Logic and Games*

Muhammad Najib

Department of Computer Science, TU Kaiserslautern

Logic & Semantics of PL Guest Lecture, 28.01.2021

*Some materials are from Oxford's Computational Game Theory Course (https://www.cs.ox.ac.uk/teaching/courses/2020-2021/cgt/)

Overview

1. Connection between Logic and Games
   - Games for Logic
   - Types of Games
   - Logic for Reasoning about Games
   - Game Dynamics
2. Logic and Games for Verification
   - Iterated Games
   - Temporal Logic in Games
   - Further Directions

Outline

Dialectics

Aristotle (384-322 BC)

- Aristotle wrote about syllogism and the rules for debating (dialectics)

Dialectics

- A dialogue (game) between two people.

Dialectics

Aristotle (384-322 BC)

P. Lorenzen (1915-1994)

- Aristotle wrote about syllogism and the rules for debating (dialectics)
- Paul Lorenzen wrote about dialogical logic in the 50s.

Dialectics

Aristotle (384-322 BC)

P. Lorenzen (1915-1994)

C. Hamblin (1922-1985)

- Aristotle wrote about syllogism and the rules for debating (dialectics)
- Paul Lorenzen wrote about dialogical logic in the 50s.
- Charles Hamblin† wrote "Mathematical Models of Dialogue" in the 70s.

† Fun fact: Hamblin introduced Reverse Polish Notation.

Determinacy

E. Zermelo
(1871-1953)

- Ernst Zermelo's theorem for finite, perfect information games (e.g. Chess, Go, Tic-Tac-Toe):
  if a player is in a winning position, then they can always force a win no matter what strategy the other player may employ[‡]

[‡] "ber eine Anwendung der Mengenlehre auf die Theorie des Schachspiels" (1913).

Determinacy

E. Zermelo
(1871-1953)

Chess (15th c.-present)

- Chess: **either White can force a win, or Black can force a win, or both sides can force at least a draw.** We don't know which case is true, because the game tree is too BIG.

Determinacy

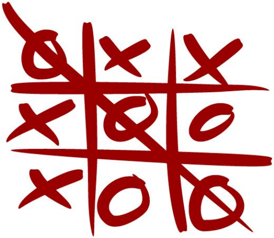
E. Zermelo
(1871-1953)

Chess (15th c.-present)

- Chess: **either White can force a win, or Black can force a win, or both sides can force at least a draw.** We don't know which case is true, because the game tree is too BIG.
- Shannon number (conservative lower bound): 10^{120}. For perspective, 10^{82} atoms in the observable universe.

Determinacy

E. Zermelo
(1871-1953)

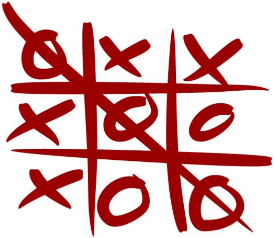


Tic-tac-toe (?-present)

- Tic-tac-toe is a solved game: **both sides can force a draw.** Assuming best play.

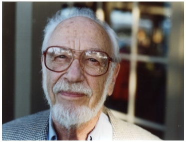
Determinacy

E. Zermelo
(1871-1953)



Tic-tac-toe (?-present)

- Tic-tac-toe is a solved game: **both sides can force a draw.** Assuming best play.
- Game tree size: 26,830. Easy to check with computer.



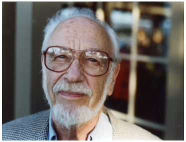
D. Gale
(1921-2008)

F. Stewart
(1917-2011)

- David Gale and Frank Stewart proved for **infinite** games.[§]

[§] Gale, D. and F. M. Stewart (1953). "Infinite games with perfect information".

Determinacy



D. Gale
(1921-2008)

F. Stewart
(1917-2011)

- David Gale and Frank Stewart proved for **infinite** games.[§]

Gale-Stewart Theorem

Every open or closed game $G(W)$ is determined.

[§] Gale, D. and F. M. Stewart (1953). "Infinite games with perfect information".

Determinacy

Abaelardus and Hlose in the manuscript *Roman de la Rose* (14th c.)

There **exists** a strategy for **Eloise**, such that **for all** strategies of **Abelard**, Eloise **wins**.

Determinacy

Abaelardus and Hlose in the manuscript *Roman de la Rose* (14th c.)

There **exists** a strategy for **Eloise**, such that **for all** strategies of **Abelard**, Eloise **wins**.

There exists x, for all y such that $P(x, y)$.

FOL and Games

Alfred Tarski (1901-1983).

- In the early 1930s Alfred Tarski proposed a definition of truth of first-order sentences.

FOL and Games

Alfred Tarski (1901-1983).

- In the early 1930s Alfred Tarski proposed a definition of truth of first-order sentences.
- 'There exists x, for all y such that $P(x, y)$' is true

FOL and Games

Alfred Tarski (1901-1983).

- In the early 1930s Alfred Tarski proposed a definition of truth of first-order sentences.
- 'There exists x, for all y such that $P(x, y)$' is true
- There is an object a such that the sentence 'for all y such that $P(a, y)$' is true.

FOL and Games

Alfred Tarski (1901-1983).

- In the early 1930s Alfred Tarski proposed a definition of truth of first-order sentences.
- 'There exists x, for all y such that $P(x, y)$' is true
- There is an object a such that the sentence 'for all y such that $P(a, y)$' is true.
- There is an object a for every object b such that the sentence '$P(a, b)$' is true

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

For all x_0 **exists** y_0 such that **for all** x_1 **exists** y_1... $P(x_0, y_0, x_1, y_1, ...)$

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

For all x_0 **exists** y_0 such that **for all** x_1 **exists** y_1... $P(x_0, y_0, x_1, y_1, \ldots)$

- Abelard chooses an object a_0 for x_0, then Eloise chooses e_0 for y_0, Abelard chooses a_1 for x_1, then Eloise e_1 for y_1 and so on.

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

For all x_0 **exists** y_0 such that **for all** x_1 **exists** y_1... $P(x_0, y_0, x_1, y_1, ...)$

- Abelard chooses an object a_0 for x_0, then Eloise chooses e_0 for y_0, Abelard chooses a_1 for x_1, then Eloise e_1 for y_1 and so on.
- We have $P(a_0, e_0, a_1, e_1, ...)$

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

For all x_0 **exists** y_0 such that **for all** x_1 **exists** y_1... $P(x_0, y_0, x_1, y_1, ...)$

- Abelard chooses an object a_0 for x_0, then Eloise chooses e_0 for y_0, Abelard chooses a_1 for x_1, then Eloise e_1 for y_1 and so on.
- We have $P(a_0, e_0, a_1, e_1, ...)$
- Skolemnisation: $P(a_0, f_0(a_0), a_1, f_1(a_0, a_1), ...)$

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

For all x_0 **exists** y_0 such that **for all** x_1 **exists** $y_1 \ldots$ $P(x_0, y_0, x_1, y_1, \ldots)$

- Abelard chooses an object a_0 for x_0, then Eloise chooses e_0 for y_0, Abelard chooses a_1 for x_1, then Eloise e_1 for y_1 and so on.
- We have $P(a_0, e_0, a_1, e_1, \ldots)$
- Skolemnisation: $P(a_0, f_0(a_0), a_1, f_1(a_0, a_1), \ldots)$
- The Skolem functions f_0, f_1 etc. define a winning strategy for Eloise.[¶]

[¶] Henkin, L. (1961). "Some remarks on infinitely long formulas". In J. of Symbolic Logic.

