EXPLORING SEMANTIC HIERARCHIES TO IMPROVE RESOLUTION THEOREM PROVING ON ONTOLOGIES

by

Stanley C. Small

A Thesis Submitted in Partial Fulfillment
of the Requirements for a Degree with Honors
(Computer Science)

The Honors College
University of Maine
March 2019

Advisory Committee:

- Dr. Torsten Hahmann, Professor of Things, Advisor
- Dr. Mark Brewer, Professor of Things
- Dr. Max Egenhofer, Professor of Things
- Dr. Sepideh Ghanavati, Professor of Things
- Dr. Roy Turner, Professor of Things

ABSTRACT

A resolution-theorem-prover (RTP) evaluates the validity (truthfulness) of conjectures against a set of axioms classified as a knowledgebase. When given a conjecture, an RTP attempts to resolve the negated conjecture with axioms from the knowledge-base until the prover finds a contradiction. If the RTP finds a contradiction between the axioms and a negated conjecture, it proves the conjecture.

The order in which the axioms within the knowledge-base are evaluated significantly impacts the runtime of the program, as the search-space increases exponentially with the number of axioms.

Ontologies, knowledge bases with semantic (and predominantly hierarchical) structures, describe objects and their relationships to other objects. For example, a 'Car' class might exist in a sample ontology with 'Vehicle' as a parent class and 'Bus' as a sibling class. Currently, any hierarchical structures within an ontology are not taken into account when evaluating the relevance of each axiom. At present, each predicate is automatically assigned a weight based on a heuristic measure (such as the number of terms or the frequency of predicates relevant to the conjecture) and axioms with higher weights are evaluated first. My research aims to intelligently select relevant axioms within a knowledge-base given a structured relationship between predicates. I will use the semantic hierarchy over predicates to assign weights to each predicate passed to a weighting function. The research aims to design heuristics based upon the semantics of the predicates, rather than solely the syntax of the statements.

I plan to develop weighting functions based upon various parameters relevant to the ontological structure of predicates contained in the ontology, such as the size and depth of a hierarchy based upon the structure of the ontology.

I will implement methods to calculate weights for each predicate and thus each axiom in attempts to select relevant axioms when proving a theorem. Then, I will conduct an experimental study to determine if my methods show any improvements over current reasoning methods.

ACKNOWLEDGEMENTS

Many thanks are given to Dr. Hahmann. This work could not be completed without his continued support and encouragement. Despite his tremendously busy schedule, he always made time to meet and answer questions.

Robert Powell also proved instrumental to the process. His utility which converts Common Logic Interchange Format (CLIF) into web ontology language (OWL). His work streamlined the testing process and allowed me to find necessary results.

The thesis committee was also instrumental in producing this undergraduate thesis.

LIST OF FIGURES

CONTENTS

1 INTRODUCTION

A theorem is a statement that can be proved from a series of asserted facts, typically referred to as axioms, using the rules of logic. By expressing facts in a formal manner, new statements can be generated mechanically by a computer program without additional user input. Computers excel at tasks which remain simple and repetitive. Formal logic defines rules of inference which allow one to transform established facts into new conclusions based on the syntax of these statements. Such inference rules are relatively easy to implement in a computer program because these rules rely only on the syntax, or the structure, of facts instead of any sense of meaning inherent in a set of facts. This is referred to as automated theorem proving. Automated theorem proving provides a versatile method for reasoning with a set of facts, and has been used to prove and verify proofs of multiple theorems. The four color map theorem is a notable example of a theorem obtained by an automated proof in 1976, and by a general-purpose theorem-proving software in 2005 [2]. Moreover, advances have been made in work on the Kepler conjecture and in finding optimal solutions for a Rubik's Cube with computer programs using formal reasoning methods.

ONTOLOGIES AIM TO CAPTURE THE SEMANTICS OF A DOMAIN'S CONCEPTS AND RELATIONS. The general-purpose nature of automated theorem proving makes the method applicable to a variety of problems, but a considerable amount of information is lost when applied on ontologies. ONTOLOGIES ARE SETS OF AXIOMS. An ontology is a set of facts describing entities and the relationships between them. Some relationships between an ontology's terms may be explicitly defined, but many are implicit. EXAMPLE OF RELATIONSHIP. Relationships between entities can often be represented as a tree, and more specifically a hierarchy. Encoded in the hierarchy is a wealth of information detailing the relationships between entities described in an ontology. Currently, knowledge encoded in structural hierarchies constructed from ontologies is not taken into account when determining which axioms might be most helpful when attempting to prove a specific theorem. This work attempts to improve automated theorem proving with ontologies by identifying relevant facts, and ignoring those less likely to yield a proof. DESCRIPTION OF THE

CONCRETE OBJECTIVE AND APPROACH OF YOUR WORK

2 BACKGROUND AND RELATED WORK

2.1 Ontologies

Ontologies describe entities and relationships between them. The word ontology ("study of being") combines Greek *onto-* ("being") and *-logia* ("logical discourse"). The act of studying knowledge and existence in philosophy has given birth to the study of formal logic and automated reasoning in computer science. Ontologies provide a "common vocabulary" for researchers to speak about a specific domain [4]. Furthermore, ontologies can often be interpreted by machines to aid in automated proofs. While ontologies at a fundamental level exist as a collection of facts, the organization and descriptions which researchers choose to represent a specific domain are important. Perhaps an ontology can best be understood with an example inspired by an example composed by Natalya F. Noy [4].

