人工知能学会研究会資料
SIG-SWO-A1101-11

集合論とOWL Full

Set Theories and OWL Full

小出誠二 [1*]　　武田英明 [2]
Seiji Koide[1]　　Hideaki Takeda[2]

[1] 情報・システム研究機構新領域融合研究センター
[1] Transdisciplinary Research Integration Center
[2] 国立情報学研究所
[2] National Institute of Informatics

Abstract: Set theories are the theoretical foundation of Semantic Web languages, RDF and OWL, as they are the foundation of mathematical theories. The W3C document of RDF Semantics mentions Zermelo-Fraenkel (ZF) Set Theory, and the W3C document of OWL Semantics invokes the comprehension principle in order to materialize RDF entities in OWL. However, this invocation arose from misunderstandings of set theories, and what is worse, it caused misdirected criticisms against RDF and OWL Full under the pretense that 'comprehension principle allows the paradox to invade upon systems'. Aiming to rescue RDF and OWL Full theory from such theoretical disorder, this paper firstly reviews the history of comprehension principle in set theories, and gives an overview of set theories of Cantor, ZF, and additionally KIF 3.0. Then, the theoretical foundation of OWL Full is introduced with the discussion of Russell's Ramified Type Theory.

1　はじめに

バートランド・ラッセルは1902年にゴットロープ・フレーゲへの手紙において，フレーゲの著わした「概念記法」(Begriffsschrift) にある命題関数 (propositional function)[1]を $\neg x(x)$ とすると矛盾を導くことを指摘した．フレーゲはわずか六日後に手紙を返し，関数 $f(x)$ は引数としてオブジェクトを取らなければならず，関数それ自身を取ることすなわち $f(f)$ は正しくないと答えたが ([1],pp.15)，それが今日ラッセルのパラドックスと呼ばれるものである．

ラッセルのパラドックスは大きく分けて二つの方法で解決された．すなわち，一つはツェルメロによって矛盾を導く無制限の包括原理 (comprehension principle) をより制約の強い原理に変更する方法であり，もう一つはラッセルによるタイプ理論の導入である．今日ではフォン-ノイマン・ベルナイス・ゲーデル (NBG) の集合論と並んで，ツェルメロ・フランケル (ZF) の集合論が数学の基礎としての地位を獲得しているが，ZF およびカントールの集合論では空集合のみを出発点に，すべての集合の要素は集合のみで構成されるため，そこに自然数の概念はあってもオントロジーとして必須のオブジェクト概念はない．

Hayes らによる RDF 意味論 [2] ではクラス[2]がメンバーシップループを有していてもパラドックスはないことの傍証として，ZF 集合論に言及するが，それ自体は間違いではないにせよ，RDF 意味論との関係は明確にされておらず，しかも後述するようにオントロジーとして世界の事物を表現するためには，ZF よりもラッセルらによるタイプ理論のほうがはるかに適切である．

一方，それ自身は集合の要素ではあっても集合とはなりえない個物から出発し，有限集合のみを考慮すれば，ZF 集合論と同様にラッセルのパラドックスは成立しない．実際それが NBG を基礎とする KIF[3] の集合論において提案された方式であり，オントロジー記述の基本原理としてより好ましい方式と考える．

W3C 勧告の「OWL 意味論および抽象構文」第 5 章「RDF-Compatible Model-Theoretic Semantics」[4] において，シーケンス記述における各構成要素がエンティティとして存在すること，owl:distinctMembers，owl:complementOf，owl:unionOf，owl:intersectionOf，owl:oneOf のサブジェクトが存在すること，owl:onProperty，owl:allValuesFrom，owl:someValuesFrom，owl:hasValue，owl:minCardinality，owl:maxCardinality，owl:cardinality のサブジェクトが存在することが要請され，それを包

*連絡先：国立情報学研究所
〒 101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: koide@nii.ac.jp

[1]フレーゲ自身は命題関数という言葉は用いていない．

[2]集合論におけるクラスとオントロジーにおけるクラスは異なるものである．第 4 節参照．

括原理と呼んで OWL 意味論における公理としている．しかし，RDF 意味論を踏まえた OWL すなわち OWL Full においては，これらトリプルの構成要素としてのサブジェクトの存在は，RDF 意味論における伴意ルールを用いれば自動的に伴意され，集合論における包括原理などという大げさな道具立てを導入する必要はない．しかも集合論の歴史はラッセルのパラドックスを解決する歴史であったにもかかわらず，RDF 意味論はラッセルのパラドックスを招くという根拠のない攻撃がコミュニティの一部から行われた．

本論文では，はじめにフレーゲによる無制限の包括原理とラッセルのパラドックスを紹介し，その数学的解決であるツェルメロの分出原理 (separation, aussonderung) と ZF 集合論について概観し，オントロジー的解決である KIF3.0 の集合論を紹介する．第 3 節では W3C の勧告におけるこれら集合論への言及について述べ，第 4 節にてラッセルの分岐タイプ理論 (Ramified Type Theory) を紹介し，第 5 節にて OWL Full との親和性および RDF 意味論と OWL 意味論は矛盾無く統合し得ることを述べる．最後にそれまでの議論を踏まえてオントロジーメタモデリングのための基準についても触れる．

2 ラッセルのパラドックスと集合論

ラッセルのパラドックスとは，集合論におけるクラス概念の自己言及によりシステムに惹起される矛盾である．本節では以下にラッセルのパラドックスの影響を歴史的に概観する．

2.1 包括原理とラッセルのパラドックス

カントールによって見出された包括原理とは直観的には次のように説明される．

> 「ある性質を共有する数学的オブジェクトは集合を形成し，それらオブジェクトはその集合のメンバーとなる」(Doets [5], pp.3)

現代の記法では，無制限の包括原理は次のように記述される．[1] 以下において，x や α はオブジェクトあるいは集合である．
(包括原理) 開かれた良形式な公式 (open well-formed formula) $\phi(x)$ について，

$$\exists\alpha\forall x[(x \in \alpha) \;\Leftrightarrow\; \phi(x)] \tag{1}$$

ここで α は $\phi(x)$ において自由ではない．
すなわち，包括原理は任意の公式を満足する要素の集合が定義可能であることを主張している．

ここで $\forall x[(x \in \alpha) \Leftrightarrow \phi(x)]$ を $\alpha = \{x \mid \phi(x)\}$ と簡潔に表記すれば $(\alpha = \{x \mid \phi(x)\} \Leftrightarrow \forall x[(x \in \alpha) \Leftrightarrow \phi(x)])$，包括原理は $\exists\alpha[\alpha = \{x \mid \phi(x)\}]$ と表記され，抽象の法則とも内包性公理とも呼ばれる．

