OWLの推論とその計算量

兼岩 憲

本解説では， Web オントロジー言語 OWL における推論とその計算量について解説する．OWL には 3 つの下位言語 (OWL Full/OWL DL/OWL Lite) が提案されており，特に OWL DL には，Description Logic (DL) に基づく理論的な基盤が整備されている．それにより，OWL DL で記述したオントロジーに対する論理的推論と決定可能性が保証されることを説明する．

This survey paper describes OWL reasoning and its complexity. The OWL Web Ontology Language has three sublangauges (OWL Full/OWL DL/OWL Lite). In particular, OWL DL is based on the theoretical foundations of Description Logic (DL). We indicate that logical reasoning on ontologies and decidability are guaranteed by the expressive power of the OWL DL.

1 はじめに

セマンティック Web の実現には，Web 情報のオントロジーを構築して計算機がその意味構造を処理する必要がある．そうした目的のために，W3C の Web オントロジー WG は Web オントロジー言語 OWL [16][4][14] の仕様を提案している．実際，オントロジー記述言語の設計にはオントロジーをどのように記述させるかが問題となるが，それに加えて重要かつ困難なのはオントロジーからの推論メカニズムを実現することである．それは形式的に安全性が保証された推論であるとともに，計算機で実装できるものでなければならない．推論の停止性が証明されなければ，ソフトウェアシステムとして望ましくない．

OWL には，ユーザーの用途を想定して，3 つの下位言語 OWL Full/DL/Lite が用意されている．OWL Lite は容易な実装を考えて簡単なクラス表現しか持たないが，OWL DL はクラスとプロパティに複雑な構文を許している．この 2 つの言語での推論は，表現力に一定の制限があるので決定可能である．一方 OWL Full は，RDF による自由な記述が許される代わりに決定可能性が保証されていない．しかしどのようにして，OWL DL/Lite の決定可能性が保証されているのだだろうか？言い換えると，決定不能な OWL Full にどのような制約を付加すれば推論は停止するのか？この答えは，Description Logic(DL) の形式的な成果を介して説明ができる．

本解説では，OWL の 3 つの下位言語のうち特に OWL DL/Lite が DL に対応づけれられ，その結果から OWL オントロジーの推論に関する計算量を示す．DL は構成要素 (constructors) の違いにより様々な表現力をもつ言語に分類されている．従って，OWL DL/Lite と同等な DL 言語を導入すれば，オントロジーの記述を DL の知識ベースに変換できる．さらに，OWL オントロジーの推論がオントロジー含意 (entailment) であるのに対して，DL の推論が概念の充足可能性判定であることを考慮しなければならない．言い換えると，OWL ではオントロジーから任意の 2 つのクラスが上位下位の関係にあるかどうかを

OWL Reasoning and its Complexity.

Ken Kaneiwa, 国立情報学研究所, National Institute of Informatics.

コンピュータソフトウェア, Vol.22, No.4(2005), pp.26–34.

2005 年 2 月 9 日受付.

導くのに対して，DL の推論は概念や知識ベースに対して意味モデルの存在を判定することである．よって，これらの表現と推論の違いを一致させれば OWL の推論に必要な計算量が明らかになる．

本稿の構成は以下の通りである．最初に，2 章で OWL の下位言語それぞれの表現力の違いを概説する．3 章では，OWL のオントロジー記述に相当する DL の構文，意味論および知識ベースを導入して，OWL DL/Lite の表現を DL に変換する．4 章では，OWL オントロジーの推論すべてがオントロジー含意に置き換えられ，さらに DL 知識ベースの充足可能性に還元される．その結果，オントロジー含意と DL の充足可能性との一致から OWL の計算量を示す．最後に，5 章で本解説のまとめを述べる．

2　Web オントロジー言語 OWL

本章では，OWL の下位言語 OWL Full/DL/Lite の表現力について述べる．OWL Full と OWL DL は同じ構成要素を持つが，OWL DL には構文に制限が課せられている．その上 OWL Lite は，OWL DL からクラスとプロパティの複雑な表現を取り除いた言語である．本解説の目的は，OWL の言語仕様を解説するのではなく OWL の計算量 (決定可能性) を説明することである．従って，OWL の構文や意味論を網羅せず，OWL Lite/DL が上位言語 (OWL DL/Full) に比べて，どのように制限されているかに注目する．

