ACCEPTED MANUSCRIPT

Title AGL: actionable granular logic for verifiable specifications and reasoningin AI decision–action systems

Authors: Andrzej Jankowski

To appear in: Technical Sciences

Received 16 December 2025;

Accepted 22 December 2025;

Available online 22 December 2025.

AGL: ACTIONABLE GRANULAR LOGIC FOR VERIFIABLE SPECIFICATIONS AND REASONING IN AI DECISION–ACTION SYSTEMS

Andrzej Jankowski
ORCID: 0000-0002-0725-6354

Department of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn, Olsztyn, Poland

Keywords: Actionable Granular Logic; LLM Hallucinations; Guarded Fragments; Neuro-symbolic AI; Clinical Decision Support; Verifiable AI; OnkoBot.

Abstract

Decision–action systems in high-stakes domains, such as oncology, face a critical barrier: the inherent unpredictability and "hallucination" risks associated with Large Language Models (LLMs). We introduce AGL (Actionable Granular Logic) as a formal logical framework designed to serve as a *verifiable symbolic wrapper* for generative AI. AGL bridges the gap between stochastic model outputs and rigorous clinical protocols by employing a *Decidability Split*. This architecture encapsulates vague or probabilistic evidence into *Information Granules* (MT-FOGL), exposing them to a verifiable core exclusively via discrete *threshold atoms*.

Unlike traditional rigid logics, AGL formalizes a flexible spectrum of operational responses (*actions* based on medical and expert knowledge)—ranging from hard procedural stops to adaptive clinical recommendations, such as flagging missing diagnostic prerequisites. To ensure rigorous control, we restrict the reasoning core to guarded profiles (GF/RGF), guaranteeing decidability while addressing computational complexity through data locality and strategic computational budgeting. The framework's practical viability is demonstrated through a prototype of the OnkoBot system, developed in collaboration with the Maria Sklodowska-Curie National Research Institute of Oncology (NIO-PIB). While comprehensive performance metrics are deferred to a subsequent report currently being prepared by the OnkoBot team from NIO-PIB, UWM, and collaborating partners, initial results indicate that AGL effectively mitigates hallucinations by grounding agentic proposals in auditable, formal specifications, providing a scalable and trustworthy foundation for AI in mission-critical deployments.

*Corresponding: Andrzej Jankowski, Katedra Metod Matematycznych Informatyki, Wydział Matematyki I Informatyki, Uniwersytet Warmińsko-Mazurski w Olsztynie, ul. Słoneczna 54, 10-710 Olsztyn, e-mail: andrzej.jankowski@uwm.edu.pl.

1. Introduction and motivation

AI systems deployed in high-stakes (*mission-critical*) domains face requirements that differ qualitatively from typical applications: decisions have operational consequences, therefore *auditability* (the ability to reconstruct premises and the structure of reasoning) and *bounded verifiability* of the decision core become essential. Rely-ing exclusively on empirical safeguards (quality testing, red-teaming, safety filters) does not establish a formal correctness contract at critical points.

Modern agentic systems based on large language models offer high expressivity, yet create deployment barriers related, among others, to hallucinations and limited logical accountability (the difficulty of formally justifying *why* a particular recommendation was produced). In practice, decision–action systems must also represent vague and uncertain notions, predictive model outputs, risk assessments, decision thresholds, and missing data.

The central premise of this paper is to couple (i) an auditable description of the knowledge state and (ii) a formal description of procedures (workflows), while keeping the verification interface within a profile that supports meta-theoretic analysis (decidability, complexity). To this end, we propose MT-FOGL (knowledge state and granularity) and AGL (procedures with regular-program syntax), with an explicit separation between the computation layer and the verifiable layer (the *Decidability Split*).

Remark 1.1 (Naming convention). In this paper, AGL denotes the canonical action-oriented formalism. The terms GF/RGF refer exclusively to the *verifiability profile* (restrictions ensuring decidability), and not to the name of the framework nor to paper titles.

For a complete list of acronyms and abbreviations used in the paper (clinical and IT/AI), see Annex A.

2. Problem setting and design goals

We consider systems that (i) maintain a knowledge state about objects (e.g., a case, a process, a patient), (ii) produce reasoning in the form of conclusions/qualifications (when a rule is applicable), and (iii) execute workflows consisting of decision and execution steps. The system must be auditable: it should support answering which premises and which logical structure led to a decision or recommendation.

We capture the design requirements as follows:

G1. *Auditability:* the reasoning core operates on explicit, discrete predicates and rules, inspectable by design.

G2. *Bounded verifiability:* core properties are subject to algorithmic verification within a decidable fragment.

G3. *Separation of responsibility:* complex computations (statistical/ML models, numerical methods) are encapsulated outside the core.

G4. *Action orientation:* the formalism describes not only "what is true" but also "what to do" via programs/procedures.

G5. *Deployability:* constructs map naturally to system components (KB-state, execution, audit trail).

Remark 2.1 (Disclaimer: illustrative nature of medical examples). All medical-context examples (formulas, rules, procedures, thresholds) are strictly illustrative (non-normative) and serve to demonstrate the formal constructions of AGL. They are not clinical recommendations nor medical advice and must not be interpreted as a formalization of any particular guideline. In real deployments, the content of rules, thresholds, and procedures requires validation by qualified experts and adaptation to local standards and data.

2.1. An example application of AGL architecture and the Decidability Split

We explain the intuition behind the AGL architecture and the Decidability Split using Figures 1–2, step by step.

Reading guide for Figure 1 (step-by-step). Figure 1 presents the operational mechanism of AGL as a compilation–verification pipeline that produces explicit audit evidence. The workflow can be read as follows:

Step 1 (Surface layer). A domain expert (e.g., a clinician) or an engineer specifies human-meaningful rules and AGL programs that describe decision conditions and workflow steps (actions, branching, iteration, escalation).

Step 2 (Grammar-grounded translation). The specification is translated by a grammar-grounded compiler into a formal representation accepted by the verification core. This step makes the mapping from the surface language to the verifiable core explicit and auditable.

Step 3 (Φ-constraints / verifiability filters). During translation, all tests and constraints used in programs are forced to belong to the selected verifiability profile, i.e., the Guarded Fragment / Regular Guarded Fragment (GF/RGF). This does *not* automatically guarantee global consistency of all domain assumptions; rather, it ensures that the verification tasks posed to the core remain algorithmically decidable and thus suitable for systematic checking.

Step 4 (Decidability Split via threshold atoms). Numerical, probabilistic, and vague premises are *not* processed inside the core. They are computed outside the core as *Information Granules* and exposed to the core only through Boolean *threshold atoms* (e.g., a computed risk score of 0.82 is mapped to the two-valued fact "risk > 0.8"). This is the Decidability Split: complex computation is encapsulated, while the core reasons over two-valued predicates.

Step 5 (Verification core). The verification core reasons about the compiled FO-PDL-style workflow under GF/RGF tests and produces machine-checkable outcomes (e.g., satisfiable/unsatisfiable, property holds/violated) together with evidence artifacts.

Step 6 (Decision policy). Verified outcomes are mapped to an operational policy such as *accept / abstain / escalate*. For example, the system may accept an action when the verified conditions hold, abstain when evidence is insufficient, and escalate when the risk or uncertainty triggers expert review.

Step 7 (Evidence artifacts / audit trail). For every decision and executed workflow, AGL can generate auditable artifacts: (i) proofs or proof sketches supporting the decision under the core assumptions, (ii) counterexamples when a property fails, and (iii) execution traces documenting which tests/rules fired and which actions were taken.

Key takeaway. Figure 1 illustrates the strict separation between the expressive layer and the verifiable core: decision thresholds appear only as Boolean *threshold atoms* in rules and tests; arithmetic is not part of the verifiable core.

Clinical reading guide for Figure 2 (three-level view). Figure 2 explains *why* the AGL approach can remain auditable and verifiable even when it uses complex AI components. It can be read as a three-level structure in which information becomes progressively more precise and suitable for formal checking.

Level 1 (bottom, blue: computation outside the core). This is where raw hospital data and complex models live (EHR, LIS/PACS, ML/statistics, simulations). Outputs here are typically numerical and uncertain (e.g., probabilities, scores). This layer is intentionally *not* subject to formal verification.

Level 2 (middle, green: auditable granulation interface, MT-FOGL). This is the key safety boundary (Decidability Split). It *translates* graded outputs into explicit Boolean *threshold atoms* that the core can use. For example, a computed risk of 0.82 is mapped into a two-valued premise such as "risk > 0.8". Because thresholds and mappings are explicit, they can be reviewed, versioned, and audited (e.g., which threshold was used and when it was crossed).

Level 3 (top, yellow: logical-procedural core). This is the only layer where formal verification is performed. The core reasons over two-valued premises and FO-PDL-style workflows with GF/RGF-bounded tests. As a result,

verification tasks remain algorithmic (decidable) and can produce evidence artifacts (proofs, counterexamples, and execution traces).

Practical implication. If a clinician asks "why did the system recommend action A?", the audit trail can point to the specific threshold atoms and workflow steps used in the core. Errors or uncertainty in numerical estimation affect *which* threshold atoms are supplied, but they do not break the verifiability of the core itself.

Thus, Figure 1 shows the end-to-end compilation and verification pipeline, while Figure 2 makes explicit the decisibility boundary that preserves auditability and algorithmic verification in the AGL core.

3. MT-FOGL: soft typing, *Information Granules*, and threshold atoms

3.1. Soft typing over a single universe

Definition 3.1 (MT-FOGL (soft typing))**.** Let L be a first-order signature containing a distinguished family of unary predicates $\{\mathsf{Type}_i(\cdot)\}_{i\in I}$ (soft types), where I is an index set enumerating available soft types and any additional symbols. An *MT-FOGL theory* is a set of first-order sentences over L, optionally extended by *typing axioms* that constrain admissible arguments of relations and functions.

Intuition. Soft types let us describe a single underlying universe U while still writing *typed* constraints and workflows. They behave like auditable "labels" (unary predicates) rather than a commitment to many-sorted semantics, which keeps the verifiable core close to standard first-order reasoning while remaining engineering-friendly.

Typed variables as syntactic sugar. For readability we use the notation:

$$\forall u : \mathsf{Patient}\ \varphi(u) \quad \text{as shorthand for} \quad \forall u\,(\mathsf{Patient}(u) \rightarrow \varphi(u)),$$

$$\exists t : \mathsf{Therapy}\ \psi(t) \quad \text{as shorthand for} \quad \exists t\,(\mathsf{Therapy}(t) \wedge \psi(t)).$$

3.2. Information Granules

In practical decision–action systems, the notions used by rules are rarely fully crisp. Concepts with vague boundaries, partial observability, and stochastic uncertainty are common. In MT-FOGL we capture these phenomena via *Information Granules*, and then expose them to the rule/procedure layer through auditable threshold atoms.

