Title	Note on the Strong Normalizability of the Logic with Self-Referential Predicates
Sub Title	
Author	岡田, 光弘(Okada, Mitsuhiro)
Publisher	三田哲學會
Publication year	1993
Jtitle	哲學 No.95 (1993. 7) ,p.1- 14
JaLC DOI	
Abstract	We consider a logical system in which the second order terms can be also used as first order terms. As a consequence, predicates can be taken as arguments of another predicate. In other words, a predicate can refer to predicates. The proposed framework seems very different from the tradition of formal logic which has been developped since Aristotle, where the distinction between the predicates and the individuals are essential. While in the traditional (Aristotelean) logical language a typical and basic sentence is of the form, e.g., Love(Socrates, Phaidon) "Socrates loves Phaidon", in our proposed language we could also express e.g., Love(Socrates, Love), Socrates "loves (the idea of) love", (which can be also express as Love(Socrates, λx, y. Love(x, y) by the usual η-conversion), or a sentence Love(God, λx ∀y Love(x, y)), "God loves (the idea of) loving everyone", or a predicate on x such as Love(x, λy Love(x, y)), "x loves x's own love". The purpose of this note is to give some proof-theoretic properties of the proposed logic; in particular, the strong normalizability is shown by slightly modifying Girard's strong normalizability proof for the traditional higher order logic. The use of this proposed logical language for the analysis of "self-references" and "intensionality" as Well as a connection with natural language semantics is discussed in Okada", and the connection with mobile communication processes and with linear logic are discussed in Okada.
Notes	
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00150430-00000095-0001

Note on the Strong Normalizability of the Logic with Self-Referential Predicates

Mitsuhiro OKADA*

We consider a logical system in which the second order terms can be also used as first order terms. As a consequence, predicates can be taken as arguments of another predicate. In other words, a predicate can refer to predicates. The proposed framework seems very different from the tradition of formal logic which has been developped since Aristotle, where the distinction between the predicates and the individuals are essential. While in the traditional (Aristotelean) logical language a typical and basic sentence is of the form, e.g., *Love*(*Socrates*, *Phaidon*) "Socrates loves Phaidon", in our proposed language we could also express e.g., *Love*(*Socrates*, *Love*), Socrates "loves (the idea of) love", (which can be also express as *Love*(*Socrates*, $\lambda x, y.$ *Love*(x, y) by the usual η-conversion), or a sentence *Love*(*God*, $\lambda x \forall y$ *Love*(x, y)), "God loves (the idea of) loving everyone", or a predicate on x such as *Love*$(x, \lambda y$ *Love*$(x, y))$, "x loves x's own love". The purpose of this note is to give some proof-theoretic properties of the proposed logic; in particular, the strong normalizability is shown by slightly modifying Girard's strong normalizability proof for the traditional higher order logic. The use of this proposed logical language for the analysis of "self-references" and "intensionality" as well as a connection with natural language semantics is discussed in Okada[4], and the connection with mobile communication processes and with linear logic are discussed in Okada[5].

* 慶應義塾大学文学部助教授 (哲学)

1. INTRODUCTION

The usual traditional logical languages are based on the language of predicate logic, where the distinction between the predicates and the individuals is essential. This traditional framework is considered as a natural extension of Aristotle's formalization based on the subject-predicate relation.

We propose the logic language[3] in which the above traditional framework is discarded. The proposed language destroys the distinction between the higher order language and the first order language, which has been the basis of the modern tradition since B. Russell, where the distinction was introduced by Russell[8] in order to avoid the well-known logical paradox., Russell's paradox[7]. In contrast to the Russell's solution, our proposed logical language identifies the first order terms and the second order terms, without producing Russell's paradox. As the consequence, the distinction between the individuals and the predicates disappears in this proposed logic, and a predicate can refer to other predicates (in particular to itself). In the traditional (Aristotelean) logical language one expresses, e.g. a sentence *Love*(*Socrates*, *Phaidon*) "Socrates loves Phaidon", while in our proposed language we could also express e.g., *Love*(*Socrates*, *Love*), "Socrates loves (the idea of) love", (which can be also express as *Love*(*Socrates*, $\lambda x, y.$ *Love*(x, y) by the usual η-conversion), or *Love*(*Socrates*, *Intelligent*), "Socrates loves the idea of "intelligent" i.e., "intelligence", or a sentence *Love*(*God*, $\lambda x \forall y$ *Love*(x, y)), "God loves (the idea of) loving everyone", or a predicate on x such as *Love*$(x, \lambda y$ *Love*$(x, y))$, "x loves x's own love". A predicate can also take a sentence as an argument in our language; for example, we can express *Love*(*Tom*, *Love*(*God*, $\lambda x \forall y$ *Love*(x, y))), which means that Tom loves (the idea of the fact) that "God loves (the idea of) loving everyone".

