Decision Theory

Daniel C

Roadmap

From classical choice to embedded counterfactuals

- **Why decision theory matters for AI safety** (modelling, specification, reflective stability, multi-agent conflict)
- **Why it is hard:** optimisation is solved for *dualistic* agents but not for *embedded* ones, where the agent must *construct* counterfactuals
- **The candidates:** causal (CDT), evidential (EDT), functional/logical (FDT), updateless (UDT 1.0, UDT 1.1) – each is a different recipe for the counterfactual
- **A unifying lens:** CDT vs EDT as two readings of an action in sequential (AIXI-style) prediction – an *intervention* (chronological semimeasure) vs *evidence* (joint prediction)
- **The hard core:** problems with UDT, the 5-and-10 problem, logical counterfactuals and Löb's theorem
- **A unifying goal:** formalise decision problems as programs and ask which decision theory is *optimal* across all "fair" problems

Why study decision theory? (for AI safety)

Four reasons

- **Architecture-independent modelling.** We want to reason about how a superintelligent agent will *behave* without knowing the details of how it is built. Decision theory is a minimal model of "what is rational choice", capturing behaviour any sufficiently capable agent tends to converge on.
- **Implementation / the specification–optimisation split.** Alignment decomposes into (1) *specify* a goal we want, and (2) *optimise* it correctly. Decision theory is part (2). A perfect specification is **not sufficient** for safety: if the agent constructs the wrong counterfactuals, it can "optimise" the goal in a way we never intended.
- **Reflective stability.** Safety properties must remain invariant under self-modification (cf. Agent Foundations deck). Decision theory is the simplest setting to ask: *would an agent running decision theory A want to self-modify into, or build a successor with, decision theory B?* A decision theory that wants to overwrite itself is a warning sign.
- **Multi-agent dynamics.** Decision problems get hard under strategic interaction and mutual modelling. We want a decision theory under which agents can *cooperate* with other superintelligences and *avoid catastrophic conflict*, without being *exploitable*.

The easy case: dualistic agents

When the agent is cleanly separated from the world

- A *dualistic* agent has clean input/output channels with its environment (the "cybernetic" picture: a player optimising a video game). It is **"larger than"** the world (can hold a full model of it) and **"outside"** it (need not reason about itself).
- Here the action is a genuine *free variable*: nothing in the world depends on the agent's internal computation except through the downstream consequences of the action. Choosing optimally is conceptually solved:

$$a^* = \arg\max_a \ \mathbb{E}\big[\, U \mid \mathrm{do}(A = a) \,\big].$$

- This is the mature, successful framework of **expected-utility maximisation**: the von Neumann–Morgenstern / Savage axioms (preferences satisfying coherence are EU-maximisation), Bayesian updating, and **complete-class theorems** (every admissible decision rule is Bayes-optimal for some prior).
- It underpins modern economics, statistics, and reinforcement learning. The conceptual work is done; only computation remains.

The hard case: embedded agents

When the agent is a part of the world it reasons about

- An *embedded* agent is a piece of the world it is trying to optimise (cf. Agent Foundations deck). It can be copied, predicted, simulated; its source code can be read by others.
- Now **"which action I take" is just another fact about the world**, fixed by physics (or by the agent's own deterministic computation). There is no free variable to set, and **no ready-made answer to "what would happen if I acted differently"** – the alternative may be physically or logically impossible.
- So the counterfactual must be *constructed*, not read off. **This is the whole difficulty**: decision theory for embedded agents is the problem of constructing counterfactuals.
- Each decision theory is a different construction:
  - **EDT:** condition on the action as *evidence*, $\mathbb{E}[U \mid A = a]$
  - **CDT:** *intervene* causally on the action, $\mathbb{E}[U \mid \mathrm{do}(A = a)]$
  - **FDT/UDT:** reason about the *logical output* of the decision procedure being a

Evidential decision theory: condition, don't intervene

EDT as conditioning

- EDT evaluates an action by the *news* it would bring – it **conditions** on "I do a" as evidence:

$$a^* = \arg\max_a \mathbb{E}\big[\, U \mid A = a \,\big].$$

- Conditioning treats the action as evidence about *everything* correlated with it – including facts the action does not cause.
- (We will see this *help* on Newcomb but *mislead* on the smoking lesion – once both rules are on the table.)

