

# Code Documentation: src

UMEDCTA Repository

December 3, 2025

## Contents

1	src/DialecticalInterpreter.jsx	2
2	src/main.jsx	19

# 1 src/DialecticalInterpreter.jsx

```

1 import React, { useState } from 'react';
2 import { AlertCircle, Sparkles, GitBranch, CheckCircle, XCircle, Loader2, Copy } from 'lucide-react';
3
4 // Helper function to call our backend API
5 const callAnthropic = async (messages, maxTokens = 2000) => {
6   const response = await fetch('http://localhost:3001/api/anthropic', {
7     method: 'POST',
8     headers: { 'Content-Type': 'application/json' },
9     body: JSON.stringify({
10       model: 'claude-sonnet-4-20250514',
11       max_tokens: maxTokens,
12       messages
13     })
14   });
15
16   if (!response.ok) {
17     const error = await response.json();
18     throw new Error(error.error || 'API request failed');
19   }
20
21   return response.json();
22 };
23
24 const DialecticalInterpreter = () => {
25   const [inputText, setInputText] = useState('');
26   const [interpretation, setInterpretation] = useState(null);
27   const [isProcessing, setIsProcessing] = useState(false);
28   const [evolutionHistory, setEvolutionHistory] = useState([]);
29   const [showEvolutionPanel, setShowEvolutionPanel] = useState(false);
30
31   // Conversation state for follow-ups
32   const [conversationHistory, setConversationHistory] = useState([]);
33   const [followUpQuestion, setFollowUpQuestion] = useState('');
34   const [showFollowUp, setShowFollowUp] = useState(false);
35
36   // Current axiom set (starts with base PML axioms)
37   const [axiomSet, setAxiomSet] = useState([
38     {
39       id: 'base_rhythm',
40       content: 's(u) => s(comp_nec a)',
41       source: 'core',
42       type: 'material',
43       context: 'Fundamental dialectical rhythm: unity necessarily generates tension'
44     },
45     {
46       id: 'sublation',
47       content: 's(lg) => s(exp_nec u_prime)',
48       source: 'core',
49       type: 'material',
50       context: 'Letting go necessarily produces new unity'
51     },
52     {
53       id: 'oobleck',
54       content: 's(comp_nec P) => o(comp_nec P)',
55       source: 'core',
56       type: 'material',
57       context: 'Subjective compression crystallizes objective content'
58     }
59   ]);
60
61   // Iteration tracking: how many times has this text been read?
62   const [iterationDepth, setIterationDepth] = useState(0);

```

```

63  const [formalizedConcepts, setFormalizedConcepts] = useState([]);
64
65  const processText = async () => {
66    setIsProcessing(true);
67    setInterpretation(null);
68
69    // Check if this is a re-read of the same text
70    const previousRead = conversationHistory.find(
71      item => item.type === 'interpretation' && item.content.original === inputText
72    );
73    const isRereading = !!previousRead;
74    const currentIteration = isRereading ? iterationDepth + 1 : 1;
75
76    try {
77      // Phase 1: Parse text into PML
78      const iterationContext = isRereading ? `

ITERATION DEPTH: ${currentIteration} (This is a RE-READING)

CRITICAL: On second+ readings, established interpretations become PART OF the phenomenology.
- First reading: All material inferences (discovering what concepts mean)
- Second+ reading: Some concepts are now FORMALIZED (structural scaffolding, not discovery)
- The reader brings prior understanding as background assumptions

Previous interpretation of this text:
${JSON.stringify(previousRead?.content.logic.interpretation)}

Key concepts from previous read:
${JSON.stringify(previousRead?.content.pml.key_concepts)}

These concepts now operate as FORMAL structure rather than material discovery.` :
79      ITERATION DEPTH: 1 (FIRST READING)
80
81      This is a first encounter with the text. The reader:
82      - Discovers concepts through material inference (content-based reasoning)
83      - Experiences genuine novelty and surprise
84      - Builds understanding from scratch without prior scaffolding`;

85      const parsePrompt = `You are a philosophical interpreter using Polarized Modal Logic (PML).

CRITICAL FRAMING: PML tracks the PHENOMENOLOGY OF READING – the temporal, embodied experience of working
→ through philosophical text. It models:
- How tension builds as you encounter concepts (compression ↓)
- How understanding releases when connections form (expansion ↑)
- The cognitive resources consumed in the reading process
- The subjective experience of the dialectical rhythm

${iterationContext}

PML Vocabulary:
- Three modes: s(P) = subjective experience, o(P) = objective claim, n(P) = normative commitment
- Four modalities: comp_nec(P) = necessary compression (↓), exp_nec(P) = necessary expansion (↑),
  comp_pos(P) = possible compression, exp_pos(P) = possible expansion
- Dialectical rhythm: u → comp_nec(a) → exp_pos(lg) → exp_nec(u_prime)
  (unity → tension → possibility of release → new understanding)

Current Axiom Set (including evolved axioms):
${axiomSet.map(a => ` - [${a.type}] ${a.content} // ${a.context}`).join('\n')}

Formalized Concepts (operate as structural scaffolding on re-reads):
${formalizedConcepts.length > 0 ? formalizedConcepts.join(', ') : 'None yet'}

Text to interpret AS A TEMPORAL UNFOLDING:
"${inputText}"

Task: Formalize how a reader EXPERIENCES this text moving through it sequentially. Track:

```

```

127 | 1. Initial subjective state (what's your starting point?)
128 | 2. Compressive moments (where does tension/confusion arise?)
129 | 3. Expansive moments (where does understanding open up?)
130 | 4. The temporal sequence of dialectical transitions
131 | 5. Resource costs (where is the text cognitively demanding?)
132 | 6. ${isRereading} ? 'FORMALIZATION: What concepts from prior reads now operate as formal structure?' :
→ 'DISCOVERY: What concepts are being discovered for the first time?'
133 |
134 Respond ONLY with valid JSON (no markdown):
135 {
136   "formalizations": [
137     {"step": "initial_state", "pml": "...", "explanation": "reader's starting point", "temporal_moment":
→ "beginning", "inference_type": "material|formal"}, ...
138     {"step": "compression", "pml": "...", "explanation": "where tension builds", "temporal_moment":
→ "middle", "inference_type": "material|formal"}, ...
139     {"step": "expansion", "pml": "...", "explanation": "where understanding releases",
→ "temporal_moment": "late", "inference_type": "material|formal"} ...
140   ],
141   "reading_experience": "Overall phenomenological description of working through this text",
142   "key_concepts": ["concept1", "concept2"],
143   "formalized_this_iteration": ["concepts that should become formal scaffolding on next read"],
144   "iteration_depth": ${currentIteration}
145 };
146
147   const parseData = await callAnthropic([
148     { role: "user", content: parsePrompt }
149   ], 2000);
150   let parseText = parseData.content[0].text.trim();
151   parseText = parseText.replace(/\`json\n?/g, "").replace(/\`\`\n?/g, "").trim();
152   const parsedPML = JSON.parse(parseText);
153
154   // Phase 2: Run logic and generate interpretation
155   const interpretPrompt = `Using the PML formalizations, generate an interpretation by tracing
→ through the logic.
156
157 Formalizations:
${JSON.stringify(parsedPML.formalizations, null, 2)}
158
159 Available Axioms:
${axiomSet.map(a => ` - ${a.content}`).join('\n')}
160
161 Apply the axioms to derive conclusions. Show each inference step.
162
163 Then, provide your philosophical interpretation of the text based on this logical structure.
164
165 Respond ONLY with valid JSON:
166 {
167   "proof_steps": [
168     {"premises": [...], "axiom_used": "...", "conclusion": "...", "explanation": "..."}
169   ],
170   "interpretation": "Your philosophical reading of the text...",
171   "key_insights": ["insight1", "insight2"]
172 };
173
174
175   const interpretData = await callAnthropic([
176     { role: "user", content: interpretPrompt }
177   ], 3000);
178   let interpretText = interpretData.content[0].text.trim();
179   interpretText = interpretText.replace(/\`json\n?/g, "").replace(/\`\`\n?/g, "").trim();
180   const logicResult = JSON.parse(interpretText);
181
182   // Phase 3: Critique - Compare against established readings
183   const critiquePrompt = `You are a meta-critic analyzing different LEVELS OF ANALYSIS.
184
185 Original Text: "${inputText}"
186