この無制限の包括原理は上記表記を見れば，集合の内包的定義を与えるものとして，一見正しそうに見える．しかしこの $\phi(x)$ にそれ自体は合理的に見える $(x \notin x)$ を採用すると (つまり自分自身をメンバーとする集合を除いた集合概念)，x が α となる場合には $(\alpha \in \alpha) \Leftrightarrow (\alpha \notin \alpha)$ となって矛盾が導かれる．個別には一見合理的に見える二つの条件から矛盾が導かれることでパラドックスと呼ばれる．このラッセルのパラドックスは，数あるパラドックスの中でも最も純粋で簡潔な形でカントールに始まる素朴集合論に含まれるパラドックスを指摘し，その後の公理的集合論の発展のきっかけとなった．

ツェルメロは，パラドックスを避けるためにこの無制限の包括原理に代えて分出原理と呼ばれる次の公式を用いた．
(分出原理) 開かれた良形式な公式 $\phi(x)$ について

$$\forall z\exists\alpha\forall x[(x \in \alpha) \Leftrightarrow (x \in z) \wedge \phi(x)] \tag{2}$$

ここで α は $\phi(x)$ 中に出現しない
分出原理では，たとえ $\phi(x)$ に $(x \notin x)$ を採用しても，$\alpha = \{x \mid (x \in z) \wedge (x \notin x)\}$ について，$(\alpha \in z) \wedge (\alpha \notin \alpha)$ を満足しないために $(\alpha \in \alpha)$ となる α は排除され，$(\alpha \notin \alpha)$ となる α であれば充足する $(\alpha \in z)$ と充足しない $(\alpha \notin z)$ があり得て，ただちにはパラドックスとならない．ただし，ここで分出される α や分出する z とは一体何かが問題となる[3]．すなわち，一旦はパラドックスが回避されたものの，結局はそのような抽象的集合の実在性についての問題が継続されていったように思われる．

2.2 超限順序数

ZF 集合論の目的は，数論に確固とした公理論的基礎を与えることであった．前述したように，我々はオントロジー記述にはラッセルのタイプ理論のほうが適切であると考えるが，後述するように，ZF 集合論が Hayes らの RDF 意味論 [2] において rdfs:Resource のメンバーシップループの根拠として引用されているため，ここで ZF 集合論の特徴を概観する．

ZF 集合論ではカントールと同様にその構成要素として，論理結合以外では集合とメンバーシップのための表示 $\in$ のみを考える．(Doets [5], pp.5) 最初にオブジェクトとして何はなくても確実に存在するのは空集合だけである．

[3] 分出原理ではある集合を構成するにはその要素がすでに別の集合あるいはクラスの要素であることが要請されている．

(空集合) $\emptyset = \{x \in a \mid x \neq x\}$
以下のような累積的 (cumulative) な集合 V_α を考える．(Doets [5], pp.3)
(cumulative hierarchy) $V_{\alpha+1} = V_\alpha \cup \wp(V_\alpha)$
ここで，$\wp(A) = \{x \mid x \subseteq A\}$ は冪集合すなわち集合 A のすべての部分集合の集合を表す．すると空集合から出発して，以下のように次々と累積的集合を生成することができる[4]．

$$\begin{aligned} V_0 &= \emptyset \\ V_1 &= V_0 \cup \wp(V_0) = \emptyset \cup \{\emptyset\} = \{\emptyset\} \\ V_2 &= V_1 \cup \wp(V_1) = \{\emptyset\} \cup \{\emptyset, \{\emptyset\}\} = \{\emptyset, \{\emptyset\}\} \\ V_3 &= V_2 \cup \wp(V_2) = \cdots = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \\ & \cdots \end{aligned}$$

詳細は省略するが，空集合に 0 を対応させ，上記 $V_{\alpha+1}$ を自然数の後続関数 $S(n) = n + 1$ と対応させることで，V_n との 1 対 1 対応により自然数のすべてを得ることができる．

このようにして得られた無限集合に対応する順序数を ω と表記する．今，空集合に代えてこの ω に対して同様の操作を適応すれば，$\omega + \omega = \omega \cdot 2$ が得られる．

$$\begin{aligned} V_\omega &= \omega \\ V_{\omega+1} &= V_\omega \cup \wp(V_\omega) = \omega \cup \{\omega\} = \{\omega\} \\ V_{\omega+2} &= V_{\omega+1} \cup \wp(V_{\omega+1}) = \cdots = \{\omega, \{\omega\}\} \\ V_{\omega+3} &= V_{\omega+2} \cup \wp(V_{\omega+2}) = \cdots = \{\omega, \{\omega\}, \{\omega, \{\omega\}\}\} \\ & \cdots \end{aligned}$$

ただしこれらの ω を含む数に用いられる $+$ および $\cdot$ は通常の数に対するそれとは異なり，交換則が成り立たないことを注意しておく．

かくして，$\omega{\cdot}2$ の次に，$\omega{\cdot}3$，...，$\omega{\cdot}\omega$，...，$\omega{\cdot}\omega{\cdot}\omega$，...，$\omega^\omega$ と，次々に数学者は「大きな」集合の概念を得る．このような ω を含む数は自然数における無限大の概念とは異なり，超限順序数 (transfinite ordinal number) と呼ばれる．ここで累積的集合 V の重要な定理を挙げる．詳細はクワインその他 [6, 7] を参照されたい．

$$x \in V \Rightarrow (x \notin x)$$
$$x \in y \in V \Rightarrow x \subset y$$

$V_\alpha \in V_{\alpha+1}$ かつ $V_\alpha \subset V_{\alpha+1}$ であることからも明らかなように，どんな「大きな」集合 V_α から出発しても $V_0 \in V_1 \in V_2 \in \cdots \in V_{\alpha-1} \in V_\alpha$ となって，メンバーシップの降下は必ず最後は空集合で停止する．これを正則性 (regularity) あるいは基礎の公理 (foundation axiom) と呼ぶ．

