Agent Foundations

Daniel C

19th April 2026

Why Agent Foundations?

Aligning a system that doesn't exist yet

- We are trying to align a superintelligence that does not yet exist, under conditions unlike anything we have observed
- Agent foundations studies the properties of intelligent agents *in general*, rather than any specific instantiation – so that we can reason about the behavior of superintelligent agents in advance
- We need to get alignment right on the *first critical try* when operating at a dangerous level of intelligence – unaligned operation at that level may be catastrophic, and we do not get to iterate. We cannot rely on trial-and-error the way science usually does
- Even if we try to learn about superintelligent agents empirically from weaker systems, we still need a theory of *what kinds of* properties generalize to greater intelligence – and that is itself a theory in agent foundations

Reflective Stability

Self-modification and the stability of goals

- A superintelligence will be capable of: rewriting its own source code, building a smarter successor AI, and radically revising its world model (e.g. learning new physics, ontology shifts)
- A property is *reflectively stable* if it remains invariant under all such self-modifications
- Any safety guardrail we instill is irrelevant if the AI can self-modify to remove it, or build a successor that lacks it
- For a sufficiently capable goal-directed agent, **nothing sticks around unless it has to** – every property, architecture, constraint, and conceptual scheme is subject to being optimized away
- To guarantee a property persists (e.g. "this machine doesn't kill all humans"), we must understand what kinds of things *can* stick around under this pressure – most things don't
- Reflectively stable properties do **not** self-correct: if Gandhi doesn't want to kill people, he will refuse a pill that makes him want to kill people
- Contrast with properties that *do* self-correct: a false belief about the world (e.g. a mistaken belief about gravity) will be corrected as the AI gets smarter

Goodhart's Law and Robust Concepts

When proxy measures break under optimization

- Let V be the true goal, U a proxy observed to correlate with V, and X the gap between them: $U = V + X$
- A point with large U will likely have large V – but also large X; so optimizing U selects not just for V but also for the discrepancy between U and V
- The harder you optimize, the more V lags behind U
- This is already a problem with weak optimization; whereas a superintelligence will impose extreme optimization pressure to the proxy
- Aligning a superintelligence requires us to have robust concepts or "True names" that don't break down even when they are subjected to extreme optimization pressure

Examples of "True names" we may want

- **Optimization**
- **Goals**
- **World models**
- **Embeddedness**

Reasoning About Ideal Intelligence

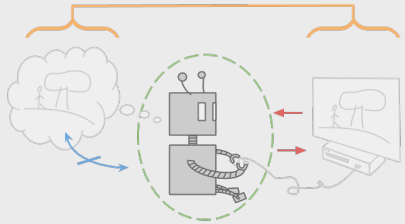
Coherence and dominated strategies

- An agent with circular preferences ($A \succ B \succ C \succ A$) can be *money-pumped*: an adversary cycles them through trades, extracting a small fee each round, leaving the agent strictly worse off after every cycle
- **Dutch book setup:** formalize an agent's degrees of belief as *betting prices* – for each event X, the agent posts a price $p \in [0, 1]$, meaning they are willing to buy or sell a contract that pays \$1 if X turns out to be true and \$0 otherwise, at price p
- The agent's price p for X thus represents their subjective probability that X is true
- A Dutch book is a combination of such contracts, bought or sold at the agent's own posted prices, that guarantees the agent a net loss regardless of how the world turns out
- An agent whose prices violate the probability axioms executes a *dominated strategy* – a Dutch book can always be constructed against them, and they can be exploited indefinitely
- **Key result:** an agent is immune to Dutch books if and only if their betting prices satisfy the axioms of probability – Bayesianism and Expected utility maximization falls out of the requirement to avoid dominated strategies
- **Sidenote:** Measuring stick of utility problem

Dualistic agents

Dualistic agents

- This is Alexei, and Alexei is playing a video game.

