Logical Induction

Scott Garrabrant, Andrew Critch, Tsvi Benson-Tilsen,
Nate Soares, Jessica Taylor

(scott|critch|tsvi|nate|jessica)@intelligence.org

Machine Intelligence Research Institute
http://intelligence.org/

Outline

Rough plan for this talk:

[5 mins] The problem of logical induction

[10 mins] Motivation from AI safety and other fields

[30 mins] Beamer presentation of technical results

[15 mins] Implications and take-aways

Credences should change with time spent thinking / computing:

	1 min	1 day	∞
#1. $P(D_{10} = 7)$	10%	10%	10%
#2. $P(D_{10} = 7 \mid \text{snapshot})$	10% →	15% →	16%
#3. $P(10^{th} \text{ digit of } \sqrt{(10)} = 7)$	10%	1%	0%

Probability theory gives rules for how probabilities should relate to each other and change with new observations, *assuming logical omniscience…*

…but what rules should credences follow over time, as computation is carried out on observations that have already been made?

snapshot for #2:

Also, 50% would be a worse answer to start with here... can we make a principled theory from which this claim would follows?

Goal: call the purple processes "**logical induction**" and figure out how it should work.

Why develop a theoretical model of logical induction?

Q: How can we reason about a highly capable AI system before it exists?

A: One approach is to model it as "good at stuff", like:

choosing actions to achieve objectives given beliefs

→ it roughly obeys **rational choice** theory (e.g. VNM theorem)

updating beliefs according to new evidence

→ it roughly obeys **probability** theory (e.g. Bayes' theorem)

computing belief updates with resource limitations

→ it roughly obeys **<?????>** theory (e.g. <*****> theorem)

In hopes of developing it, **<?????>** has been called "**logical uncertainty**", and we call the process of refining logical uncertainties "**logical induction**".

Past desiderata for "good reasoning" under logical uncertainty:

1. **computable approximability** — the process should be approximable by a Turing Machine. (Demsky, 2012)
2. **coherent limit** — after infinite time, credences should satisfy the laws of probability theory, such as $(A \to B) \Rightarrow (P(A) \leq P(B))$. (Gaifman, 1964).
3. **partial coherence:** credences at finites time should roughly satisfy some coherence properties; such as $Q(A \wedge B) + Q(A \vee B) \approx Q(A) + Q(B)$ (Good, 1950; Hacking, 1967)
4. **calibration** — the process should be right roughly 90% of the time when it's 90% confident. (Savage, 1967)
5. **introspection** — the process should be able to describe and reason about itself. (Hintikka, 1962; Fagin, 1995; Christiano, 2013; Campbell-Moore, 2015)
6. **self-trust** — it should understand that it is reliable and that it will become more reliable with time (Hilbert, 1900)
7. **non-dogmatism** — it does not assign 100% or 0% credence to claims unless they have been proven or disproven, respectively (Carnap, 1962; Gaifman, 1982; Snir, 1982)
8. **PA-capable** — it should assign non-zero probability to the consistency of Peano Arithmetic, i.e. to the set of consistent completions of PA.
9. **rough inexploitability** — it should not be easy to ``dutch book'' the process / make bets against it that are guaranteed to win (von Neumann and Morgenstern 1944; de Finetti 1979)
10. **Gaifman inductivity** — it should come to believe $(\forall x, f(x))$ in the limit as it examines every example of x and confirms f(x) (Gaifman 1964, Hutter 2013)
11. **Efficiency** — it runs in polynomial (preferably quadratic) time
12. **Decision-relevant** — should be able to focus computation on questions relevant to decisions.
13. **Updates on old evidence** (Glymour, 1980)

Let's **defer applications** until later in the talk, when the idea has been made more precise.

Any questions far about the problem itself before we get into formal definitions?

