数理解析研究所講究録
第 772 巻 1991 年 72-74

Number of Proofs for Implicational Formulas

広川 佐千男 (Sachio Hirokawa)*

An algorithm is shown which determines the number $0, 1, \cdots, \infty$ of normal form proofs for implicational formulas. The number of proofs had not been studied well. Concerning to BCK-logic, it is proved by Komori and Hirokawa [3] that the number is identical to the number of BCK-minimal formulas of α. For general implicational formulas in intuitionistic logic, Ben-Yelles [1] showed an algorithm which enumerates all the normal form proofs for α when α has finitely many proofs. But we cannot use the algorithm to decide whether α has infinitely many proofs or not. We show a limit of proof search to decide whether α has infinitely many proofs.

Given an implicational formula α, we denote by $|\alpha|$ the number of occurrences of propositional variables and the implicational symbol '$\rightarrow$'. We consider proof figures in the intuitionistic logic in Natural Deduction System (NJ) [4]. We denote by $proof(\alpha)$ the set of normal form proofs of α. The cardinality of $proof(\alpha)$ is denoted by $\sharp proof(\alpha)$. The depth of a thread in a proof π is the number of minimum formula occurrences in the thread. The depth of π, denoted by $depth(\pi)$, is the maximal depth among all the threads in π. According to the formulae-as-types correspondence [2], a normal form proof π can be represented by a closed λ-term M in β-normal form . Then the $depth(\pi)$ is identical to the depth of Böhm-tree of M.

Theorem 1 *Given an implicational formula α,*

$$\sharp proof(\alpha) = \infty$$

*Department of Computer Science, College of General Education, Kyushu University, Fukuoka 810 JAPAN. (e-mail: `hirokawa@ec.kyushu-u.ac.jp`) Supported by a Grant-in-Aid for Encouragement of Young Scientists No.02740115 of the Ministry of Education.

iff there is a normal form proof $\pi \in proof(\alpha)$ such that

(1) $depth(\pi) \leq |\alpha|\ 2^{|\alpha|+1}$ and

(2) π contains a thread in which a formula ξ occurs twice as minimum formula occurrence.

$$\pi\left\{\begin{array}{c} \left.\begin{array}{c}\left.\begin{array}{c}\vdots\\ \xi\end{array}\right\}\pi_1\\ \vdots\\ \xi\end{array}\right\}\pi_2 \\ \vdots \\ \alpha\end{array}\right.$$

Outline of proof. If-part is trivial. In fact, we can replace π_1 by π_2. We can apply this rewriting successively. Thus we have$\sharp proof(\alpha) = \infty$. To prove only-if-part, assume that $\sharp proof(\alpha) = \infty$. Then there is a proof $\pi \in proof(\alpha)$ which contains a thread with depth $\geq 2\ d$, where $d = |\alpha|\ 2^{|\alpha|}$. Then the thread contains more than $2\ d$ minimum formula occurrences. Let ξ be an arbitrary minimum formula occurrence in the thread and $\{\delta_1, \cdots, \delta_n\}$ the assumption set for the sub-proof for ξ. By the sub-formula property, all of $\xi, \delta_1, \cdots, \delta_n$ are sub-formulas of α. So we have at most d such pairs $(\xi, \{\delta_1, \cdots, \delta_n\})$. Since the depth of the thread is longer than $2\ d$, it contains *three* occurrences of the same minimum formula occurrence ξ with the same assumption set $\{\delta_1, \cdots, \delta_n\}$. Let π_1, π_2, and π_3 be sub-proof for such occurrences of ξ which π_i appears above $\pi_{i+1} (i = 1, 2)$. Then we can replace π_2 by π_1 obtaining a smaller proof of α. We can apply this transformation until we obtain a proof of α with depth $\leq 2\ d$. ∎

Theorem 2 *There is an algorithm which determines $\sharp proof(\alpha)$ for implicational formula α.*

Proof. Consider the set of normal form proofs of α with depth $\leq |\alpha|\ 2^{|\alpha|+1}$. Note that the set is finite. If this set contains some π which satisfies (2) of Theorem 1, then $\sharp proof(\alpha) = \infty$. Otherwise $\sharp proof(\alpha)$ is finite. ∎

Theorem 1 without (1) is proved in Ben-Yelles [1]. Proof of Theorem 1 would remind some readers the similarity to the proof of *uvwxy-theorem* and infinity test for context free languages. Further work shall be necessary on this similarity.

References

[1] Ben-Yelles, C.-B., Type-assignment in the lambda-calculus, PhD thesis, University College, Swansea, 1979.

[2] Howard, W.A., The formulae-as-types notion of construction, in: Hindley and Seldin Ed., *To H.B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism* (Academic Press, 1980) 479-490.

[3] Komori,Y., Hirokawa, S., Number of proofs for BCK-formulas, preprint.

[4] Prawitz, D., Natural Deduction (Almqvist and Wiksell, Stockholm, 1965)