[4]ZF では無限公理がこれを可能にする．

カントールと同様な宇宙 (universe) を対象としながらも，無制限の包括原理に代えて分出原理を用いることでツェルメロはラッセルのパラドックスを回避した．しかし今まで見たように，数学者の考える宇宙は集合あるいは数しか含まず，しかもその大きさは文字通り限界を超えている．オントロジー的立場から集合を考える場合には，数のみならず一般の事物を要素として許容しなければならず，たかだか自然数の有限の大きさで十分である．次にそのような集合論として KIF3.0 の集合論を概観する．

2.3 KIF の集合論

Knowledge Interchange Format (KIF) 3.0[3] ではその第 7 章において，ラッセルのパラドックスを踏まえて，NBG 集合論に基づく KIF 固有の集合論が展開されている．そこでは最終的に包括原理や分出原理に代えて，次式が公理とされている．

$$x \in \{\nu \mid \varphi(\nu)\} \Leftrightarrow bounded(x) \wedge \varphi_{\nu/x} \qquad (3)$$

ここで $\varphi_{\nu/x}$ は，$\varphi(\nu)$ 中のすべての ν に関する自由な出現を項 x で置き換えたものである．KIF3.0 の集合論では集合の要素に個物を許容し，すべてのオブジェクトは個物か集合のどちらかである．一方それとは独立に，すべてのオブジェクトは bounded か unbounded である．有限な集合は bounded である．unbounded な個物とは何であるか興味があるが，それについては何ら触れられていない．

> 「KIF では，個物 (individuals) と集合の間に基本的な区別がある．個物は集合ではないオブジェクトである．bounded と unbounded の間にも区別があるが，この区別は個物および集合とは直交している．bounded 個物と unbounded 個物があり，bounded 集合と unbounded 集合がある．エンティティのこの様々なタイプの間の基礎的な関係がメンバーシップである．集合はメンバーを持つが，個物は持つことができない．bounded オブジェクトは集合のメンバーになることができるが，unbounded オブジェクトはなることができない．(この条件により集合論の伝統的なパラドックスを回避することが可能になる．)」(KIF3.0[3] 集合論)

ZF における分出原理 (2) 式中の $(x \in z)$ は結局のところすでにその存在が認められた集合あるいはクラスの部分集合の記述のみを許容しているが [7]，KIF ではそれに代わって，$bounded(x)$ を用いることで，個物を許容して無限の集合を制約することで分出原理と同様

にラッセルのパラドックスを回避している[5]．オントロジーを前提にセマンティックウェブ理論の定式化を行う場合，無限集合の概念を導入しなければラッセルのパラドックスを気にせずに，(3) 式を満たすような集合のみを考えればよい．

3 RDF 意味論と OWL 意味論

3.1 RDF 意味論における集合論

W3C 勧告である RDF 意味論 [2] では，次のような記述がある．

> 「クラスが RDFS で導入されるとき，それら複数のクラスがそれら自身を含んでもよい．そのような「メンバーシップループ」は，基礎の公理であるところの，メンバーシップについて無限の降下の連鎖を禁止するという，標準 (Zermelo-Fraenkel) 集合論理の公理に違反するように見えるかもしれないが，ここで与えられる意味論的モデルでは，オブジェクトとしてのプロパティとクラスはそれらの外延 (extensions) - そのプロパティに充足するオブジェクト・値ペアの集合，もしくはそのクラスの「中」に含まれるような事物 - とは区別される．それによって，基礎の公理に違反することなくあるプロパティとクラスの外延がプロパティやクラス自身を含むことができる．特に，クラス外延写像を用いて，クラスがそれ自身を含むことができる．例えば，ある「ユニバーサル」クラス（の外延）がメンバーとしてそのクラス自身を含むことは全く問題なく，これはあるクラス階層の頂上でしばしば採用される約束事である．」(RDF 意味論 [2])

ユニバーサルクラスとは rdfs:Resource のことで，“「ユニバーサル」クラス（の外延）がメンバーとしてそのクラス自身を含む” とは次式で表現される．

$$rdfs:Resource^{\mathcal{I}} \in CEXT^{\mathcal{I}}(rdfs:Resource^{\mathcal{I}}) \quad (4)$$

$CEXT^{\mathcal{I}}(rdfs:Resource^{\mathcal{I}})$ は rdfs:Resource が表示するクラスのクラス外延であり．それはこの場合、RDF ユニバース全体を表示する．“オブジェクトとしての [...] クラスはそれらの外延 [...] とは区別される” とは $rdfs:Resource^{\mathcal{I}}$ は $CEXT^{\mathcal{I}}(rdfs:Resource^{\mathcal{I}})$ とは区別されるという意味である．上記 RDF 意味論の文章は，$rdfs:Resource^{\mathcal{I}} \in rdfs:Resource^{\mathcal{I}}$ ではないから，基礎の公理には違反しないという意味である．

確かに ZF ではどこにもメンバーシップループはないから基礎の公理に違反しない．しかし rdfs:Resource は，クラスとその外延を区別するとは言え，rdfs:Class のメンバーシップループに起因するメンバーシップループがある[6]．したがって rdfs:Resource や rdfs:Class においてメンバーシップループによる無限の降下を防ぐためには，後述するように，オーダ (階) の導入が必要である．

RDF 意味論は集合概念を基礎に持ちつつもそのクラス概念は集合とは異なるものである．すなわち，

> 「明示的な外延写像の使用はまた二つのプロパティが異なるエンティティでありながら，厳密に同じ値を持つことを可能にするし，二つのクラスが同じインスタンスを含むことを可能にする．これは RDFS のクラスは単なる集合以上のものと考えられるということを意味している．それらは単なる外延的一致を超えた，ロバストな同一性概念である “分類 (classification) ” あるいは “概念 (concept) ” と考えられる．」(RDF 意味論 [2])