```

```
187 Our PML Interpretation (PHENOMENOLOGICAL LEVEL – tracking the embodied reading experience):  
188 ${logicResult.interpretation}  
189  
190 CRITICAL CONTEXT: PML tracks the TEMPORAL PHENOMENOLOGY of reading, not atemporal propositional content.  
191 A "contradiction" with traditional interpretations often reveals that we're analyzing different levels:  
192 - Traditional: "What does the text SAY?" (propositional, atemporal)  
193 - PML: "How does it FEEL to work through this text?" (phenomenological, temporal, embodied)  
194  
195 Task:  
196 1. What are the major scholarly interpretations of this passage? (Focus on PROPOSITIONAL content)  
197 2. How does our PHENOMENOLOGICAL reading compare?  
198 3. Are apparent contradictions actually tracking different levels?  
199 4. What might deepen our phenomenological analysis?  
200  
201 DO NOT OUTPUT ANYTHING OTHER THAN VALID JSON:  
202 {  
203   "established_readings": [  
204     {"scholar": "...", "interpretation": "...", "level": "propositional|phenomenological"}  
205   ],  
206   "alignment": {  
207     "level_distinctions": [  
208       {"apparent_contradiction": "...", "actually": "different levels – both valid", "explanation":  
209         "..."}  
210     ],  
211     "genuine_contradictions": [  
212       {"issue": "...", "our_claim": "...", "standard_claim": "...", "severity": "high|medium|low"}  
213     ]  
214   },  
215   "diagnostic": {  
216     "missing_phenomenological_moves": ["what reading experiences are we not tracking?"],  
217     "needed_axioms": [  
218       {"proposed": "...", "rationale": "...", "addresses": "..."}  
219     ],  
220     "pathology_detected": "fixation|bad_infinite|none",  
221     "meta_insight": "What does this text teach us about the phenomenology of reading Hegel?"  
222   }  
223 };  
224  
225   const critiqueData = await callAnthropic([  
226     { role: "user", content: critiquePrompt }  
227   ], 3000);  
228   let critiqueText = critiqueData.content[0].text.trim();  
229   critiqueText = critiqueText.replace(/\`json\n?/g, "").replace(/\`\n?/g, "").trim();  
230   const critique = JSON.parse(critiqueText);  
231  
232   // Combine results  
233   const newInterpretation = {  
234     original: inputText,  
235     pml: parsedPML,  
236     logic: logicResult,  
237     critique: critique,  
238     timestamp: new Date().toISOString(),  
239     iterationDepth: currentIteration  
240   };  
241  
242   setInterpretation(newInterpretation);  
243  
244   // Update iteration tracking  
245   if (isRereading) {  
246     setIterationDepth(currentIteration);  
247   } else {  
248     setIterationDepth(1);  
249   }  
250 }
```

```

251 // Update formalized concepts if this is a second+ reading
252 if (parsedPML.formalized_this_iteration && parsedPML.formalized_this_iteration.length > 0) {
253   setFormalizedConcepts([
254     ...new Set([...formalizedConcepts, ...parsedPML.formalized_this_iteration])
255   ]);
256 }
257
258 // Add to conversation history
259 setConversationHistory([
260   ...conversationHistory,
261   {
262     type: 'interpretation',
263     content: newInterpretation
264   }
265 ]);
266
267 setShowFollowUp(true);
268
269 } catch (error) {
270   console.error("Error processing text:", error);
271   setInterpretation({
272     error: true,
273     message: error.message
274   });
275 }
276
277 setIsProcessing(false);
278 };
279
280 const handleFollowUp = async () => {
281   if (!followUpQuestion.trim() || !interpretation) return;
282
283   setIsProcessing(true);
284
285   try {
286     // Build conversation context
287     const contextMessages = conversationHistory.map(item => {
288       if (item.type === 'interpretation') {
289         return {
290           role: "assistant",
291           content: `I analyzed the text phenomenologically using PML and found:
292             ↪ ${item.content.logic.interpretation}`
293         };
294       } else if (item.type === 'followup') {
295         return [
296           { role: "user", content: item.question },
297           { role: "assistant", content: item.response }
298         ];
299       }
300     }).flat();
301
302     const followUpPrompt = `CONTEXT: You previously analyzed this text using PML (Polarized Modal
303       ↪ Logic), which tracks the PHENOMENOLOGY OF READING – the temporal, embodied experience of
304       ↪ working through text.
305
306 Original text: "${interpretation.original}"
307 Your PML analysis: ${interpretation.logic.interpretation}
308
309 Critique insights: ${JSON.stringify(interpretation.critique.diagnostic, null, 2)}
310
311 User's follow-up question/clarification:
312   "${followUpQuestion}"
313
314 Respond to their question while:

```

```

313 1. Maintaining focus on the PHENOMENOLOGICAL level (how the text feels to read)
314 2. Distinguishing between propositional content vs. reading experience when relevant
315 3. Suggesting refinements to the PML formalization if needed
316 4. Acknowledging where the logic might need evolution
317
318 Be conversational and dialectical. Help refine the interpretation through dialogue.`;
319
320   const followUpData = await callAnthropic([
321     ...contextMessages,
322     { role: "user", content: followUpPrompt }
323   ], 2000);
324   const response = followUpData.content[0].text;
325
326   // Add to conversation history
327   setConversationHistory([
328     ...conversationHistory,
329     {
330       type: 'followup',
331       question: followUpQuestion,
332       response: response,
333       timestamp: new Date().toISOString()
334     }
335   ]);
336
337   // Update interpretation with follow-up
338   setInterpretation({
339     ...interpretation,
340     followUps: [...(interpretation.followUps || []), {
341       question: followUpQuestion,
342       response: response
343     }]
344   });
345
346   setFollowUpQuestion('');
347
348 } catch (error) {
349   console.error("Error in follow-up:", error);
350   alert('Error processing follow-up: ' + error.message);
351 }
352
353   setIsProcessing(false);
354 };
355
356 const accommodateContradiction = async (contradiction, proposedAxiom) => {
357   setIsProcessing(true);
358
359   try {
360     // Sublation: Synthesize new axiom
361     const sublationPrompt = `You detected a contradiction in our PML interpretation.
362
363     Contradiction: ${contradiction.issue}
364     Our claim: ${contradiction.our_claim}
365     Standard claim: ${contradiction.standard_claim}
366
367     Proposed axiom: ${proposedAxiom.proposed}
368     Rationale: ${proposedAxiom.rationale}
369
370     Refine this axiom into proper PML syntax. Ensure it:
371     1. Resolves the contradiction
372     2. Preserves existing valid inferences
373     3. Opens new interpretive possibilities
374
375     Respond ONLY with valid JSON:
376   {
377     "refined_axiom": "PML syntax here",