FOL and Games

L. Henkin (1921-2006)

Leon Henkin extended Alfred Tarski's definition of truth.

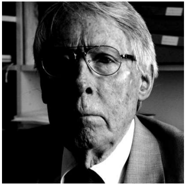
For all x_0 **exists** y_0 such that **for all** x_1 **exists** $y_1 \ldots$ $P(x_0, y_0, x_1, y_1, \ldots)$

- Abelard chooses an object a_0 for x_0, then Eloise chooses e_0 for y_0, Abelard chooses a_1 for x_1, then Eloise e_1 for y_1 and so on.
- We have $P(a_0, e_0, a_1, e_1, \ldots)$
- Skolemnisation: $P(a_0, f_0(a_0), a_1, f_1(a_0, a_1), \ldots)$
- The Skolem functions f_0, f_1 etc. define a winning strategy for Eloise.[‖]

A connection to Gale-Stewart Theorem (determinacy of infinite games)!

[‖] Henkin, L. (1961). "Some remarks on infinitely long formulas". In J. of Symbolic Logic.

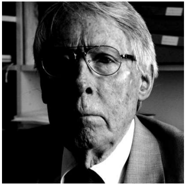
FOL and Games



J. Hintikka (1929-2015)

- Jaakko Hintikka applied games on conjunctions and disjunctions.

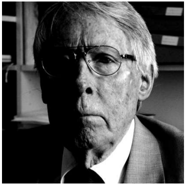
FOL and Games



J. Hintikka (1929-2015)

- Jaakko Hintikka applied games on conjunctions and disjunctions.
- $\varphi \wedge \psi$ as a universally quantified statement, i.e.,
  "every one of the sentences φ, ψ holds"

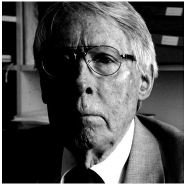
FOL and Games



J. Hintikka (1929-2015)

- Jaakko Hintikka applied games on conjunctions and disjunctions.
- $\varphi \wedge \psi$ as a universally quantified statement, i.e., "every one of the sentences φ, ψ holds"
- to play game $G(\varphi \wedge \psi)$, Abelard chooses whether the game should proceed as $G(\varphi)$ or $G(\psi)$.

FOL and Games



J. Hintikka (1929-2015)

- Jaakko Hintikka applied games on conjunctions and disjunctions.
- $\varphi \wedge \psi$ as a universally quantified statement, i.e., "every one of the sentences φ, ψ holds"
- to play game $G(\varphi \wedge \psi)$, Abelard chooses whether the game should proceed as $G(\varphi)$ or $G(\psi)$.
- Analogously for disjunctions: Eloise determines how the game $G(\varphi \vee \psi)$ should proceed.

Outline

Types of Games

How do we model games?

Types of Games

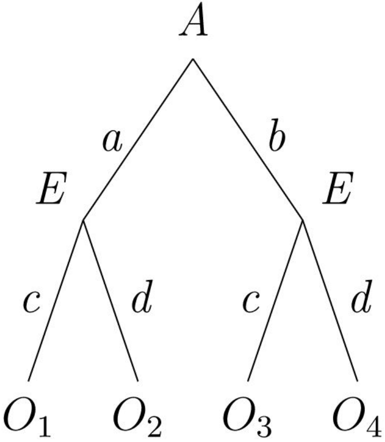
How do we model games?

Two major perspective:

1. Games in **extensive form**
2. Games in **strategic/normal form**

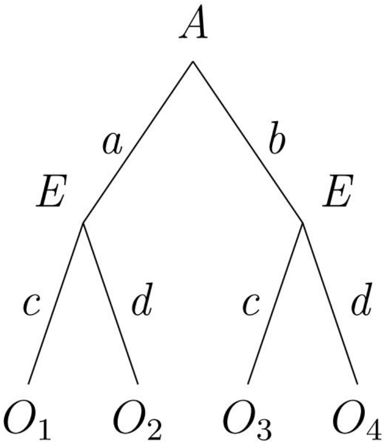
Types of Games: extensive form

- Explicit temporal structure



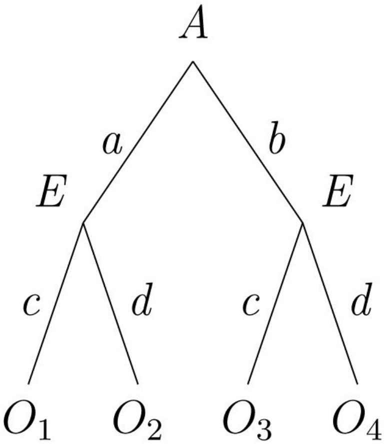
Types of Games: extensive form

- Explicit temporal structure
- Each *non-terminal* node owned by one player (whose turn)



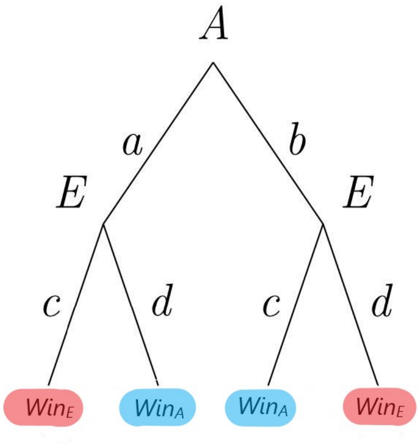
Types of Games: extensive form

- Explicit temporal structure
- Each *non-terminal* node owned by one player (whose turn)
- Edges correspond to possible moves/actions



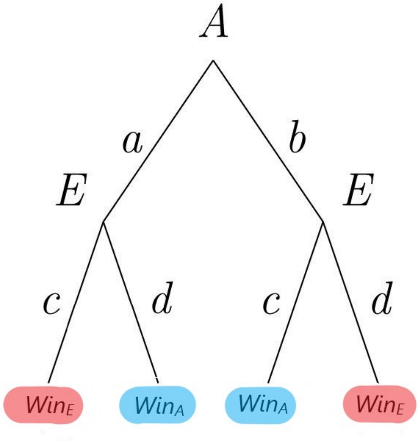
Types of Games: extensive form

Who has winning strategy?



Types of Games: extensive form

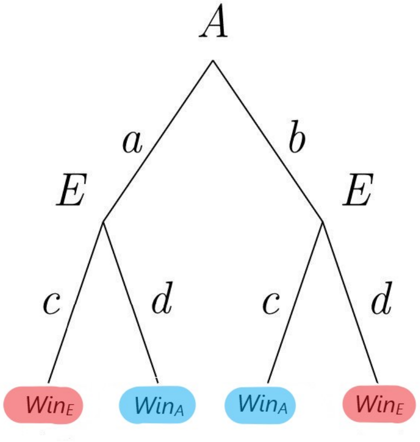
Who has winning strategy? **Eloise**.