あるいは同じことであるが，rdfs:subClassOf によるクラス包摂関係の定義は，それらのクラス外延の集合としての包含関係を意味するが，その逆は真ではない[7]．

$$\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in EXT^{\mathcal{I}}(rdfs:subClassOf^{\mathcal{I}}) \Rightarrow x^{\mathcal{I}} \in \boldsymbol{C}^{\mathcal{I}} \wedge y^{\mathcal{I}} \in \boldsymbol{C}^{\mathcal{I}} \wedge CEXT^{\mathcal{I}}(x^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(y^{\mathcal{I}}) \quad (5)$$

ここで $EXT^{\mathcal{I}}$ はプロパティ外延，$\boldsymbol{C}^{\mathcal{I}} = CEXT^{\mathcal{I}}(rdfs:Class^{\mathcal{I}})$ は論議の領域 (universe of discourse) におけるすべてのクラスの集合である．

この公理でわかるように，RDFS ではあるクラス外延が他のクラス外延の部分集合だからといって，そのクラスが他のクラスのサブクラスであるとは言えないし，あるクラス外延と他のクラス外延が集合として同じだからといって，両者が RDFS においてクラスとして等しいとは言えない．

3.2 OWL 意味論における集合論

一方対照的に，OWL におけるクラス概念は集合そのものである．「OWL 意味論および抽象構文」の第 5

[5]数学的にはこの制約は強すぎる．無限の操作の繰り返しで生成される無限集合まで拡張してもラッセルのパラドックスは回避されるが，オントロジーとしては有限で十分に用は足りる．

[6](4) 式は rdfs:Resource のクラスかつサブクラスである rdfs:Class がメンバーシップループを持つことによって成立する．詳しくは [11, 12] 参照．

[7]`http://www.w3.org/TR/rdf-mt#rdfssemcond7` 参照．

章 [4] では rdfs:subClassOf の解釈について以下のようになっている[8].

$$\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in EXT^{\mathcal{I}}(rdfs\!:\!subClassOf^{\mathcal{I}}) \Leftrightarrow x^{\mathcal{I}} \in \boldsymbol{OC}^{\mathcal{I}} \wedge y^{\mathcal{I}} \in \boldsymbol{OC}^{\mathcal{I}} \wedge CEXT^{\mathcal{I}}(x^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(y^{\mathcal{I}}) \quad (6)$$

ここで $\boldsymbol{OC}^{\mathcal{I}} = CEXT^{\mathcal{I}}(owl : Class^{\mathcal{I}})$ は OWL ユニバースにおけるすべてのクラスの集合である.

すなわち，OWL 意味論では RDF 意味論と異なって，もしあるクラス外延が他のクラス外延の部分集合であれば，先のクラスは後のクラスのサブクラスとしてよいし，あるクラス外延が他のクラス外延と等しければ両者はクラスとして等価としてよい．しかしこの意味論は正確に言えば，OWL DL の意味論である．Description Logic ではもともと概念の共通や合併はあっても rdfs:subClassOf に相当する構成がなく [8]，RDFS 語彙の OWL 解釈を決定するときに Description Logic に合わせて，そのようにしたものであろう．しかし，それは RDF 意味論に述べられた「RDFS のクラスは単なる集合以上のもの」という特徴を消し去ってしまう．RDFS の特徴を生かした OWL Full 意味論はこれとは別に定式化されなければならない [12].

3.3 包括原理とは何か？

「OWL 意味論および抽象構文」の第 5 章 [4] では，owl:oneOf や owl:intersectionOf や owl:unionOf の値であるシーケンス記述において，各要素は明示的に rdf:List で一つにまとめられるが，それら要素がエンティティとして存在することを公理として導入するときに包括原理という名前が用いられている ([4],Comprehension conditions (principles))．しかし，もともと RDF グラフ記述には意味論から見れば冗長な部分があって，意味論的には同一なものを異なるグラフで表現することができる．たとえば，ワインオントロジーにおける vin:RedWine の意味は以下のように異なるグラフで表現できる.

```
vin:RedWine rdf:type owl:Class .
vin:RedWine owl:intersectionOf _:cel1 .
_:cel1 rdf:type rdf:List .
_:cel1 rdf:first vin:Wine .
_:cel1 rdf:rest _:cel2 .
_:cel2 rdf:type rdf:List .
_:cel2 rdf:first _:gx3  .
_:cel2 rdf:rest rdf:nil .
_:gx3 rdf:type owl:Restriction .
_:gx3 owl:onProperty vin:hasColor .
_:gx3 owl:hasValue vin:Red .

vin:RedWine rdf:type owl:Class .
vin:RedWine owl:intersectionOf vin:Wine .
vin:RedWine owl:intersectionOf _:gx4 .
_:gx4 rdf:type owl:Restriction .
_:gx4 owl:onProperty vin:hasColor .
_:gx4 owl:hasValue vin:Red .
```

RDF グラフではあるサブジェクトノードのあるプロパティの値となるコレクションは単にサブジェクトから放射される複数のアークとノードによって表現することができる．私見では RDF におけるその他のコンテナプロパティrdf:Seq，rdf:Bag，rdf:Alt にはそれなりの意味があり，利用目的もはっきりしているのに比べて，rdf:List の意味や使用法は定かではなく，オントロジー記述に rdf:List 表現を用いなければならない必然性はない．rdf:List 表現を用いなければ明示的なシーケンス記述は消えうせて，その結果としてプロパティ値記述における上記混乱も解消するものと考える.

また，owl:distinctMembers，owl:unionOf，owl:oneOf，owl:intersectionOf，owl:complementOf，owl:onProperty，owl:allValuesFrom，owl:someValuesFrom，owl:hasValue，owl:minCardinality，owl:maxCardinality，owl:cardinality のサブジェクトが存在することも，同じく包括原理という名前で要請されている ([4],Comprehension conditions (principles))．しかし，これも包括原理という大げさな名前を持ち出さずとも，RDF 意味論における伴意ルール **rdfs4a**[9]を適用すればよい.

