Degree-Two Formulas and Their Proofs

Hirokawa, Sachio
Department of Computer Science, College of General Education, Kyushu University

出版情報：RIFIS Technical Report. 49, 1991-11-07. Research Institute of Fundamental Information Science, Kyushu University

RIFIS-TR-CS-49

RIFIS Technical Report

Degree–Two Formulas and Their Proofs

Sachio Hirokawa

November 7, 1991

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

E-mail: hirokawa@ec.kyushu-u.ac.jp Phone: 092-771-4161

Degree Two Formulas and Their Proofs

Sachio Hirokawa *
Department of Computer Science,
College of General Education,
Kyushu University,
Fukuoka 810 JAPAN.
(e-mail: hirokawa@ec.kyushu-u.ac.jp)

Abstract

An implicational formula is said to be of degree two when it has the form

$$\alpha = (a_1^1 \to \cdots \to a_{n_1}^1 \to a^1) \to \cdots \to (a_1^m \to \cdots \to a_{n_m}^m \to a^m) \to a$$

with type variables $a_1^1, \cdots, a_{n_1}^1, a^1, \cdots, a_1^m, \cdots, a_{n_m}^1, a^m$ and a $(n_1, \cdots, n_m \geq 0)$. Given a degree two formula α, we construct, in polynormial time of the size of α, a context free grammar $G(\alpha)$ such that

$$proof(\alpha) = \{N \mid N \overset{\eta}{\twoheadleftarrow} \lambda x_1 \cdots x_m.M \ for\ some\ M \in L(G(\alpha))\}$$

where proof(α) is the set of normal form proof of α in NJ (natural deduction system of intuitionistic logic) and $L(G(\alpha))$ is the set of context free language generated by $G(\alpha)$. The set of provable formulas of degree two is characterized as the set of simple substitution instances of principal type-schemes of 'sorted' terms. A λ-term $\lambda x_1 \cdots x_m.M$ is sorted when M has no λ's and is constructed from function symbols $x_1, \cdots, x_m$ of arity $n_1, \cdots, n_m$ in the sense of many sorted algebra.

*Supported by a Grant-in-Aid for Encouragement of Young Scientists No.02740115 of the Ministry of Education.

1 Introduction

We study the structure of the set $proof(\alpha)$ of normal form proofs in NJ (natural deduction system of intuitionistic logic [8]) for implicational formulas. According to 'terms-as-proofs' correspondence [7], the set $proof(\alpha)$ can be identified to the set of closed λ-terms having α as their types. (We use the words formulas, implicational formulas and types in the same meaning throughout the paper.)

Ben-Yelles [3] showed that $\sharp proof(\alpha) = \infty$ iff there is a proof π of the form

$$\begin{array}{c}\vdots\\ \xi\\ \vdots\\ \xi\\ \vdots\\ \alpha\end{array}$$

in which some subformula ξ of α occurs twice in a thread of π. By evaluating the depth of thread, we showed an algorithm in [6] that determines whether $\sharp proof(\alpha)$ is infinite or not. Then we noticed a similarity in our proof to the proof of uvwxy-theorem of context free languages. In this paper, we clarify this similarity between proof figures and context free languages in precise form for degree two formulas.

An implicational formula is said to be of degree two when it has the form

$$\alpha = (a_1^1 \to \cdots \to a_{n_1}^1 \to a^1) \to \cdots \to (a_1^m \to \cdots \to a_{n_m}^m \to a^m) \to a$$

with type variables $a_1^1, \cdots, a_{n_1}^1, a^1, \cdots, a_1^m, \cdots, a_{n_m}^1, a^m$ and a ($n_1, \cdots, n_m \geq 0$). Given an implicational formula α of degree two, we construct a context free gramma $G(\alpha)$ such that

$$proof(\alpha) = \{N \mid N \overset{\eta}{\twoheadleftarrow} \lambda x_1 \cdots x_m.M \quad for \quad some \quad M \in L(G(\alpha))\}.$$

Here $L(G(\alpha))$ is the set of terms derived from a by production rules of $G(\alpha)$.

