ELSEVIER

Theoretical Computer Science 243 (2000) 185–197

Theoretical
Computer Science

www.elsevier.com/locate/tcs

with permutation inference II ☆

Noriko H. Arai

Department of Computer Science, Hiroshima City University, 151 Ozuka,
Asaminami-ku, Hiroshima 731-31, Japan

Received September 1997; accepted June 1998
Communicated by M. Nivat

Abstract

In Arai (1996), we introduced a new inference rule called *permutation* to propositional calculus and showed that cut-free Gentzen system LK (GCNF) with permutation (1) satisfies the feasible subformula property, and (2) proves pigeonhole principle and k-equipartition polynomially. In this paper, we survey more properties of our system. First, we prove that cut-free LK+permutation has polynomial size proofs for nonunique endnode principle, Bondy's theorem. Second, we remark the fact that permutation inference has an advantage over renaming inference in automated theorem proving, since GCNF+renaming does not always satisfy the feasible subformula property. Finally, we discuss on the relative efficiency of our system vs. Frege systems and show that Frege polynomially simulates GCNF+renaming if and only if Frege polynomially simulates extended Frege. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Automated theorem proving; Computational complexity; Proof theory

1. Preliminaries

We usually deal with a mass of objects in combinatorics; n pigeons, n different rows of 0's and 1's, etc. When one proves a combinatorial theorem in the setting of propositional calculus, he/she first has to translate it into a series of propositional formulas. The base step of the translation is to, informally, enumerate the objects. The pigeonhole principle gives us a good example. It states that there is no one-to-one mapping from $(n+1)$ objects to n objects. Ordinal numbers from 0 to n are given to identify objects in the domain and the range. The situation of the ith object mapped to the jth object, or $f(i)=j$, is expressed as a new propositional variable

☆ The work is partly supported by the Ministry of Education of Japan under Joint Research Grant-in-Aid for International Scientific Research FM-ISEE.

E-mail address: arai@cs.hiroshima-cu.ac.jp (N.H. Arai).

PII: S0304-3975(98)00186-8

$p_{i,j}$. Accordingly, the statement "the mapping is not one-to-one" is translated to the disjunction of $f(i)=f(j)=h$ $(i \neq j)$, in which no specific i or j play any special role and they are interchangeable. We are now ready to obtain the propositional pigeonhole principle, which is

$$PHP_n \bigwedge_{0 \leqslant i \leqslant n} \bigvee_{0 \leqslant j \leqslant n-1} p_{i,j} \rightarrow \bigvee_{0 \leqslant i < m \leqslant n} \bigvee_{0 \leqslant j \leqslant n-1} (p_{i,j} \wedge p_{m,j})$$

$\bigvee_{0 \leqslant i \leqslant n} A_i$ is an abbreviation for the formula $A_0 \vee \cdots \vee A_n$. $\bigwedge_{0 \leqslant i \leqslant n} C_i$ is an abbreviation for the formula $C_0 \wedge \cdots \wedge C_n$. Note that PHP_n is closed under some permutations (a subset of S_n), as most of propositional combinatorial statements are.

An elementary proof of the pigeonhole principle uses mathematical induction on the number n of objects in the domain; we assume that the pigeonhole principle holds for n, and show that it also holds for $n+1$. Let f be a mapping from $\{0,\ldots,n+2\}$ to $\{0,\ldots,n+1\}$. Without loss of generality, we can assume that $f(n+2)=n+1$. If there exists an $i \neq n+2$ such that $f(i)=n+1$, we are done. Suppose otherwise. Then the function f restricted to $\{0,\ldots,n+1\}$ is a mapping to $\{0,\ldots,n\}$. By the induction hypothesis, it is not one-to-one, and so is not f (q.e.d.). The novelty of this proof is the line, "Without loss of generality …". Here, we understand that the situation of $f(n+2)=i$ $(i=0,\ldots,n)$ is merely a variant of the situation of $f(n+2)=n+1$; we save time by representing (exponentially) many cases by just one case.

In [2], we showed that the inference rule, *permutation*, enables cut-free LK to imitate this elementary proof line by line, which gives polynomial-size propositional proofs for PHP_n. It is an interesting question to ask whether it is always the case: viewing a combinatorial theorem as a disjunction of (exponentially) many cases, are they always reduced to several typical cases?

Checking proofs of theorems of combinatorics closely, we find that not only "without-loss-of-generality" argument but also arithmetical techniques are involved in the reasoning, such as counting the number of objects. Hence, it is equivalent to ask if these arithmetical arguments are removable without increasing the size of the proof significantly.

This question is closely related to three fundamental questions in the theory of computation.