3.2.1. Fuzzy and probabilistic granules

Definition 3.2 (Information Granule (fuzzy/probabilistic))**.** Let U be a non-empty universe of objects. A *fuzzy information granule* is a function $\Phi_G : U \rightarrow [0,1]$ (membership degree), a *probabilistic information granule* is a function $P_G : U \rightarrow [0,1]$ (probabilistic score), and a *crisp information granule* is a function $C_G : U \rightarrow \{0,1\}$

Intuition. An information granule summarizes a potentially complex, model-based assessment (fuzzy membership, probability, or a crisp indicator) into a single score that can be logged, calibrated, and versioned. The verifiable core never manipulates this score numerically; it only consumes discrete facts derived from it (threshold atoms).

Remark 3.3 (Vagueness vs. probability). $\Phi_G(u)$ denotes a degree of membership (concept vagueness), whereas $P_G(u)$ denotes the probability of satisfying a criterion (stochastic uncertainty). Although both values lie in [0, 1], their interpretation and data sources differ.

3.2.2. A unified scoring symbol and threshold atoms

From the perspective of auditability and verifiability of the rule core, a *discrete interface* is essential. Therefore, numerical and probabilistic comparisons are compiled into atomic threshold predicates, which can be used in rules (GF/RGF) and in AGL program tests.

Definition 3.4 (Unified scoring symbol)**.** Let G be an information granule and let $\Psi \in \{\Phi, P, C\}$ denote the scoring family: $\Psi = \Phi$ for a fuzzy membership degree, $\Psi = P$ for a probabilistic score, and $\Psi = C$ for a crisp score with codomain $\{0, 1\} \subseteq [0, 1]$. Then $\Psi_G : U \to [0, 1]$ denotes Φ_G or P_G or C_G, respectively.

Definition 3.5 (Audit-bounded threshold set (granular signature))**.** For each application and each scoring family $\Psi \in \{\Phi, P, C\}$, we fix a finite set of rational thresholds $R_\Psi \subseteq [0, 1] \cap \mathbb{Q}$ called the *granular signature*. Only threshold predicates with $r \in R_\Psi$ are admitted in the verifiable core.

Intuition. Without an explicit bound, the family $\{G^\Psi_{\geq r}(\cdot)\}_{r \in [0,1]}$ is uncountable and cannot be treated as a practical predicate signature. The granular signature makes the interface finite, reviewable, and stable under audit: it is an explicit design choice that can be justified clinically/organizationally and version-controlled.

Intuition. The symbol Ψ_G is a notational unification: it lets us write common interface rules without committing to whether a score is fuzzy (Φ_G), probabilistic (P_G), or crisp (C_G). This improves readability and supports uniform compilation into threshold atoms.

Definition 3.6 (Threshold atoms for fuzzy/probabilistic granules)**.** For any threshold $r \in R_\Psi$ we introduce unary predicate symbols:

$$G^\Psi_{\geq r}(\cdot), \quad G^\Psi_{\leq r}(\cdot), \quad G^\Psi_{<r}(\cdot), \quad G^\Psi_{>r}(\cdot), \quad G^\Psi_{=r}(\cdot),$$

with the intended semantics (for $u \in U$):

$$M \models G^\Psi_{\geq r}(u) \iff \Psi^{\mathrm{M}}_G(u) \geq r, \qquad M \models G^\Psi_{\leq r}(u) \iff \Psi^{\mathrm{M}}_G(u) \leq r,$$

and analogously for <, >, and =.

Intuition. Threshold atoms are the *only* way numerical evidence enters the verifiable core. They act like a discretization boundary: the computation layer may change (models, calibration, population statistics), but the core sees only a finite, auditable set of two-valued facts.

Parameterized atoms and provenance (auditable evidence carriers). In the surface specification (e.g., an expert-facing DSL) one often writes parameterized conditions such as "$\Psi_G(u) \geq r$" with a configurable threshold r. In AGL this is compiled into the fixed predicate signature determined by the granular signature R_Ψ: only comparisons at $r \in R_\Psi$ become atoms in the verifiable core. The choice of R_Ψ, as well as the construction of

Ψ_G (expert-defined, learned/calibrated from data, or hybrid), must be recorded with provenance metadata (version,

data window, calibration method, responsible role), so that every threshold fact used in a decision–action workflow is traceable.

Remark 3.7 (*Decidability Split*). The complex estimation mechanisms behind Φ_G, P_G, and C_G (numerical/ML models, calibration, population-level analysis) are isolated outside the verifiable core. The rule and procedure layer operates solely on discrete threshold atoms, which stabilizes auditability and supports the GF/RGF verifiability profile.

3.2.3. Approximation granules in the Pawlak tradition

In Pawlak's classical approach, the starting point for rough-set granules is an information/decision system based on a fixed set of attributes. These are used to construct lower and upper approximations of vague concepts. Intuitively, such approximations arise from observability: attributes (our "glasses") induce an indiscernibility relation.

Definition 3.8 (Attribute language and atomic facts)**.** Let U be a universe of objects and let A be a set of attributes. Each attribute $a \in A$ has a value domain $V(a)$. Atomic facts are expressions of the form $a(x) = v$, where $x \in U$ and $v \in V(a)$.

Intuition. The attribute language isolates *what can be observed* (auditable atomic facts) from *how it is computed*. It provides a stable vocabulary for rules and program tests, so that verification concerns only this discrete interface, not the underlying estimation pipelines.

Definition 3.9 (Indiscernibility w.r.t. an attribute set)**.** Let $B \subseteq A$. Define the indiscernibility relation $\sim_B$ on U by:

$$x \sim_B y \quad \Longleftrightarrow \quad \forall b \in B \;\; b(x) = b(y).$$

3.2.4. Elementary granules and crisp granules in Pawlak's view

Let $[x]_B$ denote the equivalence class of x w.r.t. $\sim_B$. This is the *indiscernibility class induced by* B.

In particular, $[x]_B$ is the basic unit of granulation induced by $\sim_B$ and describes the finest level of distinguishability available under observation limited to the attribute set B.

Intuitively: an elementary granule groups objects that *cannot be distinguished* using observable attributes from B.

For a concept $X \subseteq U$:

$$\underline{X}^B := \{x \in U \mid [x]_B \subseteq X\}, \qquad \overline{X}^B := \{x \in U \mid [x]_B \cap X \neq \emptyset\}.$$

In the auditable core we use predicates $X^{\mathrm{low},B}$, $X^{\mathrm{up},B}$, $X^{\mathrm{bd},B}$.

For a fixed attribute set B, the class $[x]_B$ can be described by a conjunction of atomic conditions that crisply fix attribute values. In this perspective, crisp concepts are naturally described as alternatives (disjunctions) of elementary-granule descriptors, i.e., as unions of indiscernibility classes, corresponding to a set $C = \bigcup_{i=1}^{k} [x_i]_B$. This is an important semantic distinction: an *elementary granule* describes a single indiscernibility class, whereas a *crisp granule* (a crisp concept) can be a composition of many such classes via disjunction.

Remark 3.10 (Relevance for the auditable interface). The above view aligns well with the AGL architecture: descriptors of elementary granules and crisp granules are classical formulas, hence they can feed the auditable rule core without introducing arithmetic into the verifiable layer. Rough approximations $\underline{X}^B$ and $\overline{X}^B$ can then be understood as operations over families of such granules arising from limited observability (i.e., from the choice of B).

Remark 3.11 (Rough atoms as an auditable interface). The atomic interface for rough granules is two-valued (crisp) and has three key predicates:

$$X^{\text{low},B}(x) \equiv x \in \underline{X}^B, \quad X^{\text{up},B}(x) \equiv x \in \overline{X}^B, \quad X^{\text{bd},B}(x) \equiv x \in \overline{X}^B \setminus \underline{X}^B.$$

3.3. From clinical data to threshold atoms

The table identified as Table 5 illustrates the bridge: clinical data/source → granule (fuzzy/probabilistic/rough) → threshold atom used in a rule or a program test. It shows how clinical inputs and predictive model outputs are mapped to information granules and then to discrete predicates that form the only interface to the auditable rule core.

4. AGL: a procedural extension in the style of FO-PDL

4.1. Regular programs, tests, and (optionally) parallelism

In AGL we introduce a workflow/program language with regular-program syntax in the sense of PDL, with an additional (optional) parallel-composition constructor:

$$\pi ::= a \mid \pi_1; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \mid \varphi? \mid \pi_1 \parallel \pi_2.$$

Operator intuition (short, "for system users").

- a *(atomic action):* a single step in a procedure (e.g., ordering a test, starting therapy, updating the KB-state).
- $\varphi?$ *(test):* checks a condition without changing the KB-state; if φ fails, the execution path is blocked.
- $\pi_1; \pi_2$ *(sequence):* execute π_1 first, then π_2.
- $\pi_1 \cup \pi_2$ *(choice):* allow alternative paths (the system chooses one).
- π^* *(iteration):* repeat the program zero or more times (follow-up / retry pattern).
- $\pi_1 \parallel \pi_2$ *(parallelism):* two subprocesses may progress concurrently ("two tracks at once").

Tests as the verifiability interface (base profile). In practice, tests $\varphi?$ are the key interface between the MT-FOGL knowledge state and the procedural layer. Hence, in the base profile we assume that formulas φ used in tests belong to a core restricted to GF/RGF, to preserve control over decidability and complexity. Additionally, all numerical and probabilistic comparisons are admissible only through threshold atoms.

Why $\parallel$ is optional (expressivity vs. verifiability). The operator $\parallel$ increases language expressivity, but typically complicates semantics and worsens meta-theoretic properties of the verifiable core, in particular complexity and often decidability of verification problems. For this reason, we recommend not using $\parallel$ in the base AGL profile.

Two safe alternatives to ‖ in the base profile.

1. *Interleaving when order does not matter:*
$$(\alpha;\beta) \cup (\beta;\alpha).$$

2. *Atomic action as a bundled protocol:* when two actions are treated as one "package," introduce a new atomic action a_{combo}.

Remark 4.1 (Engineering takeaway). In this paper we focus on the base profile (without ‖), because it offers the best trade-off between expressivity and verifiability. Extensions with concurrency are treated as specialized variants and a direction for future work.

A note on decidability (base profile). Here we combine the procedural core (regular programs without explicit parallelism) with tests in a GF/RGF profile, which preserves decidability of verification-relevant problems (albeit with high worst-case complexity). If stronger verification guarantees are required in a given application, one can adopt an even more restrictive test subprofile (e.g., more constrained guarded variants), at the cost of expressivity.

4.2. Modalities and operational meaning

The modalities $[\pi]\,\varphi$ and $\langle\pi\rangle\varphi$ express properties that hold after executing a procedure. From an engineering viewpoint, programs are interpreted as transitions between knowledge-base states (KB-states): actions update facts in the KB, and tests filter admissible execution paths. This yields readable, auditable procedures with explicit sequencing and branching structure.

4.3. Formal semantics of AGL (core)

Definition 4.2 (KB-state as an MT-FOGL structure)**.** A *KB-state* (knowledge-base state) is an MT-FOGL structure M with a non-empty universe U^{M} and an interpretation of the signature L (in particular, type predicates and application predicates). Additionally, for each information granule G the structure provides external scoring functions $\Phi_G^{\mathrm{M}} : U^{\mathrm{M}} \to [0,1]$ (fuzzy) and/or $P_G^{\mathrm{M}} : U^{\mathrm{M}} \to [0,1]$ (probabilistic), which are *not* part of the first-order signature and may be computed by external, validated modules.