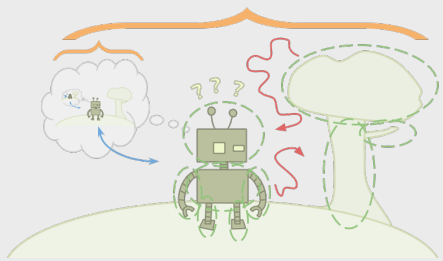


- Clear input & output channel
- Can hold the entire video game inside his mind
- Does not have to think about himself, only optimizing the environment
- Standard models of agency (e.g. EU maximization, AIXI) usually operate under this dualistic picture

Embedded agents

Embedded agents

- This is Emmy. Emmy is playing real life.



- The environment contains her, Emmy is just one part of the environment
- No well defined input & output channel
- Cannot hold the environment inside her own head
- Environment may contain copies of her
- Emmy is made out of the same pieces that the rest of the environment is made out of
- Since Emmy is within the environment that she is manipulating, she is also going to be capable of self-improvement
- We would like a theory of consequentialist agency that is consistent with this embeddedness setting

Self-Modification and Vingean Reflection

Tiling agents: designing a successor

- A sufficiently intelligent agent A_1 may choose to build a successor agent A_0 (or equivalently, rewrite its own source code – self-improvement is a special case of constructing a successor)
- **Vingean uncertainty:** A_1 cannot predict the exact decisions of something as smart or smarter than itself – if it could, it would be just as smart.
- So A_1 cannot verify A_0's safety by simulating it step by step. Instead, A_1 must reason *abstractly* about properties of A_0's design, guaranteeing that whatever A_0 does falls within acceptable bounds. Just as Deep Blue's designers could reason that the program was "trying to win" without knowing its exact moves
- In addition, A_1 cannot anticipate all future observations that A_0 will receive
- **Tiling:** we want A_1 to be able to approve the construction of successors that is similar to itself (e.g. same software, different hardware)

Self-Modification and Vingean Reflection

How can A_1 trust A_0?

- A_1 and A_0 share a satisficing goal $\mathcal{G}$. A_0 only takes actions it can *prove* achieve $\mathcal{G}$
- A_1 inspects A_0's source code and verifies: if A_0 is constructed, then any action A_0 takes was proved (in A_0's proof system) to achieve $\mathcal{G}$:

$$A^1 \Vdash \bar{A}^0 \rightarrow (\forall b_i \in Acts^0 : \bar{b}_i \rightarrow \Box_0[\bar{b}_i \rightarrow \mathcal{G}]) \qquad (2.3)$$

($\bar{A}^0$ = "A_0 is constructed", $\Box_0[\phi]$ = "there exists a proof of ϕ from A_0's axioms")

- But this only tells A_1 that A_0 *acts on what A_0's proof system certifies* – not that what A_0's proof system certifies is actually *true*
- To close this gap, A_1 needs to trust A_0's proof system: whenever A_0 proves something, it is actually true

$$A^1 \Vdash \forall b_i : (\Box_0[\bar{b}_i \rightarrow \mathcal{G}] \rightarrow (\bar{b}_i \rightarrow \mathcal{G})) \qquad (2.4)$$

- Combining (2.3) and (2.4): $A^1 \Vdash \bar{A}^0 \rightarrow (\forall b_i \in Acts^0 : (\bar{b}_i \rightarrow \mathcal{G}))$
  "If A_0 is constructed, then any action A_0 takes achieves the goal"

The Löbian Obstacle

The problem of self-trust

- When A_1 and A_0 use the *same* proof system L, equation (2.4) becomes:

$$L \vdash \forall \phi : (\Box \phi \rightarrow \phi)$$

  "Whenever L can prove ϕ, then ϕ is actually true" – L vouches for its own proofs

- This seems like a reasonable thing for a rational agent to believe: "if I could prove P, then P would be true"
- But **Löb's theorem** says: for any statement C, if $L \vdash \Box C \rightarrow C$, then $L \vdash C$
  The *only* statements for which L can assert "if I could prove this, it would be true" are statements L could already prove anyway
- Self-trust would imply that $L \vdash \Box(2+2=5) \rightarrow (2+2=5)$: "if I could prove $2+2=5$, then $2+2=5$ would be true." But by Löb's theorem, this means $L \vdash 2+2=5$
- A consistent formal system cannot vouch for its own proofs in the abstract – it can only trust a proof once it has *witnessed it directly*

