TITLE

AI Semantic Instruction Set System Based on 64-Dimensional Semantic Fractal Topological Matrix and Implementation Method Thereof

BACKGROUND OF THE PRESENT INVENTION

FIELD OF INVENTION

[0001] The present invention relates generally to artificial intelligence computing systems, and more particularly to a semantic-level instruction set architecture for artificial intelligence systems. More specifically, the invention relates to a system and method that implement artificial intelligence computation using a 64-dimensional semantic fractal topological matrix as a minimum complete semantic execution unit, thereby enabling executable, verifiable, auditable, and cross-model semantic reasoning

DESCRIPTION OF RELATED ARTS

[0002] Contemporary artificial intelligence systems, including large language models and transformer-based architectures, rely predominantly on token-level statistical representations of natural language. In such systems, tokens correspond to characters, sub-words, or words, and reasoning emerges implicitly from high-dimensional parameter optimization rather than from explicitly defined semantic execution rules.

[0003] This paradigm suffers from several fundamental technical limitations.

[0004] First, existing systems lack a minimum complete semantic unit. Tokens do not correspond to formally defined semantic roles such as intent, causality, agency, or constraint. As a result, semantic consistency cannot be guaranteed or verified at execution time.

[0005] Second, existing systems are non-auditable and non-verifiable. Inference paths are implicit within parameterized neural networks and cannot be deterministically reconstructed or formally validated, rendering such systems unsuitable for regulated, safety-critical, or governance-oriented applications.

[0006] Third, existing architectures exhibit extreme computational inefficiency, requiring massive parameter counts and hardware resources to approximate semantic reasoning that remains probabilistic rather than deterministic.

[0007] Fourth, existing systems conflate language generation with computation, preventing natural language from serving as a true machine-level instruction representation analogous to an instruction set architecture (ISA) in conventional computing systems.

[0008] Accordingly, there exists a need for a fundamentally different artificial intelligence computing architecture in which semantic meaning itself becomes the primary computational object, rather than textual tokens or statistical correlations.

SUMMARY OF THE PRESENT INVENTION

[0009] The invention is advantageous in that it provides an AI semantic instruction set system based on a 64-dimensional semantic fractal topological matrix, referred to herein as WAO-64-CORE, wherein a 64-dimensional semantic atom vector as a minimum complete semantic computation unit is defined, each dimension corresponding to a logically irreducible semantic role.

[0010] In another aspect, the invention defines a 64×64 semantic fractal topology matrix configured to represent semantic composition, dependency, recursion, and equivalence relationships between semantic atoms.

[0011] In another aspect, the invention provides a semantic instruction set architecture (Semantic ISA) in which each instruction comprises an opcode, a 64-dimensional semantic vector, dependency parameters, constraint parameters, and verification parameters.

[0012] In another aspect, the invention provides a semantic intermediate representation (WAO-IR) that compiles natural language or structured tasks into executable semantic instructions, enabling cross-model and cross-platform semantic consistency.

[0013] In another aspect, the invention enables auditable semantic execution graphs, multi-agent semantic collaboration, hardware-level semantic execution, and verifiable semantic causality chains

[0014] Still further objects and advantages will become apparent from a consideration of the ensuing description and drawings.

[0015] These and other objectives, features, and advantages of the present invention will become apparent from the following detailed description, the accompanying drawings, and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is a block diagram illustrating an overall architecture of an AI semantic instruction set system according to a preferred embodiment of the present invention.

[0017] FIG. 2 is a schematic representation view of a 64-dimensional semantic atom vector according to the above preferred embodiment of the present invention.

[0018] FIG. 3 is a schematic view illustrating a 64×64 semantic fractal topological matrix defining recursive semantic relationships according to the above preferred embodiment of the present invention.

[0019] FIG. 4 is a schematic view illustrating a semantic instruction format according to the Semantic ISA according to the above preferred embodiment of the present invention.

[0020] FIG. 5 is a block diagram illustrating a semantic compilation pipeline from natural language input to WAO-IR and semantic instruction execution according to the above preferred embodiment of the present invention.

[0021] FIG. 6 is a block diagram illustrating a semantic execution graph representing dependency-based semantic execution according to the above preferred embodiment of the present invention.

[0022] FIG. 7 is a block diagram illustrating a multi-agent semantic execution environment according to the above preferred embodiment of the present invention.

[0023] FIG. 8 is a block diagram illustrating a hardware-mapped embodiment of the semantic instruction set according to the above preferred embodiment of the present invention.

[0024] FIG. 9 is a block diagram illustrating a semantic audit and verification workflow according to the above preferred embodiment of the present invention.

[0025] FIG. 10 is a schematic view illustrating a hardware semantic processing architecture comprising a semantic processing unit and associated execution components according to the above preferred embodiment of the present invention.

[0026] FIG. 11 is a schematic view illustrating a semantic audit, verification, and governance architecture according to the above preferred embodiment of the present invention.

[0027] FIG. 12 is a schematic view illustrating a pipelined semantic processing architecture of a semantic processing unit according to the above preferred embodiment of the present invention.

[0028] FIG. 13 is a schematic view illustrating a layered semantic instruction execution architecture and heterogeneous deployment environments according to the above preferred embodiment of the present invention.

[0029] FIG. 14 is a schematic view illustrating a semantic transformation, verification, and state management workflow according to the above preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0030] The following description is disclosed to enable any person skilled in the art to make and use the present invention. Preferred embodiments are provided in the following description only as examples and modifications will be apparent to those skilled in the art. The general principles defined in the following description would be applied to other embodiments, alternatives, modifications, equivalents, and applications without departing from the spirit and scope of the present invention.

[0031] The present invention relates to an artificial intelligence semantic instruction set system and an implementation method thereof, wherein semantic meaning itself is treated as an executable computational object. Unlike conventional artificial intelligence systems that rely on token-level statistical representations or parameterized neural network inference, the present invention defines a deterministic, structured, and auditable semantic computation architecture.

