Reasoning about Knowledge and Strategies

Bastien Maubert and Aniello Murano
Program synthesis

Basic idea:

“Program synthesis is the task to automatically construct a program that satisfies a given high-level specification.”

We are interested in programs that:

- **Interact** with an environment
- **May run** forever

Example: operating systems, controllers in power plants...

Specification language: **LTL**

Propositional logic +

- **Xφ**: “φ holds at next step”
- **φUψ**: “φ will hold until ψ holds”
- **Gφ**: “φ always holds”
- **Fφ**: “φ eventually holds”
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment

- Environment chooses valuations for $I$
- System chooses valuations for $O$
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment

- Environment chooses valuations for $I$
- System chooses valuations for $O$
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

**Game between system and environment**
- Environment chooses valuations for $I$
- System chooses valuations for $O$

$i_0$

$o_0$
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

**Game between system and environment**
- Environment chooses valuations for $I$
- System chooses valuations for $O$

\[
i_0 \quad i_1 \quad o_0
\]
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

**Game between system and environment**

- Environment chooses valuations for $I$
- System chooses valuations for $O$

\[
i_0 \quad i_1 \\
o_0 \quad o_1
\]
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment
- Environment chooses valuations for $I$
- System chooses valuations for $O$

\[
i_0 \quad i_1 \quad i_2 \\
o_0 \quad o_1
\]
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment

- Environment chooses valuations for $I$
- System chooses valuations for $O$

\[
\begin{array}{c c c}
  i_0 & i_1 & i_2 \\
  o_0 & o_1 & o_2
\end{array}
\]
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment
- Environment chooses valuations for $I$
- System chooses valuations for $O$

\[
i_0 \quad i_1 \quad i_2 \quad \ldots \\
o_0 \quad o_1 \quad o_2 \quad \ldots
\]
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment

- Environment chooses valuations for $I$
- System chooses valuations for $O$

- $i_0$  $i_1$  $i_2$  $\ldots$
- $o_0$  $o_1$  $o_2$  $\ldots$

LTL synthesis problem

Given a specification $\varphi \in \text{LTL}$ over $I \cup O$, synthesize a strategy $\sigma : (2^I)^* \to 2^O$ such that all resulting behaviours satisfy $\varphi$. 
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

**Game between system and environment**
- Environment chooses valuations for $I$
- System chooses valuations for $O$

$$i_0 \quad i_1 \quad i_2 \quad \ldots$$
$$o_0 \quad o_1 \quad o_2 \quad \ldots$$

**LTL synthesis problem**

- Given a specification $\varphi \in \text{LTL}$ over $I \cup O$, synthesize a strategy $\sigma : (2^I)^* \to 2^O$ such that all resulting behaviours satisfy $\varphi$.

Synthesize a finite representation of this infinite object.
LTL synthesis (Pnueli and Rosner, 1989)

- $I$: input variables,
- $O$: output variables

Game between system and environment

- Environment chooses valuations for $I$
- System chooses valuations for $O$

$LTL$ synthesis problem

Given a specification $\varphi \in LTL$ over $I \cup O$, synthesize a strategy $\sigma : (2^I)^* \rightarrow 2^O$ such that all resulting behaviours satisfy $\varphi$.

Synthesize a finite representation of this infinite object.

What about synthesis of distributed systems?
Distributed synthesis

\begin{itemize}
\item \(p, q\) are atomic propositions
\item \(\circ, \bullet\) are actions
\end{itemize}

strategies \(\sigma : \text{Histories} \rightarrow \text{Actions}\)

Input: A concurrent game structure and a formula \(\varphi \in \text{LTL}\)
Output: A distributed strategy to enforce \(\varphi\)
Distributed synthesis

\[(\bullet, \circ, \bullet) \quad (\bullet, \circ, \bullet) \]
\[\rightarrow \quad p \quad \rightarrow \quad \]
\[\rightarrow \quad (\circ, \circ, \circ) \quad (\circ, \circ, \circ) \]
\[\rightarrow \quad (\bullet, \bullet, \circ) \quad (\bullet, \bullet, \circ) \]
\[\rightarrow \quad q \quad \rightarrow \quad \]
\[\rightarrow \quad (\circ, \bullet, \bullet) \quad (\bullet, \bullet, \bullet) \]

\[p, q \text{ are atomic propositions} \]
\[\circ, \bullet \text{ are actions} \]

strategies \(\sigma : \text{Histories} \rightarrow \text{Actions} \)

indistinguishability relations \(\sim_a\)

Input: A concurrent game structure and a formula \(\varphi \in \text{LTL} \)
Output: A distributed strategy to enforce \(\varphi \)
Distributed synthesis

$p, q$ are atomic propositions
$\circ, \bullet$ are actions
strategies $\sigma : \text{Histories} \rightarrow \text{Actions}$
indistinguishability relations $\sim_{\alpha}$

**Input:** A concurrent game structure and a formula $\varphi \in \text{LTL}$

**Output:** A distributed strategy to enforce $\varphi$
Imperfect information

1 Strategies must be consistent with players’ information

Constraint on strategies:

If \( h \sim_a h' \), then \( \sigma_a(h) = \sigma_a(h') \).

