

# Automated Reasoning

Brian Kuhns

April 2019

## 1 Introduction

Someone better at writing do this part.

## 2 Algorithm

The algorithm is an extension of ordered resolution, which attempts to find finite representations of infinite clause families, and proceed with ordered resolution on these families.

### 2.1 Anatomy of a Scheme

A scheme has 2 parts: a list of implications and a list of base clauses.

Each implication is a list of clauses and a clause. Signifying that the and of all of left clauses implies the right clause, hereafter the hypotheses and the conclusion.

An implication is syntactically valid if each hypothesis subsumes the conclusion. A scheme is syntactically valid if each implication is valid and each hypothesis of each implication subsumes each base clause.

These properties are important because they ensure that any implication can be applied to any set of clauses in the scheme. Without these properties resolution and subsumption become much more complex, and may be undecidable.

A scheme is said to contain a clause iff it can be derived by applying some finite sequence of the implications to the base clauses.

### 2.2 Learning Implications

Learning implications is done in two phases: finding candidates and constructing implications from them.

### 2.2.1 Candidates

A candidate is a tree whose nodes are clauses and whose leafs are clauses. Suppositions are also clauses but they are handled differently to construct implications.

A candidate should be derived whenever the history of a clause contains a subtree which subsumes it. A clause-tree subsumes another clause-tree if the graphs are isomorphic and each node subsumes the corresponding node in the other tree.

Two clauses are of the same form if some clause subsumes both of them.

The important property to have is that, if any set of clauses of some form can derive a new clause of one of that forms. A scheme is derived which subsumes all such resolutions.

### 2.2.2 Implications

When given a candidate the important property to ensure is that if it can be resolved indefinitely, then a scheme which subsumes all of these clauses is derived.

First mark all leaf clauses of the same form as the conclusion as suppositions.

The following properties should be vetted.

- There is at least one supposition
- The conclusion can larger than all of the hypothesis
- The conclusion can be a non-tautology

Once these properties are verified find the most specific form of the suppositions and conclusion.

This can be found by a process much like unification except when two clauses are not unifiable instead of failing replace them with a variable.

Next resolve clauses in the tree, with all the suppositions replaced by the most specific form. The resulting clauses are the conclusions and the suppositions are the implications.

## 2.3 Subsumption