Towards Fully Observable Non-deterministic Planning as Assumption-based Synthesis

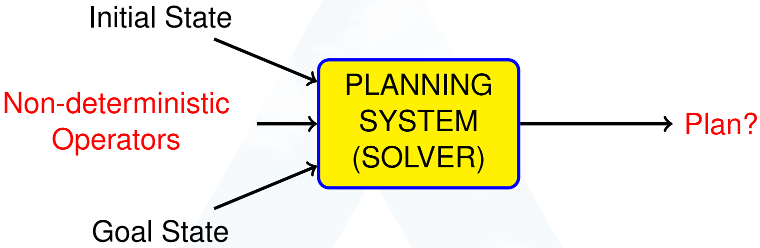
Sebastian Sardina[1] Nicolas D'Ippolito[2]

[1]School of Computer Science and IT
RMIT University
Melbourne, AUSTRALIA
sebastian.sardina@rmit.edu.au

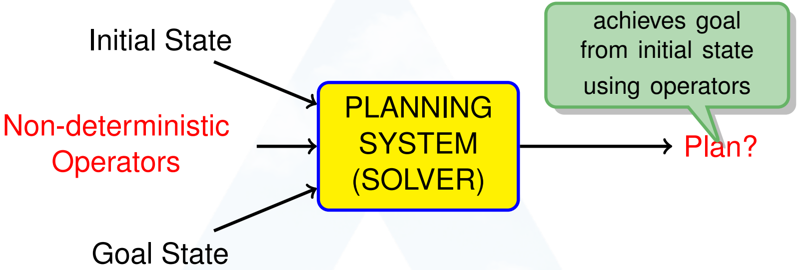
[2]Departamento de Computación, FCEN
Universidad de Buenos Aires
Buenos Aires, ARGENTINA
ndippolito@dc.uba.ar

IJCAI'15
July 25, 2015
Buenos Aires, Argentina

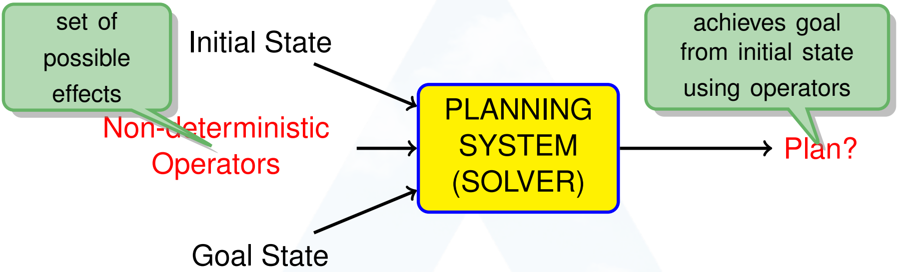
Fully Observable Non-deterministic (FOND) Planning



Fully Observable Non-deterministic (FOND) Planning

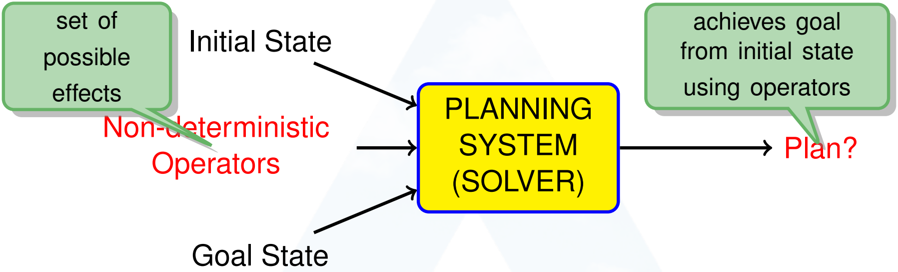


Fully Observable Non-deterministic (FOND) Planning



$$\texttt{Pick}(x) = \langle \neg\textsc{Holding}(x) \wedge \neg\textsc{Heavy}(x), \underline{\textsc{Holding}(x) \mid \textsc{Broken}(x)} \rangle$$

Fully Observable Non-deterministic (FOND) Planning



$\texttt{Pick}(x) = \langle \neg\textsc{Holding}(x) \wedge \neg\textsc{Heavy}(x), \underline{\textsc{Holding}(x) \mid \textsc{Broken}(x)} \rangle$

Questions / Contributions

1. What is an adequate plan given non-deterministic actions?
2. Where do these plans work? Which environments?
3. How is this problem related to reactive synthesis?

Solution Concepts for FOND Planning

Weak Plan

At least one execution of the plan reaches the goal.