2 Makes epistemic reasoning meaningful and useful

Example: opacity

A system is opaque for property \( P \) if a spy never knows whether the current execution is in \( P \).

Classic definition:

\[
\forall h, \exists h' \text{ s.t. } h \sim_{spy} h' \text{ and } h' \notin P
\]

With epistemic temporal logic:

\[
\Box \neg K_{spy} P
\]
Imperfect information

1. Strategies must be consistent with players’ information

**Constraint on strategies:**

If \( h \sim_a h' \), then \( \sigma_a(h) = \sigma_a(h') \).

2. Makes epistemic reasoning meaningful and useful

**Example: opacity**

A system is *opaque* for property \( P \) if a spy never knows whether the current execution is in \( P \).

**Classic definition:**

\[
\forall h, \exists h' \text{ s.t. } h \sim_{\text{spy}} h' \text{ and } h' \notin P
\]

With epistemic temporal logic:

\[
\exists \sigma(c, \sigma) \mathbf{G} \neg K_{\text{spy}} P
\]
Semantics of knowledge when reasoning about strategies

Yellow subtree: controller’s strategy

$\sim_{\text{spy}}$: spy’s indistinguishability relation

Two possible semantics:
- spy ignores controller’s strategy
  $\rightarrow K_{\text{spy}}P$ does not hold
- spy knows controller’s strategy
  $\rightarrow K_{\text{spy}}P$ holds
Semantics of knowledge when reasoning about strategies

Yellow subtree: controller’s strategy

\( \sim_{\text{spy}} \): spy’s indistinguishability relation

Two possible semantics:
- spy ignores controller’s strategy
  \[ \rightarrow K_{\text{spy}}P \text{ does not hold} \]
- spy knows controller’s strategy
  \[ \rightarrow K_{\text{spy}}P \text{ holds} \]
Semantics of knowledge when reasoning about strategies

Yellow subtree: controller’s strategy
$\sim_{\text{spy}}$: spy’s indistinguishability relation

Two possible semantics:
- spy ignores controller’s strategy
  $\rightarrow K_{\text{spy}}P$ does not hold
- spy knows controller’s strategy
  $\rightarrow K_{\text{spy}}P$ holds
Semantics of knowledge when reasoning about strategies

Yellow subtree: controller’s strategy
\( \sim_{\text{spy}} \): spy’s indistinguishability relation

Two possible semantics:

- spy ignores controller’s strategy
  \[ \rightarrow K_{\text{spy}}P \text{ does not hold} \]

- spy knows controller’s strategy
  \[ \rightarrow K_{\text{spy}}P \text{ holds} \]
In the literature

Both semantics have been used, but implicitly.

**Informed semantics:**
Distributed synthesis from epistemic temporal specifications
- van der Meyden and Vardi, 1998
- van der Meyden and Wilke, 2005

**Uninformed semantics:**
All epistemic extensions of ATL and SL (that we know of)

One paper talks about this issue: Puchala, 2010
In the litterature

Both semantics have been used, but implicitly.

**Informed semantics:**
Distributed synthesis from epistemic temporal specifications
- van der Meyden and Vardi, 1998
- van der Meyden and Wilke, 2005

**Uninformed semantics:**
All epistemic extensions of ATL and SL (that we know of)

One paper talks about this issue: Puchala, 2010

What is known about distributed synthesis?
Peterson and Reif (1979), Pnueli and Rosner (1990)

Distributed synthesis for reachability objective is undecidable.

Two known ways of retrieving decidability for temporal objectives:

1. Public actions
2. Hierarchical information

For epistemic temporal objectives and

- informed semantics:
  - decidable for public actions
  - undecidable for hierarchical information

  [van der Meyden and Wilke, 2005]

- uninformed semantics:
  - decidable for public actions
  - decidable for hierarchical information

  [Belardinelli et al., 2017] [Puchala, 2010]
Peterson and Reif (1979), Pnueli and Rosner (1990)

Distributed synthesis for reachability objective is undecidable.