The first question is in the theory of automated reasoning: what kind of mathematical problem is automatically solvable in polynomial time? Cut-free LK with permutation is known to satisfy the *feasible subformula property*, which means that if P is a cut-free LK+permutation proof of a theorem T, then one can assume that any formula appearing in P is a subformula of T. Or even stronger, any line (*sequent*) in P expresses a 'subcase' of T. Accordingly, the range of proof-search is quite limited compared to other powerful proof systems such as Frege. By virtue of its subformula property, cut-free LK+permutation is ready to be implemented for automated reasoning. At the same time, we have experienced that cut-free LK+permutation is quite efficient on tautologies which are closed under permutations. Hence, we hope, a wide range of universal combinatorial principles which are closed under S_n can be automatically provable efficiently through implementation of cut-free LK+permutation.

We can find two other questions in the field of propositional proof complexity.

It is a classical result by Gentzen [15] that any tautology can be proved in LK without using any cut inferences. However, it does not guarantee that one can remove cut inferences from a given proof in short time. It is well-known that cut inferences, of even restricted complexity, are not removable in polynomial time [10, 17, 3], but it is not known if it is also the case for LK+renaming, which is polynomially equivalent to extended Frege. Here, our question can be generalized as follows: is a superpolynomial function required to carry out cut-elimination for LK+renaming? We conjecture that it is so, or even stronger that cut-free LK+renaming does not polynomially simulate Frege.

Frege is known to have an ability to express NC^1 concepts. In NC^1, we can deal with elementary arithmetic. As suggested in [8], the base step for the translation of an arithmetical statement to a series of propositional formulas is to encode an integer of length n into a vector of n 0's and 1's, and a free variable of length n into a vector of n propositional variables; p_i represents the ith digit of a free variable a. As a result, p_i and p_j with $i \neq j$ have different "weight" in the obtained propositional formula, and usually they are not interchangeable. For example, a statement of $x_0 = x_1 + x_2$ can be translated to

$$Add_\rho(\vec{\phi}^0, \vec{\phi}^1, \vec{\phi}^2) = \bigwedge_{1 \leqslant i \leqslant \rho} \left((\phi_0^0 \leftrightarrow \phi_0^1 \oplus \phi_0^2) \right.$$
$$\left. \wedge \left(\phi_i^0 \leftrightarrow \phi_i^1 \oplus \phi_i^2 \oplus \bigvee_{0 \leqslant j < i} \left(\phi_j^1 \wedge \phi_j^2 \wedge \bigwedge_{j < k < i} (\phi_k^1 \oplus \phi_k^2) \right) \right) \right),$$

where $\vec{\phi}^l = \phi_0^l, \ldots, \phi_\rho^l$ $(0 \leqslant l \leqslant 2)$ with propositional variables ϕ_k^l's. Frege polynomially proves elementary arithmetical statements, such as the associativity of addition, but it is questionable whether cut-free LK+renaming does.

It is also a fundamental question in propositional proof complexity whether or not Frege system can efficiently simulate Frege system with extension rule (extended Frege system). In Section 3, we show that Frege system polynomially simulates extended Frege system if and only if it polynomially simulates cut-free LK+renaming.

Definition 1. A finite (possibly empty) sequence of formulas are called a *cedent*. Cedents are usually denoted by capital Greek letters. An ordered pair of cedents written in the form

$$A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$$

is called a *sequent*, where $A_1, \ldots, A_n$ is called an *antecedent* and $B_1, \ldots, B_m$ *succedent*. The intuitive meaning of a sequent of the form $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ is $A_1 \wedge \cdots \wedge A_n \rightarrow B_1 \vee \cdots \vee B_m$. When the succedent is empty, then it simply means that from the set of assumptions $A_1, \ldots, A_n$, we get a contradiction.

Definition 2. A *cut-free LK proof* is a sequence of sequents in which every sequent is an *initial sequent* of the form, $p \to p$ (p is a variable) or derived from previous sequents by one of following *inference rules*.

1. Structural rule:

$$\frac{\Gamma \to \Delta}{\Gamma^* \to \Delta^*}$$

where $\Gamma^* \supseteq \Gamma$ and $\Delta^* \supseteq \Delta$ as sets.

2. $\neg$-*left*:

$$\frac{\Gamma \to \Delta, A}{\neg A, \Gamma \to \Delta}$$

3. $\neg$-*right*:

$$\frac{A, \Gamma \to \Delta}{\Gamma \to \Delta, \neg A}$$

4. $\wedge$-*left*:

$$\frac{A, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta} \quad \text{and} \quad \frac{B, \Gamma \to \Delta}{A \wedge B, \Gamma \to \Delta}$$

5. $\wedge$-*right*:

$$\frac{\Gamma \to \Delta, A \, \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}$$

6. $\vee$-*left*:

$$\frac{A, \Gamma \to \Delta \, B, \Gamma \to \Delta}{A \vee B, \Gamma \to \Delta}$$

7. $\vee$-*right*:

$$\frac{\Gamma \to \Delta, A}{\Gamma \to \Delta, A \vee B} \quad \text{and} \quad \frac{\Gamma \to \Delta, B}{\Gamma \to \Delta, A \vee B}$$

We define the notions of *ancestors*, *descendants* and so on as usual [18].