Too optimistic...

Strong Plan

All executions are guarantee to reach the goal.

Too pessimistic, almost no solutions...

Solution Concepts for FOND Planning

Weak Plan

At least one execution of the plan reaches the goal.

Too optimistic...

Strong Plan

All executions are guarantee to reach the goal.

Too pessimistic, almost no solutions...

Strong Cyclic Plan

The plan always has the possibility to reach the goal.

Good middle-ground; good for "trying"

Strong Cyclic Plans

Definition ([Daniele *et al.* 2000, Pistore and Traverso 2001])

A policy π is a *strong cyclic plan solution* for a FOND problem $\mathcal{P}$ iff

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\mathsf{EF}\phi_{\mathsf{goal}} \mathsf{W} \phi_{\mathsf{goal}}).$$

- $\mathcal{K}_{\mathcal{P}}^{\pi}$: transition system implicit when running π in planning prob. $\mathcal{P}$

Strong Cyclic Plans

Definition ([Daniele *et al.* 2000, Pistore and Traverso 2001])

A policy π is a *strong cyclic plan solution* for a FOND problem $\mathcal{P}$ iff

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\mathsf{EF}\phi_{\text{goal}}\mathsf{W}\phi_{\text{goal}}).$$

- $\mathcal{K}_{\mathcal{P}}^{\pi}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: in every execution (of π over $\mathcal{P}$)
  - $(\cdots \mathsf{W}\phi_{\text{goal}})$: until ϕ_{goal} becomes true, it is always holds that
    - $\mathsf{E}(\cdots)$: there exists an execution
      - $\mathsf{F}(\phi_{\text{goal}})$: where ϕ_{goal} is eventually true

Strong Cyclic Plans

Definition ([Daniele *et al.* 2000, Pistore and Traverso 2001])

A policy π is a *strong cyclic plan solution* for a FOND problem $\mathcal{P}$ iff

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\mathsf{EF}\phi_{\mathsf{goal}}\mathsf{W}\phi_{\mathsf{goal}}).$$

- $\mathcal{K}_{\mathcal{P}}^{\pi}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: in every execution (of π over $\mathcal{P}$)

  $(\cdots \mathsf{W}\phi_{\mathsf{goal}})$: until ϕ_{goal} becomes true, it is always holds that

  $\mathsf{E}(\cdots)$: there exists an execution

  $\mathsf{F}(\phi_{\mathsf{goal}})$: where ϕ_{goal} is eventually true

✓ We know what exactly a strong cyclic plan is.

✓ We have ways to compute them (FOND planners).

??? But, when are these plans adequate? When do they do the job?

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}}).$$

- $\mathcal{K}_{\mathcal{P}}^{\pi}$: transition system implicit when running π in planning prob. $\mathcal{P}$

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}}).$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: for every execution (of π over $\mathcal{P}$),

  $\Phi_{\mathcal{P}} \rightarrow \cdots$: **if the execution satisfies** $\Phi_{\mathcal{P}}$, then ...

  $\mathsf{F}\phi_{\mathsf{goal}}$: the goal is eventually reached

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}}).$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: for every execution (of π over $\mathcal{P}$),
  - $\Phi_{\mathcal{P}} \rightarrow \cdots$: **if the execution satisfies** $\Phi_{\mathcal{P}}$, then ...
    - $\mathsf{F}\phi_{\mathsf{goal}}$: the goal is eventually reached

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}}).$$

- $\mathcal{K}_{\mathcal{P}}^{\pi}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: for every execution (of π over $\mathcal{P}$),
  - $\Phi_{\mathcal{P}} \rightarrow \cdots$: **if the execution satisfies** $\Phi_{\mathcal{P}}$, then ...
    - $\mathsf{F}\phi_{\mathsf{goal}}$: the goal is eventually reached

Where do strong cyclic plans work?

What are we after?

execution is "fair"

We want a condition $\Phi_{\mathcal{P}}$ such that:

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}}).$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: for every execution (of π over $\mathcal{P}$),

  $\Phi_{\mathcal{P}} \rightarrow \cdots$: **if the execution satisfies** $\Phi_{\mathcal{P}}$, then ...

  $\mathsf{F}\phi_{\text{goal}}$: the goal is eventually reached

Where do strong cyclic plans work?

What are we after?