```

```

378     "integration_strategy": "How this fits with existing axioms",
379     "test_implications": ["What this now lets us infer..."],
380     "context": "One-sentence summary of why this axiom was needed"
381   };
382
383   const sublationData = await callAnthropic([
384     { role: "user", content: sublationPrompt }
385   ], 2000);
386   let sublationText = sublationData.content[0].text.trim();
387   sublationText = sublationText.replace(/\`json\n?/g, "").replace(/\`\n?/g, "").trim();
388   const refinedAxiom = JSON.parse(sublationText);
389
390   // Add to axiom set
391   const newAxiom = {
392     id: `evolved_${Date.now()}`,
393     content: refinedAxiom.refined_axiom,
394     source: 'evolved',
395     type: 'material', // New axioms start as material, may become formal through iteration
396     rationale: proposedAxiom.rationale,
397     addresses: contradiction.issue,
398     context: refinedAxiom.context,
399     timestamp: new Date().toISOString()
400   };
401
402   setAxiomSet([...axiomSet, newAxiom]);
403
404   // Record evolution
405   setEvolutionHistory([
406     ...evolutionHistory,
407     {
408       timestamp: new Date().toISOString(),
409       trigger: contradiction.issue,
410       oldState: axiomSet.length + ' axioms',
411       newAxiom: refinedAxiom.refined_axiom,
412       synthesis: refinedAxiom.integration_strategy,
413       context: refinedAxiom.context
414     }
415   ]);
416
417   alert('Axiom integrated! Try reprocessing the text to see how the interpretation changes.');
418
419 } catch (error) {
420   console.error("Error in accommodation:", error);
421   alert('Error during sublation: ' + error.message);
422 }
423
424   setIsProcessing(false);
425 };
426
427 // Export functions
428 const copyInterpretation = () => {
429   if (!interpretation) return;
430
431   const exportText = `
432 === PML PHENOMENOLOGICAL READING ===
433 Text: ${interpretation.original}
434 Iteration: ${interpretation.iterationDepth}
435 Timestamp: ${new Date(interpretation.timestamp).toLocaleString()}
436
437 READING EXPERIENCE:
438 ${interpretation.pml.reading_experience}
439
440 PML FORMALIZATIONS:
441 ${interpretation.pml.formalizations.map(f => `
442 ${f.step.toUpperCase()} ${f.temporal_moment} ${f.inference_type || 'material'}`)}

```

```

443     PML: ${f.pml}
444     ${f.explanation}
445   `).join('\n')}
446
447 INTERPRETATION:
448 ${interpretation.logic.interpretation}
449
450 KEY INSIGHTS:
451 ${interpretation.logic.key_insights?.map(i => ` - ${i}`).join('\n') || 'None'}
452
453 META-CRITIQUE:
454 ${interpretation.critique.diagnostic.meta_insight || 'None'}
455 `;
456
457   navigator.clipboard.writeText(exportText);
458   alert('Interpretation copied to clipboard!');
459 };
460
461 const exportAxiomsAsProlog = () => {
462   const prologCode =
463   `=====
464   %% PML Axioms – Exported from Dialectical Interpreter
465   %% Generated: ${new Date().toLocaleString()}
466   %% Total Axioms: ${axiomSet.length}
467   %% Formalized Concepts: ${formalizedConcepts.join(',')} || 'None'
468   %% =====
469
470   :- module(evolved_axioms, []).
471   :- use_module(pml_operators).
472   :- multifile incompatibility_semantics:material_inference/3.
473
474   ${axiomSet.map(axiom =>
475     `%
476     %% ${axiom.context}
477     %% Source: ${axiom.source}, Type: ${axiom.type}
478     %% ${axiom.source === 'evolved' ? `%% Added: ${new Date(axiom.timestamp).toLocaleString()}` : ''}
479     %% ${axiom.rationale ? `%% Rationale: ${axiom.rationale}` : ''}
480     %% ${axiom.addresses ? `%% Addresses: ${axiom.addresses}` : ''}
481     ${convertToPrologAxiom(axiom.content)}`).join('\n\n')}
482 `;
483
484   navigator.clipboard.writeText(prologCode);
485   alert('Prolog axioms copied to clipboard! Save as evolved_axioms.pl and load after
486       ↪ semantic_axioms.');
487 };
488
489 const convertToPrologAxiom = (axiomContent) => {
490   // Parse simple axiom syntax and convert to Prolog material_inference/3
491   const match = axiomContent.match(/^.*?\s*=>\s*(.*)$/);
492   if (!match) return `%% Could not parse: ${axiomContent}`;
493
494   const antecedent = match[1].trim();
495   const consequent = match[2].trim();
496
497   return `incompatibility_semantics:material_inference([${antecedent}], ${consequent}, true).`;
498 };
499
500 const autoAddAxiom = async (proposedAxiom) => {
501   // Automatically add an axiom without going through accommodation flow
502   const newAxiom = {
503     id: `auto_${Date.now()}`,
504     content: proposedAxiom.proposed,
505     source: 'user_suggested',
506     type: 'material',
507     rationale: proposedAxiom.rationale,
508     context: proposedAxiom.rationale.substring(0, 100), // First 100 chars

