Tuning Natural Deduction Proof Search by Analytic Methods.

Alexander Bolotov[1] Alexander Gorchakov[2]

[1] University of Westminster, W1W 6UW, London, UK, A.Bolotov@Wmin.ac.uk
[2] Laboratory of information systems in Humanites, Faculty of Philosophy, Lomonosov Moscow State University, 27-4, GSP-1, Moscow, Russian Federation, gorchakov@philos.msu.ru

Abstract: This paper is a result of the analysis of the efficiency of natural deduction proof search and the major weaknesses affecting it. We introduce new analytic strategies based on a new concept "Truth Set of Support". We present a combined proof search algorithm for classical propositional logic where a crucially new step is the guidance of the searching procedure by "Truth sets of Support" and establish the correctness. We describe the implementation of this new search technique and exemplify its advantages on the strong version of the Pigeon Hole Principle.

1 Introduction

The natural deduction (ND) proof search we optimise in this paper, was initially formulated for classical setting [2] and then extended to a number of logics – propositional linear-time temporal logic PLTL [1, 3], paracomplete [6] and paraconsistent [5]. Our recent work on the complexity of the method [4] and the implementation of the technique have shown that the method should be tuned to make proofs more efficient. In particular, we were interested in improving the performance of the algorithm on the class of formulae corresponding to the famous pigeon hole principle (PhP) which is often considered as an important 'testing' step for theorem provers. In this work we introduce new analytic strategies based on a new concept which we call "Auxiliary Truth Set (ATS)". We present a combined proof search algorithm for classical propositional logic where the searching procedure is guided by the ATS and establish the correctness. This technique has been implemented exemplifying its advantages on the strong version of the Pigeon Hole Principle.

2 Natural Deduction System

Figure 1 shows elimination (*el*) and introduction (*in*) rules.

Elimination Rules:

$$\wedge_{el_1}\frac{A\wedge B}{A} \quad \wedge_{el_2}\frac{A\wedge B}{B}$$

$$\neg_{el}\frac{\neg\neg A}{A} \quad \Rightarrow_{el}\frac{A\Rightarrow B \quad A}{B} \quad \vee_{el}\frac{A\vee B \quad \neg A(\neg B)}{B(A)}$$

Introduction Rules:

$$\vee_{in_1}\frac{A}{A\vee B} \quad \vee_{in_2}\frac{B}{A\vee B}$$

$$\wedge_{in}\frac{A \quad B}{A\wedge B} \quad \Rightarrow_{in}\frac{[C] \quad B}{C\Rightarrow B} \quad \neg_{in}\frac{[C] \quad B \quad \neg B}{\neg C}$$

Figure 1: ND Rules

If the conclusion of $\Rightarrow_{in}$ or $\neg_{in}$ is at step n, then $[C]$, where C is the most recent alive assumption, means that C is discharged and all formulae from C up to n are discarded.

An ND-derivation or inference of a formula B from a (possibly empty) set of assumptions Γ is a finite sequence of formulae $A_1,\ldots,A_n = B$ such that every A_i $(1 \leqslant i \leqslant n)$ is either an initial assumption or a conclusion of one of the rules applied to some preceding formulae. If a set of initial assumptions Γ is empty then B is a theorem.

3 New Proof Search Algorithm

The proof search algorithm is represented as a sequence of *algo-steps* $\Gamma \vdash \mathcal{G}$, where Γ is an ordered set of formulae in the proof and $\mathcal{G}$ is a stack of goals. The stack control mechanism is based on the LIFO principle. The proof commences with the initial task, of deriving goal g_0 from some given set of formulae $\Gamma = F_1, F_2, \ldots, F_m$ $(1 \leq m)$, abbreviated as $F_1, F_2, \ldots, F_m \vdash g_0$ (if $\Gamma = 0$, we have a task of proving g_0 as a theorem). Γ can be classified into the following six subsets:
Γ^{init} (initial assumptions in Γ), discarded formulae F^{disc}, formulae F^{el} - premises of the elimination rules, formulae F^{src} that generated new goals, auxiliary assumptions F^{assmp}, all other formulae $F^{poten} = F\backslash(F^{disc} \cup F^{in} \cup F^{el} \cup F^{src})$, where $F^{el} \cap F^{assmp} \neq \emptyset$.
A goal $g_i \in \mathcal{G}$, is reached iff

- if $g_i \neq \bot$ then g_i is *reached* iff $\exists f_i \in \Gamma$ such that $f_i = g_i$ and $f_i \notin F^{disc}$
- if $g_i = \bot$ then g_i is *reached* iff $\exists f_k, f_l$ such that $\{f_k, f_l\} \subset \Gamma$ and $f_l = \neg f_k$ and $f_k \notin F^{disc}$ and $f_l \notin F^{disc}$

If the current goal g_c is reached then it is deleted from $\mathcal{G}$ and the immediately preceding goal becomes our new current goal.

The heuristics are classified depending on the main logical connective of the goal.

(i) 'implication': If $g_c = A \Rightarrow B$ then F^{assmp} is updated with A and $\mathcal{G}$ is updated with $g_c = B$.

(ii) 'conjunction': If $g_c = A \wedge B$ then we set up goal $g_c = A$ (unless it has been already reached) and A needs to be reached before $g_c = B$ in the same fashion.