2.1　OWL Lite/DL

OWL Lite は，単純で読みやすいオントロジーの記述を許した言語で，フレーム言語に似た構文を持つ．表 1(対応関係の左側) に示すようなクラス，個体とプロパティの公理を使ってオントロジーが記述される．`A`,`Ai`,`Ai'` はクラス名，`R`,`Ri` はオブジェクトプロパティ，`d`,`di`,`di'` はデータ型，`T`,`Ti` はデータ型プロパティ，`Si` はクラス名または Restriction，`o`,`oi` は個体名，`v`,`vi` はデータ値，`Ii` は個体名または個体の公理を表す[†1]．例えば，以下はクラス `Human` の下位クラスとしてクラス `Man` を新しく導入している．

```
Class(Man partial Human)
```

本稿では人間が読みやすいこの抽象構文を用いるが，RDF/XML の構文で表すと以下のようになる．

```
<owl:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#Human"/>
</owl:Class>
```

また，クラスの公理に現れる `Si` は，`Human` のようなアトミックなクラス名だけでなく，

```
restriction(hsaChild allValuesFrom(Human))
```

「子供をもち，それが人間である個体」即ち「人間の親」のようにオブジェクトプロパティ `hsaChild` とクラス `Human` からなる Restriction を記述できる．

OWL Lite は，その上位言語 OWL DL(表 2 に示す) と比べて次の制限がある．

1. プロパティ値の基数宣言 (cardinarity) が 0 か 1 に限られる．
2. 上位クラスやクラス間の等式がクラス名で宣言され，複雑なクラスを用いない．即ち，否定 (complement)，選言 (union)，連言 (intersection) の組み合わせによる表現がない．
3. 個体の列挙によるクラス表現 (enumerated class) を許さない．
4. `restriction(R value(o))` によるプロパティ値の個体表現を導入しない．

最初の制限により，プロパティが一意に値を取るかどうかの指定だけ許される．OWL DL では，基数は 0 以上の整数で自由に指定できる．オントロジーに複雑なクラス定義が可能な OWL DL に対して，OWL Lite では 2 つ目の制限により簡潔なクラス名の概念階層でオントロジーが構築される．最後の 2 つの制限は，共に個体クラス (インスタンスが唯一であるクラス) を禁止している．なぜならば，個体クラスは推論の計算量を増やす要因になるからである．

2.2　OWL DL/Full

OWL DL は OWL Lite の制限を緩めて，クラスの論理表現，プロパティ値の基数宣言および個体クラス

†1　クラスは個体集合，オブジェクトプロパティは個体間の 2 項関係，データ型プロパティは個体とデータ値との 2 項関係を示す．また，Restriction はプロパティにより限定された個体集合を表す．

を備える．表 2 に示すように，`C`,`Ci`,`Ci'` が Description(クラス名，Restriction または否定，選言，連言と個体の列挙の組み合わせによる表現) を表し，`D`,`Di`,`Di'` がデータ域 (データ型またはデータ値の列挙) を表し，構文がより構造的になる．即ち，クラス/個体/プロパティの公理に現れる `C`,`Ci`,`Ci'` はクラス名だけでなく，Restriction や論理結合子を含むことができる．例えば，次のように `intersectionOf(...)` 部分「男の人かつ人間の親」即ち「父親」を表す Description により，クラス `Father` が定義される．

```
Class(Father complete
      intersectionOf(Man restriction(
          hsaChild allValuesFrom(Human))))
```

OWL Full は，OWL DL の全ての表現を含むだけでなく，RDF で許す限り自由な記述が可能である．2 つの表現力の差違は，OWL DL に対する以下の制限から生じる．

1. クラス，プロパティ，個体の名前は互いに排反である (同じ名前を別の目的で使えない).
2. クラスのクラス，および個体が別個体のインスタンスとなるような高階な表現を排除する．
3. 次のいずれかが成り立つとき，オブジェクトプロパティは推移性の宣言ができない．
   (a) 関数的または逆関数的である．
   (b) 基数宣言にそのプロパティが使われている．
   (c) 推移性の宣言ができないプロパティの逆プロパティまたは下位プロパティである．