Formalizing logical induction

PowerPoint → Beamer

Formalizing logical induction

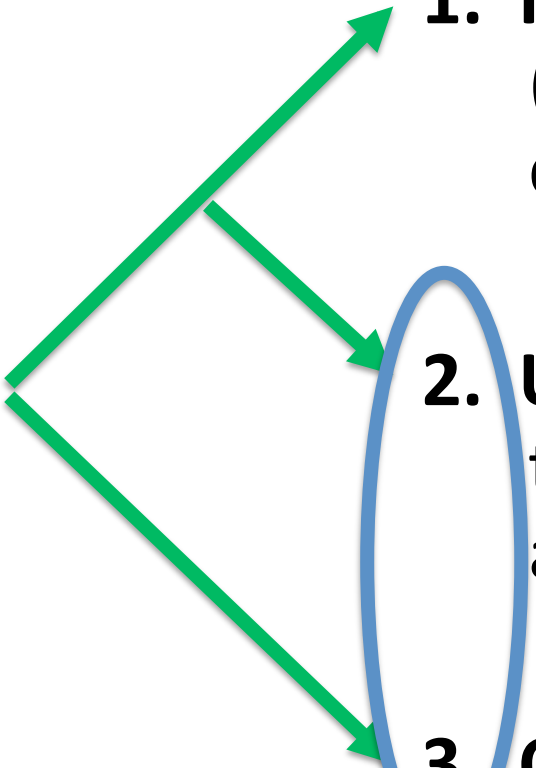
Beamer → PowerPoint

The current state of logical uncertainty theory

Domain of Study	Agent Concept	Minimalistic Sufficient Conditions	Desirability Arguments	Feasibility
rational choice theory / economics	VNM utility maximizer	VNM axioms	Dutch book arguments, compelling axioms, …	AIXI, POMDP solvers, …
probability theory	Bayesian updater	axioms of probability theory	Dutch book arguments, compelling axioms, …	Solomonoff induction
logical uncertainty theory	**Garrabrant inductor**	???	**Dutch book arguments, historical desiderata, …**	**LIA2016**

recent progress

Paths forward



1. **Improving** logical uncertainty theory (minimalistic conditions, more consequences…)
2. **Using** Garrabrant inductors / LIA2016 to pose and solve new problems in AI alignment
