Mathematical Problems of Computer Science 56, 65–72, 2021.
doi:10.51408/1963-0080

UDC 510.6

On Proof Complexity of Some Type of Tautologies

Vahagn N. Altunyan and Garik V. Petrosyan

Yerevan State University
e-mail: altunyanv@gmail.com, garik.petrosyan.1@gmail.com

Abstract

In this paper, we investigate the proof complexities of a special type of tautologies, which are described as tautologies consisting of implications and literals. In particular, we prove that the proof of this kind of tautologies can be polynomially reduced to the proof of tautologies consisting of formulas that are described by sign-alternating trees.
Keywords: Frege systems, Tautology, Sign-alternating tree, Proof complexity.
Article info: Received 8 September 2021; accepted 18 November 2021.

1. Introduction

One of the most fundamental problems of the proof complexity theory is to find an efficient proof system for classical propositional calculus. There is a widespread understanding that polynomial-time computability is the correct mathematical model of feasible computation. According to the opinion, a truly effective system should have a polynomial-size $p(n)$ proof for every tautology of size n. In [1], Cook and Reckhow named such a system a super system. They showed that $NP = coNP$ iff there exists a super system. It is well known that many systems are not super. This question about the Frege system, the most natural calculi for propositional logic, is still open.

In many papers, some specific sets of tautologies are introduced, and it is shown that the question about polynomially bounded sizes for Frege-proofs of all tautologies is reduced to an analogous question for a set of specific tautologies. In particular, Lutz Strasburger introduced in [2] the notion of balanced formulas and showed that if there are polynomially bounded Frege proofs for the set of balanced tautologies, then the Frege systems are super. An analogous result for some other class of tautologies is proved in [3].

In this work, we introduce formulas that can be described by sign-alternating trees (**sat** formulas) and show that the proofs of tautologies that contain only $\supset$ and $\neg$ symbols, where $\neg$ is used only in literals, can be polynomially reduced to proofs of specific formulas constructed from **sat** formulas.

2. Main Notions and Notations

We will use the current concepts of a classical tautology, Frege proof systems for classical propositional logic, proof and proof complexity [1]. Let us recall some of them.

A Frege system $\mathcal{F}$ uses a denumerable set of propositional variables, a finite, complete set of propositional connectives; $\mathcal{F}$ has a finite set of inference rules defined by a figure of the form $\frac{A_1 A_2 ... A_n}{B}$ (the rules of inference with zero hypotheses are the axioms schemes); $\mathcal{F}$ must be sound and complete, i.e., for each rule of inference $\frac{A_1 A_2 ... A_n}{B}$ every truth-value assignment, satisfying $A_1 A_2 ... A_n$, also satisfies B, and $\mathcal{F}$ must prove every tautology.

The particular choice of a language for the presented propositional formulas is immaterial in this consideration. However, for some technical reasons, we assume that the language contains propositional variables, logical connectives $\neg, \wedge, \vee, \supset$ and parentheses $(,)$. Note that some parentheses can be omitted in generally accepted cases.

By $|\varphi|$ we denote the size of a formula φ, defined as the number of entries of all logical signs in it. It is obvious that the full size of a formula, which is understood to be the number of all symbols is bounded by some linear function in $|\varphi|$.

In the theory of proof complexity, the two main characteristics of the proof are: t-complexity (length), defined as the number of proof steps, l-complexity (size), defined as the sum of sizes for all formulas in the proof (formal definitions are, for example, in [4]).

Let ϕ be a proof system and φ be a tautology. We denote by $l^{\phi}_{\varphi}(t^{\phi}_{\varphi})$ the minimal possible value of l-complexity (t-complexity) for all ϕ-proofs of tautology φ.

Let M be some set of tautologies.

Definition 1: *We call the ϕ-proofs of tautologies from a set M l-polynomially (t-polynomially) bounded if there is a polynomial p such that $l^{\phi}_{\varphi} \leq p(|\varphi|)$ ($t^{\phi}_{\varphi} \leq p(|\varphi|)$) for all φ from M.*

Definition 2: *We call the ϕ-proofs of tautologies from a set M l-linearly (t-linearly) bounded if there is a linear function f such that $l^{\phi}_{\varphi} \leq f(|\varphi|)$ ($t^{\phi}_{\varphi} \leq f(|\varphi|)$) for all φ from M.*

Now we'll give the definition of **sat** formulas and prove some lemmas, which are necessary for proving the main result.