Definition 3. A *literal* is a propositional variable p or a conjugate $\bar{p}$. A *clause* is a finite set of literals, where the meaning of the clause is the disjunction of the literals in the clause. A finite set of clauses is called a *cedent*.

When we restrict our interest to conjunctive normal form formulas, two inference rules are extracted out of nine to fulfill the requirements. The part of cut-free LK for conjunctive normal forms is called *GCNF*.

Definition 4. *GCNF refutation* is a sequence of cedents in which every sequent is an *initial sequent* of the form, $p, \bar{p}$ or derived from previous cedents by one of following *inference rules*:

structural inference $\dfrac{\Gamma}{\Gamma, \Delta}$

logical inference $$\frac{\Gamma, C_1, \dots, C_k \ \ \Pi, l}{\Gamma \cup \Pi, C_1 l, \dots, C_k l}(l)$$

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

Now we introduce new inference rules, *renaming* and *permutation*, to cut-free LK and GCNF.

renaming $$\frac{\Gamma}{\Gamma(q/p)}(q/p)$$

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

permutation $$\frac{\Gamma(p_1, \dots, p_m)}{\Gamma(\pi(p_1)/p_1, \dots, \pi(p_m)/p_m)}\pi$$

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

Γ is either a sequent or a cedent according to the context.

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

Definition 5. (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 *simulates* S_2 if and only if there exists a polynomial function p such that for any formula A and any proof P_2 of A in S_2, there exists an S_1-proof P_1 of A (translated into S_1 language) so that

$$size(P_1) \leqslant p(size(P_2)).$$

In other words, a system S_1 simulates S_2 if S_1 is not less efficient than S_2 as a proof system.

(3) In particular, we say that S_1 *polynomially simulates* (*p-simulates*) S_2 if there is a polynomial-time algorithm which, given an S_2-proof of a formula A, produces an S_1-proof of A.

Note that GCNF in tree form and resolution in tree form polynomially simulate each other.

2. Short proofs without using a cut

Cut-free LK with permutation is suitable for proving combinatorial theorems since combinatorial statements put into series of propositional formulas are usually closed under (some) permutations. Pigeonhole principles and mod-k principles are counted among hard examples for bounded depth Frege [1], though GCNF+permutation proves them rather easily [2]. One may speculate that GCNF+permutation (or cut-free LK+

permutation) polynomially proves non-unique endnode principle and Bondy's theorem observing that they are equivalent to or weaker than mod-2 principle by constant depth polynomial size Frege proofs [6, 4, 7]. In general, such equivalence does not promise the existence of polynomial-size cut-free LK+permutation proofs of the equivalents. However, the odds are on our side in these cases. In this section, we show that cut-free LK+permutation does polynomially proves non-unique endnode principle and Bondy's theorem.

2.1. Non-unique endnode principle

The *non-unique endnode principle* is a statement on graphs. Suppose that G is a finite simple undirected graph such that any edge x in G has at most 2 edges adjacent to x. Then, G cannot have a unique endnode. According to the fomalization given in [11], the non-unique endnode principle with vertex set $\{1,\dots,n\}$ is translated into a propositional sequent, $ENDNODE_n$, given by $\Gamma \to \Box$ where Γ is the cedent consisting of (1)–(6) and $\Box$ is an empty cedent.

1. $\neg r_{i,i}$ for all $1 \leqslant i \leqslant n$.
2. $\neg r_{i,j} \vee r_{j,i}$ for all $1 \leqslant i,j \leqslant n$.
3. $\bigvee_{1 \leqslant j \leqslant n} r_{j,n}$.
4. $\neg r_{j,n} \vee \neg r_{j',n}$ for all $1 \leqslant j < j' < n$.
5. $\bigvee_{1 \leqslant j < j' < n} (r_{i,j} \wedge r_{i,j'})$ for all $1 \leqslant i < n$.
6. $\neg r_{i,j} \vee \neg r_{i,j'} \vee \neg r_{i,j''}$ for all $1 \leqslant i < n$ and $1 \leqslant j < j' < j'' \leqslant n$.

Note that the vertex n is meant to be the unique endnode.

