Title	Tractability of Cut-free Gentzen Type Propositional Calculus with Permutation Inference
Author(s)	Arai, Noriko H.
Citation	数理解析研究所講究録 (1996), 950: 181-188
Issue Date	1996-05
URL	http://hdl.handle.net/2433/60322
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
950巻 1996年 181-188

Tractability of Cut-free Gentzen Type Propositional Calculus with Permutation Inference

広島市立大学 新井 紀子 (Noriko H. Arai)

March 4, 1996

1 Introduction

One of the most fundamental problems of the complexity theory and the automated reasoning theory is to find an efficient proof system for propositional calculus which is applicable for automated reasoning. The statement contains two intuitive concepts. First, we have to make it clear what the notion "efficient" means. There is a wide spread understanding that polynomial time computability is the correct mathematical model of feasible computation. According to the opinion, truly "effective" system must have a polynomial size, $p(n)$ proof for every tautology of size n. In [5], Cook and Reckhow named such a system, a *super system*. They showed that if there exists a super system, then $NP = co - NP$; many people are highly skeptical about the validity of this equality. Secondly, we have to have some criteria for propositional calculi to be applicable for automatic theorem proving. Intuitively, we say that tautologies are automatically proved when we can construct a deterministic machine which says yes if the input is a tautology and says no otherwise. If we interpret our goal most strictly, we have to obtain a sound proof system which proves any tautology polynomially and the construction of the proof is completely determined by the structure of the tautology. Then obviously $P = NP$ is necessary.

How can we relax our criteria so that it is theoretically meaningful but still practical? One fairly natural approach is to give up to prove every tautology polynomially but confine ourselves to "familiar" tautologies.

Gentzen's Hauptsatz suggests us that cut-free Gentzen type sequent calculus is one of the most reasonable systems to be applied to automatic reasoning: we can obtain a proof-tree automatically for any given tautology. Furthermore, the construction procedure can be determined solely by the structure of the given tautology. However, it is already known that the number of steps required in the search procedure increases exponentially with the length of inputs[9]. Resolution is another propositional calculus which is frequently mentioned in automatic theorem proving. It is also known that there are sequences of tautologies which require exponential size proofs[7]. Unfortunately, the hard examples for cut-free Gentzen system or for resolution are not rare nor pathological, but they are rather commonly found combinatorial problems[10].

We suggest another possible approach; if it is too much to ask to construct a deterministic machine accepting tautologies in polynomial time, it is worth trying to construct a nondeterministic machine but the chance to obtain a sound proof for a given tautology is relatively high. Gentzen system with cut-rule and Frege system are known to be strictly more powerful system than resolution[7],[3]. However, they do not satisfy the subformula property: the existence of cut-rule and modus-ponens allows unpredictable formulas to coming into proofs. As a result, chance to obtain appropriate proofs by machine is very low even for simple tautologies. On the contrary, if a system satisfies the subformula property, the bound for search will be relatively limited.

It is sensible to note that many hard examples for propositional calculus such as pigeonhole principles are originally first-order sentences. Translating them into propositional formulas, these propositions share an evident similarity, symmetries. If we can express as an inference rule that a tautology remains invariant under permutation of variables, proofs of propositions of this kind can be shortened dramatically[2].

In this paper, we introduce a new inference rule to play the role: *permutation rule.* We first show that a cut-free Gentzen type sequent calculus plus permutation, called GCNF'+permutation, satisfies the subformula property. Then, we show that the system have polynomial size proofs for both the pigeonhole principle and the k-equipartition.

2 Gentzen system GCNF'

Definition 1

Resolution proves a formula to be a tautology by showing that its negation, which is put into conjunctive normal form, is unsatisfiable.

A *propositional variable* is denoted by p, q, r, x. Each propositional variable has a conjugate (or negation) denoted by $\overline{p}$. Also $\overline{\overline{p}} = p$. A *literal* is a propositional variable p or a conjugate $\overline{p}$. A *clause* is a finite set of literals, where the meaning of the clause is the disjunction of the literals in the clause. For example $\{p_1, \overline{p}_2, p_3\}$ means $p_1 \vee \overline{p}_2 \vee p_3$.

Resolution has no axiom. It has only one inference rule called *resolution rule*

$$\textit{resolution rule} \qquad \frac{C_1 \cup \{x\} \quad C_2 \cup \{\overline{x}\}}{C_1 \cup C_2}$$

When we try to show that a set of clauses **C** is unsatisfiable, we take **C** to be a set of hypotheses to which we apply the resolution rule until we obtain the empty clause.

GCNF' is a variant of cut-free Gentzen system introduced by Gallier (see page 120 of [6]). It is also a refuting system.

