Opportunistic Synthesis in Reactive Games under Information Asymmetry

Abhishek N. Kulkarni

PhD Qualifying Exam Presentation

Advisor: Prof. Jie Fu

Committee: Prof. Carlo Pinciroli, Mitchell Colby

4 December 2019

Presentation Outline

I. Research Interest:

Reactive Synthesis (Formal Methods in Robotics)

II. Research Direction:

Opportunistic Synthesis

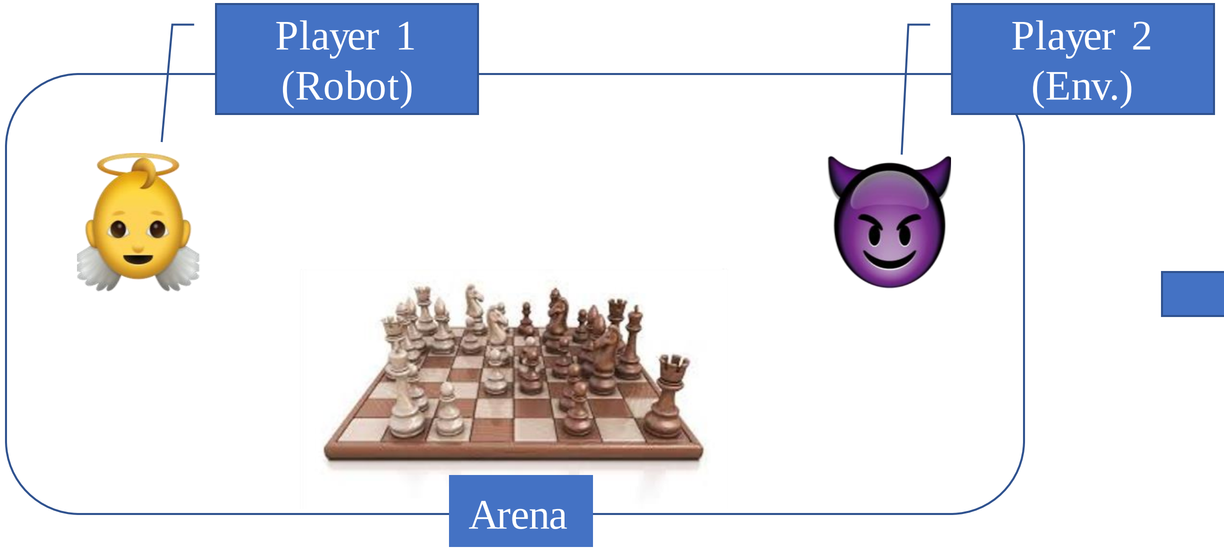
III. Future Directions:

Generalization of Opportunistic Synthesis

Section I: Reactive Synthesis

- What is reactive synthesis?
- What questions the state-of-the-art answers?
- Challenges in reactive synthesis?

Reactive Synthesis: Intuition



Reactive System

Interaction between a robot (P1) and its **dynamic** and **uncontrollable** environment (P2).

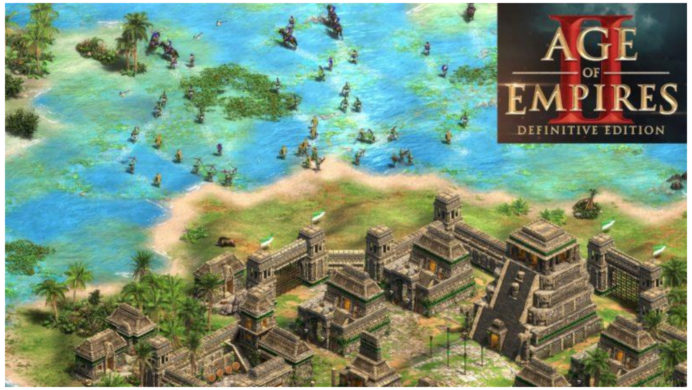
Reactive Synthesis

Automatic synthesis of strategy for P1, guaranteed to satisfy given **logical specification**.

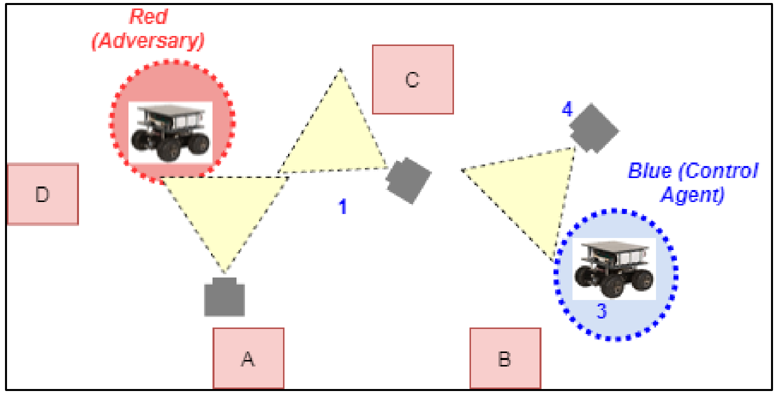
Strategic Applications

Cyber-Security

Human-Robot Interactions



Computer Game AI

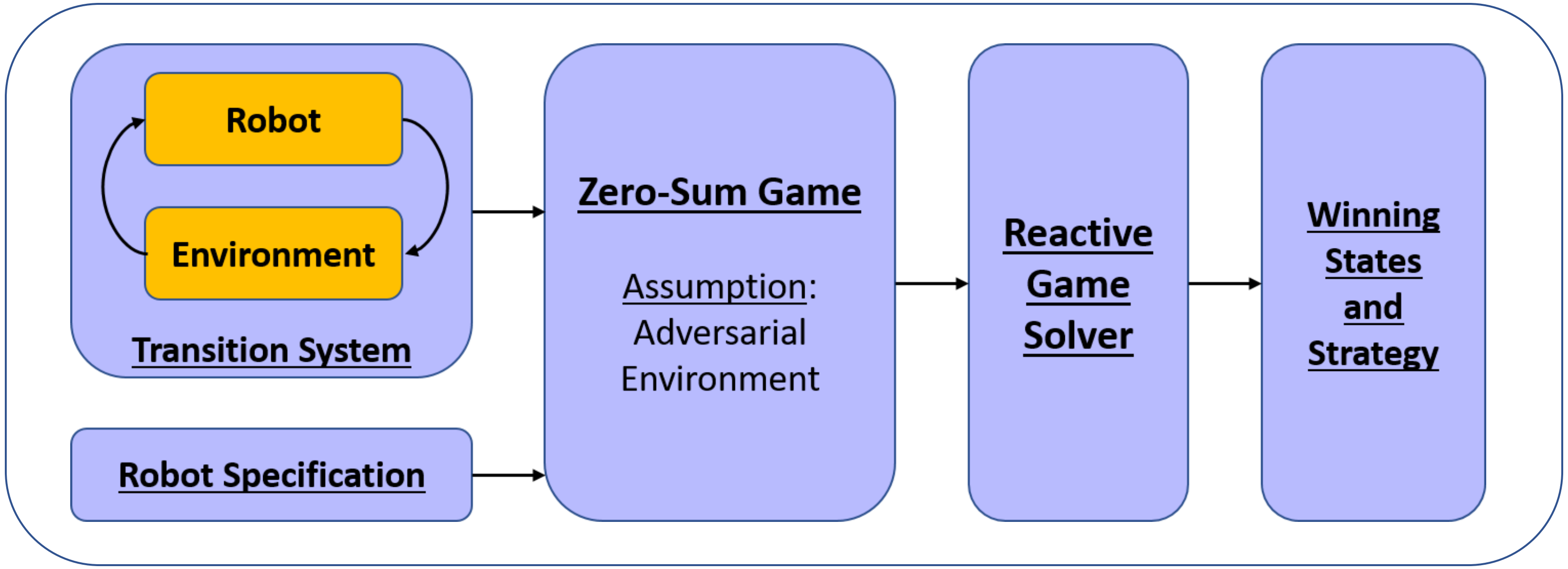


Multi-Robot Mission Planning

From: https://images.app.goo.gl/brSycsnaE5eVLhu29

State-of-the-Art

Given a model of interaction between robot and its environment, what strategy should robot play to ensure satisfaction of its objectives.



