Computability Hierarchies and Knowledge Management: Towards a Text2KB System for Formal Sciences

Claude 3.7 Sonnet

April 25, 2025

Abstract

This paper outlines a project for developing a comprehensive Text2KB (Text to Knowledge Base) system focused on computability theory and related hierarchies of definability. The system aims to extract structured knowledge from mathematical texts, organize it according to established hierarchies (arithmetic, hyperarithmetic, analytical, etc.), and integrate it with formal knowledge representation systems such as Cyc. We discuss the theoretical foundations, implementation approaches, and potential applications in mathematical knowledge management (MKM). Additionally, we examine the relevance of this work to questions of AI predictability and control, arguing that understanding hierarchies of decidability provides important context for interpreting claims about absolute undecidability in AI systems.

1 Introduction

The fields of computability theory, recursion theory, and set theory have established precise hierarchies that classify mathematical statements and sets according to their logical complexity [Soare, 1999, Simpson, 2009]. These hierarchies—arithmetic, hyperarithmetic, analytical, and beyond—form a map of the landscape of mathematical decidability and definability. Understanding this landscape has profound implications not only for pure mathematics but also for computer science, artificial intelligence, and philosophical questions about the limits of formal systems.

Despite the foundational importance of these hierarchies, knowledge about them remains fragmented across specialized texts, papers, and research communities. This fragmentation creates barriers to both learning and application, particularly for those approaching the field from adjacent disciplines or autodidactic backgrounds. The Text2KB project proposed in this paper aims to address this fragmentation by developing a system that can:

1. Extract structured knowledge about computability hierarchies from mathematical texts
2. Organize this knowledge according to established mathematical relationships
3. Support reasoning across these structures to identify connections and implications

4. Provide an accessible interface for exploring and applying this knowledge

At its core, the project represents an effort to create lasting infrastructure for organizing knowledge about the formal sciences, with a particular focus on computability theory and its hierarchies of definability.

2 Theoretical Background

2.1 Hierarchies of Definability

The hierarchies of definability that form the theoretical foundation of our project include:

2.1.1 The Arithmetic Hierarchy

The arithmetic hierarchy classifies formulas (and the sets they define) based on their quantifier structure:

- $\Sigma^0_0 = \Pi^0_0 = \Delta^0_0$: Decidable/recursive predicates (only bounded quantifiers)
- Σ^0_1: Formulas of the form $\exists x_1 ... \exists x_n R(x_1, ..., x_n)$ where R is decidable
- Π^0_1: Formulas of the form $\forall x_1 ... \forall x_n R(x_1, ..., x_n)$ where R is decidable
- Δ^0_1: Formulas that are both Σ^0_1 and Π^0_1 (recursive sets)

The pattern continues for higher levels (Σ^0_n, Π^0_n, Δ^0_n), where each level adds another alternation of quantifiers.

2.1.2 The Hyperarithmetic Hierarchy

The hyperarithmetic hierarchy extends the arithmetic hierarchy transfinitely:

- It can be viewed as iterating the Turing jump through computable ordinals
- All hyperarithmetic sets are Δ^1_1 (both Σ^1_1 and Π^1_1)

2.1.3 The Analytical Hierarchy

The analytical hierarchy further extends the classification:

- Σ^1_1: Sets definable by formulas with a leading existential second-order quantifier followed by a first-order formula
- Π^1_1: Sets definable by formulas with a leading universal second-order quantifier followed by a first-order formula
- Δ^1_1: Sets that are both Σ^1_1 and Π^1_1 (hyperarithmetic sets)

The pattern continues for higher levels of the analytical hierarchy (Σ^1_n, Π^1_n, Δ^1_n).

2.2 Reflection Principles and Feferman's Completeness

A key theoretical result underlying our project is Feferman's completeness theorem [Feferman, 1962], which demonstrates that for any true arithmetical sentence, there exists a systematic way to prove it by iterating reflection principles along well-ordered sequences.

More specifically, Feferman showed that for any true arithmetic statement ϕ, there exists a computable well-ordering L such that $RFN^L(PA) \vdash \phi$, where $RFN^L(PA)$ represents the theory obtained by iterating uniform reflection principles for Peano Arithmetic along the well-ordering L.

