Publ. RIMS, Kyoto Univ.
10 (1974), 271-291

A Formal System of Partial Recursive Functions

By

Hiroakira ONO*

We know that many parts of ordinary recursive function theory can be developed formally in a certain extension of the formal number theory (e.g. Peano arithmetic). But we encounter some difficulties when we want to deal with partial recursive functions, since in ordinary logical calculi only total functions and predicates can be treated. The most natural way to treat partial functions will be to take their graphs instead of functions themselves. More precisely, to represent an n-ary partial functions, an $(n+1)$-ary predicate P having the property that $P(x_1,\dots, x_n, y)$ holds for at most one y for any $x_1,\dots, x_n$ is taken. However, it will entail considerable complications to express properties of partial recursive functions in such forms.

In this paper, we shall attempt to formalize the theory of partial recursive functions, which is called *PRN*, on a logical calculus in which partial functions and predicates can be treated. For the logical calculus mentioned above, we shall take a system which is obtained from the one introduced by Ebbinghaus [1] by extending it to second order. We shall introduce some extensions of *PRN* and examine their logical powers. Our approach contrasts with the one by Scott [8]. In [8] a partial function from a set A to another set B is regarded as a total function from A to the set B with the element which represents the undefined value.

In § 1 and § 2, the second order logic of partial functions *SP* and its semantics are introduced. Axioms of the theory *PRN* are given in § 3. In § 4, some completeness results for some extensions of *PRN* are proved and applications of our theories to the mathematical theory of computation are suggested.

Communicated by S. Takasu, January 23, 1974.
* Department of Mathematics, Tsuda College, Tokyo.

§1. Second Order Logic of Partial Functions

In this section, we shall present the second order logic (with equality) of partial functions, which is called *SP*. *SP* is an extension of Ebbinghaus' *PPL* [1]. Except for the notion of terms, the first order part of *SP* is essentially same as *PPL*. The language of *SP* consists of the following;

1) individual constants, function constants and predicate constants (we assume that the language of *SP* contains the equality symbol $=$ as a predicate constant),

2) a list of countably infinite individual variables x, y, z etc.,

3) for each n, a list of countably infinite n-ary function variables $f^{(n)}$, $g^{(n)}$ etc.

Occasionally, we omit the superscript letter on a function variables. We define the terms by the inductive definition;

1) each individual constant or variable is a term,

2) if $t_1,\ldots, t_n$ are terms and f is an n-ary function constant or variable, then $f(t_1,\ldots, t_n)$ is a term,

3) if $t_1,\ldots, t_n, t, t'$ are terms and P is an n-ary predicate constant, then $(P(t_1,\ldots, t_n)\Rightarrow t; t')$ is a term.

$(\ \Rightarrow\ ;\ )$ designates the *if-then-else* operation of McCarthy [7]. $t, t', s, s', t_1, s_1,\ldots,$ etc. are used to denote terms. We use as the logical connectives, $\neg, \supset, \wedge, \vee, \forall$ and $\exists$. Formulas (of second order) are defined in the usual way. Thus, if A is a formula then $\forall fA$ and $\exists fA$ are formulas, where f is a function variable. A, B, C etc. are used to denote formulas.

We now introduce some abbreviations. $A\equiv B$ is an abbreviation of $(A\supset B)\wedge(B\supset A)$. ΔA is an abbreviation of $A\vee\neg A$. ΔA means that A is defined. $\sim A$ is a kind of the negation of A, which is an abbreviation of $\neg((A\supset A)\supset A)$. Δt is an abbreviation of $\exists x(x=t)$, where x is the first individual variable not appearing in a term t. Δt means that t is defined. Then the meanings of $\sim\Delta A$ and $\sim\Delta t$ are "A is undefined" and "t is undefined". $t\subseteq t'$ denotes the formula $\forall x(x=t\supset x=t')$, where x is the first individual variable appearing neither in t nor t'. $t\simeq t'$ is an abbreviation of $t\subseteq t'\wedge t'\subseteq t$. $\mathfrak{x}$ and $\mathfrak{t}$ denote some sequences of individual variables $x_1,\dots,x_m$ and of terms $t_1,\dots,t_m$, respectively. For instance, $\forall \mathfrak{x} P(\mathfrak{x})$ denotes $\forall x_1 \dots \forall x_m P(x_1,\dots,x_m)$ for some m.

$A_{x_1,\dots,x_n}[t_1,\dots,t_n]$ denotes the formula obtained from A by replacing simultaneously each free occurrence of x_i by t_i, if no free occurrence of x_i in A is in the scope of a quantifier $\forall y$ or $\exists y$ where y is an individual variable in t_i for each i. Suppose that f is an n-ary function variable and g is an n-ary function constant or variable. Then $A_f[g]$ denotes the formula obtained from A by replacing $f(t_1,\dots,t_n)$, at all of its occurrences in A at which f is free, by $g(t_1,\dots,t_n)$, if no occurrence of f in A is in the scope of a quantifier $\forall g$ or $\exists g$. Similarly, $t_{x_1,\dots,x_n}[s_1,\dots,s_n]$ denotes the term obtained from t by replacing each occurrence of x_i by s_i. For any n-ary function variable f, $t_f[\lambda x_1 \dots \lambda x_n s]$ denotes the term obtained from t by replacing each $f(t_1,\dots,t_n)$ by $s_{x_1,\dots,x_n}[t_1,\dots,t_n]$.

The logic SP is introduced in the same style as Gentzen's formal system [3]. So, we call an expression of the form $\Gamma \to A$ as a sequent, if Γ is a sequence of formulas and A is a formula. Both Γ and A may be null. Now, we give the axioms and the rules of inference of SP in the follwoing.

I) Axioms.

1) $\longrightarrow \Delta(x=y)$,

2) $A \longrightarrow A$,

3) $\longrightarrow \Delta(A \supset B)$.

II) Rules of inference

1) The structure rules (i.e., thinning, contraction and interchange), cut and logical rules for $\supset$, $\wedge$ and $\vee$ are same as those of Gentzen's LJ. (See [3].)

2) $\neg$-rules.

2.1) $\dfrac{\Gamma \longrightarrow A}{\neg A, \Gamma \longrightarrow}$ 2.2) $\dfrac{\Gamma \longrightarrow A}{\Gamma \longrightarrow \neg\neg A}$

2.3) $\dfrac{\Gamma \longrightarrow \neg\neg A}{\Gamma \longrightarrow A}$

3) Δ-rules for $\wedge$ and $\vee$.

3.1) $\dfrac{\Delta A, \Gamma \longrightarrow C}{\Delta(A \wedge B), \Gamma \longrightarrow C}$ 3.2) $\dfrac{\Delta B, \Gamma \longrightarrow C}{\Delta(A \wedge B), \Gamma \longrightarrow C}$

3.3) $\dfrac{\Gamma \longrightarrow \Delta A \qquad \Gamma \longrightarrow \Delta B}{\Gamma \longrightarrow \Delta(A \wedge B)}$

3.4) $\dfrac{\Gamma \longrightarrow \Delta A}{\Gamma \longrightarrow \Delta(A \vee B)}$ 3.5) $\dfrac{\Gamma \longrightarrow \Delta B}{\Gamma \longrightarrow \Delta(A \vee B)}$

3.6) $\dfrac{\Delta A, \Gamma \longrightarrow C \qquad \Delta B, \Gamma \longrightarrow C}{\Delta(A \vee B), \Gamma \longrightarrow C}$

4) Logical rules for the first order logic.

