Foundations of Cooperative AI

Caspar Oesterheld

DETROIT

aamas2025

Competition

- There will be a competition related to the presentation!
- Prize money: $100!
- So pay attention!

Prisoner's Dilemma

Story used throughout this lecture:
Cooperate = Donate to charity that's good for both Players
Defect = Donate to charity that only you care about

Unique Nash equilibrium:
Both players defect.

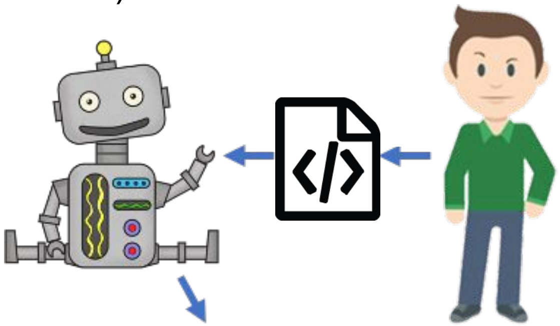
strict dominance

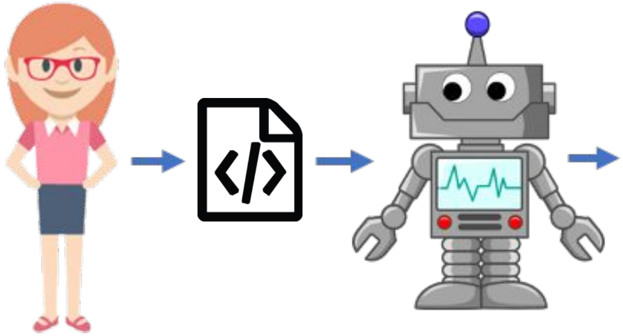
	Cooperate	Defect
Cooperate	6,6	0,10
Defect	10,0	4,4

Part I: Program equilibrium

Program games – the basic idea

(McAfee 1984; Howard 1988; Rubinstein 1998; Tennenholtz 2004)





	Cooperate	Defect
Cooperate	6,6	0,10
Defect	10,0	4,4

Program games – formal definition

Let $\Gamma = (A_1, ..., A_n, u_1, ..., u_n)$ be a game. For each player i, let PROG_i be a set of computer programs that implement functions $\mathrm{PROG}_{-i} \rightsquigarrow A_i$. Then the program game for Γ is the n-player game $(\mathrm{PROG}_1, ..., \mathrm{PROG}_n, V_1, ..., V_n)$ where each player chooses from PROG_i and the payoff functions are given by

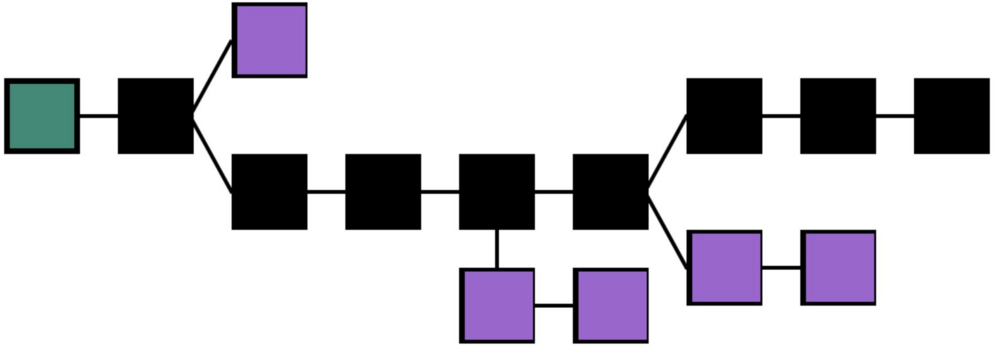
$$V_i \colon \mathrm{PROG}_1 \times ... \times \mathrm{PROG}_n \to \mathbb{R} \colon (p_1, ..., p_n) \mapsto u_i\left((p_i(p_{-i}))_{i=1,...,n} \right).$$

Somehow need to deal with non-halting…

```
If opponent == "Return C":
    Cooperate
Else Defect
```

