Comment. Math.
Univ. St. Pauli
XXIII—1, 1974

Gentzen-type formal system representing properties of functions*

by

Toshio NISHIMURA

(Received March 30, 1974)

§ 0. Introduction

First, we shall roughly explain the system given in this paper, which is mainly based on the 2-valued logic. By some little modifications, we shall be able to give the systems based on the 3-valued logic presented by S. C. Kleene [2] or J. McCarthy, which will be treated in the forthcoming paper.

Now let F or G be a function of type $\alpha \to \beta$ or $\beta \to \gamma$ respectively, then we shall denote the composition of F and G by

$$F \cdot G \quad (F \cdot G(x) = G(F(x))) .$$

And if F and G are compatible and of the same type, then the join of F and G shall be denoted by

$$F \vee G .$$

Let P be a formula, then P has a truth value. The function Φ which has always the value ϕ (the empty-set) represents the truth value 'false', and the identity function I the 'true'. (In 3-valued case we shall take the totally undefined function Ω as another value 'undefined'). Then a formula P can be considered as if it is a function, (*cf.* 1.5.4) and the composition

$$P \cdot F$$

has the value F if P is true and Φ if P is false. (In 3-valued case, if P is undefined then the value is totally undefined function Ω).

$$\text{if } P(x) \text{ then } y \text{ else } f(x)$$

has the value y if $P(x)$ is true and $f(x)$ if $P(x)$ is false. Thus, this can be represented by the composition

$$P(x) \cdot y \vee \neg P(x) \cdot f(x) ,$$

where $\neg P(x)$ represents the negation of $P(x)$. And the program

F ; G ;

* This is partly supported by CUDI foundation.

loop: if P then begin H; K; goto *loop* end

can be representd by

$$F \cdot G \cdot \left(\bigvee_{n=0}^{\infty} (P \cdot H \cdot K)^n \right) \neg P ,$$

where $A^0 = I$, $A^{n+1} = \underbrace{A \cdot A \cdots A}_{n+1 \text{ times}}$ and $\bigvee_{n=0}^{\infty} A^n = A^0 \vee A^1 \vee A^2 \vee \cdots$

Now let A and B be formulas. Then the composition $A \cdot B$ means 'A and B', because $I \cdot I = I$, $I \cdot \Phi = \Phi$, $\Phi \cdot I = \Phi$ and $\Phi \cdot \Phi = \Phi$ And $A \vee B$ means 'A or B' because $I \vee I = I$, $I \vee \Phi = I$, $\Phi \vee I = I$ And $\Phi \vee \Phi = \Phi$.

The program

$i := 1$; $s := 0$;

loop: if $\neg i > N$ then begin $s := s + a_i$; $i := i + 1$; goto *loop* end

shall give the result $s := a_1 + \cdots + a_N$. This is represented by the expression of the form

$$(i := 1)(s := 0) \bigvee_{n=0}^{\infty} (\neg (i > N)(s := s + a_i)(i := i + 1))^n (i > N) \subset s := a_1 + \cdots + a_N$$

which is called a sequent. This can be proved in our system. '$F \subset G$' means that G is an extension of F. In the 2-valued case, for formulas A and B, '$A \subset B$' means that A implies B, and so is the same as $A \rightarrow B$ in the Gentzen's original form.

We shall consider the following recursive definition of the function F (of type α)

$$F(x, y) = \text{if } p(x) \text{ then } y \text{ else } h(F(k(x), y)) .$$

It is well-known that F can be defined as the least fixed point $\bigvee_{n=0}^{\infty} f^n(\Phi)$ (denoted by ∂F), under the following definition of the function f of type $\alpha \rightarrow \alpha$

$$f^0(\Phi) = I(\Phi) = \Phi$$

$$f^{n+1}(\Phi) = \text{if } p(x) \text{ then } y \text{ else } h(f^n(\Phi)(k(x), y)$$

$$(\text{represented as } p(x) \cdot y \vee \neg p(x) \cdot h(f^n(\Phi)(k(x), h)))$$