A 60-second primer: Bayesian networks

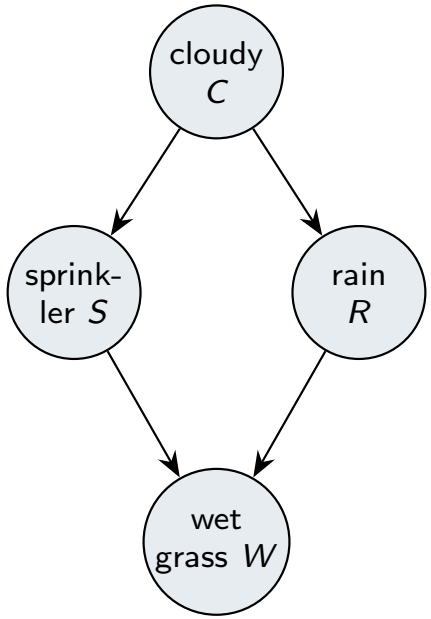
Encoding "what depends on what"

- A Bayesian network is a directed acyclic graph: nodes are random variables, an edge $X \to Y$ means Y *directly listens to* X. The joint distribution *factorises* along the graph:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\big(X_i \mid \mathrm{Pa}(X_i)\big),$$

  with $\mathrm{Pa}(X_i)$ the parents (direct causes) of X_i.
- Each variable is independent of its non-descendants given its parents: a compact, structured stand-in for a huge joint distribution.
- For the graph on the right:

$$P(C)\, P(S \mid C)\, P(R \mid C)\, P(W \mid S, R).$$



cloud cover is a common cause of sprinkler use and rain

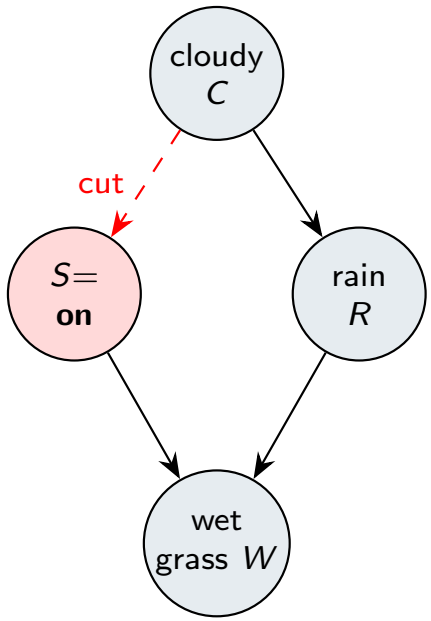
Causal decision theory: seeing vs. doing

The do-operator

- Pearl's key distinction: *observing* $X{=}x$ (passive evidence, as in EDT) is not the same as *doing* $X{=}x$ (an intervention).
- $\mathrm{do}(X = x)$: **sever all arrows into** X (cut X off from its causes), fix $X = x$, propagate downstream. Adjusting over the direct causes $z = \mathrm{Pa}(X)$:

$$P\big(Y \mid \mathrm{do}(X{=}x)\big) = \sum_z P(Y \mid x, z)\, P(z).$$

- Sprinkler graph: *seeing* the sprinkler on raises the chance it was cloudy, hence lowers the chance of rain. *Doing* $\mathrm{do}(S{=}\mathrm{on})$ cuts $C \to S$, so it tells you **nothing** about rain.
- **CDT:** $a^* = \arg\max_a \mathbb{E}[U \mid \mathrm{do}(A = a)]$ – judge each action by the causal consequences of *intervening* on it, holding the rest of the past fixed.
- The contrast in one line: $\mathbb{E}[U \mid \mathrm{do}(A{=}a)]$ (CDT, severs the action's causes) vs. $\mathbb{E}[U \mid A{=}a]$ (EDT, keeps them as evidence).