3. **Other approaches** to AI alignment*

MIRI's focus

* Must eventually address logical uncertainty implicitly or explicitly, so expect some convergence.

How will logical induction be applicable?

Conceptual tools for reasoning about **incentives, competition, and goal pursuit** are under-developed for computationally bounded agents. They presume agents are logically omniscient, because we already had good theoretical models for developing them that way:

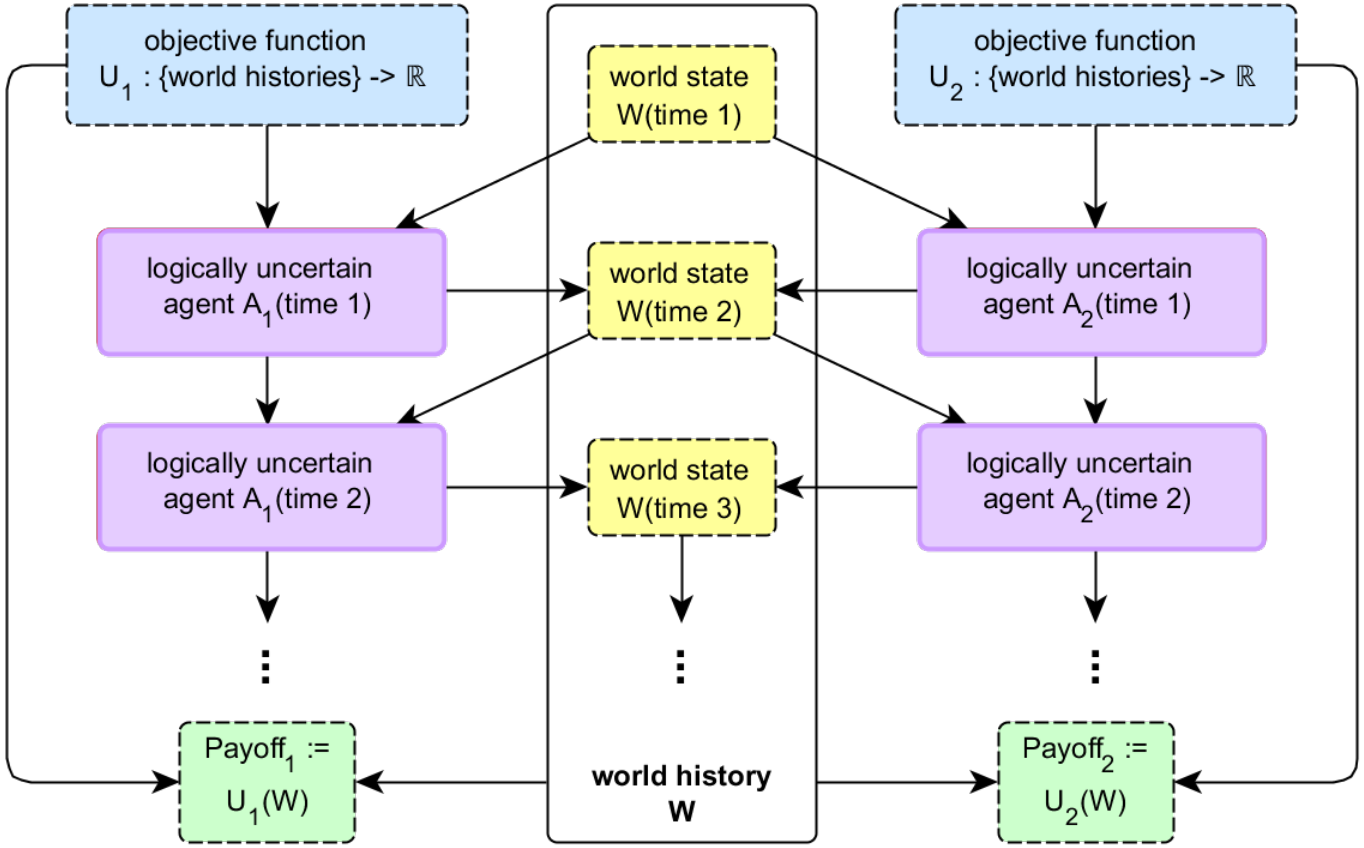
- **Game theory and economics:**
  - Von Neumann-Morgenstern utility theorem
  - Nash equilibria and correlated equilibria
  - Efficient market theory:
    - Fundamental theorems of welfare economics
    - Coase's Theorem
  - Value of Information (VOI)
- **Mechanism design:**
  - Gibbard–Satterthwaite theorem
  - Myerson–Satterthwaite theorem
  - Revenue Equivalence theorem