We characterize the set of provable formulas of degree two as the set of simple substitution instances of principal type-schemes of 'sorted' terms. A term $\lambda x_1 \cdots x_m.M$ in β-normal form is sorted when M has no λ's and is constructed from function symbols $x_1, \cdots, x_m$ of arity $n_1, \cdots, n_m$ in the sence of many sorted algebra. A principal type-scheme of a λ-term is a most general type-scheme of th term with repect to substituion. A simple substitution is a substitution θ of type variables such taht $c\theta$ is a type variable for any type variable c. When a λ-term $\lambda x_1 \cdots x_m.xM_1 \cdots M_n$ has no λ's in its body $xM_1 \cdots M_n$, then the body is considered as a 'word' constructed from free variables of $xM_1 \cdots M_n$ in language theory. Such terms are said to be *proper*. (See exercise 8.5.15, Barendregt [2] page 184.). Sorted λ-terms are special cases of proper terms.

Degree of a type α, denoted by $deg(\alpha)$, is the depth of the nest of arrows in the left direction. It is defined inductively by $deg(a) = 0$ for type variable a and $deg(\alpha_1 \to \cdots \to \alpha_m \to a) = 1 + max\{deg(\alpha_i) \mid 1 \leq i \leq m\}$. A transformation is known in [9] that decreases the degree of types and that preserves the provability of types in intuitionistic logic. By this transformation, we can obtain α^* from α in polynomial time of the size of α. (The author learned this transformation in [10].) Consider a type $\alpha = (((a \to b) \to c) \to d) \to (e \to f \to g) \to h$ whose degree is 4. Replace the occurrence of $a \to b$, which makes α of degree four, by a new type variable p. Then we obtain a type $\alpha' = ((p \to c) \to d) \to (e \to f \to g) \to h$ of degree three. We can put $\alpha^* = ((a \to b) \to p) \to \alpha'$. But, we cannot decrease the degree any more by this transformation.

It is known that the decision problem of provability of implicational formula is p-space complete [11]. Therefore the decision problem of degree three formula is p-space complete. We construct a context free grammar that generates $proof(\alpha)$ for degree two formula α. The construction is in polynomial time of the size of α. It is known that the problem of emptyness test for context free languages is in polynomial time. Therefore, our result implies that the decision problem for degree two formulas is in polynomal time. So we might say that there is a deep gap between degree two formulas and degree three formulas.

2 Context Free Grammar determined by Degree Two formula

We refer the basic notions in simple typed λ-calculus to [4]. The set of types is constructed from type variables by combining two types α and β with an arrow $\to$ obtainig a type $(\alpha \to \beta)$. We use lower case roman letters

$a, b, c, \cdots, a_1, a_2, \cdots, a_1^1, \cdots, a_j^i, \cdots$ for type varibles and $\alpha, \beta, \cdots, \alpha_1, \alpha_2, \cdots$ for types. We use an abbreviation $\alpha_1 \to \cdots \to \alpha_m \to a$ for $(\alpha_1 \to (\alpha_2 \to (\cdots \to (\alpha_m \to a) \cdots)))$.

Definition 1 *The degree of a type α, denoted by $deg(\alpha)$, is defined as follows.*

$$deg(a) = 0 \textit{ for type variable } a.$$

$$deg(\alpha_1 \to \cdots \to \alpha_m \to a) = 1 + max\{deg(\alpha_i) \mid i = 1, \cdots, m\}.$$

For example we have $deg((a \to b \to a) \to (b \to a \to b) \to a \to b \to a) = 2$ and $deg((p \to q) \to r) \to (p \to q) \to r) = 3$.