We want a condition $\Phi_{\mathcal{P}}$ such that:

execution is "fair"

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}}).$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$
- $\mathsf{A}(\cdots)$: for every execution (of π over $\mathcal{P}$),
  - $\Phi_{\mathcal{P}} \rightarrow \cdots$: **if the execution satisfies** $\Phi_{\mathcal{P}}$, then ...
    - $\mathsf{F}\phi_{\text{goal}}$: the goal is eventually reached

"If an action is executed infinitely many times, then every non-deterministic outcome will occur infinitely often."

Fairness Assumption

Example: flipping the coin

- Flipping the coin:
  $\texttt{flip} = \langle \text{HOLDING}, \underline{\neg\text{HOLDING} \wedge \text{HEADS} \mid \neg\text{HOLDING} \wedge \neg\text{HEADS}} \rangle$.
- Goal: HEADS
- Assumption: *"If the coin is flipped infinitely often, both heads and tails outcomes will occur infinitely often."*
  - In LTL: $\Phi_{\mathcal{P}} = \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\text{HEADS} \wedge \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\neg\text{HEADS}$

Fairness Assumption

Example: flipping the coin

- Flipping the coin:
  $\texttt{flip} = \langle \text{HOLDING}, \underline{\neg\text{HOLDING} \wedge \text{HEADS} \mid \neg\text{HOLDING} \wedge \neg\text{HEADS}} \rangle$.
- Goal: HEADS
- Assumption: *"If the coin is flipped infinitely often, both heads and tails outcomes will occur infinitely often."*
  - In LTL: $\Phi_{\mathcal{P}} = \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\text{HEADS} \wedge \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\neg\text{HEADS}$

Plan π

while $\neg$HEADS **do**
    pick;
    flip;
endWhile

Does $\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}))$?

Fairness Assumption

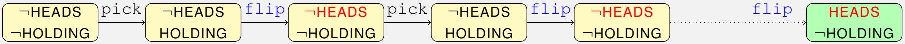
Example: flipping the coin

- Flipping the coin:
  $\texttt{flip} = \langle \text{HOLDING}, \underline{\neg\text{HOLDING} \wedge \text{HEADS} \mid \neg\text{HOLDING} \wedge \neg\text{HEADS}} \rangle$.
- Goal: HEADS
- Assumption: *"If the coin is flipped infinitely often, both heads and tails outcomes will occur infinitely often."*
  - In LTL: $\Phi_\mathcal{P} = \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\text{HEADS} \wedge \Box\Diamond\texttt{flip} \rightarrow \Box\Diamond\neg\text{HEADS}$

Plan π

while ¬HEADS **do**
  `pick;`
  `flip;`
endWhile

Does $\mathcal{K}^\pi_\mathcal{P} \models \text{A}(\Phi_\mathcal{P} \rightarrow \text{F}(\text{HEADS}))$?



A Problematic Example

- Two coins *A* and *B*.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\text{HEADS}_A \wedge \text{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\text{HEADS}_A \wedge \text{HEADS}_B)$ **do**
  `pick`;
  $\texttt{flip}_A$;
  $\texttt{flip}_B$;
endWhile

$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \wedge \text{HEADS}_B))$?

$\neg\text{HEADS}_A$
$\neg\text{HEADS}_B$
$\neg\text{HOLDING}_A$
$\neg\text{HOLDING}_B$

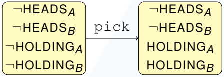
A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- <u>Goal</u>: $\textsf{HEADS}_A \wedge \textsf{HEADS}_B$.
- <u>Assumption:</u> *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\textsf{HEADS}_A \wedge \textsf{HEADS}_B)$ **do**
    $\texttt{pick}$;
    $\texttt{flip}_A$;
    $\texttt{flip}_B$;
endWhile

$$\mathcal{K}_{\mathcal{P}}^{\pi} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\textsf{HEADS}_A \wedge \textsf{HEADS}_B))?$$



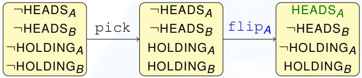
A Problematic Example

- Two coins *A* and *B*.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- <u>Goal</u>: $\text{HEADS}_A \land \text{HEADS}_B$.
- <u>Assumption:</u> *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\text{HEADS}_A \land \text{HEADS}_B)$ **do**
  $\texttt{pick}$;
  $\texttt{flip}_A$;
  $\texttt{flip}_B$;
endWhile

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \land \text{HEADS}_B))?$$



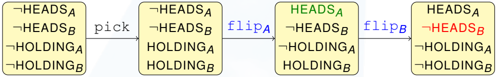
A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\text{HEADS}_A \wedge \text{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\text{HEADS}_A \wedge \text{HEADS}_B)$ **do**
    `pick`;
    $\texttt{flip}_A$;
    $\texttt{flip}_B$;
endWhile

$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \wedge \text{HEADS}_B))$?