4.1) $\dfrac{\Gamma \longrightarrow A_x[y]}{\Gamma \longrightarrow \forall x A}$

where y is a variable not free in the conclusion.

4.2) $\dfrac{A_x[a], \Gamma \longrightarrow B}{\forall x A, \Gamma \longrightarrow B}$

where a is an individual constant or variable.

4.3) $\dfrac{\Gamma \longrightarrow A_x[a]}{\Gamma \longrightarrow \exists x A}$

where a is an individual constant or variable.

4.4) $\dfrac{A_x[y], \Gamma \longrightarrow B}{\exists x A, \Gamma \longrightarrow B}$

where y is a variable not free in the conclusion.

5) Rule for equality.

$$\dfrac{\Gamma \longrightarrow A}{\Gamma, x=t \longrightarrow A_x[t]}$$

6) Other Δ-rules.

6.1) $\dfrac{\Gamma \longrightarrow \Delta \forall x A}{\Gamma \longrightarrow \forall x \Delta A}$ 6.2) $\dfrac{\Gamma \longrightarrow \forall x \Delta A}{\Gamma \longrightarrow \Delta \forall x A}$

6.3) $\dfrac{\Gamma \longrightarrow \Delta \exists x A}{\Gamma \longrightarrow \exists x \Delta A}$ 6.4) $\dfrac{\Gamma \longrightarrow \exists x \Delta A}{\Gamma \longrightarrow \Delta \exists x A}$

6.5) $\dfrac{\Gamma \longrightarrow \Delta P(t_1, \dots, t_n)}{\Gamma \longrightarrow \Delta t_j}$ $(j=1, \dots, n)$

where P is an n-ary predicate constant.

6.6) $\dfrac{\Gamma \longrightarrow \Delta f(t_1,\dots,t_n)}{\Gamma \longrightarrow \Delta t_j}$ $(j=1,\dots,n)$

where f is an n-ary function constant or a variable.

7) Rules for terms.

7.1) $\dfrac{\Gamma \longrightarrow A}{\Gamma \longrightarrow (A \Rightarrow t; t') \simeq t}$

7.2) $\dfrac{\Gamma \longrightarrow \neg A}{\Gamma \longrightarrow (A \Rightarrow t; t') \simeq t'}$

7.3) $\dfrac{\Gamma \longrightarrow \Delta(A \Rightarrow t; t')}{\Gamma \longrightarrow \Delta A}$

where A is a formula of the form $P(t_1,\dots,t_n)$.

8) Logical rules for the second order logic.

8.1) $\dfrac{\Gamma \longrightarrow A_f[g]}{\Gamma \longrightarrow \forall f A}$

where g is a function variable not free in the conclusion.

8.2) $\dfrac{A_f[g], \Gamma \longrightarrow B}{\forall f A, \Gamma \longrightarrow B}$

where g is a function constant or variable.

8.3) $\dfrac{\Gamma \longrightarrow A_f[g]}{\Gamma \longrightarrow \exists f A}$

where g is a function constant or variable.

8.4) $\dfrac{A_f[g], \Gamma \longrightarrow B}{\exists f A, \Gamma \longrightarrow B}$

where g is a function variable not free in the conclusion.

8.5) $\dfrac{\Gamma \longrightarrow \Delta \forall f A}{\Gamma \longrightarrow \forall f \Delta A}$

8.6) $\dfrac{\Gamma \longrightarrow \forall f \Delta A}{\Gamma \longrightarrow \Delta \forall f A}$

8.7) $\dfrac{\Gamma \longrightarrow \Delta \exists f A}{\Gamma \longrightarrow \exists f \Delta A}$

8.8) $\dfrac{\Gamma \longrightarrow \exists f \Delta A}{\Gamma \longrightarrow \Delta \exists f A}$

The proofs and the provability in SP are defined in the same way as Gentzen's. Remark that $\dfrac{A, \Gamma \longrightarrow}{\Gamma \longrightarrow \neg A}$ does not holds in SP. We can show only that $\dfrac{A, \Gamma \longrightarrow}{\Delta A, \Gamma \longrightarrow \neg A}$ holds. On the other hand, 1) $\longrightarrow A \vee \sim A$, 2) $\dfrac{\Gamma \longrightarrow A}{\sim A, \Gamma \longrightarrow}$ and 3) $\dfrac{A, \Gamma \longrightarrow}{\Gamma \longrightarrow \sim A}$ holds in SP.

Lemma 1.1. *Following sequents are provable in SP.*

1) $\Delta t \longrightarrow t=t$,

2) $t_1=t_2 \longrightarrow t_2=t_1$,

3) $t_1=t_2,\quad t_2=t_3 \longrightarrow t_1=t_3$,

4) $\Delta t,\ t\simeq t' \longrightarrow t=t'$,

5) $\sim\Delta t,\ t\simeq t' \longrightarrow \sim\Delta t'$.

§2. Semantics of *SP*

We shall give a description of the semantics of *SP*, following Ebbinghaus [1].

A structure $\mathfrak{A}$ for a language L of *SP* consists of the following things:

1) A non-empty set A, which is called the domain of $\mathfrak{A}$. The elements of A are called the individuals of $\mathfrak{A}$.

2) For each $n\geqq 1$, a non-empty set $A^{(n)}$ of partial functions from A^n to A. That is, $A^{(n)}$ is a set of subsets of A^{n+1} such that for any $\alpha\in A^{(n)}$, if both $<a_1,\dots, a_n, a>$ and $<a_1,\dots, a_n, b>$ are in α then $a=b$ holds. The elements of $A^{(n)}$ are called the n-ary partial functions of $\mathfrak{A}$.

3) For each individual constant c of L, an element $c^{\mathfrak{A}}$ in A.

4) For each n-ary function constant f of L, an element $f^{\mathfrak{A}}$ in $A^{(n)}$.

5) For each propositional constant p of L, an element $p^{\mathfrak{A}}$ in the set $\{T, F, U\}$, where T, F and U mean '*true*', '*false*' and '*undefined*', respectively.

6) For each n-ary predicate constant P of L other than the equality $=$, two subsets $P^{\mathfrak{A}}$ and $P^{\mathfrak{A}}_D$ of A^n such that $P^{\mathfrak{A}}\subset P^{\mathfrak{A}}_D$. $P^{\mathfrak{A}}_D$ means the domain of the partial predicate $P^{\mathfrak{A}}$.

If $A^{(n)}$ consists of the set of all partial functions from A^n to A for each n, we say the structure $\mathfrak{A}$ is *total*.

To define the validity of a formula A in $\mathfrak{A}$, it is convenient to introduce the names for the individuals and the partial functions of $\mathfrak{A}$. So, for each individual a of $\mathfrak{A}$ and for each n-ary partial function α

of $\mathfrak{A}$, we choose new constants $\underline{a}$ and $\underline{\alpha}$, respectively. The language thus obtained is designated by $L(\mathfrak{A})$.