Solution Pipeline of Reactive Synthesis

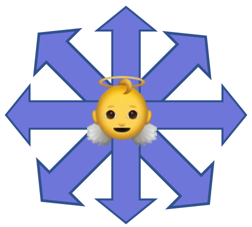
Toy Example

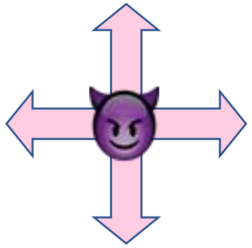
Robot (P1)

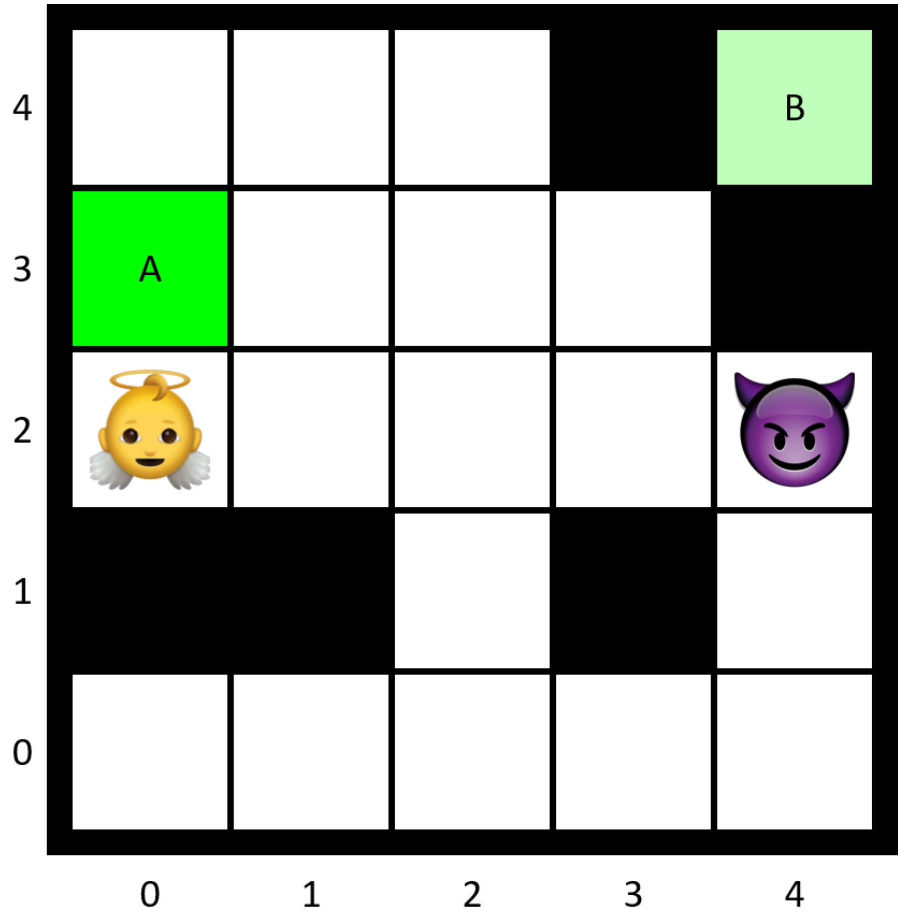
Environment (P2)

A — φ_1: P1's objective known to P2

B — φ_2: P1's objective <u>NOT</u> known to P2



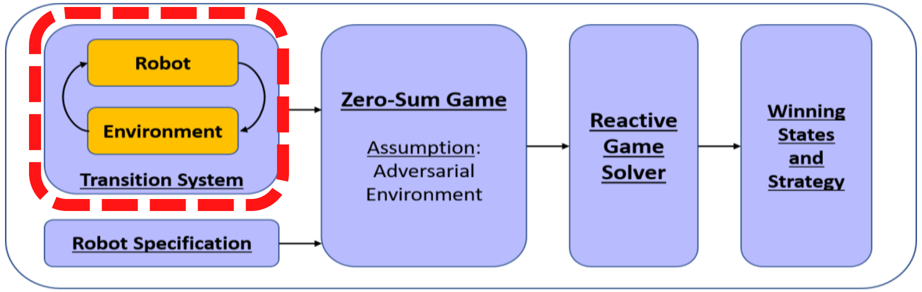




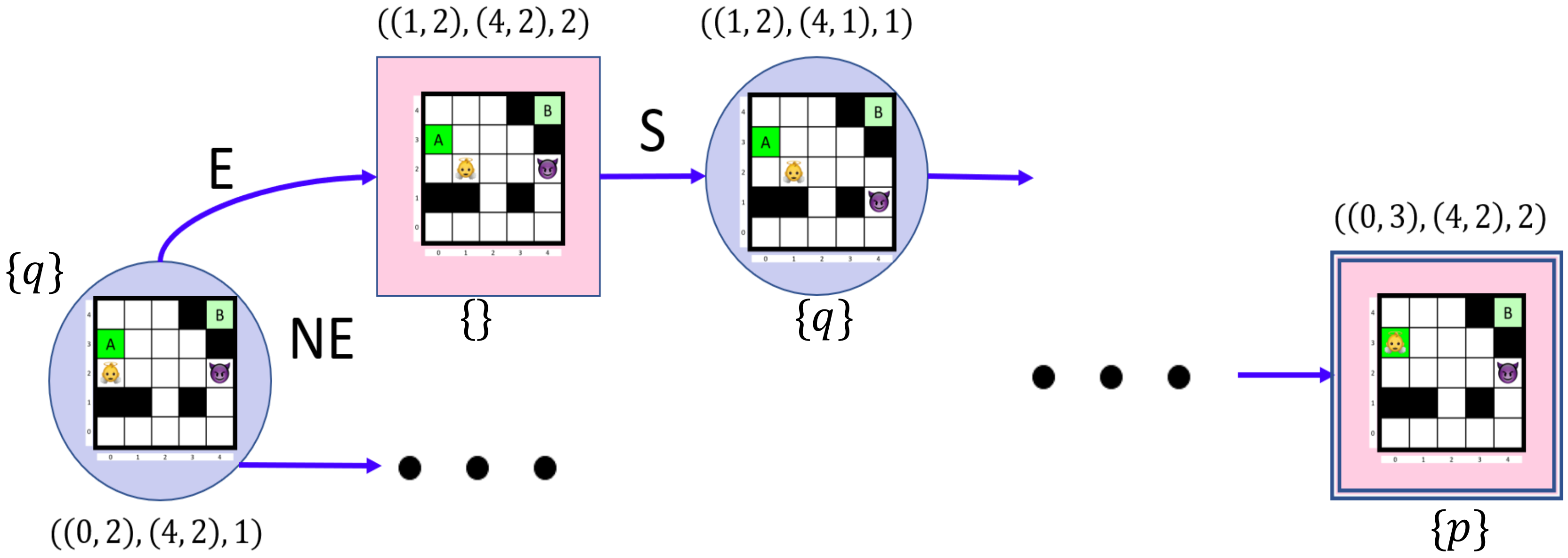
Transition System

A *deterministic turn-based* transition system TS is defined as

$$TS = \langle V, Act, \mathrm{T}, v_0, AP, L \rangle$$



Solution Pipeline of Reactive Synthesis



Atomic Propositions

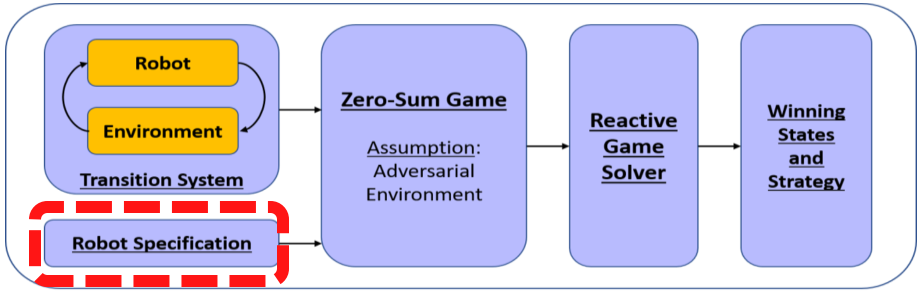
- p: P1 at A
- q: d(P1, P2) ≥ 3

Linear Temporal Logic

Linear Temporal Logic (LTL):

