \Box Q \wedge \bigwedge_i \Box\Diamond R_i \wedge \bigwedge_i \Diamond F_i$$

Theorem

Realizability of LTL_{AG} specifications is NEXPTIME-complete.

- $\varphi \in \text{LTL}_{AG}$ is realizable iff every formula $\Box(P \to Q \wedge R_i)$ and every $\Box(P \to Q \wedge F_i)$ are realizable.
- $\Box Q$ is realizable iff it is realizable by memoryless strategies.
- Realizability of LTL_{AG} is in NEXPTIME.

Fragments of LTL without $\mathcal{X}$

LTL_{AG}

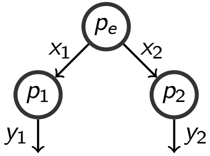
$\varphi \in \mathrm{LTL}_{AG}$ if for propositional formulae P, Q, R_i, F_i, φ is of the form

$$\varphi = \Box P \rightarrow \Box Q \wedge \bigwedge_i \Box\Diamond R_i \wedge \bigwedge_i \Diamond F_i$$

Theorem

Realizability of LTL_{AG} specifications is NEXPTIME-complete.

Dependency Quantified Boolean Formulas(DQBF) are propositional formulae with Henkin quatifiers.



$$\forall x_1 \forall x_2 \exists y_1(x_1) \exists y_2(x_2).Q(x_1, x_2, y_1, y_2)$$

- Validity of DQBF is NEXPTIME-complete.
- DQBF reduces to realizability of LTL_{AG}

Conclusions

Our contributions:

- Distributed synthesis is undecidable, even restricted to simple LTL fragments: $\text{LTL}_{\Diamond}$, $\text{LTL}_{\Box}$.
- $\text{LTL}_{\Box}$ is decidable in NEXPSPACE on the class of star architecutes with disjoint inputs.
- LTL_{AG} is NEXPTIME-complete.
- LTL_{AG} reduces to DQBF and vice versa.

Thank you!