(iii) 'negation': If $g_c = p$ ($\neg p$) (for some literal p) then F^{asspm} is updated with $\neg p$ (p) and $\mathcal{G}$ is updated with $\bot$.

(iv) 'disjunction': If $g_c = A \vee B$ then $\mathcal{G}$ is updated with $g_c = A$ to be reached by the heuristics unless the 'negation' strategy is required. If A is not reachable then all formulae and goals introduced since $g_c = A$, are deleted and $g_c = B$ and the same process applies as for $g_c = A$. If $g_c = B$ is not reached, then, after all deletions, F^{asspm} is updated by $\neg(A \vee B)$, and $\mathcal{G}$ is updated by $g_c = \bot$. For the efficiency, we also add an auxiliary rule $\vee_{el_{aux}} \dfrac{\neg(A \vee B)}{\neg A \wedge \neg B}$

(v) First, we introduce the notion of the 'proof potential', which is $\phi = ((F^{assmp} \backslash F^{el}) \cap (F^{assmp} \backslash F^{disc})) \cup F^{poten}$ 'Auxiliary Truth Set (ATS)' applies when the potential of the proof $\phi \neq \emptyset$, the current goal, $g_c = \bot$ has been generated by the last assumption, f_n, and none of the rules or other heuristics is applicable. Now we split the set Γ into two sets: $\Gamma_2 = \{f_n\}$ and $\Gamma_1 = \Gamma \backslash \Gamma_2$ and set up the new goal called 'ATS-goal'=$\neg(\bigwedge_{i=1}^{n} f_i)$, where $f_i \in \Gamma_1 \backslash F^{disc}$. In other words, we set up the goal as the negation of conjunctions of all non-discarded formulae in the proof before f_n, which we now call 'ATS-assumption'. It is shown that a proof of 'ATS-goal' from f_n is necessary and sufficient condition for the presence of a contradiction in $\Gamma_1 \backslash F^{disc}$. Now we generate all sets of truth values that make 'ATS-goal' true. Then we check if there is a variable in 'ATS-goal' which does not occur in 'ATS-assumption' and if there is one we check if this variable is 'significant' for the 'ATS-goal'. This is the process of establishing if this variable takes both values - true and false - under the fixed values of variables of the 'ATS-assumption', in which case we eliminate this variable from the consideration. Alternatively, we conclude that 'ATS-goal' does not follow from the 'ATS-assumption' and terminate proof by claiming that the desired proof from the initial set of assumptions cannot be found. Now let $\phi = \phi^1 \cup \phi^2$, where $\phi^2 = f_n$ and $\phi^1 \cup \phi^2 = \emptyset$. Now we build ATS for ATS-assumption. For each of the groups of formulae from ϕ_2 with common variables we build their truth sets. If a group does not have a variable common with the ATS-goal then we do not consider it. Similarly, we check if a group that has variables common with the ATS-goal does not contain a variable not occurring in ATS-goal and delete this group if this is not true. Comparing ATS for ATS-assumption and ATS-goal, we check if every truth set of ATS-assumption contains at least one element of the truth set of ATS-goal. If we find a combination that violates this then we terminate the proof as we found the evaluation under which the formula given for the proof is false.

4 Experimental results: Pigeon Hole Principle under the new proof search.

To illustrate how the proposed proof search works we present here its performance on the Pigeon Hole Principle comparing with the original proof search for ND [2] and with Buss's proofs for this important for theorem provers class of testing formulae. Note that Buss's results are given for the DPLL with clause learning [7].

	ND	ND with ATS	DPLL with clause learning
PHP_2	93	27	-
PHP_3	740	60	5
PHP_4	7883	115	-
PHP_5	110509	198	-
PHP_6	1914985	315	129
PHP_7	-	472	-
PHP_8	-	675	769
PHP_{10}	-	1243	-
PHP_{12}	-	2067	20000

References

[1] A. Bolotov, A. Basukoski, O. Grigoriev, and V. Shangin. Natural deduction calculus for linear-time temporal logic. In *Joint European Conference on Artificial Intelligence (JELIA-2006)*, pages 56–68, 2006.

[2] A. Bolotov, V. Bocharov, A. Gorchakov, and V. Shangin. Automated first order natural deduction. In *Proceedings of IICAI*, pages 1292–1311, 2005.

[3] A. Bolotov, O. Grigoriev, and V. Shangin. Automated natural deduction for propositional linear-time temporal logic. In *Temporal Representation and Reasoning, 14th International Symposium on*, pages 47–58, June 2007.

[4] A. Bolotov, D. Kozhemiachenko, and V. Shangin. Paracomplete logic Kl: natural deduction, its automation, complexity and applications. *IfCoLog Journal of Logics and their Applications.*, 5:221–261, 2018.

[5] A. Bolotov and V. Shangin. Natural deduction system in paraconsistent setting: Proof search for PCont. *JJournal of Intelligent Systems*, 21:1–24, 2012.

[6] A. Bolotov and V. Shangin. Tackling incomplete system specifications using natural deduction in the paracomplete setting. In *2014 IEEE 38th Annual Computer Software and Applications Conference*, pages 91–96, July 2014.

[7] S. Buss. Towards NP-P via proof complexity and search. In *Annals of Pure and Applied Logic*, volume 163, pages 906–917, 2012.