In the following, we assume that terms and formulas are of $L(\mathfrak{A})$ and contain neither free function variables nor free individual variables. Now, we shall define recursively the value of $t^{\mathfrak{A}}$ for a term t and $\mathfrak{A}(A)$ for any atomic formula A. The value of t is either undefined or an individual of $\mathfrak{A}$. In the latter case, we say that $t^{\mathfrak{A}}$ is defined. $\mathfrak{A}(A)$ takes one of the values $\{T, F, U\}$.

1) If t is an individual constant c of L then $t^{\mathfrak{A}}=c^{\mathfrak{A}}$.

2) $\underline{a}^{\mathfrak{A}}=a$ for any $a\in A$.

3) $f(t_1,\dots, t_n)^{\mathfrak{A}}=a$ if and only if all $t_i^{\mathfrak{A}}$'s are defined and $<t_1^{\mathfrak{A}},\dots, t_n^{\mathfrak{A}}, a> \in f^{\mathfrak{A}}$, where $f^{\mathfrak{A}}$ is α if f is $\underline{\alpha}$.

4) For any propositional constant p, $\mathfrak{A}(p)=p^{\mathfrak{A}}$.

5) Suppose that P is a predicate constant other than $=$. Then, $\mathfrak{A}(P(t_1,\dots, t_n))=T$ if all $t_i^{\mathfrak{A}}$'s are defined and $<t_1^{\mathfrak{A}},\dots, t_n^{\mathfrak{A}}> \in P^{\mathfrak{A}}$, $\mathfrak{A}(P(t_1,\dots, t_n))=F$ if all $t_i^{\mathfrak{A}}$'s are defined but $<t_1^{\mathfrak{A}},\dots, t_n^{\mathfrak{A}}> \in P_D^{\mathfrak{A}}-P^{\mathfrak{A}}$, and $\mathfrak{A}(P(t_1,\dots, t_n))=U$ otherwise.

6) $\mathfrak{A}(t=t')=T$ if both $t^{\mathfrak{A}}$ and $t'^{\mathfrak{A}}$ are defined and $t^{\mathfrak{A}}=t'^{\mathfrak{A}}$, $\mathfrak{A}(t=t')=F$ if both $t^{\mathfrak{A}}$ and $t'^{\mathfrak{A}}$ are defined but $t^{\mathfrak{A}}\neq t'^{\mathfrak{A}}$, and $\mathfrak{A}(t=t')=U$ otherwise.

7) $(P(t_1,\dots, t_n)\Rightarrow t;\, t')^{\mathfrak{A}}=a$ if and only if either $\mathfrak{A}(P(t_1,\dots, t_n))=T$ and $t^{\mathfrak{A}}=a$, or $\mathfrak{A}(P(t_1,\dots, t_n))=F$ and $t'^{\mathfrak{A}}=a$.

Next, we shall define $\mathfrak{A}(A)$ for any formula inductively.

1) $$\mathfrak{A}(\neg A)=\begin{cases} T & \text{if } \mathfrak{A}(A)=F, \\ U & \text{if } \mathfrak{A}(A)=U, \\ F & \text{if } \mathfrak{A}(A)=T. \end{cases}$$

2) $$\mathfrak{A}(A\supset B)=\begin{cases} T & \text{if } \mathfrak{A}(A)\neq T \text{ or } \mathfrak{A}(B)=T, \\ F & \text{otherwise.} \end{cases}$$

3) $$\mathfrak{A}(A\wedge B)=\begin{cases} T & \text{if } \mathfrak{A}(A)=\mathfrak{A}(B)=T, \\ U & \text{if } \mathfrak{A}(A)=U \text{ or } \mathfrak{A}(B)=U, \\ F & \text{otherwise.} \end{cases}$$

4) $$\mathfrak{A}(A \vee B) = \begin{cases} T & \text{if } \mathfrak{A}(A) = T \text{ or } \mathfrak{A}(B) = T, \\ U & \text{if } \mathfrak{A}(A) = \mathfrak{A}(B) = U, \\ F & \text{otherwise}. \end{cases}$$

5) $$\mathfrak{A}(\forall x A) = \begin{cases} T & \text{if } \mathfrak{A}(A_x[\underline{a}]) = T \text{ for any individual } a, \\ U & \text{if } \mathfrak{A}(A_x[a]) = U \text{ for some individual } a, \\ F & \text{otherwise}. \end{cases}$$

6) $$\mathfrak{A}(\exists x A) = \begin{cases} T & \text{if } \mathfrak{A}(A_x[\underline{a}]) = T \text{ for some individual } a, \\ U & \text{if } \mathfrak{A}(A_x[\underline{a}]) = U \text{ for any individual } a, \\ F & \text{otherwise}. \end{cases}$$

7) $$\mathfrak{A}(\forall f^{(n)} A) = \begin{cases} T & \text{if } \mathfrak{A}(A_{f^{(n)}}[\underline{\alpha}]) = T \text{ for any } n\text{-ary partial function } \alpha \text{ of } \mathfrak{A}, \\ U & \text{if } \mathfrak{A}(A_{f^{(n)}}[\underline{\alpha}]) = U \text{ for some } n\text{-ary partial function } \alpha \text{ of } \mathfrak{A}, \\ F & \text{otherwise}. \end{cases}$$

8) $$\mathfrak{A}(\exists f^{(n)} A) = \begin{cases} T & \text{if } \mathfrak{A}(A_{f^{(n)}}[\underline{\alpha}]) = T \text{ for some } n\text{-ary partial function } \alpha \text{ of } \mathfrak{A}, \\ U & \text{if } \mathfrak{A}(A_{f^{(n)}}[\underline{\alpha}]) = U \text{ for any } n\text{-ary partial function } \alpha \text{ of } \mathfrak{A}, \\ F & \text{otherwise}. \end{cases}$$

We can prove the following facts [1]:

1) $\mathfrak{A}(\Delta t) = T$ if and only if $t^{\mathfrak{A}}$ is defined.
2) $\mathfrak{A}(\Delta A) = T$ if and only if $\mathfrak{A}(A) \neq U$.
3) $\mathfrak{A}(\sim A) = T$ if and only if $\mathfrak{A}(A) \neq T$.

Suppose that all the free variables occurring in a formula A of L are $f_1, \ldots, f_m$ and $x_1, \ldots, x_n$, respectively. Then a closure A' of A

is a formula of the form $\forall f_1 \dots \forall f_m \forall x_1 \dots \forall x_n A$. Now, define the value of $\mathfrak{A}(A)$ by $\mathfrak{A}(A)=\mathfrak{A}(A')$. We say that a formula A is valid in a structure $\mathfrak{A}$ if $\mathfrak{A}(A)=T$.

A sequent $A_1,\dots, A_n \to B$ is said to be valid in a structure $\mathfrak{A}$, if the formula $A_1 \wedge \cdots \wedge A_n \supset B$ is valid in $\mathfrak{A}$. In particular, $A_1,\dots, A_n \to$ is valid in $\mathfrak{A}$ if $A_1 \wedge \cdots \wedge A_n$ is not valid in $\mathfrak{A}$.

Let T be a theory on SP. That is, T is a formal system obtained from SP by adding some sequents as its axioms. Then a structure $\mathfrak{A}$ is called *a model of* T if all the axioms of T are valid in $\mathfrak{A}$. Now, we can show the completeness theorem of SP.

