Integration of the Discovery-Invention Spectrum into the UIL Framework

1 Overview

We aim to create a unified framework that:

- Combines the knowledge generation processes (discovery and invention) with UIL's knowledge evolution and interaction mechanisms.
- Incorporates confidence measures and security levels into UIL's mathematical structures.
- Enhances the existing Nibbler Algorithm and graph representations to accommodate new functionalities.

2 Mathematical Integration

2.1 Extending the UIL Framework

Definitions:

Let $G = (V, E)$ be the directed graph representing the knowledge structure in UIL.

- V: Set of nodes (vertices) representing knowledge states or linguistic units.
- E: Set of edges representing interactions or transformations between nodes.

We introduce additional functions and measures:

- **Discovery Function** Δ: Captures knowledge deduced within the existing graph.
- **Invention Function** Φ: Represents the introduction of new knowledge based on external inputs.
- **Confidence Measures** $c_d(\omega)$ and $c_i(\omega)$: Quantify the reliability of discovered and invented knowledge.
- **Security Levels** $\lambda(\omega)$: Assign security attributes to knowledge elements.

2.2 Formal Definitions

2.2.1 Discovery Function Δ

Function: $\Delta : G \to \Omega_d$

- G: Current knowledge graph.
- Ω_d: Set of discoverable knowledge objects within G.

Condition for Discovery:

$$\Delta(G) = \{\omega \in \Omega_d \mid \exists \text{ a proof path } p \subseteq G \text{ to } \omega\} \tag{1}$$

Proof Path p: A sequence of nodes and edges in G leading to ω.

2.2.2 Invention Function Φ

Function: $\Phi : I \times G \to \Omega_i$

- I: Set of external information inputs.
- Ω_i: Set of invented knowledge objects.

Result of Invention:

$$\Phi(i, G) = \omega_i \quad \text{where } \omega_i \notin V \text{ prior to the invention} \tag{2}$$

2.2.3 Confidence Measures

For Discovery:

$$c_d(\omega) = \begin{cases} 1, & \text{if } \omega \text{ is deducible via a valid proof path in } G \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

For Invention:

$$c_i(\omega_i) = f(v(\omega_i), s(\omega_i)) \tag{4}$$

- $v(\omega_i)$: Validation level of ω_i.
- $s(\omega_i)$: Source reliability.
- f: A function mapping v and s to a confidence score $[0, 1]$.

2.2.4 Security Levels $\lambda(\omega)$

Assignment: Each knowledge object ω is assigned a security level $\lambda(\omega) \in \Lambda$, where Λ is an ordered set of security levels.

2.3 Integration into Graph Structures

2.3.1 Augmented Node Representation

Node Attributes: Each node $v \in V$ is now represented as:

$$v = (\omega, c(\omega), \lambda(\omega)) \tag{5}$$

where:

- ω: Knowledge object.
- $c(\omega)$: Confidence measure ($c_d(\omega)$ or $c_i(\omega)$).
- $\lambda(\omega)$: Security level.

2.3.2 Edge Representation

Edges $e \in E$ may also carry attributes, such as:

- Transformation Type: Indicates whether the edge represents discovery or invention.
- Security Constraints: Ensures interactions comply with security policies.

2.4 Updated Nibbler Algorithm (NA)

Algorithm Adjustments:

- **Incorporate Confidence Measures**: During compression and propagation, the NA considers $c(\omega)$ to prioritize high-confidence knowledge.
- **Security Compliance**: The NA checks $\lambda(\omega)$ to ensure that knowledge propagation adheres to security constraints.

Differential Encoding Function with Confidence and Security: For nodes v_i and v_{i+1}:

$$\Delta(v_i, v_{i+1}) = \min(\text{information change between } v_i \text{ and } v_{i+1} \mid \lambda(v_{i+1}) \geq \lambda_{\min}) \tag{6}$$

where $\lambda_{\min}$ is the minimum required security level.

Compression Criteria:

- **High Confidence**: Prioritize paths where $c(\omega)$ is high.
- **Security Levels**: Ensure $\lambda(v_{i+1})$ is appropriate for the propagation.

3 Formal Theorems and Proofs

3.1 Knowledge Compression with Confidence

Theorem 1: Confidence-Weighted Knowledge Compression

$$K = \sum_{i=1}^{n} c(\omega_i) \cdot \Delta(v_i, v_{i+1}) \tag{7}$$

The total compressed knowledge K is the sum of the weighted differential encodings, factoring in confidence measures.

3.2 Security-Constrained Path Optimization

Theorem 2: Secure Path Optimization

$$P_{\text{opt}}(v_i, v_j) = \min_{P(v_i \to v_j)} d(v_i, v_j) \quad \text{subject to } \lambda(v_k) \geq \lambda_{\min} \quad \forall v_k \in P \tag{8}$$

4 Operational Rules

4.1 Discovery Process

- Verify proof path existence.
- Confirm graph boundary adherence.
- Ensure $\lambda(\omega_d) \geq \min_{v \in p} \lambda(v)$.

Confidence Assignment: $c_d(\omega_d) = 1$ if all conditions are met.

4.2 Invention Process

- Validate external information source i.
- Create new graph boundary G_{ω_i}.
- Define shell boundaries and security domains.
- Compute confidence: $c_i(\omega_i) = f(v(\omega_i), s(\omega_i))$

Security Level Assignment: $\lambda(\omega_i) = \min(\lambda_{\text{source}}, \lambda_{\text{domain}})$

5 Conclusion

Integrating the Discovery-Invention Spectrum Framework into the UIL framework enriches the model by:

- Enhancing Knowledge Generation: Incorporating both discovery and invention processes.

- Improving Reliability: Using confidence measures to assess knowledge trustworthiness.
- Ensuring Security: Assigning and enforcing security levels throughout the knowledge graph.
- Strengthening Formal Foundations: Extending mathematical models to accommodate new functionalities.