A *cedent* is a finite set of clauses, expressed as a sequence of clauses punctuated by commas. The meaning of a cedent is the conjunction of the clauses in the cedent. For example $C_1, C_2, \ldots, C_n$ means $C_1 \wedge C_2 \wedge \ldots \wedge C_n$. We use capital Greek letters Γ, Δ, Π for cedents. The semantics of cedents implies that a cedent $C_1, \ldots, C_n$ is false iff the formula $C_1 \wedge \ldots \wedge C_n \supset \perp$ is valid.

$$\textit{axioms} \qquad p, \overline{p}$$

$$\textit{structural inference} \qquad \frac{\Gamma}{\Gamma, \Delta}$$

$$\textit{logical inference} \qquad \frac{\Gamma, C_1, \ldots, C_k \quad \Pi, l}{\Gamma \cup \Pi, C_1 l, \ldots, C_k l}\,(l)$$

l is an arbitrary literal, which is called the *auxiliary literal* of this inference.

It is fairly easy to show the soundness and the completeness of GCNF'(see chapter 4 in [6].)

Proposition 1

GCNF' is sound and complete.

Now we define a scale to measure the efficiency of a proof system.

Definition 2

1. Let S be a proof system which is sound and complete, and let P be a proof system of S. The *size* of P is the number of all the symbols used in P, that is denoted by $size(P)$.

2. Let S_1 and S_2 be proof systems for propositional calculus. S_1 *p-simulates* S_2 iff there exists a polynomial function p such that for any formula f and any proof P_2 of f in S_2, there exists a S_1-proof P_1 of f (translated into S_1 language) so that

In the following argument, we understand proofs of GCNF' or resolution to be in DAG form. If P is a GCNF' (resolution) proof, then $size(P)$ means the number of symbols appearing in different cedents (clauses) in P. Now we examine hard examples for GCNF'. Haken showed an exponential lower bound for resolution in [7]: he proved that there exists a constant c , $c > 1$ so that, for sufficiently large n, every resolution refutation of the pigeonhole principle (PHP_n) contains at least c^n different clauses. Ajtai showed in [1] a superpolynomial lower bound for constant depth Frege proofs for the pigeonhole principle, and later showed a superpolynomial lower bound for constant depth Frege proofs for $2-$*Equipartition* even assuming the pigeonhole principle. Their proofs can be translated to prove a superpolynomial lower bound for GCNF'.

Definition 3 (Pigeonhole principle)

The pigeonhole principle states that for each n, if $f : \{0,\dots,n\} \to \{0,\dots,n-1\}$ then f is not one-to-one.

For each i and j with $0 \le i \le n$ and $0 \le j \le n-1$ we will have the variable $p_{i,j}$ which 'means' $f(i) = j$.

$$PHP_n \qquad \bigwedge_{0\le i\le n}\ \bigvee_{0\le j\le n-1} p_{i,j},\ \bigwedge_{0\le i<m\le n}\ \bigwedge_{0\le j\le n-1} (\overline{p}_{i,j}\overline{p}_{m,j})$$

$\bigvee_{0\le i\le n} p_i$ is an abbreviation for the clause $p_0,\dots,p_n$. $\bigwedge_{0\le i\le n} C_i$ is an abbreviation for the cedent $C_0,\dots,C_n$.

The number of all literals contained in PHP_n is $n^3 + 2n^2 + n$.

Definition 4 (k-equipartition)

The k-equipartition states that if an integer n is not evenly divisible by k, then there is no partition of $\{1,\dots,n\}$ into disjoint sets of size k.

Let $J_n^k = \{(j_1,\dots,j_k) : 1 \le j_1 < \dots j_k \le n\}$ For $\vec{j} \in J$, we write $i \in \vec{j}$ to mean that there exists $1 \le l \le k$ such that $i = j_l$. Suppose that $n \ne 0(mod\ k)$. We introduce new variables $x_{i,(j_1,\dots,j_k)}$ for $1 \le i, j_1,\dots,j_k \le n$ to mean that $(j_1,\dots,j_k)$ is a partition of $\{1,\dots,n\}$ and $i \in \{j_1,\dots,j_k\}$.

