Sema Language Overview

Sema is a Scheme-like Lisp with LLM primitives built in Rust.
It provides 450+ builtins across 19 modules including math,
strings, lists, maps, I/O, HTTP, regex, and LLM operations.

Key features:
- Native LLM completions, chat, and structured extraction
- Tool use and agentic loops as first-class forms
- 11 providers auto-configured from environment variables
- In-memory vector store with cosine similarity search
- PDF text extraction and vision capabilities

Getting Started

Install Sema via Homebrew: brew install helgesverre/tap/sema-lang

Or download a binary from GitHub releases.

Configure your LLM provider by setting environment variables:

- ANTHROPIC_API_KEY for Claude models
- OPENAI_API_KEY for GPT models
- GEMINI_API_KEY for Google Gemini

Run a file: sema examples/hello.sema

Start the REPL: sema

Architecture

Sema is a Cargo workspace with 6 crates:

- sema-core: Value types, environment, error handling
- sema-reader: Lexer and s-expression parser
- sema-eval: Trampoline-based TCO evaluator
- sema-stdlib: 350+ native functions across 19 modules
- sema-llm: LLM providers, embeddings, vector store
- sema: CLI binary with REPL and integration tests