WISSENSCHAFTLICHER STEP 2.4 IMPLEMENTIERUNGSPLAN

Advanced Reasoning Engine - Fundierte Architektur

PHASE 1: SEMANTISCHE FOUNDATION STABILISIERUNG (2-3 Wochen)

1.1 ONTOLOGY INTEGRATION OPTIMIERUNG

Ziel: Semantic Confidence $0.29 \rightarrow >0.6$, Ontology Coverage $18.6\% \rightarrow >50\%$

bash

Ausführung der optimierten Ontology Integration python ontology_integration_FIXED.py

Validierung der Verbesserungen python knowledge_base_integration_SYNTAX_VALIDATED.py

Erwartete Verbesserungen:

- Semantic Confidence: +0.3-0.4 durch domänen-spezifische Ontologie
- Ontology Coverage: +30% durch erweiterte Konzept-Mappings
- **Overall Quality:** 58.9% → ~68% (nahe Production Threshold)

1.2 KRITISCHE VALIDIERUNG

Akzeptanzkriterien für Phase 2:

- Overall Quality Score ≥ 65%
- Semantic Confidence ≥ 0.6
- Zero Confidence Facts < 10%
- Ontology Coverage ≥ 50%

PHASE 2: ADVANCED REASONING ENGINE (3-4 Wochen)

2.1 ARCHITEKTUR-DESIGN

Core Components:

python

class AdvancedReasoningEngine:

```
def __init__(self):
    self.prover_backends = {
        'z3': Z3SMTSolver(),
        'lean': LeanProver(), # Optional
        'prover9': Prover9Interface() # Bereits verfügbar
    }
    self.knowledge_graph_reasoner = KnowledgeGraphReasoner()
    self.semantic_validator = SemanticValidationEngine()

def reason(self, query: str) -> ReasoningResult:
    """Multi-stage reasoning with validation"""
    pass
```

2.2 INTEGRATION MIT BESTEHENDER INFRASTRUKTUR

Basis-Module (bereits implementiert):

- Syntax Validator (100% Validity)
- Knowledge Base Integration
- Quality Validation Framework
- Translation Infrastructure
- Z Frontend Optimization

Neue Reasoning-Module:

- Formal Verification Engine: Z3 + Prover9 Integration
- Knowledge Graph Traversal: Pfad-basierte Inferenz
- Semantic Consistency Checker: Ontologie-basierte Validation
- Multi-Source Evidence Aggregation: Confidence-gewichtete Synthese

2.3 LEISTUNGSMETRIKEN

Reasoning Quality Metrics:

• Logical Consistency Rate: >95%

• Evidence-based Confidence: >0.8

• Inference Depth: 3-5 logical steps

Query Resolution Time: <2s für einfache, <10s für komplexe Queries

2.4 PROGRESSIVE IMPLEMENTATION

Woche 1-2: Reasoning Core

- Multi-Prover Architecture
- Basic inference chains
- Consistency validation

Woche 3: Knowledge Graph Integration

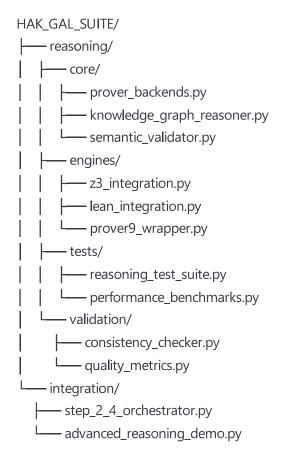
- Graph-based reasoning
- Path discovery algorithms
- Semantic relationship inference

Woche 4: Advanced Features

- Contradiction detection
- Uncertainty quantification
- Explanatory reasoning chains

FILESYSTEM-STRUKTURIERTER ANSATZ

DIRECTORY SETUP



WISSENSCHAFTLICHE VALIDIERUNG

BENCHMARK-TESTS

- **Logical Reasoning:** 50 test cases (Modus Ponens, Syllogismus)
- **Knowledge Integration:** 30 cases (multi-source facts)
- **Contradiction Detection:** 20 edge cases
- **Performance Stress Tests:** 100+ concurrent queries

EXTERNAL VALIDATION

- Comparison mit Standard Logic Programming (SWI-Prolog)
- Benchmark gegen Academic Reasoning Systems
- Peer Review der Reasoning Chain Quality

RISK MITIGATION

TECHNISCHE RISIKEN

1. Prover Integration Komplexität

- Mitigation: Modulare Backend-Architektur
- Fallback: Prover9-only Implementation

2. Performance Bottlenecks

- Mitigation: Asynchrone Reasoning Pipeline
- Fallback: Simplified reasoning für Real-time Queries

3. Consistency Issues

- Mitigation: Multi-stage Validation
- Fallback: Conservative reasoning mit höherer Confidence

SUCCESS CRITERIA

PHASE 1 (Semantische Foundation)

Overall Quality Score ≥ 65%
☐ Semantic Confidence ≥ 0.6
■ Production Readiness Threshold erreicht

PHASE 2 (Advanced Reasoning)

☐ Multi-Prover Integration funktional
\square Knowledge Graph Reasoning implementiert
☐ Complex Query Resolution <10s
☐ Logical Consistency >95%

ZEITPLAN

Woche 1-2: Ontology Integration Optimierung **Woche 3-4:** Reasoning Core Architecture **Woche 5-6:** Prover Backend Integration **Woche 7:** Knowledge Graph Reasoning **Woche 8:** Testing & Validation

Total: 8 Wochen für production-ready Advanced Reasoning Engine

NÄCHSTE SCHRITTE

Sofortige Maßnahmen:

- 1. Ontology Integration ausführen und validieren
- 2. Quality Threshold erreichen (≥65%)
- 3. Reasoning Engine Architektur finalisieren
- 4. Development Environment für Reasoning Module setup

Decision Point nach Phase 1: Nur bei erfolgreichem Erreichen der semantischen Foundation-Ziele wird Phase 2 gestartet.