We shall consider another function G of the same type as F, which is defined by the following:

$$G(x, y) = \text{if } p(x) \text{ then } y \text{ else } G(k(x), h(y)))$$

Then

$$G(x, y) = \partial g(x, y) = \Phi \vee \bigvee_{n=0}^{\infty} (p(x) \cdot y \vee \neg p(x) \cdot g^n(\Phi)(k(x), h(y)))$$

In order to prove that F is the same function as G, it is sufficient to show the following two sequents:

$$\partial f(x, y) \subset \partial g(x, y) \ and \ \partial g(x, y) \subset \partial f(x, y)$$

In our system, the notation '$\subset$' plays the similar role to the Gentzen's original notation '$\rightarrow$'. Rules of inference shall be given symmetrically for the left hand-side and the right hand-side of $\subset$.

In this paper, we shall give the formal system and its interpretation. From this we can easily obtain the plausibility of the system. And similarly to [4], we can obtain 'the completeness theorem' and 'the cut-elimination theorem'. Concerning to formal systems representing properties of functions, Platek, D. Scott [5] and M. Takahashi gave several axiomatized system, but Gentzen-type formulation including compositions of formulas and functions has not been given as yet. Gentzen-type formulation shall give the following profits to us:

(1) This will make us easy transformation to 3-valued cases from 2-valued case.

(2) This will suggest that the back-tracking is a powerful method in order to obtain the proofs for equivalence or correctness of programs. In fact, the theorem-prover [1] based on our system are now working as a powerful processor, which processed most problems presented in 'Inductive Methods for Proving Properties of Programs' by Z. Manna, S. Ness and J. Vjillemin [3].

§1. Formal system and its interpretation

In this section we shall give the formal system and its interpretation.

1.1. Constant symbols for types are o and ι. Types are defined by the following. (1) o or ι is a type. (2) If α and β are types, so is $\alpha \rightarrow \beta$. (3) Types are obtained only by applying (1) and (2) (Extreme clause). In what follows, we shall often omit the extreme clause. A predicate-type is defined by the following. (1). α is a type, then $\alpha \rightarrow \iota$ is a predicate-type. (2) If α is a type and β is predicate-type, then $\alpha \rightarrow \beta$ is a predicate-type. Types other than predicate-type are called 'object-type'.

1.2. Basic symbols other than those for types are following: functional (abbreviated by fl) constants of type α Φ_α etc.; free fl variables of type α F_α, G_α etc.; bound variables of type α X_α, Y_α etc.; overall-functional (abbreviated by ofl) constants K, L etc.; free ofl variables U, V etc.; index constants 0 etc.; free index variables a, b, c etc.; index functions $+, \cdot, '$ etc.; logical connectives $\cdot, \vee, \neg, \vee, \partial, \exists, \forall, =$; other symbols $\subset$ etc.. Especially, Φ_α is the constant of type α for every type α and T_α for every predicate-type α.

1.3. Now we shall give mathematical domains, in which our formal system will be interpreted.

Let D_{-1} be the domain of individuals. The totally undefined function on D_{-1} is denoted by Φ_0 or ω. We put

$$D_0=\{a^* \mid \text{We have } a\in D_{-1} \text{ such that } a^*(x)=a \text{ for ever } x\in D_{-1}\}\cup\{\omega\}$$

For $a, b\in D_0$, the relation $a\subset_0 b$ is defined by

$$a\subset_0 b \Leftrightarrow \mathrm{dom}\,(a)\subset\mathrm{dom}\,(b) \text{ and } a(x)=b(x) \text{ for every } x\in D_0\,.$$

Then we have $\omega\subset_0 a$ and $a\subset_0 a$ for every $a\in D_0$.

We put $D_\iota=\{\langle\ \rangle, \phi\}$, where $\langle\ \rangle$ denote the empty word and ϕ the empty set. And the relation $\subset_\iota$ is defined by

$$\phi\subset_\iota\phi\ ,\quad \phi\subset_\iota\langle\ \rangle \quad \text{or} \quad \langle\ \rangle\subset_\iota\langle\ \rangle$$

We shall often use the notation I or Φ instead of $\langle\ \rangle$ or ϕ respectively, because, as we shall see in 1.5, $\langle\ \rangle$ shall play the same role as the identity function I and ϕ as the Φ which has always the value ϕ.