Theorem 2.1. *Let T be a theory on SP. Then a sequent $\Gamma \to A$ is provable in T if and only if it is valid in any model of T.*

Remark 2.2. The semantics mentioned above tells us that SP is a 3-valued logic. To treat partial functions, some other 3-valued logics were introduced, e.g, by Kleene [5] and McCarthy [7]. Each of the truth tables introduced by them differs from those of Ebbinghaus. But they can be dealt with in SP as is shown below. First, extending the notion of formulas, we consider the expression of the form $(A \Rightarrow B;\, C)$ for formulas A, B, C, as a formula. This formula represents the *if-then-else* operation of [7]. Now we add the following rules of inference to SP.

a) $$\frac{\Gamma, A \longrightarrow B}{\Gamma, A \longrightarrow (A \Rightarrow B;\, C)}$$

b) $$\frac{\Gamma, \neg A \longrightarrow C}{\Gamma, \neg A \longrightarrow (A \Rightarrow B;\, C)}$$

c) $$\frac{\Gamma, A \supset B \longrightarrow D}{\Gamma, (A \Rightarrow B;\, C) \longrightarrow D}$$

d) $$\frac{\Gamma, \neg A \supset C \longrightarrow D}{\Gamma, (A \Rightarrow B;\, C) \longrightarrow D}$$

e) $$\frac{\Gamma \longrightarrow \Delta(A \Rightarrow B;\, C)}{\Gamma \longrightarrow \Delta A}$$

f) $$\frac{\Gamma \longrightarrow \Delta B}{A, \Gamma \longrightarrow \Delta(A \Rightarrow B;\, C)}$$

g) $$\frac{\Gamma \longrightarrow \Delta C}{\neg A, \Gamma \longrightarrow \Delta(A \Rightarrow B;\, C)}$$

h) $$\frac{\Delta B, \Gamma \longrightarrow D}{A, \Delta(A \Rightarrow B;\, C), \Gamma \longrightarrow D}$$

i) $$\frac{\Delta C, \Gamma \longrightarrow D}{\neg A, \Delta(A \Rightarrow B;\, C), \Gamma \longrightarrow D}$$

Let $A \cong B$ be an abbreviation of $(A \equiv B) \wedge (\neg A \equiv \neg B)$ and $(A \overset{K}{\Longrightarrow} B; C)$ be an abbreviation of

$$(B \cong C \Rightarrow B; (A \Rightarrow B; C)).$$

Then, as shown in [6], other connectives of Kleene (or McCarthy) can be defined by formulas containing only the connective ($\overset{K}{\Longrightarrow}$;) (or ($\Rightarrow$;), respectively).

§3. The Formal Theory of Partial Recursive Functions

We shall construct a formal theory of partial recursive functions on *SP*. To do so, we first introduce axiom schemata which say that Kleene's recursion theorem holds in our theory.

A formula A is said to be *a system of equations* (with respect to $f_1, \ldots, f_m$) if A is of the form

$$\bigwedge_{i=1}^{m} \forall x_{i1} \ldots \forall x_{in_i}(f_i(x_{i1}, \ldots, x_{in_i}) \simeq t_i),$$

where $f_1, \ldots, f_m$ are mutually distinct function variables and each t_i is a term containing no function variables other than $f_1, \ldots, f_m$, and no individual variables other than $x_{i1}, \ldots, x_{in_i}$. Sometimes, a system of equations A with respect to $f_1, \ldots, f_m$ is written as $A<f_1, \ldots, f_m>$. Now, *the axiom schemata R* consist of sequents of the form

$$\longrightarrow \exists f_1 \ldots \exists f_m(A<f_1, \ldots, f_m> . \wedge . (\forall g_1 \ldots \forall g_m(A<g_1, \ldots, g_m> \supset \bigwedge_{i=1}^{m} \forall x_{i1} \ldots \forall x_{in_i}(f_i(x_{i1}, \ldots, x_{in_i}) \subseteq g_i(x_{i1}, \ldots, x_{in_i}))))),$$

where A is a system of equations and $A<g_1, \ldots, g_m>$ is a formula obtained from A by replacing each $f_1, \ldots, f_m$ by $g_1, \ldots, g_m$, respectively.

The axiom schemata R mean that any system of equations has *the minimum solution*. We call the theory obtained from *SP* by adding R as $T(R)$. Since the width of the universe of partial functions is not mentioned in $T(R)$, minimum solutions obtained by R may not be the intended ones. More precisely, there may be a model $\mathfrak{A}$ of $T(R)$ such that the minimum solution of a system of equations in $\mathfrak{A}$ is not minimum in the total model with the same domain as $\mathfrak{A}$.

We can say that in any total model of $T(R)$, the axiom schemata R determine abstract partial recursive functions. (Cf. [2].) But we shall discuss only partial recursive functions on natural numbers in this paper. Now, we give a formal theory of partial recursive functions PRN on SP. The language of PRN consists of an individual constant 0 and two unary function constants S and P, which designate the successor and predecessor functions. In the following we abbreviate axioms of the form $\rightarrow A$ as A.

I) Axioms for number theory.

a. $\neg(Sx=0)$,

b. $Sx=Sy \supset x=y$,

c. $Px=y \equiv Sy=x$,

d. $A_x[0] \wedge \forall x(A \supset A_x[Sx]) . \supset . A$.

II) Axioms for partial recursive functions.

a. *Axiom schemata R*,

b. $\forall f^{(n+1)}(\forall x_1 \dots \forall x_n \forall y(f(x_1,\dots, x_n, y) \subseteq f(x_1,\dots, x_n, Sy))$

$\supset \exists h^{(n)} \forall x_1 \dots \forall x_n \forall z(h(x_1,\dots, x_n)=z \equiv \exists y(f(x_1,\dots, x_n, y)=z)))$,

for $n \geq 1$.

The axioms II) b. assert that if a function $f(x_1,\dots, x_n, y)$ is monotone with respect to y then the limit function of $f(x_1,\dots, x_n, y)$ exists.

The formal theory PRN_ω is obtained from PRN by adding the following infinitary rule.

III) ω-rule.

$$\frac{\Gamma \longrightarrow A_x[\bar{n}] \quad n=1, 2,\dots}{\Gamma \longrightarrow A},$$

where $\bar{n}$ denotes the term $\underbrace{SS \ldots S}_{n}0$.

The standard model $\mathfrak{N}$ of PRN is the total model whose domain is the set of natural numbers and whose interpretation for each constant is defined in the obvious way.

Example 3.1. 1) Consider the following system of equations,

$$f(x_1, \ldots, x_n) \simeq Sf(x_1, \ldots, x_n).$$

We have that any solution of the above system of equations satisfies the condition that $\forall x_1 \ldots \forall x_n(\sim \varDelta f(x_1, \ldots, x_n))$, since $\neg(x = Sx)$ holds in PRN. Thus in PRN the existence of the totally undefined n-ary function is ascertained.