Degree two types are types α with $deg(\alpha) \leq 2$. Any degree two type α has the following form

$$\alpha = (a_1^1 \to \cdots \to a_{n_1}^1 \to a^1) \to \cdots \to (a_1^m \to \cdots \to a_{n_m}^m \to a^m) \to a$$

with type variables $a_1^1, \cdots, a_{n_1}^1, a^1, \cdots, a_1^m, \cdots, a_{n_m}^1, a^m$ and a $(n_1, \cdots, n_m \geq 0)$.

Definition 2 *Let $B = \{x_i : a_1^i \to \cdots \to a_{n_i}^i \to a^i \mid i = 1, \cdots, m\}$ be a base and $a_1^i, \cdots, a_{n_i}^i, a^i$ be type variables $(i = 1, \cdots, m)$. Then we define a context free grammaer $G(B)$ as follows.*

(1) The set of terminal symbols is $\{x_1, \cdots, x_m\}$.

(2) The set of non-terminal symbols is the union of type variables in $\alpha_i (i = 1, \cdots, m)$.

(3) The set of production rules is $\{a^i \Rightarrow x_i a_1^i \cdots a_{n_i}^i \mid i = 1, \cdots, m\}$.

When the initial non-terminal symbol a is specified, we denote it by $G(B, a)$. Given a degree two type $\alpha = \alpha_1 \to \cdots \to \alpha_m \to a$, we define a context free grammar $G(\alpha)$ by

$$G(\alpha) = G(\{x_1 : \alpha_1, \cdots, x_m : \alpha_m\}, a)$$

where $x_1, \cdots, x_m$ is a sequence of distinct term variables.

Remark 1 *Given a base* $B = \{x_i : a_1^i \to \cdots \to a_{n_i}^i \to a^i \mid i = 1, \cdots, m\}$*, the language* $L(G(B,a))$ *derived from non-terminal symbol* a *is identical with the set* $T(a)$ *of terms (of 'sort'* a *in a many-sorted algebra generated by* $x_1, \cdots, x_m$*) that is defined by simaltaneous recursion as follows:*

$$\textit{if}\ \ t_j \in T(a_j^i)\ \ (j = 1, \cdots, n_i),\ \ \textit{then}\ \ x_j t_1 \cdots t_{n_j} \in T(a^i)\ \ (i = 1, \cdots m).$$

(In particular, if $n_i = 0$ *then* $x_i \in T(a^i)$*.)*

Example 1 *Let* $\alpha = (a \to a) \to a \to a$*. Then the context free grammar* $G(\alpha)$ *contains the following two rules.*

$$a \Rightarrow xa$$

$$a \Rightarrow y$$

Thus the context free language for this grammar is

$$\{y, xy, x(xy), \cdots, x(x(\cdots(xy)\cdots)), \cdots\}.$$

Lemma 1 *Let* $B = \{x_1 : \alpha_1, \cdots, x_m : \alpha_m\}$ *and* $deg(\alpha_i) \leq 1$ $(i = 1, \cdots, m)$*. For any* λ*-term* M *in* β*-normal form and for any type varible* a *in* $var(\alpha_1) \cup \cdots \cup var(\alpha_m)$*, we have*

$$B' |\!- M : a \quad iff \quad M \in L(G(B,a)).$$

Here $B' = \{x_j : \alpha_j \in B \mid x_j \in FV(M)\}$*.*

Proof. (If-part) Note that any λ-term $M \in L(G(B,a))$ has $x_1, \cdots, x_m$ and has no λ's in it. We prove the lemma by induction on the structure of M.

Base step. $M = x_i$ for some i $(i = 1, \cdots, m)$. Then $x_i \in L(G(B,a))$. Therefore $a = \alpha_i$ and $G(B)$ contains a production rule $a \Rightarrow x_i$. Thus $x_i : a \in B$. Therefore $\{x_i : a\} |\!- x_i : a$.