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigcirc\varphi \mid \varphi\, \mathcal{U} \varphi$$

$$\Diamond\varphi = \top\, \mathcal{U} \varphi \qquad \Box\varphi = \neg\Diamond\neg\varphi$$



Solution Pipeline of Reactive Synthesis

Atomic Propositions

- p: P1 at A
- q: d(P1, P2) $\geq$ 3

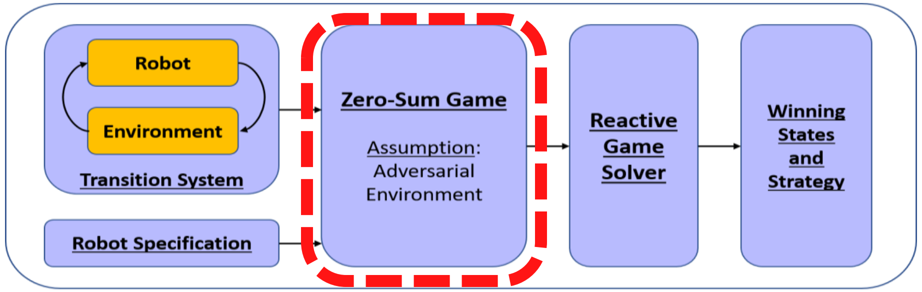
- Examples:
  - $\Diamond p$: Eventually atomic proposition p is true.
  - $\Box p$: Always atomic proposition p is true.
  - $\Diamond(p \wedge \Diamond q)$: Eventually p will be true and then q will be true.

Game with Symmetric Information

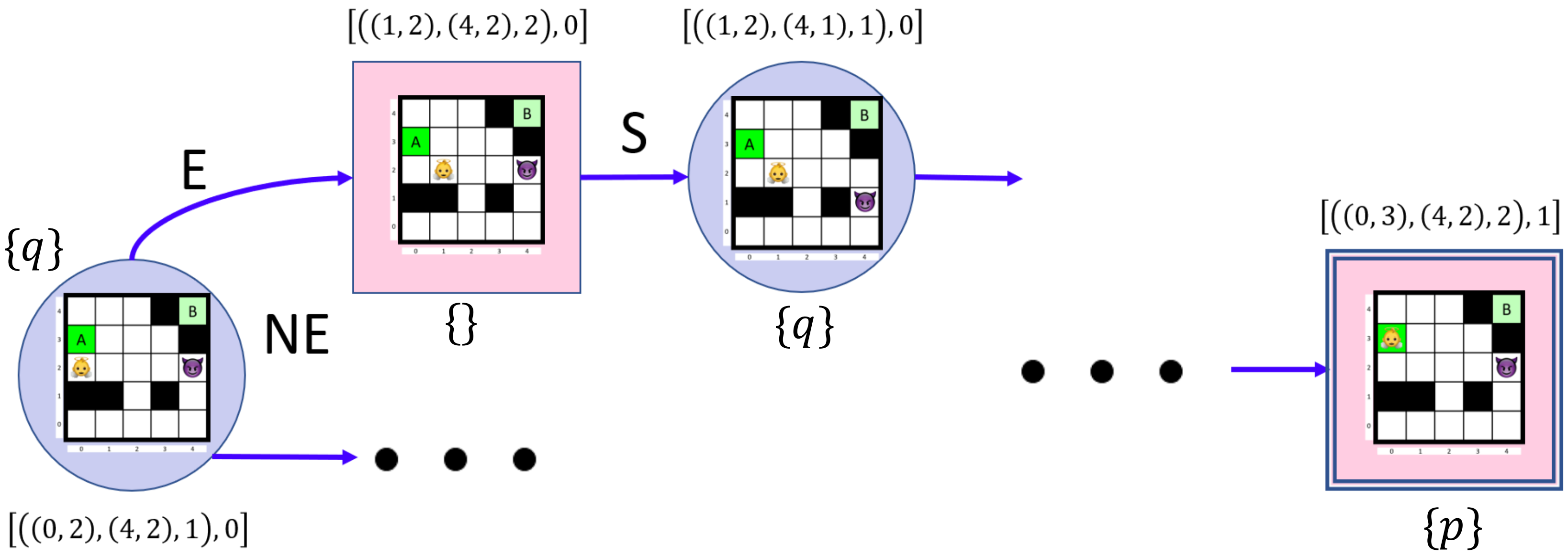
Given TS and LTL objective φ,

a *deterministic turn-based* game is defined as

$$\mathcal{G} = \langle S, Act, \Delta, s_0, F, Acc \rangle$$



Solution Pipeline of Reactive Synthesis



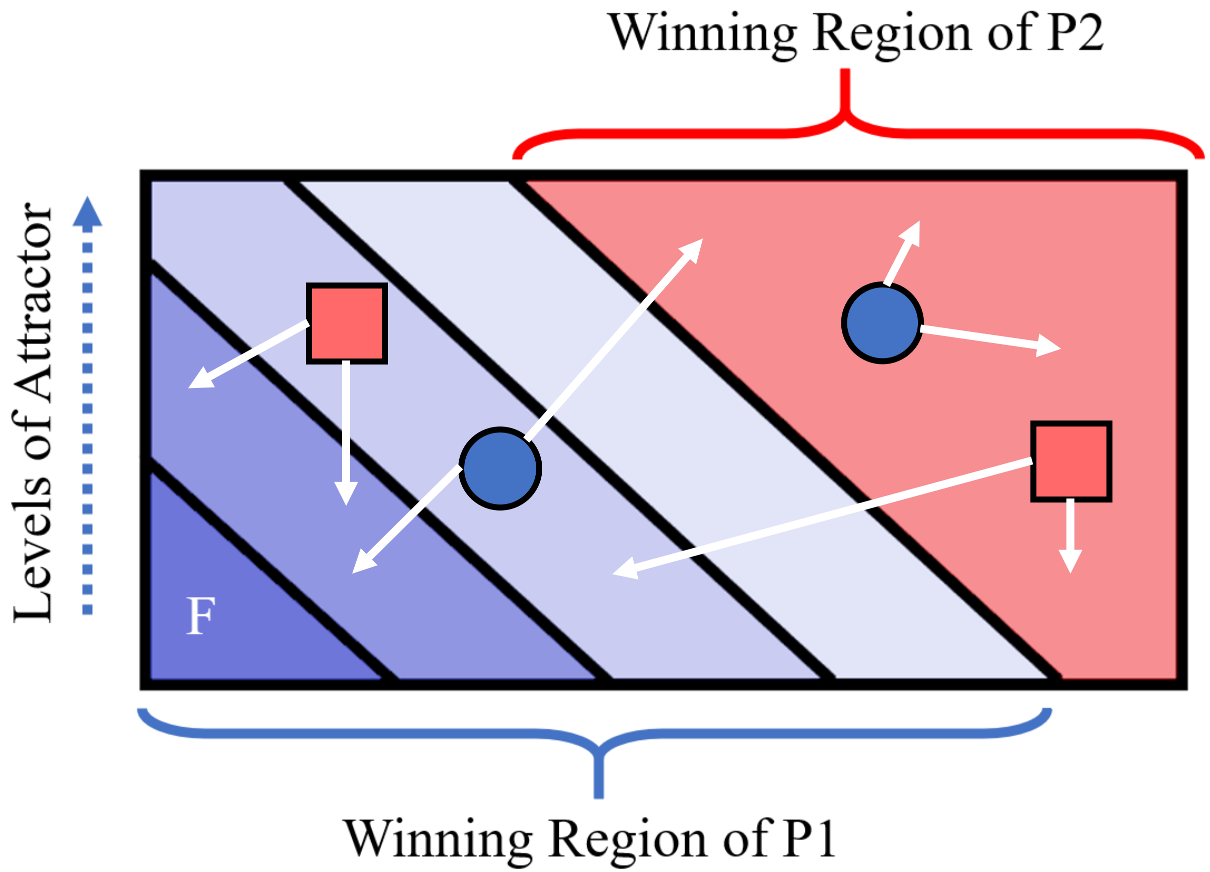
Atomic Propositions

- p: P1 at A
- q: d(P1, P2) > 3

$$\varphi = \Diamond p$$

Winning Region and Winning Strategy

- Zielonka's Recursive Algorithm
- Runs by Recursively Adding States to Winning Region of P1
- Zero-Sum Games are Determined
- Partitions State-Space in at-most Two Parts



Reference:
[1] Wieslaw Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees, Theoretical Computer Science, Volume 200, Issues 1–2, 1998.