Definition 4.3 (Atomic actions as state transformers)**.** Each atomic action a is interpreted as a binary relation $\Rightarrow_a$ on KB-states. We write $\mathrm{M} \overset{a}{\Longrightarrow} \mathrm{M}'$ iff $(\mathrm{M},\mathrm{M}') \in \Rightarrow_a$. In the base profile, actions may update the fact layer (extensions of selected predicates), while the granule scoring functions Φ_G^{M}, P_G^{M} are treated as external and recomputed after updates.

Definition 4.4 (Program semantics and satisfaction of dynamic modalities)**.** Let $[\![\pi]\!] \subseteq \mathbf{M} \times \mathbf{M}$ denote the relation induced by program π on the set of KB-states **M**, defined inductively by:

$$[\![a]\!] := \Rightarrow_a, \quad [\![\pi_1;\pi_2]\!] := [\![\pi_1]\!] \circ [\![\pi_2]\!], \quad [\![\pi_1 \cup \pi_2]\!] := [\![\pi_1]\!] \cup [\![\pi_2]\!], \quad [\![\pi^*]\!] := ([\![\pi]\!])^*,$$

and for tests:

$$[\![\varphi?]\!] := \{(M,M) \mid M \models \varphi\}.$$

Then satisfaction of dynamic modalities is defined by:

$$\mathrm{M} \models [\pi]\varphi \iff \forall \mathrm{M}' \, ((\mathrm{M}, \mathrm{M}') \in [\![\pi]\!] \Rightarrow \mathrm{M}' \models \varphi),$$

$$\mathrm{M} \models \langle\pi\rangle\varphi \iff \exists \mathrm{M}' \, ((\mathrm{M}, \mathrm{M}') \in [\![\pi]\!] \wedge \mathrm{M}' \models \varphi).$$

Remark 4.5 (Profile restriction for verifiability). In the base AGL profile, tests $\varphi?$ are restricted to formulas from GF/RGF, and all numerical/probabilistic comparisons appear only via threshold atoms.

4.4. Why procedures are essential

A purely rule-based description may suffice for classification, yet is often insufficient when decisions require sequencing, alternatives, or control loops (follow-up, retry). AGL models workflow as regular programs, and tests $\varphi?$ act as auditable checkpoints, to which we apply GF/RGF profile restrictions.

5. Verifiability profile: Guarded Fragment (GF) and Regular Guarded Fragment (RGF)

5.1. Guarded Fragment (GF)

Definition 5.1 (Guarded quantification (informal)). An occurrence $\exists y\, \varphi(x, y)$ is *guarded* if it has the form $\exists y\, (\alpha(x, y) \wedge \varphi(x, y))$, where the guard atom $\alpha(x, y)$ contains all free variables of φ. Analogously, $\forall y\, \varphi(x, y)$ is guarded if it has the form $\forall y\, (\alpha(x, y) \rightarrow \varphi(x, y))$. A formula belongs to GF if all quantifiers occur in guarded form.

The guard as a data relation ("patient-centered" perspective). In practice, a guard corresponds to a relation that anchors new variables in the data: e.g., $\mathsf{HasImaging}(p, img)$, $\mathsf{HasLab}(p, lab)$, $\mathsf{HasReport}(p, r)$. Thus quantification ranges over explicit links rather than over arbitrary objects in the database. For example, the condition "there exists an imaging study for the patient" is guarded:

$$\exists img\big(\mathsf{HasImaging}(p, img) \wedge \mathsf{StudyType}(img, \mathrm{CT})\big),$$

where $\mathsf{HasImaging}(p, img)$ serves as the guard anchoring img to the patient context p.

5.2. RGF (Regular Guarded Fragment) as a controlled enrichment

In practice, a guard may be path-based: instead of a single relational atom, we want to refer to a regular pattern of links. Such constructs are captured by the *RGF* profile, which preserves decidability of satisfiability with typical 2ExpTime complexity [3].

Example 5.2 (Why RGF matters: anchoring along a path (schematic))**.** Consider a situation in which an object of interest is related to a patient only through an intermediate record. Let $\mathsf{PatientHasRecord}(p, r)$ and $\mathsf{RecordHasSample}(r, s)$ be binary relations. In many data models, samples s are not linked to the patient directly but via records r. A path-guard can be expressed as a regular relation

$$\pi := \mathsf{PatientHasRecord} \cdot \mathsf{RecordHasSample},$$

which anchors s in the context of p through r. Intuitively, it enables quantification "along a data path," rather than using a single guard atom.

Theorem 5.3 (Decidability facts (GF/RGF profile))**.** *Satisfiability of GF is decidable [1] and is 2ExpTime-complete with the finite model property [10]. Satisfiability of RGF is decidable and is 2ExpTime-complete in standard formulations [3].*

Practical verification pipeline (fragment selection → compilation → SAT → auditable result). In practice, the decidability facts above can be used as an *engineering pipeline*: (i) select the verifiability profile (GF or RGF) for all rule conditions and program tests; (ii) compile the chosen fragment of the specification (tests $\varphi?$, guards, and rule antecedents) into a normal form suitable for automated checking; (iii) reduce the resulting bounded satisfiability/consistency checks to a propositional encoding and run a SAT solver (or an SMT solver if a fixed, safe background theory is used *outside* the verifiable core); (iv) return an auditable certificate: the set of fragment assumptions, the compiled fragment identifier, the solver outcome, and (when available) a witness model/counterexample trace. This keeps numerical evidence outside the core (only threshold atoms appear in the compiled fragment) while providing a concrete, repeatable verification path for deployments.

Remark 5.4 (Computational perspective: asymptotic complexity vs. practical feasibility). The 2ExpTime-completeness results concern the worst case for the full class of formulas in the given fragment. In AGL, verification targets a restricted core (GF/RGF profile and specific procedures), and granule estimation is isolated outside the core (the *Decidability Split*), preventing arithmetic from being pulled into the verification problem. In practical decision–action systems, verification typically concerns constrained classes of queries and programs, often over highly structured data, which tends to reduce instance difficulty.

6. Elementary clinical *Information Granules* (non-normative)

6.1. Clinical acronyms used in illustrative examples

To avoid ambiguity, Table 1 lists the clinical abbreviations used in the illustrative examples of this paper. They are included only to keep the examples readable; the examples remain non-normative and are not intended as guideline encodings.

Acronyms. For a complete, alphabetically ordered list of clinical and IT acronyms used throughout the paper (including those used in the illustrative examples), see Annex A (Table 1).

This section is didactic: it shows how *Information Granules* (the computation layer) may look in practice, and how we move from numerical scores to auditable threshold atoms used in the rule/procedure core. All examples are *non-normative* and do not constitute a formalization of any specific guideline.

In particular, we treat these snippets as *narrative benchmarks* for illustrating the Decidability Split, rather than as a clinically binding encoding of standards of care. This distinction matters because European oncology guidelines are regularly updated by leading scientific societies—for example, EAU for prostate cancer and ESMO for lung and chest tumours [6, 8].

6.2. Basic diagnostic granules (prostate cancer, schematic)

Example 6.1 (Elementary clinical granules (schematic))**.** Let u denote a patient. We illustrate several *fuzzy* granules $\Phi_G(u) \in [0,1]$ and one *crisp* granule.
(1) *PSA granule (fuzzy):*

$$\Phi_{\mathrm{G_{PSA}}}(u) = \begin{cases} 0, & if\ PSA(u) < 4 \\ \dfrac{PSA(u) - 4}{6}, & if\ 4 \le PSA(u) \le 10 \\ 1, & if\ PSA(u) > 10 \end{cases}$$

(2) PSA granule (fuzzy):

$$\Phi_{\mathrm{G_{PSAD}}}(u) = \min\left(1, \frac{PSAD(u)}{0.15}\right)$$

(3) *Age as risk-factor granule (fuzzy)*

$$\Phi_{\mathrm{G_{Age\_HR}}}(u) = \begin{cases} 0, & if\ Age(u) < 70 \\ \frac{Age(u)-70}{10}, & if\ 70 \le Age(u) \le 80 \\ 1, & if\ Age(u) > 80 \end{cases}$$

(4) *Gleason threshold crisp)*

$$\Phi_{\mathrm{G_{Gleason\ge 7}}} = \begin{cases} 0, & if Age(u) < 70 \\ \dfrac{Age(u) - 70}{10}, & if\ 70 \le Age(u) \le 80 \\ 1, & if\ Age(u) > 80 \end{cases}$$

(4) *Adherence (probabilistic)*

$$P_{G_{\mathrm{Compliance}}}(u) = P(\mathrm{adherence} \mid \mathrm{History}(u), \mathrm{FamilySupport}(u)).$$

Example 6.2 (Probabilistic granule: risk of lymph-node metastases (LN+))**.** Consider a probabilistic granule $G_{\mathrm{LN+}}$ that assigns to a patient u a pre-operative estimate of the risk of lymph-node metastases. Let

$$P_{G_{\mathrm{LN+}}}(u) \in [0,1]$$

denote a probabilistic score computed by an external, validated module (e.g., the Briganti nomogram; 4). For instance, given a set of clinical features one may use a logistic model of the form:

$$P_{G_{\mathrm{LN+}}}(u) = \big(1 + \exp - (-1.78 + 0.35 \cdot \ln(\mathrm{PSA}(u)) + 1.15 \cdot \mathrm{GS_{primary}}(u) + 0.73 \cdot \mathrm{cT}(u) + 0.02 \cdot \mathrm{PBx_{pos}}(u))\big)^{-1}.$$

where $\mathrm{GS}_{\mathrm{primary}}(u)$ is the dominant Gleason pattern, $\mathrm{cT}(u)$ is the clinical stage, and $\mathrm{PBx}_{\mathrm{pos}}(u)$ is the fraction of positive biopsy cores.

In accordance with the *Decidability Split*, the computation above is not part of the verifiable core. The AGL core sees only a discrete interface in the form of threshold atoms, e.g., for a chosen policy threshold $r \in [0,1]$:

$$G^{P}_{\mathrm{LN+}\geq r}(u).$$

Such an atom can then feed qualification rules or program tests $\varphi?$ within the GF/RGF profile.

6.3. From numerical scores to auditable threshold atoms

In line with the *Decidability Split*, comparisons such as $\Phi_G(u) \geq r$ do not enter the rule core. Instead, we use *threshold atoms* as discrete predicates employed in rules and in program tests.

Example 6.3 (Threshold atoms for multi-level decisions (schematic))**.** For illustrative thresholds $r_1, r_2 \in [0,1]$ define atoms:

$$G^{\Phi}_{\mathrm{PSA}\geq r_1}(u), \quad G^{\Phi}_{\mathrm{PSAD}\geq r_2}(u), \quad G^{\Phi}_{\mathrm{Gleason}\geq 7}(u).$$

A simple qualification condition (in the classical core) can be written as:

$$G^{\Phi}_{\mathrm{Gleason}\geq 7}(u) \wedge \left(G^{\Phi}_{\mathrm{PSA}\geq r_1}(u) \vee G^{\Phi}_{\mathrm{PSAD}\geq r_2}(u)\right).$$

All arithmetic remains encapsulated in Φ_G and is not part of the verifiable core.