Induction Step. $M = x_i M_1 \cdots M_q$. Since $M \in L(G(B,a))$, we have a derivation of M form a as follows

$$a \Rightarrow x_i a_1^i \cdots a_{n_i}^i \overset{*}{\Rightarrow} x_i M_1 \cdots M_{n_i},$$

where $a_j^i \overset{*}{\Rightarrow} M_j (j = 1, \cdots, n_i)$. Therefore $G(B)$ contains a production rule $a \Rightarrow x_i a_1^i \cdots a_{n_i}^i$. Thus we have $x_i : a_1^i \to \cdots \to a_{n_i}^i \to a \in B$. On the other hand, we have $M_j \in L(G(B), a^i_j)$ $(j = 1, \cdots, n_i)$. By induction hypothesis, we have $B_j|\!-M_j : a^i_j$ where $B_j = \{x_k : \alpha_k \in B \mid x_k \in FV(M_j)\}(j = 1, \cdots, n_i)$. Thereore we have $B_1 \cup \cdots \cup B_{n_i} \cup \{x_i : a^i_1 \to \cdots \to a^i_{n_i} \to a\}$ $|\!-x_i M_1 \cdots M_{n_i} : a$ by the following type assignment figure.

$$\frac{x : a^i_1 \to \cdots \to a^i_{n_i} \to a \quad M_1 : a^i_1 \quad \cdots \quad M_n : a^i_{n_i}}{x M_1 \cdots M_{n_i} : a}$$

(Only-if-part) By induction on the structure of M. Since a is a type variable, M is not an abstraction.

Case 1. $M = x_i$ is a variable. Then we have $B' = \{x_i : a\}$ and $\alpha_i = a$. Thus $G(B)$ contains a production rule $a \Rightarrow x_i$. Thus we have $x_i \in L(G(B), a)$.

Case 2. $M = x_p M_1 \cdots M_{n_p} (n_p \geq 1)$. Then $B'|\!-M : a$ has the following type assignment figure.

$$\frac{x_p : a^p_1 \to \cdots \to a^p_{n_p} \to a \quad M_1 : a^p_1 \quad \cdots \quad M_{n_p} : a^p_{n_p}}{x_p M_1 \cdots M_{n_p} : a}$$

Thus we have $B_i|\!-M : a^p_i (i = 1, \cdots, n_p)$. By induction hypothesis we have $M_i \in L(G(B), a^p_i)$ for $i = 1, \cdots, n_p$. On the other hand we have $x_p : a^p_1 \to \cdots \to a^p_{n_p} \to a \in B$. Therefore $G(B)$ contains the rule $a \Rightarrow x_p a^p_1 \cdots a^p_{n_p}$. Thus we have $M = x_p M_1 \cdots M_{n_p} \in L(G(B), a)$. ∎

Remark 2 *In the type assignment figure for $B'|\!-M : a$, note that any type variable c in it occurs as a predicate of some type assignment formula of the form $L : c$.*

Theorem 1 *Let $\alpha = \alpha_1 \to \cdots \to \alpha_m \to a$ be a type of degree two. Then we have $proof(\alpha) =$*

$$\{N \mid N \overset{\eta}{\twoheadleftarrow} \lambda x_1 \cdots x_m . M forsome M \in L(G(\{x_1 : \alpha_1, \cdots, x_m : \alpha_m\}, a))\}.$$

Proof. $(\subseteq)$ Since η-reduction preserves types, it suffieces to show $|\!-\lambda x_1 \cdots x_m . M : \alpha$ for all $M \in L(G(\{x_1 : \alpha_1, \cdots, x_m : \alpha_m\}, a))$. By Lemma 1, we have $\{x_{i_1} : \alpha_{i_1}, \cdots, x_{i_k} : \alpha_{i_k}\}|\!-M : a$, where $FV(M) = \{x_{i_1}, \cdots, x_{i_k}\} \subseteq \{x_1, \cdots, x_m\}$. By abstaction of $x_1 : \alpha_1, \cdots, x_m : \alpha_m$ in this order, we obtain $|\!-\lambda x_1 \cdots x_m . M : \alpha_1 \to \cdots \to \alpha_m \to a$. Note that some $x_j : \alpha_j$ might be discharged vacantly when $x_j \notin FV(M)$.