The above examples would suggest that our new logical language

could be more suitable to express intensionality, while the traditional logical framework (i.e., the traditional predicate logic and their higher order versions) is based on extensionality. In fact, the description from the point of view of extensionality has been the most standard stand-point in the development of the traditional logic since Aristotle, which could explain why the traditional predicate logic, higher order logic and set theory have been well studied with the classical (Tarski-style) model theory based on the extensional interpretation of logical language. The recent development of the Kripke-style possible world semantics is also considered as an "extensional" interpretation of "modality", of "intuitionistic implication" and of "intensionality", instead of a direct intensional interpretation, in the sense that each possible world is interpreted totally by extensionality. On the other hand, the proposed language in which predicates can refer to predicates seems to give a very different framework to analyze intensionality, as is discussed in Okada[4]. The ability of self-references is another distinguished feature of this proposed logical language. It is due to the very flexible typing definition of this language. On the other hand, a completely untyped setting would give not only the ability of self-references but also cause the contradiction, as well-known in the early history of modern logic. This is exactly the situation of Russell's paradox of the Frege's and Cantor's naive (untyped or non-hierarchical) logical languages. Our proposed language is strong enough to allow self-referential expressions, but at the same time is strict enough to avoid the Russell's Paradox or the likes. We shall shortly remark what is the essential difference in the syntactical level between those two languages in Section 3, by analyzing Russell's Paradox in our framework. An important point which we would like to claim here is that semantic interpretation by means of the hierarchical or typed structure of sets (e.g. Russell's type theory and the Zermelo-Freankel's set theory which is based

on the cumulative models) is not essential to avoid the Russell's paradox or the likes, although it is an easy sufficient condition to avoid the paradox and the development of modern logic has heavely depended on those naive ways for a solution, to the paradox. Explicite or implicite extensionalisms and its various implications have influenced logical pholosophy and logical semantics theory. (For example, the traditional logical framework for the hierarchical (or cumulative) interpretation of set structure implies a "hidden" logical atomism, which in turn implies the theory of description. See (4) for the discussion.) A more flexible logical framework and a more dynamic semantic interpretation could replace the traditional framework based on a naive extensionalism. The goal of the series of our papers is to provide such a new framework of logic. The purpose of this note is to give some proof-theoretic properties of the proposed basic logical language in which predicates can refer to predicates; in particular, the strong normalizability is shown by slightly modifying Girard's strong normalizability proof for the traditional higher order logic.

2. LANGUAGE FOR SELF-REFERENTIAL PREDICATES

We recall the self-referential logical language from Okada[4]. The language of the self-referential logic contains all usual logical conne ctives, ecept that the distinction between the second order quantifiers and the first order quantifiers disappears in this language. The language is obtained by identifying the first order terms and the second order terms from the usual second order logical language. The terms and the formulas are defined simultaneously as follows;

1. *A* predicate symbol, say a, of n-arity is a term.
2. *A* variable, say x, of n-arity is a term.
3. If a is a constant (of n-arity) and t is a vector of terms (of length n) then $a(t)$ is an atomic formula.

4. If x is a variable (of n-arity) and t is a vector of terms (of length n) then $x(t)$ is an atomic formula.

5. A formula is composed of smaller formulas by using the logical connectives in the usual ways from the atomic formulas.

6. If A is a formula and x is a vector of variables of length n (including the case $n=0$), and if all variables x appearing in A appear at the first order positions, then $\lambda x.A$ is a term of arity n. (In particular, any formula A is a term of 0-arity.) Here the notions of first order position and of second order position are given below.

