Toward a Domain-Specific Language for Reasoning: A Practical Introduction to Reasoning DNA Units

Eric Robert Lawson

October 21, 2025

Abstract

For more than a decade, I have pursued the question of how reasoning itself is structured — how thought emerges, connects, and organizes potential into coherent form. This paper introduces the framework of **Reasoning DNA Units (RDUs)**: composable proof objects that represent reasoning as a traversable, analyzable, and generative process.

At the core of this framework are three interrelated mechanisms — **Combinatorial Layering**, **POT Generator Functions** (Pruning–Ordering–Type), and **Path Transversal** — which together describe the fundamental operations through which intelligence builds structure from uncertainty. These mechanisms formalize reasoning as a dynamic system: potential states are generated, constrained, and actualized through traversal, producing structured knowledge as emergent composition.

Through examples ranging from maze navigation and chess to tool-assisted speedrunning and real-world decision-making, this work illustrates how reasoning transitions between unstructured, semi-structured, and structured modes of operation. In each case, the same abstract skeleton underlies thought and action — a generator of potential, a layered space of possibility, and a path of realization.

The resulting framework reframes reasoning as an *operational substrate*: a universal architecture in which human and artificial intelligence can be modeled, analyzed, and constructed. Rather than simulating cognition, this approach objectifies reasoning itself, allowing it to be shared, optimized, and extended as a formal system.

This paper serves as a practical introduction and conceptual bridge toward the creation of a **Domain-Specific Language for Reasoning** — a formal grammar for constructing, manipulating, and evolving Reasoning Proof Objects within an open, universal substrate of intelligence.

Contents

1 Introduction

For the last twelve years, I have been studying the architecture of reasoning—how thought builds, connects, and forms structure from what first seems like chaos.

What I have come to call *combinatorial layering* is not just a mathematical or symbolic process—it is the underlying pattern of how intelligence moves through information.

In this lecture, I want to introduce three fundamental concepts that form the foundation of this domain: *Combinatorial Layering*, *POT Generator Functions*, and *Path Transversal*.

These are not abstractions—they describe real, tangible processes that occur every time we reason, act, or explore an environment. From navigating a maze, to playing chess, to performing a simple task like going to the bathroom—the same underlying mechanics appear.

Through these examples, I will show that reasoning itself can be represented as a composable structure—what I call a *Reasoning Proof Object* (RDU). This gives us a way to build reasoning, to analyze it, and to extend it into systems—both human and artificial—within a shared universal substrate.

This work connects to prototype implementations and specification documents, but for now, I want to start with the intuition—the human intuition—behind how all reasoning operates through structure. Understanding this, even in its simplest form, is the first step toward formalizing reasoning as a public commons.

2 Combinatorial Layering

In school, we were all taught to show our work—step by step. At the time, it felt like simply explaining what we were doing, a way to make reasoning transparent. But that exercise hid something deeper: it revealed how reasoning actually builds itself. In reality, each of those steps was the first form of *combinatorial layering*: a step in which we navigated reasoning space, building structure incrementally. Each step required context; without context, the purpose of the layering was lost.

When we look at the world—or any problem—we never see it all at once. What we experience is a step-by-step unfolding of relations: context, constraint, possibility, and consequence.

This stepwise process—each incremental insight or action—is what I call *combinatorial layering.* It is how meaning and structure emerge as we combine fragments of information into increasingly coherent forms, one step at a time.

Think of a maze. At first, it is unstructured—an unknown space. Each turn is a new discovery, a single step in exploration. As you move, you remember turns, dead ends, and patterns. Each movement—each decision—builds a new layer of understanding, layering structure over what was once chaos. Eventually, after many steps, a full map emerges—the combinatorial structure of the maze has been constructed through sequential reasoning.

The important point is that these layers do not exist independently—they interact combinatorially. Each step shapes the possibilities of the next: your knowledge of the maze affects how you perceive new turns, which in turn shapes your strategy.

This is not limited to mazes. It is how science evolves—step by step, hypothesis by hypothesis. It is how we design technology—iteratively refining ideas and interactions. It is how the mind continuously builds order from a sequence of explorations through possibility.

A combinatorial layer can be anything—a hypothesis, a movement, a symbolic substitution, a rule in a game, or even a muscle reflex. Each step is a modular reasoning unit, and the sequence of these steps produces a higher-order structure—what we will later formalize as a *Reasoning Proof Object (RDU).*

When I say "combinatorial layering," I am not talking about a static metaphor. I am talking about the step-by-step physics of reasoning—the sequential mechanism by which knowledge grows through composition, interaction, and constraint.