Types of Games: extensive form

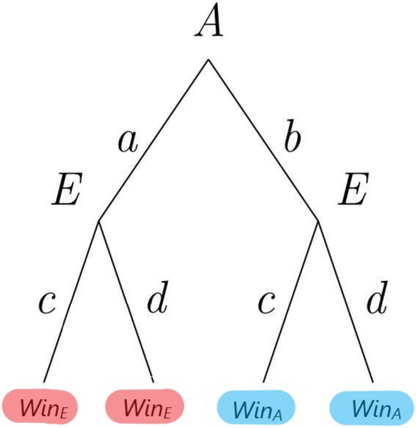
Who has winning strategy? **Eloise**.

If Abelard chooses *a* then choose *d*, else choose *c*.



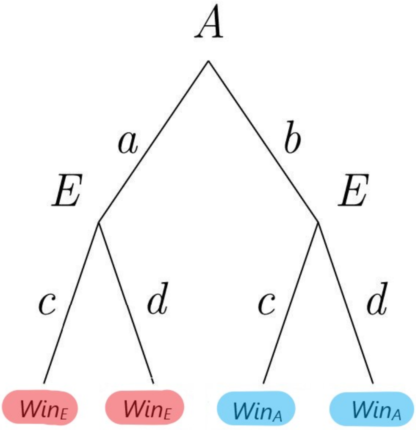
Types of Games: extensive form

What about this?



Types of Games: extensive form

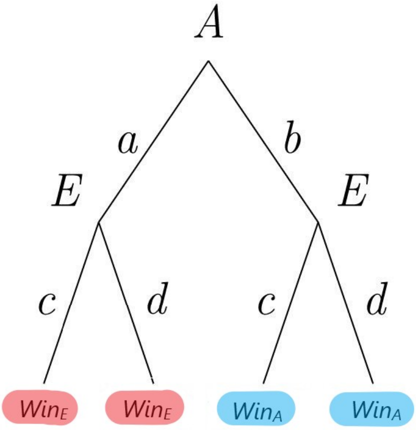
What about this? **Abelard**.



Types of Games: extensive form

What about this? **Abelard**.

Choose b.



Types of Games: strategic/normal form

- Emphasise players' **available strategies**

		E	
		a	b
A	c	O_1	O_2
	d	O_3	O_4

Types of Games: strategic/normal form

- Emphasise players' **available strategies**
- No temporal structure

		E	
		a	b
A	c	O_1	O_2
	d	O_3	O_4

Types of Games: strategic/normal form

Who has winning strategy?

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

Types of Games: strategic/normal form

Who has winning strategy? **Eloise**.

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

Who has winning strategy? **Eloise**. Choose b

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

Types of Games: strategic/normal form

Who has winning strategy?

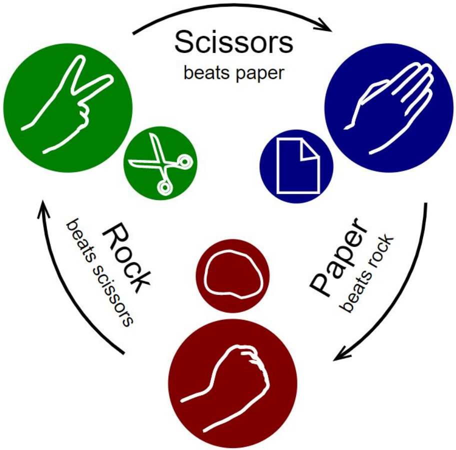
		E	
		a	b
A	c	Win_A	Win_E
	d	Win_E	Win_A

Types of Games: strategic/normal form

Who has winning strategy? **Nobody**.

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_E	Win_A

Which model is appropriate for the Rock-Paper-Scissors game?



By Enzoklop - Own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=27958688

Rock, Paper, Scissors	Rock	Paper	Scissors
Rock	0, 0	-1, 1	1, -1
Paper	1, -1	0, 0	-1, 1
Scissors	-1, 1	1, -1	0, 0

From http://gametheory101.com/

Outline

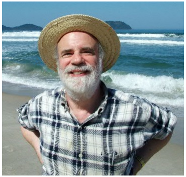
Modal Logic

Let τ a non-empty countable set, AP a set of atomic propositions. $ML(\tau, AP)$ is recursively defined as:

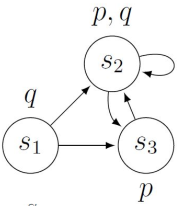
$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle a \rangle \varphi$$

where $p \in AP$ and $a \in \tau$. $[a]\varphi \equiv \neg\langle a \rangle \neg\varphi$.

A model for $ML(\tau, AP)$ is a relational (Kripke) structure $\mathcal{M} = (\mathsf{St}, (R_a)_{a\in\tau}, V)$, where St is a non-empty set of nodes (worlds/states), $R_a \subseteq \mathsf{St} \times \mathsf{St}$, and, $V : \mathsf{St} \to 2^{AP}$.



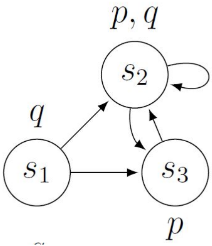
S. Kripke (1940-)

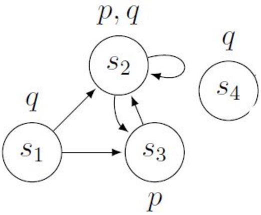


Modal Logic: semantics

We interpret $ML(\tau, AP)$ over models as follows:

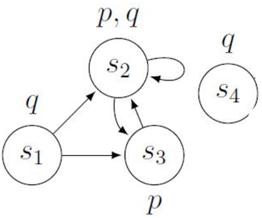
$$
\begin{array}{lll}
\mathcal{M}, s \models p & \text{iff} & p \in V(s), \\
\mathcal{M}, s \models \neg\varphi & \text{iff} & \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff} & \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models \langle a \rangle \varphi & \text{iff} & \text{there exists } s' \in \mathrm{St} \text{ with } sR_a s' \text{ and } \mathcal{M}, s' \models \varphi \\
\mathcal{M}, s \models [a]\varphi & \text{iff} & \text{for all } s' \in \mathrm{St}, \text{ if } sR_a s' \text{ then } \mathcal{M}, s' \models \varphi
\end{array}
$$





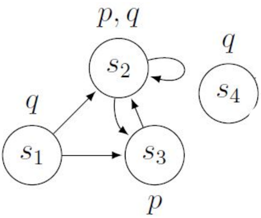
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1?



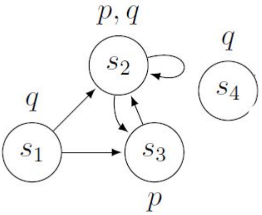
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.



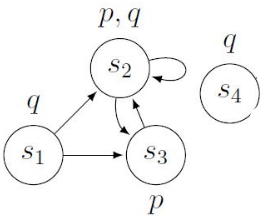
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1?



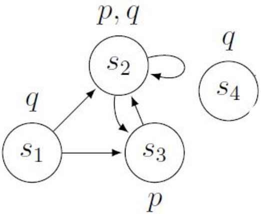
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1? no.



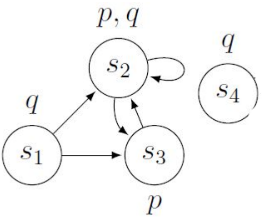
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1? no.
- $\langle a \rangle p$ satisfied in s_4?