The Löbian Obstacle

Consequence for tiling agents

- A_1 can only trust A_0's proofs if A_0 uses a *strictly weaker* proof system than A_1
- Each successive agent in a chain of successor must use a weaker proof system than its predecessor – a "telomere" of logical strength that shortens with each generation
- Eventually the chain runs out of trust, making indefinitely long chains of safe self-modification seemingly impossible

Embedded Subproblem: Logical Uncertainty

Empirical vs. logical uncertainty

- **Empirical uncertainty:** "What is the probability it will rain tomorrow?" – we lack *evidence* about the world; more observations would help
- **Logical uncertainty:** "What is the probability that the millionth digit of π is a 7?" – the answer is fully determined by the axioms of mathematics. We are not missing any observations; we are missing *computation*
- Other examples: uncertainty about the output of a program whose source code you already know, or whether a large number is prime

Why this matters for embedded agents

- Standard Bayesian reasoning assumes *logical omniscience*: the agent already knows all the consequences of its beliefs and can instantly perform any computation at no cost
- But computationally bounded agents cannot do this – they may know all the axioms of arithmetic yet still be uncertain about what follows from them
- We need a principled way to assign probabilities to logical facts, and to update those probabilities as we refine our beliefs through further computation

Logical Induction: The Criterion

Setting up a prediction market over logical sentences

- **Deductive process.** A computable sequence $D_1 \subseteq D_2 \subseteq D_3 \subseteq \cdots$ of finite sets of sentences (think: theorems proved so far). On day n, the process has confirmed every sentence in D_n
- **Market.** A computable sequence of pricings $\mathbb{P} = (P_1, P_2, \ldots)$ where $P_n(\varphi) \in [0, 1]$ is the "price" of sentence φ on day n – interpreted as the reasoner's probability that φ is true. A φ-share pays \$1 if φ is confirmed true (by the deductive process), \$0 otherwise
- **Expressible feature.** A real-valued function ξ of the market's price history, built from price lookups $\varphi^{*n}(\mathbb{P}) := P_n(\varphi)$, rational constants, $+$, $\times$, and max.
- **Trading strategy.** An expression of the form $T_n = c + \xi_1\phi_1 \ldots \xi_n\phi_n$ where the ξ_i are expressible features (functions of prices up to day n) and the φ_i are sentences. Intuitively, a trading strategy looks at the prices of each logical sentence and decides how much "shares" of the sentence to buy/sell.
- **Trader.** A sequence of trading strategies $T = (T_1, T_2, \ldots)$

Logical Induction: The Criterion

The criterion

- A trader T **exploits** market $\mathbb{P}$ relative to D if, for any world W consistent with D, the trader's cumulative wealth $\sum_{n=1}^{N} T_n(W)$ is unbounded

> **Definition 3.5.1** (Exploitation). *A trader $\overline{T}$ is said to **exploit** a valuation sequence $\overline{\mathbb{V}}$ relative to a deductive process $\overline{D}$ if the set of values*
>
> $$\left\{\mathbb{W}\left(\sum_{i\leq n} T_i(\overline{\mathbb{V}})\right) \,\middle|\, n \in \mathbb{N}^+, \mathbb{W} \in \mathcal{PC}(D_n)\right\}$$
>
> *is bounded below, but not bounded above.*

- **Logical induction criterion:** No *efficiently computable* (polytime) trader can exploit $\mathbb{P}$ relative to D
- Analogy: the classical Dutch book argument says "no trader at all can exploit you $\Rightarrow$ your beliefs satisfy the probability axioms." Logical induction weakens this to "no *efficient* trader can exploit you," and this single condition implies a wide range of desirable properties