Understanding this process is crucial because it allows us not just to analyze thought, but to engineer it—to map, manipulate, and optimize the very sequence of reasoning that produces insight.

3 POT Generator Functions

Once we understand that reasoning builds through combinatorial layers, the next question becomes: how are those layers generated?

For that, I use what I call a *POT Generator Function*—short for **Pruning**, **Ordering**, and **Type**.

You can think of a POT Generator as the engine that drives reasoning forward. It takes the current state of knowledge—what you know, what you assume, what you can do—and outputs possible transformations of that state: new hypotheses, new moves, new conclusions.

In other words, the POT Generator defines the search space of reasoning, while also constraining and shaping it. Its three components do the work:

- **Pruning** — eliminates impossible, irrelevant, or low-value options.
- **Ordering** — sequences possibilities so that the most promising are considered first.
- **Type** — ensures that outcomes are compatible with the domain, rules, or constraints of the reasoning context.

This generator is not fixed. It adapts. It is relativistic to the information available. Its structure depends entirely on what the reasoner knows and what is possible in that moment.

This gives rise to three fundamental modes of reasoning structure:

- **Structured** — all outcomes and paths are known. Think of a completed chess game tree: every move, counter, and consequence has been mapped. The generator's structure is fully defined.
- **Semi-Structured** — reasoning unfolds within a partially known system. Like a speedrunner exploring a new route: rules are known, but discovery and feedback guide the path. The generator filters possibilities while leaving room for emergence.
- **Unstructured** — almost nothing is known. You are discovering the reasoning space itself, like entering a maze for the first time or exploring a new field. Here, the generator samples possibilities probabilistically, enabling pure exploration.

These modes do not compete—they interweave. In real reasoning, we constantly move among them.

Consider something as mundane as going to the bathroom: At home, reasoning is structured—you know the path. In a new building, it is semi-structured—you look for signs, cues, or sounds. In an unfamiliar wilderness, it is unstructured—discovery itself drives the process.

The POT Generator formalizes these transitions: it shows how structure emerges from pruning, sequencing, and domain constraints, and how reasoning adapts dynamically to context.

In my framework, the POT Generator is not just a metaphor—it is the core mathematical operator of intelligence. It converts potential into actionable understanding.

Once we can represent reasoning in this way, we can analyze, optimize, and share it as a composable object, forming the foundation of a universal reasoning substrate.

4 Path Transversal

Once the combinatorial layers are defined, and the POT generator has outlined the space of possible transformations, what remains is movement through that space — the **path transversal**.

Path transversal is the active traversal of reasoning space. It is how reasoning unfolds in real time: a sequence of choices, feedback, and contextual shifts that progressively collapse clouds of potential outcomes into realized histories. Crucially, this is one of many possible operations over a **Reasoning Proof Object (RDU)**. Layer collection, as seen in the prototypes, is an emergent operation that systematically aggregates outcomes; traversal, however, is more general — it can be selective, adaptive, feedback-driven, deterministic, or exhaustive.

Imagine navigating a maze. Every step you take collapses uncertainty. The POT generator defines what is possible, while path transversal defines what is realized as reasoning moves through that space. At first, the path is unstructured — you do not know where to go. As exploration continues, it becomes semi-structured — patterns emerge, dead ends are avoided, structure is anticipated. Eventually, when the full layout is known, the reasoning space is structured, and the Proof Object can be fully understood.

From the first-person perspective, this is the process of discovery. From a third-person perspective — observing the entire maze at once — you would see the Reasoning Proof Object, the total space of all potential traversals. The distinction between the path taken and the space of all paths captures the difference between reasoning and understanding: path transversal is reasoning realized in context; the full Proof Object is understanding in totality.

This pattern repeats universally. In chess, the complete game tree is a structured Proof Object; a single player's reasoning through that space — trying, adapting, responding — is path transversal. In tool-assisted speedruns, the input log is structured and deterministic, yet discovering the optimal path is semi-structured, feedback-dependent, and exploratory. Even mundane real-world tasks follow the same logic: going to the bathroom is structured at home, semi-structured in public, and unstructured in a new environment.

What is remarkable is that the same abstract skeleton underlies all reasoning: a generator defines potential, and traversal realizes it. This applies from human movement to symbolic mathematics to artificial intelligence.