Literature Survey

Reactive Games with Complete Information

- Buchi and Landweber (1969): Equivalence of Synthesis Problem with Zero-Sum Games
- Zielonka (1998): Solution of Zero-Sum Turn-based Games on Graph
- L. de Alfaro, T. Henzinger(2000): Solution of Zero-sum Concurrent Games on Graph

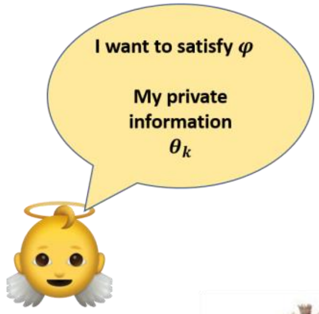
Reference:
[1] Buchi, J. R., & Landweber, L. H. (1969). Solving Sequential Conditions by Finite-State Strategies. *Transactions of the American Mathematical Society*, *138*, 295–311.
[2] Wieslaw Zielonka, **Infinite games on finitely coloured graphs with applications to automata on infinite trees**, *Theoretical Computer Science*, Volume 200, Issues 1–2, 1998.
[3] de Alfaro, Luca, and Thomas A. Henzinger. "Concurrent omega-regular games." Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332). IEEE, 2000.

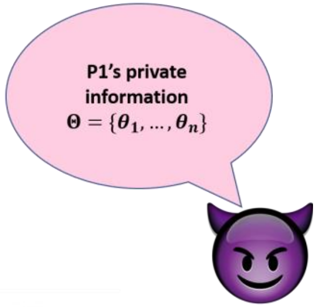
How do we make decisions?

Literature: Reactive Games with Asymmetric Information

I. Bayesian Games

- Harsanyi (1967): Bayesian Games to Model Games with Incomplete Information
  - P1 has some private information (not known to P2)
  - Type of P1 is the private information it has.
  - P2 maintains and updates belief over types: Θ





- Zhuang and Bier (2009): Secrecy and Deception at Equilibrium
- Huang and Zhu (2018): Dynamic Bayesian Games for Cyber-deception

Reference:
[1] Harsanyi, John C. "Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model." Management science 14, no. 3 (1967): 159-182.
[2] Huang, Linan, and Quanyan Zhu. "Dynamic Bayesian Games for Adversarial and Defensive Cyber Deception." arXiv preprint arXiv:1809.02013 (2018).
[3] Jun Zhuang & Vicki M. Bier (2011) SECRECY AND DECEPTION AT EQUILIBRIUM, WITH APPLICATIONS TO ANTI-TERRORISM RESOURCE ALLOCATION, Defence and Peace Economics, 22:1, 43-61

Literature: Reactive Games with Asymmetric Information

II. Hypergames

- Bennett (1977): Hypergames as Game of Games
  - Discussed in next section

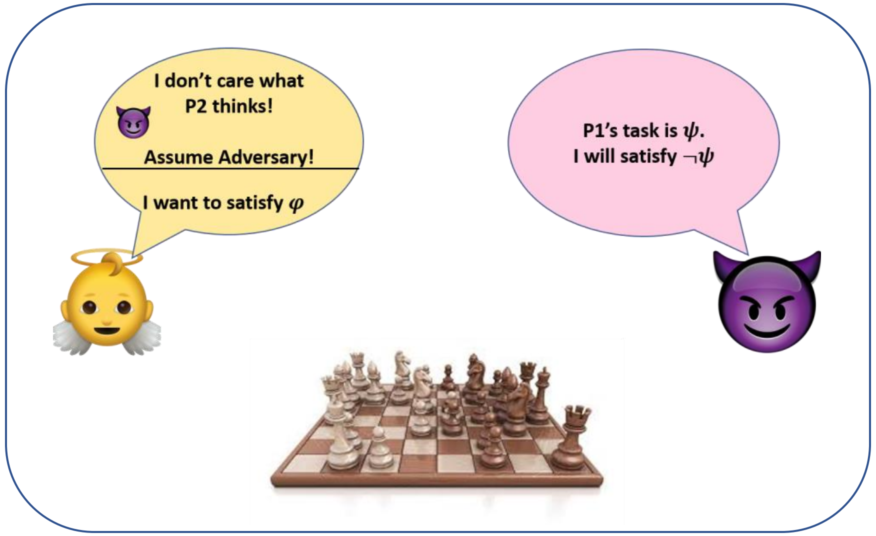
- Gharesifard (2011, 12): Exploration and Evolution of Misperceptions in Hypergames
- Nicholas Kovach (2016): Temporal Framework of Games
- Ehab Al-Shaer et al. (2019): Modeling and Analysis of Normal-form Deception Games

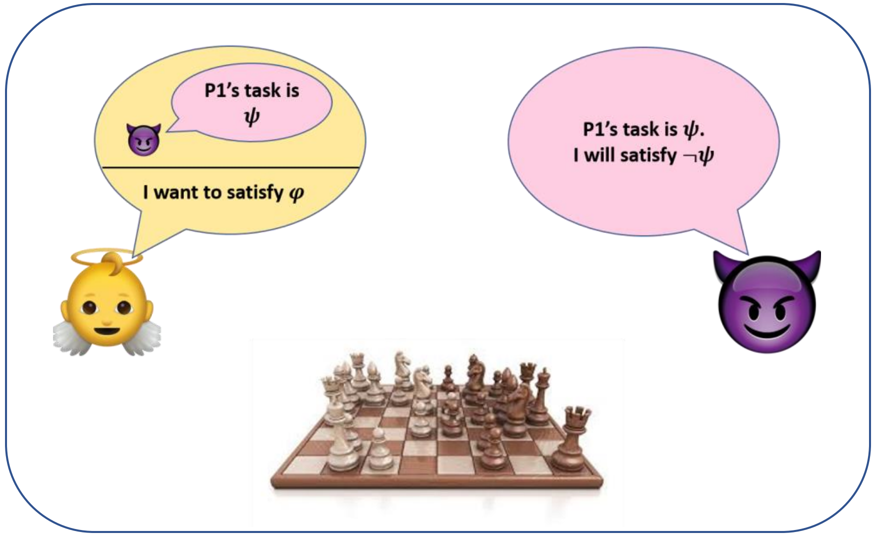
Reference:
[1] Al-Shaer E., Wei J., Hamlen K.W., Wang C. (2019) Modeling and Analysis of Deception Games Based on Hypergame Theory. In: Al-Shaer E., Wei J., Hamlen K., Wang C. (eds) Autonomous Cyber Deception. Springer, Cham
[2] PG Bennett, Toward a theory of hypergames, Omega, Volume 5, Issue 6, 1977, Pages 749-751.
[3] Gharesifard, B., & Cortes, J. (2011). Exploration of misperceptions in hypergames. *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1565–1570.
[4] Gharesifard, B., & Cortés, J. (2012). Evolution of players' misperceptions in hypergames under perfect observations. *IEEE Transactions on Automatic Control*, *57*(7), 1627–1640.
[5] Kovach, Nicholas S.. "A Temporal Framework For Hypergame Analysis Of Cyber Physical Systems In Contested Environments." (2016).