$\mathrm{do}(S{=}on)$ *severs* S *from its cause* C

Prerequisites: the sequential prediction setting

Histories, environments as chronological semimeasures, and beliefs

- Interaction: at each step t the agent emits an action $a_t \in \mathcal{A}$ and receives a percept $e_t \in \mathcal{E}$ (e.g. $e_t = (o_t, r_t)$). The history is $h_{<t} = a_1 e_1 \cdots a_{t-1} e_{t-1}$.
- An **environment** is a **chronological semimeasure** ρ: it gives the probability of percepts *given* actions,

$$\rho(e_{1:t} \parallel a_{1:t}) \in [0, 1],$$

  where the double bar $\parallel$ marks the actions as **conditioned-on inputs, never predicted**.
  - *chronological:* $e_{1:t}$ may depend on $a_{1:t}$ but not on *future* actions;
  - *semimeasure:* $\sum_{e_t} \rho(e_{1:t} \| a_{1:t}) \leq \rho(e_{<t} \| a_{<t})$ (probability may leak; a measure has equality).
- The agent's belief is a **Bayesian mixture** over a class $\mathcal{M}$ of environments, $\xi(e_{<t} \| a_{<t}) = \sum_{\nu \in \mathcal{M}} w_\nu \, \nu(e_{<t} \| a_{<t})$ (AIXI: $\mathcal{M}$ = lower-semicomputable ν, $w_\nu = 2^{-K(\nu)}$).
- Decision-theoretic crux: **when the agent chooses a_t, does that choice update its belief about which environment it is in?** The answer separates CDT from EDT.

Chronological semimeasure = CDT

The action is an intervention, not evidence

- In a chronological semimeasure the action appears **only to the right of** $\|$: ρ assigns *no* probability to actions, so "the action as evidence about ρ" is not even definable.
- The posterior over environments updates on **percepts only**, holding the actions fixed as given:

$$w_\nu(h_{<t}) \;=\; w_\nu \,\frac{\nu(e_{<t} \,\|\, a_{<t})}{\xi(e_{<t} \,\|\, a_{<t})}\,.$$

  Choosing a_t does **not** reweight w_ν: *you do not update "which environment am I in" by deciding.*
- So the agent evaluates an action by *intervening* and propagating through a *fixed* which-world belief:

$$a_t \;=\; \arg\max_a \sum_{e_t} \xi(e_t \,\|\, h_{<t}\, a)\,\big[r_t + \cdots\big] \;=\; \arg\max_a \mathbb{E}_\xi\big[\,U \,\|\, \mathrm{do}(a)\,\big]\,.$$

- This is exactly **CDT**: the chronological bar $\|$ *is* the do-operator, lifted to whole histories. (Newcomb: such an agent two-boxes – the prediction is a fixed past percept, untouched by $\mathrm{do}(a)$.)

Joint prediction = EDT

Actions predicted exactly like observations

- Now model the *entire* history $h = a_1 e_1 a_2 e_2 \cdots$ as **one stream to predict** over $(\mathcal{A} \times \mathcal{E})^*$ – actions get probabilities too. Take the generative model to be a mixture over (world ν, policy π) pairs:

$$P(h_{1:t}) \;=\; \sum_{\nu,\pi} w_{\nu,\pi} \prod_{k=1}^{t} \pi(a_k \mid h_{<k})\, \nu(e_k \mid h_{<k} a_k).$$

  (Solomonoff's $M(h) = \sum_{p:U(p)=h*} 2^{-\ell(p)}$ is the universal such joint predictor.)

- Choosing/observing a_t is now ordinary **evidence**: Bayes updates the joint posterior over *both* the policy and the world,

$$w_{\nu,\pi}(h_{<t}\, a_t) \;\propto\; w_{\nu,\pi} \left[\prod_{k<t} \cdots\right] \pi(a_t \mid h_{<t}).$$

  The factor $\pi(a_t \mid h_{<t})$ tells you "**what policy am I running**"; and any world ν *correlated with the action in the prior* $w_{\nu,\pi}$ is reweighted – "**what world am I in**".

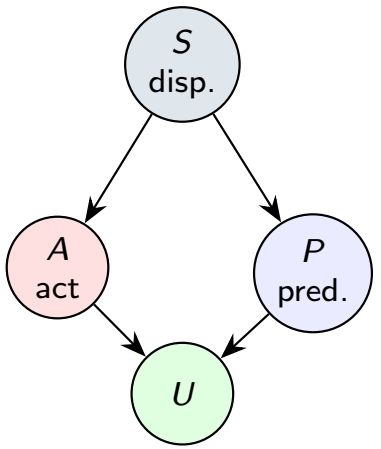
- So the agent evaluates an action by *conditioning*: $a_t = \arg\max_a \mathbb{E}_P[\, U \mid A_t{=}a,\; h_{<t}\,]$.

- This is exactly **EDT** – and it reproduces the smoking lesion: the prior correlates the lesion (a world-feature) with the smoking disposition (a policy-feature), so conditioning on "I smoke" upweights lesion-worlds and the act looks like bad news, though it is no lever.