$k - Eq(n)$ is defined as the following cedent;

$$\bigwedge_{1\le i\le n}\ \bigvee_{\vec{j}\in J_n^k, i\in\vec{j}} x_{i,\vec{j}},\quad \bigwedge_{\vec{j}\in J_n^k,\ i_1,i_2\in\vec{j},\ i_1\ne i_2} (\overline{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\quad \bigwedge_{\substack{j_1,\vec{j}_2\in J_n^k\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\ne\vec{j}_2}} (\overline{x}_{i,\vec{j}_1}\overline{x}_{i,\vec{j}_2})$$

The number of all literals contained in $k - Eq(n)$ is
$n\begin{pmatrix} n-1 \\ k-1 \end{pmatrix} + 2\begin{pmatrix} n \\ k \end{pmatrix}\begin{pmatrix} k \\ 2 \end{pmatrix} + n\begin{pmatrix} n-1 \\ k-1 \end{pmatrix}^2 - n\begin{pmatrix} n-1 \\ k-1 \end{pmatrix}.$
The first $\bigwedge$ of clauses expresses that "each i is contained in some partition whose size is k." The second $\bigwedge$ of clauses expresses that "if $(i_1, \ldots, i_k)$ is a partition containing i_1, then it is also a partition containing i_2,... and i_k." The last $\bigwedge$ of clauses means that "if $i_s = j_t$ for some $1 \leq s \leq k$ and $1 \leq t \leq k$ and if $(i_1, \ldots, i_k) \neq (j_1, \ldots, j_k)$, then either $(i_1, \ldots, i_k)$ or $(j_1, \ldots, j_k)$ is not a partition."
(Note: The definition given above is slightly different from the formulation given in [4], but they are equivalent.)

Proposition 2 (Haken [7])

There exists a constant c , $c > 1$ such that, for sufficiently large n, every GCNF' refutation of PHP_n contains at least c^n different cedents.

Proposition 3 (Ajtai [1])

There exists a constant c , $c > 1$ so that, for sufficiently large n, every GCNF' refutation of $k - Eq(n)$ contains at least c^n different cedents.

We introduce new inference rules, called *renaming*, *restricted renaming* and *permutation.*

$$\textit{renaming} \qquad \frac{\Gamma}{\Gamma(p \to q)}\ p \to q$$

$\Gamma(p \to q)$ is obtained by replacing every occurrence of p by q in Γ.

$$\textit{restricted renaming} \qquad \frac{\Gamma}{\Gamma(p \Rightarrow q)}\ p \Rightarrow q$$

$\Gamma(p \Rightarrow q)$ is obtained by replacing every occurrence of p in Γ by a variable q, which does not appear in Γ.

$$\textit{permutation} \qquad \frac{\Gamma(p_1, \ldots, p_m)}{\Gamma(\pi(p_1), \ldots, \pi(p_m))}\ \pi$$

π is a permutation on $\{p_1, \ldots, p_m\}$ and $\Gamma(\pi(p_1), \ldots, \pi(p_m))$ is the result of replacing every occurrence of p_i ,$1 \leq i \leq m$ in $\Gamma(p_1, \ldots, p_m)$ by $\pi(p_i)$.

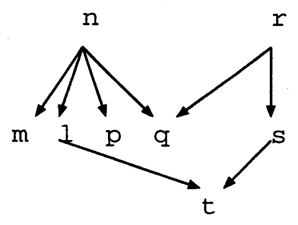
It is straightforward to show that GCNF'+restricted renaming p-simulates GCNF'+permutation.

Proposition 4

GCNF'+restricted renaming p-simulates GCNF'+permutation.

In general, GCNF'+permutation does not satisfy the subformula property. However, one can translate a given GCNF'+permutation refutation into a GCNF'+permutation refutation satisfying the subformula property without increasing its size too much. Before we start, we need some definitions.

Definition 5



Let D be a directed acyclic graph. Suppose that n, m are nodes appearing in D. When m appears below n and no other node appears between n and m, we say that m is a *son* of n. When $n_1, \ldots, n_k$ are the sons of n, and when n_1 is the leftmost occurrence among them, we say that n_1 is the *direct son* of n. $n_2, \ldots, n_k$ are called *nondirect sons* of n. A sequence of nodes $m_1, \ldots, m_l$ is called a *direct line* of n_l in D when n_1 is either a leaf or a nondirect son of a node in D, and every n_i for $1 < i \leq l$ is the direct son of n_{i-1}.

In the following, we frame a 2-dimensional image of directed acyclic graphs so that we can fix the order of right and left of nodes.

Theorem 1 (Subformula property of GCNF'+permutation)

Let P be a GCNF'+permutation refutation of $C_1, \ldots, C_n$. Then, there exists P', a refutation of $C_1, \ldots, C_n$ such that $size(P') = O(size(P)^3)$ and P' satisfies the subformula property; every clause $C = l_1 \ldots l_m$ appearing in P' is a subformula of one of $C_1, \ldots, C_n$.

(proof)

We shall transform P into P' inductively from the bottom to the top.