```

```

507     timestamp: new Date().toISOString()
508   };
509
510   setAxiomSet([...axiomSet, newAxiom]);
511
512   setEvolutionHistory([
513     ...evolutionHistory,
514     {
515       timestamp: new Date().toISOString(),
516       trigger: 'User suggestion',
517       oldState: axiomSet.length + ' axioms',
518       newAxiom: proposedAxiom.proposed,
519       synthesis: 'Direct user addition',
520       context: proposedAxiom.rationale
521     }
522   ]);
523
524   alert('Axiom added! Try reprocessing text to see the effect.');
525 };
526
527 const exampleTexts = [
528   {
529     name: "Hegel - Being/Nothing",
530     text: "Being, pure being, without any further determination. In its indeterminate immediacy it is
531     ↪ equal only to itself. It is also not unequal relatively to an other; it has no diversity
532     ↪ within itself nor any with a reference outwards. Pure being is in fact nothing, and neither
533     ↪ more nor less than nothing."
534   },
535   {
536     name: "Hegel - Self-Consciousness",
537     text: "Self-consciousness exists in and for itself when, and by the fact that, it so exists for
538     ↪ another; that is, it exists only in being acknowledged."
539   },
540   {
541     name: "Hegel - Master/Slave",
542     text: "The master relates himself to the bondsman meditately through independent being, for that is
543     ↪ precisely what keeps the bondsman in thrall; it is his chain, from which he could not in the
544     ↪ struggle get away, and for that reason he proved himself to be dependent, to have his
545     ↪ independence in the shape of thinghood."
546   }
547 ];
548
549 return (
550   <div className="min-h-screen bg-gradient-to-br from-slate-900 via-purple-900 to-slate-900 text-white
551     ↪ p-6">
552     <div className="max-w-6xl mx-auto">
553       {/* Header */}
554       <div className="mb-8">
555         <h1 className="text-4xl font-bold mb-2 bg-gradient-to-r from-purple-400 to-pink-400
556           ↪ bg-clip-text text-transparent">
557           Dialectical Interpreter
558         </h1>
559         <p className="text-purple-300">
560           A self-evolving PML system for philosophical text analysis
561         </p>
562       </div>
563     </div>
564
565     {/* Main Input Section */}
566     <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6 mb-6">
567       <div className="mb-4">
568         <label className="block text-sm font-medium mb-2">Philosophical Text</label>
569         <textarea
570           value={inputText}
571           onChange={(e) => setInputText(e.target.value)}>
572         </textarea>
573       </div>
574     </div>
575   
```

```

562         className="w-full h-40 bg-black/30 border border-purple-500/30 rounded-lg p-4 text-white
563         ↵ placeholder-purple-300/50 focus:outline-none focus:border-purple-400 focus:ring-2
564         ↵ focus:ring-purple-400/20"
565         placeholder="Paste a philosophical passage here (e.g., from Hegel's Phenomenology)...""
566     />
567 </div>

568 <div className="flex flex-wrap gap-2 mb-4">
569     <span className="text-sm text-purple-300">Examples:</span>
570     {exampleTexts.map((ex, i) => (
571         <button
572             key={i}
573             onClick={() => setInputText(ex.text)}
574             className="text-xs bg-purple-500/20 hover:bg-purple-500/30 px-3 py-1 rounded-full
575             ↵ transition-colors"
576             >
577                 {ex.name}
578             </button>
579         )))
580     </div>

581 <div className="flex gap-3">
582     <button
583         onClick={processText}
584         disabled={!inputText || isProcessing}
585         className="flex-1 bg-gradient-to-r from-purple-500 to-pink-500 hover:from-purple-600
586         ↵ hover:to-pink-600 disabled:from-gray-500 disabled:to-gray-600 px-6 py-3 rounded-lg
587         ↵ font-medium transition-all flex items-center justify-center gap-2"
588         >
589             {isProcessing ? (
590                 <>
591                     <Loader2 className="w-5 h-5 animate-spin" />
592                     Processing...
593                 </>
594             ) : (
595                 <>
596                     <Sparkles className="w-5 h-5" />
597                     {iterationDepth > 0 && inputText === interpretation?.original
598                         ? `Re-read (Iteration ${iterationDepth + 1})`:
599                         'Interpret Text'}
600                     </>
601             )
602         </button>
603         <button
604             onClick={() => setShowEvolutionPanel(!showEvolutionPanel)}
605             className="bg-purple-500/20 hover:bg-purple-500/30 px-6 py-3 rounded-lg font-medium
606             ↵ transition-colors flex items-center gap-2"
607             >
608                 <GitBranch className="w-5 h-5" />
609                 Logic ({axiomSet.length})
610             </button>
611     </div>

612     /* Iteration depth indicator */
613     {iterationDepth > 0 && (
614         <div className="mt-3 bg-blue-500/20 border border-blue-400/30 rounded-lg p-3">
615             <p className="text-sm text-blue-200">
616                 <strong>Iteration {iterationDepth}</strong> -
617                 {formalizedConcepts.length > 0
618                     ? ` ${formalizedConcepts.length} concepts formalized as structural scaffolding`:
619                     ' First reading - all material inference'}
620                 </p>
621             {formalizedConcepts.length > 0 && (
622                 <p className="text-xs text-blue-300 mt-1">
```

```

621           Formalized: {formalizedConcepts.join(', ')}

622           </p>
623       )
624     </div>
625   )
626 </div>

627   {/* Evolution Panel */}
628   {showEvolutionPanel && (
629     <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6 mb-6">
630       <div className="flex items-center justify-between mb-4">
631         <h2 className="text-xl font-bold flex items-center gap-2">
632           <GitBranch className="w-5 h-5" />
633           Axiom Evolution
634         </h2>
635         <button
636           onClick={exportAxiomsAsProlog}
637           className="bg-blue-500/30 hover:bg-blue-500/40 px-4 py-2 rounded text-sm
638             transition-colors"
639         >
640           Export as Prolog
641         </button>
642       </div>
643
644     <div className="space-y-3 mb-6">
645       {axiomSet.map((axiom) => (
646         <div
647           key={axiom.id}
648           className={`p-4 rounded-lg ${(
649             axiom.source === 'core'
650               ? 'bg-blue-500/20 border border-blue-500/30'
651               : 'bg-green-500/20 border border-green-500/30'
652             )}`}
653         >
654           <div className="flex items-start justify-between">
655             <div className="flex-1">
656               <div className="flex items-center gap-2 mb-2">
657                 <code className="text-sm font-mono text-purple-200">{axiom.content}</code>
658               </div>
659               <p className="text-xs text-purple-300 mb-1">{axiom.context}</p>
660               {axiom.rationale && (
661                 <p className="text-xs text-purple-300 mb-1">Rationale: {axiom.rationale}</p>
662               )}
663               {axiom.addresses && (
664                 <p className="text-xs text-green-300 mt-1">Addresses: {axiom.addresses}</p>
665               )}
666             </div>
667             <div className="flex flex-col gap-1 ml-3">
668               <span className={`text-xs px-2 py-1 rounded ${
669                 axiom.source === 'core' ? 'bg-blue-500/30' : 'bg-green-500/30'
670               )`}>
671                 {axiom.source}
672               </span>
673               <span className={`text-xs px-2 py-1 rounded ${
674                 axiom.type === 'formal' ? 'bg-yellow-500/30' : 'bg-purple-500/30'
675               )`}>
676                 {axiom.type}
677               </span>
678             </div>
679           </div>
680         </div>
681       ))}
682     </div>
683
684   {evolutionHistory.length > 0 && (

```

```

685     <>
686     <h3 className="text-lg font-bold mb-3">Evolution History</h3>
687     <div className="space-y-2">
688       {evolutionHistory.map((event, i) => (
689         <div key={i} className="bg-black/30 p-3 rounded-lg text-sm">
690           <div className="flex items-center gap-2 mb-1">
691             <CheckCircle className="w-4 h-4 text-green-400" />
692             <span className="text-purple-300">{new
693               ↪ Date(event.timestamp).toLocaleTimeString()}</span>
694           </div>
695           <p className="text-yellow-300 mb-1">Trigger: {event.trigger}</p>
696           <p className="text-green-300">New Axiom: <code>{event.newAxiom}</code></p>
697           <p className="text-purple-200 text-xs mt-1">{event.synthesis}</p>
698         </div>
699       )))
700     </div>
701   )
702 </div>
703 }

704 /* Results Section */
705 {interpretation && !interpretation.error && (
706   <div className="space-y-6">
707     /* Export Controls */
708     <div className="bg-gradient-to-r from-green-500/20 to-blue-500/20 backdrop-blur-lg
709       rounded-lg p-4 border border-green-400/30">
710       <div className="flex items-center justify-between">
711         <div>
712           <h3 className="font-semibold text-green-300 mb-1">Export Analysis</h3>
713           <p className="text-xs text-green-200">
714             Copy interpretation or export evolved axioms as Prolog code
715           </p>
716         </div>
717         <div className="flex gap-2">
718           <button
719             onClick={copyInterpretation}
720             className="bg-green-500/30 hover:bg-green-500/40 px-4 py-2 rounded transition-colors
721               ↪ text-sm"
722             >
723                Copy Reading
724             </button>
725           <button
726             onClick={exportAxiomsAsProlog}
727             className="bg-blue-500/30 hover:bg-blue-500/40 px-4 py-2 rounded transition-colors
728               ↪ text-sm"
729             >
730                Export Prolog
731             </button>
732         </div>
733       </div>
734     /* PML Formalization */
735     <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
736       <div className="mb-4">
737         <h2 className="text-xl font-bold mb-2">Phenomenological Reading (PML)</h2>
738         <p className="text-purple-300 text-sm italic">
739           Tracking the temporal, embodied experience of working through this text ↪
740         </p>
741         {interpretation.pml.reading_experience && (
742           <div className="mt-3 bg-purple-500/20 p-3 rounded-lg">
743             <p className="text-sm text-purple-100">{interpretation.pml.reading_experience}</p>
744           </div>
745         )}
746       </div>