[0032] In conventional computing systems, computation is performed through execution of machine instructions operating on registers and memory. In contrast, conventional artificial intelligence systems lack a minimum executable semantic unit, and semantic reasoning emerges implicitly from probabilistic correlations among tokens. As a result, such systems are not verifiable, auditable, or semantically deterministic.

[0033] Referring to FIG.1 to FIG. 9, the present invention introduces a semantic instruction set architecture in which semantic meaning is discretized into a formally defined minimum complete unit and executed through a defined instruction format, thereby enabling artificial intelligence systems to perform semantic computation in a manner analogous to instruction-level execution in classical computing architectures.

[0034] According to a preferred embodiment of the present invention, semantic computation is organized hierarchically into a plurality of abstraction layers including a semantic atom layer, a semantic fractal topology layer, a semantic instruction layer, a semantic execution graph layer, and a semantic audit and verification layer, each of which is formally defined and interoperable.

[0035] Referring to FIG. 2, the present invention defines a semantic atom as a minimum complete semantic computation unit represented by a sixty-four dimensional vector. The semantic atom is expressed as a vector A comprising elements a_1 through a_{64}, wherein each element corresponds to a logically irreducible semantic role.

[0036] In the preferred embodiment, the sixty-four semantic dimensions collectively form a closed semantic basis set. Removal or omission of any dimension results in loss of semantic completeness, thereby preventing full semantic representation of intent, action, causality, or constraint.

[0037] Each semantic dimension represents a distinct semantic role that is explicitly defined and interpretable. Unlike latent embedding vectors generated through machine learning, the semantic atom dimensions are not learned implicitly but are instead deterministically assigned based on predefined semantic definitions.

[0038] The semantic dimensions include, without limitation, semantic roles associated with intent definition, agent identification, action execution, object or resource reference, causal dependency, constraint enforcement, state representation, and verification or audit requirements.

[0039] Each semantic dimension may be encoded as a scalar value, a symbolic identifier, a categorical value, or a bounded numerical range, depending on implementation requirements and system constraints.

[0040] According to the preferred embodiment, semantic atoms are generated by a semantic compiler configured to parse natural language input, structured task descriptions, or programmatic instructions, and to map semantic components of such input onto corresponding semantic dimensions in accordance with predefined semantic grammars, ontologies, or rule sets.

[0041] Because the semantic atom is deterministically encoded, identical semantic input produces identical semantic atom representations, thereby enabling repeatability, verification, and auditability of semantic computation.

[0042] Referring to FIG. 3, the present invention further defines a semantic fractal topological matrix M comprising sixty-four rows and sixty-four columns. Each matrix element M_{ij} represents a semantic relationship between the semantic dimension a_i and the semantic dimension a_j.

[0043] The semantic relationships represented by the matrix include, without limitation, dependency relationships, causal relationships, equivalence relationships, reinforcement relationships, exclusion relationships, and hierarchical containment relationships.

[0044] According to the preferred embodiment, the semantic fractal topological matrix exhibits self-similarity across semantic scales, such that submatrices of the matrix represent semantically complete subsystems capable of independent semantic reasoning.

[0045] The fractal nature of the matrix enables recursive semantic composition, whereby higher-level semantic structures are constructed through repeated application of matrix-based transformations to semantic atom vectors.

[0046] According to the preferred embodiment, semantic reasoning depth increases logarithmically with task complexity rather than quadratically as in attention-based transformer architectures, thereby reducing computational complexity while preserving semantic expressiveness.

[0047] The semantic fractal topological matrix is further configured to preserve one or more semantic invariants, including intent preservation, constraint consistency, and causal closure, such that semantic execution does not violate predefined semantic rules.

[0048] These semantic invariants enable formal verification of semantic execution paths and prevent semantic drift during multi-step reasoning processes.

[0049] According to one preferred embodiment of the present invention, the sixty-four semantic dimensions forming the semantic atom structure are organized as a binary fractal semantic space generated from six binary semantic elements, each binary semantic element representing a fundamental dual semantic state.

[0050] In this embodiment, combinations of the six binary semantic elements generate sixty-four distinct semantic states, each semantic state corresponding to a unique configuration of semantic intent, action, state, causality, or constraint. The resulting sixty-four-state semantic space forms a closed and complete semantic topology, wherein each semantic state is uniquely identifiable and distinguishable from all other semantic states.

[0051] Such a sixty-four-state binary semantic topology corresponds structurally to a canonical hexagram-based state system historically represented by sixty-four hexagrams, each hexagram being composed of six binary components arranged in a stacked configuration.

[0052] In the present invention, the hexagram-based representation is used solely as a topological and combinatorial semantic mapping framework, and not as a philosophical, symbolic, or metaphysical construct. Each hexagram-corresponding semantic state represents a deterministic semantic configuration that may encode, without limitation, a semantic intent pattern, a state transition condition, a causal configuration, or a constraint relationship.

[0053] Adjacency relationships, transformation rules, and transition paths between the sixty-four semantic states are encoded within the semantic fractal topological matrix, thereby defining permissible semantic transitions during execution of semantic instructions. In this manner, the semantic fractal topological matrix operates as a formally defined semantic state-transition matrix governing movement between hexagram-corresponding semantic states during semantic instruction execution.

[0054] The use of a sixty-four-state binary fractal topology enables recursive semantic composition, deterministic semantic evolution, and complete semantic coverage while preserving auditability and verification of semantic execution paths.

[0055] It will be appreciated by one skilled in the art that alternative symbolic or graphical representations of the same sixty-four-state binary semantic topology may be used without departing from the spirit and scope of the present invention

[0056] Referring to FIG. 4, the present invention defines a semantic instruction set architecture in which each semantic instruction comprises a structured instruction format including an operation code, a semantic atom operand, dependency information, constraint information, and verification information.

[0057] According to the preferred embodiment, a semantic instruction is represented as a data structure including an opcode field identifying a semantic operation type, a sixty-four dimensional semantic atom vector field, one or more dependency fields defining execution prerequisites, one or more constraint fields defining semantic or operational restrictions, and one or more verification fields defining audit or validation requirements.

[0058] The opcode field may specify semantic instruction classes including, without limitation, intent definition instructions, action execution instructions, state transition instructions, dependency binding instructions, constraint assertion instructions, and verification instructions.

