Position: Beyond Reasoning Zombies — AI Reasoning Requires Process Validity

Rachel Lawrence[*1], *Jacqueline Maasch*[*2] | *Equal contribution [1]Microsoft Research Cambridge [2]Cornell Tech

Reasoning is a (Learnable) Rule-Based Process

TL;DR Reasoning in generative AI has experienced unnecessary and addressable definitional ambiguity. This work advocates for the use of operational definitions for reasoning based on *process validity*, positioning reasoning as a learnable process grounded in *exact rule application*.

CORE POSITIONS

Thesis 1 ***Define, then measure.***
- 1.1 Research concerned with AI reasoning should provide formal operational definitions for the type of reasoning under investigation.
- 1.2 The construct validity of reasoning evaluation should be explicitly justified with respect to the operational definitions provided.

Thesis 2 ***Reasoning is a (learnable) rule-based process.***
- 2.1 Reasoning is a process of *exact rule application*. Learnable rules unambiguously map reasoning inputs to outputs and can include theorems, functions, policies, assumptions, etc., including rules pertaining to stochasticity, uncertainty, and approximation.

Thesis 3 ***Rule-based reasoning is valid.***
- 3.1 The *validity* of a reasoning process arises from exact rule application, independent of rule selection.

Construct Validity: Process ≠ Product

Current "reasoning benchmarks" often measure QA accuracy. We echo Chollet [2] on the risks of "confusing the process" (reasoning, in our case) "with the artifact produced by this process" (e.g., QA responses):

> "In the case of AI, the focus on achieving task-specific performance while placing no conditions on *how the system arrives at this performance* has led to systems that, despite performing the target tasks well, *largely do not feature the sort of human intelligence that the field of AI set out to build*."

Operationalizing Reasoning

Reasoning as a "rule-governed operation" [1]: ***Rules*** are operators whose operands are *information*, partitioned into (1) extrinsically obtained information (***evidence***) and (2) intrinsically generated information (***beliefs***).

Def. 1 (Reasoning, informal). The process of selecting and applying sequences of rules that act on prior beliefs and current evidence to obtain principled belief updates in evolving states.

Def. 2 (Reasoner). Goal-oriented decision-maker that implements reasoning.

Def. 3 (Reasoning, formal). Let $\mathcal{S}_t \coloneqq \langle \mathcal{B}_t, \mathcal{E}_t, \mathcal{R}_t \rangle$ denote the reasoner's state at time step t, where $\mathcal{B}_t$ denotes current belief, $\mathcal{E}_t$ denotes aggregated evidence up to time t, and $\mathcal{R}_t$ denotes the current set of established rules. Then, *reasoning* is the iterated application over steps t of rules $r \in \mathcal{R}_{t-1}$ to prior beliefs $\mathcal{B}_{t-1}$ and current evidence $\mathcal{E}_t$, by which we obtain dynamically updated states $\mathcal{S}_t$, and where every output $\mathcal{B}_t$ for $t > 0$ is the result of a rule application $r(\mathcal{B}_{t-1}, \mathcal{E}_t)$ to the contents of state $\mathcal{S}_{t-1}$.