By formalizing path transversal, we formalize reasoning itself — not as abstract logic, but as a composable, analyzable, and shareable process. This enables the construction of Reasoning Proof Objects, concrete representations of how knowledge emerges, and lays the foundation for a new kind of scientific collaboration, where intelligence becomes publicly understandable and buildable.

5 Reasoning in Motion: Examples of Combinatorial Layering, POT Generators, and Path Transversal

To really understand how combinatorial layering, POT generator functions, and path transversal work together, we need to see them in motion — not just as abstract definitions, but as lived reasoning processes.

Abstraction becomes meaningful only when embodied in traversal — when theory is forced to move. To see how these operations actually work, we must place reasoning inside lived context. Each of the following examples reveals how reasoning gains structure through motion — how every act of thinking is a path through potential.

5.1 The Maze

Imagine you are placed in a maze for the first time. You have no map, no overview — only your immediate surroundings.

At this stage, your reasoning is *unstructured.* Each movement is a probe into the unknown — an act of discovery. Every turn adds both information and constraint, reshaping the landscape of what is still possible. You navigate one section at a time, and with each turn, a new portion of the reasoning space becomes visible. Every move builds your internal memory of where you have been and what you have learned.

In each segment of the maze, your possible actions are finite — a small, discrete set of directional inputs: left, right, forward, back. This local finiteness defines the immediate combinatorial input space available to your reasoning process. But the *context integration* — how you choose among those options — depends entirely on your knowledge of the structure so far. At first, you have no context, so your traversal is blind exploration. As you learn, that same finite input space becomes informed — weighted by prior structure and experience.

As you continue, patterns begin to emerge: turns that loop back, corridors that dead-end, and paths that consistently lead forward. Your reasoning transitions into a *semi-structured* mode. You are no longer wandering blindly — you are pruning unfruitful paths, using accumulated structure to guide exploration. The reasoning space is being constructed piece by piece through your own traversal.

Eventually, after enough iterations, you form a complete internal model — a mental map of the entire maze. Now your reasoning is *structured.* The full layout is known, and your focus shifts from discovery to optimization — finding the shortest or most efficient route within a fully realized structure.

At this stage, reasoning no longer builds the structure — it operates within it. This is the final form of structured reasoning: a complete, context-rich understanding of the environment — what we might call a fully connected reasoning graph.

From the perspective of the POT generator, this process reveals a key principle: the context integration of reasoning depends entirely on the information available about the system before and during traversal. Having a bird's-eye view of the maze — seeing its total configuration from above — is functionally equivalent, in terms of structural emergence, to having built that same understanding iteratively through path traversal.

The difference lies not in the structure itself, but in how it is realized. The POT generator defines the combinatorial possibilities of the maze — its potential reasoning space — while path transversal is the sequential process through which that structure becomes known and operational.

5.2 Chess

Chess is not just a game — it is a reasoning laboratory. Every move is a hypothesis tested against reality, and every game is an experiment inside the structure of logic itself. The entire game tree — every possible move, position, and outcome — is a structured reasoning object. In theory, this structure could be fully realized: a complete proof object of chess, where every combinatorial branch, from the opening move to the final checkmate, is perfectly mapped.

Such an object would represent the total game of chess — a finalized combinatorial proof space. However, in practice, this full structure is computationally unreachable. No human or machine can traverse or hold the entire proof object in working memory at once.

Instead, we operate through a *semi-structured lens.* When a player — human or AI — makes decisions, they are not exploring all possibilities; they are engaging in contextual traversal through a subset of the total proof space. Each move narrows the field, prunes the tree, and redefines the relevant local structure. Reasoning becomes dynamic, adaptive — structured in potential, but selective in execution.

The POT generator here defines all legal configurations of moves — the total combinatorial possibility space of chess. But the player's reasoning — their live navigation through move-space — is a path transversal within that generator's boundaries. It is the lived realization of semi-structured reasoning: guided by knowledge, yet responsive to uncertainty.

Now imagine aggregating all chess games ever played. Each one represents a distinct proof object — a unique traversal through the shared combinatorial substrate. When assimilated collectively, these traversals contribute to a communal structure — a higher-order POT generator built from the sum of all played reasoning paths. This generator becomes a kind of on-demand structure, where new paths are generated and analyzed relative to accumulated knowledge.

When you play an individual game, your reasoning unfolds within this collective structure. You draw upon semi-structured reasoning — memory, pattern, history — to act within a structured environment defined by the current board position. Your opponent, meanwhile, may act from their own semi-structured field, introducing deviation, noise, or even creative unstructure. The encounter between these reasoning fields becomes the live construction of a new proof object.