尚，OWL Lite は OWL DL の下位言語なので，OWL DL の構文的な制限は OWL Lite にも適用される．OWL DL では，各表現がクラス，プロパティ，個体のいずれか 1 つの役割しかもたない．従って，2 つ目の制限にも関連して，クラスが他のクラスのインスタンスにはならない．これは論理的な推論のサポートに必要不可欠な条件だが，オントロジー記述の上では大きな制約と考えられる．実際，クラスのクラスのような表現は容易に思いつく．例えば，クラス「色」のインスタンスとして「赤」があるが，それも「赤いもの」をインスタンスとするクラスでもある．OWL Full はグラフ表現に相当しており論理的な制約が無く，プロパティをクラスや個体とみなすことさえ可能である．最後の制限は，それらのオブジェクトプロパティの推移性を許すと決定不能性が導かれしまう理由から導入されている．

3 OWL と Description Logic

本章では，Description Logic(DL) [3][18] の構文，意味論と知識ベースを定義し，OWL Lite/DL のオントロジー記述から DL 言語の知識ベースへの変換を示す．

3.1 Description Logic

DL 言語は，構成要素 (constructors) により異なる表現力をもつ言語ファミリーを形成する．最初に，基本的な DL 言語 $\mathcal{ALC}$ [15] を導入する．$\mathcal{ALC}$ の言語は，概念名 A の集合 $\mathbf{C}$，抽象ロール名 R の集合 $\mathbf{R}_A$，個体名 o の集合 $\mathbf{I}$，論理結合子 $\sqcap$(連言), $\sqcup$(選言), $\neg$(否定) と量化子 $\exists$(存在), $\forall$(全称) から構成される[†2]．$\mathcal{ALC}$ 言語の概念 ($\mathcal{ALC}$ 概念と呼ぶ) は，以下の構文規則によって帰納的に定義される．

$$A \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall R.C \mid \exists R.C$$

ここで，A は概念名，R は抽象ロール名，C, C_1, C_2 は一般的な概念を表す．

次に $\mathcal{ALC}$ 言語の意味論を定義する．解釈 $\mathcal{I}$ は，対象領域 $\Delta^{\mathcal{I}}$ (空でない集合) と解釈関数 $\cdot^{\mathcal{I}}$ との 対 $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ である．解釈関数 $\cdot^{\mathcal{I}}$ は，概念名，抽象ロール名，個体名に対して，$\Delta^{\mathcal{I}}$ の部分集合，$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ の部分集合，$\Delta^{\mathcal{I}}$ の要素をそれぞれ割り当てる．$\mathcal{ALC}$ 概念の解釈は，以下のように帰納的に定義される．

$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} - C^{\mathcal{I}}$$
$$(C_1 \sqcap C_2)^{\mathcal{I}} = (C_1)^{\mathcal{I}} \cap (C_2)^{\mathcal{I}}$$
$$(C_1 \sqcup C_2)^{\mathcal{I}} = (C_1)^{\mathcal{I}} \cup (C_2)^{\mathcal{I}}$$
$$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y[(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}]\}$$
$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y[(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}]\}$$

特に，概念名 $\top$(最大概念) と $\bot$(空概念) の解釈は $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ と $\bot^{\mathcal{I}} = \emptyset$ である．ある概念 C に対して，$C^{\mathcal{I}} \neq \emptyset$ である解釈 $\mathcal{I}$ が存在するとき，C は充足可能であり，$\mathcal{I}$ は C を充足するという．

Horrocks ら [13][11] は，セマンティック Web など

†2 概念名は個体集合を示し，ロールは 2 項関係を示す．

の応用向けに推移的ロール (推移的ロール名の集合を $\mathbf{R}_A^+ \subseteq \mathbf{R}_A$ で表す) や逆ロール R^- を導入した DL 言語を提案している．言語名には，推移的ロールを $\mathcal{S}$，逆ロールを $\mathcal{I}$ で記して，これらを追加した $\mathcal{ALC}$ 言語を略して $\mathcal{SI}$ と呼ぶ．推移的ロールと任意のロール R の逆ロール R^- の解釈を以下に定義する．

$$R^{\mathcal{I}} = (R^{\mathcal{I}})^+ (\text{但し，} R \in \mathbf{R}_A^+)$$
$$(R^-)^{\mathcal{I}} = \{(x,y) \mid (y,x) \in R^{\mathcal{I}}\}$$