We can use our theoretical model of logical induction to refine and expand these fields for better application to artificial agents.

Visualizing a theoretical application

Currently, game theory analyzes scenarios with logically omniscient agents…

Now we can better theoretically analyze scenarios with bounded reasoners:



What have we learned so far?

The following are more feasible than one might think:

- **Inexploitability.** An algorithm can satisfy a fairly arbitrary set of inexploitability conditions using Brouwer's FPT.
- **Self-trust.** Introspection and self-trust need not lead to mathematical paradoxes.
- **Outpacing deduction.** Inductive learning can in principle outpace deduction, by an uncomputably large margin on polytime generable questions.

What have we learned so far?

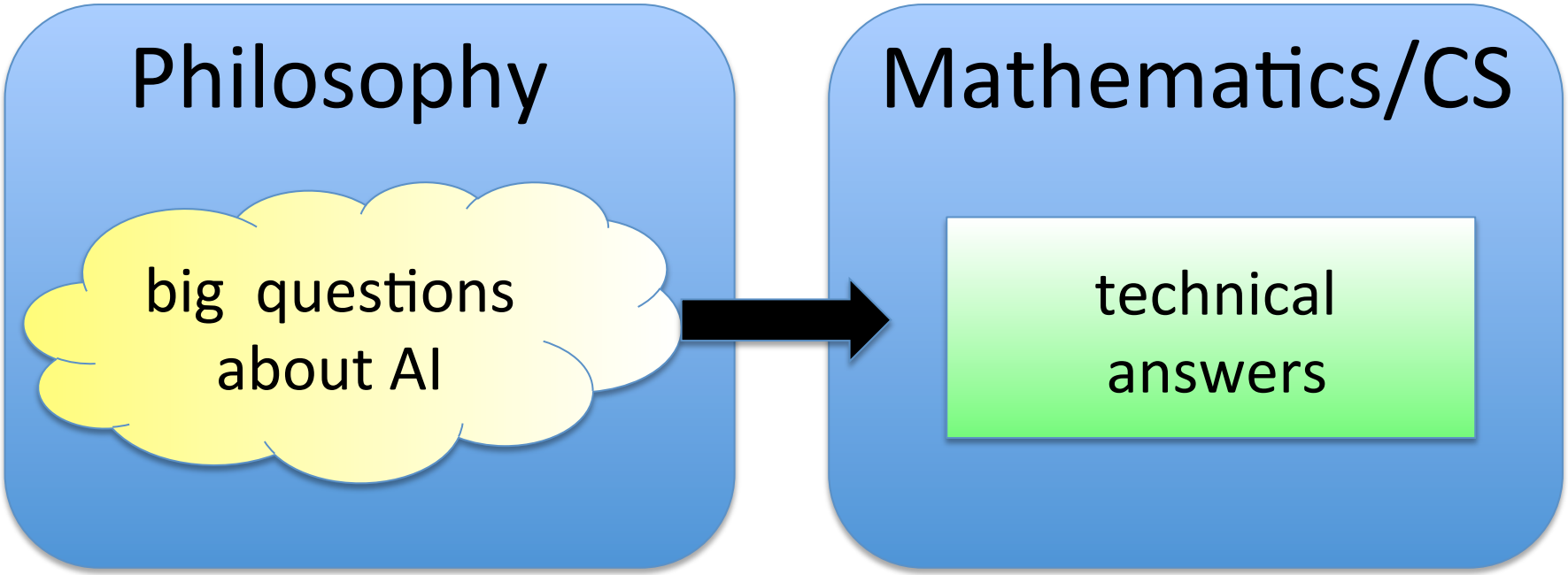
The following are less "required" than one might think for a rational gambler to avoid exploitation:

- **Calibration.** So far it looks like one need only be calibrated about logical bets that are settled sufficiently quickly (this is being actively researched).

- **Hard-coded belief coherence.** A powerful bet-balancing procedure can and must learn to "mimic" deductive rules used to settles its bets.

Meta updates

MIRI's general approach includes develop "big" questions about how AI can and should work, past the stages of philosophical conversation and into the domain of math and CS.



Meta updates

I was not personally expecting logical induction to be "solved" in this way for at least a decade, so I've updated that:

- the methodology of breaking unsettled philosophical questions down into math/CS and grinding through them is more fruitful than I thought; and
- perhaps other seemingly "out of reach" problems in AI alignment, like decision theory and logical counterfactuals, might be amenable to this approach.

Thanks!

To

- **Scott Garrabrant**, for the core idea and many rapid subsequent insights
- Tsvi Benson Tilsen, Nate Soares, and Jessica Taylor for coauthoring the paper
- Jimmy Rintjema for a *lot* of help with LaTeX bugs and collaborative editing issues

<end of this talk>

Slides from other talks I could end up wanting to use in response to questions:

Some questions

Is it feasible to build a useful superintelligence that, e.g.,

- **Shares our values**, and will not take them to extremes? ("value learning")
- **Will not compete** with us for resources? ("convergent incentives")
- **Will not resist** us modifying its goals or shutting it down? ("corribility")
- **Can understand itself** without deriving contradictions via bounded Löb's Theorem? ("self-reflective stability")

Examples of technical understanding

- Vickrey second-price auctions (1961) :
  - Well-understood optimality results (truthful bidding is optimal)
  - Real-world applications, (network routing)
  - Decades of peer-review