これまで概観してきたように，集合論における (無制限の) 包括原理とは (1) 式のことである．一方，OWL 意味論に言う包括原理とは，伴意ルールに現れる個物の存在を要請する原理のようである[10]．このような違いがどこから生まれたか定かではないが，プログラム言語においては *list comprehension* というものがある[11]．これは set-builder の機能を有するプログラム言語において，与えられた公式を満足するリストを自動的に生成することを言うが，別の集合から要素を抽出するようになっており，この機能は正確には分出原理と呼ばれるべきである．集合論における包括原理 (1) 式は，個物の存在についての議論なのではなく，集合あるいはクラスの存在についての議論であり，集合の要素が問題とされているのではない．さらに問題なのは，(無制限の) 包括原理はラッセルのパラドックスを導くということを根拠として，RDF 意味論に対する攻撃がこれまでコミュニティの一方の陣営から行われてきた.

[8] この iff 条件は OWL2 においても同様である．http://www.w3.org/TR/2009/REC-owl2-rdf-based-semantics-20091027/#table-semcond-rdfs を参照のこと

[9] If (uuu aaa xxx .) then (uuu rdf:type rdfs:Resource .)

[10] http://www.w3.org/TR/2009/REC-owl2-rdf-based-semantics-20091027/#Appendix:_Comprehension_Conditions_.28Informative.29 参照のこと.

[11] http://en.wikipedia.org/wiki/List_comprehension

4 分岐タイプ理論とユニバーサルクラス

ラッセル自身はいわゆるラッセルのパラドックスなるものを他のパラドックスと同様に悪循環 (vicious circle) の一種として捉えた．

> 「避けるべきパラドックスを調べると，それらは全てある種の悪循環の結果であるということがわかる」([10],pp.39)

ラッセルのパラドックスを避けるために，以下のような「悪循環の法則」が設定される．

> 「ある集まり (collection) のすべてを含むようなものは何にせよ，その集まりの一つであってはならない．逆に言えば，もしある集まりが全体であって，それがその全体という点でのみ定義可能なメンバーを有するのなら，その集まりはもはや全体ではない」([10],pp.40)

集合論にいうユニバーサルクラスはまさにそのようなものであり，論議の領域全体を表示するクラスである．ちなみに，集合論において集合はクラスでもあるが，集合ではないクラスは本来のクラスと呼ばれる．ZF ではユニバーサルクラスは集合とは見なされない本来のクラスである．NBG では集合は小文字記号で表現され，クラスは大文字記号で表現され，それぞれにほとんど同様な公理が定式化される．

> 「メンバーとなりうるクラスは集合と呼ばれる．その他のクラスは,『本来のボストン』や『真部分』とのアナロジーで，不完全にも『真のクラス』(proper class) とよばれてきた」(クワイン [6], 序論, 大出・藤村訳)

クワインはそれを究極クラス (ultimate classes) と呼ぶことにした．ユニバーサルクラスはその内に無限個のメンバーを含み，構成主義的な集合論ではそれは無限回の集合の生成の極限にあるものであるが，それを「単なる言い回し」とする立場，それを数と同様な意味で「実在する」とする立場など，哲学的議論も含めて様々である．

ラッセルは集合論の語彙を用いてではなく，新たに命題関数 (propositional function) という考え方を導入する．命題関数とは，それに含まれる変数にある値を割り当てると命題となるようなものである．そして，この命題関数についてこの悪循環の法則を満たすものとして分岐タイプ理論 (Ramified Type Theory) を考案した．しかし，その定式化は現在の数学的に厳密な目から見ると曖昧であり，その後多くの研究者がこれを様々に解釈した．ここでは Kamareddine [1] を参考に，分かりやすさを目的に，1 座と 2 座の命題関数のみ言いかえればメンバーシップと 2 項関係のみを考えて，RDF 意味論に従って Tarski 流の表示意味論を加味する．

- $\mathcal{A}$ は個物の語彙であり，$a_1, a_2, \ldots, b_1, b_2, \ldots$ は $\mathcal{A}$ 上を走る個物のシンボルである．$a_i^{\mathcal{I}}$ は a_i の，$b_j^{\mathcal{I}}$ は b_j の表示である．
- $\mathcal{C}$ は 1 座の述語語彙であり，C_i は $\mathcal{C}$ 上を走る単項述語のシンボルである．$C_i^{\mathcal{I}}$ は C_i の表示である．
- $\mathcal{R}$ は 2 座の述語語彙であり，R_i は $\mathcal{R}$ 上を走る 2 項述語のシンボルである．$R_i^{\mathcal{I}}$ は R_i の表示である．
- $\mathcal{V}$ は変数の語彙であり，$x_1, x_2, \ldots, y_1, y_2, \ldots$ は $\mathcal{V}$ 上を走る変数のシンボルである．$x_i^{\mathcal{I}}$ は x_i の表示である．$y_i^{\mathcal{I}}$ は y_i の表示である．

(原子命題) $C_i^{\mathcal{I}}(a_j^{\mathcal{I}})$ と $R_i^{\mathcal{I}}(a_j^{\mathcal{I}}, b_k^{\mathcal{I}})$ は原子命題である．

命題関数は「論理結合と量化」，および「先に生成された命題関数の抽象の法則による抽象化」の二つの方法により，原子命題より生成される．
(命題関数)$\mathcal{P}$ は命題関数の表記，$\boldsymbol{P}$ は $\mathcal{P}$ の各要素 $f(f \in \mathcal{P})$ の表示 $f^{\mathcal{I}}$ の集合である命題関数の集合 $(f^{\mathcal{I}} \in \boldsymbol{P})$ である．$\boldsymbol{P}$ は次のように定義される．

1. もし $j \in \mathcal{A} \cup \mathcal{V}$ および $k \in \mathcal{A} \cup \mathcal{V}$ ならば，$C_i(j) \in \mathcal{P}$ および $R_i(j,k) \in \mathcal{P}$，そして $C_i^{\mathcal{I}}(j^{\mathcal{I}}) \in \boldsymbol{P}$ および $R_i^{\mathcal{I}}(j^{\mathcal{I}}, k^{\mathcal{I}}) \in \boldsymbol{P}$ である．
2. もし $f, g \in \mathcal{P}$ ならば，$f^{\mathcal{I}} \vee g^{\mathcal{I}} \in \boldsymbol{P}$ かつ $\neg f^{\mathcal{I}} \in \boldsymbol{P}$ である．
3. もし $f \in \mathcal{P}$ かつ x が f の自由変数ならば，$\forall x^{\mathcal{I}}[f^{\mathcal{I}}] \in \boldsymbol{P}$ である．
4. もし $j \in \mathcal{A} \cup \mathcal{V} \cup \mathcal{P}$ そして $k \in \mathcal{A} \cup \mathcal{V} \cup \mathcal{P}$ ならば，$z_i^{\mathcal{I}}(j^{\mathcal{I}}) \in \boldsymbol{P}$ かつ $z_j^{\mathcal{I}}(j^{\mathcal{I}}, k^{\mathcal{I}}) \in \boldsymbol{P}$ である．ただし，$z_i \in \mathcal{C} \cup \mathcal{V}$，$z_j \in \mathcal{R} \cup \mathcal{V}$ である．
5. すべての命題関数は上記ルールを用いてのみ作られる．