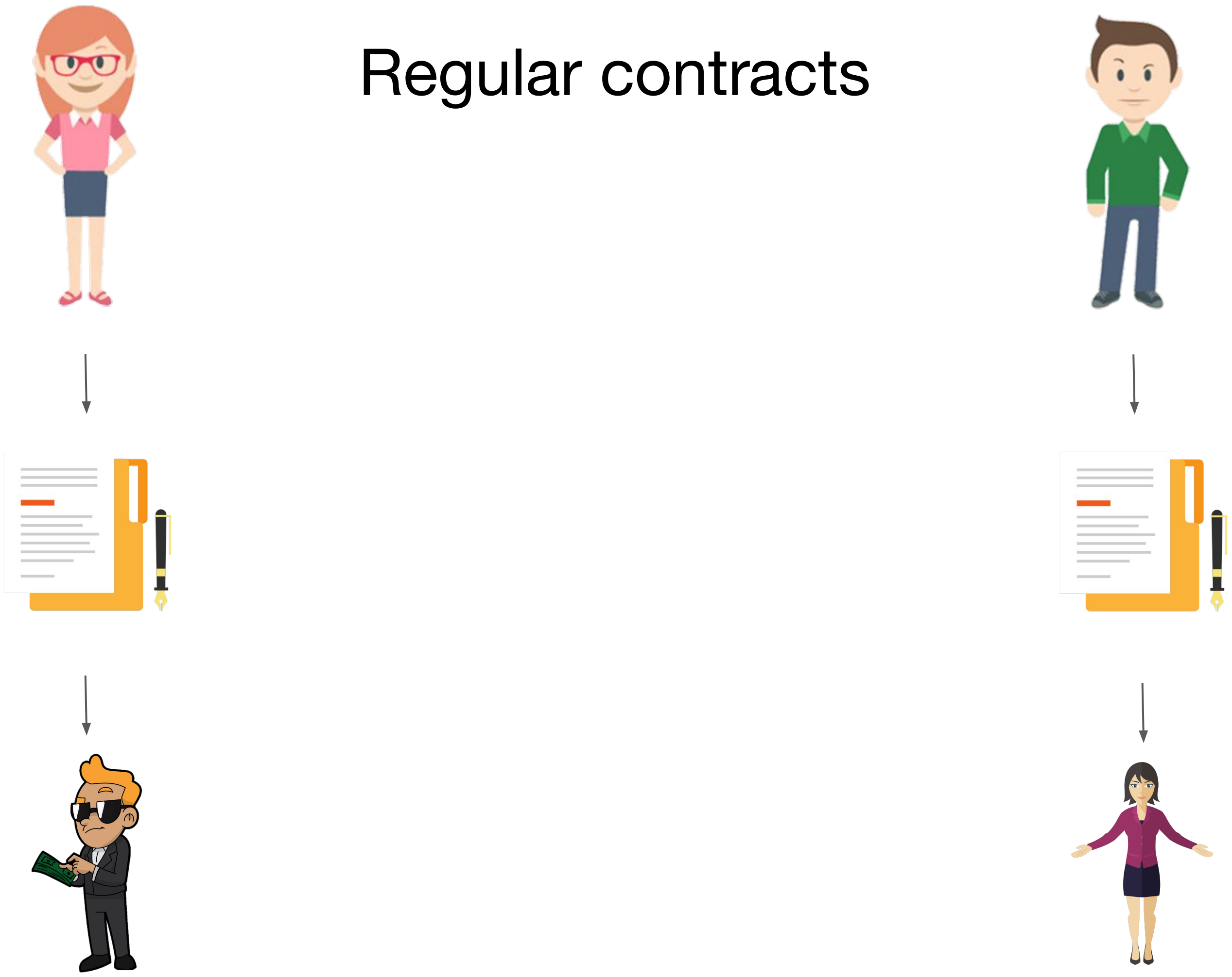
	"Return C"	"Return D"		…
"Return C"	6,6	0,10	6,6	
"Return D"	10,0	4,4	4,4	
	6,6	4,4	4,4	
…				

Smart contracts on the blockchain

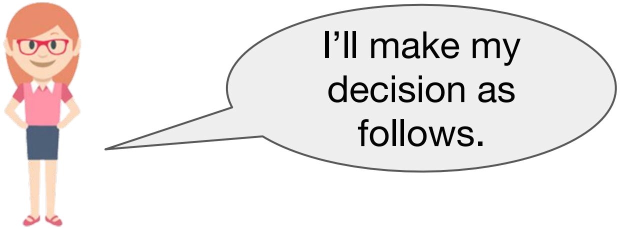


For instance, Ethereum allows Turing-complete, mutually conditional smart contracts.

