Decision Theory

Daniel C Satya Benson
Williams College

September 8, 2026

1 Prerequisites

- Basic probability and expected value; comfort reading pseudocode.
- Helpful but not required: Löb's theorem (covered on the Agent Foundations day; it underlies FairBot and the tournament's cooperation results) and a first acquaintance with Bayesian networks and the causal do-operator (re-introduced briefly in the lecture).
- No prior decision-theory background assumed; Newcomb's problem and the rest are introduced from scratch.

What you'll learn

(What.) By the end of the day, students can explain why decision theory is hard for an *embedded* agent (its action is just another fact about the world, so counterfactuals must be constructed, not read off) and can state the four main theories and what each gets right and wrong: CDT (intervene), EDT (condition), FDT (choose your decision function's output), and UDT 1.0/1.1 (act on the policy you would have committed to in advance). They can work the canonical problems (Newcomb, smoking lesion, counterfactual mugging, twin prisoner's dilemma) and say which theory each one separates. They understand the multi-agent layer: open-source game theory and Löbian cooperation (FairBot), commitment races, and safe Pareto improvements. They apply all of this by designing and submitting a bot to the open-source Prisoner's Dilemma tournament.
(Why.) To make a superintelligence safe we will likely prove safety properties under assumptions about *how the agent decides*. Decision theory is a *reflectively-consistent degree of freedom*: unlike a mistaken factual belief, a bad decision theory is not automatically corrected as an agent gets smarter, so a load-bearing assumption ("the agent is CDT") can silently fail if the agent self-modifies. The multi-agent failures (commitment races, conflict, exploitable cooperation) are direct AI-safety concerns, and safe Pareto improvements are one of the few constructive tools against them.
(How.) A morning lecture builds the theories in order (CDT/EDT to FDT

to UDT), motivated by the problems that break each one. A two-hour reading and discussion block has students wrestle with the primary sources and a transparency-and-cooperation prompt. An afternoon lecture moves to the multi-agent setting (open-source game theory, commitment races, safe Pareto improvements). The day ends with a programming tournament that operationalizes program equilibrium and Löbian cooperation.

2 Content

Slides: Deck I (Decision Theory: CDT to UDT) and Deck II (Open-Source Game Theory, Commitment Races, and Safe Pareto Improvements), linked from the Decision Theory session page. Tournament handout: Open-Source Prisoner's Dilemma Tournament.

2.1 Fast-track

To get the core in about an hour, or to catch up after missing the day:

- Read "The four decision theories" and "The multi-agent layer" below.
- Read the FDT paper (at least chapters 1-3) for the FDT/UDT picture, and skim Towards a new decision theory.
- Read the tournament handout and write a one-paragraph bot (even "cooperate only if the opponent provably cooperates with me" is enough to engage with the ideas).

2.2 Main content

The day has four sub-modules: the **morning lecture** (single-agent decision theory), the **reading and discussion** block, the **afternoon lecture** (multi-agent decision theory), and the **tournament**, closing with a **daily checkpoint**.

2.2.1 Morning lecture

Why decision theory, and why it is hard. For a *dualistic* agent cleanly separated from the world, the action is a free variable and the rule is just `a* = argmax_a E[U | do(A=a)]`. For an *embedded* agent the action is itself a fact about the world (predictors may have modelled it, copies may share it), so "what happens if I act differently" is a counterfactual that must be *constructed*. The decision theories differ in how they construct it.

EDT and CDT: condition vs intervene.

- *Evidential decision theory* conditions on the action as evidence: `a* = argmax_a E[U | A=a]`. This treats the action as news about everything correlated with it.

- *Causal decision theory* intervenes: `a* = argmax_a E[U | do(A=a)]`, severing the arrows *into* the action in a causal (Bayesian-network) model, so the action is news about its effects only.

- They split on the canonical cases. On **Newcomb's problem** (a predictor fills a box based on your disposition), EDT one-boxes and wins \$1,000,000; CDT two-boxes (the prediction is already made) and wins \$1,000. On the **smoking lesion** (a common cause produces both a disposition to smoke and the disease), EDT wrongly abstains (smoking is bad *evidence* though not a *cause*); CDT correctly smokes. Neither rule handles both, which points to a third kind of dependence.