ルール 1 から 3 までの記述は 1 階述語論理に相当する．ルール 4 によって高階まで拡張される．
(命題) 命題関数の自由変数の個数がゼロの場合，すなわち変数の個数がゼロかすべての変数が量化されたとき，それは命題である．

たとえば，$Obama \in \mathcal{A}$，そして $Human \in \mathcal{C}$ の場合，$Human^{\mathcal{I}}(Obama^{\mathcal{I}})$ や $\forall x^{\mathcal{I}}[Human^{\mathcal{I}}(x^{\mathcal{I}})]$ は命題であり，$x^{\mathcal{I}}(Obama^{\mathcal{I}})$ や $Human^{\mathcal{I}}(x^{\mathcal{I}})$ は命題関数である．

ここまでのところでまだ「悪循環の法則」は設定されていない．たとえば，$\neg z(z)$ はいわゆるラッセル集合[12]に相当する命題関数であるが，それはまだ定義可能である．ここでラッセルはタイプを導入する.「同じタイプ」とは次のように，step by step で定義される．二つのオブジェクト $u^{\mathcal{I}}$ と $v^{\mathcal{I}}$ は，

1. もし二つの個物ならば同じタイプである．
2. 同じタイプの引数をとる基本命題関数[13]は同じタイプである．
3. $u^{\mathcal{I}}$ が命題関数で $v^{\mathcal{I}}$ がその否定ならば同じタイプである．
4. u が $\phi(x)$ または $\psi(y)$ で，$v^{\mathcal{I}}$ が $\phi^{\mathcal{I}}(x^{\mathcal{I}})\vee\psi^{\mathcal{I}}(y^{\mathcal{I}})$(ここで $\phi^{\mathcal{I}}(x^{\mathcal{I}})$ と $\psi^{\mathcal{I}}(y^{\mathcal{I}})$ は基本命題関数) ならば $u^{\mathcal{I}}$ と $v^{\mathcal{I}}$ は同じタイプである．u が $\phi(x_j, x_k)$ または $\psi(y_j, y_k)$ で，$v^{\mathcal{I}}$ が $\phi^{\mathcal{I}}(x_l^{\mathcal{I}}, x_m^{\mathcal{I}}) \vee \psi^{\mathcal{I}}(y_l^{\mathcal{I}}, y_m^{\mathcal{I}})$ (ここで $\phi^{\mathcal{I}}(x_l^{\mathcal{I}}, x_m^{\mathcal{I}})$ と $\psi^{\mathcal{I}}(y_l^{\mathcal{I}}, y_m^{\mathcal{I}})$ は基本命題関数) ならば，$u^{\mathcal{I}}$ と $v^{\mathcal{I}}$ は同じタイプである．
5. $u^{\mathcal{I}}$ が $\forall y^{\mathcal{I}}[\phi^{\mathcal{I}}(x^{\mathcal{I}}, y^{\mathcal{I}})]$ でかつ $v^{\mathcal{I}}$ が $\forall z^{\mathcal{I}}[\phi^{\mathcal{I}}(x^{\mathcal{I}}, z^{\mathcal{I}})]$ (ここで $\phi^{\mathcal{I}}(x^{\mathcal{I}}, y^{\mathcal{I}})$ と $\phi^{\mathcal{I}}(x^{\mathcal{I}}, z^{\mathcal{I}})$ は同じタイプ) ならば同じタイプである．
6. $u^{\mathcal{I}}$ と $v^{\mathcal{I}}$ が基本命題ならば同じタイプである．
7. $u^{\mathcal{I}}$ が命題関数で $v^{\mathcal{I}}$ がその否定なら同じタイプである．
8. もし $u^{\mathcal{I}}$ が $\forall x^{\mathcal{I}}[\phi^{\mathcal{I}}(x^{\mathcal{I}})]$ で $v^{\mathcal{I}}$ が $\forall y^{\mathcal{I}}[\psi^{\mathcal{I}}(y^{\mathcal{I}})]$ (ここで $\phi^{\mathcal{I}}(x^{\mathcal{I}})$ と $\psi^{\mathcal{I}}(y^{\mathcal{I}})$ は同じタイプ) ならば，$u^{\mathcal{I}}$ と $v^{\mathcal{I}}$ は同じタイプである．(以上，[10],*9·131, Vol.1,pp.138)

分岐タイプとは次のように定義される．

1. 0^0 は分岐タイプである．
2. もし t^m が分岐タイプならば，$(t^m)^n$ も分岐タイプである．ここで n や m は自然数で $n > m$ である．
3. すべての分岐タイプは上記ルールを用いてのみ作られる．

以下は分岐タイプの例である．

- 0^0;
- $(0^0)^1$;
- $((0^0)^1, (0^0)^4)^5$;

$((0^0)^1, (0^0)^4)^4$ は分岐タイプではない．

分岐タイプのうち，$n = m + 1$ なるものを記述可能 (predicative) タイプと呼ぶ．n や m をオーダと呼ぶ．たとえば，$Obama \in \mathcal{A}$ であるとき，$Obama^{\mathcal{I}}$ はタイプ 0^0 であり，$Human^{\mathcal{I}}(Obama^{\mathcal{I}})$ はタイプ $(0^0)^1$ である．すると，$rdfs{:}Class^{\mathcal{I}}(Human^{\mathcal{I}})$ はタイプ $((0^0)^1)^2$ である．