ここで，$(R^{\mathcal{I}})^+$ は $R^{\mathcal{I}}$ の推移的閉包を示す．この $\mathcal{SI}$ は，他の構成要素を追加して $\mathcal{SI}$ 言語ファミリーを形成する．例えば，以下のロール階層 ($\mathcal{H}$)

$$R_1 \sqsubseteq R_2$$

(R_1 は R_2 の下位ロールである) と関数的ロール ($\mathcal{F}$) を追加した言語は，$\mathcal{SHIF}$ で表される．ロール階層 $R_1 \sqsubseteq R_2$ に対する解釈は，次のように定義される．

$$(R_1)^{\mathcal{I}} \subseteq (R_2)^{\mathcal{I}}$$

抽象ロール R が関数的ならば，以下を満たす．

$$(x,y) \in R^{\mathcal{I}} \wedge (x,z) \in R^{\mathcal{I}} \Rightarrow y = z$$

以下の構成要素は，数量限定 (number restriction)($\mathcal{N}$) と個体概念 (nominal)($\mathcal{O}$) である．

$$\leq nR,\ \geq nR,\ \{o_1, \ldots, o_n\}$$

$\leq nR$ は「高々n 個の要素をもつ」を表し，$\geq nR$ は「少なくとも n 個の要素をもつ」を表す．個体概念は，個体 $o_1, \ldots, o_n$ を列挙した概念を表している．数量限定と個体概念の解釈は，次のように与えられる．

$$(\leq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \| \{y \in \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}| \leq n\}$$
$$(\geq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \| \{y \in \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}| \geq n\}$$
$$(\{o_1, \ldots, o_n\})^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$$

ここまでに扱った概念や個体は抽象的対象であったが，他に文字列や自然数などのデータ型 (concrete datatype) を扱う表現がある．ここで，データ型 d の集合 $\mathbf{D}$ とデータ型ロール T の集合 $\mathbf{R}_{\mathbf{D}}$ を導入すると，次の概念表現が追加される．

$$\forall T.d,\ \exists T.d,\ \leq nT,\ \geq nT$$

例えば，$\forall age.Nat$ は「年齢をもつならば，それは自然数である個体」を意味する．また，個体 o と同様にデータ型の要素であるデータ値 v を用いる (例えば，1 はデータ型 Nat のデータ値である)．データ値の列挙による概念は，以下のように表される．

$$\{v_1, \ldots, v_n\}$$

データ型の対象領域は $d^{\mathbf{D}}$ で表され，すべてのデータの対象領域は $\Delta_{\mathbf{D}}^{\mathcal{I}} (\supseteq d^{\mathbf{D}})$ で表される．ここで解釈関数 $\cdot^{\mathcal{I}}$ を拡張して，すべてのデータ型ロール名に $\Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$ の部分集合を割り当てる．それによりデータ型に関する概念の解釈は，次のように定義される．

$$(\forall T.d)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y[(x,y) \in T^{\mathcal{I}} \rightarrow y \in d^{\mathbf{D}}]\}$$
$$(\exists T.d)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y[(x,y) \in T^{\mathcal{I}} \wedge y \in d^{\mathbf{D}}]\}$$
$$(\{v_1, \ldots, v_n\})^{\mathcal{I}} = \{v_1^{\mathbf{D}}, \ldots, v_n^{\mathbf{D}}\}$$

$\mathcal{SHIF}(D)$ は $\mathcal{SHIF}$ にデータ型とデータ型ロールを追加した DL 言語である．さらに，$\mathcal{SHOIN}(D)$ は $\mathcal{SHIF}(D)$ に数量限定と個体概念を追加している†3.