Lemma 1. *If P is a cut-free LK+permutation proof of $A \vee B, \Gamma \to \Delta$, then there exist cut-free LK+permutation proofs P_1 of $A, \Gamma \to \Delta$ and P_2 of $B, \Gamma \to \Delta$ with size$(P_i) <$ size(P) and len$(P_i) <$ len(P) for $i = 1, 2$.*

Proof. Find all the direct ancestors of the indicated $A \vee B$. Change them to A or B, as needed. The result may fail to be a proof. Discard some unnecessary $\vee$-*left* inferences and change names of variables to obtain proper proofs of $A, \Gamma \to \Delta$ and $B, \Gamma \to \Delta$. □

Theorem 1. *There exists a polynomial function p and a cut-free LK+permutation proof P_n such that the end-sequent of P_i is $ENDNODE_n$ and size$(P_n) \leqslant p(n)$.*

Proof. We prove $ENDNODE_n$ backwards and reduce it to a proof of $ENDNODE_{n-1}$. Then, we show that the length of the proof of $ENDNODE_n$ is bounded by $O(n^2)$ by induction on n.

First, we break down the formula $\bigvee_{1 \leqslant j \leqslant n} r_{j,n}$ in $ENDNODE_n$ by using $\vee$-*left* backwards. Then, we obtain sequents $\Gamma^k \to \Box$ where for each k $(1 \leqslant k \leqslant n)$ Γ^k is a cedent consisting of the following formulas.

1. $\neg r_{i,i}$ for all $1 \leqslant i \leqslant n$.
2. $\neg r_{i,j} \vee r_{j,i}$ for all $1 \leqslant i,j \leqslant n$.
3. $r_{k,n}$.

4. $\neg r_{j,n} \vee \neg r_{j',n}$ for all $1 \leqslant j < j' < n$.
5. $\bigvee_{1 \leqslant j<j' \leqslant n}(r_{i,j} \wedge r_{i,j'})$ for all $1 \leqslant i<n$.
6. $\neg r_{i,j} \vee \neg r_{i,j'} \vee \neg r_{i,j''}$ for all $1 \leqslant i<n$ and $1 \leqslant j<j'<j'' \leqslant n$.

Obviously Γ^n is reducible to an initial sequent $r_{n,n} \rightarrow r_{n,n}$. For k $(1 \leqslant k \leqslant n-2)$, Γ^k can be obtained from Γ^{n-1} by exchanging $r_{k,n}$ by $r_{n-1,n}$ and $r_{n,k}$ by $r_{n,n-1}$. Hence, we only need to consider Γ^{n-1}.

Second, we apply $\vee$-*left* backwards to Γ^{n-1} to decompose the formula $\bigvee_{1\leqslant j<j' \leqslant n}(r_{n-1,j} \wedge r_{n-1,j'})$. Then, we obtain two sequents which we have to prove: $\bigvee_{1\leqslant j<j'<n}(r_{n-1,j} \wedge r_{n-1,j'}), \Gamma^* \rightarrow \Box$ and $\bigvee_{1\leqslant j<n}(r_{n-1,j} \wedge r_{n-1,n}), \Gamma^* \rightarrow \Box$ where Γ^* is a cedent obtained from Γ^{n-1} by deleting the formula $\bigvee_{1\leqslant j<j' \leqslant n}(r_{n-1,j} \wedge r_{n-1,j'})$. We have a short proof for $r_{n-1,n}, r_{n-1,j}, r_{n-1,j'}, \neg r_{n-1,n} \vee \neg r_{n-1,j} \vee \neg r_{n-1,j'} \rightarrow \Box$, and so for $\bigvee_{1\leqslant j<j'<n}(r_{n-1,j} \wedge r_{n-1,j'})$, $\Gamma^* \rightarrow \Box$. Now we focus on the latter sequent, $\bigvee_{1\leqslant j<n}(r_{n-1,j} \wedge r_{n-1,n})$, $\Gamma^* \rightarrow \Box$. We, again, apply $\vee$-*left* backwards to the sequent and decompose the formula $\bigvee_{1\leqslant j<n}(r_{n-1,j} \wedge r_{n-1,n})$. Then, we obtain the sequent $\Delta^k \rightarrow \Box$ where Δ^k consists of the following formulas.

1. $\neg r_{i,i}$ for all $1 \leqslant i \leqslant n$.
2. $\neg r_{i,j} \vee r_{j,i}$ for all $1 \leqslant i, j \leqslant n$.
3. $r_{n-1,n}$.
4. $\neg r_{j,n} \vee \neg r_{j',n}$ for all $1 \leqslant j<j'<n$.
5. $\bigvee_{1\leqslant j<j' \leqslant n}(r_{i,j} \wedge r_{i,j'})$ for all $1 \leqslant i<n-1$.
6. $r_{n-1,k} \wedge r_{n-1,n}$.
7. $\neg r_{i,j} \vee \neg r_{i,j''} \vee \neg r_{i,j'}$ for all $1 \leqslant i<n$ and $1 \leqslant j<j'<j'' \leqslant n$.