($\supseteq$) Let $N = \lambda x_1 \cdots x_p.xN_1 \cdots N_q \in proof(\alpha)$. Then a type assignment figure for $|\!-N : \alpha$ has the following form

$$\left.\begin{array}{c} \dfrac{\dfrac{x : a_1 \rightarrow \cdots \rightarrow a_q \rightarrow \xi \quad N_1 : a_1 \quad \cdots \quad N_q : a_q}{xN_1 \cdots N_q : \xi}}{\lambda x_1 \cdots x_p.xN_1 \cdots N_q : \alpha_1 \rightarrow \cdots \rightarrow \alpha_p \rightarrow \xi} \end{array}\right\} P_0 \quad ,$$

where $\xi = a_{q+1} \rightarrow \cdots \rightarrow a_{q+r} \rightarrow a$, $m = p + r$ and $\alpha_{p+1} = a_{q+1}, \cdots, \alpha_{p+r} = a_{q+r}$ are type variables. Now let $x_{p+1}, \cdots, x_{p+r}$ be new variables and $B = \{x_{i_i} : \alpha_{i_1}, \cdots, x_{i_n} : \alpha_{i_n}\}$ for $FV(N) = \{x_{i_1}, \cdots x_{i_n}\}$. Then we can construct from P_0, a type assignment figure for $B \cup \{x_{p+1} : a_{q+1}, \cdots, x_{p+r} : a_{q+r}\}$ $|\!-xN_1 \cdots N_q x_{p+1} \cdots x_{p+r}$ as follows.

$$\dfrac{\dfrac{x : a_1 \rightarrow \cdots \rightarrow a_q \rightarrow \xi \quad N_1 : a_1 \quad \cdots \quad N_q : a_q}{xN_1 \cdots N_q : \xi} \qquad x_{p+1} : a_{q+1} \cdots x_{p+r} : a_{q+r}}{xN_1 \cdots N_q x_{p+1} \cdots x_{p+r} : a}$$

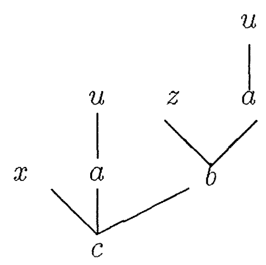
By Lemma 1, we have $xN_1 \cdots N_q x_{p+1} \cdots x_{p+r} \in L(G(\{ x_1 : \alpha_1, \cdots, x_m : \alpha_m \}, a))$. Since $x_{p+1}, \cdots, x_{p+r} \notin FV(xN_1 \cdots N_q)$, it follows that $\lambda x_1 \cdots x_p x_{p+1} \cdots x_{p+r}. xN_1 \cdots N_q x_{p+1} \cdots x_{p+r}$ is η-reducible to $\lambda x_1 \cdots x_p.xN_1 \cdots N_q$. ∎

Example 2 *Let $\alpha = (a \rightarrow b \rightarrow c) \rightarrow (c \rightarrow b \rightarrow a) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$. If we use $\{x, y, z, u\}$ as terminal symbols, then the grammar $G(\alpha)$ has the following four rules.*

$$\begin{array}{lcl} c & \Rightarrow & xab \\ a & \Rightarrow & ycb \\ b & \Rightarrow & za \\ a & \Rightarrow & u \end{array}$$

Since the initial non-terminal symbol is c, we have $xu(zu) \in L(G(\alpha))$ by the following derivation tree. This corresponds to the following type assignment figure for $\{x : a \rightarrow b \rightarrow c, z : a \rightarrow b, u : a\}|\!-xu(zu) : c$.

$$\dfrac{x : a \rightarrow b \rightarrow c \quad u : a \quad \dfrac{z : a \rightarrow b \quad u : a}{zu : b}}{xu(zu) : c}$$



3 Degree Two Formulas and Sorted λ-terms

In this section we show a characterization of degree two formulas in terms of 'sorted' λ-terms.