Section II: Opportunistic Synthesis

- What is opportunistic synthesis?
- How to identify, and represent the opportunities?
- How to synthesize an opportunistic strategy?
- Is opportunistic synthesis better than reactive synthesis?

Idea behind Opportunistic Synthesis





Reactive Synthesis

- P2 is Adversary
- P2 has Complete Information

Opportunistic Synthesis

- P2 is an Adversary
- P2 has Partial Information about P1's Objective

Problem Statement

Given φ and ψ such that

$$\boxed{\begin{array}{c}\varphi = \varphi_1 \land \varphi_2 \\ \psi = \varphi_1\end{array}}$$

Problem:

Synthesize an *opportunistic* strategy $\pi: S \to Act$ for P1, which guarantees that P1 will satisfy φ with probability one.

What do we expect?

- Strategy π is better than the strategy computed by reactive synthesis!
  - How do we decide if one strategy is better than other?

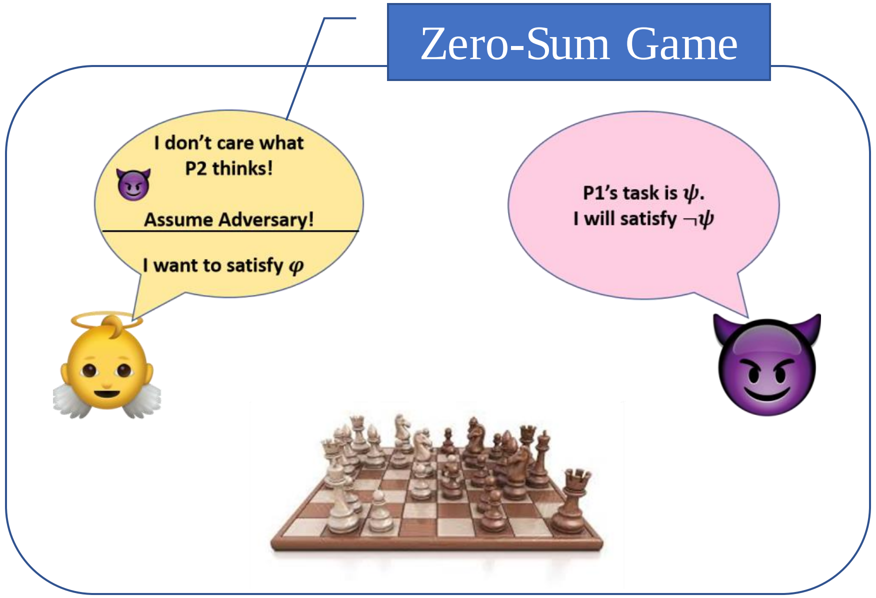
- P1 should be able to leverage Information Asymmetry
  - Try to satisfy $\varphi = \varphi_1 \land \varphi_2$, if possible.
  - If not, try to satisfy φ_2, which is not known to P2.

- Computational cost of new algorithm should be comparable to, if not better than, reactive synthesis.

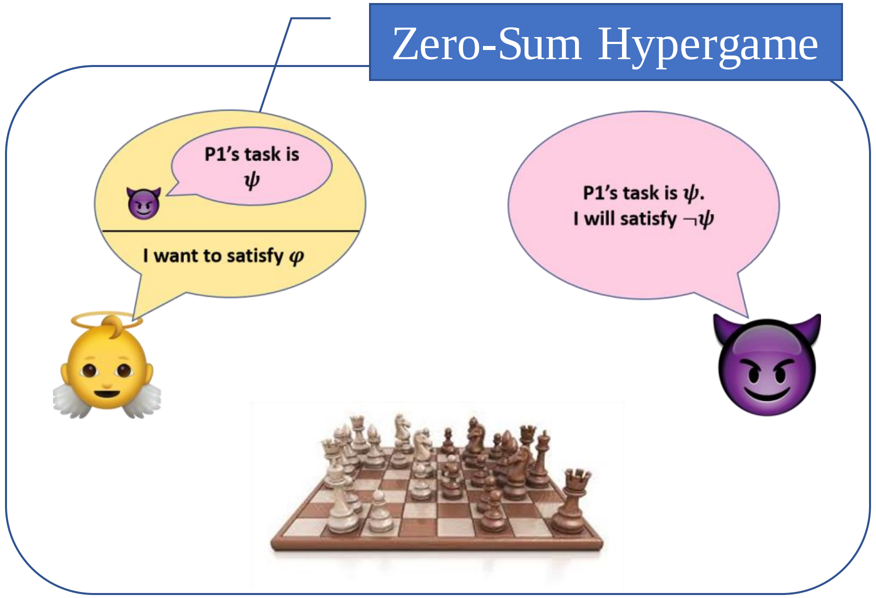
Hypergame, not Game!

- A Zero-sum Hypergame is Game of Zero-sum Games

$$\mathcal{H} = \langle \mathcal{G}_1, \mathcal{G}_2 \rangle$$

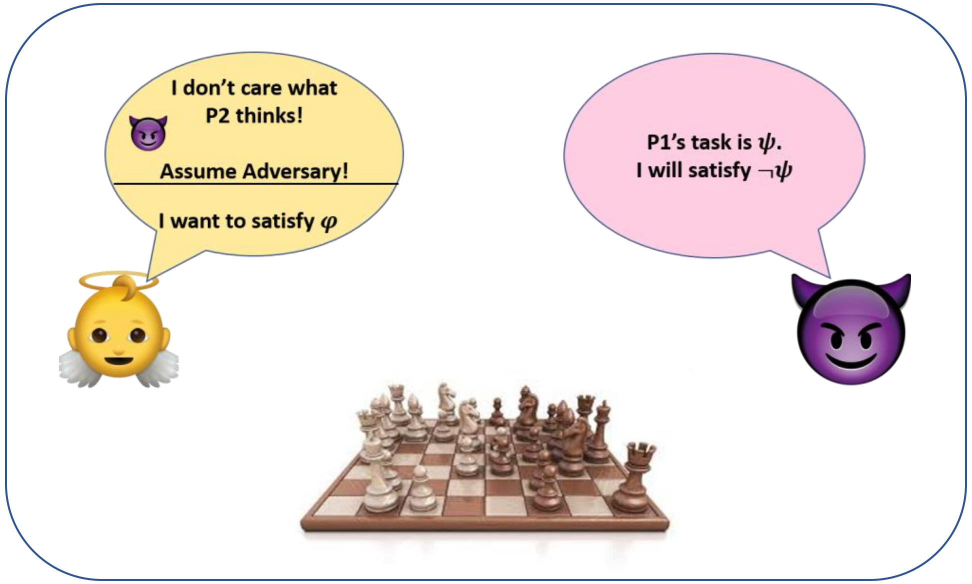


Reactive Synthesis



Opportunistic Synthesis

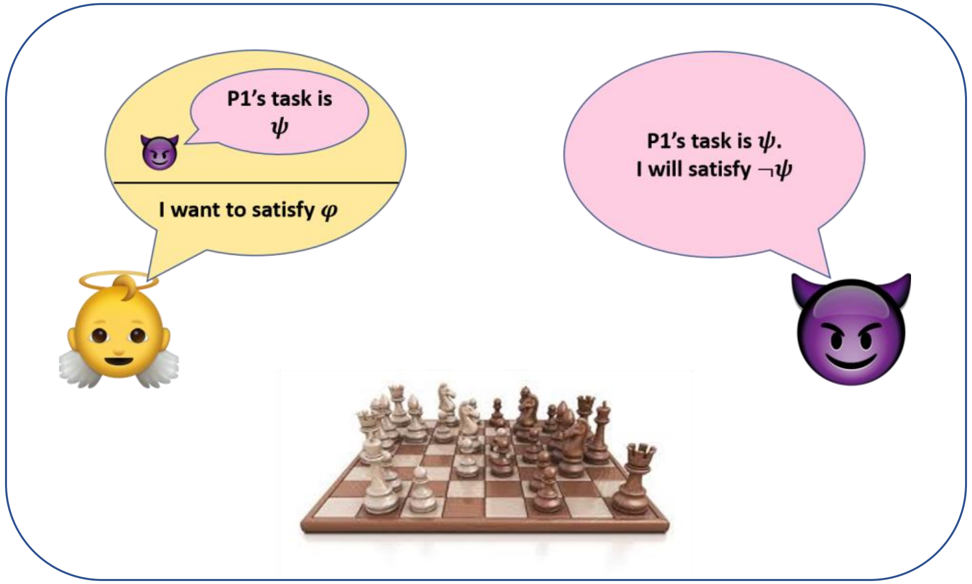
Hypergame Model with Information Asymmetry