Although we identify the first order terms and the second order terms, we want to distinguish an occurrence of a variable at a first order position and at a second order position. An occurrence of a variable x of the form $a(t, x, s)$, where s and t are vectors of terms and a is a constant or a variable, for a subexpression $a(t, x, s)$ of a term is said to occur at a first order position. An occurrence of a variable is said to be at a second order position if it is not at a first order position.

Examples: if $x(x)$ occurs in a formula or a term, the first x is at the second order position and the second x is at the first order position. In $x(\lambda y.(b(y)\vee x(y, x)), c)\wedge D$, the first and the second occurrences of x are at second order positions and the third one is at the first order position.

Note that our formalism does not have β-reduction nor η-reduction as a formal rule (but as the meta-level) and does not have the application operator, either. "( )" is used only for expressing the argument places of a predicate (and is not the application oparator in the sense of λ-calculus.).

The syntax for the basic logic of the self-referential language is defined in the same way as the usual traditional logic, except for the inference rules of the quantifiers. Here the quantifier rules behaves as the second order quantifier rules of the traditional

second order logic. Here we consider the Natural Deduction formulation of Gentzen (Prawitz[6]).

The elimination for $\forall$ is;

$$\frac{\begin{array}{c}\vdots\\ \forall x\, A[x]\end{array}}{A[t/x]}$$

where t is an arbitrary term of the same arity as that of x.

On the other hand, the introduction rule is as usual;

$$\frac{\begin{array}{c}\vdots\\ A[x]\end{array}}{\forall x\, A[x]}$$

where x is an "eigenvariable" (in the sense of Takeuti[9]).

The rules for $\exists$ are the dual of the above.

Both the classical and the intuitionistic versions of natural deduction system for our self-referential predicate language are defined in the same way as Prawitz[6], with the above quantifier rules.

It is very natural to ask if or not such a new language with its basic logical syntax is consistent. Thie answer might not be very clear at a slight look since the language gives a very flexible typing mechanism; the types of the arguments of a predicate may be essentially any types in the proposed language. As we shall see in the next Section, the language becomes inconsistent if we relax the constraint of the language slightly. In fact, Russell's paradox can be accommodated if we relax the type constraint a little more, as will be seen below. On the other hand, it was shown that the proposed constraint taken in our language above is strong enough to be sound and complete with respect to a natural extension of the traditional (type-theoretic) denotational semantics. Soundness and completeness are given in Okada[4].

THEOREM. [Okada 4]. The logic with self-referential predi-

cates, SR, is sound and complete with respect to the term models.

There are several important versions of this self-referential logic proposed in Okada[4],[5]. In particular, various kinds of intensionality, propositional attitude, modality, could be defined by giving additional axioms on the basic logic above. We only refer to a simple additional remark in this Section. The reader is invited to counsult[4],[5] for such examples.

For example, "human" means the set of properties that the human beings have. If *human*(*intelligent*), *human*(*having-two-legs*), ... are true, then the intensional meaning of $human = \lambda x.\, human(x)$ is composed of those properties, intelligent, having two legs, ... On the other hand, one could say that "intelligent" contains "human", hence *intelligent*(*human*), although the extensions of those two may be different; for example, there might be an artifitial intelligence or an expert system which is intelligent but not human. Hence *intelligent*(*expert*−*system*). There might also be an individual human who is human but not intelligent (due to a certain disease). This situation on "human" and "intelligent", for example, is very hard to interprete the above concepts by the usual extensional (set-theoretic) interpretation (cf. (4)). (For example, one cannot interprete the above "membership" relation by neither the extensional membership nor the extensional set-inclusion. It is important to note that the s-intensional or t-intentional meaning of a word is often independent of the existance of some counter-examples. For example, a concept of "bird" contains "can fly" even if we know that there are some counter-examples, such as some penguines).

It is, of couse, more interesting to consider both "intensionality" and "extensionality" in one logical framework. In fact, one could introduce the extensional interpretation of entities in order to consider a more real setting in which both intensional meaning and extensional meaning are treated; one simple (but naive) way to introduce extensional entities is to set up a name of an entity, e.g.

mitsu, and add the following axiom;

$$\exists x\, mitsu(x) \equiv mitsu(mitsu)\ .$$

This means that being Mitsu has an only property being Mitsu itself. This setting gives the effect of the traditional use of the individual (first order) terms for the entity names. The extensional interpretation of, e.g., japanese' may be characterized as *japanese* (*mitsu*), *japanese*(*hanako*), ...