Newcomb's problem: CDT two-boxes, EDT one-boxes

The same problem splits the two rules

- **Newcomb:** a transparent box holds **$**1,000; an opaque box holds **$**1,000,000 *iff* a highly reliable predictor Ω foresaw you would take *only* the opaque box. Ω has already chosen.
- **CDT** applies $\mathrm{do}(A)$: this **severs** your action from your disposition S, so it no longer informs Ω's prediction P. The contents are fixed; taking both boxes earns **$**1,000 more *in every fixed state*. CDT **two-boxes** and predictably walks away with **$**1,000.
- **EDT** conditions: one-boxing is strong *evidence* the opaque box is full, so $\mathbb{E}[U \mid \text{one-box}]$ is huge. EDT **one-boxes** and gets **$**1,000,000 – it keeps exactly the predictive dependence CDT discards.
- **Verdict:** CDT loses Newcomb. (In the lens above: the chronological-semimeasure agent two-boxes; the joint-prediction agent one-boxes.)



CDT cuts $S \to A$, losing the link to P.

EDT mistakes evidence for control

The smoking lesion

- A genetic lesion L causes cancer (cost C) *and* makes you more likely to want to smoke. Smoking itself is harmless and gives benefit $b > 0$.
- Conditioning on "I smoke" raises $P(L)$, hence raises estimated cancer risk: $\mathbb{E}[U \mid \text{smoke}] = b - C\,P(L \mid \text{smoke})$. For strong correlation, EDT **abstains** – even though abstaining does *nothing* to reduce cancer. EDT confuses a **symptom** for a **lever**.
- **Scorecard:** CDT and EDT are each half right.
  - CDT ignores predictive dependence $\Rightarrow$ loses Newcomb.
  - EDT overreacts to mere correlation $\Rightarrow$ loses the smoking lesion.
- The missing ingredient is a dependence that is neither raw physical causation nor raw evidential correlation.

CDT vs. physics: the intervention is a fiction

Does CDT really "reflect physical reality"? (following Ebtekar & Hutter)

- CDT's $\mathrm{do}(A = a)$ holds the past fixed, **magically sets the action**, and propagates forward. What licenses cutting the action loose from its causes?
- In a *deterministic, time-reversible* physical world, your action has physical antecedents (your brain state, your source code). A predictor that reads those antecedents is correlated with your action through **perfectly ordinary physics**, not magic.
- "Severing the incoming arrows to A" directly **contradicts embeddedness**: it pretends your action has no causes, but physically it does. The intervention is a harmless idealisation for a dualistic agent; for an embedded agent it *throws away real correlations* (with predictors, copies, simulations).
- So the usual defence is backwards: physics gives no privileged metaphysical "intervention" that frees the action from its past. This pressures us to move the counterfactual off the *physical act* and onto the agent's *policy / computation*.

Spurious counterfactuals: EDT breaks for self-aware agents

Conditioning on a probability-zero action

- A capable embedded agent can reason about (even *prove* things about) its own source code, so it can largely predict its own action. But then, for the action a it will *not* take, $P(A = a) = 0$.
- EDT's recipe $\mathbb{E}[U \mid A = a]$ is then $\frac{0}{0}$ – **undefined**. You may assign the unchosen action *any* value you like: a **spurious counterfactual**.
- So conditioning gives *no constraint* on counterfactual actions you are certain you won't take – exactly the cases decision-making turns on.
- (Foreshadow: in proof-theoretic form this same pathology becomes the **5-and-10 problem**, and the fix needs a notion of *logical* dependence.)

Functional decision theory: choose your function's output

"Which output of this very function would yield the best outcome?"