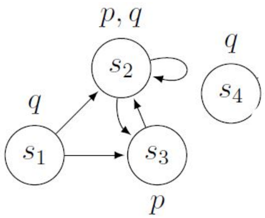
suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1? no.
- $\langle a \rangle p$ satisfied in s_4? no.



suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1? no.
- $\langle a \rangle p$ satisfied in s_4? no.
- $[a]p$ satisfied in s_4?

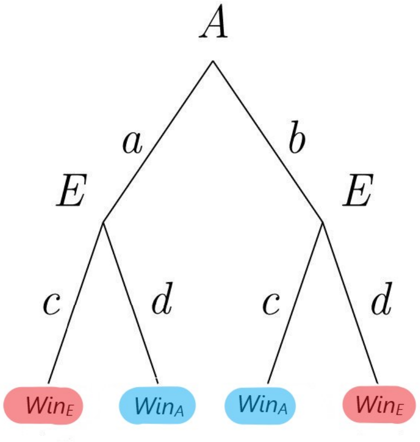


suppose all edges are labelled a

- $\langle a \rangle p$ satisfied in s_1? yes.
- $[a]q$ satisfied in s_1? no.
- $\langle a \rangle p$ satisfied in s_4? no.
- $[a]p$ satisfied in s_4? yes.

Modal Logic for Extensive Games

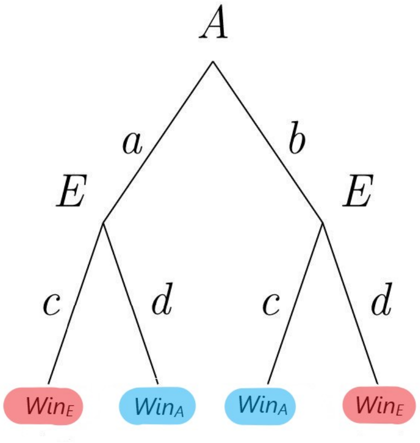
We can think of extensive form game structure as a model for $ML(\tau, AP)$



Modal Logic for Extensive Games

$$\varphi_E := [move_A]\langle move_E \rangle Win_E$$

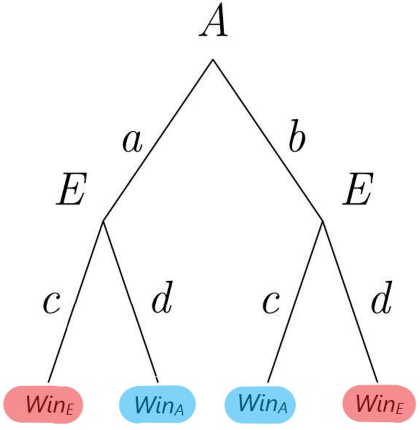
Is the formula φ_E satisfied by the model?



Modal Logic for Extensive Games

$$\varphi_E := [move_A]\langle move_E \rangle Win_E$$

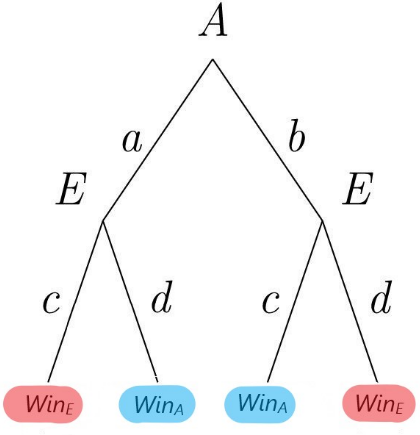
Is the formula φ_E satisfied by the model? YES.



Modal Logic for Extensive Games

$$\varphi_E := [move_A]\langle move_E\rangle Win_E$$

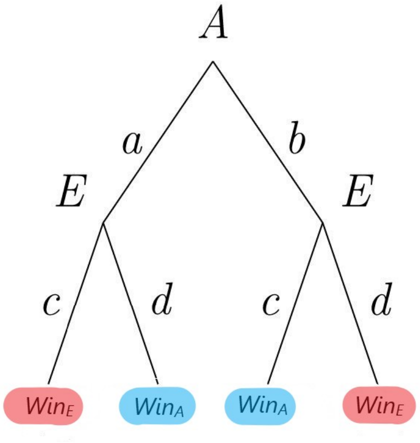
Is the formula φ_E satisfied by the model? YES.
φ_E expresses "Eloise has a winning strategy"



Modal Logic for Extensive Games

$$\varphi_A := \neg\varphi_E = \langle move_A \rangle [move_E] \neg Win_E$$

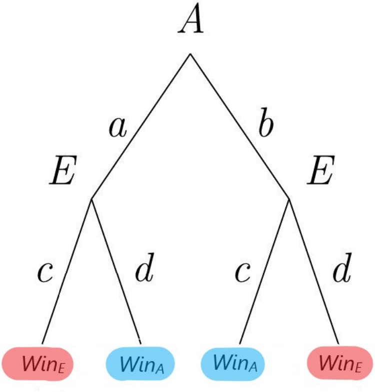
Is the formula φ_A satisfied by the model?



Modal Logic for Extensive Games

$$\varphi_A := \neg\varphi_E = \langle move_A \rangle [move_E] \neg Win_E$$

Is the formula φ_A satisfied by the model? NO (by law of excluded middle).

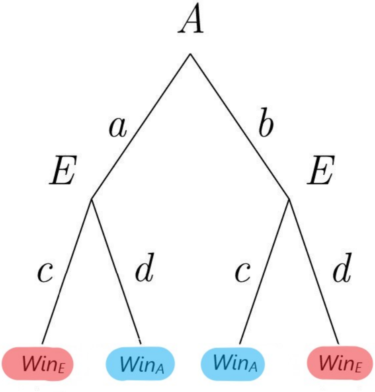


Modal Logic for Extensive Games

$$\varphi_A := \neg\varphi_E = \langle move_A \rangle [move_E] \neg Win_E$$

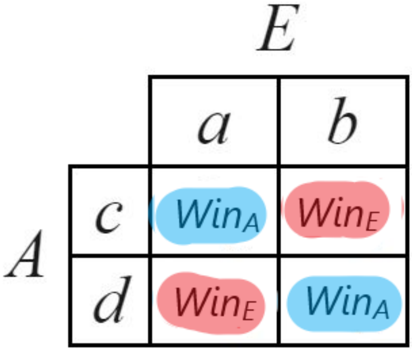
Is the formula φ_A satisfied by the model? NO (by law of excluded middle).

Zermelo's theorem: Eloise has a winning strategy iff Abelard does not have one.



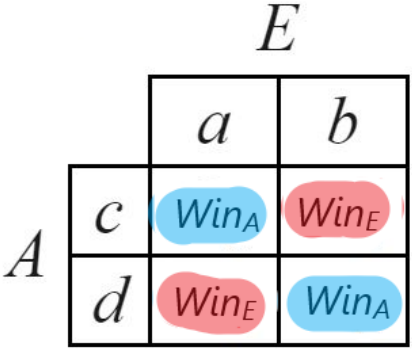
Outline

Unstable Game



Everytime players try to fix their actions, one of them wants to change the action, i.e., the game is **unstable**

Unstable Game



Everytime players try to fix their actions, one of them wants to change the action, i.e., the game is **unstable**

Zermelo's theorem does not apply here!