Supposing that the domains D_α, D_β and the relations $\subset_\alpha$, $\subset_\beta$ are already defined, we put

$$D'_{\alpha\to\beta}=\{F_{\alpha\to\beta} \mid F_{\alpha\to\beta}\colon D_\alpha\longrightarrow D_\beta \quad\text{and}\quad \mathrm{dom}\,(F_{\alpha\to\beta})=D_\alpha\}$$

Now we shall define the relation $F_{\alpha\to\beta}\subset_{\alpha\to\beta}G_{\alpha\to\beta}$ by

$$F_{\alpha\to\beta}(h)\subset_\beta G_{\alpha\to\beta}(h) \quad \text{for every} \quad h\in D_\alpha$$

$F_{\alpha\to\beta}$ is said to be monotonic, if $F_{\alpha\to\beta}(h)\subset_\beta F_{\alpha\to\beta}(g)$ for every $h, g\in D_\alpha$ such that $h\subset_\alpha g$. Then $D_{\alpha\to\beta}$ is defined by

$$D_{\alpha\to\beta}=\{F \mid \text{monotonic } F\in D'_{\alpha\to\beta}\}$$

Next, let $\mathscr{T}$ be the set of all types and $\mathscr{D}$ be $\bigcup_{\alpha\in\mathscr{T}}D_\alpha$. We define the set $\mathscr{F}$ by

$\mathscr{F}=\{f \mid f$ is an fl defined on $\mathscr{D}$ and for every $\alpha\in\mathscr{T}$ the restriction of f to D_α (denoted by $f\restriction D_\alpha$) is an fl of type $\alpha\longrightarrow\alpha\}$

Let N be $\{0, 1, 2, \cdots\}$ and we consider the number theoretic functions corresponding to the index-functions.

Let $\Phi^*_{\alpha\to\beta}$ be the function such that $\Phi^*_{\alpha\to\beta}[h]=\Phi^*_\beta$ for every $h\in D_\alpha$, where $\Phi^*_0=\omega$ and $\Phi^*_\iota=\phi$ In what follows, we shall often omit type-subscript (i.e. F instead of F_α).

1.4. We shall assign an element of D_α (or $\mathscr{F}$) to every fl (or ofl)-constant or every fl (or ofl)-variable of type α and an element of N to every index-constant or-variable as follows. We denote this assignment by φ.

$$\varphi(\Phi_\alpha)=\Phi^*_\alpha$$

$\varphi(f)\in D_\alpha$ for every free fl-variable or fl-constant f of type α
$\varphi(f)\in\mathscr{F}$ for every free ofl-variable or ofl-constant f
$\varphi(0)=0$ for the index constant 0

$\varphi(a) \in N$ for the index variable a

$\varphi(f)=f^*$ for the index-function f, where f^* is the corresponding number theoretic function. (In what follows, $\varphi(E)$ is represented by E^*.)

1.5. Fl's, ofl's, indices and formulas and the extension of the assignment over these are defined by the following. In what follows, we denote one assigned to a formal expression E by E^*

1.5.1. An fl-constant of type α (ofl-constant) or a free fl-variable of type α (free ofl-variable) is an fl of type α (ofl). An ofl is an fl. An fl of type ι is called a formula.

1.5.2. If F or f is an fl of type $\alpha\to\beta$ or α respectively, so is $F(f)$ of type β. Especially, if $f_1, \cdots, f_n$ and F are of types $\alpha_1, \cdots, \alpha_n$ and $\alpha_1\to(\alpha_2\to \cdots (\alpha_n\to\beta)))$ respectively, $F(f_1)(f_2)\cdots(f_n)$ is of type β, abbreviated by $F(f_1, \cdots, f_n)$. If F is an ofl and f is an fl of type α, then $F(f)$ is an fl of type α. $(F(f))^*$ is $F^*(f^*)$.