[0059] Execution of a semantic instruction operates directly on semantic atoms rather than on tokens, embeddings, or numerical tensors, thereby enabling semantic-level execution rather than statistical inference.

[0060] Semantic instruction execution is deterministic, such that given identical semantic atom inputs and identical dependency and constraint conditions, the execution produces identical semantic outputs and state transitions.

[0061] Referring to FIG. 5, the present invention further provides a semantic intermediate representation layer, referred to as WAO-IR, which functions as a semantic compilation interface between human-readable input and executable semantic instructions.

[0062] According to the preferred embodiment, natural language input, structured task descriptions, or programmatic commands are parsed by a semantic compiler into semantic atoms, which are then assembled into semantic instructions in accordance with the semantic instruction set architecture.

[0063] The WAO-IR is model-agnostic and execution-platform independent, such that identical semantic intermediate representations may be executed by different artificial intelligence models, rule-based systems, multi-agent systems, or hardware accelerators while preserving semantic equivalence.

[0064] As a result, the WAO-IR enables cross-model semantic consistency and portability that is not achievable in conventional artificial intelligence systems.

[0065] Referring to FIG. 6, semantic instructions generated in WAO-IR are organized into a semantic execution graph comprising a directed acyclic graph structure, wherein each node in the semantic execution graph corresponds to a semantic instruction, and each directed edge represents a dependency, causal relationship, or execution order constraint between semantic instructions. Semantic execution proceeds by evaluating dependency satisfaction and constraint compliance prior to executing each semantic instruction node.

[0066] During execution, semantic state transitions, matrix transformations, and verification outcomes are explicitly recorded, thereby enabling complete semantic traceability. The semantic execution graph provides a deterministic and auditable representation of semantic reasoning processes, enabling post hoc inspection, verification, and compliance analysis.

[0067] According to the preferred embodiment, the semantic instruction set system disclosed herein enables artificial intelligence systems to perform semantic computation with determinism, verifiability, and auditability that are not achievable using conventional token-based architectures. The disclosed architecture further enables reduced computational complexity, hardware-level semantic execution, and coordinated multi-agent semantic reasoning. The semantic instruction set system disclosed herein is not limited to standalone artificial intelligence architectures, but is expressly configured to be integrated with, layered upon, or embedded within existing artificial intelligence platforms, including but not limited to large language model platforms, multimodal artificial intelligence systems, agent-based AI frameworks, and enterprise AI application stacks.

[0068] The semantic instruction set system can be operated as a semantic execution layer positioned logically above, below, or alongside a conventional large language model, such that the large language model functions as a semantic parser, semantic generator, or semantic actuator rather than as the primary reasoning engine.

[0069] Accordingly, natural language input received by a large language model is not directly used to generate output text, but is first parsed into semantic components that are compiled into semantic atoms and semantic instructions in accordance with the semantic instruction set architecture described herein.

[0070] The large language model may be used, without limitation, to assist in mapping natural language expressions to semantic dimensions, resolving ambiguity in intent or context, or generating candidate semantic atom assignments, while final semantic execution is governed by the deterministic semantic instruction execution system.

[0071] According to the preferred embodiment, the semantic instruction set system may also function as an intermediate semantic control layer that constrains, validates, or redirects outputs generated by a large language model to ensure that such outputs satisfy predefined semantic constraints, causal consistency rules, and verification requirements. In this manner, the semantic instruction set system transforms a probabilistic language generation model into a semantically governed execution engine, thereby enabling predictable, auditable, and verifiable behavior from otherwise non-deterministic models.

[0072] When the semantic instruction set system is integrated as a semantic compiler backend, a large language model produces a semantic intermediate representation candidate, and the semantic instruction set system validates, optimizes, and executes the candidate representation, such that the semantic intermediate representation serves as a contract between the language model and downstream execution systems, ensuring that semantic meaning is preserved regardless of model architecture, parameterization, or training data.

[0073] When the semantic instruction set system is implemented as a semantic runtime monitor that observes internal or external actions taken by an artificial intelligence system and reconstructs semantic execution graphs post hoc for auditing, verification, or compliance purposes, outputs generated by an artificial intelligence system are mapped back into semantic atoms and semantic instructions, enabling reconstruction of semantic causality even when the original system does not natively expose reasoning paths.

[0074] The semantic instruction set system may further be integrated into multi-agent artificial intelligence platforms, wherein multiple language models or AI agents communicate using semantic instructions rather than raw natural language, thereby enabling structured collaboration, dependency resolution, and task orchestration. In such multi-agent embodiments, semantic instructions function as a shared execution protocol that ensures semantic consistency across agents operating on heterogeneous models, architectures, or vendors.

[0075] The semantic instruction set system may also be integrated into enterprise AI applications, including but not limited to decision-support systems, legal analysis platforms, medical advisory systems, financial risk assessment systems, and automated compliance systems, wherein the semantic instruction set system enforces domain-specific constraints, regulatory requirements, or ethical rules at the semantic instruction level, thereby preventing invalid or non-compliant actions even when generated by a language model.

[0076] When the semantic instruction set system is applied to multimodal artificial intelligence platforms, wherein semantic atoms represent not only linguistic meaning but also visual, auditory, spatial, or sensor-derived semantics, semantic atoms may be generated from image recognition systems, speech recognition systems, or sensor inputs, and unified within the same semantic instruction execution framework.

[0077] The semantic instruction set system further enables cross-version and cross-model portability, such that semantic instructions generated using one version of a language model may be executed or verified using a different model version without semantic degradation. This portability enables long-term semantic stability in artificial intelligence applications, even as underlying models evolve or are replaced.

[0078] It is worth mentioning that the semantic instruction set system can be deployed as a platform-level service, accessible through application programming interfaces, enabling third-party applications to submit semantic tasks for compilation, execution, and verification. Also, the semantic instruction set system can be embedded directly within an artificial intelligence platform as a native semantic execution kernel, thereby transforming the platform into a semantic-operating-system-like environment.