Obviously, Δ^{n-1} is reducible to an initial sequent $r_{n-1,n-1} \rightarrow r_{n-1,n-1}$. For k $(1 \leqslant k \leqslant n-3)$, Δ^k is obtainable from Δ^{n-2} by exchanging $r_{k,n-1}$ by $r_{n-2,n-1}$ and $r_{n-1,k}$ by $r_{n-1,n-2}$. Hence, we only need to consider the sequent $\Delta^{n-2} \rightarrow \Box$.

Third, we apply, to $\Delta^{n-2} \rightarrow \Box$, a logical inference of which auxiliary literal is $\neg r_{n-1,n}$ then a structural inference backwards so that we can obtain the sequents $\neg r_{n-1,n}$, $r_{n-1,n} \rightarrow \Box$ and $\Delta^* \rightarrow \Box$ where Δ^* consists of the following formulas.

1. $\neg r_{i,i}$ for all $1 \leqslant i \leqslant n-1$.
2. $\neg r_{i,j} \vee r_{j,i}$ for all $1 \leqslant i, j \leqslant n-1$.
3. $\neg r_{j,n-1} \vee \neg r_{j',n-1}$ for all $1 \leqslant j<j'<n$.
4. $\bigvee_{1\leqslant j<j' \leqslant n-1}(r_{i,j} \wedge r_{i,j'})$ for all $1 \leqslant i<n-1$.
5. $r_{n-1,n-2}$.
6. $\neg r_{i,j} \vee \neg r_{i,j'} \vee \neg r_{i,j''}$ for all $1 \leqslant i<n-1$ and $1 \leqslant j<j'<j'' \leqslant n-1$.

By Lemma 1 and the induction hypothesis, $\Delta^* \rightarrow \Box$ has a cut-free LK+permutation proof of length less than $O(n^2)$. The length of the proof of *ENDNODE*$_n$ given above is obviously bounded by $O(n^2)$. The size of this proof is bounded by $O(n^6)$ since the size of every line is bounded by $O(n^4)$.

2.2. Bondy's theorem

Bondy's theorem states that in any $n \times n$ (0,1)-matrix containing n pairwise distinct rows, there exists a column such that, if the column is deleted, the resulting $(n-1) \times n$

matrix still contains n pairwise distinct rows. Propositional Bondy's theorem $BONDY_n$ is obtained by translating the $\{i,j\}$-entry of the given matrix by a propositional variable $p_{i,j}$.

$$BONDY_n \quad \left(\bigwedge_{1\leqslant k_0\leqslant n} \bigvee_{1\leqslant i<j\leqslant n} \bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq k_0}} p_{i,k}\equiv p_{j,k} \right) \to \left(\bigvee_{1\leqslant i<j\leqslant n} \bigwedge_{1\leqslant k\leqslant n} p_{i,k}\equiv p_{j,k} \right)$$

Theorem 2. *There exists a polynomial function p and a cut-free LK+permutation proof P_n such that the end-sequent of P_n is $BONDY_n$ and $size(P_n)\leqslant p(n)$.*

Proof. We prove $BONDY_n$ backwards and show that the length of the proof of $BONDY_n$ is bounded by $O(n^4)$ by induction on n.

We denote the formula $\bigvee_{1\leqslant i<j\leqslant n}\bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq k_0}}(p_{i,k}\equiv p_{j,k})$ by Γ_{k_0} and the succedent of $BONDY_n$ by Δ_n. Hence, $BONDY_n$ is written as follows:

$$\Gamma_1,\dots,\Gamma_n\to\Delta_n.$$

First, we apply $\vee$-*left* backwards to decompose the formula Γ_1 in $BONDY_n$. As a result, we obtain $(n(n-1)/2)$-many sequents $\Gamma_1^{g,h},\Gamma_2,\dots,\Gamma_n\to\Delta_n$ where $\Gamma_1^{g,h}$ $(1\leqslant g<h\leqslant n)$ is a formula defined by

$$\bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 1}} p_{g,k}\equiv p_{h,k}.$$

$\Gamma_1^{g,h}$ intuitively means that the gth and the hth columns coincide except for the first row. $\Gamma_1^{g,h},\Gamma_2,\dots,\Gamma_n\to\Delta_n$ is obtainable from $\Gamma_1^{1,2},\Gamma_2,\dots,\Gamma_n\to\Delta_n$ by using a permutation inference. Hence, we only need to consider $\Gamma_1^{1,2},\Gamma_2,\dots,\Gamma_n\to\Delta_n$.