Two known ways of retrieving decidability for temporal objectives:

1. Public actions
2. Hierarchical information

For epistemic temporal objectives and

- informed semantics:
  - decidable for public actions
  - undecidable for hierarchical information

  [van der Meyden and Wilke, 2005]

- uninformed semantics:
  - decidable for public actions
  - decidable for hierarchical information

  [Belardinelli et al., 2017] [Puchala, 2010]
SL with imperfect information and knowledge

<table>
<thead>
<tr>
<th>SL</th>
<th>(Chatterjee et al. 2010, Mogavero et al. 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTL +</td>
<td>⊨∃σφ \hspace{1cm} “there exists a strategy σ s.t. φ”</td>
</tr>
<tr>
<td></td>
<td>⊨(a,σ)φ \hspace{1cm} “when player a plays strategy σ, φ”</td>
</tr>
</tbody>
</table>
SL with imperfect information and knowledge

<table>
<thead>
<tr>
<th>SL_{ii}</th>
<th>(Berthon et al. 2017)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTL +</td>
<td></td>
</tr>
<tr>
<td>\exists^o \sigma \varphi</td>
<td>“there exists a strategy \sigma with observational power \sigma \text{ s.t. } \varphi”</td>
</tr>
<tr>
<td>(a, \sigma) \varphi</td>
<td>“when player \alpha plays strategy \sigma, \varphi”</td>
</tr>
</tbody>
</table>
SL with imperfect information and knowledge

\[ \exists^o \sigma \varphi \]

“there exists a strategy \( \sigma \) with observational power \( o \) s.t. \( \varphi \)”

\[ (a, \sigma) \varphi \]

“when player \( a \) plays strategy \( \sigma \), \( \varphi \)”

\[ K_a \varphi \]

“player \( a \) knows that \( \varphi \)”

\[ A \varphi \]

“in all outcomes, \( \varphi \)”
What can ESL express?

- **Distributed synthesis:**

  \[ \exists^{o_1} x_1 \ldots \exists^{o_n} x_n (a_1, x_1) \ldots (a_n, x_n) \forall^{o_e} y (e, y) \psi \]

- **Existence of Nash equilibria:**

  \[ \exists^{o_a} x \exists^{o_b} y \exists^{o_c} z (a, x)(b, y)(c, z) \]
  \[ \land_{d \in \{a, b, c\}} \exists^{o_d} x' (d, x') \text{ Win}_d \rightarrow \text{ Win}_d \]

- **Players changing observation:**

  \[ \exists^{o_1} x_1 (a, x_1) \text{ AF} \exists^{o_2} x_2 (a, x_2) \text{ AF} \text{ Win}_a \]

  "First I find my glasses, then I play for real."
Hierarchical instances

An ESL formula $\Phi$ is hierarchical if:

- innermost strategies observe better than outermost ones
- epistemic subformulas do not talk about current strategies

Considering the uninformed semantics of knowledge:

Theorem

Model-checking hierarchical instances of ESL is decidable.
Corollaries:

On systems with **hierarchical information**, for **epistemic temporal specifications** with **uninformed semantics**, we can solve

- distributed synthesis,
- module checking,
- synthesis of Nash equilibria,
- rational synthesis,
- ...
Interested? Come to Napoli!
Rational distributed synthesis

Rational synthesis

Fisman et al. (2010), Condurache et al. (2016), Kupferman et al. (2016)

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \)
- System made of one component \( a \)
- LTL specification \( \psi_g \)

\[
\Phi_{\text{c-RS}} := \exists x \exists y_1 \ldots \exists y_m (a, x)(e, y) \varphi_{\text{NE}} \land A \psi_g
\]
\[
\Phi_{\text{nc-RS}} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{\text{NE}} \rightarrow A \psi_g
\]

where

\[
\varphi_{\text{NE}} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A \psi_i) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- individual LTL goals \( \psi_i \)
- System made of one component \( a \)
- LTL specification \( \psi_g \)

\[
\Phi_{c-RS} := \exists x \exists y_1 \ldots \exists y_m (a, x)(e, y) \varphi_{NE} \land A\psi_g \\
\Phi_{nc-RS} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{NE} \rightarrow A\psi_g \\
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A\psi_i) \rightarrow A\psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o^{e_i}_i \)
- System made of one component \( a \)
- LTL specification \( \psi_g \)