Reasoning components		**Reasoner** components	
$t \in [0, ..., T]$	Reasoning step.	$\mathtt{s}_\mathtt{L} : \mathbf{R} \times \mathbf{B} \times \mathbf{E} \to \mathcal{R}^L$	Local rule selector.
$\{\mathcal{B}_i\}_{i=0}^T,\ \mathcal{B}_i \in \mathbf{B}$	Beliefs.	$\mathtt{s}_\mathtt{M} : \mathbf{R} \times \mathbf{B} \times \mathbf{E} \to \mathcal{R}^M$	Meta rule selector.
$\{\mathcal{E}_i\}_{i=0}^T,\ \mathcal{E}_i \in \mathbf{E}$	Evidence.	$\mathtt{s}_\mathtt{stop} : \mathbf{S} \to \{0,1\}$	Stopping rule.
$\{\mathcal{R}_i\}_{i=0}^T,\ \mathcal{R}_i \in \mathbf{R}$	Rule set.	$\mathtt{tr} : \mathbf{S} \times \mathcal{R}^L \times \mathcal{R}^M \times \mathbf{S} \to \Sigma^*$	Trace writer.
$\mathcal{S}_i \coloneqq \langle \mathcal{B}_i, \mathcal{E}_i, \mathcal{R}_i \rangle,\ \mathcal{S}_i \in \mathbf{S}$	States.	$\mathcal{T} \coloneqq \left\{\mathtt{tr}\left(\mathcal{S}_{i-1}, r_i^L, r_i^M, \mathcal{S}_i\right)\right\}_{i=1}^T$	Reasoning trace.
$\mathcal{R}_t^L \coloneqq \{r \in \mathcal{R}_t \mid r : \mathbf{B} \times \mathbf{E} \to \mathbf{B}\}$		where $r_i^L \coloneqq \mathtt{s}_\mathtt{L}(\mathcal{R}_t, \mathcal{B}_t, \mathcal{E}_{t+1})$ and $r_i^M \coloneqq \mathtt{s}_\mathtt{M}(\mathcal{R}_t, \mathcal{B}_t, \mathcal{E}_{t+1})$.	
$\mathcal{R}_t^M \coloneqq \{r \in \mathcal{R}_t \mid r : \mathbf{R} \times \mathbf{B} \times \mathbf{E} \to \mathbf{R}\}$			

Local rules $\mathcal{R}^L$ update beliefs, while ***meta rules*** $\mathcal{R}^M$ update the rule set.

Validity & Soundness

Rule selection and discovery can employ ***agency, intelligence, or creativity***. Rule application requires ***exactness***. This exactness gives rise to process validity, which is independent of soundness. This echoes Broome [1]:

> "Correct reasoning is not reasoning you are *required* to do by rationality, but reasoning you are *permitted* to do by rationality" (emphasis added).

Def. 4 (Validity). A transition from state $\mathcal{S}_t$ to $\mathcal{S}_{t+1}$ is *valid* if and only if it arises from the application of a rule $r \in \mathcal{R}_t$ to components of state $\mathcal{S}_t$.

Def. 5 (Soundness). A valid transition from $\mathcal{S}_t$ to $\mathcal{S}_{t+1}$ is *sound* if and only if all premises (as encoded by $\mathcal{B}$, $\mathcal{E}$, and $\mathcal{R}$) are true w.r.t. *external evaluation*.

Def. 6 (Reasoning zombie (r-zombie)). A system that superficially behaves as an autonomous reasoner, but lacks the mechanisms necessary for validity.

Open Q: What use cases *require reasoners*, and when do r-zombies suffice?

Algorithm 1 Valid reasoning as exact rule application.

Input. Initial rules $\mathcal{R}_0$, beliefs $\mathcal{B}_0$, evidence stream $\{\mathcal{E}_i\}_{i=1}^T$.

$\mathcal{R}, \mathcal{B}, \mathcal{E} \leftarrow \mathcal{R}_0, \mathcal{B}_0, \mathcal{E}_0$
$\mathcal{S} \leftarrow (\mathcal{R}, \mathcal{B}, \mathcal{E})$
$t \leftarrow 0$
while not $\mathtt{s}_\mathtt{stop}(\mathcal{S})$ **do**
  $\mathcal{E}' \leftarrow \mathcal{E}_{t+1}$
  $r^L \leftarrow \mathtt{s}_\mathtt{L}(\mathcal{R}, \mathcal{B}, \mathcal{E}')$ {Select local rule.}
  $\mathcal{B}' \leftarrow r^L(\mathcal{B}, \mathcal{E}')$ {Apply local rule, update beliefs.}
  $r^M \leftarrow \mathtt{s}_\mathtt{M}(\mathcal{R}, \mathcal{B}', \mathcal{E}')$ {Select meta rule.}
  $\mathcal{R}' \leftarrow r^M(\mathcal{R}, \mathcal{B}', \mathcal{E}')$ {Apply meta rule, update rules.}
  $\mathcal{S}' \leftarrow (\mathcal{R}', \mathcal{B}', \mathcal{E}')$
  $\mathcal{T}.\mathtt{append}(\mathtt{tr}(\mathcal{S}, r^L, r^M, \mathcal{S}'))$ {Update trace.}
  $\mathcal{R}, \mathcal{B}, \mathcal{E}, \mathcal{S} \leftarrow \mathcal{R}', \mathcal{B}', \mathcal{E}', \mathcal{S}'$
  $t \mathrel{+}= 1$
end while
Return $\mathcal{B}, \mathcal{T}$