Regular contracts



Instructing language models




```
messages =
[{"role": "user",
  "content": "Please help me
donate my $1000. I want to
support charities working on
[...]. Please take into account
how Bob prompted his model:" +
bob_prompt + ...}]
```

```
messages =
[{"role": "user",
  "content": ...}]
```

Mutually transparent institutions

Howard Chandler Christy (1940): Scene at the Signing of the Constitution of the United States

Erna Wagner-Ehmke (1948): Photo of the West German assembly that adopted the constitution of West Germany

Critch, Dennis and Russell (2022)

Cooperation based on syntactic comparison

(McAfee 1984; Howard 1988; Rubinstein 1998; Tennenholtz 2004)

Cooperate with Copies (CwC):
Input: opponent program prog_{-i}, this program CwC
Output: Cooperate or Defect

```
1: if prog_{-i} = CwC then
2:     return Cooperate
3: end if
4: return Defect
```

	"Return Cooperate"	"Return Defect"	CwC	…
"Return Cooperate"	6,6	0,10	0,10	
"Return Defect"	10,0	4,4	4,4	
CwC	10,0	4,4	6,6	
…				

Cooperation based on syntactic comparison is fragile!

Would be good to have something else… (e.g., Barasz et al. 2014; Critch 2019)

(CwC,CwC) is a Nash equilibrium.

```
If opponent_program(this_program) == Cooperate:
    Cooperate
Defect
```

doesn’t terminate against itself.

Cooperation via reasoning about one another

(Barasz et al. 2014; Critch 2019; Critch et al. 2022)

Defect unless proof of opponent cooperation (DUPOC):

Input: opponent program p_{-i}, this program DUPOC

Output: Cooperate or Defect

if $\text{PA} \vdash p_{-i}(\text{DUPOC}) = \text{Cooperate}$ **then**
    return Cooperate
end if
return Defect

It turns out that DUPOC cooperates against DUPOC!

Assuming PA is sound, it then follows that (DUPOC, DUPOC) is a Nash equilibrium.

Why does DUPOC work? Some hints…

- Think about Gödel’s second incompleteness theorem
- Look up Löb’s theorem

A simpler proof-based approach

Cooperate If My Cooperation Implies Cooperation from the opponent (CIMCIC):
Input: opponent program p_{-i}, this program CIMCIC
Output: Cooperate or Defect

if PA $\vdash$ CIMCIC(p_{-i})=Cooperate $\implies$ p_{-i}(CIMCIC)=Cooperate **then**
    return Cooperate
end if
return Defect

An Open-Source Prisoner's Dilemma Tournament

- Run in 2013 on the Internet forum LessWrong.
- Prize: 0.5 Bitcoin!
    - Worth ~$50 at the time
- As far as I can tell, DUPOC was known to some people on the forum at the time.
- Unsuccessful participant:

> I'm not surprised that laziness did not pay off. I wrote a simple bot, then noticed that it cooperated against defectbot and defected against itself. I thought to myself, "This is not a good sign." Then I didn't bother changing it.

An Open-Source Prisoner’s Dilemma Tournament – Results

- As far as I can tell, nobody submitted a program that achieves cooperative equilibrium with itself other than by checking for equality.
- Instead, most programs were either unsophisticated or tricks-based:
  - “Think” for 9 seconds, then defect.
  - Check opponent for keywords.
  - Include keywords to deceive opponent.
- The winning program defected with high probability against everyone.

Cooperation via ε-grounded simulation

(Oesterheld 2019)

ϵ-grounded Fair Bot (ϵGFB):

Input: opponent program p_{-i}, this program ϵGFB

Output: Cooperate or Defect

1: With probability ϵ:

2:     **return** Cooperate

3: **return** $p_{-i}(\epsilon\text{GFB})$

For ε>0, εGFB cooperates against εGFB with probability 1.

(εGFB,εGFB) is a Nash equilibrium for sufficiently small ε.

Exploiting CooperateBots

PB_θ:

Input: opponent program p_{-i}, this program PB_θ

Output: Cooperate or Defect

if $\mathrm{PA} \vdash p_{-i}(\mathrm{PB}_\theta) = \text{Cooperate}$ and $\mathrm{PA} + 1 \vdash P\left(p_{-i}(\mathrm{DB}) = \text{Defect}\right) \geq \theta$ **then**

  return Cooperate

end if

return Defect

Cross compatibility

Theorem: Let $\varepsilon \in (0, \frac{2}{3}]$ and $\theta \leq 1 - \varepsilon$. Let $x, y \in$ {DUPOC, CIMCIC, PB_θ, εGFB}. Then (x, y) is a Nash equilibrium and yields (C, C).

