Justification Proof Search Implementation in Python

Bachelorarbeit

der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Judith Fuog

2014

Leiter der Arbeit:
Prof. Dr. Thomas Studer
Institut für Informatik und angewandte Mathematik

Abstract

In short what's it all about.

Contents

Chapter 1

Introduction

1.1 Motivation

What's the motivation behind it? Not **MY** motivation, but the scientific motivation.

1.2 Goal

The initial goal was to extend an existing proof search engine [1] such that it could also handle Justification Logic. Deeper investigation into that project revealed that to make it handle also Justification Logic the given interface in Python would not be enough, but the expenses it would require to get so much deeper into the material that the actual indented work would be only secondary. So instead of extending *Z3* the actual goal changed to implementing a simplified proof search for Justification Logic.

Z3

The program should satisfy to following conditions:

Input The formula to be proven as well as a list of formulas needed for the proof is given as string. It may be presumed that the string is exactly formatted in the way needed. It must not be checked for syntax error or general typing mistakes.

Output A simple *True* or *False* for the provability of the formula. [2]

1.3 Overview

The second chapter will present a short introduction to Justification Logic, but will go only as deep as needed to understand the problem as well as the develop algorithm.

The heart of the third chapter will introduce the algorithm used in the program. Since this thesis concerns itself more with the practical side of imple-

j-logic

[1] Z3 from Microsoft Research

[2] Optional the output could give information about how a proof was found if the formula is provable.

mentation and not the theoretical side of mathematical logic theory there will be little proof here but instead many example to show how the algorithm works.

Finally the last chapter will discuss the result of the work and give some ideas about how the work of a Justification Logic proof search implementation could be improved.

Chapter 2

Background

2.1 Justification Logic

2.2 Order of Operation Tree

Chapter 3

A Divide and Conquer Algorithm

3.1 Core Idea

3.2 Divide

3.2.1 Atomize

Sumsplit

Simplify Bang

Remove Bad Bang

3.2.2 Get Must

3.3 Conquer

3.3.1 Configurations and Conditions

3.3.2 Merge

3.4 Implementation

Chapter 4

Results

4.1 Application

4.2 Enhancement

Authorship

Todo list