3. 2 DL の知識ベース

前節で導入した DL 概念に関する知識ベースを†4定義する．知識ベースは，概念階層の言明 (T-Box 言明と呼ぶ) と，概念とロールのインスタンス言明 (A-Box 言明と呼ぶ) に区分される．次の 2 つは，概念とロールの A-Box 言明である．

$$o: C,\ (o_1, o_2): R$$

個体 o が概念 C のインスタンスであること，個体 o_1, o_2 の対がロール R のインスタンスであることを示す．さらに T-Box 言明は，次のように表される．

$$C_1 \sqsubseteq C_2,\ R_1 \sqsubseteq R_2$$

前者は概念 C_2 が C_1 を包含する (subsume)，または C_1 が C_2 の下位概念であることを示す．後者はロール R_2 が R_1 を包含する，または R_1 が R_2 の下位ロールであることを示す．知識ベース KB は，A-Box 言明と T-Box 言明の有限集合として定義される．

データ型を扱う場合，データ型ロール T に関する A-Box 言明は以下により与えられる．

$$(o, v): T$$

データ型ロールは，抽象ロールと違って個体 o とデータ値 v の対がインスタンスとなる．抽象ロール同様に，以下はデータ型ロールの T-Box 言明である．

$$T_1 \sqsubseteq T_2$$

尚，データ型は既に固定した領域 (自然数の集合や文

†3 数量限定 ($\leq 1R$) は関数的ロールを記述できるので，$\mathcal{SHOIN}(D)$ は関数的ロールを含むとみなせる．

†4 一般的に，知識ベースは事実や規則などの集合のことを言う．

字列の集合) をもつので，データ型のインスタンス (A-Box 言明) は宣言されない．

A-Box 言明の解釈は，次のように与えられ，

$o^{\mathcal{I}} \in C^{\mathcal{I}}$, $(o_1^{\mathcal{I}}, o_2^{\mathcal{I}}) \in R^{\mathcal{I}}$, $(o^{\mathcal{I}}, v^{\mathrm{D}}) \in T^{\mathcal{I}}$

T-Box 言明の解釈は，以下のように与えられる．

$(C_1)^{\mathcal{I}} \subseteq (C_2)^{\mathcal{I}}$, $(R_1)^{\mathcal{I}} \subseteq (R_2)^{\mathcal{I}}$, $(T_1)^{\mathcal{I}} \subseteq (T_2)^{\mathcal{I}}$

これにより，知識ベース KB 内のすべての A-Box 言明と T-Box 言明に対して，以上を満たす解釈 $\mathcal{I}$ が存在するとき，$\mathcal{I}$ は KB を充足するという．

3.3 OWL と DL の対応関係

Web オントロジー WG の主要メンバー Horroks と Patel-Schneider は，[9] [12] において OWL Lite/DL から DL 言語への変換を定義した．まず，OWL のクラス，オブジェクトプロパティ，データ型プロパティは，以下のように DL の要素に対応づけられる．

クラス `A` $\Leftrightarrow$ 概念名 A

オブジェクトプロパティ `R` $\Leftrightarrow$ 抽象ロール R

データ型プロパティ `T` $\Leftrightarrow$ データ型ロール T

個体 `o`，データ型 `d`，データ値 `v` は，DL でも同じ名称の表現 (個体 o，データ型 d，データ値 v) がある．

表 1 と表 2 により，$\mathcal{T}$ は $\mathcal{T}(\mathtt{A}) = A$ かつ $\mathcal{T}(\mathtt{d}) = d$ であるような OWL から DL への変換である．OWL Lite の公理は $\mathcal{SHIF}(D)$ で表現され (表 1)，OWL DL の公理は $\mathcal{SHOIN}(D)$ で記述される (表 2) †5．$\sqcap_{1\le i\le n} X_i (\sqcup_{1\le i\le n} X_i)$ は，$X_1 \sqcap \cdots \sqcap X_n (X_1 \sqcup \cdots \sqcap X_n)$ を表す．OWL の構文内で `[ ]` はオプションを示しており，個体の公理では `[o]` のように個体名を明示してもしなくてもいい．もし，`o` が明示されているならば，DL への変換で $x\colon A_i$ (表 1) と $x\colon \mathcal{T}(C_i)$ (表 2) において x は o を表す．明示されてなければ，x は新しい変数とする．さらに `value(Ri Ii)` を変換する際，`Ii` 内で個体名 `oi` が明示されていれば y_i は o_i を表し，無記名ならば $\mathcal{T}(I_i)$ で導入された新しい変数を示す．また，$Trans(R)$ は R が推移的ロールであることを示す (即ち，$R \in \mathbf{R}_A^+$)．表 2 で，`DisjointClasses(C1...Cn)` は，$1 \le i < j \le n$ に対する全ての $\mathcal{T}(C_i) \sqsubseteq \neg\mathcal{T}(C_j)$ へ変換される．`EquivalentClasses(C1...Cn)` は，$1 \le i \le n-1$ に対する全ての $\mathcal{T}(C_i) \equiv \mathcal{T}(C_{i+1})$ へ変換される．