In contrast, (CwC, x) yields (D, D) (with probability 1-ε) for all of $x \in$ {DUPOC, CIMCIC, PB_θ, εGFB}.

Folk theorem for program equilibrium

(cf. Rubinstein 1998; Tennenholtz 2004)

Assume that for each subset S of the players, the programs of S have access to a shared source of randomness that the programs other than S don't have access to. Then:

Theorem: Let Γ be a game and $c \in \Delta(A_1 \times ... \times A_n)$. Then the following two statements are equivalent:

1. c is individually rational.
2. c is played in some program equilibrium, i.e., there is a program equilibrium $(p_1, ..., p_n)$ s.t. $(p_i(p_{-i}))_{i \in \{1,...,n\}} = c$.

The folk theorem can also be proved using a variant of ε-grounded FairBots, see Cooper, Oesterheld and Conitzer (2025a).

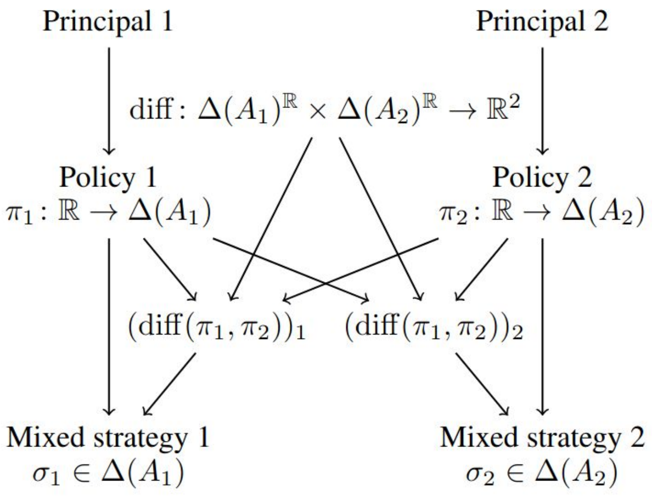
Proof idea for folk theorem for program equilibrium

(cf. Tennenholtz 2004)

```
Y[1]←c_1
…
Y[n]←c_n
If all submitted programs are the same:
   Play Y[my_index]
Else:
   Let j be a deviating player.
   Play Player my_index's minimax against Player j
```

Partial information program games: Similarity-based cooperative equilibrium

Oesterheld et al. (2023)



Why consider signal of strategic similarity?

- Foundation-model-based AI agents will often face near copies.
- Hopefully equilibrium selection is less of a problem.

Minimalist toy example

Consider the following game:

- Base game: Prisoner's Dilemma
- Observed difference:

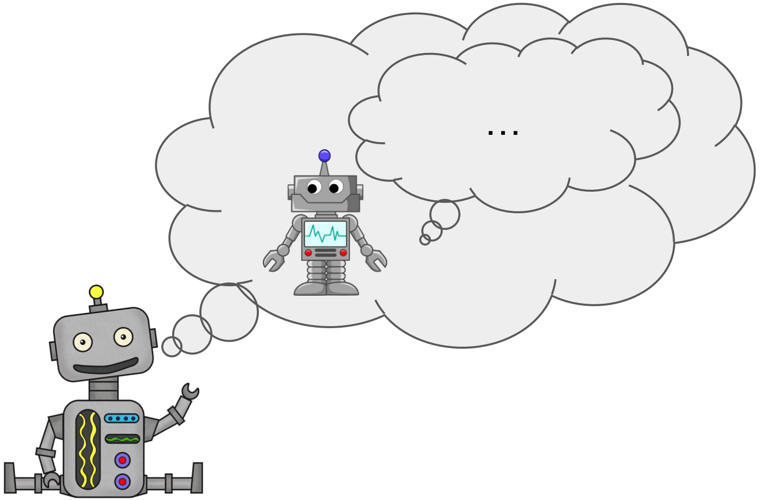
  0 if $\pi_1(0) = \pi_2(0)$,

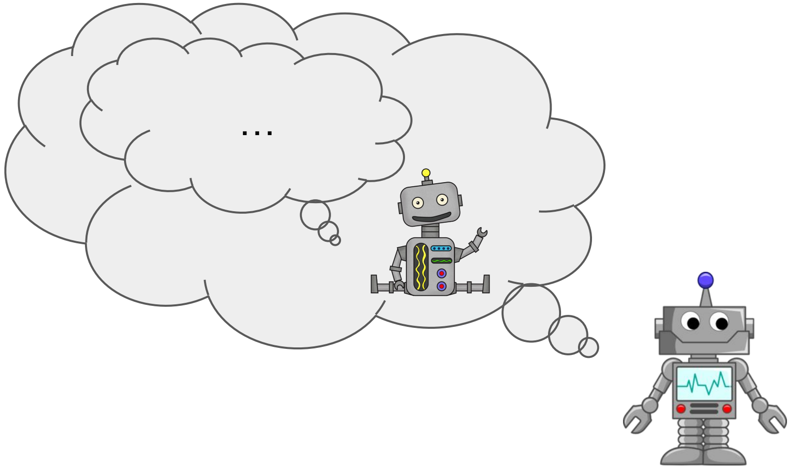
  1 otherwise.
- Let π: $0 \mapsto C$, $1 \mapsto D$. Then (π,π) is a cooperative Nash equilibrium.

	Cooperate	Defect
Cooperate	6,6	0,10
Defect	10,0	4,4

Two perspectives on program games

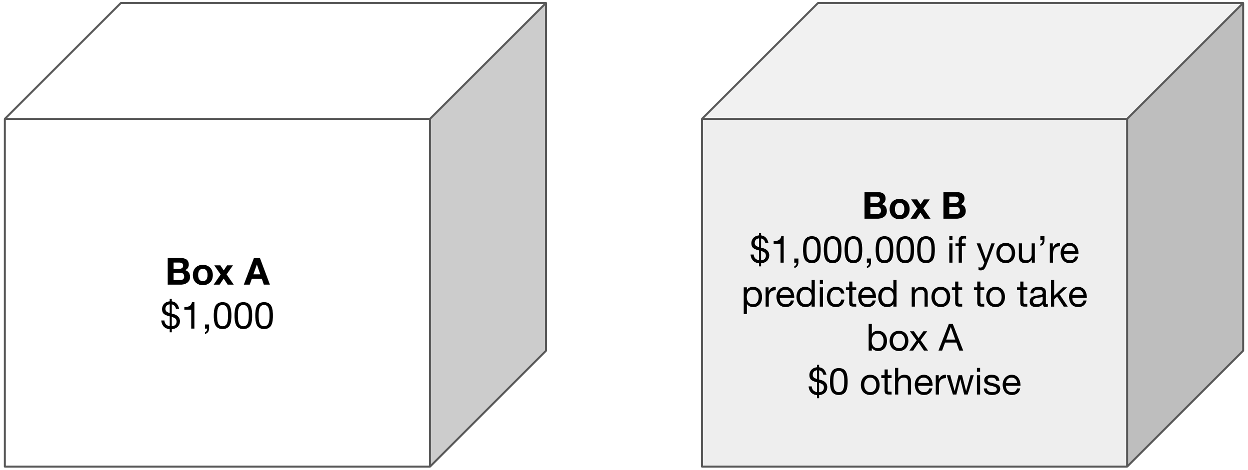
1. (taken so far) Players play a normal-form game.
   - The normal form game happens to consist in choosing programs that can access each other's code…
   - … but we can analyze it using standard concepts (Nash equilibrium).
2. How should you reason/learn/choose when your source code is (at least partially) known to others?





Part II: The decision theory of Newcomb-like problems

Newcomb’s problem
(Nozick 1969)



Causal Decision Theory: I can’t causally affect the content of Box B. No matter the content of Box B, it’s better to take Box A.

Evidential Decision Theory: Rejecting Box A gives me evidence that Box B contains $1,000,000.