Level-1 Hypergame

$$\mathcal{H}^1 = \langle \mathcal{G}_1, \mathcal{G}_2 \rangle$$

- P2 perceives P1's objective as ψ
- P2 uses strategy to satisfy $\neg\psi$
- P1 uses strategy to satisfy φ



Level-2 Hypergame

$$\mathcal{H}^2 = \langle \mathcal{H}^1, \mathcal{G}_2 \rangle$$

- P2 perceives P1's objective as ψ
- P2 uses strategy to satisfy $\neg\psi$
- P1 knows P2 misperceives φ as ψ and uses this information

Hypergame Transition System

Given a second-level hypergame; $\mathcal{H}^2 = \langle\langle\mathcal{G}_1, \mathcal{G}_2\rangle, \mathcal{G}_2\rangle$,
P1 computes strategy by solving $\mathcal{H}^1 = \langle\mathcal{G}_1, \mathcal{G}_2\rangle$.

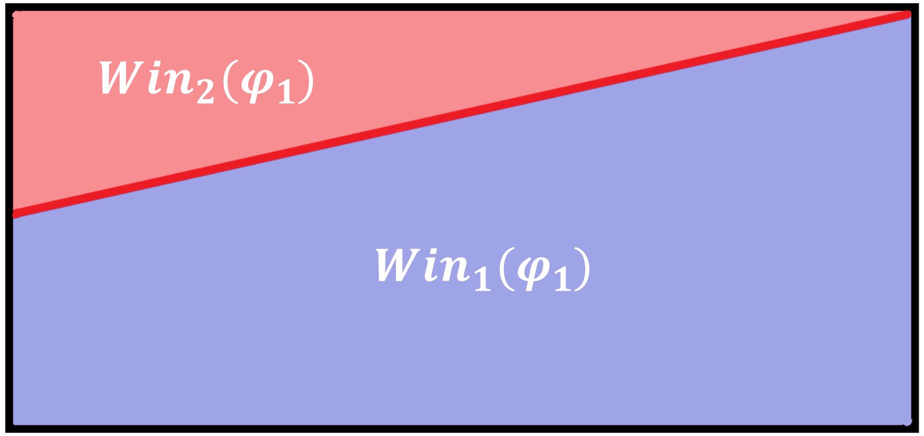
Then, the *deterministic turn-based* hypergame transition system is defined as

$$\boxed{\mathcal{H} = \langle S, Act, \Delta, s_0, \mathcal{F}, Acc\rangle}$$

where

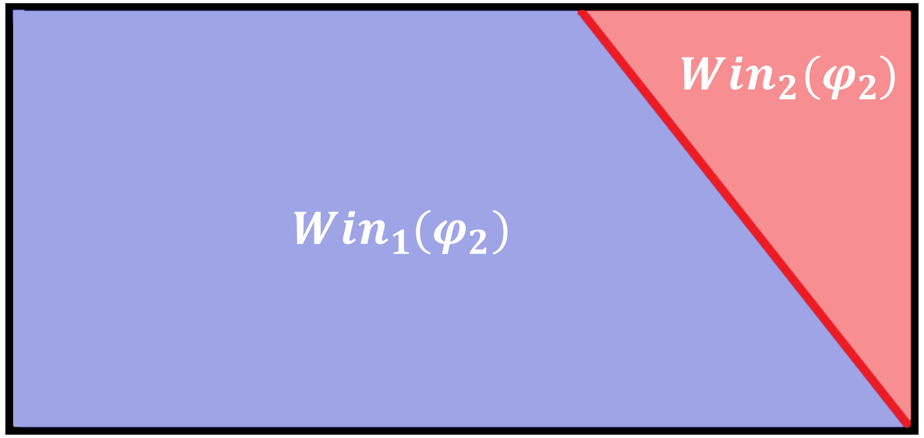
- S, Act, Δ, s_0, Acc have same definitions as that for game (See [slide on games](#))
- $\mathcal{F} = (F_1, F_2, F_{12})$ is a collection of set of final states such that
  - F_1 is set of states corresponding to satisfying φ_1
  - F_2 is set of states corresponding to satisfying φ_2
  - F_{12} is set of states corresponding to satisfying both φ_1 <u>and</u> φ_2.

Solution of Hypergame



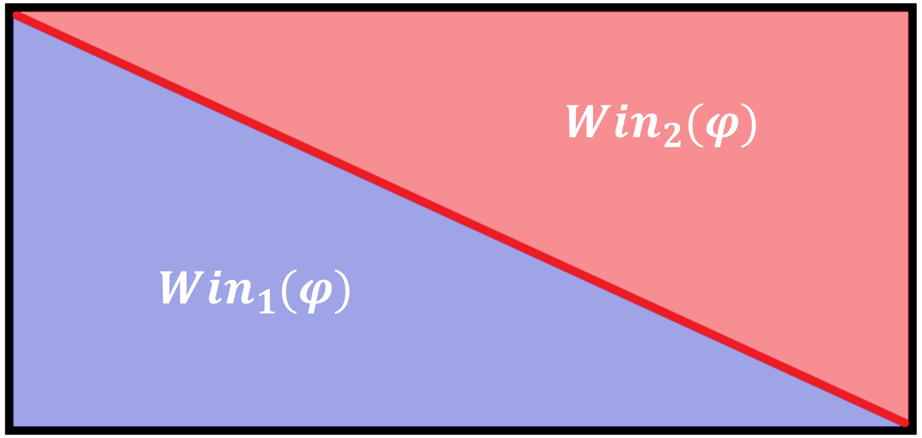
$\mathcal{G}(\varphi_1)$: P1's task is φ_1