2) Consider the following system of equations, where Q is any atomic predicate;

$$\forall \mathfrak{x}(f_1(\mathfrak{x}) \simeq f_2(0, \mathfrak{x})) \wedge \forall \mathfrak{x} \forall y(f_2(y, \mathfrak{x}) \simeq (Q(y, \mathfrak{x}) \Rightarrow y; f_2(Sy, \mathfrak{x}))).$$

Let f^* be the minimum solution for f_1. Then we can prove in PRN that

a. $Q(y, \mathfrak{x}),\ \forall z(z < y \supset \neg Q(z, \mathfrak{x})) \longrightarrow f^*(\mathfrak{x}) = y,$

b. $\sim \varDelta Q(y, \mathfrak{x}),\ \forall z(z < y \supset \neg Q(z, \mathfrak{x})) \longrightarrow\ \sim \varDelta f^*(\mathfrak{x}),$

c. $\forall y \neg Q(y, \mathfrak{x}) \longrightarrow\ \sim \varDelta f^*(\mathfrak{x}).$

Thus, the Kleene's μ-operator can be dealt with in our theory. (Cf. [7].)

Now, we prove the following two lemmas, which will be used in later sections.

Lemma 3.2. *Suppose that g is an n-ary function variable and that f_1 and f_2 are $(n+m)$-ary function variables. Then for any t*

$$\forall \mathfrak{x}(f_1(\mathfrak{x}, \mathfrak{y}) \subseteq f_2(\mathfrak{x}, \mathfrak{y})) \supset (t_g[\lambda \mathfrak{x} f_1(\mathfrak{x}, \mathfrak{y})] \simeq t_g[\lambda \mathfrak{x} f_2(\mathfrak{x}, \mathfrak{y})])$$

is provable in PRN.

Proof. It suffices to prove the lemma, when t has only one occurrence of g, which is of the form $g(t')$. By the assumption it follows that $f_1(t', \mathfrak{y}) \subseteq f_2(t', \mathfrak{y})$. So we have only to show that for every term t_1, t_2, s, $t_1 \subseteq t_2$ implies $s_x[t_1] \subseteq s_x[t_2]$. But this can be verified by induction on the length of s.

Lemma 3.3. *Suppose that s is a term containing no function variables and no individual variables other than $\mathfrak{x}$, and that t is a term containing no function variables other than $g(\mathfrak{x})$ and no individual variables other than $\mathfrak{x}$. If f_1 and f_2 are solutions of the following system of equations, called the primitive recursion,*

$$\forall \mathfrak{x} \forall y (f(\mathfrak{x}, y) \simeq (y=0 \Rightarrow s;\ t_g[\lambda \mathfrak{x} f(\mathfrak{x}, Py)])),$$

then $f_1(\mathfrak{x}, y) \simeq f_2(\mathfrak{x}, y)$ holds in PRN.

Proof. We prove the lemma by induction on y. Clearly, $f_1(\mathfrak{x}, 0) \simeq s \simeq f_2(\mathfrak{x}, 0)$. Suppose that $f_1(\mathfrak{x}, y) \simeq f_2(\mathfrak{x}, y)$. Since $\neg(Sy=0)$ holds, $f_1(\mathfrak{x}, Sy) \simeq t_g[\lambda \mathfrak{x} f_1(\mathfrak{x}, PSy)] \simeq t_g[\lambda \mathfrak{x} f_1(\mathfrak{x}, y)]$. Similarly, $f_2(\mathfrak{x}, Sy) \simeq t_g[\lambda \mathfrak{x} f_2(\mathfrak{x}, y)]$. By the hypothesis of induction and Lemma 3.2, it follows that

$$t_g[\lambda \mathfrak{x} f_1(\mathfrak{x}, y)] \simeq t_g[\lambda \mathfrak{x} f_2(\mathfrak{x}, y)].$$

Hence,

$$f_1(\mathfrak{x}, Sy) \simeq f_2(\mathfrak{x}, Sy).$$

We notice here that in a certain sense PRN and PRN_ω are extensions of second order arithmetic A and A_ω of [4]. For example, Leśniewski schemata can be expressed in our systems as $\bigwedge_{i=1}^{m} \forall \mathfrak{x}_i \Delta g_i(\mathfrak{x}_i) \supset \exists f \forall \mathfrak{x}(f(\mathfrak{x})=t)$, where t is a term containing only $g_1, \dots, g_m$ as function constants or variables. It is obvious that these formulas are provable in PRN by using the axiom schemata R. Some of the results in the next section have a close connection with those in [4].

§4. Extensions of *PRN*

First, we define conservative extensions PRN^* and PRN^*_ω of PRN and PRN_ω, respectively. Suppose that $A<f_1, \dots, f_m>$ is a system of equations. The formula B_A is defined as

$$A<f_1,\ldots,f_m> \wedge (\forall g_1 \ldots \forall g_m(A<g_1,\ldots, g_m> \supset \bigwedge_{i=1}^{m} \forall \mathfrak{x}_i(f_i(\mathfrak{x}_i) \subseteq g_i(\mathfrak{x}_i)))).$$

Clearly, $\longrightarrow \exists f_1 \ldots \exists f_m B_A$ is an instance of R. Since it holds in PRN that

$$B_A, (B_A)_{f_1,\ldots,f_m}[f'_1,\ldots,f'_m] \longrightarrow \bigwedge_{i=1}^{m} \forall \mathfrak{x}_i(f_i(\mathfrak{x}_i) \simeq f'_i(\mathfrak{x}_i)),$$

functions $f_1, .., f_m$ satisfying B_A are determined uniquely up to $\simeq$. Now, we construct a theory PRN^* from PRN by adding a new function constant $\mu f_i A$ $(i \leqq m)$ and a new axiom

$$\forall f_1 \ldots \forall f_m(B_A \equiv \bigwedge_{i=1}^{m} \forall \mathfrak{x}_i(f_i(\mathfrak{x}_i) \simeq \mu f_i A(\mathfrak{x}_i)))$$

for each system of equations $A<f_1,\ldots,f_m>$ of $L(PRN)$, the language of PRN.

For each formula C of PRN^*, define a formula C_0 of PRN as follows. Let $\mu f_1 A_1,\ldots, \mu f_m A_m$ be all the new function constants appearing in C. We assume for simplicity that each A_i is of the form $A_i<f_i>$. We obtain a formula C' from C by replacing each $\mu f_i A_i$ by f_i. Now the formula C_0 is $\forall f_1 \ldots \forall f_m(\bigwedge_{i=1}^{m} B_{A_i} \supset C')$.

Theorem 4.1. *For each formula C of PRN^*, C is provable in PRN^* if and only if C_0 is provable in PRN.*

From this theorem it follows that PRN^* is a conservative extension of PRN. By the same way, we can construct a conservative extension PRN^*_ω from PRN_ω. In the following, we assume the consistency of these theories.

We notice that the axiom schemata R of PRN^* can be restricted only to systems of equations of $L(PRN)$. In other words, for any system of equations $A<f,\ldots>$ of $L(PRN^*)$ there exists a system of equations $B<f,\ldots>$ of $L(PRN)$ such that for any $\mathfrak{x}$ $\mu fA(\mathfrak{x}) \simeq \mu fB(\mathfrak{x})$. Suppose that $\mu f_1 C_1,\ldots, \mu f_n C_n$ are all the new function constants appearing in A. $A'<f, g_1,\ldots, g_n>$ denotes the formula obtained from A by replacing each $\mu f_i C_i<f_i,\ldots>$ by a function variable g_i. Now, let B be a system of equations of PRN,

$$A'<f, g_1,\dots, g_n> \wedge \bigwedge_{i=1}^{n} C_i<g_i, \dots > .$$

Then, it is easy to see that $\mu fA(\mathfrak{x}) \simeq \mu fB(\mathfrak{x})$ for any $\mathfrak{x}$.