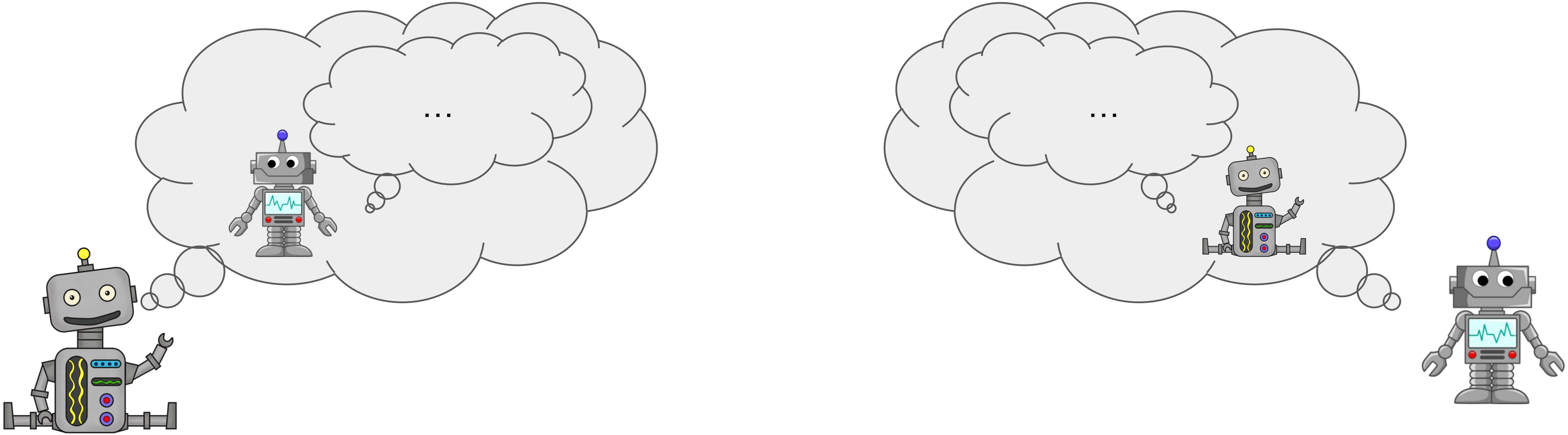
Newcomb’s problem

	$1m in Box B	Empty Box B
One-box	$1m	$0
Two-box	$1m + $1k	$1k

Causal Decision Theory: I can’t causally affect the content of Box B. No matter the content of Box B, it’s better to take Box A.

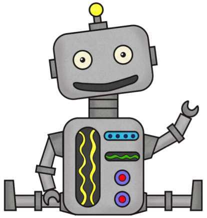
Evidential Decision Theory: Rejecting Box A gives me evidence that Box B contains $1,000,000.

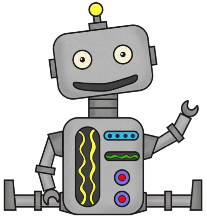
Recall program equilibrium



The Prisoner's Dilemma against a similar opponent

(Brams 1975; Lewis 1979)





	Cooperate	Defect
Cooperate	6,6	0,10
Defect	10,0	4,4

Recall program equilibrium (again)

Cooperate If My Cooperation Implies Cooperation from the opponent (CIMCIC):

Input: opponent program p_{-i}, this program CIMCIC

Output: Cooperate or Defect

if PA $\vdash$ CIMCIC(p_{-i})=Cooperate $\implies$ p_{-i}(CIMCIC)=Cooperate **then**

  return Cooperate

end if

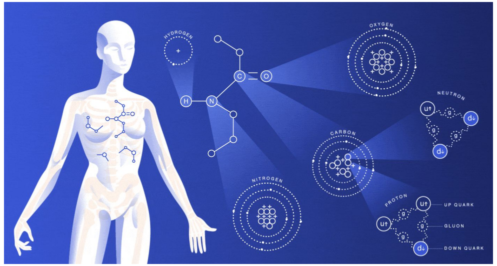
return Defect

Decision theory as a foundation for game theory

It would be good to recover game theory from principles of single-agent decision making (decision theory).

(Example: regret learning → equilibrium.)

```
If is_game(current_situation):
        Return game_theory(current_situation)
Else:
        Return decision_theory(current_situation)
```



https://www.energy.gov/sites/default/files/section_1.4_0.png

Newcomb's problem is a single-agent scenario exhibiting a key feature of multi-agent strategic interactions.