Traditionally, logical analysis has been haevily based on the individuals-predicates relation and on the extensional interpretation (or its modification by using a set of extensional interpretation, which is called possible-world semantics), which depends on the standard model (cummulative model) of set theory (either Russell's (both simple and ramified) type theory, or Zermero-Frenkael's or Godel-Bernays' set theory) where the entities are interpreted as urelements (the basis) of the cumulative models and the an "atomic predicate" are interpreted as a set of urelements. In other words, those interpretations are based on a certain "logical atomism", although the doctorine of the logical atomism was defeated in the history of theory of knowledge. On the other hand, the above situation leads us to the proposed logical language in which some membership relations among concepts or among concepts and entities can not be interpreted by the traditional (and very naive) cummulative hierarchy of the static set theoretic structure. The main purpose of our proposed language is to provide the more dynamic logical framework of the language and concept analysis. Another important featur of this proposed language is in its dynamic typing structure, as mentioned before. In our language, a predicate, for example, "know" can take either, a sentence or a concept (which is expressed by a second order lambda term) or an entity; for example, *know*(*Tom*, *Mary*), *know*(*Tom*, $\lambda x, y\ Love(x, y)$), *know*(*Tom*, $\forall x\ Love(x, y)$), *know*(*Tom*, $\forall x\ Love(Mary, x)$) are all type-matched in our language. Although our typing rule is flexible or

rough compared with, e.g. Russell's type theory, this rough typing is already strict enough to avoid Russell's paradox. We shall analyse this fact in the next Section.

3. UNTYPED SELF-REFERENTIAL LANGUAGE AND RUSSELL'S PARADOX

If we relax condition 6 of the above definition of the terms and the formulas, by deleting the second if-clause of 6, the resulting language is called the language of untyped self-referential logic. The form of 6 for the untyped self-referential logic is, hence;

6′. If A is a formula and x is a vector of variables of length n, then $\lambda x.A$ is a term of arity n.

Hence, some variables x in A may occur at a second order position. We also add.

7. If $\lambda x.A$ is a term, and if t is a vector of terms with the same arity as that of vector x, then $\lambda x.A(t)$ is a formula.

Now we consider the β-reduction;

$$(\lambda x.A)(t) \Longrightarrow A[t/x] .$$

Here, $A[t/x]$ stands for a substitution of t on the all free occurrences of x in A.

Consider Russell's construction R such that

$$\neg(x \in x) \equiv x \in R .$$

Then by interpreting $a \in b$ by $b(a)$ we can define the above R in our system. Now we consider a unary predicate $\lambda x.R$ as $\lambda x.\neg x(x)$. In the original argument of Russell[7], $R \in R \wedge \neg R \in R$, which can be simulated in our language by considering $R(R) \equiv \lambda x.\neg x(x)(\lambda x.\neg x(x)) \equiv \neg x(x)[\lambda y \neg y(y)/x] \equiv \neg R(R)$. Hence, the untyped self-referential logic implies the contradiction.

On the other hand, the restricted version of our language in

Section 2 cannot follow the above Russell's argument because in the Russell's predicate $\lambda x.x(x)$ the λ binds x at a second order position (as well as at a first order position), hence this is not a term in the sense of the language of Section 2. In fact, it is consistent, more over is complete with respect to a natural denotational semantics, as shown in Okada[4].

It is very natural to ask what is the natural extension of this system which is still consistent. One natural way to construct such an extension is to restrict the rule 7 as follows;

In 7, t is a term of the (typed) referential logic language.

In this case, for example, above R is permitted as a term, but $R(R)$ is not permit ted as a term since R is not a term of the typed referential logic language.

4. STRONG NORMALIZABILITY

The main purpose of this Section is to prove the strong normalization of our system, by modifying the Tait-Girard argument[1]. The proof reduction rules are defined in the same way as in Prawitz[7] (or Girard[2], Jouannaud-Okada[3]), for which a redex is defined as a successive application of an introduction rule and an elimination rule., which is called the β-rule, and as a successive application of an elimination rule and an introduction rule, which is called η-ruler. For our quantifier $\forall$, we have the following β- and η-rules.