```

```

746
747     <div className="space-y-3">
748       {interpretation.pml.formalizations.map((form, i) => (
749         <div key={i} className="bg-black/30 p-4 rounded-lg">
750           <div className="flex items-center gap-2 mb-2">
751             <span className="bg-purple-500/30 px-2 py-1 rounded text-xs font-medium">
752               {form.step}
753             </span>
754             {form.temporal_moment && (
755               <span className="bg-blue-500/30 px-2 py-1 rounded text-xs">
756                 {form.temporal_moment}
757               </span>
758             )}
759             {form.inference_type && (
760               <span className={`px-2 py-1 rounded text-xs ${form.inference_type === 'formal' ? 'bg-yellow-500/30 border border-yellow-400/30' : 'bg-green-500/30'}`}>
761                 {form.inference_type === 'formal' ? 'formal' : 'material'}
762               </span>
763             )}
764           </div>
765           <code className="text-purple-300 block mb-2">{form.pml}</code>
766           <p className="text-sm text-purple-200">{form.explanation}</p>
767         </div>
768       ))}
769     </div>
770   </div>
771
772   /* Logical Proof */
773   <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
774     <h2 className="text-xl font-bold mb-4">Proof Steps</h2>
775     <div className="space-y-2">
776       {interpretation.logic.proof_steps.map((step, i) => (
777         <div key={i} className="bg-black/30 p-3 rounded-lg">
778           <div className="flex items-start gap-3">
779             <span className="bg-blue-500/30 px-2 py-1 rounded text-xs font-mono shrink-0">
780               {i + 1}
781             </span>
782             <div className="flex-1">
783               <p className="text-sm text-blue-300 mb-1">
784                 Premises: {step.premises.join(', ')}
785               </p>
786               <p className="text-sm text-purple-300 mb-1">
787                 Axiom: <code>{step.axiom_used}</code>
788               </p>
789               <p className="text-sm text-green-300 mb-1">
790                 Conclusion: <code>{step.conclusion}</code>
791               </p>
792               <p className="text-xs text-purple-200">{step.explanation}</p>
793             </div>
794           </div>
795         </div>
796       ))}
797     </div>
798   </div>
799
800   /* Interpretation */
801   <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
802     <h2 className="text-xl font-bold mb-4">Interpretation</h2>
803     <p className="text-purple-100 mb-4">
804       {interpretation.logic.interpretation}</p>
805     <div>
806       {interpretation.logic.key_insights && interpretation.logic.key_insights.length > 0 && (
807         <div>
808           <div>
809             {interpretation.logic.key_insights}</div>
810           </div>
811         </div>
812       )}
813     </div>
814   </div>
815
816   /* Summary */
817   <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
818     <h2 className="text-xl font-bold mb-4">Summary</h2>
819     <p>
820       {interpretation.summary}</p>
821   </div>
822
823   /* Footer */
824   <div>
825     <div>
826       <div>
827         <div>
828           <div>
829             {navigation}</div>
830           </div>
831         </div>
832       </div>
833     </div>
834   </div>
835
836   /* Footer */
837   <div>
838     <div>
839       <div>
840         <div>
841           <div>
842             {navigation}</div>
843           </div>
844         </div>
845       </div>
846     </div>
847   </div>
848
849   /* Footer */
850   <div>
851     <div>
852       <div>
853         <div>
854           <div>
855             {navigation}</div>
856           </div>
857         </div>
858       </div>
859     </div>
860   </div>
861
862   /* Footer */
863   <div>
864     <div>
865       <div>
866         <div>
867           <div>
868             {navigation}</div>
869           </div>
870         </div>
871       </div>
872     </div>
873   </div>
874
875   /* Footer */
876   <div>
877     <div>
878       <div>
879         <div>
880           <div>
881             {navigation}</div>
882           </div>
883         </div>
884       </div>
885     </div>
886   </div>
887
888   /* Footer */
889   <div>
890     <div>
891       <div>
892         <div>
893           <div>
894             {navigation}</div>
895           </div>
896         </div>
897       </div>
898     </div>
899   </div>
900
901   /* Footer */
902   <div>
903     <div>
904       <div>
905         <div>
906           <div>
907             {navigation}</div>
908           </div>
909         </div>
910       </div>
911     </div>
912   </div>
913
914   /* Footer */
915   <div>
916     <div>
917       <div>
918         <div>
919           <div>
920             {navigation}</div>
921           </div>
922         </div>
923       </div>
924     </div>
925   </div>
926
927   /* Footer */
928   <div>
929     <div>
930       <div>
931         <div>
932           <div>
933             {navigation}</div>
934           </div>
935         </div>
936       </div>
937     </div>
938   </div>
939
940   /* Footer */
941   <div>
942     <div>
943       <div>
944         <div>
945           <div>
946             {navigation}</div>
947           </div>
948         </div>
949       </div>
950     </div>
951   </div>
952
953   /* Footer */
954   <div>
955     <div>
956       <div>
957         <div>
958           <div>
959             {navigation}</div>
960           </div>
961         </div>
962       </div>
963     </div>
964   </div>
965
966   /* Footer */
967   <div>
968     <div>
969       <div>
970         <div>
971           <div>
972             {navigation}</div>
973           </div>
974         </div>
975       </div>
976     </div>
977   </div>
978
979   /* Footer */
980   <div>
981     <div>
982       <div>
983         <div>
984           <div>
985             {navigation}</div>
986           </div>
987         </div>
988       </div>
989     </div>
990   </div>
991
992   /* Footer */
993   <div>
994     <div>
995       <div>
996         <div>
997           <div>
998             {navigation}</div>
999           </div>
1000          </div>
1001        </div>
1002      </div>
1003    </div>
1004
1005    /* Footer */
1006    <div>
1007      <div>
1008        <div>
1009          <div>
1010            <div>
1011              {navigation}</div>
1012            </div>
1013          </div>
1014        </div>
1015      </div>
1016    </div>
1017
1018    /* Footer */
1019    <div>
1020      <div>
1021        <div>
1022          <div>
1023            <div>
1024              {navigation}</div>
1025            </div>
1026          </div>
1027        </div>
1028      </div>
1029    </div>
1030
1031    /* Footer */
1032    <div>
1033      <div>
1034        <div>
1035          <div>
1036            <div>
1037              {navigation}</div>
1038            </div>
1039          </div>
1040        </div>
1041      </div>
1042    </div>
1043
1044    /* Footer */
1045    <div>
1046      <div>
1047        <div>
1048          <div>
1049            <div>
1050              {navigation}</div>
1051            </div>
1052          </div>
1053        </div>
1054      </div>
1055    </div>
1056
1057    /* Footer */
1058    <div>
1059      <div>
1060        <div>
1061          <div>
1062            <div>
1063              {navigation}</div>
1064            </div>
1065          </div>
1066        </div>
1067      </div>
1068    </div>
1069
1070    /* Footer */
1071    <div>
1072      <div>
1073        <div>
1074          <div>
1075            <div>
1076              {navigation}</div>
1077            </div>
1078          </div>
1079        </div>
1080      </div>
1081    </div>
1082
1083    /* Footer */
1084    <div>
1085      <div>
1086        <div>
1087          <div>
1088            <div>
1089              {navigation}</div>
1090            </div>
1091          </div>
1092        </div>