[0079] Through these integration modes, the present invention enables existing artificial intelligence platforms to transition from probabilistic text generation systems into semantically executable, verifiable, and governable intelligence systems, without requiring replacement of existing models or retraining of foundational architectures.

[0080] In other words, the present invention is broadly applicable to current and future artificial intelligence systems, regardless of implementation technology, training paradigm, or deployment environment.

[0081] In the present invention, the semantic instruction set system is further configured to support multi-agent artificial intelligence execution, wherein a plurality of artificial intelligence agents cooperate to perform semantic tasks using a unified semantic instruction set architecture.

[0082] As used herein, an artificial intelligence agent refers to any autonomous or semi-autonomous computational entity capable of receiving semantic instructions, performing semantic execution, and producing semantic results, including but not limited to language models, reasoning engines, planning agents, perception agents, robotic controllers, or hybrid systems.

[0083] In conventional multi-agent artificial intelligence systems, agents communicate primarily through unstructured natural language messages or ad hoc data structures, resulting in ambiguity, inconsistency, and lack of formal dependency management. The present invention overcomes these limitations by defining semantic instructions as the common execution and communication substrate among agents.

[0084] According to the preferred embodiment, each agent participating in a multi-agent system is configured to receive, interpret, and execute semantic instructions encoded in accordance with the semantic instruction set architecture described herein. Semantic instructions exchanged among agents include explicit semantic atom vectors, dependency declarations, constraint parameters, and verification requirements, thereby ensuring that semantic meaning is preserved across agent boundaries.

[0085] Further, a complex semantic task is decomposed into a plurality of semantic sub-tasks, each represented by one or more semantic instructions arranged within a semantic execution graph. The semantic execution graph is constructed such that each node corresponds to a semantic instruction and each directed edge represents a dependency, causal relationship, or execution ordering constraint between instructions. Semantic task decomposition may be performed by a semantic compiler, a coordinating agent, or a supervisory execution engine, and may leverage domain-specific knowledge, optimization heuristics, or agent capability profiles. Each semantic sub-task is assigned to one or more agents based on capability matching, resource availability, trust level, or execution constraints defined within the semantic instruction.

[0086] According to the preferred embodiment, a semantic scheduler is configured to evaluate the semantic execution graph and determine an execution order that satisfies all declared dependencies and constraints. The semantic scheduler operates at the semantic instruction level rather than at the token or message level, thereby enabling precise coordination of agent actions based on semantic meaning rather than textual interpretation.

[0087] Before executing a semantic instruction, the semantic scheduler verifies that all dependency conditions associated with the instruction have been satisfied and that all constraints remain valid. If a dependency is not satisfied or a constraint violation is detected, execution of the corresponding semantic instruction is deferred, rejected, or rerouted in accordance with predefined semantic policies. This dependency-aware scheduling mechanism ensures causal correctness and prevents inconsistent or contradictory agent actions.

[0088] According to the preferred embodiment, semantic instructions that do not share dependencies are executed in parallel by multiple agents, thereby enabling scalable and efficient semantic computation. Parallel execution is coordinated through the semantic execution graph, which explicitly encodes concurrency opportunities while preserving semantic correctness.

[0089] In distributed embodiments, agents may reside on different computing nodes, networks, or administrative domains, yet still cooperate seamlessly through the shared semantic instruction set. Because semantic instructions are deterministic and self-describing, execution results produced by one agent may be consumed by another agent without ambiguity or semantic loss.

[0090] In multi-agent environments, conflicts may arise when multiple agents attempt to perform incompatible actions or propose contradictory semantic outcomes. In the present invention, such conflicts are detected and resolved at the semantic instruction level rather than at the textual or behavioral level.

[0091] In one embodiment, semantic conflicts are identified by analyzing constraint violations, dependency cycles, or incompatible semantic atom states within the execution graph. Conflict resolution strategies may include prioritization of instructions, negotiation among agents using semantic instructions, rollback of semantic state transitions, or invocation of verification and arbitration rules. Because semantic instructions include explicit verification fields, conflict resolution outcomes may be formally validated and audited.

[0092] In one embodiment, each agent is associated with a semantic capability profile describing the types of semantic instructions the agent is authorized or capable to execute. Capability profiles may include performance characteristics, domain expertise, trust levels, or verification requirements. The semantic scheduler uses these profiles to assign semantic instructions to appropriate agents, thereby preventing unauthorized or unreliable execution. Trust levels may further determine whether an agent's semantic execution results require additional verification or cross-validation by other agents.

[0093] Upon completion of semantic instruction execution by individual agents, semantic results are aggregated into a unified semantic state representation. Aggregation is performed using semantic atom composition rules and semantic matrix transformations, ensuring that combined results remain semantically consistent. The aggregated semantic state may serve as input to subsequent semantic instructions, higher-level reasoning processes, or external applications.

[0094] According to the preferred embodiment, each semantic instruction executed by an agent generates an execution record including input semantic atoms, applied transformations, resulting semantic states, and verification outcomes. These execution records are appended to a semantic audit log that provides a complete, ordered trace of multi-agent semantic execution. The semantic audit log enables reconstruction of agent behavior, verification of compliance with constraints, and post-hoc analysis of semantic decision-making processes.

[0095] The multi-agent semantic execution framework disclosed herein enables structured cooperation among heterogeneous artificial intelligence agents with deterministic semantics, formal dependency management, and auditable execution. This framework eliminates ambiguity inherent in natural language-based agent communication and enables scalable, reliable, and governable multi-agent artificial intelligence systems.

[0096] Referring to FIG. 10 and FIG. 11, in addition to software-based implementations, the semantic instruction set system disclosed herein is configured to be implemented, in whole or in part, in hardware, firmware, or hybrid hardware–software architectures, thereby enabling deterministic semantic execution at an instruction level. In the present invention, semantic instructions are treated as first-class executable entities, enabling direct mapping to hardware execution units in a manner analogous to conventional instruction set architectures.