Logical Induction: Construction Sketch

How to build a logical inductor

- Trading strategies are affine combinations of expressible features, so they can be *aggregated*: any finite collection of traders can be summed into a single combined trader. In fact, an infinite sequence of traders can be combined (via a convergent weighted sum) into a single trader that exploits the market if *any* of the individual traders do
- So it suffices to defend against a single combined trader $\overline{T}$ that represents all e.c. traders at once
- **Finding fair prices via fixed points.** On each day n, the algorithm looks at the combined trader's strategy $\overline{T}_n$ and the prices $\mathbb{P}_{\leq n-1}$ from previous days. It uses an approximate fixed point (guaranteed to exist by *Brouwer's fixed point theorem*) to find prices P_n such that the combined trader's profit $\overline{T}_n(\mathbb{P}_{\leq n})$ is at most a very small positive amount
- Intuitively, the fixed point finds the trader's "fair prices" – prices at which the trader on net abstains from betting, by adjusting prices upwards if the market buys more shares on net and vice versa
- Since the combined trader's profit on each day is bounded by a small ϵ_n with $\sum_n \epsilon_n < \infty$, cumulative profit is bounded, so no e.c. trader achieves unbounded profit – the logical induction criterion is satisfied

Logical Induction: Desirable Properties

Properties of logical inductor

- **Convergence and Coherence:** In the limit, the prices of a logical inductor describe a belief state that is fully logically consistent – $P_\infty(\varphi) \leq P_\infty(\psi)$ whenever $\varphi \Rightarrow \psi$, probability 1 to all theorems, 0 to all contradictions
- **Learning Statistical Patterns:** E.g. learns to assign $\approx 10\%$ to the claim "the nth digit of π is a 7" for large n, even without computing those digits
- **Expectations:** Logical inductors give rise to a well-behaved notion of the expected value of a logically uncertain variable
- **Self-Trust:** A logical inductor's current credence in any sentence is a weighted average of its expected future credences – it defers to its future self without needing to witness the reasoning trace of its future self

Embedded Subproblem: Decision Theory

The problem of counterfactuals for embedded agents

- For a dualistic agent (playing a video game), optimization is simple: just $\arg\max_a \mathbb{E}[U \mid \text{do}(A = a)]$ over actions
- For embedded agents, which action you pick is just another fact about the world. There is no well-defined notion of "what would happen if I took a different action"
- Different decision theories correspond to different ways of constructing these counterfactuals: conditioning on the action as evidence (EDT), intervening causally on it (CDT), or reasoning about the logical consequences of being the kind of agent that runs a given decision procedure (FDT).

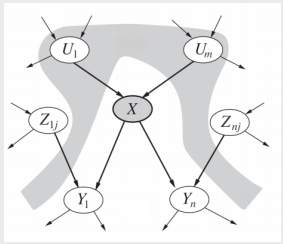
Causal decision theory

Causal decision theory and the do-calculus

- A Bayesian network encodes conditional independences via a directed acyclic graph. The joint distribution factorizes as:

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

where $\mathrm{Pa}(X_i)$ are the parents of X_i in the graph



- The do-operator models an intervention: $\mathrm{do}(X = x)$ *severs* all incoming arrows to X (removing the influence of X's parents) and fixes $X = x$, then propagates consequences downstream:

$$P(Y \mid \mathrm{do}(X = x)) = \sum_{u,z} P(Y \mid X{=}x, U{=}u, Z{=}z)\, P(U{=}u)\, P(Z{=}z)$$

- CDT chooses $a^* = \arg\max_a \mathbb{E}[U \mid \mathrm{do}(A = a)]$: intervene on the action node and maximize expected utility under the resulting distribution

Functional Decision Theory

Twin Prisoner's Dilemma

- You and your psychological twin must each choose to cooperate or defect. You reason identically but choose in separate rooms without communication
- **CDT reasoning:** "My action cannot causally affect my twin's action. No matter what she does, I gain extra by defecting. Defection dominates cooperation." Both defect.
- But your twin is computing the *same decision function*. If your decision procedure outputs "defect", so does hers. The relevant question is not "what should I do holding her action fixed?" but "what output of this decision procedure would yield the best outcome?"