Examples from Domain-Specific Reasoning

Example	**Beliefs** $\mathcal{B}_t$	**Evidence** $\mathcal{E}_t$	**Rules** $\mathcal{R}_t$	**Goal / Conclusion**
Logical deduction	Derived formulas in the proof state Γ.	Premises E_0 are given at $t = 0$ and not updated thereafter.	Fixed inference rules (e.g., modus ponens, introduction/elimination rules)	Derive a target formula φ such that $\Gamma \vdash \varphi$.
Bayesian inference	Current posterior $p(\theta \mid D_{1:t})$	Newly observed data D_t (possibly aggregated with prior observations).	Bayes' rule and any auxiliary update rules (e.g., conjugate prior updates, approximation schemes).	Obtain updated posterior beliefs $p(\theta \mid D_{1:t})$.
Reinforcement learning	Current value function estimates, policy parameters, and internal state representations.	Observed environment states, state transitions, and rewards obtained by interaction with the environment.	Update rules such as temporal difference or policy gradient updates, plus any meta rules adapting learning rates or architectures.	Learn a policy that maximizes expected return (i.e., select approximately optimal actions over time).
Nonmonotonic logic	Current set of accepted conclusions, including defeasible ones.	New information that may conflict with existing conclusions (e.g., exceptions, defaults).	Nonmonotonic inference rules that support belief revision and retraction, plus meta rules for revising the rule set itself.	Maintain a coherent, defeasible belief set that updates appropriately under new, possibly contradictory evidence.
Turing machine	Current configuration of the machine: tape contents $\sigma_t \in \Sigma^*$, head position h_t, and control state $q_t \in Q$, collectively encoded as $\mathcal{B}_t = \langle \sigma_t, h_t, q_t \rangle$.	Input word $w \in \Sigma^*$ (typically fixed at $t = 0$).	Transition relation or function $\delta_t : Q \times \Sigma \to Q \times \Sigma \times \{L, R\}$.	Compute the value of a (partial) function $f : \Sigma^* \rightharpoonup \Sigma^*$ on input w, i.e., reach a halting configuration with output tape σ_T such that σ_T encodes $f(w)$.
Probabilistic next-token prediction	Current token given prior tokens.	Token(s) provided at initialization (e.g., the first word of a sentence, the prompt to an instruction-tuned LLM, etc.).	Probability rules (e.g., chain rule), structural assumptions (e.g., Markovianity), maximum likelihood formulae.	Iterate procedure until query is answered (e.g., string is of desired length, etc.).

Open Questions on Rules & Validity in Neural Reasoning. Can the parameters of a neural network store rules, and if so, how do we locate them? Circuit discovery and other mechanistic interpretability methods aim to answer such questions [4, 6, 5]. Where are the mechanisms ensuring *exact application* of these rules? Neural program synthesis is a form of rule learning where exact rule application can be outsourced to a compiler [3].

References. **[1]** Broome. Rationality Through Reasoning. 2013. **[2]** Chollet. On the measure of intelligence. 2019. **[3]** Li, et al. Combining induction and transduction for abstract reasoning. 2025. **[4]** Olah, et al. Zoom in: An introduction to circuits. 2020. **[5]** Sharkey, et al. Open problems in mechanistic interpretability. 2025. **[6]** Shi, et al. Hypothesis testing the circuit hypothesis in LLMs. 2024.

https://bit.ly/ai-reasoning