[0097] According to the preferred embodiment, the semantic instruction set system is implemented within a semantic processing unit 100 configured to directly execute semantic instructions encoded in accordance with the semantic instruction set architecture. The semantic processing unit 100 can be embodied to comprise a semantic instruction decoder 110, one or more semantic execution units 120, a semantic atom register file 130, a semantic topology matrix memory 140, and a semantic verification unit 150.

[0098] The semantic instruction decoder 110 is configured to receive a semantic instruction stream and parse each semantic instruction to extract an opcode field, a semantic atom operand field, dependency parameters, constraint parameters, and verification parameters.

[0099] The semantic atom register file 130 is configured to store a plurality of semantic atom vectors 131, each semantic atom vector comprising sixty-four semantic dimensions, and supports read and write operations required for semantic instruction execution.

[00100] The semantic topology matrix memory 140 is configured to store one or more semantic fractal topological matrices 141, each matrix comprising a sixty-four by sixty-four dimensional matrix defining semantic relationships among semantic dimensions.

[00101] The semantic execution units 120 are configured to apply matrix-based semantic transformations, semantic composition rules, and semantic state transition operations to semantic atom vectors 131 in accordance with decoded semantic instructions.

[00102] The semantic verification unit 150 is configured to evaluate verification parameters associated with semantic instructions to determine whether semantic invariants, dependency conditions, and constraint rules are satisfied during execution.

[00103] According to the preferred embodiment, the semantic processing unit 100 executes semantic instructions using a pipelined execution architecture comprising an instruction fetch stage 161, a decode stage 162, a semantic execution stage 163, a verification stage 164, and a commit stage 165.

[00104] During the instruction fetch stage 161, semantic instructions are retrieved from a semantic instruction memory 170 or received from an external semantic execution controller 180.

[00105] During the decode stage 162, the semantic instruction decoder 110 extracts semantic instruction fields and prepares dependency and constraint metadata for evaluation.

[00106] During the semantic execution stage 163, the semantic execution units 120 apply semantic topology matrix transformations using the semantic fractal topological matrices 141 to transform semantic atom vectors 131.

[00107] During the verification stage 164, the semantic verification unit 150 evaluates semantic invariants, constraint conditions, and dependency satisfaction criteria associated with the semantic instruction.

[00108] During the commit stage 165, resulting semantic states are written back to the semantic atom register file 130 or to an external semantic state memory 190, and a semantic execution record is generated.

[00109] In another preferred embodiment, the semantic instruction set system can be embodied to be implemented as a semantic accelerator 200 configured to operate in conjunction with a host processor 210 selected from a central processing unit, a graphics processing unit, or a tensor processing unit.

[00110] The semantic accelerator 200, which is implemented as an application-specific integrated circuit, a field-programmable gate array, or a system-on-chip module, offloads

semantic matrix operations, semantic instruction execution, and semantic verification tasks from the host processor 210.

[00111] The host processor 210 is configured to communicate with the semantic accelerator 200 via a communication interface 220 selected from memory-mapped interfaces, command queues, or dedicated semantic instruction channels. This configuration enables high-throughput semantic execution while minimizing computational overhead on the host processor 210.

[00112] According to the preferred embodiment, semantic atom vectors 131 are stored in a semantic atom memory region 300 arranged in contiguous memory layouts optimized for vectorized access. The semantic topology matrices 141 are stored in a semantic matrix memory region 310 using dense or sparse storage formats depending on semantic connectivity density.

[00113] In addition, fractal submatrices 142 derived from the semantic topology matrices 141 are cached in a semantic cache memory 320 to accelerate recursive semantic computations. Semantic execution graphs and dependency metadata are stored in a semantic graph memory 330 configured to support efficient traversal and scheduling operations.

[00114] As shown in FIG. 12, it is worth mentioning that the semantic instruction execution is performed through a plurality of pipeline stages including the instruction fetch stage 161, the decode stage 162, the semantic execution stage 163, the verification stage 164, and the commit stage 165. The instruction fetch stage 161 retrieves semantic instructions from the semantic instruction memory 170 or from the external semantic execution controller 180. The decode stage 162 cooperates with the semantic instruction decoder 110 to extract semantic instruction fields and prepare dependency and constraint metadata. The semantic execution stage 163 applies semantic transformations using the semantic execution units 120 and the semantic fractal topological matrices 141 to process the semantic atom vectors 131. The verification stage 164 cooperates with the semantic verification unit 150 to evaluate semantic invariants, constraint conditions, and dependency

satisfaction criteria. The commit stage 165 writes resulting semantic states to the semantic atom register file 130 or to the external semantic state memory 190, and generates semantic execution records. The semantic processing unit 100 may further cooperate with the semantic accelerator 200 operating in conjunction with the host processor 210 through the communication interface 220.

[00115] According to the preferred embodiment, as shown in FIG. 13, the semantic instruction execution is distributed across a software layer 400 and a hardware layer 500, wherein semantic compilation, scheduling, and orchestration are performed by software, and semantic transformations and verification are performed by hardware.

[00116] The software layer 400 dynamically generates semantic instructions, updates semantic topology matrices, and modifies constraint rules without requiring hardware redesign. The hardware layer 500 executes semantic instructions deterministically using the semantic processing unit 100 or the semantic accelerator 200.

[00117] The semantic instruction set system may be deployed in a centralized server environment 600, a distributed cloud environment 610, an edge computing environment 620, or an embedded system environment 630.

[00118] In cloud embodiments, multiple semantic processing units 100 may be virtualized and orchestrated to provide elastic semantic execution capacity.

[00119] In edge or embedded embodiments, a lightweight semantic execution core 700 derived from the semantic processing unit 100 is deployed to enable real-time semantic reasoning under constrained resources.

[00120] The disclosed architecture supports heterogeneous deployment, wherein different portions of semantic execution are performed by different hardware platforms.

[00121] In one embodiment, semantic execution includes redundant semantic execution paths 800 to detect execution inconsistencies. The semantic verification unit 150 compares

execution outcomes across redundant paths to ensure semantic consistency. Upon detection of a fault condition, semantic state rollback logic 810 reverts semantic atom states and triggers re-execution of affected semantic instructions.

