From Additive to Classical Proof Search

Adam Lassiter
Department of Computer Science
University of Bath

Willem Heijltjes
Department of Computer Science
University of Bath

April 24, 2019

Abstract

We investigate a natural algorithm for proof search within classical logic and bounds on the complexity class of such a search. We further examine natural optimisations to this algorithm and how they affect complexity bounds.

Introduction

Building on the work done by Heijltjes and Hughes (2015), we investigate proof search in classical logic through additive linear logic (ALL). The process we investigate, called *coalescence*, is a top-down proof search from axiom links down to the conclusion. This method is promising as it boasts great efficiency for ALL proof search and has a natural transformation to sequent calculus proofs.

Sequent Calculus and ALL

A classical formula can be proved by an n-dimensional additive proof, for some n dependant upon the formula. We propose some simple classes for formulae — boolean constants *only* are 1-dimensional, normal additive proofs are 2-dimensional amongst others etc. We prove that this idea is consistent through *additive stratification* of the sequent calculus — that is, any sequent proof may be 'rearranged' up to the order of rules applied, in particular with *weakening* at the top and *contraction* at the bottom. Coalescence is then exactly (additively stratified) proof search.

Coalescence

We construct our proof search through use of 'natural' deductions. Using a system analogous to *petri nets*, we construct a net through the n-dimensional cross product of places in a formula and associated transitions across the net. Each token in our net begins at an axiom and transitions are exhaustively applied up the formula's syntax tree — or down an equivalent sequent proof tree. A place is a coordinate in the n-dimensional grid representing a context of n places in the formula that is provable. The process either halts when the root of the formula syntax tree is reached and a proof is constructed, or restarts in a higher dimension when applicable transitions are exhausted and the dimensionality must be increased. The dimensionality of a proof is then the dimensionality of our grid when the root is reached.

Motivation

Clearly complexity scales with dimensionality and our motivation is then: *'What dimension is sufficient for a given formula?'*. In essence, this gives an upper bound for proof search.

Some Examples

We find that this does not yield results as expected — for example, in $(a \vee \neg a) \wedge (b \vee \neg b)$ we would expect a dimension of 2 since each component may be proved in 2 dimensions and the conjunction should be trivial. We instead find, unsatisfyingly, that this increases dimension quickly and unreasonably, with the aforementioned case yielding dimensionality 3 and linear growth for subsequent additional variables.

Solution

To solve this issue, we then investigate liberating the search algorithm and generalising over the properties of sequents — namely, idempotency and commutativity. This includes: some notion of applying conjunctions 'diagonally' and switching from tuple or multiset links to set links. The latter takes us into more familiar/obvious proof search territory.

Beyond these two points, there exist various other implementation trade-offs: dense/sparse representation, high-performance data structures and assorted other ad-hoc optimisations.

Further Examples

Our optimisation through generalising over sequents yields favourable results for conjunctive normal form (CNF) but maintains poor performance for disjunctive normal form (DNF) formulae. We construct a simple algebra of classes, with conjunction and disjunction of proven formulae equivalent to maximum and minimum functions of left and right subformulae dimensionality. Finally, we hypothesise a generalised bound for any formula and provide the 'essence' of the associated proof — we expect dimensionality to be equivalent to the most number of variables in any DNF subformula.

Ongoing Work

We continue to study how coalescence proof search relates to traditional proof search methods, but we expect it to be similar to either the 'connections' or 'matrix' method — see: Renne (2004), Bibel (1981), Otten and Kreitz (1995) and Andrews (1981). Progress is still to be made as to DNF formulae — we expect generalisation over associativity of ALL terms and automatic construction of subformulae may hold the key to a more natural computation.

References

Andrews, Peter B. Theorem proving via general matings. In *Journal of the ACM (JACM)*, vol. 28(2):193–214 (1981).

Bibel, Wolfgang. On matrices with connections. In *J. ACM*, vol. 28(4):633–645 (1981).

Heijltjes, Willem and Hughes, Dominic JD. Complexity bounds for sum-product logic via additive proof nets and petri nets. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, (pp. 80–91). IEEE (2015).

Otten, Jens and Kreitz, Christoph. A connection based proof method for intuitionistic logic. In *International Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, (pp. 122–137). Springer (1995).

Renne, Bryan. Tableaux for the logic of proofs. In *CUNY PhD Program in Computer Science, Technical Report TR-2004001, New York*, (pp. 148–158) (2004).