

Natural Proof Search for Classical Logic

Adam Lassiter
Department of Computer Science
University of Bath

Willem Heijltjes
Department of Computer Science
University of Bath

February 10, 2019

Abstract

We investigate a natural algorithm for proof search within classical logic and prove bounds on the complexity class of such a search.

1 Classical Logic

Definition 1.1 (Formulae).

A *formula* within classical logic is constructed as follows:

$$\begin{aligned} A, B, C &::= \perp \mid \top \mid a \mid \neg a \mid A \vee B \mid A \wedge B \\ \Gamma, \Delta, \Sigma &::= A_1 \dots A_n \end{aligned}$$

where \vee, \wedge are additive linear logic disjunction and conjunction respectively and Γ, Δ, Σ are contexts..

Example.

Definition 1.2 (Sequent Proofs).

Within *classical logic*, a *sequent proof* is constructed from the following rules:

$$\begin{array}{ccc} \frac{}{\vdash \top} \top & \frac{\vdash \Gamma, A}{\vdash \Gamma, A \vee B} \vee R & \frac{\vdash \Gamma}{\vdash \Gamma, A} w \\ \frac{}{\vdash a, \neg a} ax & \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \wedge B} \wedge R & \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} c \end{array}$$

where A, B, C are formulae and Γ, Δ, Σ are sequents. A sequent proof provides, without context, a proof of its conclusion and each line of the proof represents a tautology.

Example.

Remark 1.3.

Within the context of weakening and contraction, *additive* and *multiplicative* rules are inter-derivable.

Definition 1.4 (Derivations).

Given *tops* $\Gamma_1 \dots \Gamma_n$ for the sequent proof $\vdash \Delta$, a *derivation* is a tree providing a proof of $\Gamma_1 \dots \Gamma_n \Rightarrow \Delta$.

A derivation is written as:

$$\frac{\frac{\vdash \Gamma_1 \quad \dots \quad \vdash \Gamma_n}{\vdash \Delta} [label]}$$

where the *label* describes which rules may be used within the derivation.

Corollary 1.5 (Derivation Equivalence).

A sequent proof is a derivation where all top derivations of the tree are $\vdash \top, ax$. Equivalence of derivations may be weakly defined up to equivalence of leaves and conclusion.

Example.**Definition 1.6** (Additive Stratification).

A proof tree is said to be *additively stratified* if $\vdash P$ is structured as follows:

$$\frac{\frac{\frac{\vdash A_1}{\vdash \Gamma_1} w \quad \dots \quad \frac{\frac{\vdash A_n}{\vdash \Gamma_n} w}{\vdash P \dots P} \wedge, \vee}{\vdash P} c$$

That is, the inferences made in an additively stratified proof are strictly ordered by:

1. Top/Axiomatic
2. Weakening
3. Conjunction/Disjunction
4. Contraction

Example.**Proposition 1.7** (Stratification Equivalence).

Given $\vdash A$, there exists an additively stratified proof of A .

Proof. For each instance of a weakening below another inference, there exists an equivalent subproof that is additively stratified:

$$\frac{\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B} \vee}{\vdash \Gamma, A \vee B, C} w \quad \rightsquigarrow \quad \frac{\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A, B, C} w}{\vdash \Gamma, A \vee B, C} \vee$$

$$\begin{array}{ccc}
\frac{\frac{\frac{\vdash \Gamma, A}{\vdash \Gamma, A \wedge B} \wedge}{\vdash \Gamma, A \wedge B, C} w}{\vdash \Gamma, A \wedge B, C} w & \rightsquigarrow & \frac{\frac{\frac{\vdash \Gamma, A}{\vdash \Gamma, A, C} w}{\vdash \Gamma, A \wedge B, C} w}{\vdash \Gamma, A \wedge B, C} w \wedge \\
\\
\frac{\frac{\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} c}{\vdash \Gamma, A, B} w}{\vdash \Gamma, A, B} w & \rightsquigarrow & \frac{\frac{\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A, A, B} w}{\vdash \Gamma, A, B} c}{\vdash \Gamma, A, B} c
\end{array}$$

