

Conversation with Gemini

Please write a review of hehner.ca/EGT2.pdf

- Query successful

Review: *Epimenides, Gödel, Turing: an Eternal Gölden Twist* by Eric C.R. Hehner

Monograph URL: hehner.ca/EGT2.pdf

Author: Eric C.R. Hehner (University of Toronto)

Core Theme: Paradoxes, Inconsistency, Self-Reference, and the Reinterpretation of the Halting Problem

Executive Summary

Epimenides, Gödel, Turing: an Eternal Gölden Twist is a provocative mathematical essay that revisits foundational limits of computer science and logic. Paying homage in its title to Douglas Hofstadter's *Gödel, Escher, Bach: an Eternal Golden Braid*, Hehner's paper takes aim at the holy trinity of mathematical self-reference: the Epimenides (Liar) paradox, Gödel's Incompleteness Theorems, and Turing's proof of the unsolvability of the Halting Problem.

Hehner's central, radical claim is that **the Halting Problem is not an uncomputable problem, but rather a subjective, inconsistently defined specification.** By identifying a structural flaw he terms "twisted self-reference," Hehner argues that standard proofs do not demonstrate a limit on what computers

can calculate, but rather a limit on what humans can logically declare as valid questions.

The Core Argument: Objective vs. Subjective Specifications

The philosophical engine of the paper relies on a sharp distinction between **objective** and **subjective** specifications:

- **Objective Specifications:** The correct output depends strictly on the input, regardless of who or what is performing the computation (e.g., "What is the square of 5?").
- **Subjective Specifications:** The correct output changes depending on the agent performing the evaluation (e.g., "What is your IP address?" or "What is your name?").

Hehner argues that traditional proofs of the Halting Problem sneak subjectivity into a domain that claims to be objective. When a hypothetical halting-detector program H is passed a "twisted" program that asks H what it will do and then does the opposite, the specification becomes inherently subjective. The question is no longer a generic "Does this program halt?" but rather, "Can *you specifically* predict your own reaction to a program designed to spite you?"

Because the specification changes based on who is asking, Hehner argues that the resulting mathematical contradiction doesn't prove that a halting function is *incomputable*. Instead, it proves that the definition of the function is **inconsistent** under self-reference.

Demystifying the Trinity

Hehner systematically breaks down how this "twisted self-reference" manifests across historical milestones:

1. The Epimenides Paradox (The Liar)

If a sentence G states, "This sentence is false," standard logic struggles. Hehner models this mathematically by defining a truth-appraiser function B . When passed G , the definition forces $B("G")=false$ if $B("G")=true$. Hehner asserts that this is simply a broken, inconsistent definition of a function, not a fundamental breakdown of truth itself.

2. Gödel's Incompleteness

Hehner extends this logic to Gödel numbering. He posits that Gödel's proof constructs a mathematical sentence that essentially mimics a faulty definition. Rather than proving that arithmetic is fundamentally incomplete (containing truths that cannot be proven), Hehner argues it shows that certain self-referential mathematical definitions are fundamentally inconsistent.

3. Turing's Halting Problem

This is where the essay focuses its heaviest ammunition. Turing's proof relies on a program (often called twist or diagonal) that calls a halting detector, then loops infinitely if the detector says it halts, and halts if the detector says it loops.

Hehner writes out the full code of this paradox to show that the assignment of the variable halts results in an algebraic contradiction ($true=false$). He notes:

"It is inconsistent to ask for an X-program to compute halting for all X-programs due to a twisted self-reference... So it may be consistent and satisfiable to ask for a Y-program to compute halting for all X-programs."

In short: A computer *can* predict whether another computer's program will halt; a computer simply cannot consistently predict whether a program specifically engineered to dynamically contradict that computer's own prediction will halt.

Critique and Practical Impact

Strengths

- **Accessible Subversion:** Like much of Hehner's work, the paper is written with refreshing clarity. He dispenses with dense, obfuscated notations to lay out foundational paradoxes in clean, readable logic.
- **De-mystification of Computability:** For students paralyzed by the abstract finality of "incomputability," Hehner's framing treats computers as tools bounded by logical consistency, not magical limits. It reframes a computing limitation as a linguistic and definitional limitation.

Weaknesses / Controversies

- **Philosophical Divergence from Mainstream CS:** The mainstream computer science community views Turing's work as a definitive mathematical bound on algorithms. Hehner's reframing—arguing that the problem lies in the *specification* rather than the *capability* of the machine—can be viewed by critics as a semantic shift rather than a mathematical invalidation of Turing's architecture.
- **Practical Tooling Bounds:** Even if one accepts Hehner's philosophical stance that a language Y can compute the halting status of language X , building a practical, generalized static analyzer that evades all forms of adversarial self-reference remains an incredibly elusive engineering challenge.

Final Verdict

Epimenides, Gödel, Turing: an Eternal Gölden Twist is a compelling intellectual exercise that demands its reader to

unlearn a deeply entrenched computer science dogma. It is an excellent read for theoreticians, philosophers of logic, and software engineers who enjoy questioning the absolute boundaries of their craft. Hehner successfully strips the "mystique" away from diagonal arguments, replacing existential mathematical dread with clean, matter-of-fact boolean algebra.

Gemini is AI and can make mistakes.