ここで RDF 意味論に言うクラスは 1 座の記述可能タイプであるとする．すなわち，$x^{\mathcal{I}} \in CEXT^{\mathcal{I}}(C^{\mathcal{I}})$ は $C^{\mathcal{I}}(x^{\mathcal{I}})$ と同じことであり，かつ $x^{\mathcal{I}}$ のオーダが n ならば，$C^{\mathcal{I}}$ のオーダは $n+1$ とする．すると，たとえば $Obama^{\mathcal{I}}$ のタイプ 0^0 に対して，$Human^{\mathcal{I}}$ はタイプ $(0^0)^1$ である．ユニバーサルクラスの (4) 式において，左辺の $rdfs:Resource^{\mathcal{I}}$ のオーダが n のとき，右辺の $rdfs:Resource^{\mathcal{I}}$ のオーダは $n+1$ である[14]．両者は一見同じオブジェクトを表示するように見えるが，分岐タイプ理論ではオーダの違いにより区別されなければならない．こうして，任意のオーダ n におけるユニバーサルクラスのメンバーシップループの降下は $n \to n-1 \to n-2 \to \ldots$ となって最後には停止する．それでは最後のボトムになる $rdfs:Resource^{\mathcal{I}}$ のオーダはいくつであろうか？$rdfs{:}comment^{\mathcal{I}}$ や $rdfs{:}label^{\mathcal{I}}$ をオーダ 0 とすると[15]，$rdf:Property^{\mathcal{I}}$ はオーダ 1，そのスーパクラスである $rdfs{:}Resource^{\mathcal{I}}$ はオーダ 1，そしてメンバーシップループ (4) 式によってそれ以上と計算される．それでは $rdfs{:}Class^{\mathcal{I}}$ のオーダはいくつであろうか？$rdfs{:}Resource^{\mathcal{I}} \in CEXT^{\mathcal{I}}(rdfs{:}Class^{\mathcal{I}})$ であることからオーダ 2 以上である．

5 RDF 意味論と整合する OWL

RDF 意味論において，OWL 定義ファイル[16]にあるように，

$$\langle owl{:}Class^{\mathcal{I}}, rdfs{:}Class^{\mathcal{I}}\rangle \in EXT^{\mathcal{I}}(rdfs{:}subClassOf^{\mathcal{I}})$$

とすることで，(5) 式より

$$owl{:}Class^{\mathcal{I}} \in \boldsymbol{C}^{\mathcal{I}} \wedge CEXT^{\mathcal{I}}(owl{:}Class^{\mathcal{I}}) \subseteq \boldsymbol{C}^{\mathcal{I}}$$

となって，OWL 意味論 [4] における owl:Class の条件を満足させることができる．そして，RDF 意味論の有効な RDF ユニバース中において，たとえば OWL 定義ファイルにあるように，

$$\langle owl{:}Thing^{\mathcal{I}}, owl{:}Class^{\mathcal{I}}\rangle \in EXT^{\mathcal{I}}(rdf:type^{\mathcal{I}})$$

[12] ラッセルのパラドックスを導く ($x \notin x$)

[13] 基本命題関数とは，その値として基本命題のみをとるような命題関数のことである．基本命題とは，いかなる変数も含まないようなものである．[10],pp.95 参照.

[14] ラッセル自身はユニバーサルタイプを導入していない．

[15] これらのプロパティはインスタンスであり，そのインスタンスは決して定義されない

[16] `http://www.w3.org/2002/07/owl.rdf`

とすることで，RDFS の伴意ルール **rdfs9**[17]により，

$$\langle owl{:}Thing^{\mathcal{I}}, rdfs{:}Class^{\mathcal{I}}\rangle \in EXT^{\mathcal{I}}(rdf{:}type^{\mathcal{I}})$$

と伴意され，さらに **rdfs8**[18]により，

$$\langle owl{:}Thing^{\mathcal{I}}, rdfs{:}Resource^{\mathcal{I}}\rangle \in EXT^{\mathcal{I}}(rdfs{:}subClassOf^{\mathcal{I}})$$

と伴意されて，OWL 意味論 [4] における owl:Thing の条件を満足させることができる．

ところで，RDF 意味論において OWL 定義ファイルを適応しただけでは，$\boldsymbol{OC}^{\mathcal{I}} \subset \boldsymbol{OT}^{\mathcal{I}}$ とはならない[19]．そこで RDF 意味論における公理 $\boldsymbol{C}^{\mathcal{I}} \subset \boldsymbol{R}^{\mathcal{I}}$ と同様に，

$$\langle owl{:}Class^{\mathcal{I}}, owl{:}Thing^{\mathcal{I}}\rangle \in EXT^{\mathcal{I}}(rdfs{:}subClassOf^{\mathcal{I}})$$

という公理を設定する．この公理によって，すべての OWL クラスが OWL ユニバース中に存在するようになる．詳細は [11, 12] を参照されたい．

6 OWL Full のためのメタモデリング基準

これまで見てきたように，メンバーシップの無限の降下を防止するためには，$x^{\mathcal{I}} \in CEXT^{\mathcal{I}}(y^{\mathcal{I}})$ において，$x^{\mathcal{I}}$ のオーダは必ず $y^{\mathcal{I}}$ のオーダよりも小さくなければならず，メンバーシップを表す $\in$ において両辺のオーダが逆順になったり同一オーダになるような，メンバーシップのループがあってはならない．しかし，実際にはそのようなオントロジーや predicative ではない[20]オントロジーなどが公開されている．我々はすでに Common Lisp Object System (CLOS) を用いた OWL Full 処理系を開発し [13, 14]，公開しているが[21]，そこでは CLOS の意味論に合致した OWL Full のためのメタモデリング基準を提案している．本節では改めて，本稿にてこれまで見てきたようなラッセルらの分岐タイプ理論に基づいた，OWL Full のためのメタモデリング基準を提案する．

1. rdfs:Resource や owl:Thing のインスタンスでかつ rdfs:Class や owl:Class のインスタンスではないオブジェクトは個物であり，そのオーダは 0 である．

2. rdfs:Resource や owl:Thing のインスタンスでかつ rdfs:Class や owl:Class のインスタンスであるが，rdfs:Class や owl:Class のサブクラスではないオブジェクトは，クラスであるがそのオーダは 1 である．