(β-rule)

$$\dfrac{\dfrac{\dfrac{\vdots\,\pi[x]}{A[x]}}{\forall x\,A[x]}}{A[t/x]} \quad\Longrightarrow\quad \begin{array}{l}\vdots\,\pi[t/x]\\ A[t/x]\end{array}$$

(η-rule)

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi[x]}{\forall x\, A[x]}}{A[y/x]}}{\forall y\, A[y/x]} \quad \Longrightarrow \quad \begin{array}{c}\vdots\ \pi[y/x] \\ \forall y\, A[y/x]\end{array}$$

We use the notations from Girard[1] and Jouannaud-Okada[3]. We assume that the veader is familiar with the basic technique of strong normalizability proof in eg. Girard[1],[2] or Jouannaud-Okada[3]. For any formula A a set α_A is a set of proofs of A (i.e. of type A) which satisfies the following *reducibility* conditions;

C1. If a proof p is *reducible*, p is strongly normalizable.

C2. If p is neutral and if for any p' obtained from p by one step reduction is *reducible*, then p is *reducible*.

C3. If p is *reducible* and if q is obtained from p by finitely many reductions then q is also *reducible*.

In C2, by a neutral proof we mean a proof whose last inference rule is an elimination rule, (cf. (2), (3)).

If t is of the form $\lambda x.A$, α_t is the set of functions $\{f\colon f(u) \in \alpha_{A[u]}$ for any vector of terms u (of the same length as that of $x)\}$.

We consider the following specific construction of the reducibility set (i.e., the reducibility predicates). We use the notation $R_{A[\alpha_t,t]}$ (the reducibility predicate of type $A[\alpha_t, t]$) to denote the specific reducibility set of semantic type $A[\alpha_t,t]$. If t is of the form $\lambda x.A$, $R_{t[\alpha_s;s]}$ is the set of functions $\{f\colon f(u) \in R_{A[\alpha_s;s,u/x]}$ for any vector of terms u (of the same length as that of $x)\}$. α_t may be abbreviated as α when the type t is not important in the context.

R1. For atomic formula $a(x)$ (where a is a constant), $p[t] \in R_a(t)$ if p is a strong normalizable proof of $a(t)$.

R2. For atomic formula $x(y)$, $p[t;\ s] \in R_{x(y)[\alpha_t;s]}$ if $p[t;\ s] \in a_t(s)$.

R3. Let $q[x;\ x]$ be a non-neutral proof of type $A\to B$, and $p[x;\ x]$ be the subproof obtained by deleting the last inference rule frm q. If proof p (of type $A[t]$) in $R_A[\alpha_t;\ t]$, $q[p/x]$ is in $R_B[\alpha_t;\ t]$, then

$q[\alpha_t, t]$ is in $R_{A\to B[\alpha_t;t]}$.

R4. Let $q[y; y]$ be a non-neutral proof of type $\forall x\, A[x; y]$, and $p[x, y; x, y]$ be the subproof obtained by deleting the last inference rule frm q. Let s be a vector of terms whose arities match those of y, and let $\gamma \in \alpha_t$. If q is of type $\forall x\, A[x, y]$, and if for any t (with the same arity as x) and for any $\eta \in \alpha_t$, $q[\eta, \gamma; t, s] \in R_{A[\alpha_t,\alpha_s;t,s]}$, then q is in $R_{\forall x A[\alpha_s;s]}$.

R5. If q of type $A[x]$ is neutral, q is in $R_A[\alpha_t; t]$ if

1. q is strongly normalizable, or
2. If any p' obtained from q by one step reduction is in $R_{A[\alpha_t;t]}$.

For a proof p whose list of free variables is x we use the notation $p[x; x]$ where the second x indicates the occurrences of vector x inside of some "( )", and the first x indicates the other occurrences (i.e., the occurrences used as second order variables). For any proof $p[x, x]$, $p[q, t]$ is obtained from $p[x, x]$ by substituting $q(s, t)$ on the open premisses $x(s, x)$ of p, and substituting t on all occurrences of x in p.