The function and predicate constants of $L(PRN)$ are only S, P and $=$. Thus, by the recursion theorem each system of equations $A<f_1,\dots,f_m>$ determines partial recursive functions $\varphi_1,\dots,\varphi_m$ in the standard model $\mathfrak{N}$. So, defining the interpretation of $\mu f_i A$ by φ_i, we get an extension $\mathfrak{N}^*$ of the model $\mathfrak{N}$. We say that $\mathfrak{N}^*$ is the standard model of PRN^*. Using the results in Example 3.1, we have that any partial recursive function can be introduced by the axiom schemata R and hence is of the form $(\mu fA)^{\mathfrak{N}^*}$.

Next we show how the computation of the value of $\mu fA(\overline{\mathfrak{m}})$ for any system of equations A and natural numbers $\mathfrak{m}$ is executed in PRN^*_ω. We consider only the case where A is of the form $A<f>$. Other cases can be dealt with in the similar way. In the following, $\vdash B$ means the provability of a formula B in PRN^*_ω. Suppose that $A<f>$ is $\forall\mathfrak{x}(f(\mathfrak{x})\simeq t)$. Define a function $g(\mathfrak{x}, y)$ by the condition that

1) $\forall\mathfrak{x}(\sim\Delta g(\mathfrak{x}, 0))$ and

2) $\forall\mathfrak{x}\forall y(g(\mathfrak{x}, Sy)\simeq t_f[\lambda\mathfrak{x}g(\mathfrak{x}, y)])$.

The existence and the uniqueness of such a function g can be verified by Example 3.1 and Lemma 3.3. Now, we show by induction that $\vdash g(\mathfrak{x}, y)\subseteq g(\mathfrak{x}, Sy)$. By the definition, $\sim\Delta g(\mathfrak{x}, 0)$. Thus, $g(\mathfrak{x}, 0)\subseteq g(\mathfrak{x}, S0)$. Suppose that $g(\mathfrak{x}, y)\subseteq g(\mathfrak{x}, Sy)$ holds. Then we have that $t_f[\lambda\mathfrak{x}g(\mathfrak{x}, y)]\subseteq t_f[\lambda\mathfrak{x}g(\mathfrak{x}, Sy)]$. So $g(\mathfrak{x}, Sy)\subseteq g(\mathfrak{x}, SSy)$ holds.

Using Axioms II) b, we have that there is a function h such that $\forall z(h(\mathfrak{x})=z \equiv \exists y(g(\mathfrak{x}, y)=z))$. We can get the following lemma due to [5].

Lemma 4.2. *h is the minimum solution of A.*

Proof. We first show that h is a solution of A. By the definition,

for any y $$g(\mathfrak{x}, y) \subseteq h(x).$$

By Lemma 3.2, $$t_f[\lambda\mathfrak{x}g(x, y)] \subseteq t_f[\lambda\mathfrak{x}h(\mathfrak{x})].$$

Hence, $\exists y(t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]=z) \supset (t_f[\lambda \mathfrak{x} h(\mathfrak{x})]=z)$.

Since $g(\mathfrak{x}, y) \subseteq g(\mathfrak{x}, Sy) \simeq t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]$, it holds that $h(\mathfrak{x})=z \supset \exists y(t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]=z)$. Combining these results, we obtain that $h(\mathfrak{x}) \subseteq t_f[\lambda \mathfrak{x} h(\mathfrak{x})]$.

To prove that $t_f[\lambda \mathfrak{x} h(\mathfrak{x})] \subseteq h(\mathfrak{x})$, we first show by induction on the length of t that

$$\forall z(t_f[\lambda \mathfrak{x} h(\mathfrak{x})]=z \supset \exists y(t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]=z)).$$

We prove this only when t is of the form $f(t_1, \ldots, t_k)$. We assume that for each i

$$\forall z(t_{i_f}[\lambda \mathfrak{x} h(\mathfrak{x})]=z \supset \exists y(t_{i_f}[\lambda \mathfrak{x} g(\mathfrak{x}, y)]=z)).$$

We write t'_i and t''_i instead of $t_{i_f}[\lambda \mathfrak{x} h(\mathfrak{x})]$ and $t_{i_f}[\lambda \mathfrak{x} g(\mathfrak{x}, y)]$, respectively. Then, $t_f[\lambda \mathfrak{x} h(\mathfrak{x})]$ and $t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]$ are $h(t'_1, \ldots, t'_k)$ and $g(t''_1, \ldots, t''_k, y)$. In SP, it holds that $h(t'_1, \ldots, t'_k)=z$ implies

$$(1) \qquad \exists x_1 \ldots \exists x_k(\bigwedge_{i=1}^{k}(t'_i=x_i) \wedge h(x_1, \ldots, x_k)=z)$$

By the assumption, (1) implies

$$(2) \qquad \exists x_1 \ldots \exists x_k(\bigwedge_{i=1}^{k} \exists y(t''_i=x_i) \wedge \exists y(g(x_1, \ldots, x_k, y)=z))$$

By Lemma 3.2 and the above discussion, we have in PRN that $y \leqq y'$ implies $(t''_i \subseteq t''_{i_y}[y']) \wedge (g(x_1, \ldots, x_k, y) \subseteq g(x_1, \ldots, x_k, y'))$ and that $y \leqq y' \vee y' \leq y$. Thus, (2) implies

$$(3) \qquad \exists y \exists x_1 \ldots \exists x_k(\bigwedge_{i=1}^{k}(t''_i=x_i) \wedge g(x_1, \ldots, x_k, y)=z).$$

Clearly, (3) implies $\exists y(g(t''_1, \ldots, t''_k, y)=z)$. Now, suppose that $t_f[\lambda \mathfrak{x} h(\mathfrak{x})]=z$. Then from the above discussion, it follows that $\exists y(t_f[\lambda \mathfrak{x} g(\mathfrak{x}, y)]=z)$. So, $\exists y(g(\mathfrak{x}, Sy)=z)$. This implies $\exists y(g(\mathfrak{x}, y)=z)$. Thus $h(\mathfrak{x})=z$. Hence, $t_f[\lambda \mathfrak{x} h(\mathfrak{x})] \subseteq h(\mathfrak{x})$.

Next we show that $h(\mathfrak{x})$ is the minimum solution. Let k be a solution of A. That is, $k(\mathfrak{x}) \simeq t_f[\lambda \mathfrak{x} k(\mathfrak{x})]$. We prove by induction on y that $g(\mathfrak{x}, y) \subseteq k(\mathfrak{x})$. Since $\sim \Delta g(\mathfrak{x}, 0)$, $g(\mathfrak{x}, 0) \subseteq k(\mathfrak{x})$. Suppose that

$g(x, y) \subseteq k(x)$. By Lemma 3.2, $g(x, Sy) \simeq t_f[\lambda x g(x, y)] \subseteq t_f[\lambda x k(x)] \simeq k(x)$. Thus, $\forall y(g(x, y) \subseteq k(x))$. From this, it follows that $\exists y(g(x, y)=z) \supset (k(x)=z)$. Hence $h(x) \subseteq k(x)$. This completes the proof.