1.5.3. Let F and G be fl's of type $\alpha\to\beta$ and $\beta\to\gamma$ respectively. The $F\cdot G$ (abbreviated by FG) is the fl of type $\alpha\to\gamma$. If F or G is an ofl, then FG is the fl of the same type of another. $(FG)^*$ is F^*G^* (the composition of F^* and G^*) i.e.

$$(FG)^*=F^*G^*=\{\langle fg\rangle \mid \exists h(\langle fh\rangle \in F^* \quad \text{and} \quad \langle hg\rangle \in G^*)\}$$

1.5.4. If P is a formula and F is an fl of type α (an ofl), then PF or FP is an fl of type α (an ofl). $(PF)^*=(FP)^*=$the direct product of P^* and F^*. Thus $(PF)^*=(FP)=F$ or ϕ according to $P^*=\langle\ \rangle$ or $P^*=\phi$. This shows that in our system $\langle\ \rangle$ or ϕ plays the same role as the ofl I^* or Φ^* respectively. We shall often use I^* or Φ^* instead of $\langle\ \rangle$or ϕ respectively. Especially, if F is also a formula, so is PF or FP. In this case, PF (FP) means 'P and F', because $I^*I^*=I^*$, $I^*\Phi^*=\Phi^*$, $\Phi^*I^*=\Phi^*$ and $\Phi^*\Phi^*=\Phi^*$

1.5.5. If P and Q are formulas, so are $\neg P$, $P\vee Q$, $P\supset Q$ and $P\equiv Q$. $\neg I^*=\Phi^*$, $\neg\Phi^*=I^*$, $(P\vee Q)^*=P^*\cup Q^*$, $(P\supset Q)^*=(\neg P\vee Q)^*$ $(P\equiv Q)^*=(P\supset Q)^*\vee(Q\supset P)^*$.

1.5.6. Let F be a formula, g a free fl-variable of type α and h a bound fl-variable of type α not contained in F. Then $\exists hF[h/g]$ and $\forall hF[h/g]$ is a formula, where $F[h/g]$ denote the result obtained by replacing h for g. $(\exists hF[h/g])^*$ has the value Φ^* if $(F[f/g])^*=\Phi^*$ for every $f\in D_\alpha$, otherwise I^*. $(\forall hF[h/g])^*$ has the value I if $(F[f/g])^*=I^*$ for every $f\in D_\alpha$, otherwise Φ^*.

1.5.7. An index-constant or a free index-variable is an index. If f is an n-ary index-function and $t_1, \cdots, t_n$ are indices, then $f(t_1, \cdots, t_n)$ is an index. $(f(t_1, \cdots, t_n))^*=f^*(t_1^*, \cdots, t_n^*)$

1.5.8. Let R be an fl of type $\alpha\to\alpha$ (or an ofl). Then R^t is of type $\alpha\to\alpha$ and ∂R of type α (or an ofl), where t is an index. $(R^t)^*=$

R^{*t^*} and $(\partial R)^* = \bigvee_{n=0}^{\infty} R^*(\Phi^*)$

1.5.9. Formulas of the forms Q_1PQ_2 and $R_1 \nearrow PR_2$ are said mutually disjoint. If formulas P and Q are mutually disjoint, then fl's APB and AQC are mutually disjoint. Two fl's are said compatible if they are mutually disjoint. And any two of $F^m(\Phi)$, $F^n(\Phi))$ and ∂F are compatible. Any two of formulas are always compatible. If f and g are compatible, so are AfB and AgB, or $F(f)$ and $F(g)$. Clearly, if two fl's A and B are mutually disjoint, then one of A^* and B^* is Φ^*. And

$$F^{*m}(\Phi^*) \subset F^{*n}(\Phi^*) \subset \partial F^* \quad \text{for} \quad m \leqq n$$
$$(\forall hP[h/g])^* \subset (P[f/g])^* \subset (\exists hP[f/g])^*$$

And if $f^* \subset g^*$, then $A^*f^*B^* \subset A^*g^*B^*$ and $F^*(f^*) \subset F^*(g^*)$. Thus, if A and B are compatible fl's, then $A^* \subset B^*$ or $B^* \subset A^*$.

1.5.10. If S and T are compatible fl's of type α (or ofl), then $S \vee T$ is an fl of type α (or ofl). Then $(S \vee T)^*$ = the join of S^* and T^*, so one of S^* and T^*.