Suppose that n is a node in P. Let $n_1, \ldots, n_l$ are the list of sons of n. Suppose that n_1 is the direct son of n. When

$$\frac{n}{n_1}$$

is weakening, no change is made. If

$$\frac{n \quad m}{n_1}$$

is a logical inference, no change is made. Suppose that the inference between n and n_1 is permutation, say

$$\frac{\Gamma(p_1, \ldots, p_m)}{\Gamma(\pi(p_1), \ldots, \pi(p_m))}\ \pi$$

Then, replace every occurrence of p_i by $\pi(p_i)$ $(1 \leq i \leq m)$ in each cedent on every direct line containing the upper cedent , $\Gamma(p_1, \ldots, p_m)$. The result may fail to be a GCNF'+permutation refutation: there may exist a gap between a node and its nondirect son. Suppose that n in P is replaced by n', and its nondirect son n_k is replaced by n'_k. Suppose that the inference between n and n_k is a permutation. Note that a product of permutations is again a permutation. Hence,

$$\frac{n'}{n'_k}$$

is a sound permutation inference. Suppose that the inference between n and n_k is either structural or logical, then insert one permutation inference necessary. Now we obtain a sound GCNF'+permutation refutation, P'.

We show that P' satisfies the subformula property by induction on the construction of P'. Let m be a node in P'. Let m_1 be the direct son of m. Then, by the induction hypothesis, m_1 satisfies the subformula property. The inference between m and m_1 is either a logical inference, structural inference, or a special kind of restricted renaming, which is

$$\frac{\Gamma}{\Gamma}.$$

Hence, m also satisfies the subformula property.

We remark that a close examination of the proof of theorem 2 gives us a polynomial algorithm to translate a GCNF'+permutation refutation to GCNF'+permutation which satisfies the subformula property.

A resolution refutation R is called *regular* iff for every resolution

$$\frac{C_1 \cup \{x\} \quad C_2 \cup \{\overline{x}\}}{C_1 \cup C_2}\,(I)$$

appearing in R, no resolution of the form,

$$\frac{D_1 \cup \{x\} \quad D_2 \cup \{\overline{x}\}}{D_1 \cup D_2}$$

appears below I. This notion was introduced by Tseitin [8]. He proved that regular resolution is not super before Haken's work. By analogy, we say a GCNF' (or GCNF'+permutation) refutation P is *regular* iff for every logical inference I whose auxiliary literal is l in P, no logical inference having the same auxiliary literal l appears below I.

We show that regular GCNF'+permutation has polynomial size refutations for PHP_n and $k - Eq(n)$.

Theorem 2

There exists a regular GCNF'+permutation refutation of PHP_n whose *size* $\leq O(n^6)$.
(proof)

Assume that we already have a regular GCNF'+permutation refutation P_{n-1} of

$$\bigvee_{j=0}^{n-2} p_{0,j}, \ldots, \bigvee_{j=0}^{n-2} p_{n-1,j}, \quad \bigwedge_{0\leq i<m\leq n-1} \; \bigwedge_{0\leq j\leq n-2} (\overline{p}_{i,j}\overline{p}_{m,j})$$

such that $size(P_{n-1}) \leq O((n-1)^6)$. We supplement some lines below P_{n-1} to obtain $P_{n,n-1}$.First, we add a logical inference of which auxiliary literal is $p_{n-1,n-1}$.

$$\frac{\bigvee_{j=0}^{n-2} p_{0,j}, \ldots, \bigvee_{j=0}^{n-2} p_{n-1,j}, \quad \bigwedge_{0\leq i<m\leq n-1} \; \bigwedge_{0\leq j\leq n-2} (\overline{p}_{i,j}\overline{p}_{m,j}) \qquad \overline{p}_{n-1,n-1}, p_{n-1,n-1}}{\overline{p}_{n-1,n-1}, \bigvee_{j=0}^{n-2} p_{0,j}, \ldots, \bigvee_{j=0}^{n-2} p_{n-2,j} \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0\leq i<m\leq n-1} \; \bigwedge_{0\leq j\leq n-2} (\overline{p}_{i,j}\overline{p}_{m,j})}\,(p_{n-1,n-1})$$

Similarly, add logical inferences whose auxiliary literals are $p_{n-2,n-1}, \ldots, p_{0,n-1}$, and whose right upper cedents are axioms. Then, we get

$$\overline{p}_{0,n-1}, \ldots, \overline{p}_{n-1,n-1}, \bigvee_{j=0}^{n-1} p_{0,j}, \ldots, \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0\leq i<m\leq n-1} \; \bigwedge_{0\leq j\leq n-2} (\overline{p}_{i,j}\overline{p}_{m,j})$$