Remark 6.4 (Where to place richer clinical example libraries). If a broader library of examples (diagnostic/therapeutic/follow-up) is needed, a natural place is an appendix or a separate application-focused paper. In P1 we keep them intentionally short so as not to dilute the formal contribution of AGL.

6.4. Canonical clinical formulas using threshold atoms (non-normative)

In this subsection we show how typical "clinical formulas" can be expressed in the classical core (MT-FOGL + GF/RGF profile) exclusively via *threshold atoms*. All numerical comparisons are encapsulated in the definitions of Φ_G and P_G and exposed as discrete predicates $G^{\Psi}_{\Phi \geq r}(u)$.

Example 6.5 (Qualification for biopsy (multi-level, threshold-atom style))**.** Let u denote a patient. Introduce the following threshold atoms (illustrative thresholds):

$$G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.9}(u), \quad G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.7}(u), \quad G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.3}(u),$$

$$G^{\Phi}_{\mathrm{PSAD}\geq 0.8}(u), \quad G^{\Phi}_{\mathrm{PSAD}\geq 0.5}(u), \quad G^{\Phi}_{\mathrm{PSA}\geq 0.3}(u).$$

Define three decision conditions in the classical core:

$$\psi_{\mathrm{urgent}}(u) := G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.9}(u) \vee G^{\Phi}_{\mathrm{PSAD}\geq 0.8}(u),$$

$$\psi_{recommended}(u) := G\Phi_{PI-RADS \geq 0.7}(u) \vee \left(G\Phi_{PI-RADS \geq 0.3}(u) \wedge G\Phi_{PSAD \geq 0.5}(u)\right),$$

$$\psi_{optional}(u) := G\Phi_{PSA \geq 0.3}(u) \wedge \left(G\Phi_{FamilyRisk = 1}(u) \vee G\Phi_{Age-HR \geq 0.5}(u)\right).$$

Example 6.6 (Qualification for Active Surveillance (AS) (threshold-atom style)). Let u denote a patient. Assume the following atoms (illustratively):

$$G^{\Phi}_{\text{Gleason}\leq 6}(u), \quad G^{\Phi}_{\text{cT1-2a=1}}(u), \quad G^{\Phi}_{\text{PSA}<0.5}(u), \quad G^{\Phi}_{\text{CoresPos}\leq 0.33}(u).$$

Qualification condition in the core:

$$\psi_{\text{AS-qual}}(u) := G^{\Phi}_{\text{Gleason}\leq 6}(u) \wedge G^{\Phi}_{\text{cT1-2a=1}}(u) \wedge G^{\Phi}_{\text{PSA}<0.5}(u) \wedge G^{\Phi}_{\text{CoresPos}\leq 0.33}(u).$$

Optionally, a conservative exclusion:

$$\psi_{\text{noAS}}(u) := \neg\psi_{\text{AS-qual}}(u) \vee G^{\Phi}_{\text{Volume}>0.5}(u) \vee G^{P}_{\text{Compliance}<0.7}(u) \vee G^{\Phi}_{\text{LifeExpectancy}>10y=0}(u).$$

Example 6.7 (Follow-up: detecting biochemical recurrence (BCR) and triggering the next step). Let u denote a patient after definitive treatment. Introduce atoms:

$$G^{\Phi}_{\text{PSA}\geq 0.2}(u), \qquad G^{\Phi}_{\text{PSA-confirmed=1}}(u).$$

Define the BCR condition:

$$\psi_{\text{BCR}}(u) := G^{\Phi}_{\text{PSA}\geq 0.2}(u) \wedge G^{\Phi}_{\text{PSA-confirmed=1}}(u).$$

Optionally, a trigger for imaging:

$$\psi_{\text{TriggerImaging}}(u) := \psi_{\text{BCR}}(u) \wedge \left(G^{\Phi}_{\text{PSA}>0.2}(u) \vee G^{\Phi}_{\text{PSA-DT}<6m}(u) \right).$$

A follow-up procedure (orders, iterations, stop conditions) is naturally expressed in AGL programs, and the predicates above serve as tests $\varphi?$.

7. Examples: decision–action patterns (non-normative)

7.1. End-to-end hero example: from granules to a verifiable workflow

This example illustrates the full pipeline targeted by AGL: a computation layer produces graded evidence, the evidence is exposed to the verifiable core via a finite threshold signature, and a profiled rule core drives an actionable FO-PDL workflow. The example is schematic and non-normative.

Computation layer (granules). Assume three graded outputs for a patient u: (i) a biochemical suspicion score $\Phi_{\text{psa}}(u) \in [0,1]$, (ii) an imaging suspicion score $\Phi_{\text{img}}(u) \in [0,1]$, (iii) a fitness score $\Phi_{\text{fit}}(u) \in [0,1]$. Assume a finite, auditable threshold signature $R_{\text{psa}} = \{0.6, 0.8\}$, $R_{\text{img}} = \{0.6\}$, $R_{\text{fit}} = \{0.4\}$.

Auditable interface (threshold atoms). The computation layer emits only two-valued facts such as:

$$G^{\Phi_{\text{psa}}}_{\geq 0.8}(u), \qquad G^{\Phi_{\text{img}}}_{\geq 0.6}(u), \qquad G^{\Phi_{\text{fit}}}_{\leq 0.4}(u),$$

together with crisp observations (e.g., $\mathsf{BiopsyAvailable}(u)$).

Verifiable rule core (profiled to GF/RGF). Introduce two action-relevant predicates: $\mathsf{HighRisk}(u)$ and $\mathsf{EscalateToMDT}(u)$ (MDT = multidisciplinary team). A simple, guarded-style rule (written here in a classical implication form) can be:

$$\forall u \left(\mathsf{Patient}(u) \wedge G^{\Phi_{\mathrm{psa}}}_{\geq 0.8}(u) \wedge G^{\Phi_{\mathrm{img}}}_{\geq 0.6}(u) \rightarrow \mathsf{HighRisk}(u)\right),$$

and an escalation rule driven by risk and low fitness:

$$\forall u \left(\mathsf{Patient}(u) \wedge \mathsf{HighRisk}(u) \wedge G^{\Phi\mathrm{fit}}_{\leq 0.4}(u) \rightarrow \mathsf{EscalateToMDT}(u)\right).$$

The essential point is that the core reasons only over two-valued atoms, while all graded computation remains outside.

Action workflow (FO-PDL style). Let collect, compute, applyRules, requestEvidence, and escalate be primitive actions. A schematic workflow specification is:

$$\mathrm{collect};\ \mathrm{compute};\ \mathrm{applyRules};\ \left(\mathsf{EscalateToMDT}(u)?;\ \mathrm{escalate} \cup \neg\mathsf{EscalateToMDT}(u)?;\ \mathrm{requestEvidence}\right).$$

Since the tests are built from profiled predicates (including threshold atoms), the verification boundary remains explicit.

In this section we present six short examples illustrating typical decision–action patterns in AGL. Examples are demonstrative (non-normative). The canonical principle remains unchanged: numerical and probabilistic comparisons are encapsulated as threshold atoms, and the rule/procedure core operates on discrete predicates.

Audit trail generation. A deployment can log: (i) provenance identifiers for the computed granule values (or the values themselves where permitted), (ii) which threshold atoms from the finite signature R_Ψ were asserted for patient u, (iii) which profiled rules/tests fired in the verifiable core, and (iv) which workflow branch/action was executed (including escalation). For example, the log may state that u was escalated because a specific rule fired based on a threshold atom such as $G^{\Phi\mathrm{risk}}_{\geq 0.8}(u)$, together with the compiled fragment identifier and verification outcome.

7.2. Static patterns: threshold atoms as the rule interface

Example 7.1 (E1: Fuzzy thresholds as auditable atoms (PSA + PI-RADS, schematic))**.** Let u denote a patient. Assume two fuzzy granules $\Phi_{G_{\mathrm{PSA}}}(u)$ and $\Phi_{G_{\mathrm{PI\text{-}RADS}}}(u)$. The exposed interface:

$$G^{\Phi}_{\mathrm{PSA}\geq 0.5}(u), \qquad G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.8}(u).$$

Qualification condition in the classical core:

$$G^{\Phi}_{\mathrm{PSA}\geq 0.5}(u) \wedge G^{\Phi}_{\mathrm{PI\text{-}RADS}\geq 0.8}(u).$$

Example 7.2 (E2: A probabilistic threshold as a decision interface (LN+ risk, schematic))**.** Let $P_{G_{\mathrm{LN+}}}(u) \in [0,1]$ be a probabilistic granule estimating LN+ risk. For a policy threshold $r \in [0,1]$ define the atom:

$$G^{P}_{\mathrm{LN+}\geq r}(u).$$

Such an atom can then feed qualification rules or tests $\varphi?$ in procedures.

7.3. Procedural patterns: AGL programs (FO-PDL)

Example 7.3 (E3: Branching "verify, then act" (schematic))**.** Let $\mathsf{Patient}(u)$ be a type predicate. Consider an action a_{confirm}, after which predicate $\mathsf{MarkerConfirmed}(u)$ may become available in the KB-state. Let a_{treat} and a_{treatAlt} be alternative actions. A branching program:

$$\pi := \mathsf{Patient}(u)?;\ a_{\text{confirm}};\ (\mathsf{MarkerConfirmed}(u)?;\ a_{\text{treat}}\ \cup\ \neg\mathsf{MarkerConfirmed}(u)?;\ a_{\text{treatAlt}}).$$

Interpretation. The program makes the decision point auditable by separating verification tests from actions. In particular, the system can record which test succeeded ($\mathsf{MarkerConfirmed}(u)?$ or $\neg\mathsf{MarkerConfirmed}(u)?$), and therefore which branch enabled a_{treat} versus a_{treatAlt}.

Example 7.4 (E4: Follow-up loop with a stop condition (iteration))**.** Let a_{follow} be a follow-up action and let $\mathsf{Stop}(u)$ be a stop condition expressed in the verifiable profile. A loop pattern:

$$\pi := (\neg\mathsf{Stop}(u))?;\ a_{\text{follow}} \quad \text{and} \quad \pi^*.$$

Example 7.5 (E5: Order-insensitive steps via safe interleaving)**.** If two steps α and β are order-insensitive, model interleaving:

$$(\alpha;\beta)\ \cup\ (\beta;\alpha).$$

This is a substitute for $\|$ in the base profile.

Interpretation. Instead of introducing true parallel composition in the base profile, we encode permissible interleavings explicitly. This keeps verification within standard program constructs while still capturing the intended independence of steps.

7.4. Complex decision rules with quantifiers (MT-FOGL + GF/RGF style)

Example 7.6 (A high-specialization center policy (schematic, auditable))**.** This example emphasizes that the AGL core is (profiled) first-order predicate logic (GF/RGF), while complex risk criteria are supplied via threshold atoms.