LEMMA 1. If $p[\eta, t]$ is a proof of type $A[\eta, t]$ for some (vector of) terms t and for some (vector of) reducible proofs η of t, i.e., any $\eta \in \alpha_t$, and if p is in $R_{A[\eta;t]}$, then p is strongly normalizable. Moreover, the above R_t satisties the reducibility conditions C1–C3. (Note that the strong normalizability is one of these conditions.) Hence R_t is one example of α_t.

Proof is carried out by induction on the construction of the reducibility predicate $R_{A[\alpha_t/x;t/x]}$. The usual proof for the traditional logical language can be applied with a slight modification (with the above notation for substitution $p[q; t]$ of $p[x; x]$).

Case 1. In particular, if $A[x; x]$ is of the form $y(z)$ where y and z come from the vector x, then p is composed only of the premiss $y(z)$. Here note that some of variables z may be y itsely in our language. Then for any $\eta \in \alpha_t$, $p[\eta, t] = \eta(t) \in \alpha_t(t) = R_{A[\alpha_t/x;t/x]}$. Since $\alpha_t(t)$ satisfies the reducibility conditions by definition, $R_{A[\alpha_t/x;t/x]}$

also satisfies them. In particular, $p[\eta, t]$ is strongly normalizable.

Case 2. If $p[y; y]$ is of type $\forall x\, A[x, y]$ and is not neutral. Let s be a vector of terms whose arities match those of y, and let $\gamma \in \alpha_s$. $p[\gamma; s]$ is in $R_{\forall x A[\alpha_s; s]}$. Then, by the definition of R, for any t (with the same arity as x) and for any $\eta \in \alpha_t$, $q[\eta, \gamma; t, s] \in R_{A[\alpha_t, \alpha_s; t, s]}$, where q is the subproof of p obtained by deleting the last inference rule ($\forall$-introduction). Then, by the induction hypothesis, $q[\eta, \gamma; t, s]$ is strongly normalizable and $R_{A[\alpha_t, \alpha_s; t, s]}$ satisfies the reducibility conditions. The claim follows from these facts.

The other cases are the same as the usual proof (in Girard[2] and in Jouannaud-Okada[3]).

LEMMA 2. If $p[\eta, t]$ is a proof of $A[t, t]$ for any (vector of) terms t and for any (vector of) reducible proofs η of t, i.e., any $\eta \in R_t$, then $p[\eta, t]$ is in $R_{A[\eta; t]}$.

The proof is by the induction on the length of proof, as the case of the traditional logics (cf. Girard[2], Jouannaud-Okada[3]).

Then from these two Lemmas, we have immediately

THEOREM 1. Our System of Self-Referential Predicates (SR) is strongly normalizable.

The well-known Church-Rosser property of the intuitionistic second order fragment with the $\rightarrow$ (implication) and $\forall$ (universal quantifier) is preserved in our system SR.

THEOREM 2. The above fragment of system SR has the Church-Rosser property.

Proof is carried out by checking all critical pairs, as usual, (cf. (2)).

5. REFERENCES

(1) J. Y. Girard, Interpretation fonctionnelle et elimination des coupures dans l'arithmetique d'ordre superieur, These de doctorat d'etat, 1972, Universite Paris VII. Also in Proceedings of the Second Scandinavian

Logic Symposium, 1973, North Holland.
(2) J. Y. Girard, Y. Lafont and P. Taylor. Proofs and Types, Oxford University Press.
(3) J. P. Jouannaud-M. Okada, A Computation model for executable hogher order algebraic specification language., Logic in Computer Science 6, 1991, IEEE.
(4) M. Okada, Logic with self-referential predicates, to appear.
(5) M. Okada, Mobile Linear Logic for a framework of asynchronous and syncronous communication processes, to appear. Also in The Lecture Notes on the Linear Logic and its Application, The Software Science Society of Japan 1993, June.
(6) D. Prawitz, Natural Deduction, 1963, Stockholm. See also, G. Gentzen, Untersuchungen über das logische Schliessen, Collected works, Horth Holland, 1969.
(7) B. Russell, Principles of Mathematics, 1903.
(8) B. Russell and A. N. Whitehead, Principia Mathematica, 1910–1913.
(9) G. Takeuti, Proof theory, 1985, North Holland.