Let x be a free variable in a λ-term M. To distinguish the occurrences of the x's, we use superscripts $x^1, x^2, \cdots$. For example $M = x(xyz)$ has two occurrences of x. We write this by $x^1(x^2yz)$.

Definition 3 *Let x^i be an occurrence of free variable x in a λ-term M that has no λ's. The number of arguments of x^i, denoted by $arg(x^i)$, is the maximal integer m such that suberm $x^iM_1 \cdots M_m$ contains the occurrence of x^i in its head position.*

Definition 4 *A closed λ-term $M = \lambda x_1 \cdots x_m.x_iM_1 \cdots M_n$ in β-normal form is said to be* **sorted** *iff any two occurrences x^1 and x^2 of the same free variable $x \in FV(x_iM_1 \cdots M_n)$ have the same number of arguments. A λ-term N in β-normal form is sorted if $\lambda x_1 \cdots x_n.N$ is sorted where $\{x_1, \cdots, x_n\} = FV(M)$.*

Theorem 2 *Any sorted λ-terms has a principal type-scheme of degree two.*

Proof. It suffieces to show the folowing claim.

> **Claim:** Any sorted λ-term M having no λ's has a principal type assignment $\{x_1 : \alpha_1, \cdots, x_m : \alpha_m\} |\!- M : a$ such that $deg(\alpha_i) \leq 1 (i = 1, \cdots, m)$ and a is a type variable, where $\{x_1, \cdots, x_m\} = FV(M)$.

Since M is in β-normal form and has no λ's, it has the form $M = xM_1 \cdots M_q$. We prove the claim by induction on q.

Base Step. $q = 0$. Then $M = x$. Thus $\{x : a\}||\!-x : a$ for type variable a. Thus the claim is true.

Induciton Step. $M = xM_1 \cdots M_q (q \geq 1)$. By induction hypothesis, we have a principal type assignment for $B_i = \{x_1^i : \alpha_1^i, \cdots, x_{n_i}^i : \alpha_{n_i}^i\}||\!-M_i : a_i$ such that $deg(\alpha_1^i), \cdots, deg(\alpha_{n_i}^i) \leq 1$ and a_i is a type variable $(i = 1, \cdots, q)$. We construct a type assignment figure for $xM_1 \cdots M_q$ from these type assignment figures by combining them in the minor premisses of $(\rightarrow E)$ rule of the form

$$\frac{x : a_1 \rightarrow \cdots \rightarrow a_q \rightarrow a \quad M_1 : a_1 \cdots M_q : a_q}{xM_1 \cdots M_q : a}$$

with a new type variable a. But this figure cannot be a type assignment figure if

(a) some variable y occurs in $M_{i_1}, \cdots, M_{i_p}$ with the distinct subjects $y : \alpha^{i_1}, \cdots, y : \alpha^{i_p}$ or

(b) x occurs in $M_{j_1}, \cdots, M_{j_q}$ with the distinct subjects $x : \gamma^{j_1}, \cdots, x : \gamma^{j_q}$.

If (a) is the case, then $B_{i_1}, \cdots, B_{i_p}$ contains type assignment formulas $y : \alpha^{i_1}, \cdots, y : \alpha^{i_p}$ respectively. Since $\alpha_{i_1}, \cdots, \alpha_{i_p}$ are of degree 2, they have the following form.

$$\alpha_{i_1} = a^{i_1} \rightarrow \cdots \rightarrow a^{i_1} \rightarrow c^{i_1}$$

$$\cdots$$

$$\alpha_{i_p} = a^{i_p} \rightarrow \cdots \rightarrow a^{i_p} \rightarrow c^{i_p}$$

