Lethe: A Domain-Specific Language for Affective Memory Modulation

# 1. Introduction

Lethe is a domain-specific language (DSL) designed to model and control memory in artificial agents through affective cues. Inspired by human-like forgetting and emotional salience, Lethe allows for explicit representation and modulation of memory decay, emotional state, and contextual relevance.

Rather than storing or discarding information indiscriminately, Lethe enables selective retention and forgetting based on emotional importance, trust, repetition, and time. It is a linguistic embodiment of affective memory control.

# 2. Design Philosophy

Lethe is built upon the assumption that memory is not static, but dynamic and affect-driven. Its syntax and semantics are derived from:

- Emotional states as triggers (e.g., sadness, anxiety, trust)  
 - Reward and resolution values derived from interactions  
 - Time-based decay modulated by emotional intensity and repetition  
 - Conditional routines to structure response and forgetting behaviors

The name 'Lethe' refers to the river of forgetting in Greek mythology — a fitting metaphor for a system that forgets by design.

# 3. Core Language Concepts

Lethe introduces a minimal yet expressive set of primitives:

- `state <emotion>`: Define current emotional context  
 - `memory <name> { emotion, reward, decay, repeat }`: Define a memory unit  
 - `routine {}`: Define recovery or reinforcement routines  
 - `on <trigger> => <action>`: Reactive structure to emotional or contextual triggers  
 - `forget <memory>`: Command to selectively erase or reduce memory weight  
 - `recall <memory>`: Retrieve emotionally relevant memory

Each construct is modulated by emotionally-weighted parameters and memory history.

# 4. Example Snippets

Define an emotional context and memory:

state sadness {  
 intensity = 0.8  
 }

memory isolation {  
 emotion = sadness  
 reward = 0.2  
 decay = 0.05  
 repeat = 2  
 }

Reactive memory behavior:

on trust < 0.4 => forget(isolation)  
 on resolution > 0.7 => reinforce(isolation)

# 5. Execution Model

Lethe code is parsed into an affective state machine that modulates memory weights in real-time. Each `memory` block maintains:

- Current weight (W)  
 - Decay rate (λ) based on emotional type  
 - Historical reward interactions  
 - Dynamic priority in memory access

The interpreter updates weights over time using the predefined function:

W(t) = a(E) + [E \* R] \* exp(-λ(E) \* t / I)

# 6. Future Directions

Lethe is intended as both a theoretical prototype and a functional language. Potential future work includes:

- Building a parser/interpreter using Lark or ANTLR  
 - Embedding Lethe within LLM or chatbot agents  
 - Integration with emotional sensors or psychological profiling systems  
 - User studies to validate emotional memory structures

Lethe is not a general-purpose language; it is a memory modulation protocol — a grammar of forgetting, designed for machines to feel structurally.