Let $F_1, \cdots, F_m$ and $G_1, \cdots, G_n$ be sets of fl's of type α or ofl, in which any two of F's or Gs are compatible. Then the figure of the following form is called a sequent.

$$F_1, \cdots, F_m \subset G_1, \cdots, G_n ,$$

where it may happen that $m=0$ or $n=0$. This is interpreted by $F_1^* \cap \cdots \cap F_m^* \subset_\alpha G_1^* \cup \cdots \cup G_m^*$

§ 2. Proof-figure

In what follows, Greek capital letters, Γ, Π etc. shall represent finite set of fl's such as $F_1, \cdots, F_m$. A proof-figure is a tree constructed by sequents, in which every uppermost sequent is an axiom and by a rule of inference upper sequents and a lower sequent are connected.

2.1. We can give various assumptions as axioms, but we shall give here only logical axioms as the most basic ones. Logical axioms are sequents of the following form.

1. $\Phi \subset \Delta$, where Φ is the particular constant.
2. $\Gamma_1, F, \Gamma_2 \subset \Delta_1, F, \Delta_2$
3. $\Gamma_1, AP_1 \cdots P_mB, \Gamma_2 \subset \Delta_1, AP_{i_1} \cdots P_{i_k}B, \Delta_2$, where P_i is a formula.

It is clear that the logical axioms are true under any interpretations, because Φ^* is the least element, $F^* \subset F^*$ and $A^*P_1^* \cdots P_m^*B^* \subset A^*P_{i1}^* \cdots P_{ik}^*B^*$

2.2. Rules of inference

2.2.1. Rules of Replacement

(1) IF, FI, $\Phi \vee F$ or $F \vee \Phi$ can be replaced by F and conversely.

(2) ΦF or $F\Phi$ can be replaced by Φ and conversely.

(3) $\neg I$ or $\neg \Phi$ can be replaced by Φ or I respectively and conversely.

(4) $P\equiv Q$, $P\supset Q$, $\neg(P\cdot Q)$, $\neg(P\vee Q)$ or $\neg\neg P$ can be replaced by $(P\supset Q)\wedge(Q\supset P)$, $\neg P\vee Q$, $\neg P\vee \neg Q$, $\neg P\cdot \neg Q$ or P respectively and conversely.

(5) If Φ occurs in the left hand-side, then the left is replaced by Φ. Φ in the right is omitted.

In every rule of replacement, a true sequent is transformed to the true because A is replaced by B such that $A^*=B^*$.

2.2.2. Rules of inference with respect to logical connectives.

(1) $\vee$-left

$$\frac{\Gamma_1,\ AFB,\ \Gamma_2\subset\Delta \quad \Gamma_1,\ AGB,\ \Gamma_2\subset\Delta}{\Gamma_1,\ A\cdot\{F\vee G\}\cdot B,\ \Gamma_2\subset\Delta}$$

$\vee$-right

$$\frac{\Gamma\subset\Delta_1 AFB,\ AGB,\ \Delta_2}{\Gamma\subset\Delta_1,\ A\cdot\{F\vee G\}\cdot B,\ \Delta_2.}$$

These rules of inference shall give the true lower sequent from the true upper sequents under any interpretation, because F and G are compatible.

(2) ∂-left

$$\frac{\Gamma_1,\ A\cdot F^n(\Phi)\cdot B, \Gamma_2\subset\Delta \quad n=0,\ 1,\ 2,\ \cdots}{\Gamma_1,\ A\cdot\partial F\cdot B,\ \Gamma_2\subset\Delta}$$

∂-right

$$\frac{\Gamma\subset\Delta_1,\ A\cdot F^m(\Phi)\cdot B, \Delta_2,\ A\cdot\partial F\cdot B}{\Gamma\subset\Delta_1,\ A\cdot\partial F\cdot B_1\Delta_2}$$

It is easily shown from the compatibility of $F^n(\Phi)$ and ∂F that these rules of inference are reasonable.

(3) $\forall$-left

$$\frac{\Gamma_1,\ A\cdot P[g/h]\cdot B,\ \Gamma_2,\ A\cdot\forall hP\cdot B\subset\Delta}{\Gamma_1,\ A\cdot\forall hP\cdot B,\ \Gamma_2\subset\Delta}$$

where g is an arbitrary fl of the same type as h.