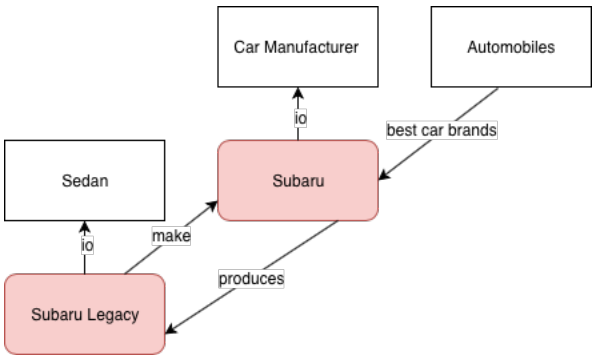

Figure 1: This is a simple ontology describing classes (black) and instances (red) in the "automobiles" ontology.

2.2 Theorem Proving

Formal logic, specifically first-order logic, defines a structure for statements which can be used to form logical and mathematical proofs. Asserted facts, called axioms, are used to derive facts which logically follow. Consider the following set of facts.

$$isSedan(SubaruLegacy)$$
$$isSedan(X) \rightarrow hasFourSeats(X)$$

The first statement asserts the Subaru Legacy is a sedan. The second statement asserts all sedans have four seats. These two statements do not directly state the Subaru Legacy has four seats. However, one can derive the statement $hasFourSeatsl(SubaruLegacy)$ by using the inference rule *modus ponens*, defined below.

$$A$$
$$A \rightarrow B$$
$$\therefore B$$

One can think of A and B as variables representing statements, and any statement can replace them. An inference rule defines a valid rule for statements. By expressing facts in a formal notation, one makes proofs using such statements mechanical and easily parsed by a computer. Expressing statements in formal logic poses multiple challenges. First, each object and relationship must be explicitly defined.

Resolution is one of many methods for automated theorem proving. Historically, resolution has significance and is widely used [1, p. 51]. In order to use resolution as a proof technique, axioms must first be expressed in Conjunctive Normal Form. This can be done by expressing the set of facts as a conjunction of disjunctions.

$$isSedan(X) \rightarrow hasFourSeats(X)$$
$$\neg isSedan(X) \vee hasFourSeats(X)$$

One can then resolve the statements.

$$\frac{isSedan(X) \vee hasFourSeats(X), \neg Winter \vee Cold}{Hot \vee Cold}$$

In resolution, the search space increases exponentially with the addition of each new axiom. Currently only the syntax of statements is taken into account

Syntax defines the structure of a sentence, whereas semantics describe the meaning of a sentence.

An ontology defines categories and relationships among objects. One can think of an ontology as a "vocabulary" used to desrcibe a domain [5, p. 308]. Typically, the objects can be arranged in a hierarchy.

At a simplistic level, semantic hierarchies describing an ontology can be compared to a family tree. Given a pair of two individuals, if one were tasking with determining if two individuals are related, a family tree would prove quite useful.

3 APPROACH

3.1 Weighting Functions

Given an ontology and a conjecture, one can generate weights for classes and properties to reduce the number of clauses generated in a proof. Functions are evaluated by their influence on the number of clauses generated with a proof.

After a hierarchy has been generated, weights can be assigned to each class and subproperty. The same weighting function is applied to both the classes and sub-properties.

The weighting functions are currently applied by hand to the ontologies, with the beginnings of an automated program underway.

3.2 Computing Weighting Functions

4 EXPERIMENTS

The study was a success!

4.1 Setup

My experiments were conducted using Prover9, written by William McCune [3]. Many tests were conducted using a version of the program which supports a GUI, but a command line version is available for running automated tests. Git was used for version control and a repository containing source code can be found at `https://github.com/stanleysmall/thesis`.

4.2 Results

5 CONCLUSIONS

In many cases, the algorithm increases the number of clauses generated for each test, but does not do so to the point where the tests are unusable. For some very specific ontologies, the number of clauses generated decreases. This can be attributed to, in part, by the small number of ontologies available for testing, along with the specific pattern and hierarchy of each ontology.

Many opportunities for further research include fully automating the search procedure, working with a larger number of ontologies to ensure the weighting functions actually do as they say, developing a new approach towards automatically weighting the predicates.

REFERENCES

[1] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer, 2018.

[2] Georges Gonthier. "Formal proof–the four-color theorem". In: *Notices of the AMS* 55.11 (2008), pp. 1382–1393.

[3] William McCune. *Prover9 and mace4*. 2005.

[4] Natalya F Noy, Deborah L McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*. 2001.

[5] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

A TESTS

AUTHOR'S BIOGRAPHY

Stanley C. Small grew up in Hampden, Maine with his mother Diane and his father Scott. He attended the University of Maine and received a Bachelor of Science degree in Computer Science in May of 2019.