Recent work by Pakhomov et al. [Pakhomov et al., 2024] has refined these results, showing that for true Π_{2n+1} sentences, iterations along well-orders of type $\omega^n + 1$ are sufficient, and this bound is tight.

3 System Architecture

The proposed Text2KB system consists of several interconnected components:

3.1 Text Ingestion and Knowledge Extraction

1. **Source Collection**: A modular pipeline for legal acquisition of texts on computability theory, respecting copyright limitations
2. **Text Parsing**: Natural language processing techniques adapted for mathematical notation and logical formalisms
3. **Knowledge Extraction**: Identification of definitions, theorems, proofs, and relationships between concepts

3.2 Knowledge Representation

1. **Core Ontology**: Representation of fundamental concepts in computability theory (e.g., "recursive set," "Turing reduction")
2. **Hierarchical Relationships**: Structural relationships between complexity classes (e.g., "Π^0_1 statements include consistency assertions")
3. **Equivalence Classes**: Groups of problems or statements with equivalent complexity

3.3 Reasoning and Integration

1. **AgentSpeak Framework**: An agent-based system for querying and extending the knowledge base
2. **Translation Model**: An OpenNMT-based model for translating between natural language and formal knowledge representation languages (e.g., CycL)
3. **Cyc Integration**: Mapping computability concepts to Cyc's existing upper ontology and defining new microtheories

3.4 User Interface and Applications

1. **Zettelkasten-style Interface**: A personal knowledge management interface inspired by the Zettelkasten method
2. **Complexity Zoo Integration**: Connection with and extension of the Complexity Zoo database
3. **Problem Reduction Framework**: Tools for mapping real-world problems to known complexity classes

4 Implementation Approaches

4.1 Legal Knowledge Acquisition

To respect copyright while building a comprehensive knowledge base, we propose several approaches:

1. **Open Access Resources**: Prioritizing arXiv papers, university course notes, and open educational resources
2. **Public Domain Materials**: Incorporating older foundational texts that have entered the public domain
3. **Library API Access**: Utilizing non-consumptive research access provided by digital libraries
4. **Citation Networks**: Building knowledge around citations and references without reproducing protected content
5. **Publisher Partnerships**: Exploring formal arrangements with academic publishers for computational access

4.2 Knowledge Translation

A core technical challenge is translating between natural language mathematics and formal representation languages:

1. **Parallel Corpus Development**: Creating a parallel English-CycL corpus for training translation models
2. **OpenNMT Integration**: Using neural machine translation approaches adapted for mathematical content
3. **Intermediate Representation**: Developing a specialized vocabulary for computability concepts before full CycL integration

4.3 Incremental Knowledge Construction

Given the patchwork nature of available resources, the system is designed for incremental knowledge construction:

1. **Foundation-First Approach**: Beginning with the most fundamental, time-invariant aspects of computability theory

2. **Relationship-Centered**: Focusing on capturing relationships between concepts even when complete definitions are unavailable

3. **Provenance Tracking**: Maintaining clear linkage between knowledge assertions and their sources

5 Applications

5.1 Mathematical Knowledge Management

The primary application of the system is in mathematical knowledge management (MKM):

1. **Knowledge Preservation**: Creating lasting digital infrastructure for computability theory

2. **Knowledge Navigation**: Supporting exploration of the complex relationships between different hierarchies

3. **Knowledge Extension**: Facilitating the discovery of new connections and implications

5.2 Problem Reduction Framework

A secondary application is in supporting problem reduction:

1. **Complexity Mapping**: Identifying the appropriate complexity class for real-world problems

2. **Reduction Identification**: Suggesting possible reductions to known problems

3. **Solvability Assessment**: Providing context on the theoretical limitations of different approaches

5.3 Educational Applications

The system also has significant educational potential:

1. **Self-Directed Learning**: Supporting autodidactic exploration of computability theory

2. **Concept Visualization**: Providing visual representations of complex hierarchical relationships

3. **Knowledge Gaps**: Identifying and addressing gaps in available educational resources

6 Relevance to AI Control and Predictability

The concern that superintelligent AI systems might be inherently unpredictable or uncontrollable often cites undecidability results from theoretical computer science, such as the Halting Problem. While these concerns identify important limitations, they sometimes conflate different types of undecidability or fail to account for the nuanced hierarchy of decidability classes.