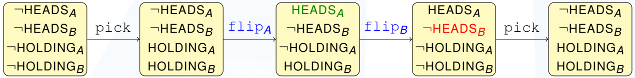
A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B)$ **do**
    $\texttt{pick}$;
    $\texttt{flip}_A$;
    $\texttt{flip}_B$;
endWhile

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B))?$$



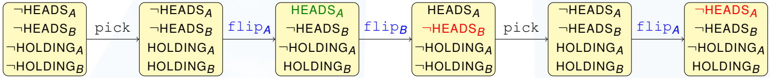
A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\text{HEADS}_A \land \text{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\text{HEADS}_A \land \text{HEADS}_B)$ **do**
  `pick`;
  $\texttt{flip}_A$;
  $\texttt{flip}_B$;
endWhile

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \land \text{HEADS}_B))?$$



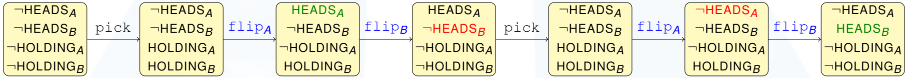
A Problematic Example

- Two coins *A* and *B*.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- <u>Goal</u>: $\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B$.
- <u>Assumption:</u> *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B)$ **do**
    `pick`;
    $\texttt{flip}_A$;
    $\texttt{flip}_B$;
endWhile

$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\mathsf{HEADS}_A \wedge \mathsf{HEADS}_B))$?



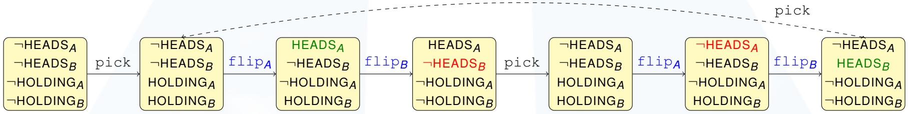
A Problematic Example

- Two coins A and B.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- <u>Goal</u>: $\text{HEADS}_A \wedge \text{HEADS}_B$.
- <u>Assumption:</u> *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

Plan π

while $\neg(\text{HEADS}_A \wedge \text{HEADS}_B)$ **do**
 `pick`;
 $\texttt{flip}_A$;
 $\texttt{flip}_B$;
endWhile

$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \wedge \text{HEADS}_B))$?



A Problematic Example

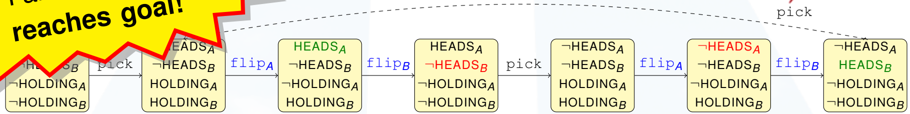
- Two coins *A* and *B*.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\text{HEADS}_A \wedge \text{HEADS}_B$.
- Assumption: *“If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often.”*

Plan π

while $\neg(\text{HEADS}_A \wedge \text{HEADS}_B)$ **do**
```
    pick;
    flipA;
    fli...;
en...
```

“Fair” but never reaches goal!

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models A(\Phi_{\mathcal{P}} \rightarrow F(\text{HEADS}_A \wedge \text{HEADS}_B))?$$



A Problematic Example

- Two coins *A* and *B*.
- Can `pick` up both at the same time, but can $\texttt{flip}_X$ one at a time.
- Goal: $\text{HEADS}_A \wedge \text{HEADS}_B$.
- Assumption: *"If a coin is flipped infinitely often, heads and tails outcomes will occur infinitely often."*

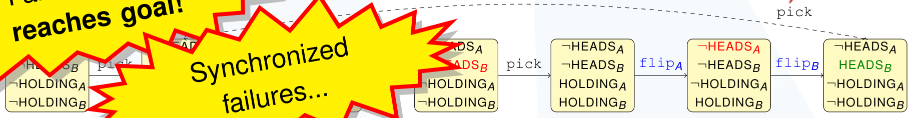
Plan π

while $\neg(\text{HEADS}_A \wedge \text{HEADS}_B)$ **do**
  pick;
  $\texttt{flip}_A$;
  fli…;
en…

$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi_{\mathcal{P}} \rightarrow \mathsf{F}(\text{HEADS}_A \wedge \text{HEADS}_B))$?