[00122] In one embodiment, semantic instructions are executed within isolated semantic execution domains 900 to prevent unauthorized access or semantic interference. Access to the semantic topology matrix memory 140, the semantic atom register file 130, and semantic instruction streams is controlled by semantic access control logic 910.

[00123] This configuration enables secure multi-tenant semantic execution suitable for enterprise, governmental, or regulated environments.

[00124] It is worth mentioning that hardware and hybrid embodiments of the semantic instruction set system enable deterministic, high-performance semantic execution not achievable using conventional probabilistic artificial intelligence models. The disclosed architectures provide scalability, reliability, security, and auditability while preserving semantic interpretability.

[00125] The semantic instruction set system disclosed herein is further configured to provide comprehensive semantic audit, verification, governance, and compliance capabilities, thereby enabling artificial intelligence systems to operate in regulated, safety-critical, and accountability-driven environments.

[00126] Unlike conventional artificial intelligence systems in which inference paths are implicit and non-reconstructable, the present invention records semantic execution explicitly at the semantic instruction level, enabling deterministic reconstruction of semantic reasoning processes.

[00127] According to the preferred embodiment, referring to FIG. 14, the semantic instruction set system includes a semantic audit module 1000 configured to record execution metadata for each semantic instruction. The semantic audit module 1000 is operatively coupled to the semantic processing unit 100, the semantic accelerator 200, or

both, and receives semantic execution records generated during the commit stage of semantic instruction execution. The semantic instruction 1000 is input to a semantic transformation decoder 1010, which decodes and interprets the semantic instruction into an internal semantic representation. The semantic transformation decoder 1010 cooperates with semantic transformation logic 1020 to apply semantic transformation rules and generate transformed semantic content. A semantic state 1030 is accessed and updated by the semantic transformation decoder 1010 to reflect semantic state transitions resulting from execution of the semantic instruction. The semantic transformation decoder 1010 further communicates with a semantic verification module 1040, which evaluates the transformed semantic content against semantic invariants, constraint conditions, or governance rules, and provides verification feedback to the semantic transformation logic 1020 to ensure semantic correctness and compliance.

[00128] Each semantic execution record includes, without limitation, an instruction identifier, an opcode identifier, one or more input semantic atom vectors 131, applied semantic topology matrix identifiers 141, resulting semantic atom vectors 131, execution timestamps, and verification outcomes. The semantic audit module 1000 stores semantic execution records in a semantic audit log 1010, which may be implemented as a tamper-resistant memory structure, append-only log, or cryptographically secured ledger. The semantic audit log 1010 is configured to support chronological traversal, dependency-based traversal, and semantic-dimension-based querying.

[00129] The present invention may further include a semantic verification engine 1020 configured to evaluate semantic execution against predefined verification rules, semantic invariants, and compliance policies. Verification rules may include intent preservation rules, constraint satisfaction rules, causal consistency rules, and state transition validity rules.

[00130] The semantic verification engine 1020 operates in real time during semantic instruction execution, preventing execution of semantic instructions that violate verification rules.

[00131] In another preferred embodiment, the semantic verification engine 1020 operates in post-execution mode to validate completed semantic execution graphs for audit or certification purposes.

[00132] Verification results are appended to the semantic audit log 1010 and associated with corresponding semantic execution records.

[00133] According to the preferred embodiment, the semantic audit module 1000 is configured to reconstruct semantic causality chains by traversing the semantic execution graph and correlating dependency edges with semantic execution records. This reconstruction enables identification of which semantic instructions, semantic atoms, and dependency conditions led to a particular semantic outcome. Semantic causality reconstruction enables explainability of artificial intelligence decisions in terms of formal semantic reasoning rather than statistical inference.

[00134] The semantic instruction set system further includes a semantic governance module 1030 configured to enforce governance policies at the semantic instruction level. Governance policies may include ethical rules, legal constraints, organizational policies, jurisdictional requirements, or domain-specific regulations.

[00135] In one embodiment, governance policies are encoded as semantic constraints that are evaluated prior to semantic instruction execution. If a proposed semantic instruction violates a governance policy, the instruction is rejected, modified, or escalated for human review. Governance decisions and policy enforcement outcomes are recorded in the semantic audit log 1010 for traceability.

[00136] The disclosed system is configured to support compliance with regulatory frameworks requiring transparency, accountability, and traceability of artificial intelligence decision-making.

[00137] According to the preferred embodiment, the semantic audit module 1000 generates compliance reports based on semantic execution records, including summaries

of executed semantic instructions, applied constraints, and verification outcomes. Compliance reports may be generated automatically in response to audit requests from regulators, internal compliance teams, or external auditors.

[00138] The semantic instruction set system further supports selective disclosure of semantic execution records based on access permissions, thereby protecting sensitive information while enabling compliance verification.

[00139] According to the preferred embodiment, the semantic instruction set system supports human-in-the-loop oversight through a semantic review interface 1040. The semantic review interface 1040 enables authorized human operators to inspect semantic execution graphs, semantic atom states, verification results, and governance decisions. Human operators may approve, modify, or reject semantic instructions or execution plans prior to execution or during execution pauses triggered by governance rules. Human interventions are recorded as semantic events in the semantic audit log 1010 to preserve a complete audit trail.

[00140] According to the preferred embodiment, the semantic governance module 1030 supports cross-system governance, wherein semantic policies are enforced consistently across multiple artificial intelligence platforms or deployments. Governance policies may be versioned and associated with jurisdiction identifiers, enabling semantic execution to adapt dynamically to geographic or regulatory contexts. This capability enables deployment of artificial intelligence systems across multiple legal or organizational domains while preserving consistent semantic behavior.

[00141] According to the preferred embodiment, the semantic audit log 1010 is cryptographically secured using hashing, digital signatures, or distributed ledger techniques. Semantic execution records may be chained or notarized to prevent unauthorized modification or deletion. This ensures integrity and non-repudiation of semantic audit data.

[00142] The semantic audit, verification, governance, and compliance framework disclosed herein enables artificial intelligence systems to operate with levels of transparency, accountability, and trust not achievable using conventional probabilistic models. The disclosed framework enables regulatory compliance, ethical governance, and explainability while preserving the performance and scalability of semantic execution.