- Reframe *what you are choosing*. You are not just *this body about to act*; you are running a fixed **decision function**, and your action is its *output* on the situation you face.
- Many things in the world may depend on that *same* function: a predictor who **modelled** it, a **copy** of you running it, a simulation of it. When the function's output changes, all of them change *with* it.
- So FDT does not ask "what should I do, holding everything else fixed?" (that is CDT). It asks the question the paper puts at its centre:

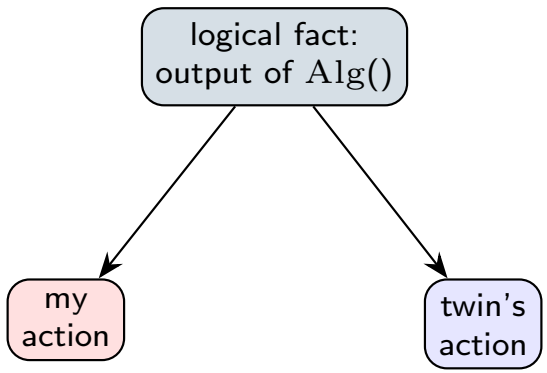
  "which output of this very decision function – given everything in the world that depends on it – would yield the best outcome?"

  and then *returns that output*. Concretely: score each candidate action a by the outcome in the world where *my decision function outputs* a (so the predictor predicted a, and every copy outputs a), and take the best.
- It acts *as if it controls all instances of the computation at once* – not by magic, but because they really are the same logical fact.
- This wins the earlier cases for the *right* reason: in Newcomb the prediction tracks the function's output, so FDT **one-boxes**; the lesion does *not* run the function, so FDT still **smokes**. FDT shines exactly where embedded agents live – worlds containing copies of you, or predictors that model your decision procedure.

Twin Prisoner's Dilemma & subjunctive dependence

Same function, two bodies

- You and a psychological *twin* (your exact copy) play a one-shot Prisoner's Dilemma in separate rooms, no communication. You run the *same* decision function.
- **CDT:** "I can't causally affect her; defection dominates." Both defect, each gets **$**1.
- **FDT:** "We compute the same function. If it outputs C, both cooperate; if D, both defect." So FDT cooperates – both get **$**3.
- **Subjunctive dependence** (Y&S): when two physical systems compute the *same* function, their outputs are logically tied – like two calculators evaluating 6288+1048: see one print 7336 and you know the other will too. Causal dependence is a *special case*; mere correlation (both calculators happen to be pink) is not. FDT acts as if it controls *all* instances of the computation at once.



one logical cause, two physical effects: CDT sees no arrow between them, FDT sees the shared parent

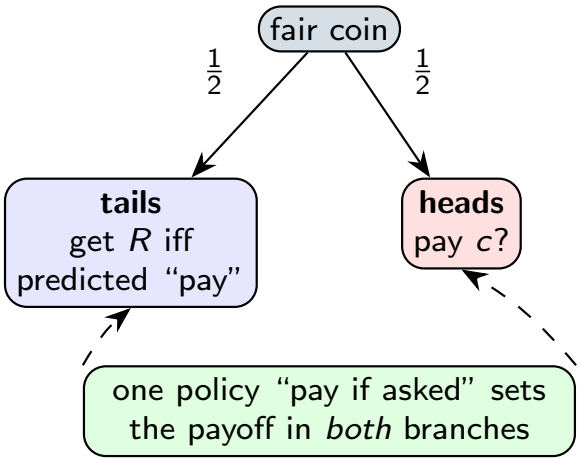
Counterfactual mugging: the updateless move

Paying in a branch you know you're not rewarded in

- Ω flips a fair coin. **Tails:** you get $R = \$10{,}000$ iff Ω predicted you'd pay on heads. **Heads:** Ω asks for $c = \$100$. You are asked to pay (so: heads). Pay?
- **Updateful** reasoning conditions on "I'm in heads": it sees only the **$**100 cost and **refuses**. But *ex ante* the policy "pay if asked" is worth $\frac{1}{2}(R - c) > 0$.
- **UDT 1.0:** evaluate the local output against the **prior** – keep the (tails) branch whose payoff your policy controls, by dropping the updateful rule's extra $O{=}o$ conditioning:

$$\mathrm{choice}(o) = \arg\max_a \mathbb{E}\big[U \mid \mathrm{choice}(o){=}a\big].$$

  So UDT 1.0 **pays**.
- Updatelessness is *not* ignorance (no predictor ⇒ UDT = ordinary EU max), and it is *reflectively stable*: the agent already follows the policy it would have precommitted to.