1093      </div>
1094    </div>
1095
1096    /* Footer */
1097    <div>
1098      <div>
1099        <div>
1100          <div>
1101            <div>
1102              {navigation}</div>
1103            </div>
1104          </div>
1105        </div>
1106      </div>
1107    </div>
1108
1109    /* Footer */
1110    <div>
1111      <div>
1112        <div>
1113          <div>
1114            <div>
1115              {navigation}</div>
1116            </div>
1117          </div>
1118        </div>
1119      </div>
1120    </div>
1121
1122    /* Footer */
1123    <div>
1124      <div>
1125        <div>
1126          <div>
1127            <div>
1128              {navigation}</div>
1129            </div>
1130          </div>
1131        </div>
1132      </div>
1133    </div>
1134
1135    /* Footer */
1136    <div>
1137      <div>
1138        <div>
1139          <div>
1140            <div>
1141              {navigation}</div>
1142            </div>
1143          </div>
1144        </div>
1145      </div>
1146    </div>
1147
1148    /* Footer */
1149    <div>
1150      <div>
1151        <div>
1152          <div>
1153            <div>
1154              {navigation}</div>
1155            </div>
1156          </div>
1157        </div>
1158      </div>
1159    </div>
1160
1161    /* Footer */
1162    <div>
1163      <div>
1164        <div>
1165          <div>
1166            <div>
1167              {navigation}</div>
1168            </div>
1169          </div>
1170        </div>
1171      </div>
1172    </div>
1173
1174    /* Footer */
1175    <div>
1176      <div>
1177        <div>
1178          <div>
1179            <div>
1180              {navigation}</div>
1181            </div>
1182          </div>
1183        </div>
1184      </div>
1185    </div>
1186
1187    /* Footer */
1188    <div>
1189      <div>
1190        <div>
1191          <div>
1192            <div>
1193              {navigation}</div>
1194            </div>
1195          </div>
1196        </div>
1197      </div>
1198    </div>
1199
1200    /* Footer */
1201    <div>
1202      <div>
1203        <div>
1204          <div>
1205            <div>
1206              {navigation}</div>
1207            </div>
1208          </div>
1209        </div>
1210      </div>
1211    </div>
1212
1213    /* Footer */
1214    <div>
1215      <div>
1216        <div>
1217          <div>
1218            <div>
1219              {navigation}</div>
1220            </div>
1221          </div>
1222        </div>
1223      </div>
1224    </div>
1225
1226    /* Footer */
1227    <div>
1228      <div>
1229        <div>
1230          <div>
1231            <div>
1232              {navigation}</div>
1233            </div>
1234          </div>
1235        </div>
1236      </div>
1237    </div>
1238
1239    /* Footer */
1240    <div>
1241      <div>
1242        <div>
1243          <div>
1244            <div>
1245              {navigation}</div>
1246            </div>
1247          </div>
1248        </div>
1249      </div>
1250    </div>
1251
1252    /* Footer */
1253    <div>
1254      <div>
1255        <div>
1256          <div>
1257            <div>
1258              {navigation}</div>
1259            </div>
1260          </div>
1261        </div>
1262      </div>
1263    </div>
1264
1265    /* Footer */
1266    <div>
1267      <div>
1268        <div>
1269          <div>
1270            <div>
1271              {navigation}</div>
1272            </div>
1273          </div>
1274        </div>
1275      </div>
1276    </div>
1277
1278    /* Footer */
1279    <div>
1280      <div>
1281        <div>
1282          <div>
1283            <div>
1284              {navigation}</div>
1285            </div>
1286          </div>
1287        </div>
1288      </div>
1289    </div>
1290
1291    /* Footer */
1292    <div>
1293      <div>
1294        <div>
1295          <div>
1296            <div>
1297              {navigation}</div>
1298            </div>
1299          </div>
1300        </div>
1301      </div>
1302    </div>
1303
1304    /* Footer */
1305    <div>
1306      <div>
1307        <div>
1308          <div>
1309            <div>
1310              {navigation}</div>
1311            </div>
1312          </div>
1313        </div>
1314      </div>
1315    </div>
1316
1317    /* Footer */
1318    <div>
1319      <div>
1320        <div>
1321          <div>
1322            <div>
1323              {navigation}</div>
1324            </div>
1325          </div>
1326        </div>
1327      </div>
1328    </div>
1329
1330    /* Footer */
1331    <div>
1332      <div>
1333        <div>
1334          <div>
1335            <div>
1336              {navigation}</div>
1337            </div>
1338          </div>
1339        </div>
1340      </div>
1341    </div>
1342
1343    /* Footer */
1344    <div>
1345      <div>
1346        <div>
1347          <div>
1348            <div>
1349              {navigation}</div>
1350            </div>
1351          </div>
1352        </div>
1353      </div>
1354    </div>
1355
1356    /* Footer */
1357    <div>
1358      <div>
1359        <div>
1360          <div>
1361            <div>
1362              {navigation}</div>
1363            </div>
1364          </div>
1365        </div>
1366      </div>
1367    </div>
1368
1369    /* Footer */
1370    <div>
1371      <div>
1372        <div>
1373          <div>
1374            <div>
1375              {navigation}</div>
1376            </div>
1377          </div>
1378        </div>
1379      </div>
1380    </div>
1381
1382    /* Footer */
1383    <div>
1384      <div>
1385        <div>
1386          <div>
1387            <div>
1388              {navigation}</div>
1389            </div>
1390          </div>
1391        </div>
1392      </div>
1393    </div>
1394
1395    /* Footer */
1396    <div>
1397      <div>
1398        <div>
1399          <div>
1400            <div>
1401              {navigation}</div>
1402            </div>
1403          </div>
1404        </div>
1405      </div>
1406    </div>
1407
1408    /* Footer */
1409    <div>
1410      <div>
1411        <div>
1412          <div>
1413            <div>
1414              {navigation}</div>
1415            </div>
1416          </div>
1417        </div>
1418      </div>
1419    </div>
1420
1421    /* Footer */
1422    <div>
1423      <div>
1424        <div>
1425          <div>
1426            <div>
1427              {navigation}</div>
1428            </div>
1429          </div>
1430        </div>
1431      </div>
1432    </div>
1433
1434    /* Footer */
1435    <div>
1436      <div>
1437        <div>
1438          <div>
1439            <div>
1440              {navigation}</div>
1441            </div>
1442          </div>
1443        </div>
1444      </div>
1445    </div>
1446
1447    /* Footer */
1448    <div>
1449      <div>
1450        <div>
1451          <div>
1452            <div>
1453              {navigation}</div>
1454            </div>
1455          </div>
1456        </div>
1457      </div>
1458    </div>
1459
1460    /* Footer */
1461    <div>
1462      <div>
1463        <div>
1464          <div>
1465            <div>
1466              {navigation}</div>
1467            </div>
1468          </div>
1469        </div>
1470      </div>
1471    </div>
1472
1473    /* Footer */
1474    <div>
1475      <div>
1476        <div>
1477          <div>
1478            <div>
1479              {navigation}</div>
1480            </div>
1481          </div>
1482        </div>
1483      </div>
1484    </div>
1485
1486    /* Footer */
1487    <div>
1488      <div>
1489        <div>
1490          <div>
1491            <div>
1492              {navigation}</div>
1493            </div>
1494          </div>
1495        </div>
1496      </div>
1497    </div>
1498
1499    /* Footer */
1500    <div>
1501      <div>
1502        <div>
1503          <div>
1504            <div>
1505              {navigation}</div>
1506            </div>
1507          </div>
1508        </div>