"Fair" but **never reaches goal!**

Synchronized failures...



State-strong Fairness

State Strong Fair

Given FOND problem $\mathcal{P}$ with goal ϕ_{goal}, we provide a condition $\Phi^*_{\mathcal{P}}$ s.t.:

Plan π is a strong cyclic plan for problem $\mathcal{P}$

iff

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}})$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$

State-strong Fairness

State Strong Fair

Given FOND problem $\mathcal{P}$ with goal ϕ_{goal}, we provide a condition $\Phi^*_{\mathcal{P}}$ s.t.:

Plan π is a strong cyclic plan for problem $\mathcal{P}$

iff

$$\mathcal{K}^{\pi}_{\mathcal{P}} \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}})$$

- $\mathcal{K}^{\pi}_{\mathcal{P}}$: transition system implicit when running π in planning prob. $\mathcal{P}$

Condition $\Phi^*_{\mathcal{P}}$:

1. Forces every non-deterministic action to be fair.
2. But also makes all those actions independent of each other.
   - No synchronized failures!
3. Corresponds to the semantic notion of fairness given in [Geffner and Bonet 2013] (in terms of runs/traces).
4. Is in CTL, as [Daniele *et al.* 2000, Pistore and Traverso 2001].

FOND as Reactive Synthesis

$\mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$ can be used for LTL synthesis!

Find controller π such that: $\mathcal{P} \times \pi \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$

FOND as Reactive Synthesis

$\mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$ can be used for LTL synthesis!

Find controller π such that: $\mathcal{P} \times \pi \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$

2EXPTIME!

FOND as Reactive Synthesis

$A(\Phi^*_{\mathcal{P}} \rightarrow F\phi_{goal})$ can be used for LTL synthesis!

Find controller π such that: $\mathcal{P} \times \pi \models A(\Phi^*_{\mathcal{P}} \rightarrow F\phi_{goal})$

2EXPTIME!

FOND-IE = FOND with Intended Effects

- Every ND-action has an intended effect.
  - Robot grabs an object: *holding the object*.
  - Open door: *door opened*.
  - Going to x: *be located at x*
  - Flip a coin: ???

FOND as Reactive Synthesis

$\mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$ can be used for LTL synthesis!

Find controller π such that: $\mathcal{P} \times \pi \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\mathsf{goal}})$

2EXPTIME!

FOND-IE = FOND with Intended Effects

- Every ND-action has an intended effect.
  - Robot grabs an object: *holding the object*.
  - Open door: *door opened*.
  - Going to x: *be located at x*
  - Flip a coin: ???
- Reduce FOND-IE to SGR(1) synthesis [D'Ippolito *et al.* 2011]
  - There is eventually a "window of opportunity" *with no failures*.
  - SGR(1) can be reduced to GR(1) synthesis [Piterman *et al.* 2006].
    - $(\Box\Diamond\phi_1 \wedge \cdots \wedge \Box\Diamond\phi_n) \rightarrow (\Box\Diamond\psi_1 \wedge \cdots \wedge \Box\Diamond\psi_n)$
  - Polynomial complexity.

FOND as Reactive Synthesis

$\mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}})$ can be used for LTL synthesis!

Find controller π such that: $\mathcal{P} \times \pi \models \mathsf{A}(\Phi^*_{\mathcal{P}} \rightarrow \mathsf{F}\phi_{\text{goal}})$

2EXPTIME!

FOND-IE = FOND with Intended Effects

- Every ND-action has an intended effect.
  - Robot grabs an object: *holding the object*.
  - Open door: *door opened*.
  - Going to x: *be located at x*
  - Flip a coin: ???
- Reduce FOND-IE to SGR(1) synthesis [D'Ippolito *et al.* 2011]
  - There is eventually a "window of opportunity" *with no failures*.
  - SGR(1) can be reduced to GR(1) synthesis [Piterman *et al.* 2006].
    - $(\Box\Diamond\phi_1 \wedge \cdots \wedge \Box\Diamond\phi_n) \rightarrow (\Box\Diamond\psi_1 \wedge \cdots \wedge \Box\Diamond\psi_n)$
  - Polynomial complexity.

on size of "game" TS!

Conclusions

1. Characterized the "fairness" assumption for FOND planning in the CTL framework.
   - Complements foundational [Daniele *et al.* 2000] work.
   - Syntactic counterpart of semantic defs. [Geffner and Bonet 2013].
   - "Naive" version does not work: allows for synchronized failures.
2. Linked FOND planning to assumption-based reactive synthesis.
   - General case is too expensive.
   - Special easier case: FOND + Intended Effects.
     - Exploits techniques from SE [D'Ippolito *et al.* 2011].

