From Additive to Classical Proof Search

Adam Lassiter
Department of Computer Science
University of Bath

Willem Heijltjes
Department of Computer Science
University of Bath

Introduction

Building on the work done by Heijltjes & Hughes (2015), we investigate proof search in Classical Logic (CL) 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. The interesting features are that: proof search steps are simple, $\mathcal{O}(n)$, there is no need for backtracking, the method has built-in sharing (or memoization); but dimensionality increases rapidly, and sometimes unexpectedly.

Additive Linear Logic

ALL is the fragment of linear logic that concerns sum ($+$) and product ($\times$), with their units 0 and 1. A *formula* of ALL is constructed:

$$A, B, C \quad ::= \quad 0 \mid 1 \mid a \mid \overline{a} \mid A + B \mid A \times B$$

We write $|\mathsf{A}|$ for the set of subformula occurrences of A (we distinguish uniquely both occurrences of a in $a \times a$).

A sequent for ALL is a pair $\vdash A, B$, and a sequent calculus for ALL is given by the following rules (where the symmetric rules operating on the second element of the pair $\vdash A, B$ are omitted):

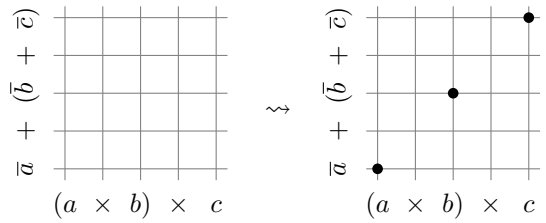
$$\frac{}{\vdash a, \overline{a}}\, ax \qquad \frac{}{\vdash 1, A}\, 1 \qquad \frac{\vdash A, C}{\vdash A + B, C}\, +_1$$

$$\frac{\vdash B, C}{\vdash A + B, C}\, +_2 \qquad \frac{\vdash A, C \qquad \vdash B, C}{\vdash A \times B, C}\, \times$$

Coalescence Proof Search

Naively searching for a sequent proof, starting from a given conclusion, is exponential: the additive conjunction rule ($\times$) duplicates its context C, and reciprocated duplication between both formulas in the sequent creates exponential growth. A more efficient algorithm, first observed by Galmiche & Marion (1995) and later by Heijltjes & Hughes (2015), is given by placing proof search for a sequent $\vdash A, B$ in the product space $|\mathsf{A}| \times |\mathsf{B}|$. This set contains all *sub-sequents* of $\vdash A, B$ that might occur in a sequent proof of $\vdash A, B$, without redundancy. We represent $|\mathsf{A}| \times |\mathsf{B}|$ by a

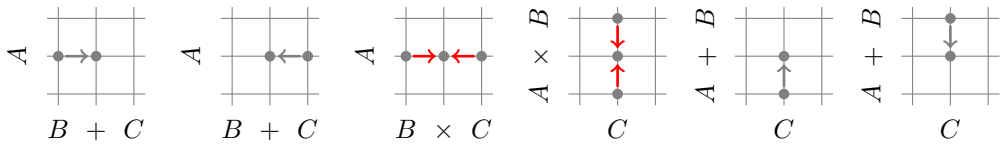
grid, e.g. that for $\vdash \overline{a} + (\overline{b} + \overline{c}), (a \times b) \times c$ is below:



The *coalescence proof search* algorithm for ALL is then as follows. We place tokens on this grid to indicate a sub-sequent is provable. We initially place tokens on each position $(\overline{a}, a)$, $(1, A)$, and $(A, 1)$, as shown above right for our example, then apply the following local rules (and the symmetric variants):

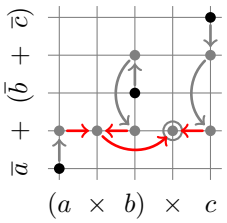
- Place a token on $(A + B, C)$ if (A, C) has a token.
- Place a token on $(A + B, C)$ if (B, C) has a token.
- Place a token on $(A \times B, C)$ if (A, C) and (B, C) have a token.

We illustrate these graphically as follows; note, though, that the *neighbours* of a position in the grid are given by the subformula relation, and not by adjacency in the plane:



Observe also that the initial token placement and the propagation rules correspond directly to the axioms and sequent rules for ALL respectively.

Finally, if the root position (A, B) for $\vdash A, B$ has a token, we succeed; otherwise, if no more tokens can be placed we fail. For our example, we get the following trace:



This algorithm succeeds if and only if there is an ALL proof of the given formula. Clearly, each progression is local (i.e. constant time), and the number of tokens placed is bounded by the size of the grid $|\mathsf{A}| \times |\mathsf{B}|$. The algorithm thus runs in polynomial time. The interesting observation is that, naively, the additive, context-duplicating conjunction rule seems very inefficient for proof search. The coalescence approach solves this problem by using a natural data structure, the grid, to obtain an algorithm with natural memoization, that is efficient for this reason.

Classical Logic and Coalescence

We investigate how the coalescence technique applies to classical propositional logic. The main idea is that a classical formula A can be proved by additive rules applied to a sequent $\vdash A, \ldots, A$ with n copies of A. This is easily shown by a *stratification* of sequent proofs with additive rules, where all contractions are pushed to the bottom (conclusion), and weakenings to the top (axioms).

