
Simple Natural Language Reasoning Agent — R01

1. Objective

To build a **Simple Natural Language Reasoning Agent** that can:

1. Parse English sentences into structured logical forms.
2. Reason formally over those forms (using Z3 or Prolog).
3. Explain its deductions in natural language.
4. Learn new transformation patterns over time — *without hard-coding*.

This prototype serves as the seed for a scalable **neuro-symbolic AGI framework** combining modern language models with symbolic reasoning.

2. Guiding Principles

The Golden Rule of AGI Design

“Don’t reinvent the wheel — integrate and extend.”

True intelligence emerges from **composing existing competencies** rather than re-implementing them.

We build on mature tools (spaCy, Stanza, AllenNLP SRL, Z3, Prolog) and focus our innovation on the *interfaces and abstractions* that link them.

Corollaries

1. **Never parse language by hand.**
Always use tested, learnable parsers.
 2. **Never fake generality with toy cases.**
Design every stage for scaling and unseen inputs.
 3. **Build bridges, not silos.**
Use shared representations (JSON, FOL, graphs) for all modules.
-

3. System Overview

English



spaCy / Stanza Parser + SRL



Semantic Mapper (learned model)



Logic Form (FOL / Prolog / Z3)



Symbolic Reasoner



Explanation Generator



Example Store / Meta-Learner (for self-improvement)

This architecture separates **linguistic perception**, **symbolic inference**, and **meta-learning**, enabling generalization and abstraction.

4. Rationale — Why Not Hand-Written Rules

Early prototypes often used ad-hoc regex such as:

```
m = re.match(r"all\s+(.)s+are\s+(.)", t)
```

These approaches are anti-scalable and **anti-AGI**.

They simulate intelligence for a few patterns but fail to generalize.

Our rule: *no hand parsing, no brittle shortcuts*.

Instead, we rely on:

Task	Modern Tool
Sentence splitting, POS, deps	spaCy / Stanza
Semantic roles	AllenNLP SRL
Logical mapping	Learned semantic parser (T5-SP / LLM)
Reasoning	Z3 / Prolog
Explanation	Template + LLM summarizer

5. Integration with DeepAgent Concepts

Recent work — **DeepAgent: A General Reasoning Agent with Scalable Toolsets** (Li et al., 2026) — extends our Golden Rule even further by allowing agents to **discover and extend their toolsets** autonomously.

What DeepAgent Achieves

- **Dynamic tool retrieval** from large registries.
- **Unified reasoning trace** without fixed step order.
- **Memory folding** (episodic / working / tool).
- **Reinforcement learning for tool selection (ToolPO)**.

These directly support our modular philosophy.

What DeepAgent Misses

It omits a **symbolic reasoning substrate**.

No mention of Prolog, Z3, or formal logic checking.

Reasoning remains procedural and neural rather than semantic and verifiable.

Capability	DeepAgent	Our Agent
Tool autonomy	✓	✓
Symbolic reasoning	✗	✓

Capability	DeepAgent	Our Agent
Logical consistency checking	✗	✓
Rule induction / abstraction	✗	✓
Explainable proofs	partial	✓

Our Extension

We insert a **symbolic reasoning tier** into the same tool ecosystem:

LLM Controller ↔ Tool Registry

↓

(symbolic tools)

- └─ english_to_logic
- └─ logic_reasoner_z3
- └─ prolog_infer
- └─ logic_explainer

The meta-controller learns which symbolic tool best supports each task.

6. Symbolic Reasoning Layer

Symbolic engines provide:

- **Generalization** through quantifiers and variables.
- **Abstraction** via explicit predicates and hierarchies.
- **Verifiability** of each inference step.
- **Structured learning signals** (proof success/failure).

Together, they transform an LLM pattern-matcher into a true reasoner.

7. Implementation Plan (R01)

1. **Parsing & semantics:**
Use `spaCy` + `AllenNLP SRL` to obtain dependency and role structure.
2. **Logic conversion:**
Start with a few hand-verified sentence→logic examples, train a lightweight seq2seq mapper.
3. **Reasoning:**
Interface with **Z3** for entailment checks and consistency tests.
4. **Learning:**
Store (`sentence`, `logic`, `result`) triples; meta-controller refines mappings from success/failure.
5. **Explanation:**
Template generator:

“Because *{premise1}* and *{rule}*, therefore *{conclusion}*.”

8. Future Work (R02+)

Stage	Focus
R02	Add Prolog reasoning; enable transitive inference.
R03	Rule induction from examples (meta-learner).
R04	Integrate DeepAgent-style tool discovery and memory folding.
R05	Continuous self-improvement via symbolic feedback loops.

9. Key Insight

DeepAgent teaches us how to scale the agentic process — memory, autonomy, and tool use.

Our symbolic layer provides the semantic grounding — understanding *why*.

10. Reference

Li et al. (2026). *DeepAgent: A General Reasoning Agent with Scalable Toolsets*.
[arXiv:2510.21618](https://arxiv.org/abs/2510.21618)

File: `SimpleNaturalLanguageReasoningAgent_R01.md`

Purpose: baseline document for integrating DeepAgent-style autonomy with symbolic reasoning and logic-based generalization.