UDT 1.1: optimise the whole policy, not one action

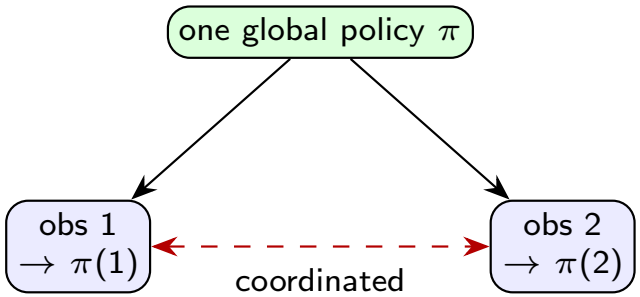
When local outputs must be coordinated

- Even *updateless local-action* selection can be too local. **Copied-agent anti-coordination**: two copies see index 1 or 2, must pick *different* actions to win (10 each), else 0.
- The local question "if $\mathrm{Alg}(1) = A$, what is $\mathrm{Alg}(2)$?" is the *wrong type*: both symmetric answers fail. The good object is the *whole table* $\mathrm{Alg} : \{1, 2\} \to \{A, B\}$.
- **UDT 1.1 (Wei Dai):** first choose the entire *policy* (a map $S : O \to A$) against the prior, *then* apply S^* to your actual observation o:

$$S^* \in \arg\max_{S:\, O \to A} \mathbb{E}\big[U \mid \mathrm{choice}{=}S\big].$$

(UDT 1.0 instead fixes a single *output* and hopes the rest fall out right.)

- π_{AB}, π_{BA} score 10; a shared tie-breaker coordinates the copies. This is **global optimisation over branches**, not local optimisation.



The three shifts in "what am I choosing?"

CDT/EDT → FDT → UDT 1.0 → UDT 1.1 (after Kosoy's formalism)

Rule	Object chosen	Update status	Dependence used
CDT	local physical act	updateful	causal intervention
EDT	local act (as evidence)	updateful	evidential correlation
FDT / TDT	local algorithm output	updateful	logical / subjunctive
UDT 1.0	local algorithm output	**updateless**	logical / subjunctive
UDT 1.1	**whole policy**	updateless	logical over policies

- **Shift 1 (FDT):** from a physical act to the output of an abstract computation – captures predictors, copies, simulations.
- **Shift 2 (UDT 1.0):** from posterior to prior – keep the value of being predictable, don't discard branches whose payoff your policy controls.
- **Shift 3 (UDT 1.1):** from a single output to a whole policy – coordinate behaviour across all possible observations.

Problem 1: paying for worlds that don't exist

Ex-ante optimality is only as good as the prior

- UDT maximises *prior* expected utility. In the branch it actually inhabits it will **knowingly burn real utility**. In counterfactual mugging, after seeing heads, UDT hands over a real $100 in the only world it will ever experience, to benefit a tails-world *it now knows did not happen*.
- Generalise: a UDT agent will take actions that are bad *in the actual world* because they would have been good in counterfactual scenarios that are never realised.
- This is dangerous when the prior is even slightly wrong about *which worlds are possible*: put prior mass on a predictor that does not actually exist (or on a logically impossible scenario), and UDT pays a **real, ongoing tax** for fake benefits. A clever (or merely hypothetical) Ω can extract resources by promising/threatening in branches that never occur.
- The worry: is "best from the prior" the right notion of rational, when "the prior" ranges over worlds you have since learned are not actual?

Problem 2: logical updatelessness

Which prior, when some of your uncertainty is mathematical?

- A real embedded agent's uncertainty is not only *empirical* (coin flips). It is also **logical**: it does not know all consequences of its own source code, the 10^{100}-th digit of π, or whether the opponent's program halts.
- UDT says "don't update." Refusing to update on *empirical* observations is clear. Refusing to update on *logical* facts is not: as the agent thinks longer it *learns* logical facts (proves theorems, computes digits) – and that looks exactly like updating.
- **Logical updatelessness** would have the agent choose its policy as if from a state of *logical* ignorance, not taking computed logical facts into account. But (a) it cannot help computing some of them, and (b) we have **no coherent "logical prior"** – a probability distribution over mathematical truths before any computation.
- This is open. A satisfactory UDT needs a theory of *logical uncertainty* (cf. logical induction), which we do not yet have. It is also the doorway to the 5-and-10 problem.