(1) All high-risk patients must be discussed at an MDT. Let $G_{\text{high-risk}}$ be a granule (e.g., fuzzy) describing "high risk," and let $G^{\Phi}_{\text{high-risk}\geq 0.8}(u)$ be its threshold atom in the core. Let $\mathsf{MDT}(u)$ denote the fact "patient u's case is referred/discussed at an MDT." In the classical core:

$$\forall u\ \left(\mathsf{Patient}(u) \wedge G^{\Phi}_{\text{high-risk}\geq 0.8}(u)\ \rightarrow\ \mathsf{MDT}(u)\right).$$

(2) If the patient has contraindications to all options, consider best supportive care. Let $\mathsf{FitRP}(u)$, $\mathsf{FitRT}(u)$, and $\mathsf{FitADT}(u)$ be schematic predicates expressing clinical eligibility for three treatment options: *radical prostatectomy* (RP), *radiotherapy* (RT), and *androgen deprivation therapy* (ADT), respectively. (These predicates are used here only to illustrate how domain predicates enter the verifiable core; they are *non-normative* and require clinical definition and governance in any real deployment.) Let $\mathsf{BSC}(u)$ mean "consider best supportive care." Then:

$$\forall u \left(\mathsf{Patient}(u) \wedge \neg\mathsf{FitRP}(u) \wedge \neg\mathsf{FitRT}(u) \wedge \neg\mathsf{FitADT}(u) \rightarrow \mathsf{BSC}(u)\right).$$

(3) There exist patients for whom AS is safer than treatment (canonical variant). Let $G_{\text{very-low-risk}}$ be a granule (crisp or fuzzy) describing "very low risk" in the sense of eligibility for a conservative strategy. In the auditable profile assume its interface as:

$$G^{\Phi}_{\text{very-low-risk}=1}(u).$$

Moreover, the risk comparison (e.g., "AS vs treatment") is computed outside the core as a probabilistic granule $P_{G_{\text{ASsaferThanTx}}}(u) \in [0,1]$ and exposed only via a threshold atom $G^{P}_{\text{ASsaferThanTx}\geq r}(u)$ for a chosen policy threshold $r \in [0,1]$. Then:

$$\exists u\, \mathsf{Patient}(u) \wedge G^{\Phi}_{\text{very-low-risk}=1}(u) \wedge G^{P}_{\text{ASsaferThanTx}\geq r}(u).$$

Remark 7.7 (Clinical interpretation and compliance with the *Decidability Split*). The formulas above show that AGL keeps the decision core as first-order predicate logic (profiled to GF/RGF), supporting auditability and verifiability. Vagueness and probability are modeled in the granule layer; the core sees only Boolean predicates (threshold atoms or crisp atoms) that serve as the interface to rules and program tests.

8. Related work

The choice of GF as a verifiability profile builds on classical results on guarded fragments of first-order logic, including decidability and the finite model property [1, 10]. The procedural layer of AGL relies on ideas from dynamic logics (PDL) and regular-program syntax [11], which enables a formal description of sequencing, choice, and iteration of actions.

GF/RGF as a mechanism for controlling verifiability. Full first-order dynamic logic is often undecidable; AGL therefore imposes engineering restrictions: (1) program tests $\varphi?$ are restricted to the GF/RGF profile, (2) arithmetic, probability, and vagueness are exposed to the core solely via threshold atoms, which stabilizes semantics and keeps the verification interface within two-valued logic.

Probabilistic reasoning, inductive logic, and epistemic background. Threshold atoms in AGL deliberately decouple the verifiable two-valued core from graded evidence (probability / fuzziness), which connects naturally to classical probabilistic KR and inductive-logical perspectives on evidential support (Bayesian networks and probabilistic inference in particular) [7, 14, 18]. Moreover, since AGL is intended for workflow-like decision–action systems deployed in multi-agent settings (human experts, services, and automated components), epistemic notions and reasoning about knowledge form an important meta-theoretic backdrop [9].

RGF and path-based data anchoring. The regularization of guards [3] is motivated by the need to

express data-link patterns in quantifier guards and tests (e.g., patient–record–sample relations) that are typical for workflow systems.

Granular computing as a computation layer and an auditable interface. The concept of *Information Granules* and granular views of vagueness and uncertainty have deep roots in the literature on granular computing and information granulation [2, 19–21], while rough sets [13] provide a natural model of concept boundaries driven by observability. In AGL these mechanisms are deliberately separated from the verifiable core (the Decidability Split).

Meta-theoretic background: non-classical logics and algebraic approaches. The classical monographs of Rasiowa and Sikorski [16] and Rasiowa [15] provide reference points for a broad spectrum of non-classical logics. In this paper, however, such logics are not the core: instead, we use a two-valued verification interface and move "gradedness" into the granule layer.

9. Discussion: expressivity vs. verifiability

The richer the description language, the harder it becomes to achieve algorithmic verification and complexity control. AGL adopts a profiled approach: we allow expressivity in the surface layer (e.g., an expert-facing DSL), provided that it has an explicit mapping to a controlled verifiability interface. The verifiable core is restricted to the GF/RGF profile and to auditable predicates (threshold atoms).

In practice, we realize this via two mechanisms: (i) the *Decidability Split*—all arithmetic and probability are encapsulated in *Information Granules* and exposed as threshold atoms, (ii) restricting program tests $\varphi?$ to GF/RGF, which stabilizes the properties of the verifiable core. The architectural rationale is illustrated in Fig. 1.

Parallelism as an illustration of the trade-off. Extending the program language with $\|$ increases expressivity (concurrent action tracks), but typically worsens verification properties (complexity and, in many variants, decidability). Therefore, in the base profile we recommend avoiding $\|$ and using safe substitutes: interleaving $(\alpha;\beta) \cup (\beta;\alpha)$ when order does not matter, or treating a bundle of actions as a single atomic action.

Figure 2 clarifies where the decidability boundary lies in AGL and how the granule layer acts as an auditable interface between computations and the verifiable core.

10. Engineering applications of AGL in AI/IT

The AGL logic presented in this paper has broad potential for engineering applications in AI/IT, because it is designed as an *auditable control and verification layer* for decision–action systems.

The OnkoBot project is a year-long collaboration between clinical experts at NIO-PIB and engineering and research teams at UWM, conducted under a formal Letter of Intent and consolidated in a comprehensive internal project charter document for the OnkoBot program [5]. During this period, the team iteratively advanced proof-of-concept prototypes across multiple platform subsystems and formalized successive milestones via mutually agreed project charters.

From a deployment viewpoint, it is crucial that AGL does not compete with mature technologies based on fragments of FOL (Datalog, DL/OWL, SAT/SMT), but complements them with *actionable granules*: threshold-based, auditable interfaces between rich input information and a classical, verifiable rule core.

We distinguish two complementary application directions:

(1) *Enriching existing FOL-fragment applications in AI/IT with AGL mechanisms* (guardrails, audit, verifiability profiling, decision correction and escalation).

(2) *A layered view of non-classical logics, including modal and temporal logics:* temporal, deontic, and "possibility/necessity" properties are captured procedurally in FO-PDL (workflow), without introducing many-valued truth into the fact core.

10.1. Examples of potential AGL applications in AI/IT around key FOL fragments

From the perspective of *engineering practice in AI/IT*, the most important FOL fragments are those that combine: *(a)* natural modeling (rules, classes, relations), *(b)* decidability and controlled complexity, *(c)* mature implementations (Datalog/ASP engines, DL reasoners, SAT/SMT solvers).

In data technologies, CQ/UCQ/EPFO and Datalog (with recursion) dominate, while knowledge systems rely on DL/OWL. When inference must generate new facts under constraints, the family TGDs/Datalog± appears. When exceptions and defaults are essential, NAF and ASP enter practice. For controlled negation and locality in relations, GNFO and the guardedness principle (GF) are particularly relevant. Finally, when systems touch arithmetic and data structures at the level of code and constraints, SAT/SMT provides the computational backbone for verification and analysis.

Table 2 summarizes these fragments in a practical view: what is restricted, what is gained, and where they dominate in AI/IT. *Note:* FO-PDL (procedural modal logic) is discussed in this paper as the procedural layer of AGL in Subsection 1; the table below concerns FOL fragments dominating in KR/DB/ATP.

Engineering note: complexity regimes (data vs. combined). In Table 2 we list inference complexity in a compact form because AI/IT practice commonly uses two standard regimes:

- *Data complexity*: treat the query/rules/ontology (TBox) as fixed and vary only the size of input data (ABox, database instance). This is the dominant perspective in database systems (e.g., for CQ) and analytics pipelines.
- *Combined complexity*: treat both data and the problem description (query, rules, TBox) as variable. This typically yields higher bounds but better reflects costs when logic/ontologies/rules are generated or frequently modified (e.g., for Description Logics and guarded logics).

Hence the complexity classes in Table 2 should be interpreted in the context of the dominant use cases of each fragment.

10.2. Potential AGL applications related to non-classical predicate logics

For many years, various areas of potential applications of non-classical logics in AI/IT have been intensively studied, in particular:

- many-valued logics (e.g., Łukasiewicz, Post) — allowing more than two truth/assessment values,
- modal logics — enabling necessity and possibility,
- intermediate logics — modeling intermediate states and varying degrees of non-constructiveness,
- paraconsistent logics — tolerating inconsistencies without trivialization,
- temporal logics — describing time-dependent properties (e.g., disease progression),

- deontic logics — describing obligations and prohibitions (norms, procedures, policies).

In decision–action systems, these classes matter because data and domain knowledge may be: (i) vague/graded, (ii) stochastically uncertain, (iii) incomplete or inconsistent, and (iv) embedded in time and, crucially, in procedures. Formally, one can also generalize MT-FOGL constructions to predicate calculi with algebraic semantics, opening broad perspectives for integration with non-classical logics. Interested readers are referred to the classical monographs by H. Rasiowa and R. Sikorski [16] and H. Rasiowa [15].

In this paper, however, we adopt a classical core because it offers the most transparent engineering compromise between expressivity and verifiability. In particular, a classical approach provides:

(1) mature meta-theoretic foundations (including decidability results for selected profiles),

(2) broad availability of automated reasoning and verification tools,

(3) a simple, auditable interpretation in the rule/procedure core,

(4) straightforward mapping to existing decision-support architectures.

Crucially, the key distinction that drives the AGL architecture is: *uncertainty, gradedness, and probabilistic aspects are not encoded as degrees of truth in the logical core*. Instead, they are encapsulated at the *Information Granules* level and exposed to the core only via *threshold atoms*. This keeps the rule/procedure core within classical two-valued first-order logic, profiled to GF/RGF, simplifying audit, stabilizing semantics, and enabling algorithmic verification while retaining rich inputs (the *Decidability Split*). Temporal and deontic properties are handled as workflow properties in FO-PDL, without introducing many-valued truth into the fact core.

