Introduction

This project delves into the complex family relationships presented in the song "I'm My Own Grandpa" by modeling them in the Prolog coding language. The song describes a paradox where, through a series of marriages and parent-child connections, the narrator unwittingly becomes his own grandfather. Our group went through to encode these relationships in Prolog's logical framework, focusing on a minimal number of facts while maximizing rule-based inference. Our approach shows how prolog programming can get through seemingly contradictory situations in the real world by clearly defining family tree ties.

Background

The story begins with the narrator marrying a widow who has an adult daughter named Red hair. The narrator's father then marries red hair, leading to a confusing web of relationships: the narrator's father becomes his son-in-law, his wife turns into his grandmother, and his son is now his uncle. Prolog is particularly effective for modeling these complexities, thanks to its rule-based evaluation of easy to read rule cases. Unlike other programming languages, Prolog enables abstract definitions of relationships (for example, "X is Y's grandparent if X is the parent of Y's parent"), showing the dynamic connections without needing to list every fact exhaustively.

Implementation

The implementation employs a small collection of facts to represent individuals and their direct relationships, such as male(i), married(i, widow), and parent(father, i). Most of the logical structure is built around rules that generalize kinship terms. For instance, grandparent(X, Y) is determined by linking two parent relationships, while son_in_law(X, Y) checks if X is married to Y's daughter. Importantly, rules like uncle(X, Y) utilize existing definitions (like brother and parent) to minimize redundancy. This structure supports scalability; to add new relationships (like step-siblings), only incremental rules are needed instead of new facts.

Our group started by finding what all connections we needed to start defining by writing out the connections through the song. Then created functions to easily connect the differing family ties through a sort of twisted family tree. The rules use variables X, Y, Z to easily navigate the complex relationships like son in law, mother, aunt, step-parent, etc. The rules/functions needed to carefully walk through which of our predefined facts needed to be used with the relationship functions.

Testing

Conclusion

This project showcases the capabilities of Prolog in modeling recursive rules, and carefully running the complexity of family kinship. By emphasizing rules over facts, the

implementation easily shows the song's intricacies with a minimal amount of code while remaining open to future extensions. The successful queries, like grandfather(i, i), highlight how logical inference can clarify real-world contradictions that seem counterintuitive.