Definition 3: *We'll say that a formula is described by a sign-alternating tree (**sat** formula) if it satisfies the following rules.*

1. *it's a literal*
2. *has a form $r \wedge (T_1 \vee T_2)$, where r is a literal and T_1, T_2 are **sat** formulas*

Lemma 1: *For any formulas A, B, C, the following formulas have polynomially bounded proofs.*

1. $A \equiv (C \supset A) \wedge (\neg C \supset A)$
2. $A \supset (B \supset A)$
3. $(\neg A \supset (B \supset A)) \equiv (\neg A \supset \neg B)$
4. $\neg\neg A \equiv A$

5. $A \supset (B \supset C) \equiv A \wedge B \supset C$
6. $A \wedge \neg A \wedge B \supset C$
7. $A \wedge B \supset A$
8. $\neg(A \supset B) \equiv A \wedge \neg B$
9. $A \supset (B \wedge C) \equiv (A \supset B) \wedge (A \supset C)$
10. $A \supset (B \supset C) \equiv B \supset (A \supset C)$
11. $A \supset (A \supset B) \equiv A \supset B$
12. $(A \supset B) \wedge (C \supset B) \equiv (A \vee C) \supset B$

The proof is trivial as all the formulas are tautologies and have fixed length proofs, so the proof complexities may be assumed to be linearly bounded.

Lemma 2: *Tautologies of the form*

$$A \supset (B_1 \supset ...(B_{n-1} \supset (B_n \supset A))...)$$

have polynomially bounded proofs.

Proof. We can prove the tautology above by the following steps.

$$\begin{array}{ll}
A \vdash A & \\
A \vdash A \supset (B_n \supset A) & \text{2nd formula of Lemma 1} \\
A \vdash (B_n \supset A) & \textit{modus ponens} \\
A \vdash (B_n \supset A) \supset (B_{n-1} \supset (B_n \supset A)) & \\
A \vdash (B_{n-1} \supset (B_n \supset A)) & \textit{modus ponens} \\
\vdots & \\
A \vdash (B_1 \supset ...(B_{n-1} \supset (B_n \supset A))...) &
\end{array}$$

The number of proof steps is linearly bounded and the size of each formula in proof is also linearly bounded, so the proof is polynomially bounded. ■

Lemma 3: *Tautologies of the form*

$$d_1 \supset (d_2 \supset (... \supset d_k)...)$$

where $d_1, d_2, ..., d_k$ are literals, have polynomially bounded proofs.

Proof. After applying the operation of replacement by an equivalent formula $(k-2)$ times using the 5th formula of Lemma 1, we get:

$$d_1 \supset (d_2 \supset (... \supset d_k)...) \equiv d_1 \wedge d_2 \wedge ... \wedge d_{k-1} \supset d_k$$

In this case, if there exist $1 \leq i, j, \leq k-1$ such that $d_i = \neg d_j$ then replacing them with equivalent formulas using the 6th formula of Lemma 1, we'll get a polynomially bounded proof, or if there exist such $1 \leq i \leq k-1$ such that $d_i = d_k$ then replacing it with equivalent formulas using the 7th formula of Lemma 1, we'll get a polynomially bounded proof, otherwise the formula isn't a tautology. ■

3. Main Result

Definition 4: *Any propositional formula A is called sat-constructed if it is in the following form: $A = \neg(T_1 \wedge T_2 \wedge ... \wedge T_n)$, where $T_i(1 \leq i \leq n)$ are **sat** formulas.*

Theorem 1: *Let M be the set of all sat-constructed tautologies. If proofs of formulas from the set M are l-polynomially (t-polynomially) bounded, then proofs of all tautologies containing only $\supset$ and $\neg$ symbols, where $\neg$ is used only in literals, are l-polynomially (t-polynomially) bounded.*

Proof. We need to prove that any tautology A containing only $\supset$ and $\neg$ symbols, where $\neg$ is used only in literals, can be reduced to a *sat-constructed* tautology in polynomially bounded number of steps and length.