- *(Aside, for the mathematically inclined.)* The two rules drop out of two ways to model the agent's interaction history: CDT is a chronological-semimeasure predictor (actions are conditioned-on inputs, never predicted, exactly the do-operator), while EDT is a joint predictor over actions and observations (conditioning on an action updates beliefs about which world and which policy you are). The slides develop this; it can be skipped without loss.

FDT: choose your function's output. Functional decision theory reframes *what you are choosing*. You run a fixed decision function; your action is its output, and everything that depends on that function (predictors who modelled it, copies running it, simulations of it) moves together with the output. FDT asks "which output of this decision function, given everything that depends on it, yields the best outcome?" It one-boxes on Newcomb and smokes on the smoking lesion. Its signature case is the **twin prisoner's dilemma**: two copies of the same agent share the same logical output (*subjunctive dependence*), so FDT cooperates (each gets \$3) where CDT defects (each gets \$1). Causal dependence is a special case of subjunctive dependence; mere correlation (the lesion) is not.

UDT: act on the policy you would have committed to.

- *UDT 1.0* chooses each action as the one a prior-stage self would have committed to: `choice(o) = argmax_a E[U | choice(o)=a]`. On **counterfactual mugging** (a coin you have already seen land the wrong way, where paying in this branch is what makes paying profitable across branches), the updateful agent refuses and the updateless agent pays, because ex ante paying is worth `(R-c)/2`. Updatelessness is not ignorance; it is refusing to update away a commitment that is good in expectation, which makes it reflectively stable.

- *UDT 1.1* optimizes the whole *policy* (a map from observations to actions) rather than each action separately, then applies it: `S* in argmax_{S:O->A} E[U | choice=S]`. This is needed when copies must coordinate their actions across branches, which per-observation optimization can get wrong.

Where this is heading. The shift across CDT/EDT to FDT to UDT 1.0 to UDT 1.1 is a shift in *what you are choosing*: an action, then a function's output, then an output you do not update away, then a whole policy. Two hard problems remain:

UDT pays real utility in the actual branch and is only as good as its prior, and *logical updatelessness* (what is the right "prior" when some uncertainty is mathematical?) has no clean answer. The **5-and-10 problem** shows the deeper trouble: a naive proof-searching agent can be driven by a spurious Löbian proof to take the worse action, because counterfactuals over one's own action break down. The frontier goal is to formalize decision problems as programs and ask which theory is *optimal* over a well-defined class of "fair" problems (where the world depends only on the agent's input/output behaviour); the two demands such a criterion forces (coordinate across calls; detect an isomorphic copy of yourself) are exactly UDT 1.1 and FDT turned into a specification.

2.2.2 Reading and discussion

Readings (arranged roughly chronologically; the FDT paper is the primary reference):

- Functional decision theory: a new theory of instrumental rationality (chapters 1-5).
- Towards a new decision theory.
- Updateless decision theory.
- Conceptual problems with UDT and policy selection.
- Pitfalls of building UDT agents.

Discussion prompt (transparency and cooperation): consider two agents A1 and A2 whose code, preferences, and decision algorithms become perfectly transparent to each other at time T (state your interpretation of "transparent" where it matters). Under these conditions, when might A1 and A2 end up in a *Pareto-inefficient* outcome? Where inefficiency looks plausible but a non-obvious argument rules it out, give that argument. (This prompt is the bridge to the afternoon: it is exactly the open-source-game-theory setting.) Supporting afternoon readings: When would AGIs engage in conflict? and Individually incentivized safe Pareto improvements in open-source bargaining.

3 Learn more

Single-agent decision theory. The reading list above, plus the FDT paper for the full treatment; on the CDT-vs-EDT-as-prediction view, the chronological-semimeasure framing follows the algorithmic-thermodynamics line (Ebtekar and Hutter). The 5-and-10 problem and logical counterfactuals are developed in the MIRI decision-theory literature (Demski and Garrabrant).