11. Planned next steps for AGL (Next steps)

11.1. The key value of AGL in mission-critical *systems*

We assume that AGL provides a formal core for constructing *auditable, interpretable, and verifiable* decision mechanisms under *vagueness and uncertainty*. In *mission-critical* contexts (clinical decision support, agent control, autonomous systems), the key is not prediction alone but the ability to *document* and *enforce* procedural constraints: from inputs, through granulation, to a verifiable verdict. Below we indicate priority, directly measurable next steps.

11.2. Auditable procedural knowledge models as safety boundaries

- *Lead goal.* Develop a methodology for building *symbolic, auditable models of procedural knowledge* (e.g., action rules, dynamic constraints, eligibility conditions) that can serve as a *safety boundary* (*safety policy / guardrail*) for decision systems in complex operational environments.

- *Role of AGL.* Use native granulation mechanisms to construct *procedural granules* (conditions–actions–constraints) with a full *audit trail*: (i) input conditions, (ii) granule assignment, (iii) rules/procedures fired, (iv) a formal justification of the verdict, and (v) identification of critical rules and boundary points (vagueness thresholds).

- *Benchmark and metrics.* Validate on domains requiring strict procedural compliance, in particular: (a) NSCLC scenarios (e.g., histopathology-driven diagnostics, diagnostic–therapeutic pathways as a *narrative benchmark* independent of guideline versions), and (b) simulations of autonomous agents/systems. Evaluate using measurable indicators: *coverage* (fraction of cases for which AGL produces a procedural verdict), *conflict rate* (number of detected rule conflicts and their classification), *audit completeness* (ability to reconstruct the full decision path), and *explainability fidelity* (agreement between explanations and the formal reasoning trace).

11.3. Online verification of LLM agents: symbolic control, correction, and escalation

- *Application goal.* Integrate AGL as a *verifiable control mechanism* (*verifier*) for LLM-based agents operating in *mission-critical* environments. The agent proposes an action (or recommendation), while AGL enforces procedural safety constraints.

- *Role of AGL in online control.* AGL maps imprecise and uncertain agent outputs (e.g., proposed actions, textual rationales, plan parameters) to *linguistic/procedural granules*, then verifies them in real time against formal procedural rules (see Subsection 2). The verification outcome is always auditable: it includes the set of active rules, their satisfaction conditions, and a minimal formal justification.

- *Reaction policy.* Instead of binary accept/reject, consider three modes: (i) allow (decision admissible), (ii) revise (AGL computes a correction to the nearest formally admissible decision), (iii) abstain & escalate (hold and escalate to an expert). This is particularly important in clinical and control settings where borderline decisions require controlled correction or escalation, not only hard rejection.

- *Boundary verification (uncertainty band).* Prioritize cases in which an LLM agent generates decisions within an *uncertainty band* (i.e., near admissibility boundaries described by granules and thresholds). The goal is a mechanism that: (a) identifies the nature of borderline status, (b) generates an auditable correction (*revise*) or a formal justification for escalation, and (c) minimizes unnecessary rejections while maintaining safety.

- *Benchmark and metrics.* Measure: *allow/revise/escalate rates*, correction effectiveness (fraction of inadmissible cases becoming admissible after *revise*), *time-to-verdict* (online cost), and *audit latency* (time to produce the audit artifact).

11.4. Human-in-the-Loop (HITL): governance, triggers, and evidence artifacts

In AGL, *Human-in-the-Loop (HITL)* is not an informal "manual override", but a controlled governance layer that is explicitly triggered by verifiable signals produced by the GF/RGF-bounded verification core. Operationally, HITL is the mechanism that closes the safety loop in *mission-critical* settings: it routes borderline, inconsistent, incomplete, or out-of-profile situations to qualified experts and records a traceable justification for the final operational decision (allow / revise / abstain&escalate).

Design principle (Decidability Split + finite evidence). The verifiable core never consumes raw clinical data or high-dimensional model internals. Instead, it consumes only *auditable finite evidence* (threshold atoms, bounded retrieved sets, and bounded proof objects). HITL uses this evidence to (i) validate/approve actions, (ii) request additional evidence/provenance, or (iii) revise policy parameters (e.g., threshold signatures) under explicit authorization and version control.

A concise proposal of HITL triggers and corresponding auditable artifacts (the operational governance interface for AGL) is summarized in Table 3. It makes the escalation mechanism explicit: the core outputs verifiable *reasons* for escalation, while HITL produces a governance-grade *decision record* linked to finite evidence. This supports traceability, accountability, and the controlled evolution of thresholds, granules, and procedures.

11.5. Towards Interactive Granular Computing (IGrC)

- *Main Remark* A key engineering signal is the discrepancy between *expected* and *observed* action outcomes: the rule/procedure core may predict that a certain action should achieve a target state, while the operational environment yields a different observation. IGrC provides a vocabulary for handling such mismatches through *physical semantics* (linking symbols to measurable outcomes) and through *c-granules* (complex granules) that encapsulate richer, composite evidence structures. Practically, this supports adaptive responses such as re-granulation, threshold recalibration, or revising the finite threshold signature R_{Ψ} under explicit governance.

- *Vision.* Extend AGL toward *Interactive Granular Computing (IGrC)* [12, 17], enabling controlled, auditable interaction of experts and environments with granulation processes and reasoning, in extended AGL variants aligned with IGrC.

- *What is interactive (without losing auditability).* Interaction is not ad-hoc "manual rule editing," but formally described operations on AGL artifacts: (i) updating granules (e.g., redefining linguistic concepts), (ii) tuning thresholds/preference relations, (iii) prioritizing rules and resolving conflicts, (iv) introducing fixes via versioned *change requests*. Every modification generates an audit trail (who, what, when, why, with what effect).

- *Benefits for agent control.* IGrC enables real-time feedback: experts can adjust *procedural granules* and verification parameters in a controlled way, increasing adaptability of hybrid agents and facilitating updates of symbolic safety boundaries as operational conditions change (e.g., procedure updates, new evidence, data drift).

- *Metrics.* Evaluate: time and number of iterations required to stabilize policy, impact on *allow/revise/escalate rates*, and consistency measures for versioned rules (e.g., conflict reduction across versions and decreased escalation without safety loss).

12. Conclusions and future work

In this paper, we have introduced AGL (Actionable Granular Logic), a formal framework designed to bridge the gap between stochastic AI agents and the rigorous requirements of clinical decision-making. By implementing a *Decidability Split* and utilizing the guarded profiles (GF/RGF), we have created a symbolic wrapper capable of grounding Large Language Model outputs in verifiable procedural structures.

The unique positioning of AGL compared to existing knowledge representation frameworks is summarized in Table 4. Unlike traditional logic fragments, AGL is specifically designed to handle the "hallucination" risks of generative AI by acting as a formal verifier that ensures procedural compliance.

The practical utility of AGL is already being explored within the OnkoBot prototype, developed in collaboration with NIO-PIB. Preliminary findings suggest that AGL does not merely act as a safety filter, but as a sophisticated clinical advisor. For instance, it can suggest missing diagnostic prerequisites when an LLM proposes a treatment plan prematurely.

Locality and Computational Budgeting. To address the theoretical complexity of the RGF profile, our implementation relies on the principle of data locality enforced by guards. Furthermore, we employ a "computational budgeting" strategy: if a verification task exceeds predefined time limits, the system triggers a mandatory HITL escalation. This ensures that safety is never compromised by hardware limitations.

Long-term outlook: solver progress and specialized computing. While AGL is defined independently of hardware, advances in solver technology and specialized computing (including emerging quantum-computing paradigms) may expand the practical verification budget for richer granular interfaces. We therefore treat hardware acceleration as a research direction rather than an assumption for near-term deployments.

IGrC outlook. Future work will leverage Interactive Granular Computing (IGrC) to manage discrepancies between expected and observed action results through adaptive re-granulation and controlled updates of threshold signatures, strengthening the link between formal decisions and their physical semantics in deployments.

13. Acknowledgements

The author thanks colleagues and clinical collaborators at NIO-PIB, UWM, and PAN for discussions that helped clarify the engineering goals of auditable and verifiable decision–action systems. Any remaining errors are the author's responsibility.

14. References

References

[1] Andréka, H., Németi, I., van Benthem, J. 1998. Modal Languages and Bounded Fragments of Predicate Logic. Journal of Philosophical Logic, 27(3), 217–274.

[2] Bargiela, A., Pedrycz, W. 2003. Granular Computing: An Introduction. Kluwer Academic Publishers.

[3] Bednarczyk, B., Kieroński, E. 2025. Guarded Fragments Meet Dynamic Logic: The Story of Regular Guards. In Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR 2025), 89–99.

[4] Briganti, A., et al. 2019. A novel tool to predict lymph node metastases in prostate cancer: the 2018 Briganti nomogram. European Urology Oncology, 2(4), 420–426.

[5] Dąbkowski, M., Wawrzuta, D., Żarłok, E., Jankowski, A., Polkowski, L., Skowron, A., Artiemjew, P. 2025. *OnkoBot: Propozycja Karty Projektu. Projekt Zintegrowanego Systemu AI dla Narodowego Instytutu Onkologii PIB*. Internal project document (NIO-PIB and UWM), Warsaw/Olsztyn, version dated 5 October 2025. Available upon request from the authors (internal circulation).

[6] European Association of Urology (EAU). 2025. EAU Guidelines. 2025 edition (presented at the EAU Annual Congress, Madrid 2025), available via Uroweb.

[7] Eagle, A. 2025. Probability and Inductive Logic. Cambridge University Press.

[8] European Society for Medical Oncology (ESMO). 2025. ESMO Clinical Practice Guidelines: Lung and Chest Tumours. Online guideline collection (accessed 2025).

[9] Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y. 1995. Reasoning About Knowledge. MIT Press.

[10] Grädel, E. 1999. On the Restraining Power of Guards. The Journal of Symbolic Logic, 64(4), 1719–1742.

[11] Harel, D., Kozen, D., Tiuryn, J. 2000. Dynamic Logic. MIT Press.

[12] Jankowski, A. 2017. Interactive Granular Computations in Networks and Systems Engineering: A Practical Perspective. Springer.

[13] Pawlak, Z. 1982. Rough Sets. International Journal of Computer & Information Sciences, 11, 341–356.

[14] Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann.

[15] Rasiowa, H. 1974. An Algebraic Approach to Non-Classical Logics. North-Holland Publishing Company, Amsterdam.

[16] Rasiowa, H., Sikorski, R. 1963. The Mathematics of Metamathematics. Polish Scientific Publishers (PWN), Warsaw.

[17] Skowron, A., Jankowski, A., Dutta, S. 2025. Interactive Granular Computing: Toward Computing Model for Complex Intelligent Systems. In Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS), Bolanowski, M., Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D. (Eds.).

[18] Williamson, J. 2017. Lectures on Inductive Logic. Oxford University Press.

[19] Yao, Y.Y. 2004. Granular Computing: Basic Issues and Possible Solutions. In Proceedings of the 5th Joint Conference on Information Sciences.

[20] Zadeh, L.A. 1965. Fuzzy Sets. Information and Control, 8(3), 338–353.

[21] Zadeh, L.A. 1997. Toward a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic. Fuzzy Sets and Systems, 90(2), 111–127.