Similarly, we decompose the formula Γ_2 in $\Gamma_1^{1,2},\Gamma_2,\dots,\Gamma_n\to\Delta_n$ by applying $\vee$-*left* backwards. Then, we obtain the sequents $\Gamma_1^{1,2},\Gamma_2^{g,h},\Gamma_3,\dots,\Gamma_n\to\Delta_n$ where $\Gamma_2^{g,h}$ is defined by

$$\bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 2}} p_{g,k}\equiv p_{h,k}.$$

For $(g,h)=(1,2)$, the given sequent means that "if the first and the second column coincide except for the first row, and at the same time they coincide except for the second row, then there exist two columns which coincide". Obviously, the first and second columns are those which coincide. Thus, we can reduce it by applying structural inference backwards to the sequent S_1 defined as follows:

$$\bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 1}} p_{1,k}\equiv p_{2,k},\quad \bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 2}} p_{g,k}\equiv p_{h,k}\to \bigwedge_{1\leqslant k\leqslant n} p_{1,k}\equiv p_{2,k}.$$

S_1 follows from the transitivity of equivalence, and has a proof of length $O(n)$. For $(g,h)\neq(1,2)$, it can be obtained by using a permutation from $\Gamma_1^{1,2},\Gamma_2^{2,3},\Gamma_3,\dots,\Gamma_n\to\Delta_n$.

Again, we decompose the formula Γ_3 by applying $\vee$-*left* backwards. We obtain three different type of sequents which require different treatments. The first type of sequents means that "if two columns coincide except for the ith row and at the same time they coincide except for the jth row, then they must, actually, coincide". Sequents falling in this type can be obtained from S_1 by using a permutation. The second type means that "Suppose that there are three columns which satisfy the following. The i_0th and the i_1th columns coincide except for the j_0th row, the i_1th and the i_2th columns coincide except for the j_1th row, and the i_2th and the i_0th columns coincide except for the j_2th row. Then two of them has to coincide". Define S_2 by the sequent as follows:

$$\bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 1}} p_{1,k}\equiv p_{2,k},\qquad \bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 2}} p_{2,k}\equiv p_{3,k},\qquad \bigwedge_{\substack{1\leqslant k\leqslant n\\ k\neq 3}} p_{3,k}\equiv p_{1,k}\rightarrow \Delta_n.$$

The sequents falling in the second type can be obtained from S_2 by a permutation. S_2 follows from the transitivity of equivalence, and has a proof of length O(n).

We keep going on until we obtain the sequent S_n of the following form:

$$\left(\bigwedge_{2\leqslant k\leqslant n}(p_{1,k}\equiv p_{2,k}),\ldots,\bigwedge_{\substack{k\neq n-1\\ 1\leqslant k\leqslant n}}(p_{n-1,k}\equiv p_{n,k}),\bigwedge_{1\leqslant k\leqslant n-1}(p_{n,k}\not\equiv p_{0,k})\right)\rightarrow \Delta_n.$$

Again, S_n follows from the transitivity of equivalence and has a proof of length O(n). The length of the whole proof is bounded by O(n^4).

3. Permutation vs. renaming

In [2], we showed that GCNF+permutation satisfies the feasible subformula property in the following sense. Let R be a GCNF+permutation refutation of size m. Then, there exists a GCNF+permutation refutation R^* such that the last lines of R^* and R are the same, the size of R^* is bounded by polynomial of m, and every formula appearing in R^* is a subformula of some formula in the last line. In this section, we show that GCNF+renaming does not satisfy this property; the pigeonhole principle gives a counter example.

Definition 6. A GCNF+renaming refutation P is *normal* if it satisfies the subformula property; every formula appearing in P is a subformula of some formula in the end-sequent of P.

Lemma 2. *If P is a GCNF+renaming refutation of $l, \bar{l}C_1, \ldots, \bar{l}C_n, \Gamma$ with all the occurrences of l and $\bar{l}$ indicated, then there exists a GCNF+renaming refutation P^* of $C_1, \ldots, C_n, \Gamma$ with size(P^*)<size(P), len(P^*)< len(P) and neither l nor $\bar{l}$ occurring in P^*.*

Proof. First, we replace every occurrence of l (resp. $\bar{l}$) which is not an ancestor of an occurrence of l (resp. $\bar{l}$) in the end-cedent by a new literal k (resp. $\bar{k}$). Then we obtain another GCNF+renaming refutation P' of $l, \bar{l}C_1, \ldots, \bar{l}C_n, \Gamma$ with $size(P') \leqslant size(P)$ and $len(P') \leqslant len(P)$. By deleting every occurrences of l from P' and by replacing every occurrences of $\bar{l}C_i$ by C_i in P', we obtain a GCNF+renaming refutation P^* of $C_1, \ldots, C_n, \Gamma$ with $size(P^*) < size(P)$ and $len(P^*) < len(P)$. □

From Lemma 2, we can conclude the following.