Similarly, for each instance of a contraction above another inference, there exists an equivalent subproof that is additively stratified:

$$\begin{array}{ccc}
\frac{\frac{\frac{\vdash \Gamma, A, A, B}{\vdash \Gamma, A, B} c}{\vdash \Gamma, A \vee B} \vee}{\vdash \Gamma, A \vee B} \vee & \rightsquigarrow & \frac{\frac{\frac{\frac{\vdash \Gamma, A, A, B}{\vdash \Gamma, A, A, B, B} w}{\vdash \Gamma, A \vee B, A, B} \vee}{\vdash \Gamma, A \vee B, A \vee B} \vee}{\vdash \Gamma, A \vee B} c \\
\\
\frac{\frac{\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} c}{\vdash \Gamma, A \wedge B} \wedge}{\vdash \Gamma, A \wedge B} \wedge & \rightsquigarrow & \frac{\frac{\frac{\frac{\vdash \Gamma, B}{\vdash \Gamma, A, B} w}{\vdash \Gamma, A, A \wedge B} \wedge}{\vdash \Gamma, A \wedge B, A \wedge B} \wedge}{\vdash \Gamma, A \wedge B} c
\end{array}$$

By induction from the leaves downwards on a finite height tree, apply the associated rule to each pair of inferences of the form (c above inf). Any given $\vdash P$ may be rewritten:

$$\frac{\frac{\frac{\frac{\vdash A_1}{\vdash \Gamma_1} \wedge, \vee, w}{\vdash \Gamma_1} \dots \frac{\frac{\frac{\vdash A_n}{\vdash \Gamma_n} \wedge, \vee, w}{\vdash \Gamma_n} c}{\vdash P} c$$

Again, by induction from the root upwards on this partially stratified tree, apply the associated rule to each pair of inferences of the form (w below inf). $\vdash P$ may then be further rewritten:

$$\frac{\frac{\frac{\frac{\vdash A_1}{\vdash \Gamma_1} w}{\vdash \Gamma_1} \dots \frac{\frac{\frac{\vdash A_n}{\vdash \Gamma_n} w}{\vdash \Gamma_n} \wedge, \vee}{\vdash P \dots P} c}{\vdash P} c$$

□

Example.

2 Coalescence

Definition 2.1 (Petri Nets).

For the purposes required here, a *petri net* \mathcal{N} is $(\mathcal{P}, \mathcal{F})$ where $f \in \mathcal{F} : \mathcal{P}^m \times \mathcal{P}$. In particular, \mathcal{P} is a set of places and \mathcal{F} a set of flows. A *configuration* is a set $\mathcal{C} \subset \mathcal{P}$ of tokens in places.

Example.

Definition 2.2 (Firing Petri Nets).

Given a petri net \mathcal{N} and configuration \mathcal{C} , a *firing* of the net \mathcal{N} is a new configuration generated by application of a transition $f \in \mathcal{F}$ on m tokens $c_1 \dots c_m \in \mathcal{C}$. In particular, $(\mathcal{N} = (\mathcal{P}, \mathcal{F}), \mathcal{C}) \mapsto (\mathcal{N}, \mathcal{C} \cup f_{right} \setminus f_{left})$ for some $f = (f_{left}, f_{right}) \in \mathcal{F}$

A petri net is said to be *exhaustively fired* if it is fired until there does not exist any such $f \in \mathcal{F}$ to fire.

Example.

Remark 2.3.

Implementing fireable petri nets is straightforward using an n -dimensional boolean array of places visited and a collection of n -tuples representing tokens.

Definition 2.4 (Coalescence).

Given a formula P , the coalescence algorithm is as follows:

1. Set $n := 1$
2. Construct a n -dimensional petri net \mathcal{N} from P where each subformula is a place, each conjunction and disjunction a flow
3. Construct a configuration \mathcal{C} with a token at each place $p = (\dots, a, \dots, \neg a, \dots)$ the intersection of a pair of tautological atoms
4. Exhaustively fire the petri net \mathcal{N} using the *spawning* method.
5. If there exists a token in the configuration \mathcal{C}^* at the root of the formula P , halt and return n
6. Otherwise, increment $n := n + 1$ and go to step 2

Example.