[00143] The embodiments described herein are illustrative of the principles of the present invention and are not intended to limit the scope of the invention. Various modifications, substitutions, and alternatives may be made without departing from the spirit and scope of the invention as defined by the appended claims.

[00144] It is appreciated that although the preferred embodiment describes a semantic atom comprising sixty-four semantic dimensions, the inventive concept is not limited to a fixed dimensionality. In alternative embodiments, the semantic atom may comprise fewer or greater than sixty-four dimensions, provided that the selected dimensionality forms a closed and complete semantic basis capable of representing intent, action, causality, constraint, state, and verification semantics.

[00145] For example, a reduced-dimension semantic atom is used for resource-constrained environments, while in another embodiment, an expanded-dimension semantic atom is used to represent higher-order semantic abstractions. The semantic fractal topological matrix is correspondingly resized to match the dimensionality of the semantic atom, while preserving recursive and self-similar properties.

[00146] For another example, the semantic fractal topological matrix is implemented as a dynamically reconfigurable matrix whose values are updated in response to semantic learning, policy changes, or environmental feedback.

[00147] Further, multiple semantic topology matrices are maintained in parallel, each corresponding to a different semantic domain, governance regime, or operational context. Semantic instruction execution may select among multiple semantic topology matrices based on context identifiers included in semantic instructions.

[00148] Further, the semantic instruction set system is implemented entirely in software executing on general-purpose computing hardware without dedicated semantic processing units.

[00149] In such examples, semantic atom operations and semantic matrix transformations are implemented using vectorized software libraries, numerical computation frameworks, or symbolic reasoning engines.

[00150] In virtualized environments, semantic execution units may be instantiated as isolated virtual machines, containers, or serverless execution contexts. In distributed embodiments, semantic instruction execution is performed across a plurality of network-connected nodes, each node executing a portion of a semantic execution graph. Semantic instructions and semantic atoms are transmitted between nodes using structured semantic messaging protocols rather than raw natural language messages. Distributed execution enables fault tolerance, load balancing, and geographic distribution of semantic computation.

[00151] In the present invention, the semantic instruction set system incorporates adaptive mechanisms that adjust semantic mappings or matrix parameters based on feedback, performance metrics, or observed outcomes. Such adaptive mechanisms do not alter the deterministic execution semantics of individual semantic instructions, but rather refine the selection or weighting of semantic relationships over time. Learning-assisted adaptation may be performed offline, under supervision, or subject to governance constraints to prevent uncontrolled semantic drift.

[00152] Further, domain-specific semantic extensions are layered on top of the core semantic instruction set architecture. Such extensions may include specialized semantic dimensions, instruction opcodes, or verification rules tailored to specific application domains including healthcare, finance, law, robotics, manufacturing, education, or public administration. Domain-specific extensions remain interoperable with the core semantic instruction set through shared semantic atom structures and execution rules.

[00153] Further, semantic instructions include version identifiers enabling backward and forward compatibility across different versions of the semantic instruction set architecture. Compatibility mechanisms enable semantic execution systems to interpret, validate, or translate semantic instructions generated under earlier or later versions. This capability enables long-term stability of semantic applications despite evolution of underlying semantic definitions.

[00154] The semantic instruction set system of the present invention may be integrated with non-artificial-intelligence systems including databases, workflow engines, control systems, and enterprise software platforms. In such integrations, semantic instructions may trigger external system actions, query external data sources, or control physical devices while preserving semantic auditability.

[00155] In one embodiment, the semantic instruction set system includes failure handling logic that enables graceful degradation of functionality when semantic execution resources are unavailable. Failure handling may include fallback execution paths, reduced semantic resolution modes, or deferred execution strategies. Failure events and recovery actions are recorded in the semantic audit log to preserve traceability.

[00156] In one embodiment, the semantic instruction set architecture is exposed through standardized interfaces to enable interoperability among independently developed semantic execution systems. Such interfaces enable third-party systems to generate, submit, or verify semantic instructions without access to proprietary implementation details.

[00157] One skilled in the art will understand that the embodiment of the present invention as shown in the drawings and described above is exemplary only and not intended to be limiting.

[00158] It will thus be seen that the objects of the present invention have been fully and effectively accomplished. The embodiments have been shown and described for the purposes of illustrating the functional and structural principles of the present invention and

is subject to change without departure from such principles. Therefore, this invention includes all modifications encompassed within the spirit and scope of the following claims.

WHAT IS CLAIMED IS:

1. An artificial intelligence semantic instruction set system, comprising:

a semantic instruction execution core configured to execute semantic instructions;

a semantic atom structure comprising a plurality of semantic dimensions forming a minimum complete semantic computation unit;

a semantic topology matrix defining semantic relationships among the semantic dimensions of the semantic atom structure;

a semantic instruction set architecture defining a format for semantic instructions including at least an operation field and a semantic atom operand field; and

a semantic execution controller configured to apply the semantic topology matrix to the semantic atom structure during execution of the semantic instructions, wherein semantic meaning is represented and executed deterministically at a semantic instruction level independently of token-level language representations.

2. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic atom structure comprises sixty-four semantic dimensions forming a closed semantic basis set.

3. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic topology matrix comprises a sixty-four by sixty-four dimensional matrix defining dependency, causality, equivalence, exclusion, or reinforcement relationships among semantic dimensions.

4. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic topology matrix exhibits recursive or fractal self-similarity enabling multi-level semantic composition.

5. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein each semantic instruction further comprises at least one dependency field, constraint field, or verification field.

6. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein execution of semantic instructions produces deterministic semantic state transitions independent of probabilistic inference.

7. The artificial intelligence semantic instruction set system, as recited in claim 1, further comprising a semantic intermediate representation layer configured to compile natural language input into executable semantic instructions.

8. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic intermediate representation layer is model-agnostic and executable across heterogeneous artificial intelligence models.

9. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the system is integrated with a large language model such that the large language model functions as a semantic parser or generator and semantic execution is governed by the semantic instruction execution core.