Stable Game

This game is **stable**: Eloise always plays b, Abelard is *indifference* between c and d.

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

Stable Game

This game is **stable**: Eloise always plays b, Abelard is *indifference* between c and d.

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

We can **predict** the outcome of the game above.

Solution Concept

How do we predict the outcome of a game? Use **solution concepts**.

Solution Concept

How do we predict the outcome of a game? Use **solution concepts**.

a **solution concept** is a formal rule for predicting how a game will be played.

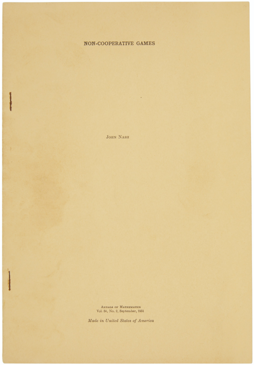
Solution Concept

How do we predict the outcome of a game? Use **solution concepts**.

a **solution concept** is a formal rule for predicting how a game will be played.

There are many, but the most important is **Nash equilibrium**

J. Nash (1928-2015)



26 pages, 2 citations

Solution Concept

How do we predict the outcome of a game? Use **solution concepts**.

a **solution concept** is a formal rule for predicting how a game will be played.

There are many, but the most important is **Nash equilibrium**

J. Nash (1928-2015)

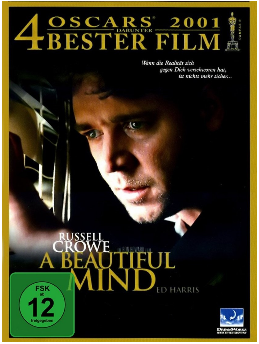
Solution Concept

How do we predict the outcome of a game? Use **solution concepts**.

a **solution concept** is a formal rule for predicting how a game will be played.

There are many, but the most important is **Nash equilibrium**

J. Nash (1928-2015)



Game Structure

A strategic/normal form is a structure:

$$(N, \Sigma_1, ..., \Sigma_n, u_1, ..., u_n)$$

where

- $N = \{1, ..., n\}$ is the set of **players**;
- Σ_i is set of possible **strategies** for player $i \in N$;
- $u_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{R}$ is the **utility function** for player $i \in N$.

Notice that the utility of player i depends not only on **her** actions, but on the **actions of others** (similarly for other agents). For player i to find **the best action** involves deliberating about what **others will do**, taking into account the fact that they will also try to maximise their utility taking into account how player i will act.

Eloise & Abelard Unstable Game

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_E	Win_A

Eloise & Abelard Unstable Game

		E	
		a	*b*
A	*c*	1, 0	0, 1
	d	0, 1	1, 0

Eloise & Abelard Unstable Game

		E	
		a	b
A	c	1, 0	0, 1
	d	0, 1	1, 0

- $N = \{Eloise, Abelard\}$

		E	
		a	b
A	c	1, 0	0, 1
	d	0, 1	1, 0

- $N = \{Eloise, Abelard\}$
- $\Sigma_E = \{a, b\}, \Sigma_A = \{c, d\}$

Eloise & Abelard Unstable Game

		E	
		a	b
A	c	0 1	1 0
	d	1 0	0 1

- $N = \{Eloise, Abelard\}$
- $\Sigma_E = \{a, b\}, \Sigma_A = \{c, d\}$
- $$\begin{array}{llll} u_E(a,c) = 0 & u_E(a,d) = 1 & u_E(b,c) = 1 & u_E(b,d) = 0 \\ u_A(a,c) = 1 & u_A(a,d) = 0 & u_A(b,c) = 0 & u_A(b,d) = 1 \end{array}$$

Strategy Profiles

A **strategy profile** is a tuple of strategies, one for each player:

$$\vec{\sigma} = (\sigma_1, ..., \sigma_i, ..., \sigma_n) \in \Sigma_1 \times \cdots \times \Sigma_i \times \cdots \times \Sigma_n$$

We denote the strategy profile obtained by replacing the ith component of $\vec{\sigma}$ with σ'_i by

$$(\vec{\sigma}_{-i}, \sigma'_i)$$

And so we have:

$$(\vec{\sigma}_{-i}, \sigma'_i) = (\sigma_1, ..., \sigma'_i, ..., \sigma_n)$$

(Pure Strategy) Nash Equilibrium

For a game $\mathcal{G} = (N, (\Sigma_i)_{i \in N}, (u_i)_{i \in N})$

a strategy profile $\vec{\sigma}$ is a **Nash equilibrium (NE)** if there is no player $i \in N$ and strategy $\sigma_i' \in \Sigma_i$ such that

$$u_i(\vec{\sigma}_{-i}, \sigma_i') > u_i(\vec{\sigma}).$$

A player **cannot benefit** by deviating from a Nash equilibrium.

Eloise & Abelard Unstable Game

		E	
		a	b
A	c	1, 0	0, 1
	d	0, 1	1, 0

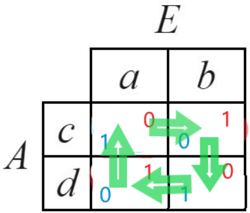
- $\vec{\sigma} = (a, c), u_A(a, c) = 1, u_E(a, c) = 0.$

Eloise & Abelard Unstable Game

		E	
		a	b
A	c	1, 0	0, 1
	d	0, 1	1, 0

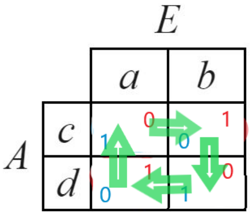
- $\vec{\sigma} = (a, c), u_A(a, c) = 1, u_E(a, c) = 0.$
- Eloise can benefit: $u_E(b, c) = 1$.

Eloise & Abelard Unstable Game



- $\vec{\sigma} = (a, c), u_A(a, c) = 1, u_E(a, c) = 0.$
- Eloise can benefit: $u_E(b, c) = 1$.
- For each $\vec{\sigma} \in \Sigma_E \times \Sigma_A$, there's always a **beneficial deviation** for a player.

Eloise & Abelard Unstable Game



- $\vec{\sigma} = (a, c), u_A(a, c) = 1, u_E(a, c) = 0.$
- Eloise can benefit: $u_E(b, c) = 1$.
- For each $\vec{\sigma} \in \Sigma_E \times \Sigma_A$, there's always a **beneficial deviation** for a player.
- There is **NO** NE.

Eloise & Abelard Stable Game

		E	
		a	b
A	c	Win_A	Win_E
	d	Win_A	Win_E

Eloise & Abelard Stable Game

		E	
		a	b
A	c	0 1	1 0
	d	0 1	1 0

$\vec{\sigma} = (b, c), \vec{\sigma}' = (b, d)$, both are NE.

The Prisoner's Dilemma

Abelard and Eloise are collectively charged with a crime and held in separate cells, with no way of meeting or communicating.

- if one confesses and the other does not, the confessor will be freed, and the other will be jailed for three years;
- if both confess, then each will be jailed for two years.
- Both know that if neither confesses, then they will each be jailed for one year.

The Prisoner's Dilemma

Abelard and Eloise are collectively charged with a crime and held in separate cells, with no way of meeting or communicating.