This refutation graph is called $P_{n,n-1}$. The last cedent means that "for all $0 \leq k \leq n-1$, the k-th pigeon sits in one of the hole,$0, \ldots, n-1$. At the same time, the pigeon does not sit the (n-1)-th hole." Define a permutation π_k by a product of $(n-1)$ transpositions,

$$(p_{0,n-1} \quad p_{0,k}) \cdots (p_{n-1,n-1} \quad p_{n-1,k})$$

for all $0 \leq k \leq n-2$. To obtain $P_{n,k}$, for each $0 \leq k \leq n-2$ add one permutation inference;

$$\frac{\overline{p}_{0,n-1}, \ldots, \overline{p}_{n-1,n-1}, \bigvee_{j=0}^{n-1} p_{0,j}, \ldots, \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0 \leq i < m \leq n-1} \bigwedge_{0 \leq j \leq n-2} (\overline{p}_{i,j}\overline{p}_{m,j})}{\overline{p}_{0,k}, \ldots, \overline{p}_{n-1,k}, \bigvee_{j=0}^{n-1} p_{0,j}, \ldots, \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0 \leq i < m \leq n-1} \bigwedge_{0 \leq j \leq n-1, j \neq k} (\overline{p}_{i,j}\overline{p}_{m,j}))} \; \pi_k$$

For each $0 \leq k \leq n-1$, we add a logical inference;

$$\frac{\overline{p}_{0,k}, \ldots, \overline{p}_{n-1,k}, \bigvee_{j=0}^{n-1} p_{0,j}, \ldots, \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0 \leq i < m \leq n-1} \bigwedge_{0 \leq j \leq n-1, j \neq k} \overline{p}_{i,j}\overline{p}_{m,j} \qquad p_{n,k}, \overline{p}_{n,k}}{p_{n,k}, \bigvee_{j=0}^{n-1} p_{0,j}, \ldots, \bigvee_{j=0}^{n-1} p_{n-1,j}, \quad \bigwedge_{0 \leq i < m \leq n-1} \bigwedge_{0 \leq j \leq n-1, j \neq k} \overline{p}_{i,j}\overline{p}_{m,j}, \bigwedge_{0 \leq i \leq n-1} \overline{p}_{n,k}\overline{p}_{i,k}} \; (\overline{p}_{n,k})$$

Combine these together by applying $n-1$ logical inferences to obtain P_n of PHP_n. $P_{n-1}$ is regular by the induction hypothesis, so is P_n.

$$size(P_n) \leq (len(P_{n-1}) + 2n + 2(n-1))(n + 1 + n^2(n+1)/2) \leq o(n^6).$$

We can also prove the following theorem.

Theorem 3

There exists a polynomial function $p$, independent from $n$, and a regular GCNF'+permutation refutation of $k - Eq(n)$ whose size $\leq p(n)$.

Corollary 1

Resolution does not p-simulate GCNF'+permutation.

Corollary 2

Bounded depth Frege systems do not p-simulate GCNF'+permutation.

References

[1] M.Ajtai, "The complexity of the pigeonhole principle", *29th Annual Symposium on the Foundations of Computer Science* (1988), 346-55.

[2] B. Benhamou and L. Sais, "Tractability through symmetries in propositional calculus", *J. Automated Reasoning*, Vol.12 (1994), 89-102.

[3] S. R. Buss, "Polynomial size proofs of the pigeonhole principle", *J. Symbolic Logic*, Vol.52 (1987), 916-27.

[4] P. Clote, "On polynomial size Frege proofs of certain combinatorial principles", in *Arithmetic, Proof Theory, and Computational Complexity*, Clarendon Press, Oxford (1993), 162-84.

[5] S. A. Cook and R. A. Reckhow, "The relative efficiency of propositional proof systems", *J. Symbolic Logic*, Vol.44 (1979), 36-50.

[6] J. Gallier, *Logic for Computer Science*, (John Wiley &Sons, New York, 1987).

[7] A. Haken, "The intractability of resolution", *Theoretical Computer Science*, Vol.39 (1985), 297-308.

[8] G. S. Tseitin, "On the complexity of derivation in propositional calculus", *Studies in Mathematics and Mathematical Logic Part 2*, 1968,V. A. Steklov Math. Institute.

[9] A. Urquhart, "The complexity of Gentzen systems for propositional logic", *Theoretical Computer Science*, Vol.66 (1989), 87-97.

[10] A. Urquhart, "Hard examples for resolution", *J. Assoc. Comput. Mach.*, Vol.34 (1987) 209-219.