Correspondingly, we generalize coalescence proof search to CL by applying it to a grid of n dimensions, for any n, where it was previously fixed at 2. The algorithm will start at dimension 1, and when it fails at dimension n it will try again at dimension $n+1$, up to a theoretically determined upper bound. A *formula* within CL is constructed:

$$A, B, C \quad ::= \quad \top \mid \bot \mid a \mid \overline{a} \mid A \vee B \mid A \wedge B$$

A sequent calculus for CL is then given by the following rules:

$$\frac{}{\vdash \top}\,\top \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \vee B}\,\vee \qquad \frac{\vdash \Gamma}{\vdash \Gamma, A}\,w$$

$$\frac{}{\vdash a, \overline{a}}\,ax \qquad \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \wedge B}\,\wedge \qquad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\,c$$

where A, B, C are formulae and Γ, Δ, Σ are sequents. Notice that both conjunction $\wedge$ and disjunction $\vee$ rules preserve the number of terms in a sequent, but there is no longer a bound on the number of terms per sequent through the weakening w and contraction c rules.

A proof tree is said to be *additively stratified* if it is structured as follows:

$$\dfrac{\dfrac{\dfrac{}{\vdash A_1}\,\top, ax}{\vdash \Gamma_1}\,w \quad \ldots \quad \dfrac{\dfrac{}{\vdash A_n}\,\top, ax}{\vdash \Gamma_n}\,w}{\dfrac{\vdash P \ldots P}{\vdash P}\,c}\,\wedge, \vee$$

That is, any proof tree in CL sequent calculus may be rearranged such that rules applied are ordered $\{\top, ax\}, w, \{\vee, \wedge\}, c$. This rearrangement does not affect the number of terms of sequents.

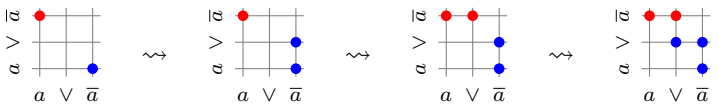
For a formula P in CL, the coalescence algorithm runs as follows:

1. Set $n := 1$
2. Construct the n-dimensional grid of possible n-term sequents $\vdash A_1 \ldots A_n \in |\mathsf{A}| \times \ldots_n \times |\mathsf{A}|$
3. Spawn tokens at all instances of axiom links $\vdash \Gamma, a, \overline{a}$
4. Exhaustively perform transitions given by the CL sequent calculus rules
5. Does there exist a token at $(P, P \ldots P) \equiv \vdash P, P \ldots P \equiv \vdash P$?
   (a) Yes — Halt with a proof for P and dimensionality n
   (b) No — Increment $n := n + 1$ and goto 2

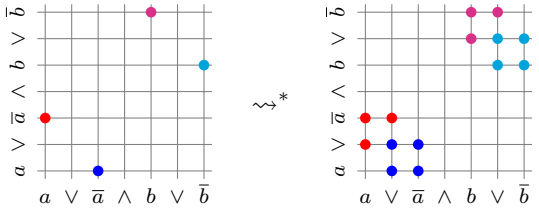
The dimensionality of a proof is then the dimensionality of our grid when the root is reached, equivalent to the number of contractions required in an equivalent sequent proof. A CL formula can thus be proved by an n-dimensional additively stratified proof, where n is the number of terms in a sequent before contraction. Through a natural transformation on steps of the algorithm to equivalent sequent proofs, coalescence up to $n = N$ is then exactly (additively stratified) proof search, with implicit weakening and contraction of all sequents up to N terms.

Some Examples

Consider a simple proof requiring disjunction through $A ::= a \vee \overline{a}$ as follows:



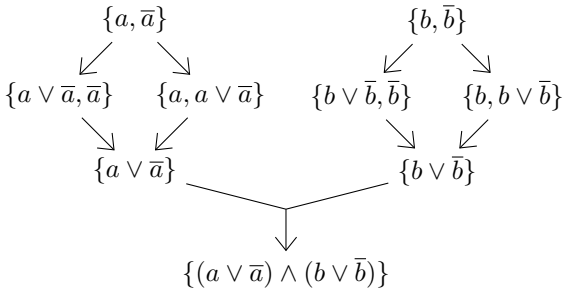
The root position (A, A) is reached in 2 dimensions so we describe the associated proof as having dimensionality 2. Consider next a simple proof requiring conjunction through $B ::= (a \vee \overline{a}) \wedge (b \vee \overline{b})$ as follows (note intermediate steps are skipped and the grid is saturated before a proof is reached):



We do not reach the root (B, B) for $n = 2$ despite having two proven 'subproofs' and need only apply a simple conjunction. Instead, a solution is reached for $n = 3$ (visualisation omitted due to lack of clarity of 3D diagrams). For a similar term in 3 variables a, b, c, a solution is reached for $n = 4$ and growth continues linearly. This is deemend unsatisfying and we readdress the mechanics of coalescence to fix this.

Current Work

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' (from $(a \vee \overline{a}, b \vee \overline{b})$ to (B, B) in one step in the above) and switching from *tuple* or *multiset* links to *set* links. The latter takes us into more familiar/obvious proof search territory:



Furthermore, we examine bounds on the dimension required for a given formula. We initially propose a trivial theoretical upper bound through number of unique atoms in a formula A. Through construction of a set of classes representing formulae provable in n dimensions and a simple algebra of these classes, we hypothesise complexity is bounded by the largest disjunctive form in $|\mathsf{A}|$.

References

Galmiche, D. & Marion, J.-Y. (1995), Semantic proof search methods for all-a rst approach, *in* '4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods, St Goar am Rhein, Germany'.

Heijltjes, W. & Hughes, D. J. (2015), 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', IEEE, pp. 80–91.