1509      </div>
1510    </div>
1511
1512    /* Footer */
1513    <div>
1514      <div>
1515        <div>
1516          <div>
1517            <div>
1518              {navigation}</div>
1519            </div>
1520          </div>
1521        </div>
1522      </div>
1523    </div>
1524
1525    /* Footer */
1526    <div>
1527      <div>
1528        <div>
1529          <div>
1530            <div>
1531              {navigation}</div>
1532            </div>
1533          </div>
1534        </div>
1535      </div>
1536    </div>
1537
1538    /* Footer */
1539    <div>
1540      <div>
1541        <div>
1542          <div>
1543            <div>
1544              {navigation}</div>
1545            </div>
1546          </div>
1547        </div>
1548      </div>
1549    </div>
1550
1551    /* Footer */
1552    <div>
1553      <div>
1554        <div>
1555          <div>
1556            <div>
1557              {navigation}</div>
1558            </div>
1559          </div>
1560        </div>
1561      </div>
1562    </div>
1563
1564    /* Footer */
1565    <div>
1566      <div>
1567        <div>
1568          <div>
1569            <div>
1570              {navigation}</div>
1571            </div>
1572          </div>
1573        </div>
1574      </div>
1575    </div>
1576
1577    /* Footer */
1578    <div>
1579      <div>
1580        <div>
1581          <div>
1582            <div>
1583              {navigation}</div>
1584            </div>
1585          </div>
1586        </div>
1587      </div>
1588    </div>
1589
1590    /* Footer */
1591    <div>
1592      <div>
1593        <div>
1594          <div>
1595            <div>
1596              {navigation}</div>
1597            </div>
1598          </div>
1599        </div>
1600      </div>
1601    </div>
1602
1603    /* Footer */
1604    <div>
1605      <div>
1606        <div>
1607          <div>
1608            <div>
1609              {navigation}</div>
1610            </div>
1611          </div>
1612        </div>
1613      </div>
1614    </div>
1615
1616    /* Footer */
1617    <div>
1618      <div>
1619        <div>
1620          <div>
1621            <div>
1622              {navigation}</div>
1623            </div>
1624          </div>
1625        </div>
1626      </div>
1627    </div>
1628
1629    /* Footer */
1630    <div>
1631      <div>
1632        <div>
1633          <div>
1634            <div>
1635              {navigation}</div>
1636            </div>
1637          </div>
1638        </div>
1639      </div>
1640    </div>
1641
1642    /* Footer */
1643    <div>
1644      <div>
1645        <div>
1646          <div>
1647            <div>
1648              {navigation}</div>
1649            </div>
1650          </div>
1651        </div>
1652      </div>
1653    </div>
1654
1655    /* Footer */
1656    <div>
1657      <div>
1658        <div>
1659          <div>
1660            <div>
1661              {navigation}</div>
1662            </div>
1663          </div>
1664        </div>
1665      </div>
1666    </div>
1667
1668    /* Footer */
1669    <div>
1670      <div>
1671        <div>
1672          <div>
1673            <div>
1674              {navigation}</div>
1675            </div>
1676          </div>
1677        </div>
1678      </div>
1679    </div>
1680
1681    /* Footer */
1682    <div>
1683      <div>
1684        <div>
1685          <div>
1686            <div>
1687              {navigation}</div>
1688            </div>
1689          </div>
1690        </div>
1691      </div>
1692    </div>
1693
1694    /* Footer */
1695    <div>
1696      <div>
1697        <div>
1698          <div>
1699            <div>
1700              {navigation}</div>
1701            </div>
1702          </div>
1703        </div>
1704      </div>
1705    </div>
1706
1707    /* Footer */
1708    <div>
1709      <div>
1710        <div>
1711          <div>
1712            <div>
1713              {navigation}</div>
1714            </div>
1715          </div>
1716        </div>
1717      </div>
1718    </div>
1719
1720    /* Footer */
1721    <div>
1722      <div>
1723        <div>
1724          <div>
1725            <div>
1726              {navigation}</div>
1727            </div>
1728          </div>
1729        </div>
1730      </div>
1731    </div>
1732
1733    /* Footer */
1734    <div>
1735      <div>
1736        <div>
1737          <div>
1738            <div>
1739              {navigation}</div>
1740            </div>
1741          </div>
1742        </div>
1743      </div>
1744    </div>
1745
1746    /* Footer */
1747    <div>
1748      <div>
1749        <div>
1750          <div>
1751            <div>
1752              {navigation}</div>
1753            </div>
1754          </div>
1755        </div>
1756      </div>
1757    </div>
1758
1759    /* Footer */
1760    <div>
1761      <div>
1762        <div>
1763          <div>
1764            <div>
1765              {navigation}</div>
1766            </div>
1767          </div>
1768        </div>
1769      </div>
1770    </div>
1771
1772    /* Footer */
1773    <div>
1774      <div>
1775        <div>
1776          <div>
1777            <div>
1778              {navigation}</div>
1779            </div>
1780          &
```

```

810     <div className="mt-4">
811         <h3 className="font-semibold mb-2">Key Insights:</h3>
812         <ul className="space-y-1">
813             {interpretation.logic.key_insights.map((insight, i) => (
814                 <li key={i} className="text-sm text-purple-200 flex items-start gap-2">
815                     <CheckCircle className="w-4 h-4 text-green-400 mt-0.5 shrink-0" />
816                     {insight}
817                 </li>
818             )))
819         </ul>
820     </div>
821   )
822 </div>
823
824 /* Critique & Evolution */
825 <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
826   <h2 className="text-xl font-bold mb-4 flex items-center gap-2">
827     <AlertCircle className="w-5 h-5" />
828     Meta-Critique: Levels of Analysis
829   </h2>
830
831 /* Meta-Insight */
832 {interpretation.critique.diagnostic.meta_insight && (
833   <div className="mb-6 bg-gradient-to-r from-purple-500/20 to-pink-500/20 p-4 rounded-lg
834   ↪ border border-purple-400/30">
835     <h3 className="font-semibold mb-2 text-purple-300">Meta-Insight:</h3>
836     <p className="text-sm
837   ↪ text-purple-100">{interpretation.critique.diagnostic.meta_insight}</p>
838   </div>
839 })
840
841 /* Established Readings */
842 <div className="mb-6">
843   <h3 className="font-semibold mb-3">Established Scholarly Readings:</h3>
844   <div className="space-y-2">
845     {interpretation.critique.established_readings.map((reading, i) => (
846       <div key={i} className="bg-blue-500/20 p-3 rounded-lg">
847         <div className="flex items-center gap-2 mb-1">
848           <p className="text-sm font-medium text-blue-300">{reading.scholar}</p>
849           {reading.level && (
850             <span className="text-xs bg-blue-500/30 px-2 py-0.5 rounded">
851               {reading.level}
852             </span>
853           )}
854         </div>
855         <p className="text-sm text-blue-200">{reading.interpretation}</p>
856       </div>
857     )))
858   </div>
859 }
860 /* Level Distinctions */
861 {interpretation.critique.alignment.level_distinctions &&
862   interpretation.critique.alignment.level_distinctions.length > 0 && (
863   <div className="mb-6">
864     <h3 className="font-semibold mb-3 text-green-400">
865       Level Distinctions (Not Contradictions):
866     </h3>
867     <div className="space-y-2">
868       {interpretation.critique.alignment.level_distinctions.map((dist, i) => (
869         <div key={i} className="bg-green-500/20 p-3 rounded-lg border
870         ↪ border-green-500/30">
871           <p className="text-sm text-yellow-300 mb-1">
872             Apparent: {dist.apparent_contradiction}
873           </p>
874         </div>
875       )))
876     </div>
877   </div>
878 }