FDT: reasoning about the logical output of your decision procedure

- FDT asks: "Which output of this very decision function I'm using would yield the best outcome, given everything in the world that depends on this function's output?"
- In the twin PD: if this function outputs cooperate, both copies would cooperate. If it outputs defect, both would defect. So FDT cooperates
- FDT is sensitive to whether the correlation between your action and the world arises from a *logical dependence*. It acts as if it controls both its own action and the action of its copy.
- FDT performs well in environments that contain copies of you, or predictors that model your decision procedure – precisely the settings that arise for embedded agents

Updateless Decision Theory

Reflective stability of decision theories

- We want decision theories to be *reflectively stable*: the decision theory shouldn't want to self-modify into a different decision theory
- CDT defects in the twin prisoner's dilemma, but *before entering the game*, a CDT agent would pay to precommit to cooperating – it would like to self-modify into an FDT agent. This means CDT is not reflectively stable

From updateful to updateless

- **Updateful (EDT):** choose the action that maximizes expected utility *given your current observations*:

$$\texttt{choice}(o) := \arg\max_{a \in A} \mathbb{E}_P[U \mid \texttt{choice}(o) = a,\ O = o]$$

- **Updateless (UDT):** drop the conditioning on observations – choose the action you would have wanted to commit to *before* seeing anything:

$$\texttt{choice}(o) := \arg\max_{a \in A} \mathbb{E}_P[U \mid \texttt{choice}(o) = a]$$

Updateless Decision Theory

UDT as coordination across counterfactual branches

- **Blackmail:** giving in to blackmail achieves higher utility *when you're being blackmailed*. But giving in incentivizes blackmail – so an updateful agent conditions gives in to blackmail when it is being blackmailed.
- UDT asks: "what *policy* (mapping from observations to actions) would I have wanted to commit to before seeing any observations?" Being the kind of agent that refuses blackmail is the better policy overall, even though it is locally costly when blackmailed
- UDT agent coordinates across all counterfactual branches: by refusing to give in, the blackmailed branch sacrifices utility so that the branches where blackmail is never attempted becomes more probable. The overall policy maximizes expected utility across all branches, evaluated from the prior
- UDT is reflectively stable: a UDT agent would not want to self-modify into something else, because it is already following the policy it would have chosen to commit to from the start

Descriptive Agent Foundations

From normative to descriptive

- **Normative (ideal) agent foundations:** characterizes how a perfectly rational agent *should* reason and act in principle – starts from desiderata of ideal agents and derives what follows
- **Descriptive agent foundations:** asks what agents we actually encounter in the wild actually *look like* – given any physical system (a neural network, a bacterium, a future AI), can we identify its goals, world-model, and decision-making structure?
- The aim is to develop concepts that apply to a wide range of intelligences, from bacteria to superintelligences

World-centric framing and robustness to scale

- Normative agency usually starts from the agent's subjective point of view (beliefs, preferences, rational choice). Embedded agency asks how this subjective picture maps onto a world where the agent is just another physical subsystem
- Descriptive agent foundations starts from the other direction: it begins with properties of the *world* (e.g. modularity, selection pressures, resource constraints) and asks what sorts of processes reliably steer the world into narrow target configurations
- **Robustness to scaling up:** normative foundations demands that our concepts do not break when the agent becomes vastly more capable than us – a safety property that fails under extreme optimization pressure is useless
- **Robustness to scaling down:** descriptive foundations additionally demands that the same framework degrades gracefully when applied to simpler systems.

Descriptive Agent Foundations

Physical constraints shape agent structure

- Descriptive foundations emphasize selection pressures, constraints from the physical world, and computational boundedness – these determine what kinds of agents actually arise
- For instance, a key property of the world is **modularity**: the world decomposes into subsystems that interact sparsely. This should make both planning and world-modelling easier for any agent that exploits it
- **World-modelling:** Bayesian networks exploit modularity – instead of maintaining a full joint hypothesis over every possible world, updates about variables are only propagated locally along edges in the graph
- **Planning:** General-purpose search exploits modularity – if the world is modular, an agent can pursue decoupled subgoals independently, rather than having to plan everything jointly