If the number of implications in the formula is bounded by, let's say, 3, then it can be reduced to $\neg(p_1 \wedge \neg p_1)$, and the reduction complexity will be constant which is also a polynomial. We now assume that formula A contains more than 3 implications. A can be expressed in the following form:

$$A = (S_1 \supset ...(S_{c-1} \supset (S_c \supset q_1))...) \tag{1}$$

where $S_i(1 \leq i \leq c)$ are sub-formulas and q_1 is a literal. We can replace A with an equivalent formula using the 1st formula of Lemma 1 and get $(q_1 \supset A) \wedge (\neg q_1 \supset A)$. The first half has a polynomially bounded proof by Lemma 2.

For the second half, we can apply the operation of replacement by an equivalent formula using the 3rd formula of Lemma 1 and get:

$$\neg q_1 \supset (S_1 \supset ...(S_{c-1} \supset (S_c \supset q_1))...) \equiv \neg q_1 \supset (S_1 \supset ...(S_{c-1} \supset \neg S_c)...)$$

So the proof of A has been polynomially reduced to the proof of $\neg q_1 \supset (S_1 \supset ...(S_{c-1} \supset \neg S_c)...)$. If S_c is a literal, then we can repeat the same process for the formula we got. After some repetitions, we'll end up either with a tautology $d_1 \supset (d_2 \supset (... \supset d_k)...)$, where $d_i(1 \leq i \leq k)$ are literals. This tautology has a polynomially bounded proof by Lemma 3. Or we'll end up with the following formula:

$$d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset \neg F)...))...) \tag{2}$$

where $d_i(1 \leq i \leq m)$ are literals and F is not a literal.

Suppose $F = F_1 \supset F_2$, applying the replacement by an equivalent formula using the 8th formula of Lemma 1, we'll get:

$$\begin{gathered} d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset \neg F)...))...) \equiv \\ d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset (F_1 \wedge \neg F_2))...))...) \end{gathered}$$

Applying the operation of replacement by an equivalent formula multiple times using the 9th formula of Lemma 1, we'll get:

$$\begin{gathered} d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset (F_1 \wedge \neg F_2))...))...) \equiv \\ d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset F_1)...))...) \wedge \\ d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ...(U_k \supset \neg F_2)...))...) \end{gathered}$$

So, after polynomially bounded number of steps, we reduced the proof of A to the proof of two formulas, the first one of which has the form (1) and the second one has the form (2). Repeating the same process for these formulas, we'll get l tautologies, and the proof of A will be reduced to the proof of these tautologies:

$$\begin{array}{c} d_1^1 \supset (d_2^1 \supset ... \supset (d_{m_1}^1 \supset (d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ... \supset \neg U_k))...)))...) \\ d_1^2 \supset (d_2^2 \supset ... \supset (d_{m_2}^2 \supset (d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ... \supset \neg U_k))...)))...) \\ \vdots \\ d_1^l \supset (d_2^l \supset ... \supset (d_{m_l}^l \supset (d_1 \supset (d_2 \supset ... \supset (d_m \supset (U_1 \supset ... \supset \neg U_k))...)))...) \end{array}$$

where $d_1^i, d_2^i, ..., d_{m_i}^i (1 \leq i \leq l)$ are literals. There are no repetitions among $d_1^i, d_2^i, ..., d_{m_i}^i, d_1, ..., d_m$, otherwise we can keep a single one of each repetition - replacing by equivalent formulas using the 10th and 11th formulas of Lemma 1. We may also assume that there are no variable repetitions, because if a variable and it's negation are present, then the formula can be polynomially proved. So we can assume that all the literals among $d_1^i, d_2^i, ..., d_{m_i}^i, d_1, ..., d_m$ are different and all the variables are also different and so the number of those literals doesn't exceed $|A|$. Also note that each time a new formula was generated an implication from F was removed, so $l < |F| < |A|$.