P2 <u>knows</u> φ_1



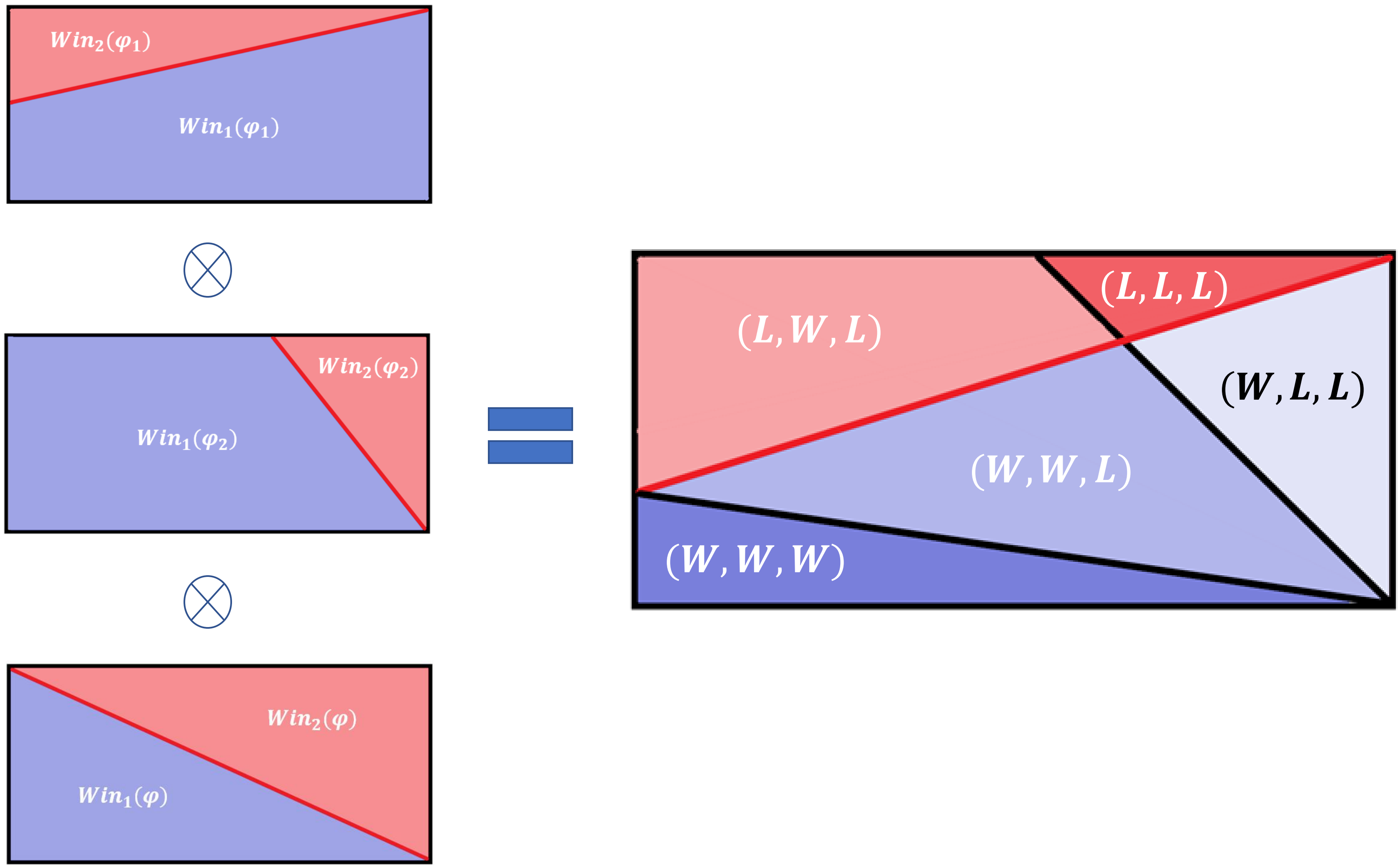
$\mathcal{G}(\varphi_2)$: P1's task is φ_2

P2 <u>does NOT know</u> φ_1



$\mathcal{G}(\varphi)$: P1's task is φ

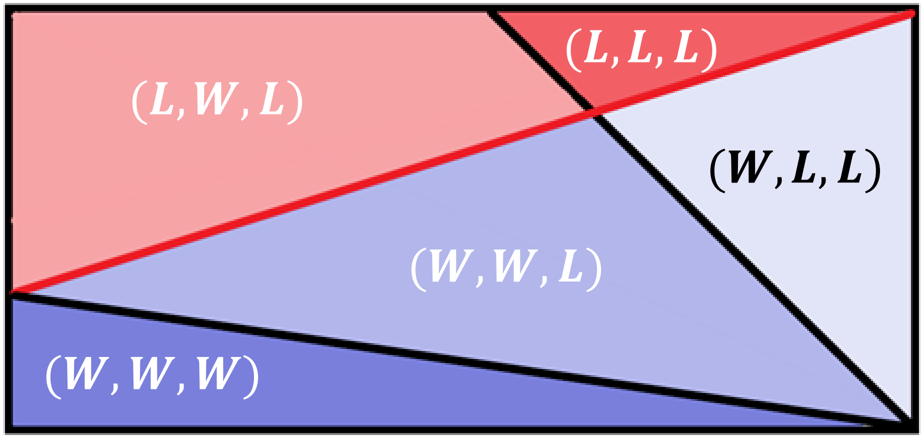
Characterization of Hypergame State-Space



Win-Label (L, W, L) means that

- **P1 is Losing in $\mathcal{G}(\varphi_1)$**
- **P1 is Winning in $\mathcal{G}(\varphi_2)$**
- **P1 is Losing in $\mathcal{G}(\varphi)$**

Solution of Hypergame



Observations

- (W, L, L): P1 has a winning-strategy to satisfy φ_1.
- (L, W, L): P1 has a winning-strategy to satisfy φ_2.
- (W, W, W): P1 has a winning-strategy to satisfy φ.
- (W, W, L): P1 has winning-strategy to satisfy φ_1 ***OR*** φ_2, but not both simultaneously.
- (L, L, L): P2 has a winning-strategy to stop P1 from satisfying φ.

Hypergame Markov Decision Process (MDP)

Assumption:

P2 plays a randomized permissive strategy.

Given hypergame $\mathcal{H}$ and P2's strategy σ, a Hypergame-MDP is defined as

$$\mathcal{H}^{\sigma} = \langle S, Act \cup \{\text{stop}\}, P, s_0, R \rangle$$

where

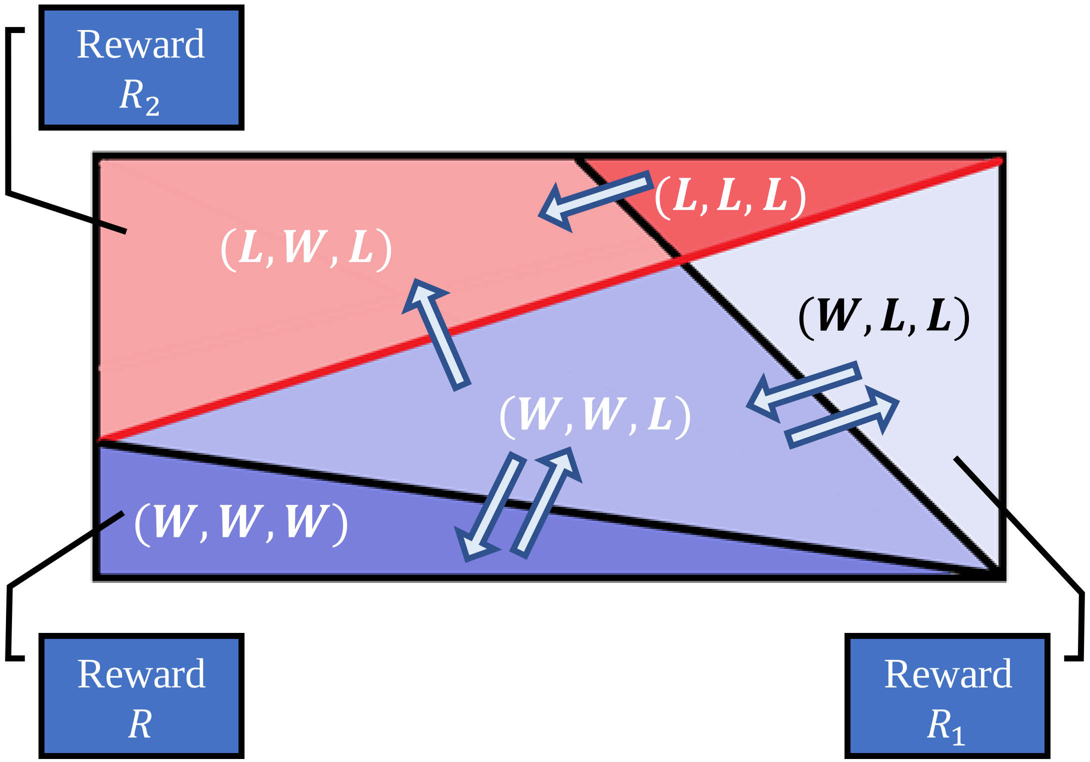
- "stop" is an additional special action that is chosen by P1, when P1 decides to switch to winning strategy in particular sub-game.
- $P: S_1 \times Act_1 \to \mathcal{D}(S_1)$ is a probability transition function.
  - P1 is constructed by marginalizing out P2's strategy σ.
- R is the reward function defined based on satisfaction of φ_1, φ_2 or φ.