Lemma 3. *Suppose that P is a GCNF+renaming refutation, and I is a renaming inference in P*

$$\dfrac{\begin{matrix} \vdots\; Q \\ \Gamma \end{matrix}}{\Gamma(q/p)}\, I.$$

If a literal p or $\bar{p}$ appears as a clause in Γ, then P can be shortened to P' so that

$$\begin{matrix} \dfrac{\begin{matrix} \vdots\; Q' & \\ \tilde{\Gamma} & q, \bar{q} \end{matrix}}{\Gamma(q/p)} \\ \vdots \end{matrix}$$

where $\tilde{\Gamma}$ is obtained from Γ by deleting all the occurrences of p and $\bar{p}$, and neither p nor $\bar{p}$ appears in Q'.

Theorem 3. *There exists a constant c, $c > 1$ such that for sufficiently large n every normal GCNF+renaming refutation of PHP_n contains at least c^n lines.*

Proof. By the result in [16], it suffices to show that a shortest normal GCNF+renaming refutation for PHP_n is actually a GCNF refutation. Suppose that I is a renaming inference in P_n,

$$\dfrac{\Gamma}{\Gamma(p_{l',h'}/p_{l,h})} I$$

with $p_{l',j'} \neq p_{l,j}$. By the definition of normal refutation, Γ consists of subformulas of formulas of the form either $\bigvee_{0 \leqslant j \leqslant n-1} p_{i,j}$ or $\bar{p}_{i,j}\bar{p}_{m,j}$. Note that $p_{l,h}$ only occurs in a subformula A of $\bigvee_{0 \leqslant j \leqslant n-1} p_{l,j}$. By Lemma 3, we can assume that A involves other variables than $p_{l,h}$. Again by the definition of normal refutation, the predecessor of A must be also a subformula of $\bigvee_{0 \leqslant j \leqslant n-1} p_{l,j}$; $l' = l$ and $j' \neq j$. On the other hand, $\bar{p}_{l,j}$ only occurs in a subformula B of $\bar{p}_{l,j}\bar{p}_{l^*,j}$ for some $l^* \neq l$ $(0 \leqslant l^* \leqslant n-1)$. By Lemma 3, we can assume that B is $\bar{p}_{l,j}\bar{p}_{l^*,j}$. However, the predecessor of B is either $\bar{p}_{l,j}\bar{p}_{l,j'}\bar{p}_{l^*,j}$ or $\bar{p}_{l,j'}\bar{p}_{l^*,j}$ with $j' \neq j$; it is not a subformula of any formula in the end-sequent. This contradicts the normality of P_n. □

4. The relative efficiency; GCNF+renaming vs. Frege

An extension rule, $p \leftrightarrow A$, allows to abbreviate a long formula A by a new propositional variable p. It saves the space to express complicated formulas, and as a result, we obtain considerably small-size proofs. Buss [9] showed that renaming rule has the same effects on lengths of proofs as extension over Frege: Frege+renaming p-simulates extended Frege.[1]

In this section, we show that the p-simulation problem of GCNF+renaming by Frege is as difficult to solve as that of extended Frege by Frege.

Theorem 4. *LK p-simulates cut-free LK+renaming if and only if LK p-simulates LK+renaming.*

Proof. ($\Leftarrow$) The backward implication is obvious.

($\Rightarrow$) Let P be an LK+renaming proof of $\Sigma \to \Pi$. For every cut inference in P,

$$\frac{\Gamma \to \Delta, A \quad A, \Gamma \to \Delta}{\Gamma \to \Delta}$$

we replace it by

$$\frac{\dfrac{\Gamma \to \Delta, A}{\neg A, \Gamma \to \Delta} \quad A, \Gamma \to \Delta}{\neg A \vee A, \Gamma \to \Delta}\ .$$

Then, we obtain a cut-free LK+renaming proof P' of

$$\neg A_1 \vee A_1, \dots, \neg A_n \vee A_n, \quad \Sigma \to \Pi$$

where $A_1, \dots, A_n$ is the list of cut-formulas in P. Note that $size(P') = \mathrm{O}(size(P))$. By the hypothesis, there exists a polynomial-time algorithm to translate P' to an LK proof P^* of $\neg A_1 \vee A_1, \dots, \neg A_n \vee A_n,\ \Sigma \to \Pi$. At the same time, there are small size LK proofs of $\to \neg A_i \vee A_i$ for all $1 \leqslant i \leqslant n$. By removing $\neg A_i \vee A_i$ by cuts, we obtain an LK proof Q of $\Sigma \to \Pi$ where $size(Q) = \mathrm{O}(size(P)^2)$. □

A similar statement holds for GCNF+permutation.

Theorem 5. *LK p-simulates cut-free LK+permutation if and only if LK p-simulates LK+permutation.*

Proof. The proof is similar to that of Theorem 4. □

Corollary 1. *Frege p-simulates GCNF+renaming if and only if Frege p-simulates extended Frege. Frege p-simulates GCNF+permutation if and only if Frege p-simulates LK+permutation.*

[1] It is open whether resolution+renaming p-simulates resolution+extension.