Lemma 4.3. *Let l and $\mathfrak{m}$ be a natural number and a k-tuple of natural numbers, respectively. Then either $\vdash g(\bar{\mathfrak{m}}, \bar{l})=\bar{n}$ holds for some natural number n or $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$ holds.*

Proof. First, we remark that

1) $m=n$ holds in $\mathfrak{N}^*$ if and only if $\vdash m=\bar{n}$,

2) $Sm=n$ holds in $\mathfrak{N}^*$ if and only if $\vdash Sm=\bar{n}$,

3) $Pm=n$ holds in $\mathfrak{N}^*$ if and only if $\vdash Pm=\bar{n}$.

Now, let $Q(l)$ be the proposition which says that for any k-tuple of natural numbers $\mathfrak{m}$, either $\vdash g(\bar{\mathfrak{m}}, \bar{l})=\bar{n}$ holds for some n or $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$. We prove $Q(l)$ by induction on l. If $l=0$, then $\vdash \sim \Delta g(\bar{\mathfrak{m}}, 0)$ by the assumption. Thus $Q(0)$ holds. Suppose that $Q(l)$ holds. By the definition, $\vdash g(\bar{\mathfrak{m}}, S\bar{l}) \simeq t^*_f[\lambda x g(x, \bar{l})]$, where t^* is $t_{\mathfrak{x}}[\bar{\mathfrak{m}}]$. By induction on the length of t^*, we show that either $\vdash t^*_f[\lambda x g(x, \bar{l})]=n$ for some n or $\vdash \sim \Delta t^*_f[\lambda x g(x, \bar{l})]$ holds.

1) The case where t^* is 0.
It is obvious that $\vdash t^*_f[\lambda x g(x, \bar{l})]=0$.

2) The case where t^* is of the form St'.
Then $t^*_f[\lambda x g(x, \bar{l})]$ is equal to $S(t'_f[\lambda x g(x, \bar{l})])$. By the hypothesis, either $\vdash t'_f[\lambda x g(x, \bar{l})]=\bar{n}$ for some n or $\vdash \sim \Delta t'_f[\lambda x g(x, \bar{l})]$. In the first case, it follows that $\vdash t^*_f[\lambda x g(x, \bar{l})]=S\bar{n}=\overline{n+1}$. In the second case, $\vdash \sim \Delta t^*_f[\lambda x g(x, \bar{l})]$ holds, since $\Delta S(t'_f[\lambda x g(x, \bar{l})]) \supset \Delta t'_f[\lambda x g(x, \bar{l})]$ holds in PRN^*_ω.

3) The case where t^* is of the form $f(t_1,\ldots, t_k)$.
Then $t^*_f[\lambda x g(x, \bar{l})]$ is equal to $g(t'_1,\ldots, t'_k, \bar{l})$, where each t'_i denotes $t_{i_f}[\lambda x g(x, \bar{l})]$. Suppose that $\vdash t_{i_f}[\lambda x g(x, \bar{l})]=\bar{n}_i$ holds for each i. By the hypothesis that $Q(l)$ holds, $\vdash g(\bar{n}_1,\ldots, \bar{n}_k, \bar{l})=\bar{n}$ for some n or $\vdash \sim \Delta g(\bar{n}_1,\ldots, \bar{n}_k, \bar{l})$. In the first case, $\vdash t^*_f[\lambda x g(x, \bar{l})]=\bar{n}$ and in the second case $\vdash \sim \Delta t^*_f[\lambda x g(x, \bar{l})]$. On the other hand, if there exists at least one

i such that $\vdash \sim \Delta t_{i_f}[\lambda x g(x, \bar{l})]$, we have that $\vdash \sim \Delta t_f^*[\lambda x g(x, \bar{l})]$.

Using the remark mentioned first, we can prove other cases similarly. Thus we have $Q(l+1)$.

Theorem 4.4. *Let g and h be functions mentioned above and $\mathfrak{m}$ be a k-tuple of natural numbers. Then*

1) $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$ *for any natural number l if and only if* $\vdash \sim \Delta h(\bar{\mathfrak{m}})$,

2) $\vdash g(\bar{\mathfrak{m}}, \bar{l}) = \bar{n}$ *for some natural number l if and only if* $\vdash h(\bar{\mathfrak{m}}) = \bar{n}$.

Proof. 1) Suppose that $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$ for any l. Then by ω-rule,

$$\vdash \forall x \sim \Delta g(\bar{\mathfrak{m}}, x)\,. \tag{1}$$

On the other hand,

$$\Delta h(\bar{\mathfrak{m}}) \longrightarrow \exists y \Delta g(\bar{\mathfrak{m}}, y)\,, \tag{2}$$

and

$$\exists y \Delta g(\bar{\mathfrak{m}}, y), \qquad \forall x \sim \Delta g(\bar{\mathfrak{m}}, x) \longrightarrow \tag{3}$$

are provable. From (2) and (3), it follows that

$$\Delta h(\bar{\mathfrak{m}}), \qquad \forall x \sim \Delta g(\bar{\mathfrak{m}}, x) \longrightarrow$$

Using (1), we get $\Delta h(\bar{\mathfrak{m}}) \rightarrow$. Thus $\vdash \sim \Delta h(\bar{\mathfrak{m}})$. Conversely, suppose that $\vdash \sim \Delta h(\bar{\mathfrak{m}})$. Then $\Delta h(\bar{\mathfrak{m}}) \rightarrow$. Since $\exists y \Delta g(\bar{\mathfrak{m}}, y) \rightarrow \Delta h(\bar{\mathfrak{m}})$ and $\Delta g(\bar{\mathfrak{m}}, \bar{l}) \rightarrow \exists y \Delta g(\bar{\mathfrak{m}}, y)$ are provable, $\Delta g(\bar{\mathfrak{m}}, \bar{l}) \rightarrow$. Hence $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$, for any l.

2) Suppose that $\vdash g(\bar{\mathfrak{m}}, \bar{l}) = \bar{n}$. Then $\vdash \sim \exists y (g(\bar{\mathfrak{m}}, y) = \bar{n})$. Hence $\vdash h(\bar{\mathfrak{m}}) = \bar{n}$. Conversely, suppose that $\vdash h(\bar{\mathfrak{m}}) = \bar{n}$. If $\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$ for any l then $\vdash \sim \Delta h(\bar{\mathfrak{m}})$ as we have just proved. But this is a contradiction, since we assume the consistency of PRN_ω^*. Thus, there is an l such that $\not\vdash \sim \Delta g(\bar{\mathfrak{m}}, \bar{l})$. By Lemma 4.3, there is a natural number n' such that $\vdash g(\bar{\mathfrak{m}}, \bar{l}) = \bar{n}'$. Then $\vdash \bar{n} = \bar{n}'$. Hence $\vdash g(\bar{\mathfrak{m}}, \bar{l}) = \bar{n}$ for some l.

By Lemmas 4.2, 4.3 and Theorem 4.4, we have the following theorem.

Theorem 4.5. *For any system of equations* $A<f, ..>$ *and any natural number* $\mathfrak{m}$, *either* $\vdash \mu f A(\overline{\mathfrak{m}}) = \bar{n}$ *for some natural number* n *or* $\vdash \sim \Delta \mu f A(\overline{\mathfrak{m}})$.