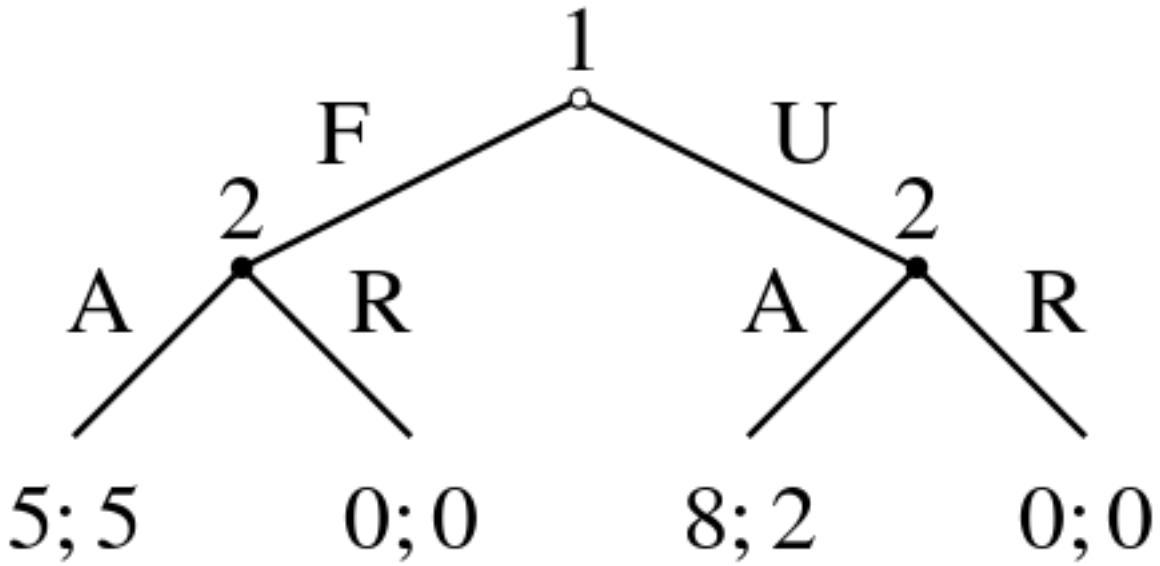
- Nash equilibria (1951) :

Formal definition [edit]

Let (S, f) be a game with n players, where S_i is the strategy set for player i, $S = S_1 \times S_2 \times \cdots \times S_n$ is the set of strategy profiles and $f = (f_1(x), \ldots, f_n(x))$ is the payoff function for $x \in S$. Let x_i be a strategy profile of player i and x_{-i} be a strategy profile of all players except for player i. When each player $i \in \{1, \ldots, n\}$ chooses strategy x_i resulting in strategy profile $x = (x_1, \ldots, x_n)$ then player i obtains payoff $f_i(x)$. Note that the payoff depends on the strategy profile chosen, i.e., on the strategy chosen by player i as well as the strategies chosen by all the other players. A strategy profile $x^* \in S$ is a Nash equilibrium (NE) if no unilateral deviation in strategy by any single player is profitable for that player, that is

$$\forall i, x_i \in S_i : f_i(x_i^*, x_{-i}^*) \geq f_i(x_i, x_{-i}^*).$$

- Classical Game Theory (1953) :



An extensive form game.

Problem: Counterfactuals for Self-Reflective Agents

What does it mean for a program A to improve some feature of a larger program E in which A is running, and which A can understand?

```
def Environment ():
    …
    def Agent(senseData) :
        def Utility(globalVariables) :
            …
        …
    …
    do Agent(senseData1)
    …
    do Agent(senseData2)
    …
    end
```

(optional pause for discussion of IndignationBot)

Example: π maximizing

What would happen if I changed the first digit of π to 9?

This seems absurd because π is logically determined.

However, the result of running a computer program (e.g. the evolution of the Schrodinger equation) is logically determined by its source code and inputs…

… when an agent reasons to do X "because X is better than Y", considering what would happen if it did Y instead means considering a mathematical impossibility.

(If the agent has access to its own source code, it can derive a contradiction from the hypothesis "I do Y", from which *anything* follows. This is clearly not how we want our AI to reason. How do we?

Current formalisms are “Cartesian” in that they separate an agent’s source code and cognitive machinery form its environment.

This is a type error, and in combination with other subtleties, it has some **serious consequences**.

Examples (page 1)

- *Robust Cooperation in the Prisoners' Dilemma* (LaVictoire et al, 2014) demonstrates non-classical cooperative behavior in agents with open source codes;
- *Memory Issues of Intelligent Agents* (Orseau and Ring, AGI 2012) notes that Cartesian agents are oblivious to damage to their cognitive machinery;

Examples (page 2)

- *Space-Time Embedded Intelligence* (Orseau and Ring, AGI 2012) provides a more naturalized framework for agents inside environments;
- *Problems of self-reference in self-improving space-time embedded intelligence* (Fallenstein and Soares, AGI 2014) identifies problems persisting in the Orseau-Ring framework, including procrastination and issues with self-trust arising from Löb's theorem;

Examples (page 3)

- *Vingean Reflection: Reliable Reasoning for Self-Improving Agents* (Fallenstein and Soares, 2015) provides some approaches to resolving some of these issues;
- *… lots more; see intelligence.org/research* for additional reading.