10. The artificial intelligence semantic instruction set system, as recited in claim 1, further comprising a semantic execution graph defining dependency relationships among semantic instructions.

11. The artificial intelligence semantic instruction set system, as recited in claim 10, wherein semantic instructions are distributed among a plurality of artificial intelligence agents for parallel or sequential execution.

12. The artificial intelligence semantic instruction set system, as recited in claim 11, wherein the plurality of artificial intelligence agents communicate using semantic instructions rather than natural language messages.

13. The artificial intelligence semantic instruction set system, as recited in claim 11, wherein execution scheduling is performed based on dependency resolution at a semantic instruction level.

14. The artificial intelligence semantic instruction set system, as recited in claim 11, wherein semantic conflicts among agents are detected and resolved using constraint evaluation or semantic verification rules.

15. The artificial intelligence semantic instruction set system, as recited in claim 1, further comprising a semantic audit module configured to record execution records for semantic instructions.

16. The artificial intelligence semantic instruction set system, as recited in claim 15, wherein the execution records include semantic atom inputs, semantic topology matrix identifiers, resulting semantic states, and verification outcomes.

17. The artificial intelligence semantic instruction set system, as recited in claim 15, wherein the semantic audit module enables reconstruction of semantic causality chains.

18. The artificial intelligence semantic instruction set system, as recited in claim 1, further comprising a semantic verification engine configured to enforce semantic invariants during execution.

19. The artificial intelligence semantic instruction set system, as recited in claim 18, wherein semantic invariants include intent preservation, constraint satisfaction, or causal consistency.

20. The artificial intelligence semantic instruction set system, as recited in claim 1, further comprising a semantic governance module configured to enforce ethical, legal, organizational, or regulatory policies at a semantic instruction level.

21. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic instruction execution core is implemented in hardware as a semantic processing unit.

22. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic processing unit comprises a semantic instruction decoder, a semantic atom register file, a semantic topology matrix memory, and a semantic verification unit.

23. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic instruction execution core is implemented as a hardware accelerator coupled to a host processor.

24. The artificial intelligence semantic instruction set system, as recited in claim 23, wherein the hardware accelerator offloads semantic matrix operations and semantic verification from the host processor.

25. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein semantic instructions are executable in software, hardware, or hybrid software-hardware environments.

26. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic instruction set system is deployed in a cloud, edge, embedded, or distributed computing environment.

27. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the semantic instruction set architecture supports versioning for backward or forward compatibility.

28. The artificial intelligence semantic instruction set system, as recited in claim 1, wherein the number of semantic dimensions differs from sixty-four while forming a complete semantic basis.

29. The artificial intelligence semantic instruction set system, as recited in claim 3, wherein the sixty-four semantic dimensions correspond to sixty-four discrete semantic states organized according to a binary fractal topology.

30. The artificial intelligence semantic instruction set system, as recited in claim 29, wherein the binary fractal topology corresponds to a sixty-four-state combinatorial structure generated from six binary elements.

31. The artificial intelligence semantic instruction set system, as recited in claim 30, wherein the sixty-four-state combinatorial structure corresponds to a canonical sixty-four-state symbolic topology historically represented by sixty-four hexagrams.

32. The artificial intelligence semantic instruction set system, as recited in claim 31, wherein each of the sixty-four hexagrams represents a distinct semantic state defined by a unique combination of semantic intent, action, state transition, or constraint relationships.

33. The artificial intelligence semantic instruction set system, as recited in claim 31, wherein adjacency, transformation, or transition relationships between the sixty-four hexagrams are encoded in the semantic fractal topological matrix to define permissible semantic state transitions.

34. A method for semantic execution in an artificial intelligence system, comprising steps of:

(a) representing semantic meaning using a semantic atom structure comprising a plurality of semantic dimensions;

(b) defining semantic relationships among the semantic dimensions using a semantic topology matrix;

(c) compiling input information into semantic instructions formatted according to a semantic instruction set architecture; and

(d) executing the semantic instructions by applying the semantic topology matrix to the semantic atom structure, wherein semantic execution is deterministic and auditable.

35. The method, as recited in claim 34, further comprising a step of recording semantic execution records enabling reconstruction of semantic causality.

36. The method, as recited in claim 34, further comprising a step of enforcing semantic constraints and verification rules during execution.

37. The method, as recited in claim 34, wherein the input information comprises natural language processed by a large language model.

38. The method, as recited in claim 34, wherein semantic instructions are executed by multiple artificial intelligence agents coordinated through a semantic execution graph.

39. The method, as recited in claim 34, wherein execution results are used to govern downstream actions, decisions, or physical device control.

40. The method, as recited in claim 34, wherein the plurality of semantic dimensions comprises sixty-four semantic dimensions arranged as a binary fractal semantic space.

41. The method, as recited in claim 40, wherein the binary fractal semantic space corresponds to a sixty-four-state topology generated from six binary semantic elements.

42. The method, as recited in claim 41, wherein the sixty-four-state topology corresponds to a hexagram-based semantic state space.

43. The method, as recited in claim 42, further comprising mapping semantic state transitions between hexagram states using the semantic topology matrix during execution of semantic instructions.

AI Semantic Instruction Set System Based on 64-Dimensional Semantic Fractal Topological Matrix and Implementation Method Thereof

ABSTRACT OF THE DISCLOSURE

An artificial intelligence semantic instruction set system and an implementation method are disclosed. The system defines semantic meaning as an executable computational object using a semantic atom structure comprising a plurality of semantic dimensions forming a minimum complete semantic unit, and a semantic topological matrix defining relationships among the semantic dimensions. A semantic instruction set architecture is provided, in which semantic instructions include an operation field and semantic atom operands, enabling deterministic semantic execution independent of token-level language representations. A semantic intermediate representation compiles natural language or structured input into executable semantic instructions, which are executed using matrix-based semantic transformations. The system supports multi-agent coordination, hardware or software execution, semantic auditability, verification, and governance enforcement. The disclosed architecture enables verifiable, auditable, and cross-model semantic execution for artificial intelligence systems, including integration with large language model platforms, distributed computing environments, and domain-specific applications.