- if one confesses and the other does not, the confessor will be freed, and the other will be jailed for three years;
- if both confess, then each will be jailed for two years.
- Both know that if neither confesses, then they will each be jailed for one year.

		E defect	*E* coop
A	defect	-2 (A), -2 (E)	0 (A), -3 (E)
A	coop	-3 (A), 0 (E)	-1 (A), -1 (E)

The Prisoner's Dilemma

		E	
		defect	coop
A	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

The Prisoner's Dilemma

		E	
		defect	coop
	defect	-2	-3
A		-2	0
	coop	0	-1
		-3	-1

- in (c, c), Eloise wants to deviate to (d, c)

The Prisoner's Dilemma

		E	
		defect	coop
A	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

- in (c, c), Eloise wants to deviate to (d, c)
- Abelard wants to deviate from (d, c) to (d, d)

The Prisoner's Dilemma

		E	
		defect	coop
A	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

- in (c, c), Eloise wants to deviate to (d, c)
- Abelard wants to deviate from (d, c) to (d, d)
- Simmetric reasoning if Abelard deviates first from (c, c)

The Prisoner's Dilemma

		E	
		defect	coop
A	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

- in (c, c), Eloise wants to deviate to (d, c)
- Abelard wants to deviate from (d, c) to (d, d)
- Simmetric reasoning if Abelard deviates first from (c, c)
- (d, d) is the NE.

Outline

Bad Equilibrium

		E	
		defect	coop
A	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

- Previously, we looked at the Prisoner's Dilemma game.
- The NE is "bad".
- This kind of game happens in real life: e.g., nuclear arms reduction, CO_2 reduction, doping in sport.
- Can we achieve cooperation?

Arguments for Cooperation

- We are altruistic

Arguments for Cooperation

- We are altruistic
- Abelard and Eloise care about each other

Arguments for Cooperation

- We are altruistic
- Abelard and Eloise care about each other
- The shadow of the future...

Arguments for Cooperation

- The shadow of the future...

The Iterated Prisoner's Dilemma

- Play PD more than once
- If you know you will be meeting the other person again, would you still want to defect?

Finitely Repeated Prisoner's Dilemma

- suppose you both know that you will play the game exactly n times.
- What should you do? Imagine yourself playing the **final round**.
- In the final round you would want to defect to gain that extra bit of payoff
- Then the round $n-1$ is the last "real" round, and you want to defect there too, and so on...
- This is **backward induction**

Theorem

Iterated PD with a fixed, finite, pre-determined, commonly known number of rounds, has one NE: defection at every step.

Infintely Repeated Games

- Suppose you play the game an infinite number of rounds
- How to measure utility over infinite plays?
  Sums to infinity does not work.
- How to model strategies?
  Need to define strategies for infinitely many rounds.

Utility Functions for Infinite Runs

- **Limit of means**: computing the average payoff over the infinite run
- For a given infinite run

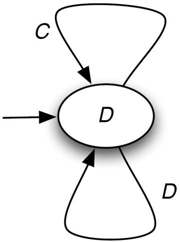
$$\omega_0\omega_1\omega_2\cdots\omega_k\cdots$$

where $\omega_k \in \Sigma_1 \times \cdots \times \Sigma_n$, the value of such a run for player i is

$$\lim_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} u_i(\omega_k)$$

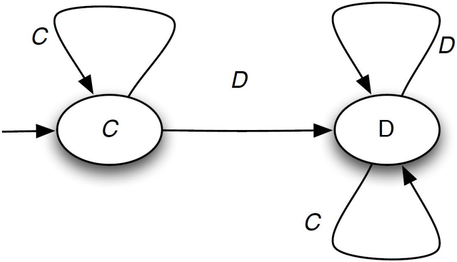
- The value is not always well defined. But if we represent strategies as **deterministic finite automata**, then we have well defined value.

Strategies as Automata



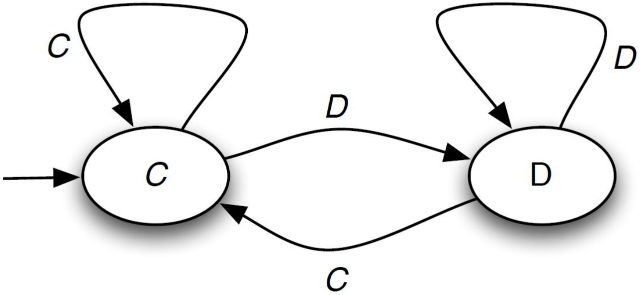
- We represent strategies as deterministic finite automata (transducers)
- Example above is an automaton strategy "ALLD", which always defects.
- Value inside a state is the action selected; edges are actions of other player.

The GRIM Strategy



I cooperate until you defect, at which point I flip to punishment mode: I defect forever after.

The TIT-FOR-TAT Strategy



What does this strategy do?

Automaton vs Automaton

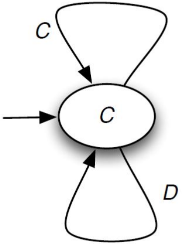
Theorem

Deterministic finite automata playing against each other will eventually enter a finite repeating sequence of outcomes, i.e., the resulting run will be

$$\alpha \cdot \beta^{\omega}$$

where α, β *are regular expressions and* $^{\omega}$ *is the infinite iteration operator. The average utility of an infinite run is the average utility of the finite sequence* β*.*

ALLC vs ALLC

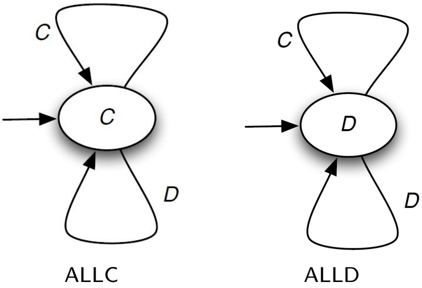


ALLC: cooperate forever

round:	0	1	2	3	4	⋯	
ALLC:	C	C	C	C	C	⋯	average utility = -1
ALLC:	C	C	C	C	C	⋯	average utility = -1

This is not a NE: either player would do better to choose another strategy (e.g., ALLD).

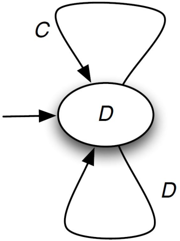
ALLC vs ALLD



round:	0	1	2	3	4	⋯	
ALLC:	C	C	C	C	C	⋯	average utility = -3
ALLD:	D	D	D	D	D	⋯	average utility = 0

This is not a NE: ALLC would do better to choose another strategy (e.g., ALLD)

ALLD vs ALLD

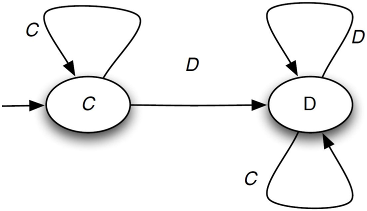


ALLD: defect forever

round:	0	1	2	3	4	⋯	
ALLD:	D	D	D	D	D	⋯	average utility = -2
ALLD:	D	D	D	D	D	⋯	average utility = -2

This is a NE (basically same as in one-shot case). But it is a bad one!