15. Annex A. Acronyms and abbreviations

This annex summarizes acronyms and abbreviations used in the paper, with emphasis on clinical/medical and IT/AI terms.

Table 1: Acronyms and abbreviations used in the paper.

Acronym	Meaning	Notes / context in this paper
ADT	Androgen Deprivation Therapy	Prostate cancer therapy abbreviation (medical).
AGL	Actionable Granular Logic	Proposed framework for verifiable specifications and reasoning in decision–action systems.
AI	Artificial Intelligence	Umbrella term for learning and reasoning components.
AKB	Algebraic Knowledge Base	Rule/knowledge-base notion referenced as a verifiable core style.
API	Application Programming Interface	System integration interface (services, modules, subsystems).
AS	Active Surveillance	Prostate cancer management strategy (medical).
ASP	Answer Set Programming	Logic programming / nonmonotonic KR paradigm (KR/ATP context).
ATP	Automated Theorem Proving	Proving in first-order and related logics; solver/ATP context.
CQ	Conjunctive Query	Database/KR query form; central in DL/ontology and DB theory.
CSP	Constraint Satisfaction Problem	Constraint-based reasoning/optimization; solver context.
CT	Computed Tomography	Imaging modality referenced in illustrative examples.
DB	Database	Data management context for KR/queries; appears in KR/DB/ATP comparisons.
DL	Description Logic	Family of decidable FOL fragments used in ontologies; OWL foundations.
DSL	Domain-Specific Language	Expert-facing specification layer that compiles to a verifiable core.
EAU	European Association of Urology	Maintains regularly updated European urology guidelines (e.g., prostate cancer).
EHR	Electronic Health Record	Clinical data source (raw data layer).
ESMO	European Society for Medical Oncology	Maintains regularly updated European oncology clinical practice guidelines.
EU AI Act	European Union Artificial Intelligence Act	Governance context for high-risk AI systems (if referenced).
FO-PDL	First-Order Propositional Dynamic Logic	Procedural layer with first-order state predicates; tests are restricted to GF/RGF.
FOL	First-Order Logic	Classical predicate logic; the verifiable core is constrained by decidable fragments.
GDPR	General Data Protection Regulation	EU data protection regulation (privacy/security context).
GF	Guarded Fragment	Decidable fragment of FOL used as a verifiability profile.
GNS-Align	Grounded Neuro-Symbolic Alignment	Risk-governed architecture notion referenced as a broader alignment/governance perspective.
HITL	Human-in-the-Loop	Workflow pattern where human experts

review/approve/escalate decisions.

continued on next page

Acronym	Meaning	Notes / context in this paper
ILP	Inductive Logic Programming	Learning logical rules from examples; ML/KR bridge.
KB	Knowledge Base	Repository of formalized knowledge (rules, facts, ontologies).
KR	Knowledge Representation	General area: logics, ontologies, rule bases used to represent domain knowledge.
LIS	Laboratory Information System	Laboratory data source (raw data layer).
LKB	Lattice Knowledge Base	Lattice-valued knowledge representation notion referenced for graded evidence modeling.
LLM	Large Language Model	Class of generative models that may require verifiable wrappers in mission-critical settings.
MDT	Multidisciplinary Team	Clinical decision forum (workflow escalation / review context).
ML	Machine Learning	Models in the computation layer (outside the verifiable core).
MRI	Magnetic Resonance Imaging	Imaging modality referenced in illustrative examples.
MT-FOGL	Multi-Typed First-Order Granular Logic	Conceptual computation layer for soft typing and graded (fuzzy/probabilistic) granules.
NIO-PIB	National Oncology Institute, Poland (Państwowy Instytut Badawczy)	Clinical partner organization referenced in the OnkoBot context.
OnkoBot	OnkoBot program/project	Collaborative clinical+engineering program referenced as a motivating deployment context.
OS	Overall Survival	Standard oncology endpoint (medical).
PACS	Picture Archiving and Communication System	Imaging archive/source (raw data layer).
PAN	Polish Academy of Sciences	The national academy of sciences in Poland, involved in high-level research in logic and computer science.
PDL	Propositional Dynamic Logic	Modal/procedural logic for reasoning about programs (actions).
PET	Positron Emission Tomography	Imaging modality (medical).
PET-CT	PET combined with CT	Combined imaging modality (medical).
PFS	Progression-Free Survival	Standard oncology endpoint (medical).
PoC	Proof of Concept	Prototype implementations validating feasibility of subsystems/ideas.
PSA	Prostate-Specific Antigen	Example laboratory variable used to illustrate granules and threshold atoms.
PSAD	PSA Density	PSA normalized by prostate volume; used as an example feature/granule.
QALY	Quality-Adjusted Life Year	Health economics/clinical outcome metric (general medical abbreviation).
QoL	Quality of Life	Clinical outcome category (medical).
RAG	Retrieval-Augmented Generation	Architecture pattern for grounding generation in retrieved sources (if referenced).
RDF	Resource Description Framework	Graph-based KR data model; Semantic Web context.
RGF	Regular Guarded Fragment	Guarded fragment with regular/path-shaped guards for data anchoring.
RP	Radical Prostatectomy	Surgical treatment abbreviation (medical).

continued on next page

Acronym	Meaning	Notes / context in this paper
RT	Radiotherapy	Common oncology treatment abbreviation (medical).
FitRP	Fit for Radical Prostatectomy	Schematic domain predicate used in Example 7.6: patient eligible for RP (non-normative).
FitRT	Fit for Radiotherapy	Schematic domain predicate used in Example 7.6: patient eligible for RT (non-normative).
FitADT	Fit for Androgen Deprivation Therapy	Schematic domain predicate used in Example 7.6: patient eligible for ADT (non-normative).
SAT	Boolean Satisfiability	Solver setting for verification/decision procedures (KR/ATP context).
SMT	Satisfiability Modulo Theories	Solver-based reasoning over background theories.
SPARQL	SPARQL Protocol and RDF Query Language	Query language for RDF graphs; KR/DB context.
SQL	Structured Query Language	Relational database query language; DB context.
UCQ	Union of Conjunctive Queries	DB/KR query form; used in ontology-mediated querying.
UWM	University of Warmia and Mazury in Olsztyn	Research/engineering partner organization referenced in the OnkoBot context.
XAI	Explainable AI	Transparency/interpretability methods; complements auditability and verification.

16. Figures and Tables

Table 2: Extended comparison of FOL fragments relevant in AI/IT (practical view; KR/DB/ATP context).

Fragment / family	Restriction / idea	Typical AI/IT applications + inference complexity
Horn logic	At most one positive literal per clause; if–then rules	Logic programming, expert/business rules, fact-based inference. *Complexity:* propositional Horn inference is *P-complete*; first-order variants depend on further restrictions but are typically well supported in practice.
Datalog	No function symbols; fixpoint over finite data	Recursive queries, graph analytics, program analysis, access-control policies. *Complexity: data complexity PTIME*; *combined complexity* typically high (classically *EXPTIME-complete*).
NAF / Datalog with negation	Negation as failure; stratified / well-founded / stable semantics	Rules with exceptions, defaults, exception-aware policies; analytics with exceptions. *Complexity:* depends on the class; for *stratified* programs often *PTIME (data)*; full stable semantics moves toward ASP.

Continued on the next page

Table 2: Continued

Fragment / family	Restriction / idea	Typical AI/IT applications + inference complexity
ASP	Rules + minimality/non-monotonicity; solutions as answer sets	Planning, scheduling, configuration, diagnosis, combinatorial tasks. *Complexity:* without disjunction typically *NP-complete* (answer-set existence); with disjunction usually higher in the polynomial hierarchy.
DL / OWL	Class/role constructors; TBox/ABox; decidable dialects	Ontologies, semantic web, classification, consistency, semantic search. *Complexity:* dialect-dependent; typically *EXPTIME* and above (for expressive dialects even *N2EXPTIME*).
CQ (conjunctive queries)	∃ + ∧ over atoms; no ¬, no ∨	SQL core (WHERE), OMQ, query optimization, data mappings. *Complexity:* evaluation (data) in *PTIME*; *combined* typically *NP-complete*; CQ containment *NP-complete*.
UCQ (union of CQs)	Disjunction of multiple CQs (UNION)	Data integration, rewriting, OMQ, views and mediators. *Complexity:* as for CQ; evaluation (data) in *PTIME*; UCQ containment typically *NP-complete*.
EPFO (existential-positive FO)	∃, ∧, ∨ without ¬, ∀	Positive querying, data transformations, rewriting to positive normal forms. *Complexity:* close to CQ/UCQ in practice: *data PTIME*, *combined* grows (often around *NP* for typical classes).
TGDs / Datalog±	Rules with ∃ in the head; chase; decidable classes	QA under constraints, data integration, data exchange, data completion, inference rules. *Complexity:* can be *undecidable* in general; for major decidable classes typically very high (often *EXPTIME–2EXPTIME*).
GF (Guarded Fragment)	Quantification only under a guard (variable locality)	Design of decidable "data+rules" formalisms; query theory over relational structures. *Complexity:* GF satisfiability is classically *2EXPTIME-complete*.
GNFO (Guarded Negation FO)	Negation allowed only in guarded contexts	Queries/rules with controlled negation, data validation, exception-aware policies. *Complexity:* satisfiability typically *2EXPTIME-complete* (as a reference point).
FO^2 (two-variable FO)	Only two variable names; reuse via quantification	Complexity control for binary relations; KB/DB patterns; links to DL. *Complexity:* FO^2 satisfiability (with equality) is classically *NEXPTIME-complete*.
Bernays–Schönfinkel / EPR	Prenex ∃*∀* without function symbols	Automated theorem proving (ATP), constraint solving in "almost propositional" form, a target for grounding + SAT/SMT pipelines. *Complexity:* *decidable*; satisfiability is classically *NEXPTIME-complete*.
Function-free FOL	No functions (often without =); facts and relations	Relational/graph data, RDF as facts, rule-based querying over databases. *Complexity:* absence of functions alone does not guarantee decidability; complexity depends on the selected fragment (e.g., EPR, GF, FO^2, CQ/UCQ).
Propositional logic (SAT)	No quantifiers; structureless atoms	Constraint solving, verification, constraint compilation, foundation of SMT. *Complexity:* SAT is *NP-complete*.
SMT (SAT + FOL theories)	Theories: EUF, LIA/LRA, arrays, bitvectors, datatypes, etc.	Formal verification, program analysis, symbolic execution, constraint synthesis. *Complexity:* theory-dependent; often *NP–EXPTIME* in theory, yet highly efficient in practice.

