Work Specification: Computational System for Autonomous Semantic Evolution

1. Project Goal:

- To create a computational system capable of exploring semantic spaces defined by formal systems.
- To develop internal mechanisms that allow the system to detect the limits of its current semantics (incompleteness or explorative exhaustion).
- To lay the conceptual and implementational foundations for a system capable of autonomously generating new semantics (sets of rules/postulates) in response to such limits, inspired by the adaptability and evolution of biological systems.
- (Long-term/Speculative Vision): To explore whether such mechanisms could be relevant to the emergence of complex properties like self-awareness/consciousness.

2. Background / Premises (From the Initial Document):

- Formal systems (mathematics, logic) and biological systems (DNA, brain) can be viewed as entities defined by rules that process information.
- Biological systems show remarkable capabilities for evolution and adaptation (by modifying their "topology" or operative rules).
- Formal systems, in the classical sense, are static and encounter intrinsic limits (incompleteness, undecidability - Gödel's Theorem).
- The project aims to bridge this gap by attempting to endow a formal computational system with intrinsic evolutionary capabilities.

3. Central Problem to Solve:

How can a computational system, operating on a defined set of rules (current semantics):

a) Autonomously detect its internal limits or incompleteness?
b) Use this information to generate a new set of rules (a new semantics) that overcomes these limits?

4. Architecture / Conceptual Components of the System:

- **Semantic Archive (Database):**
  - Structure for storing the basic elements of semantics: Postulates, Definitions, Inference Rules.
  - Organized hierarchically or structurally (e.g., Tables, Records, Fields) where fields contain parameters defining the elements.
  - Each "Database" or set of Tables represents a specific "Semantics" or formal system.
- **Processable Representation:**
  - Strings (or equivalent data structures) encoding:
    - Logical statements (potential theorems).
    - Mathematical and logical operations.
    - References to Postulates, Definitions, already Derived Theorems.
- **Processor (Semantic):**
  - Computational engine for:
    - Composing/Generating the strings (initial process still to be detailed).
    - Processing strings by applying Inference Rules.

        - Verifying the validity (correctness) of derived theorems.
        - (Hypothesis) Capability to "understand" or operate on emergent semantics (aspect still to be defined implementation-wise).
- **Provenance Structure (Tree):**
    - Data structure (e.g., tree) to trace the derivation of each valid theorem/string back to its postulates/axioms of origin.
    - Allows for "bidirectional traversal" or "inverse recursion" (from theorem to postulates and vice versa) to understand context and dependencies.
- **Feedback and Analysis Module:**
    - A feedback loop that takes the Processor's output (processed strings, verification results) and feeds it back for further analysis/processing.
    - Module for statistical analysis of properties of the strings processed in the feedback loop (e.g., frequency of repetition, complexity, distribution).

5. Key Mechanisms / Processes to Develop:

- **String Composition/Generation:** Define the rules or algorithm by which strings are created from elements in the Semantic Archive (currently a "larval idea" on how to make it intelligent).
- **Theorem Derivation and Verification:** Implement the Processor's algorithm for applying Inference Rules to strings and verifying the formal validity of derived theorems.
- **Provenance Tracking:** Implement the tree structure and functions to ascend the derivation chain of a theorem.
- **Limit / Incompleteness Detection (Crucial Research Area):**
    - Implement the feedback loop.
    - Develop mechanisms to search for **Self-Reference** in strings or processes (inspired by Gödel/Hofstadter): define the internal encoding and the computational recognition of self-reference (point still to be detailed).
    - Develop the statistical analysis of string properties in the feedback loop (e.g., measure properties, calculate distributions like Gaussian).
    - Define **how** statistical anomalies (e.g., in 2-sigma tails) and/or self-reference detection are **interpreted** as signals of semantic limit or incompleteness.
- **New Semantics Creation (The "Million-Dollar Question"):**
    - Define the mechanism (highly speculative at present) by which the system, having detected a limit, generates a *new* set of Postulates/Rules.
    - How this generation is guided by the information about the encountered limits (e.g., from statistical analysis tails or the nature of detected self-reference).
    - How the new semantics relates to previous ones (includes, modifies?).

6. Sub-Goals / Development Phases (Example):

- **Phase 1:** Implementation of Semantic Archive (Database), basic String representation, basic Processor for simple derivation/verification.
- **Phase 2:** Implementation of Tree Structure and Provenance Tracking.
- **Phase 3 (Case Study):** Application to Euclidean Geometry (Postulates 1-4), attempt to derive/not derive the 5th Postulate. Validation of the ability to probe independence.
- **Phase 4:** Implementation of Feedback Module and basic Statistical Analysis (measure properties, calculate distributions).
- **Phase 5:** Development of hypotheses and attempts to implement the interpretation of statistical/self-referential signals as limit indicators.
- **Phase 6:** (Very Long Term) Exploration of mechanisms for autonomous generation of new postulates.

7. Key Challenges / Open Questions:

- Computationally defining the Processor's "semantic" capabilities.
- Designing efficient algorithms for intelligent string composition (avoiding combinatorial explosion).
- Addressing the enormous **computational cost** of exploration and analysis in complex systems.
- Precisely defining and **computably detecting** formal self-reference within the system's string framework.
- Establishing the rigorous link between statistical anomalies/self-reference and formal/semantic limits.
- The actual mechanism of generating novel rules/semantics (the creative leap).
- Empirical validation (how to demonstrate that the system is *truly* evolving its semantics meaningfully and not just randomly?).