Proposition 2.5 (Coalescence Proof Search).

The coalescence algorithm on P is exactly a proof search on P .

Proof. In particular, consider the additively stratified $\vdash P$ with $n - 1$ contractions (should such a proof exist). The first configuration of tokens is precisely a proof all possible combinations of the *axiom* rule with $n - 3$ other terms through the *weakening*. Each flow transition followed when fired is an application of either the \vee or \wedge rule. Finally, a token at the root of the formula is $\vdash P \dots P$, with the *contraction* rule applied implicitly.

If there exists an additively stratified proof of P , the coalescence algorithm will find it.

Since for every proof there exists an additively stratified proof, coalescence is precisely proof search. \square

3 Dimensionality

Definition 3.1 (Dimensionality).

Given a formula P , the coalescence proof search produces a proof in an n dimensional petri net. The dimensionality of P is then defined $\dim(P) ::= n$. Equivalently, an additively stratified sequent proof requires $n - 1$ contractions at the bottom of the proof. Given $\vdash P$, its dimensionality is defined $\dim(P) ::= n$.

Proposition 3.2 (Deducing Dimensionality).

Given formulae P, Q such that $\exists \vdash P$ with $\dim(P) = n$ and $\exists \vdash Q$ with $\dim(Q) = m$, then:

$$\begin{aligned}\dim(P \vee Q) &= \min(n, m) \\ \dim(P \wedge Q) &= \max(n, m)\end{aligned}$$

Proof. Omitted \square

Remark (Satisfiability vs Provability).

For any formula P , there exist four distinct classes: *true*, *false*, *satisfiably true*, *satisfiably false*, where *satisfiably* differs by finding a particular assignment of values to each variable. Coalescence searches for a proof of P in *true*, whereas the SAT problem addresses a proof of P in *true*. In particular, *true* is the complement class to *satisfiably false* and *false* complement to *satisfiably true*.

Definition 3.3 (Dimensionality when not Provable).

Given a formula P such that there does not exist $\vdash P$, for all formulae Q such that there does exist $\vdash Q$, the dimensionality of P is defined as the least n such that:

$$\dim(P) ::= n \implies \begin{cases} \exists \vdash (P \vee Q) \implies \dim(P \vee Q) = \min(n, \dim(Q)) \\ \exists \vdash (P \wedge Q) \implies \dim(P \wedge Q) = \max(n, \dim(Q)) \end{cases}$$

Remark.

This definition of a ‘partial dimensionality’ amounts to finding the highest dimension in coalescence proof search that some useful deduction was made, outside of trivial cases.

Definition 3.4 (Paths in a Tree).

Path from root to any leaf (effectively iterating leaves, but tracing parents)

Proposition 3.5 (\vee -Bound on Dimensionality).

$$\begin{aligned}\dim(P) &\leq 1 + \#\{\vee \in P\} \\ \dim(P) &\leq 1 + \max\{\#\{\vee \in path\} \mid path \in tree(P)\}\end{aligned}$$

Proof. Omitted \square

Proposition 3.6 (*ax-Bound on Dimensionality*).

$$\begin{aligned} \dim(P) &\leq 1 + \#\{ax\text{-}Rule \in \vdash P\} \\ \dim(P) &\leq 1 + \#\{vars \in P\} \end{aligned}$$

Proof. The first case is trivial. Given a formula P and a constructed sequent proof $\vdash P$ through coalescence, additive stratification ensures that $\dim(P) = n \implies leaves(P) \geq n$. As each of the n branches of the tree moves ‘upwards’, they must either terminate at an *ax*-rule or branch, leaving $n + 1$ branches total. Subsequently, for each *ax*-rule forming a leaf of the tree, $\dim(P)$ increases by no more than one. Finally, consider a base case such as $P := \vdash a \vee \neg a$ where $\dim(P) = 2 \leq 1 + \#\{ax\text{-}Rule \in \vdash P\}$ (in this case, it is equal).

Second case omitted □