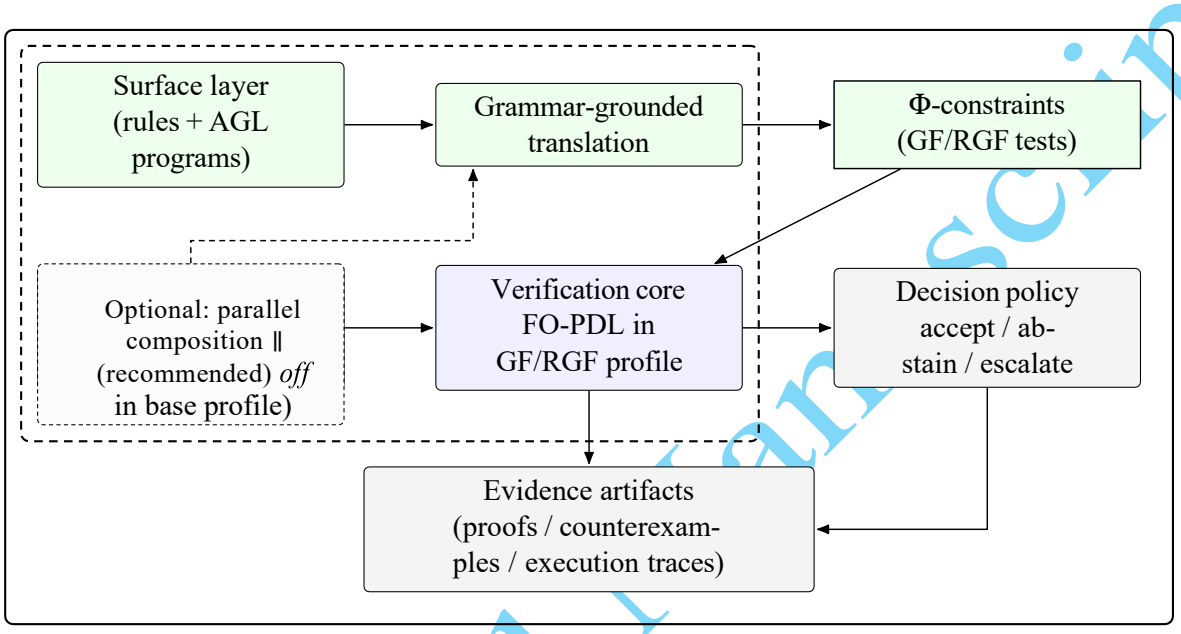

Figure 1: An example of AGL architecture as a compilation–verification pipeline (Decidability Split). Surface rules and AGL programs are translated into a GF/RGF-bounded FO-PDL verification core. Graded premises (numerical, probabilistic, vague) are computed outside the core as *Information Granules* and enter the core only via Boolean *threshold atoms*. Verification results drive the operational decision policy (accept/abstain/escalate) and yield auditable evidence artifacts (proofs, counterexamples, execution traces). For a step-by-step reading guide, see Section 1.

Trigger (when to escalate)	Core-level signal (auditable)	HITL action / role	Evidence artifact recorded
Borderline admissibility ("uncertainty band")	Verification outcome is "near boundary" (e.g., a threshold atom is within a policy-defined margin; or minimal counterexample exists)	Approve *allow* / request *revise* / confirm *escalate* policy for this band	Proof/counterexample trace + boundary explanation + chosen operational mode
Conflicting rules / inconsistent evidence	Detected conflict (incompatible obligations, mutually exclusive actions, or inconsistent preconditions)	Classify conflict type; select priority/resolution rule; request additional evidence if needed	Conflict certificate + resolution decision + justification note
Missing provenance / incomplete evidence	Required predicate cannot be evaluated as finite evidence (e.g., missing retrieval justification; missing data quality flags)	Request additional provenance; defer decision; initiate data-quality workflow	Missing-evidence report + provenance request ticket
Out-of-profile formula / undecidability risk	Compiled verification condition exceeds declared GF/RGF profile or violates the safe compilation constraints	Reject the artifact for production use; request re-compilation into the accepted profile	Profile-violation report + compilation log + accepted/rejected fragment tag
High-impact action class	Action belongs to a policy-defined high-severity class (e.g., irreversible or legally sensitive action)	Mandatory expert sign-off (dual control if required)	Sign-off record + role/identity + time stamp + linked evidence
Expected vs. observed mismatch (closed-loop control)	Observed outcome violates the expected post-action granule beyond tolerance (IGrC-style signal)	Trigger adaptation: re-granulation, threshold recalibration, or workflow revision under governance	Drift/mismatch report + adaptation decision + versioned change request

Table 3: HITL triggers and auditable artifacts (operational governance interface for AGL).

Table 4: Comparison of AGL with selected reasoning frameworks (focus: clinical deployment).

Framework	**Output verifiability / grounding**	**Support for graded clinical notions**	**Decision–action integration**	**Typical failure mode / limitation**
Pure LLM	Low: no native proof/trace; grounding depends on prompting and post-hoc checks	High (linguistic coverage), but weak handling of explicit thresholds, uncertainty and audit constraints	Low: generates text; execution/policy requires external orchestration	Non-grounded confident outputs; prompt injection / data contamination; non-auditable reasoning chain
OWL / DL	Medium–High: model-theoretic semantics; consistency checking; limited to the modeled vocabulary/axioms	Low: primarily crisp concepts; graded notions require extensions or external layers	Low: descriptive inference; action/workflow typically external	Coverage gaps and modeling brittleness; high engineering cost; limited support for procedural workflows
Datalog / ASP	High: explicit rules; derivations can be traced; deterministic (or stable-model) semantics	Low–Medium: mostly crisp; uncer-tainty/vagueness requires additional formalisms (e.g., weights, probabilities, fuzzy layers)	Medium: supports consequences and constraints; workflow still usually external	Boolean rigidity; knowledge acquisition bottleneck; scala-bility/maintenance issues for large rule sets
AGL (Pro-posed)	**High: verifiable core (GF/RGF-bounded) + auditable evidence artifacts; uncertainty isolated as granules**	**High: graded premises handled via information granules and threshold atoms; clinically tunable interfaces**	**High: explicit policy + workflow (FO-PDL) enabling accept/ abstain/ escalate and controlled execution**	**Up-front design of granules/ thresholds and governance procedures; integration overhead; requires continuous calibration**

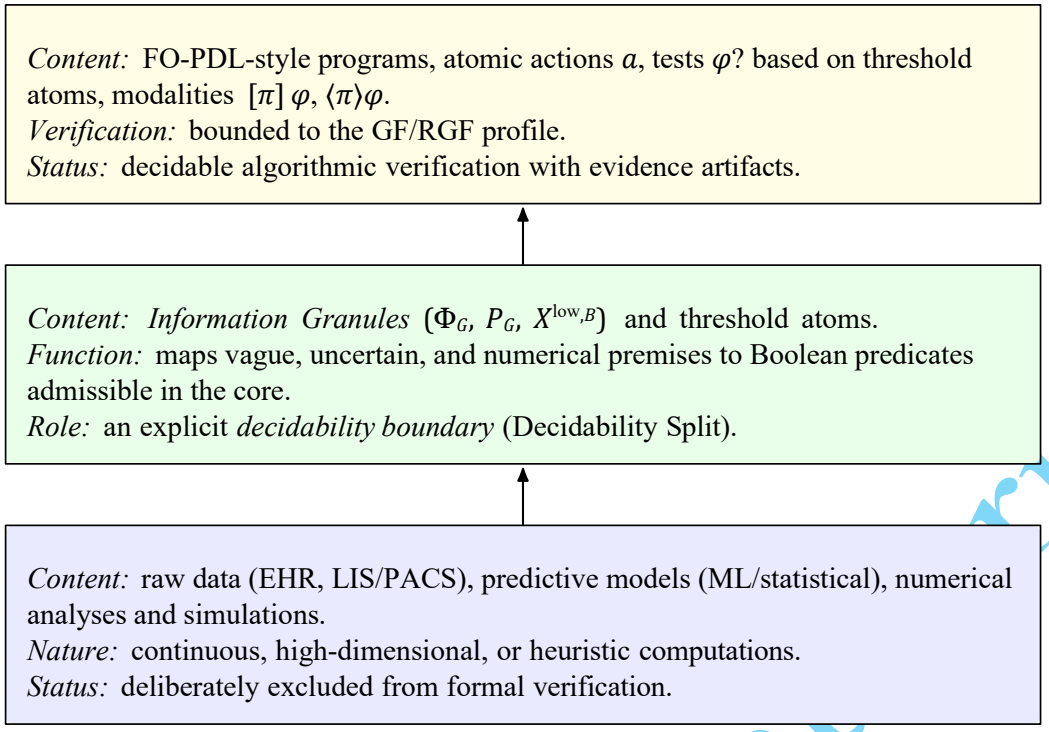

Figure 2: Layered epistemic structure of AGL and the decidability boundary. Formal verification applies only to the logical-procedural core (GF/RGF profile), while the granule layer provides an auditable interface that safely encapsulates complex computations outside the verifiable core. For a step-by-step reading guide, see Section 1.

Table 5: A bridge from clinical data to auditable threshold atoms. Thresholds r are illustrative (non-normative); in AGL arithmetic remains in the granule computation layer, while the rule/procedure core operates exclusively on threshold atoms.

Granule (type)	**Meaning (clinical intuition)**	**Typical data source**	**Atom in the core**
$\Phi_{G_{\text{PSA}}}(u)$ (fuzzy)	"How elevated PSA is" after normalization	LIS / laboratory results (PSA)	$G^{\Phi}_{\text{PSA}\geq r}(u)$
$\Phi_{G_{\text{PSAD}}}(u)$ (fuzzy)	"How elevated PSAD is" (PSA / prostate volume)	LIS (PSA) + MRI/US (volume)	$G^{\Phi}_{\text{PSAD}\geq r}(u)$
$\Phi_{G_{\text{PI-RADS}}}(u)$ (fuzzy)	"Strength of lesion suspicion in MRI" (granular view)	MRI report (PI-RADS)	$G^{\Phi}_{\text{PI-RADS}\geq r}(u)$
$\Phi_{G_{\text{Age-HR}}}(u)$ (fuzzy)	"Age as a risk factor" (gradual)	EHR / registration	$G^{\Phi}_{\text{Age-HR}\geq r}(u)$
$\Phi_{G_{\text{Gleason}\geq 7}}(u)$ (crisp)	"Whether Gleason meets the threshold" (0/1)	Histopathology / pathology report	$G^{\Phi}_{\text{Gleason}\geq 7}(u)$
$P_{G_{\text{Compliance}}}(u)$ (prob.)	"Probability of adherence" (stochastic uncertainty)	EHR + interview + models/surveys (local calibration)	$G^{P}_{\text{Compliance}\geq r}(u)$ or $G^{P}_{\text{Compliance}<r}(u)$
$P_{G_{\text{LN+}}}(u)$ (prob.)	"LN+ risk from a model/nomogram"	EHR + histopath/biopsy + validated predictive model	$G^{P}_{\text{LN+}\geq r}(u)$
$\Phi_{G_{\text{CoresPos}}}(u)$	(fuzzy/crisp)	"Share of positive biopsy cores	$_{\text{CoresPos}\leq r}$ / burden"

Histopathology / biopsy report	(u)		
G^{Φ}			
$X^{\text{low},B}(u)$ (rough)	"Certainly satisfies concept X under attributes B"	Attributes B as "observation glasses"	$X^{\text{low},B}(u)$
$X^{\text{bd},B}(u)$ (rough)	"Boundary zone: requires refining attributes"	Missingness / ambiguity in attributes B	$X^{\text{bd},B}(u)$