5. Open problems and future researches

In recent researches, it has been revealed that there is a close connection between the hierarchy of computational complexity and that of propositional calculi. Among them the relations between P vs. extended Frege systems, and NC^1 vs. Frege systems are well studied [14, 12, 5, 13]. There exist natural complexity classes known to fall between P and NC^1, for example LOGSPACE and NC. However there is no propositional calculus which is known to correspond to them. We conjecture that LK+permutation can be a good candidate for it: there exists a complexity class C such that LK+permutation "corresponds to" C in the sense of S.A.Cook:

1. $\mathrm{P} \supset \mathrm{C} \supset \mathrm{NC}^1$ and $\mathrm{P} \neq \mathrm{C} \neq \mathrm{NC}^1$.
2. If F is a universal combinatorial principle which can be proved using concepts in C, then F corresponds to a family of tautologies F_n which have polynomial-size LK+permutation proofs.

The combinatorial principles we have proved so far in GCNF (or cut-free LK) +permutation are already known to have polynomial-size Frege proofs. It will be interesting if one can find a family of tautologies such that it has polynomial-size GCNF+permutation proofs but it is not known if it has polynomial-size Frege proofs.

Another interesting open problem is to find superpolynomial lower bounds for GCNF +permutation or, even stronger, to show that GCNF+permutation does not polynomially simulate Frege systems. We conjecture the following.

1. There exists a family of combinatorial tautologies F_n such that GCNF + permutation polynomially proves F_n, however, it does not polynomially prove substitution instances of F_n.
2. Bounded depth Frege+permutation do not p-simulate Frege systems.

References

[1] M. Ajtai, The complexity of the pigeonhole principle, 29th Annual Symposium on the Foundations of Computer Science, 1988, pp. 346–355.

[2] N.H. Arai, Tractability of cut-free Gentzen type propositional calculus with permutation inference, Theoret. Comput. Sci. 170 (1996) 129–144.

[3] N.H. Arai, A proper hierarchy of propositional sequent calculi, Theoret. Comput. Sci. 159 (1996) 343–354.

[4] T. Arai, Frege system $F \vdash^{p(n)} (\bigwedge_{i<j\leqslant n} \bigvee_{k\leqslant n} \neg(p_{ik} \equiv p_{jk})) \rightarrow (\bigvee_{l\leqslant n} \bigwedge_{i<j\leqslant n} \bigvee_{k\leqslant n, k\neq l} \neg(p_{ik} \equiv p_{jk}))$ for a polynomial $p(n)$, manuscript, 1993.

[5] T. Arai, A bounded arithmetic *AID* for Frege system, Tech. Report FI-CXT1998-003.

[6] P. Beam, S. Cook, J. Edmonds, R. Impagliazzo, T. Pitassi, The relative complexity of NP search problems, 27th ACM Symp. Theory of Computing, 1995, pp. 303–314.

[7] M.L. Bonet, S. Buss, T. Pitassi, Hard examples for Frege systems, in: Feasible Mathematics 2, Birkhäuser, Boston, 1995, pp. 30–56.

[8] S.R. Buss, Some remarks on lengths of propositional proofs, Arch. Math. Logic 34 (1995) 377–394.

[9] S.R. Buss, Polynomial size proofs of the pigeonhole principle, J. Symbolic Logic 52 (1987) 916–27.

[10] S.R. Buss G. Turán, Resolution proofs of generalized pigeonhole principle, Theoret. Comput. Sci. 62 (1988) 311–317.

[11] S.R. Buss, P. Clote, Cutting plane, connectivity, and threshold logic, Arch. Math. Logic 35 (1996) 33–62.

[12] S.R. Buss, Propositional consistency proofs, Ann. Pure Appl. Logic 52 (1991) 3–29.

[13] S.A. Cook, Relating the provable collapse of P to NC^1 and the power of logical theories, in: Proof Complexity and Feasible Arithmetics, American Mathematical Society, 1998, pp. 73–92.

[14] S.A. Cook, A. Urquhart, Functional interpretations of feasibly constructive arithmetic, Ann. Pure Appl. Logic 63 (1993) 1–103.

[15] G. Gentzen, Untersuchungen über das logische Schließen, Math. Z. 39 (1934) 176–210, 405–431.

[16] A. Haken, The intractability of resolution, Theoret. Comput. Sci. 39 (1985) 297–308.

[17] J. Krajíček, Lowerbounds to the size of constant-depth propositional proofs, J. Symbolic Logic 54 (1994) 73–86.

[18] G. Takeuti, Proof Theory, North-Holland, Amsterdam, 1987.