Future work

1. "Guess" intended effects: not always possible to specify them!
2. Other solution concepts (e.g., selective fairnes, k-tolerant plans).
3. More flexible fairness assumptions.
4. Hybrid reactive synthesis & planning techniques.

References I

Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Yaniv Sa'ar.
Synthesis of reactive(1) designs.
Journal of Computer and System Sciences, 78(3):911–938, 2012.

Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso.
Weak, strong, and strong cyclic planning via symbolic model checking.
Artificial Intelligence, 147(1-2):35–84, 2003.

M. Daniele, P. Traverso, and M. Vardi.
Strong cyclic planning revisited.
Recent Advances in AI Planning, pages 35–48, 2000.

Giuseppe De Giacomo, Fabio Patrizi, and Sebastian Sardina.
Generalized planning with loops under strong fairness constraints.
In *Proceedings of the Int. Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 351–361, 2010.

References II

Nicolás D'Ippolito, Víctor A. Braberman, Nir Piterman, and Sebastián Uchitel.
Synthesis of live behaviour models for fallible domains.
In *Proceedings of the International Conference on Software Engineering*, pages 211–220, 2011.

Carmel Domshlak.
Fault tolerant planning: Complexity and compilation.
In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2013.

Jicheng Fu, Vincent Ng, Farokh Bastani, and I-Ling Yen.
Simple and fast strong cyclic planning for fully-observable non-deterministic planning problems.
In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1949–1954, 2011.

Hector Geffner and Blai Bonet.
A Concise Introduction to Models and Methods for Automated Planning.
Morgan & Claypool Publishers, 2013.

References III

U. Kuter, S. Nau, D., E. Reisner, and P. Goldman, R.
Using classical planners to solve nondeterministic planning problems.
In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 190–197, 2008.

Christian Muise, Sheila A. McIlraith, and J. Christopher Beck.
Improved non-deterministic planning by exploiting state relevance.
In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 172–180, 2012.

Fabio Patrizi, Nir Lipovetzky, and Hector Geffner.
Fair LTL synthesis for non-deterministic systems using strong cyclic planners.
In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

Marco Pistore and Paolo Traverso.
Planning as model checking for extended goals in non-deterministic domains.
In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 479–486, 2001.

References IV

Nir Piterman, Amir Pnueli, and Yaniv Sa'ar.
Synthesis of reactive(1) designs.
In *Proceedings of the International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 3855 of *Lecture Notes in Computer Science (LNCS)*, pages 364–380. Springer, 2006.

Amir Pnueli and Roni Rosner.
On the synthesis of a reactive module.
In *Proceedings of the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 179–190, 1989.

Miguel Ramirez and Sebastian Sardina.
Directed fixed-point regression-based planning for non-deterministic domains.
In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 235–243, 2014.

Intelligent Autonomous Behavior

1. Behavior-based AI: set of independent simple reactive modules.
   - intelligent behavior emerges “implicitly”.
   - popular in robotics (since the ’80).
2. Agent-oriented programming: control specified by programmer.
   - BDI systems: JACK, JASON, 3APL, etc.
   - High-level languages: Golog-like languages, FLUX, etc.
3. Learning: learn how to act based on previous experience.
   - E.g., reinforcement learning.
4. Automated Planning: automatic synthesis of behavior from model.
   - Input: model of the world + initial state + goal to be achieved.
   - Output: plan or controller to achieve the goal in the world.

Intelligent Autonomous Behavior

4 Automated Planning: automatic synthesis of behavior from model.

- Input: model of the world + initial state + goal to be a
- Output: plan or controller to achieve the goal in the world.

Here!

Fully-Observable Non-Deterministic Planning

FOND Planning Problem

A **FOND planning problem** is a tuple $\mathcal{P} = \langle A, s_0, O, \phi_{\text{goal}} \rangle$:

- state variables (or atoms) A;
- initial state s_0;
- goal condition ϕ_{goal};
- operator set O made of operators $o = \langle Pre_o, Eff_o \rangle$:
  - Pre_o is the pre-condition of operator o;
  - $Eff_o = e_1 \mid \cdots \mid e_n$ are the *non-deterministic effects*
    - each e_i is a conjunction of literals.
    - effect e_i may be true after executing o.

Classical planning: each operator has exactly one effect; $Eff_o = e_o$