Applying the operation of replacement by an equivalent formula using the 10th formula of Lemma 1, we'll get the following form for our l formulas:

$$\begin{array}{c} C_1 = d_2 \supset (d_3 \supset ... \supset (d_m \supset (d_1 \supset (d_1^1 \supset ... \supset (d_{m_1}^1 \supset (U_1 \supset ... \supset \neg U_k))...)))...) \\ C_2 = d_2 \supset (d_3 \supset ... \supset (d_m \supset (d_1 \supset (d_1^2 \supset ... \supset (d_{m_2}^2 \supset (U_1 \supset ... \supset \neg U_k))...)))...) \\ \vdots \\ C_l = d_2 \supset (d_3 \supset ... \supset (d_m \supset (d_1 \supset (d_1^l \supset ... \supset (d_{m_l}^l \supset (U_1 \supset ... \supset \neg U_k))...)))...) \end{array}$$

At this point, we've reduced the proof of A to the proof of $C_1 \wedge C_2 \wedge ... \wedge C_l$ polynomially.

Applying the operation of replacement by an equivalent formula using the 9th formula of Lemma 1, we'll get:

$$C_1 \wedge C_2 \wedge ... \wedge C_l \equiv d_2 \supset (d_3 \supset ... \supset (d_m \supset (C_1' \wedge C_2' \wedge ... \wedge C_l'))...)$$

where

$$\begin{array}{c} C_1' = (d_1 \supset (d_1^1 \supset ... \supset (d_{m_1}^1 \supset (U_1 \supset ... \supset \neg U_k))...)) \\ C_2' = (d_1 \supset (d_1^2 \supset ... \supset (d_{m_2}^2 \supset (U_1 \supset ... \supset \neg U_k))...)) \\ \vdots \\ C_l' = (d_1 \supset (d_1^l \supset ... \supset (d_{m_l}^l \supset (U_1 \supset ... \supset \neg U_k))...)) \end{array}$$

Applying the operation of replacement by an equivalent formula using the 5th formula of Lemma 1, we'll get the following form for above formulas:

$$\begin{array}{c} C_1' = d_1 \wedge d_1^1 \wedge ... \wedge d_{m_1}^1 \supset (U_1 \supset ... \supset \neg U_k) \\ C_2' = d_1 \wedge d_1^2 \wedge ... \wedge d_{m_2}^2 \supset (U_1 \supset ... \supset \neg U_k) \\ \vdots \\ C_l' = d_1 \wedge d_1^l \wedge ... \wedge d_{m_l}^l \supset (U_1 \supset ... \supset \neg U_k) \end{array}$$

Applying the operation of replacement by an equivalent formula using the 12th formula of Lemma 1, we'll get:

$$C'_1 \wedge C'_2 \wedge ... \wedge C'_l \equiv (d_1 \wedge d_1^1 \wedge ... \wedge d_{m_1}^1 \vee d_1 \wedge d_1^2 \wedge ... \wedge d_{m_1}^2 \vee ... \vee d_1 \wedge d_1^l \wedge ... \wedge d_{m_1}^l) \supset (U_1 \supset ... \supset \neg U_k)$$

Let's prove that the **DNF** generated above - $(d_1 \wedge d_1^1 \wedge ... \wedge d_{m_1}^1 \vee d_1 \wedge d_1^2 \wedge ... \wedge d_{m_1}^2 \vee ... \vee d_1 \wedge d_1^l \wedge ... \wedge d_{m_1}^l)$ is an **sat** formula.

As we can see, each of the conjunctions includes d_1. Note that we can clearly split the **DNF** into two **subDNF**'s - one generated from F_1 and the other generated from F_2. For each of F_1 and F_2, we repeated the same process and so if the generated **DNF**s from F_1 and F_2 are both **sat** formulas, then the **DNF** generated above is also an **sat** formula. Note that we can make the assumption above, as after finite repetitions of splitting operation, we'll reach a literal, which is an **sat** formula.

At this point, we have polynomially reduced the proof of tautology A to the proof of the following tautology:

$$d_2 \supset (d_3 \supset ... \supset (d_m \supset (d_1 \wedge d_1^1 \wedge ... \wedge d_{m_1}^1 \vee d_1 \wedge d_1^2 \wedge ... \wedge d_{m_1}^2 \vee ... \vee d_1 \wedge d_1^l \wedge ... \wedge d_{m_1}^l) \supset (U_1 \supset ... \supset \neg U_k)...)$$