Reward Function and Stop Action

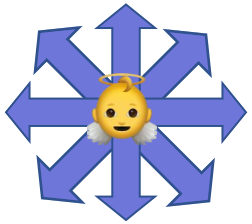
- Reward Function:

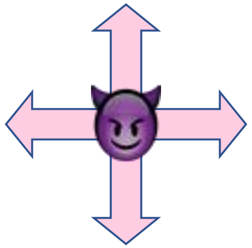
$$\boxed{R = R_1 + R_2}$$

- Case (W, W, L)
  - P1 may wait until it enters (W, W, W)
  - Or execute $STOP$ action
    - To settle with reward R_1 satisfying φ_1
    - Or settle with reward R_2 satisfying φ_2

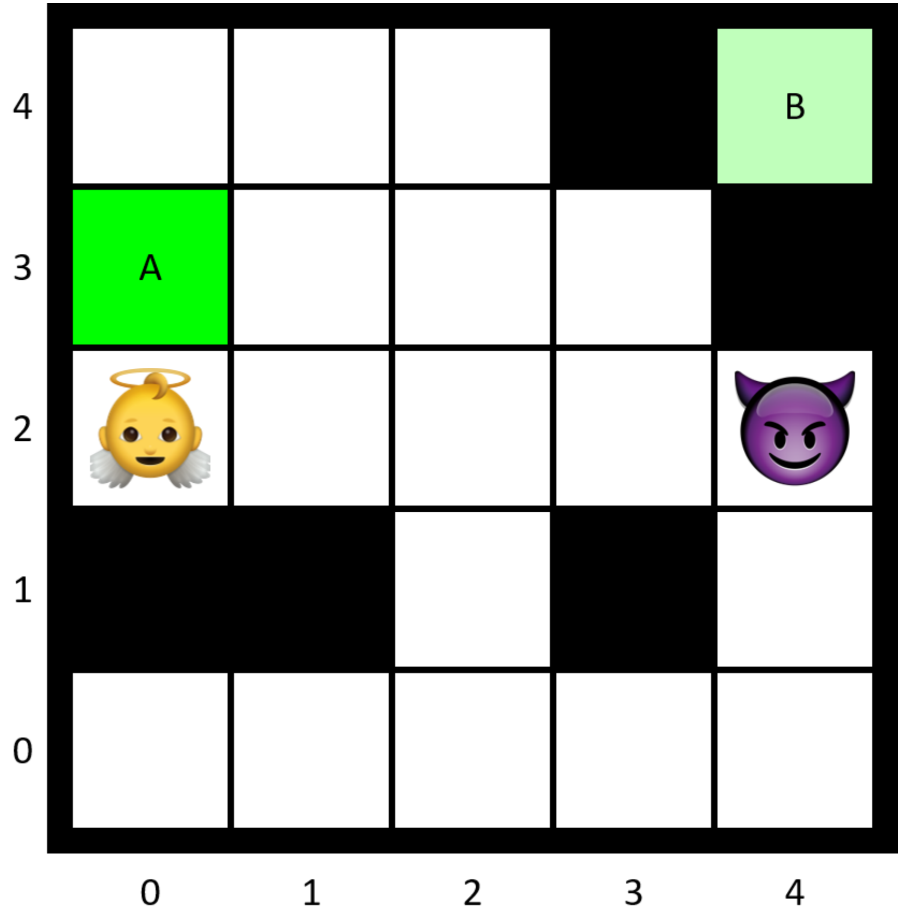


Toy Example

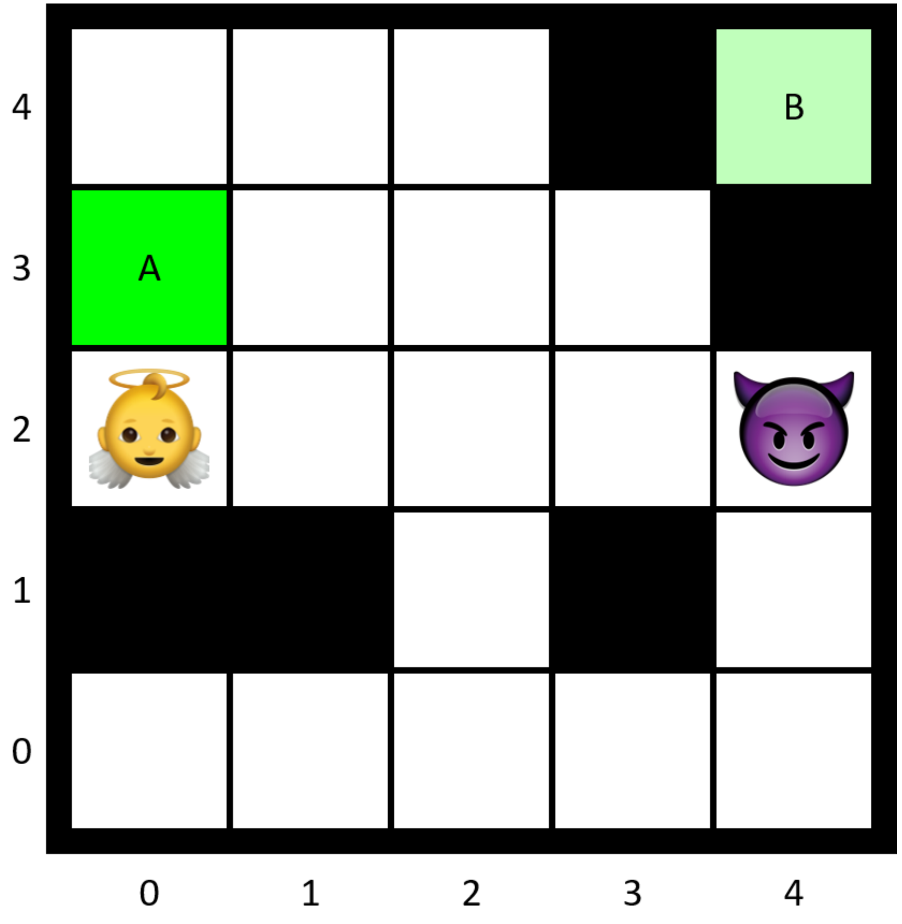




$$\varphi = \underbrace{(\neg O \, \mathcal{U} \, A)}_{\varphi_1} \wedge \underbrace{(\neg O \, \mathcal{U} \, B)}_{\varphi_2}$$



Toy Example



Partition	Number of States
(W, W, W)	1831
(W, W, L)	181
(W, L, L)	479
(L, W, L)	515
(L, L, L)	194
(W, L, W)	0
(L, W, W)	0
(L, L, W)	0

Table 1: Partition of game state-space due to information asymmetry.

A sampled run from (W, L, L) to (W, W, W) from the optimal policy:

1. State: $(((0,2),(4,2),0),1,1)$, win-label: (W, L, L)
2. State: $(((0,3),(3,2),0),0,1)$, win-label: (W, L, L)
3. State: $(((1,2),(2,2),0),0,1)$, win-label: (W, W, W)

Key Results

Result 1:

Opportunistic synthesis may provide a winning strategy for robot from its sure-losing states in reactive game.

Result 2:

Opportunistic synthesis computes a strategy to satisfy φ, φ_1 and φ_2 in order of preference given by the reward function.

Result 3:

Time-complexity of opportunistic synthesis is same as that of reactive synthesis.

Conclusions and Future Work

Opportunistic Synthesis: A Fundamental Problem

- Modelled opportunistic synthesis as a hypergame.
- Reduced the decision problem in hypergame to that of solving an MDP.
- Solution has same complexity as that of reactive synthesis.

Opportunistic and Deceptive Synthesis under

- Misperception of Actions
- Misperception of Labeling Function
- Different models of P2's Intelligence

Acknowledge: DARPA SCIENTIFIC SYSTEMS