3. rdfs:Class のオーダは 2 以上，rdfs:Resource のオーダは 1 以上である．

4. rdfs:Resource や owl:Thing のインスタンスでかつ rdfs:Class や owl:Class のインスタンスであり，しかも rdfs:Class や owl:Class のサブクラスでもあるオブジェクトは，クラスでありそのオーダは 2 以上である．

5. rdfs:subClassOf の引数のオーダは同じでなければならない．

6. rdf:type は predicative でなければならない．すなわち，rdf:type におけるメンバーをオーダ n とすると，クラスはオーダ $n+1$ でなければならない．

7. 基礎の公理を満足しなければならない．すなわち，原則として直接間接にメンバーシップループを作ってはならない．メンバーシップループが一見見えるときでも，上記基準に照らして，オーダを解釈してメンバーシップループの降下は最後に停止しなければならない．

7 むすび

本論文では RDF 意味論および OWL 意味論における集合論に焦点をあて，最初にカントールによる (無制限の) 包括原理，ツェルメロの分出原理を紹介し，ラッセルのパラドックスをめぐる集合論の歴史的経緯について概観した．次に，これらの集合論よりもオントロジーの議論により適切な集合論として，KIF 3.0 における集合論を紹介し，無限集合や全体集合 (ユニバーサルクラス) を導入しなければ，ラッセルのパラドックスは成立しないことを述べた．

しかし，RDF 意味論にはユニバーサルクラス (rdfs:Resource) やそのメタクラス (rdfs:Class) がある．RDF 意味論はクラスオブジェクトとそのクラス外延を区別することで，ツェルメロ・フランケル集合論における基礎の公理に違反しないと述べているが，本論文ではラッセルの分岐タイプ理論を紹介して，これを RDF のメンバーシップに当てはめて，オーダを考慮することによりメンバーシップループを逃れて基礎の公理に違反しなくなることを述べた．

RDF 意味論と OWL DL 意味論にはクラス包摂関係において重要な違いがあることを指摘した．また，OWL1 における *comprehension condition* の導入は OWL Full

[17] `http://www.w3.org/TR/rdf-mt/#rulerdfs9`
[18] `http://www.w3.org/TR/rdf-mt/#rulerdfs8`
[19] ここで $\boldsymbol{OT}^{\mathcal{I}} = CEXT^{\mathcal{I}}(owl:Thing^{\mathcal{I}})$ は OWL ユニバースにおけるすべてのエンティティの集合である．
[20] $x^{\mathcal{I}} \in CEXT^{\mathcal{I}}(y^{\mathcal{I}})$ において $y^{\mathcal{I}}$ のオーダが $x^{\mathcal{I}}$ のオーダよりも 2 以上大きな場合．
[21] `http://www-kasm.nii.ac.jp/~koide/SWCLOS2.htm`

では無用であることは W3C 勧告にも述べられているが，それでもこの言葉を用いることは集合論の観点から正しくないことを指摘した．

最後にこれまでの議論を踏まえて，ラッセルのパラドックスを導かない，ラッセル流の観点で言えば悪循環を導入しないためのオントロジー構成基準を「OWL Full のためのメタモデリング基準」という言葉で提案した．しかしこれといわゆる tractability の問題とは別問題であることを注意しておく．悪循環はもちろん untractable であるが，OWL には rdfs:subClassOf 以外にも様々なオントロジーモデリング機能があるため，ここで提案した記述を満たしても tractable であるとは限らない．

我々の開発した OWL Full 処理系 [13, 14] では，CLOS クリーンな観点から OWL Full メタモデリング基準を提案したが，さらにこの基準を緩めて，基礎の公理に違反することなくどこまでメタモデリングが許されるのかという問題，また untractable なメタモデリングのほとんどは工学的にナンセンスであるが，基礎の公理に違反しつつかつ実用上意味のある問題が存在するのかどうかどういう問題は興味ある問題である．

参考文献

[1] Kamareddine, F., T. Laan, and R. Nederpelt: *A Modern Perspective on Type Theory*. Kluwer Academic Publishers, (2004)

[2] Hayes, P., and B. McBride: RDF Semantics. W3C Recommendation, Feb. (2004) `http://www.w3.org/TR/rdf-mt/`.

[3] Genesereth, M.R., and R.E. Fikes: Knowledge interchange format version 3.0 reference manual, (1994) `http://logic.stanford.edu/kif/Hypertext/kif-manual.html`

[4] Patel-Schneider, P.F., P. Hayes, and I. Horrocks: OWL Web Ontology Language Semantics and Abstract Syntax section 5. rdf-compatible model-theoretic semantics. W3C Recommendation, `http://www.w3.org/TR/owl-semantics/rdfs.html`, February (2004)

[5] Doets，H.C.: Zermelo-Fraenkel Set Theory. Lecture, April (2002) `http://staff.science.uva.nl/~vervoort/AST/ast.pdf`.

[6] Quine, W.O.: *Set Theory and its Logic*. Harvard, (1963)

[7] Potter, M. *Set Theory and its Philosophy*. Oxford, (2004)

[8] Borgida, A. et al.: *The Description Logic Handbook*. Cambridge, (2003)

[9] Boolos, G.: The Iterative Conception of Set. *The Journal of Philosophy*, Vol. 68, No. 8, Philosophy of Logic and Mathematics, pp. 215–231, April (1971)

[10] Whitehead, A.N. and Russell, B.: *Principia Mathematica*, volume 1. Merchant Books, (1910)

[11] Koide, S., Takeda, H.: COMMON LANUGAGES FOR SEMANTIC WWW – Beyond RDF and OWL –, 5th Int. Conf. Evaluation of Novel Approaches to Software Engineering (ENASE2010), Springer, (2010)

[12] Koide, S.: Theory and Implementation of Object Oriented Semantic Web Language Dr. thesis, Dept. Informatics School of Multidisciplinary Sciences, The Graduate Univ. for Advanced Studies (SOKENDAI), (2010)

[13] Koide, S., Takeda, H.: OWL-Full Reasoning from an Object Oriented Perspective, Asian Semantic Web Conf. (ASWC2006), PP.263–277, Springer (2006)

[14] 小出, 武田: OWL 意味論に基づく CLOS オブジェクト指向プログラミング, コンピュータソフトウェア, Vol. 28, No. 2, pp. 236–260 (2011)