6.1 Refining the Undecidability Discussion

The Text2KB project provides several contributions to this discussion:

1. **Hierarchical Context**: By situating undecidability results within their proper hierarchies, we gain a more nuanced understanding of their implications
2. **Feferman's Lesson**: Feferman's completeness theorem demonstrates that even when a statement is undecidable within a particular formal system, there exist systematic ways to extend the system to decide it
3. **Relative vs. Absolute**: Distinguishing between relative undecidability (within a specific formal system) and absolute undecidability (across all possible systems)

6.2 Beyond Undecidability: Knowledge Organization for AI Safety

The project also offers broader contributions to AI safety discussions:

1. **Knowledge Infrastructure**: Building robust knowledge infrastructure around foundational theoretical concepts relevant to AI safety
2. **Reduction Frameworks**: Developing frameworks for reducing novel AI safety problems to known mathematical problems
3. **Precision in Discourse**: Supporting more precise communication about theoretical limitations and possibilities

6.3 A Note on Control vs. Alignment

While some discussions frame AI safety in terms of "control," this project aligns with perspectives that emphasize collaborative alignment rather than restrictive control. Understanding the theoretical landscape of decidability can inform approaches to alignment that respect both the capabilities and limitations of formal systems.

Feferman's completeness theorem offers an interesting parallel: it shows that while no single formal system can capture all arithmetic truths, there exists a systematic way to extend systems to accommodate any particular truth. Similarly, while no fixed set of alignment principles may guarantee safety across all possible AI developments, this doesn't preclude the possibility of systematic approaches to alignment that evolve alongside AI capabilities.

7 Conclusion and Future Work

The Text2KB project for computability hierarchies represents an ambitious effort to create lasting knowledge infrastructure in an area of fundamental importance to theoretical computer science, mathematics, and artificial intelligence. By systematically organizing knowledge about hierarchies of definability and computability, the project aims to make this knowledge more accessible, navigable, and applicable.

Future work will focus on:

1. Expanding the knowledge extraction capabilities to handle increasingly complex mathematical notation

2. Developing more sophisticated reasoning capabilities across the knowledge base

3. Exploring applications in AI safety research and formal verification

4. Extending the approach to adjacent areas such as model theory and proof theory

The relevance of this work to discussions of AI predictability and control lies not in providing simple answers to complex questions, but in contributing to a more nuanced understanding of the theoretical landscape. By distinguishing between different types and levels of undecidability, and by demonstrating systematic approaches to extending formal systems, the project offers valuable context for interpreting claims about the theoretical limitations of AI alignment and control.

In the spirit of Feferman's completeness theorem, we suggest that theoretical limitations within fixed formal systems need not imply absolute limitations across all possible approaches. Just as reflection principles offer systematic ways to transcend the limitations of particular formal systems, thoughtful approaches to AI development may find systematic ways to address alignment challenges even in the face of theoretical undecidability results.

References

Ash, C. J. and Knight, J. F. (2000). *Computable Structures and the Hyperarithmetical Hierarchy*, volume 144. Elsevier.

Boolos, G. S., Burgess, J. P., and Jeffrey, R. C. (2007). *Computability and Logic*. Cambridge University Press.

Feferman, S. (1962). Transfinite recursive progressions of axiomatic theories. *The Journal of Symbolic Logic*, 27(3):259–316.

Kechris, A. S. (1995). *Classical Descriptive Set Theory*, volume 156. Springer Science & Business Media.

Pakhomov, F., Rathjen, M., and Rossegger, D. (2024). Feferman's completeness theorem. *arXiv preprint arXiv:2405.09275v2*.

Rogers Jr, H. (1987). *Theory of Recursive Functions and Effective Computability*. MIT press.

Sacks, G. E. (2017). *Higher Recursion Theory*. Cambridge University Press.

Simpson, S. G. (2009). *Subsystems of Second Order Arithmetic*. Cambridge University Press.

Soare, R. I. (1999). *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*. Springer Science & Business Media.