$\forall$-right

$$\frac{\Gamma\subset\Delta_1,\ A\cdot P[f/h]\cdot B,\ \Delta_2}{\Gamma\subset\Delta_1,\ A\cdot\forall hP\cdot B,\ \Delta_2}$$

where f is an arbitray free variable of the same type as h not contained in the lower sequent.

It is clear that $\forall$-left is the reasonable inference. We shall show $\forall$-right is so. If $(\forall hP)^*=I^*$, $(A\cdot P[f/h]\cdot B)^*\subset(A\cdot\forall hP\cdot B)^*$. Provided $(\forall hP)^*=\Phi^*$, we have $g\in D_\alpha$ such that $P^*[g/h]=\Phi^*$. Considering a new assignment φ' which assigns g to f and the original one to other than f, we have $\varphi'(P[f/h])=\Phi^*$ and $\varphi'(E)=\varphi(E)$ for E which does not contain f. Then, if the lower sequent is not true under φ, so is it under φ'. And so the upper sequent is not true under φ'. This contradicts to the assumption that the upper one is true under any interpretation.

(4) $\exists$-left

$$\frac{\Gamma_1,\ A\cdot P[f/h]\cdot B,\ \Gamma_2\subset\Delta}{\Gamma_1,\ A\cdot\exists hP\cdot B,\ \Gamma_2\subset\Delta}$$

where f satisfies the condition in $\forall$-right.

$\exists$-right

$$\frac{\Gamma\subset\Delta_1,\ A\cdot P[g/h]\cdot B, \Delta_2,\ A\cdot\exists hP\cdot B}{\Gamma\subset\Delta_1,\ A\cdot\exists hP\cdot B,\ \Delta_2}$$

where g satisfies the condition in $\forall$-left.

It will be shown by the quite similar way to in (3) that these are reasonable.

2.2.3. Practical rules of inference

We can add at will some practical reasonable rules, e.g.

$$\frac{\Gamma_1, \Gamma_2 \subset \Delta_1, C, \Delta_2 \qquad \Gamma_1, C, \Gamma_2 \subset \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \subset \Delta_1, \Delta_2}$$

and

$$\frac{A_1, \cdots, A_m \subset B_1, \cdots, B_n \qquad F \subset G}{A_1F, \cdots, A_mF \subset B_1G, \cdots, B_nG}$$

where $A_1, \cdots, A_m, B_1, \cdots, B_n$ are of type $\alpha \rightarrow \beta$ and F and G of type $\beta \rightarrow \gamma$.

From the facts given in the above, we shall see the following plausibility theorem.

THEOREM 1 (Plausibility). *Let $\Gamma \subset \Delta$ be a provable sequent. Then it is true under any interpretations.*

Similarly to [4], we can obtain the following important theorem.

THEOREM 2 (Completeness and Elimination of redundance). *Let a sequent $\Gamma \subset \Delta$ be true under any interpretation. Then it is provable by applying only rules in* 2.2.1 *and* 2.2.2.

REFERENCES

[1] IWAMARU, Y., NAGATA, M., NAKANISHI, M. and NISHIMURA, T.; Implementation of Gentzen-Type Formal System Representing Properties of Functions, This Journal.

[2] KLEENE, S. C.; *Introduction to Metamathematics*, D. Van Nostrand Company, Inc., Princeton, New Jersey, 1952.

[3] MANNA, Z., NESS, S. and VJILLIEMIN, J.; Inductive Methods for Proving Properties of Programs, *Proceedings of An ACM Conference On Proving Assertion About Programs*, New Mexico State University, New Mexico, January 6-7, 1972.

[4] NISHIMURA, T.; Gentzen-Style Formulation of Systems of Set-Calculus, This Journal.

[5] SCOTT, D.; A Type-Theoretical Alteratives to CUCH, ISWIM, OWHY, *Proceeding of An ACM Conference On Proving Assertions About Programs*, New Mexico State University, New Mexico, January 6-7, 1972.

The University of Tsukuba