Since $B_{i_j}||\!-M_{i_j} : a_{i_j}$ is a principal type assignment, M_{i_j} contains an occcurrence of y with n_{i_j} arguments $(j = 1, \cdots, p)$. Since $xM_1 \cdots M_q$ is sorted, we have $n_{i_1} = \cdots = n_{i_p}$. Since $deg(\alpha^{i_1}), \cdots, deg(\alpha^{i_p}) \leq 1$, all $a_1^{i_1}, \cdots, a_{n_{i_1}}^{i_1}, c^{i_1}, \cdots, a_1^{i_p}, \cdots, a_{n_{i_p}}^{i_1}, c^{i_p}$ are type variables. Thus we can constuct a simple substitution θ_1 such that $\alpha^{i_1}\theta_1 = \cdots = \alpha^{i_p}\theta_1$. In this way, predicates of assumption in B_i's with the same subjects can be identified. If (b) is the case, we can construct a simple subsitution that identifies the subjects of $x : a_1 \rightarrow \cdots \rightarrow a_q \rightarrow a$ and the subjects of $x : \gamma^{j_1}, \cdots, x : \gamma^{j_q}$ in some B_j similary to the case (a). Thus wc can construct a type assignment to $xM_1 \cdots M_q$. Besides the subjects of the assumption are of degree one. Thus the lemma was proved.∎

Theorem 3 *For any type α, the following (1) and (2) are equivalent.*

(1) $NJ \mid\!- \alpha$ *and* $deg(\alpha) \leq 2$.

(2) α *is a simle substituion instance of a principal type-scheme of a sorted closed* λ*-term in* β*-normal form.*

Proof. (2) $\Rightarrow$ (1) follows from Theorem 2. (1) $\Rightarrow$ (2): Assume that $\mid\!- M : \alpha$ for closed λ-term M in β-normal form and $deg(\alpha) \leq 2$. Let $\alpha = \alpha_1 \rightarrow \cdots \rightarrow \alpha_m \rightarrow a$. By Theorem 1 we have $M \in L(G(\{x_1 : \alpha_1, \cdots, x_m : \alpha_m\}, a))$. By the definition of production rules of the grammar, we can see that M is sorted. By Remark 2, every type variable c occurs in the form $N : c$ in the type assignment figure of $\mid\!- M : \alpha$. By characterization of principal type assignment figures [5] α is a simple subsitution instance of a principal type-scheme of M. ∎

Further works

We showed that the set of normal form proofs for degree two formulas is a context free language. The grammar was constructed in polynomial time of the size of the formula. Akama [1] proved that the set of normal form proof for implicational formula, which does not have to be of degree 2, is recognizable by a tree automaton. His construction takes exponential time of the size of the term. Since the frontier set of regular set of trees is a context free, there will be a possibility to construct a context free grammar that generages all the λ-terms having the formula as their types.

References

[1] Akama, Y., Number of proofs for implicational formuas and tree automata, preprint.

[2] Barendregt, H.P., The Lambda-Calculus, its Syntax and Semantics, 2nd ed.(North-Holland,1981)

[3] Ben-Yelles,C.-B., Type-assignment in the lambda-calculus, Ph.D. thesis, (University College, Swansea,1979)

[4] Hindley,J.R., Seldin, J.P., Introduction to Combinators and Lambda-Calculus (Cambridge University Press, London, 1986)

[5] Hirokawa, S., Principal type assignment to lambda terms, International Journal of Foundation of Computer Sciences (to appear).

[6] Hirokawa, S., Number of proofs of implicational formulas, preprint.

[7] 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.

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

[9] Rose, G.F., Propositional calculus and realizability, *Trans. Am. Math. Soc.* **175** (1953) 1-19.

[10] Soloviev, S.V., Preservation of equivalence of derivations under reduction of depth of formulas, *Zapiski nauchnykh Seminarov Lenigradskogo Otdeleniya matematichskogo Instituta im. V.A. Steklova An SSSR* **88** (1979) 197-207.

[11] Statman, R., Intuitionistic propositional logic is polynomial-space complete, *Theoret. Comput. Sci.* **9** (1989) 97-105.