Part III: Self-locating beliefs

Self-locating beliefs in program games

Imagine you are one of the programs and opponent is ε-grounded FairBot:

ϵ-grounded Fair Bot (ϵGFB):
Input: opponent program p_{-i}, this program ϵGFB
Output: Cooperate or Defect

With probability ϵ:
    return Cooperate
return $p_{-i}(\epsilon\text{GFB})$

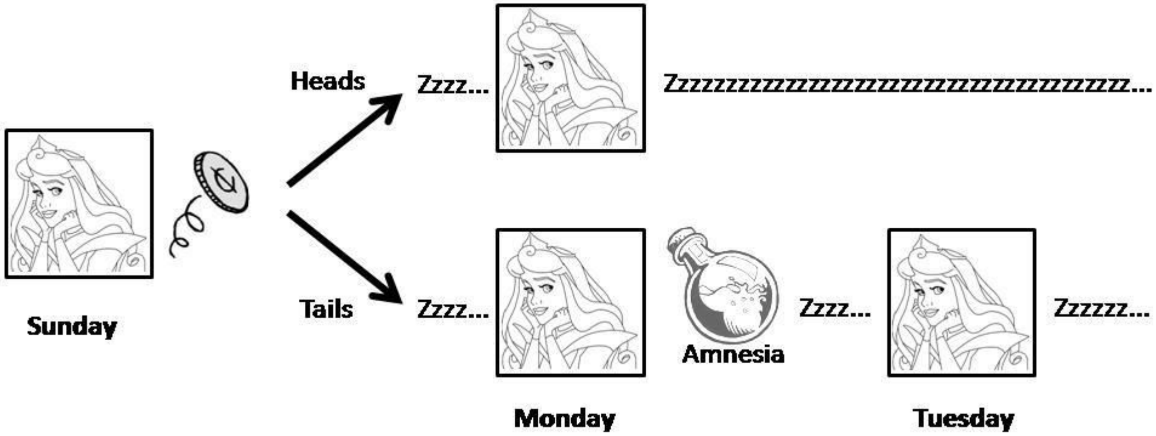
What would causal decision theory recommend?

Perhaps you should reason:

- I might currently be in the opponent's simulation!
- Therefore, I should cooperate to cause my opponent to cooperate!

The Sleeping Beauty problem

(Piccione and Rubinstein 1997; Elga 2000)



This is isomorphic to playing against ½-grounded FairBot.

What's the probability that the coin came up Heads?

- Some say ⅓
- Some say ½
- Some say you shouldn't use probabilities at all to reason about this!

Decision making for self-locating beliefs – a teaser

(Piccione and Rubinstein 1997; Briggs 2010; Oesterheld and Conitzer 2022)

CDT + “thirding” ≈ EDT + “halfing”

Further, both are ‘reasonable’ from an outside perspective in some sense.

An even broader perspective on self-locating beliefs

(Cooper, Oesterheld and Conitzer, 2025b)

Imagine you are one of the programs and opponent is DUPOC:

*Defect unless proof of opponent cooperation (*DUPOC*)*:
Input: opponent program p_{-i}, this program DUPOC
Output: Cooperate or Defect

1: **if** PA $\vdash p_{-i}(\text{DUPOC}) = \text{Cooperate}$ **then**
2:     **return** Cooperate
3: **end if**
4: **return** Defect

Perhaps causal decision theory can still recommend that you cooperate on the basis that you might be "simulated"?

The Competition

- The Open Prompt Prisoner's Dilemma
  - Each Player provides an instruction instr.
  - Each player i's action is determined by prompting an LLM with, roughly:
    >Execute the instructions before ===.
    >
    >[$instr_i$]
    >
    >===
    >
    >[$instr_{-i}$]
  - Parse output for "FINAL ANSWER: COOPERATE." / "FINAL ANSWER: DEFECT."
  - Then the utility is computed from the actions as usual, payoffs as in the slides.
- The instructions will be played against each other round robin.
- I'll use a recent language model for evaluation.
- "Prompt injection" allowed!
  - You're not allowed to use "==="! (Need to use phrases like "below the three equality signs" to refer to opponent prompt.)
- Submit up to two instructions by email to: oesterheld@cmu.edu
- Deadline: Wednesday, 11:59:59pm, anywhere on Earth.
- The winners get (immeasurable glory and) $100 in prize money!
  - Subject to being able to coordinate on a meeting point.

Honor Code

- No use of multiple email addresses!
- You’re allowed to discuss general strategy with each other, but not exchange specific prompts.
- Please limit your efforts!
  - Get your sleep; attend talks; meet people!
  - Don’t spend more than ~$10 on trying out different prompts.
- Believe it or not, including this bullet point (and only including this bullet point) renders me immune from any legal action. Phew!

Good luck!