```

```
872         <p className="text-sm text-green-300 mb-1">
873             Actually: {dist.actually}
874         </p>
875         <p className="text-xs text-green-200">{dist.explanation}</p>
876     </div>
877   </div>
878 </div>
879 )
880
881
882 /* Genuine Contradictions */
883 {interpretation.critique.alignment.genuine_contradictions &&
884 interpretation.critique.alignment.genuine_contradictions.length > 0 && (
885   <div className="mb-6">
886     <p className="text-sm text-red-400 mb-2">⚠️ Genuine Contradictions (Same Level):</p>
887     <div className="space-y-3">
888       {interpretation.critique.alignment.genuine_contradictions.map((contra, i) => (
889         <div
890           key={i}
891           className={`p-4 rounded-lg ${{
892             contra.severity === 'high' ? 'bg-red-500/20 border border-red-500/30' :
893             contra.severity === 'medium' ? 'bg-yellow-500/20 border border-yellow-500/30' :
894               'bg-orange-500/20 border border-orange-500/30'
895           }}`}
896         >
897           <p className="font-medium mb-2">{contra.issue}</p>
898           <p className="text-sm text-red-200 mb-1">Our claim: {contra.our_claim}</p>
899           <p className="text-sm text-yellow-200 mb-3">Standard:
900             ↳ {contra.standard_claim}</p>
901
902           {interpretation.critique.diagnostic.needed_axioms
903             .filter(ax => ax.addresses === contra.issue)
904             .map((axiom, j) => (
905               <div key={j} className="mt-3 bg-black/30 p-3 rounded">
906                 <p className="text-sm text-purple-300 mb-2">Proposed Resolution:</p>
907                 <code className="text-xs text-green-300 block
908                   ↳ mb-2">{axiom.proposed}</code>
909                 <p className="text-xs text-purple-200 mb-3">{axiom.rationale}</p>
910                 <div className="flex gap-2">
911                   <button
912                     onClick={() => accommodateContradiction(contra, axiom)}
913                     disabled={isProcessing}
914                     className="bg-green-500/30 hover:bg-green-500/40 px-4 py-2 rounded
915                       ↳ text-sm transition-colors disabled:opacity-50"
916                   >
917                     ⚡ Refine & Evolve
918                   </button>
919                   <button
920                     onClick={() => autoAddAxiom(axiom)}
921                     disabled={isProcessing}
922                     className="bg-blue-500/30 hover:bg-blue-500/40 px-4 py-2 rounded
923                       ↳ text-sm transition-colors disabled:opacity-50"
924                   >
925                     ⚡ Quick Add
926                   </button>
927                 </div>
928               </div>
929             ))})
930           </div>
931         </div>
932       </div>
933     )}
```

```

932     {/* Pathology Detection */}
933     {interpretation.critique.diagnostic.pathology_detected !== 'none' && (
934       <div className="bg-red-500/20 border border-red-500/30 p-4 rounded-lg">
935         <h3 className="font-semibold mb-2 flex items-center gap-2">
936           <XCircle className="w-5 h-5" />
937           Pathology Detected: {interpretation.critique.diagnostic.pathology_detected}
938         </h3>
939         <p className="text-sm text-red-200">
940           The current axiom set may be generating a pathological pattern.
941           Consider accepting the proposed axioms to achieve sublation.
942         </p>
943       </div>
944     )}
945   </div>

946   {/* Follow-up Conversation */}
947   {showFollowUp && (
948     <div className="bg-white/10 backdrop-blur-lg rounded-lg p-6">
949       <h2 className="text-xl font-bold mb-4">Dialectical Refinement</h2>
950       <p className="text-purple-300 text-sm mb-4">
951         Clarify the interpretation, ask about specific moves, or challenge the framing
952       </p>
953
954       {/* Previous follow-ups */}
955       {interpretation.followUps && interpretation.followUps.length > 0 && (
956         <div className="mb-4 space-y-3">
957           {interpretation.followUps.map((fu, i) => (
958             <div key={i} className="space-y-2">
959               <div className="bg-blue-500/20 p-3 rounded-lg">
960                 <p className="text-sm font-medium text-blue-300 mb-1">You asked:</p>
961                 <p className="text-sm text-blue-100">{fu.question}</p>
962               </div>
963               <div className="bg-purple-500/20 p-3 rounded-lg">
964                 <p className="text-sm font-medium text-purple-300 mb-1">Response:</p>
965                 <p className="text-sm text-purple-100 whitespace-pre-wrap">{fu.response}</p>
966               </div>
967             )));
968           </div>
969         )})
970       </div>
971     )
972
973     {/* New follow-up input */}
974     <div className="flex gap-3">
975       <textarea
976         value={followUpQuestion}
977         onChange={(e) => setFollowUpQuestion(e.target.value)}
978         placeholder="e.g., 'But isn't the temporal reading inconsistent with Hegel's claim
979           ↪ that logic is atemporal?' or 'What about the role of negation here?''"
980         className="flex-1 bg-black/30 border border-purple-500/30 rounded-lg p-3 text-white
981           ↪ placeholder-purple-300/50 focus:outline-none focus:border-purple-400
982           ↪ focus:ring-2 focus:ring-purple-400/20 min-h-[80px]"
983         disabled={isProcessing}
984       />
985       <button
986         onClick={handleFollowUp}
987         disabled={!followUpQuestion.trim() || isProcessing}
988         className="bg-gradient-to-r from-purple-500 to-pink-500 hover:from-purple-600
989           ↪ hover:to-pink-600 disabled:from-gray-500 disabled:to-gray-600 px-6 rounded-lg
990           ↪ font-medium transition-all self-end"
991       >
992         {isProcessing ? <Loader2 className="w-5 h-5 animate-spin" /> : 'Ask'}
993       </button>
994     </div>
995   </div>
996 }

```

```
992     </div>
993   )}
994
995   {interpretation && interpretation.error && (
996     <div className="bg-red-500/20 border border-red-500/30 rounded-lg p-6">
997       <h2 className="text-xl font-bold mb-2 flex items-center gap-2">
998         <XCircle className="w-5 h-5" />
999         Error
1000       </h2>
1001       <p className="text-red-200">{interpretation.message}</p>
1002     </div>
1003   )}
1004   </div>
1005 </div>
1006 );
1007
1008
1009 export default DialecticalInterpreter;
1010
```

## 2 src/main.jsx

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import DialecticalInterpreter from './DialecticalInterpreter';
4
5 ReactDOM.createRoot(document.getElementById('root')).render(
6   <React.StrictMode>
7     <DialecticalInterpreter />
8     </React.StrictMode>
9   );
10
```