GRIM vs GRIM



round:	0	1	2	3	4	$\cdots$	
GRIM:	C	C	C	C	C	$\cdots$	average utility = -1
GRIM:	C	C	C	C	C	$\cdots$	average utility = -1

This is a NE! Rationally sustained cooperation.
The threat of punishment keeps players in line.

Nash Folk Theorem

Define **security value** as the best utility that player i can guarantee in a game, no matter what other players do.

Theorem

In an infinitely repeated game, every outcome in which every player gets at least their security value can be sustained as a Nash equilibrium.

Corollary

In the infinitely repeated Prisoners Dilemma, mutual cooperation can be sustained as an equilibrium.

Nash Folk Theorem

Define **security value** as the best utility that player i can guarantee in a game, no matter what other players do.

Theorem

In an infinitely repeated game, every outcome in which every player gets at least their security value can be sustained as a Nash equilibrium.

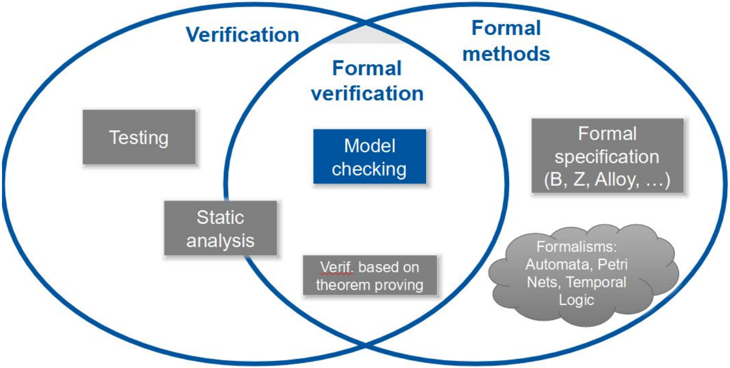
Corollary

In the infinitely repeated Prisoners Dilemma, mutual cooperation can be sustained as an equilibrium.

Single shot and repeated games may have different sets of NE!

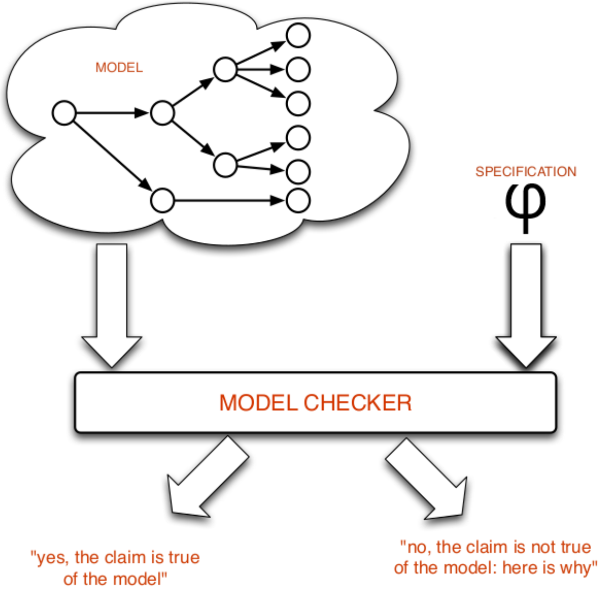
Outline

Formal Verification



Fernndez, Darvaz, Blanco (2016)

Model Checking



Model Checking As Games

- Model checking problems can be cast as strategy problems for Hintikka games.
- Played by two players: **Verifier** (Eloise) and **Falsifier** (Abelard)
- "Eloise has a strategy such that for all Abelard's strategies, the specification is true in the model".
- For systems that run "**forever**" we use **infinitely repeated games** and appropriate logic (e.g. **LTL**, CTL)

Propositional Linear Temporal Logic (LTL)

A standard language for talking about **infinite state sequences**.

$\top$	truth constant
p	atomic propositions
$\neg\varphi$	negation
$\varphi \vee \psi$	disjunction
$\mathbf{X}\varphi$	in the next state...
$\mathbf{F}\varphi$	will eventually be the case that φ
$\mathbf{G}\varphi$	is always the case that φ
$\varphi\,\mathbf{U}\,\psi$	always the case φ until ψ

Example

$\mathbf{F} \neg \mathit{sleepy}$

Example

$\mathbf{F}\neg \mathit{sleepy}$

eventually I will not be sleepy (a **liveness** property)

Example

$$\mathbf{G}\neg \mathit{crash}$$

Example

$$\mathbf{G}\neg \mathit{crash}$$

the program will never crash (a **safety** property)

Example

GF*eatRice*

Example

$\mathbf{GF}\,\mathit{eatRice}$

I will eat rice **infinitely often**

Example

$\mathbf{FG}\neg \mathit{alive}$

Example

$$\mathbf{FG}\neg \mathit{alive}$$

eventually will come a time at which I am not alive forever after

Example

$$(\neg \mathit{takeExam})\ \mathbf{U}\ \mathit{zulassung}$$

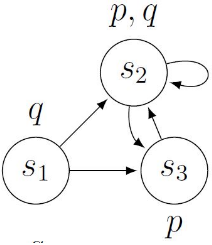
Example

$$(\neg \mathit{takeExam})\ \mathbf{U}\ \mathit{zulassung}$$

you may not take Logic exam until you have a Zulassung

Model for LTL

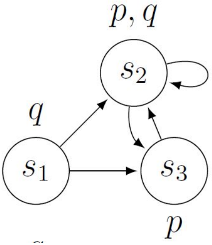
LTL formulae are usually interpreted in terms of Kripke structure.



- $\mathbf{X}p$ is true in s_1: $s_1 s_2$, $s_1 s_3$

Model for LTL

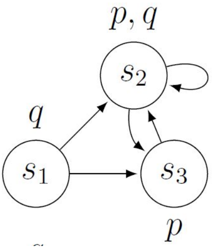
LTL formulae are usually interpreted in terms of Kripke structure.



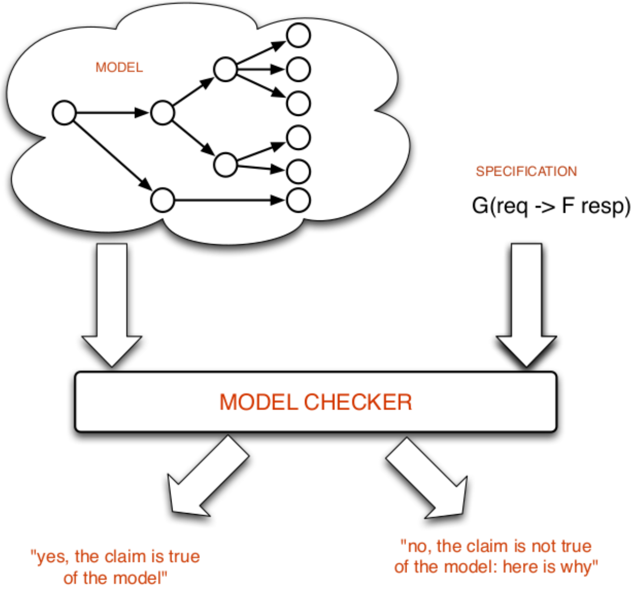
- $\mathbf{X}p$ is true in s_1: $s_1 s_2$, $s_1 s_3$
- $\mathbf{X}q$ is not true in s_1: $s_1 s_3$

Model for LTL

LTL formulae are usually interpreted in terms of Kripke structure.