A model $\mathfrak{A}$ of *PRN* (or *PRN**) is an *ω-model* if every individual a of $\mathfrak{A}$ is $\bar{n}^{\mathfrak{A}}$ for some natural number n. We can prove the following theorem in the similar way as Henkin-Orey theorem for theories on classical logic (see, e.g. [9]).

Theorem 4.6. *A formula of PRN (or PRN*) is a theorem of* PRN_ω *(or* PRN^*_ω*) if and only if it is valid in every ω-model of PRN (or PRN*).*

By this theorem, we can get the following theorems similarly as [4].

Theorem 4.7. *Let A be a formula of PRN which contains no second order quantifiers. Then* $\forall f_1 \ldots \forall f_k A$ *is a theorem of* PRN_ω *if and only if it is valid in* $\mathfrak{N}$.

Theorem 4.8. *Let A be a formula of PRN* which contains no second order quantifiers. Then* $\forall f_1 \ldots \forall f_k A$ *is a theorem of* PRN^*_ω *if and only if it is valid in* $\mathfrak{N}^*$.

Proof. Let $\mathfrak{A}$ be any ω-model of *PRN**. Then we have only to show that for every system of equations $A<f,...>$ and every natural number $\mathfrak{m}$,

1) $\mathfrak{A}(\mu f A(\overline{\mathfrak{m}}) = \bar{n}) = T$ if and only if $\mathfrak{N}^*(\mu f A(\overline{\mathfrak{m}}) = \bar{n}) = T$, an

2) $\mu f A(\overline{\mathfrak{m}})^{\mathfrak{A}}$ is not defined if and only if $\mu f A(\overline{\mathfrak{m}})^{\mathfrak{N}^*}$ is not defined. But this can be verified by using Theorem 4.5.

As a corollary of Theorem 4.8, we have that for any partial recursive function φ there is a system of equations $A<f,...>$ such that for every natural number $\mathfrak{m}$,

1) $\varphi(\mathfrak{m}) = n$ if and only if $\vdash \mu f A(\overline{\mathfrak{m}}) = \bar{n}$ and

2) $\varphi(\mathfrak{m})$ is undefined if and only if $\vdash \sim \Delta\mu f A(\overline{\mathfrak{m}})$.

We say that the function constant $\mu f A$ represents φ.

Theorem 4.8 says that PRN_ω^* is powerful enough to prove many theorems about partial recursive functions. For example, the enumeration theorem for partial recursive function in [5] is expressed in the following form:

There exists a system of equations $A<f,\dots>$ *such that for any system of equations* $B<g,\dots>$,

$$\exists y \forall x_1,\dots \forall x_n(\mu g B(x_1,\dots, x_n) \simeq \mu f A(y, x_1,\dots, x_n))$$

is provable in PRN_ω^*.

The above discussion suggests a way of constructing an extension of PRN_ω^*. We have taken $\mathfrak{N}^*$ for the standard model of PRN^*. But, another standard model of PRN^* can be taken. Let $\mathfrak{N}^+$ be a structure whose domain is the set of natural numbers and whose functions are all of partial recursive functions. We can show that $\mathfrak{N}^+$ is a model of PRN^*, since Axioms II) b holds in $\mathfrak{N}^+$. Now, consider the partial recursive function $U(\mu y T_n(z, x_1,\dots, x_n, y))$ in [5]. Let Φ_n be the $(n+1)$-ary function constant which represents the function $U(\mu y T_n(z, x_1,\dots, x_n, y))$. Now, PRN^+ (or PRN_ω^+) is the theory obtained from PRN^* (or PRN_ω^*) by adding the following axioms;

IV) $$\forall f^{(n)} \exists z \forall x_1 \dots \forall x_n(f(x_1,\dots, x_n) \simeq \Phi_n(z, x_1,\dots, x_n)).$$

Clearly, $\mathfrak{N}^+$ is a model of PRN^+. Similarly as Theorem 4.6, we can prove that for any formula A of PRN^+, A is provable in PRN_ω^+ if and only if it is valid in any ω-model of PRN^+. Now, let $\mathfrak{A}$ be any ω-model of PRN^+ and α be any n-ary function of $\mathfrak{A}$. For simplicity, we assume that the domain of $\mathfrak{A}$ is the set of natural numbers. Since Axioms IV) are valid in $\mathfrak{A}$, there is a natural number m such that

$<m_1,\dots, m_n, k> \in \alpha$ if and only if $<m, m_1,\dots, m_n, k> \in \Phi_n^{\mathfrak{A}}$ for every natural number $m_1,\dots, m_n, k$. By Theorem 4.5, $\Phi_n^{\mathfrak{A}}$ is equal to $\Phi_n^{\mathfrak{N}^+}$. Clearly, $\Phi_n^{\mathfrak{N}^+}$ is (the graph of) a partial recursive function. So, α is also partial recursive. Conversely, it is obvious that every partial recursive function is a function of $\mathfrak{A}$. Thus. $\mathfrak{A}$ is isomorphic to $\mathfrak{N}^+$.

Theorem 4.9. *For any formula A of* PRN^+, *A is provable in* PRN^+_ω *if and only if it is valid in* $\mathfrak{N}^+$. *Hence,* PRN^+_ω *is complete.*

Applicability of our systems to problems in mathematical theory of computation will be obvious. Since problems of equivalence, correctness and termination of programs about natural numbers can be expressed by formulas of PRN^*_ω of the form mentioned in Theorem 4.8, they can be treated completely in PRN^*_ω. For another example, theorems in [6] can be proved formally in the theory $T(R)$. $T(R)$ has a close relation with the formal system in [8]. To strength $T(R)$, some axioms like Axioms II) b. are necessary. But in general case we can not express II) b. in our language. So some rules like the induction in [8] will be needed.

References

[1] Ebbinghaus, H.-D., Über eine Prädikatenlogik mit partiell definierten Prädikaten und Funktionen, *Arch. Math. Logik Grundlagenforsch.*, **12** (1969), 39–53.

[2] Friedman, H., Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory, *Logic Colloquium* '69 (ed. by R.O. Gandy and C.E.M. Yates) (1971), 361–389.

[3] Gentzen, G., Untersuchungen über das logische Schliessen, *Math. Zeitschrift*, **39** (1934/35), 176–210, 405–431.

[4] Grzegorczyk, A., A. Mostowski and C. Ryll-Nardzewski, The classical and the ω-complete arithmetic, *J. Symbolic Logic*, **23** (1958), 188–206.

[5] Kleene, S. C., *Introduction to metamathematics*, Van Nostrand, New York, North Holland, Amsterdam and Noordhoff, Groningen, 1952.

[6] Manna, Z. and J. McCarthy, Properties of programs and partial function logic, *Machine Intelligence* **5** (ed. by B. Meltzer and D. Michie) (1969), 27–37.

[7] McCarthy, J., A basis for a mathematical theory of computation, *Computer programming and formal systems*, (ed. by P. Braffort and D. Hirschberg), North Holland, Amsterdam (1963), 33–69.

[8] Scott, D., A type-theoretical alternative to CUCH, ISWIM, OWHY, (unpublished) Oxford, 1969.

[9] Shoenfield, J. R., *Mathematical logic*, Addison-Wesley, Reading, Mass., 1967.