\[
\Phi_{\text{c-RS}} := \exists x \exists y_1 \ldots \exists y_m (a, x)(e, y) \varphi_{\text{NE}} \land A\psi_g \\
\Phi_{\text{nc-RS}} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{\text{NE}} \rightarrow A\psi_g
\]

where

\[
\varphi_{\text{NE}} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A\psi_i) \rightarrow A\psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \)

\[
\Phi_{c-RS} := \exists x \exists y_1 \ldots \exists y_m (a, x)(e, y) \varphi_{NE} \land A \psi_g
\]
\[
\Phi_{nc-RS} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{NE} \rightarrow A \psi_g
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ \left( \exists y'_i (e_i, y'_i) A \psi_i \right) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\begin{align*}
\Phi_{c-RS} & := \exists x \exists y_1 \ldots \exists y_m (a, x)(e, y) \varphi_{NE} \land A \psi_g \\
\Phi_{nc-RS} & := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{NE} \rightarrow A \psi_g
\end{align*}
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A \psi_i) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c\text{-RS}} := \exists x \exists^{o^e_1} y_1 \ldots \exists^{o^e_m} y_m (a, x)(e, y) \varphi_{\text{NE}} \land A \psi_g
\]
\[
\Phi_{nc\text{-RS}} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{\text{NE}} \rightarrow A \psi_g
\]

where

\[
\varphi_{\text{NE}} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A \psi_i) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c-RS} := \exists^0 x_1 \ldots \exists^0 x_n \exists^0 y_1 \ldots \exists^0 y_m (a, x)(e, y) \varphi_{NE} \land A\psi_g
\]

\[
\Phi_{nc-RS} := \exists x \forall y_1 \ldots \forall y_m (a, x)(e, y) \varphi_{NE} \rightarrow A\psi_g
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in \mathbb{[m]}} \left[ (\exists y'_i (e_i, y'_i) A\psi_i) \rightarrow A\psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c-RS} := \exists^0 x_1 \ldots \exists^0 x_n \exists^0 y_1 \ldots \exists^0 y_m (a, x)(e, y) \varphi_{NE} \land A\psi_g
\]

\[
\Phi_{nc-RS} := \exists x \forall^0 y_1 \ldots \forall^0 y_m (a, x)(e, y) \varphi_{NE} \rightarrow A\psi_g
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ \left( \exists y'_i (e_i, y'_i) A\psi_i \right) \rightarrow A\psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o^e_i \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c-RS} := \exists^1 x_1 \ldots \exists^1 x_n \exists^1 y_1 \ldots \exists^1 y_m (a, x)(e, y) \varphi_{NE} \land A \psi_g
\]

\[
\Phi_{nc-RS} := \exists^1 x_1 \ldots \exists^1 x_n \forall^1 y_1 \ldots \forall^1 y_m (a, x)(e, y) \varphi_{NE} \rightarrow A \psi_g
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ (\exists y'_i (e_i, y'_i) A \psi_i) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o_i^e \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c-RS} := \exists^{o_1} x_1 \ldots \exists^{o_n} x_n \exists^{o_1} y_1 \ldots \exists^{o_1} y_m (a, x)(e, y) \varphi_{NE} \land A \psi_g
\]

\[
\Phi_{nc-RS} := \exists^{o_1} x_1 \ldots \exists^{o_n} x_n \forall^{o_1} y_1 \ldots \forall^{o_1} y_m (a, x)(e, y) \varphi_{NE} \rightarrow A \psi_g
\]

where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ \left( \exists^{o_i} y'_i (e_i, y'_i) A \psi_i \right) \rightarrow A \psi_i \right]
\]
Rational distributed synthesis

- Environment made of several components \( \{e_1, \ldots, e_m\} \)
- Individual LTL goals \( \psi_i \) and observations \( o_i^e \)
- System made of several components \( \{a_1, \ldots, a_n\} \)
- LTL specification \( \psi_g \) and observations \( o_i \)

\[
\Phi_{c-RS} := \exists^0 x_1 \ldots \exists^0 x_n \exists^0 e_1 y_1 \ldots \exists^0 e_m y_m (a, x)(e, y) \varphi_{NE} \land A \psi_g
\]

\[
\Phi_{nc-RS} := \exists^0 x_1 \ldots \exists^0 x_n \forall^0 e_1 y_1 \ldots \forall^0 e_m y_m (a, x)(e, y) \varphi_{NE} \rightarrow A \psi_g
\]

Where

\[
\varphi_{NE} := \bigwedge_{i \in [m]} \left[ \left( \exists^{op} y'_i (e_i, y'_i) A \psi_i \right) \rightarrow A \psi_i \right]
\]