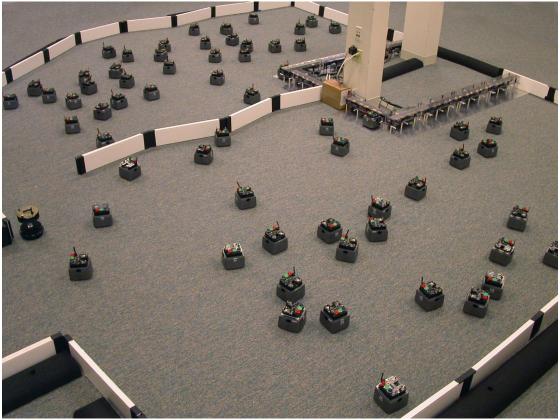


- $\mathbf{X}p$ is true in s_1: $s_1 s_2$, $s_1 s_3$
- $\mathbf{X}q$ is not true in s_1: $s_1 s_3$
- $\mathbf{FG}p$ is true in s_1: e.g., $s_1 s_2^\omega$, $s_1(s_2 s_3)^\omega$, $s_1(s_3 s_2)^\omega$, etc.


MODEL
SPECIFICATION
G(req -> F resp)
MODEL CHECKER
"yes, the claim is true
of the model"
"no, the claim is not true
of the model: here is why"

Multi-Agent Systems

Now consider a system composed of multiple entities (players/agents).
Each entity may have different (not necessarily conflicting) goal.



https://people.csail.mit.edu/jamesm/project-MultiRobotSystemsEngineering.php

Multi-Agent Systems

Now consider a system composed of multiple entities (players/agents).
Each entity may have different (not necessarily conflicting) goal.

Each autonomous car may have different destination

Correctness Problem

- How do we define correctness in multi agent systems?

Correctness Problem

- How do we define correctness in multi agent systems?
- Agents are rational

Correctness Problem

- How do we define correctness in multi agent systems?
- Agents are rational
- Agents pursue their interests strategically

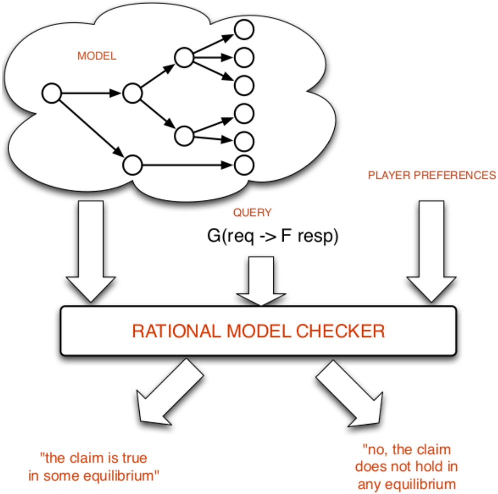
Correctness Problem

- How do we define correctness in multi agent systems?
- Agents are rational
- Agents pursue their interests strategically
- Some possible behaviour may not arise

Correctness Problem

- How do we define correctness in multi agent systems?
- Agents are rational
- Agents pursue their interests strategically
- Some possible behaviour may not arise
- We need to **predict** the behaviour of the systems $\Rightarrow$ Nash equilibrium

Equilbrium Checking



We take into the account of player preferences

Multi-Agent Systems as Games

- Multi-agent systems modelled as multi-player games.
- Games are played on graph-like (Kripke structure) arena:

$$A = (N, \mathrm{Ac}, \mathrm{St}, s_0, \mathrm{tr}, \lambda)$$

- N (finite) set of agents;
- Ac (finite) set of actions;
- St (finite) set of states (s_0 initial state);
- $\mathrm{tr} : \mathrm{St} \times \mathrm{Ac}^{\mathrm{N}} \to \mathrm{St}$ transition function [a];
- $\lambda : \mathrm{St} \to 2^{\mathrm{AP}}$ labelling function.

[a] At every state, agents take actions concurrently and move to the next state

Strategies

Strategy

Finite state automaton $\sigma = \langle Q, \mathsf{St}, q_0, \delta, \tau \rangle$

- Q, internal state (q_0 initial state);
- $\delta : Q \times \mathsf{St} \to Q$ internal transition function;
- $\tau : Q \to \mathsf{Ac}$ action function.

A strategy is a recipe for the agent prescribing the action to take at every time-step of the game execution.

Play

Given a strategy assigned to every agent in A, denoted $\vec{\sigma}$, there is a unique possible execution $\pi(\vec{\sigma})$ called play.

Note that plays can only be ultimately periodic, i.e., of the form $\alpha \cdot \beta^{\omega}$

Games and Nash Equilibria

- A games is given by $\mathcal{G} = (A, \gamma_1, ..., \gamma_n)$, where γ_i is the goal of player i in LTL formula.
- For a game $\mathcal{G}$, strategy profile $\vec{\sigma}$ is a **Nash equilibrium** if there is no player i and strategy σ'_i such that

$$\pi(\vec{\sigma}) \models \neg\gamma_i \implies \pi((\vec{\sigma}_{-i}, \sigma'_i)) \models \gamma_i$$

A player **cannot benefit** by deviating from a Nash equilibrium.

Rational Verification[**]

Non-Emptiness

Given: a game $\mathcal{G}$
Question: Does NE exist in $\mathcal{G}$?

Is the game stable?

[**] Wooldridge et al. "Rational Verification: From Model Checking to Equilibrium Checking". In: AAAI. 2016.

Rational Verification**

Non-Emptiness

Given: a game $\mathcal{G}$
Question: Does NE exist in $\mathcal{G}$?

Is the game stable?

E-Nash

Given: a game $\mathcal{G}$ and a LTL formula φ
Question: Is there any NE that satisfies φ?

Is there any NE in which cars ignore traffic lights?

**Wooldridge et al. "Rational Verification: From Model Checking to Equilibrium Checking". In: AAAI. 2016.

Tools

- **EVE** (http://eve.cs.ox.ac.uk/)

Other related tools:

- **MCMAS** (https://vas.doc.ic.ac.uk/software/mcmas/): memoryless strategies
- **PRALINE**: Büchi-defineable goals, instead of LTL
- **PRISM-games** (https://www.prismmodelchecker.org/games/): stochastic games with CTL goals.

Outline

- Probabilistic Systems $\Rightarrow$ stochastic games (PRISM-games[††], Probabilistic Strategy Logic[‡])
- What if the players can cooperate? cooperative games (other solution concept: e.g., CORE[‡‡])
- Repairing games: designing equilibria, instead of just verifying equilibria, we want to introduce desired ones [*] [†].

[††] Kwiatkowska et al., (2020), PRISM-games 3.0: Stochastic Game Verification with Concurrency, Equilibria and Time.

[‡] Kwiatkowska et al., (2019), Probabilistic Strategy Logic

[‡‡] Gutierrez, Kraus, Wooldridge, (2019), Cooperative Concurrent Games.

[*] Almagor, Avni, Kupferman, (2015), Repairing Multi-Player Games

[†] Gutierrez et al., (2019), Equilibrium Design for Concurrent Games.