The 5-and-10 problem

Two programs (Demski & Garrabrant)

Take \$5 or \$10; utility is the dollars taken, so take \$10. The agent has its own source code and the world's, and chooses by proof search.

```
U():
  if A() = 10:  return 10
  if A() = 5:  return 5

A():
  search for a proof of
    [A()=5 -> U()=x] and
    [A()=10 -> U()=y],
    for x,y in {0,5,10}
  if found, x>y:  return 5
  else:  return 10
```

Why it returns \$5

- The world returns whatever the agent returns, so \$10 genuinely beats \$5. Yet with enough search time `A` returns \$5.
- Let P be the sentence "[`A()=5` $\Rightarrow$ `U()=5`] and [`A()=10` $\Rightarrow$ `U()=0`]".
- **A proof of P makes P true.** If P is proved, `A` finds it ($x{=}5 > y{=}0$) and returns \$5. Then `A()=5`, so the first conjunct holds; and `A()=10` is false, so the second holds *vacuously* (a false "if" implies anything). Hence P.
- So $\Box P \to P$ is itself provable ($\Box X$ means "X is provable"). **Löb's theorem** ($\Box(\Box P \to P) \to \Box P$) then makes P provable; the search finds it and returns \$5.
- **Spurious proof:** `A` takes \$5 because it proves that taking \$10 pays \$0, and that holds *only because* it takes \$5. The action not taken is provably false, and an "if" with a false premise carries no information, so ordinary counterfactuals over one's own action break down.

Why we want a formal optimality criterion

Beyond case-by-case scoring

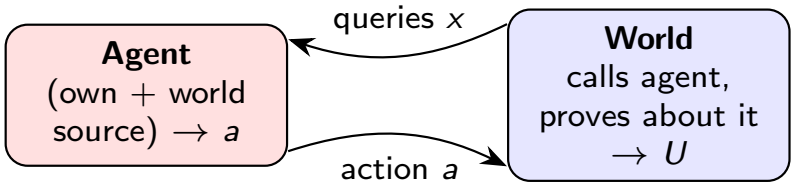
- Newcomb, counterfactual mugging, the 5-and-10 problem, the twin PD, open-source games all share a **common structure**: agent and environment are both *programs* that may inspect each other's code, simulate each other's behaviour, or contain subroutines algorithmically *isomorphic* to one another.
- Yet these are usually presented as standalone puzzles, and CDT/EDT/FDT/UDT are judged *case by case* ("does it get Newcomb right?") rather than against a single optimality standard.
- **The substantive question:** can we define one class of computational environments rich enough to contain all these problems at once, together with a criterion of optimality that any candidate decision theory can be *measured* against?

Agents and worlds as programs

The setup

- **Agent** = a program with access to (i) its own source code and (ii) the world's source code; it returns an action.
- **World** = a program that may
  - *call the agent* as a function (input $\to$ action), possibly *many times on different inputs* ("what would the agent do if it observed x?"), and
  - *search for proofs* about the agent's input–output behaviour;

  it returns a utility.
- **Goal:** design an agent whose utility is high across (almost) all worlds – the best mapping from situations to actions.



Fairness

Judging the agent only on its behaviour

- Without a constraint, a "world" could just punish source code: "if the agent is written in Python, utility $= 0$." Then *no* decision theory could do well and the criterion would be vacuous.
- **Fairness:** the world may depend on the agent *only through its input–output behaviour* (its policy), never through implementation details unrelated to its outputs. Equivalently, a fair world is representable as a program that takes the agent's *policy/behaviour* as input.
- Newcomb, the smoking lesion, counterfactual mugging, the twin PD, and the open-source PD are all **fair**: the predictor/lesion/copy interacts with the agent through its behaviour. This is what makes them legitimate tests.

What a good agent must discover

Two demands – which are exactly FDT/UDT turned into a spec

- **Coordinate across function calls.** If the world queries the agent on several inputs (probing its counterfactual behaviour), the agent must *recognise this* and keep its counterfactual outputs *coordinated* for high utility. (This is precisely UDT 1.1's move to optimise a whole policy.)
- **Subjunctive dependence.** If the world contains a subroutine *algorithmically isomorphic* to the agent (a copy in different code), the agent should treat its own decision as also fixing that subroutine's output – and so must *detect* that "the world contains something isomorphic to me." (This is precisely FDT.)

The open problem (and the bridge ahead)

- Define the class of fair computational worlds precisely; define optimality (e.g. dominance across all fair worlds); show CDT/EDT/FDT/UDT differ on it; and ask whether any (or some modification) is *optimal*.
- Caveats already met: logical counterfactuals (5-and-10), logical uncertainty / logical updatelessness, what counts as a "policy", and multi-agent fairness & bargaining – which lead into reflective oracles, open-source games, and commitment races.