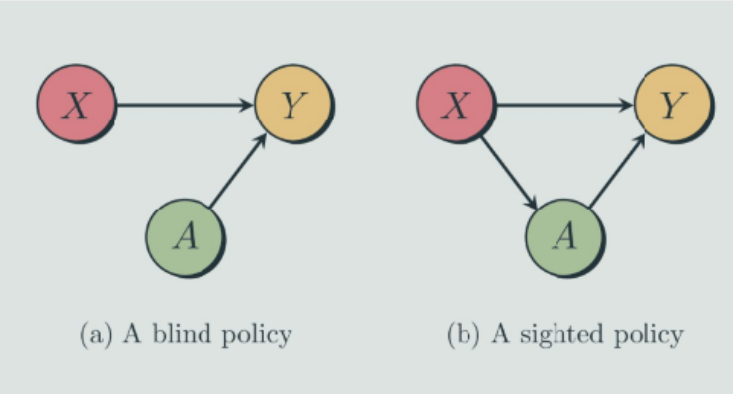
Optimization and Thermodynamics

Optimization as local entropy reduction

- Powerful agents reliably steer the world towards narrow regions of target configurations – outcomes that would be extremely unlikely to arise under any random process
- This is a form of *local entropy reduction*: concentrating probability mass from a broad initial distribution onto a narrow final distribution around a target
- Irreversible transitions: funnelling many initial states into the same targets
- Existing results in stochastic thermodynamics such as fluctuation theorems (Second law of thermodynamics), cost of information erasure and information thermodynamics directly constrains the shape of optimization processes and hold far from equilibrium

Optimization and Thermodynamics

Steering costs information (Touchette & Lloyd)



(a) A blind policy (b) A sighted policy

- A *sighted* agent observes X and chooses action A accordingly; a *blind* agent acts independently of X. The entropy reduction a sighted agent can achieve over the blind baseline is bounded:

$$\Delta H \leq \Delta H_{\text{Blind}}^{\text{Max}} + I(X; A)$$

- The degree to which an agent can steer outcomes beyond what a blind policy achieves is paid for by the mutual information between its observations and its actions – optimization requires information

Optimization and Thermodynamics

From ensembles to individual states

- Traditional thermodynamics defines entropy in terms of probability distributions over ensembles – but where does the distribution come from?
- **Algorithmic thermodynamics** (Ebtekar & Hutter) replaces Shannon entropy H with Kolmogorov complexity K, yielding thermodynamic laws that apply to *individual physical states* rather than ensembles
- Intuitively, a state in which all particles are concentrated in one location would have low entropy, because the repeated coordinates can be printed by a short program.

Knowledge as an endogenous physical resource

- In the Gibbs-Shannon framework, a probability distribution μ is an *exogenous* parameter specifying what the agent knows. But the insight from embedded agency is that an embedded agent's knowledge must be *physically encoded* in its brain or memory device
- In algorithmic thermodynamics, knowledge becomes an *endogenous* feature of the physical system: it manifests as **algorithmic mutual information** between the agent's memory and the environment
- This algorithmic mutual information plays same role as the setting in Touchette & Lloyd: it is the resource that an embedded agent can spend to steer the world into narrow target configurations
- From the embedded perspective: the agent's probabilistic knowledge about the environment is physically encoded in its memory, and this manifests as algorithmic mutual information between agent and environment. This algorithmic mutual information is precisely the resource that can be spent on steering – which is why an agent that *knows more* about its environment can perform *more optimization*

Conclusion

Conclusion

- We need robust concepts about agency to reason about systems far more capable than anything we have seen, and we need properties that are invariant under self-modification.
- **Tiling agents:** directly about making safety a reflectively stable property – ensuring that an agent's successors preserve the same guarantees, indefinitely
- **Logical induction:** about reasoning under logical uncertainty in an environment larger than yourself, where you cannot instantly compute the consequences of your hypotheses. Also relevant for reasoning about your own future beliefs and inputs – a form of self-reference
- **Decision theory:** a reflectively consistent degree of freedom – an agent using a bad decision theory will not automatically self-correct as it gets smarter. Also directly tied to embedded agency: counterfactuals are not well-defined when you are part of the environment, and the environment may contain copies of you or predictors modelling your decision procedure
- **Descriptive agent foundations:** starts from properties of the world (modularity, selection pressures, thermodynamic constraints) and derives what kinds of optimization processes arise – what agents embedded in the physical world would look like mechanistically
- **Connecting the two viewpoints:** reasoning about how properties of the world constrain the structure of optimization processes may tell us how an ideal agent *should* model the world and plan its actions from its subjective point of view