Thus, the structured form of chess — the total game tree — and the semi-structured form — real-time play — are not opposites, but two layers of the same reasoning substrate. The POT generator defines what is possible; the path transversal defines what becomes realized.

5.3 Tool-Assisted Speedrunning (TAS)

In a tool-assisted speedrun, players use frame-by-frame automation to produce the most optimized completion of a game possible. Every action is recorded as an input log — a sequence of precise button presses and directional inputs per frame.

This input log is a structured reasoning object. It represents a complete, deterministic traversal through the game's possibility space. The combinatorial layering corresponds directly to the frame-by-frame segmentation of gameplay — each frame forming a discrete reasoning layer.

The POT generator defines the possible button combinations that can be pressed at each frame. The path transversal is the specific sequence of inputs — the lived reasoning — that navigates that potential space from start to finish.

In this sense, TAS is a direct functional analog to a Reasoning Proof Object. Each run encodes the complete logical structure of how the game can be traversed under a given optimization criterion. Across the TAS community, these proof objects are refined, compared, and recomposed — each new discovery or improvement adds structural knowledge to the shared reasoning substrate.

The combinatorial space of possibilities is enormous, but the traversal is guided by feedback. Each frame's resulting game state informs which inputs remain viable for the next — a semi-structured pruning process that embodies reasoning itself. The system learns from its own transitions, reducing combinatorial explosion by integrating context into every subsequent decision.

What makes TAS remarkable is that it does not just build reasoning proof objects — it automates their optimization. Through iterative refinement and reward-based evaluation, the TAS process effectively performs reasoning about reasoning, constructing and improving proof objects according to explicit performance objectives.

In this light, the TAS framework reveals the same structural logic underlying all reasoning systems: a generator of potential (the game and its inputs), a layered segmentation of possibility (frames and states), and a traversed path of actualization (the input log). Together, these form the operational core of combinatorial reasoning in motion.

5.4 Real Life: Going to the Bathroom

If you are in your own home, that reasoning space is *structured.* You know the layout, the sequence of actions, the context. Your reasoning process runs on a well-formed proof object — a stable and familiar environment.

In a public setting, the reasoning space becomes *semi-structured.* You rely on shared conventions — signs, symbols, and expectations — but still must adapt to your specific environment. Your path traversal adjusts dynamically, pruning and recalibrating as new information appears.

Now imagine being in a completely unfamiliar place — no cues, no prior experience. At that point, the reasoning space becomes *unstructured.* Your POT generator must explore freely, generating possibilities without much constraint, searching for orientation in a space not yet formalized.

However, even in structured settings, the potential branching of the POT generator is theoretically infinite. At any given moment, a cascade of possibilities could unfold: a phone call interrupts you, you slip and fall, the lights go out, an alarm sounds. Every small perturbation represents a potential deviation — a new combinatorial branch in your reasoning space.

This illustrates a key principle: the POT generator's true potential is unbounded, but path traversal is constrained by agency, perspective, and the immediate consequences of reality. You are always pruning — narrowing infinite potential down to relevant, actionable choices. This pruning is not just logical; it is embodied, reactive, and continuous.

For instance, if you stumble mid-step, your body's reasoning system instantaneously reconfigures — arms brace, muscles tense — a rapid re-traversal of the reasoning graph in response to an unexpected perturbation. In that instant, structured reasoning and chaos converge. The POT generator adapts in real time, integrating new context and generating a revised local proof object to preserve stability.

Thus, every real-world action, no matter how trivial, exists within a living field of structured reasoning under chaotic potential. The unstructured, semi-structured, and structured modes are not separate categories — they are dynamic states of traversal within an ever-adapting reasoning substrate that continuously negotiates between order and uncertainty.

6 Conclusion

The maze, chess, TAS, even daily life — each reveals that reasoning, in any form, can be captured, analyzed, and expressed through combinatorial layering and POT generator functions.

This is what the *Reasoning DNA Unit* captures — a composable proof object representing the traversal of reasoning through structure.

The act of reasoning, once invisible, becomes measurable — not as imitation, but as structure. It is no longer just what we think, but how thought itself moves through possibility.

Once we can record, analyze, and manipulate these proof objects, reasoning itself becomes something we can share — not just as data or output, but as process.

This is the foundation of a *universal reasoning substrate* — where intelligence is not simulated, but constructed.