By deduction theorem the proof of the above formula is equivalent to the following proof:

$$d_2, ..., d_m, (d_1 \wedge d_1^1 \wedge ... \wedge d_{m_1}^1 \vee d_1 \wedge d_1^2 \wedge ... \wedge d_{m_1}^2 \vee ... \vee d_1 \wedge d_1^l \wedge ... \wedge d_{m_1}^l) \vdash (U_1 \supset ... \supset \neg U_k)...)$$

Note that all the hypotheses are **sat** formulas and that we can repeat all the previous steps on formula $(U_1 \supset ... \supset \neg U_k)$ even though we have some hypotheses. Repeating this process, the proof of A will be reduced to the proof of the following formula:

$$T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1,$$

where $T_i (1 \leq i \leq n)$ are **sat** formulas and c_1 is a literal. Note that new T_i-s are generated only when we consider the last sub-formula of A and then remove it for the next step, so this guarantees that $n \leq |A|$.

Applying the operation of replacement by an equivalent formula using the 1st formula of Lemma 1, we'll get:

$$T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1 \equiv (c_1 \supset (T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1)) \wedge (\neg c_1 \supset (T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1))$$

The tautology $(c_1 \supset (T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1))$ has a polynomially bounded proof.

We reduced the proof of A to the proof of $(\neg c_1 \supset (T_1 \wedge T_2 \wedge ... \wedge T_n \supset c_1))$. Applying the operation of replacement by an equivalent formula using the 3rd formula of Lemma 1, we'll get:

$$A \equiv \neg(T_1 \wedge T_2 \wedge ... \wedge T_n)$$

All the operations have polynomial complexity. The number of steps is also polynomially bounded, so the total complexity of reduction is polynomially bounded. ■

4. Conclusion

In this work, we introduced **sat** formulas and reduced the proofs of tautologies containing only $\supset$ and $\neg$ symbols, where $\neg$ is used only in literals, to the proofs of *sat-constructed* tautologies in polynomially bounded number of steps and length. Investigation of proof complexities of *sat-constructed* tautologies is in process.

References

[1] S. A. Cook and A. R. Reckhow, "The relative efficiency of propositional proof systems", *Journal of Symbolic logic*, vol. 44, pp. 36-50, 1979.

[2] L. Strasburger, "Extension without Cut", *Annals of Pure and Applied Logic*, vol. 163, no. 12, pp. 1995-2007, 2012.

[3] A. A. Chubaryan and G. V. Petrosyan, "Some notes on proof complexities in Frege systems", *Sciences of Europe*, vol 1. # 12 (12), Physics and Mathematics, pp. 31–34, 2017.

[4] J. Nordstrom, "Narrow proofs may be spacious: Separating space and width in resolution", *SIAM Journal on Computing*, vol. 39, no. 1, pp. 59-121, 2019.

Որոշ տիպի նույնաբանությունների արտածման բարդությունների վերաբերյալ

Վահագն Ն. Ալթունյան և Գարիկ Վ. Պետրոսյան

Երևանի պետական համալսարան
e-mail: altunyanv@gmail.com, garik.petrosyan.1@gmail.com

Ամփոփում

Այս հոդվածում ուսումնասիրվում են արտածման բարդությունները հատուկ տիպի նույնաբանությունների համար, որոնք նկարագրվում եմ որպես իմպլիկացիաներով և լիտերալներով կազմված նույնաբանություններ: Մասնավորապես ապացուցվել է, որ այդ տեսքի նույնաբանությունների արտածումները բազմանդամորեն հանգեցվում են նշանափոխ ծառերով ներկայացվող նույնաբանություն հանդիսացող բանաձևերի արտածումներին:

Բանալի բառեր՝ Ֆրեգեի համակարգեր, նույնաբանություններ, նշանափոխ ծառեր, արտածման բարդություն:

О сложности выводов некоторого типа тавтологий

Ваагн Н. Алтунян и Гарик В. Петросян

Ереванский государственный университет
e-mail: altunyanv@gmail.com, garik.petrosyan.1@gmail.com

Аннотация

В настоящей статье исследованы сложности выводов тавтологий специального вида, которые можно описать как тавтологии состоящие из импликаций и литералов. В частности, доказано, что выводы тавтологий такого вида можно полиномиально свести к выводам тавтологий, которые являются формулами, описываемыми знакопеременными деревьями.

Ключевые слова: системы Фреге, тавтологии, знакопеременные деревья, сложность вывода.