これらの対応関係から，OWL DL ではクラス名，Restriction と論理結合子からなる Description，さらにはデータ域 (データ型，またはデータ値の列挙) を含んで概念表現がより構造的になっている．そのため，OWL DL の変換では再帰的な定義が多く見られる．例えば，表 2 の左中央にある Restriction の変換で，DL に変換された右側の記述に再び変換関数 $\mathcal{T}(C)$ が現れる部分が再帰的である．

†5 表現力の高い OWL DL と $\mathcal{SHOIN}(D)$ の対応関係だけで十分だが，本稿では理解のために OWL Lite と $\mathcal{SHIF}(D)$ の対応関係も示す．

4 オントロジーの推論と計算量

OWL と DL の対応関係から，DL の研究成果を用いて OWL の計算量 (決定可能性) を示す．

4.1 OWL オントロジーの含意

ここで OWL によって構築されたオントロジーのことを，OWL オントロジーと呼ぶこととする．OWL の推論とは，ある OWL オントロジー $\mathcal{O}$ が与えられたとき，$\mathcal{O}$ において (1) クラス `C1` がクラス `C2` の下位クラスであるかどうか，および (2) 個体 `o` がクラス `C` のインスタンスであるかどうか，を導くことである．これらの推論は，以下の公理を含む他のオントロジーを導くことに他ならない．

- `SubClassOf(C1 C2)`
- `Individual(o type(C))`

例えば，オントロジーの公理を ont_i で表すとする．そのとき OWL オントロジー $\mathcal{O}$ は，公理の有限集合 $\{ont_1, \ldots, ont_n\}$ で与えられ，今述べたクラス間の上位下位関係もしくはインスタンス言明が，オントロジーの公理 $\{ont\}$ で記述できる．これより，オントロジーの推論は次のオントロジー含意に還元できる．

$$\{ont_1, \ldots, ont_n\} \models \{ont\}$$

例えば，$\mathcal{O} \models \{$`SubClassOf(C1 C2)`$\}$ のように還元される．ここでオントロジー含意の定義を与える．OWL オントロジー $\mathcal{O}$ を DL の知識ベースに変換したとき ($\mathcal{T}(\mathcal{O})$ で表す)，知識ベース $\mathcal{T}(\mathcal{O})$ を充足する (satisfies) 解釈を $\mathcal{O}$ のモデルという．2 つの OWL オントロジー $\mathcal{O}_1, \mathcal{O}_2$ が与えられたとする．このとき，

表 1 OWL Lite と $\mathcal{SHIF}(D)$ との対応関係

クラスの公理 A_C	DL $\mathcal{T}(A_C)$	プロパティの公理 A_P	DL $\mathcal{T}(A_P)$
`Class(A partial S1...Sn)`	$A \sqsubseteq \sqcap_{1\le i\le n}\mathcal{T}(S_n)$	`DataProperty(T`	
`Class(A complete S1...Sn)`	$A \equiv \sqcap_{1\le i\le n}\mathcal{T}(S_n)$	`super(T1)...super(Tn)`	$T \sqsubseteq T_i$
`EquivalentClasses(A1...An)`	$A_1 \equiv \cdots \equiv A_n$	`[Functional]`	$\top \sqsubseteq \le 1T$
Restriction r	DL $\mathcal{T}(r)$	`domain(d1)...domain(dm)`	$\ge 1R \sqsubseteq d_i$
`restriction(R allValuesFrom(A))`	$\forall R.A$	`range(d1')...range(dl'))`	$\top \sqsubseteq \forall T.d'_i$
`restriction(R someValuesFrom(A))`	$\exists R.A$	`ObjectProperty(R`	
`restriction(T allValuesFrom(d))`	$\forall T.d$	`super(R1)...super(Rn)`	$R \sqsubseteq R_i$
`restriction(T someValuesFrom(d))`	$\exists T.d$	`[InverseOf R0]`	$R \equiv (R_0)^-$
`restriction(R minCardinality(0))`	$\ge 0R$	`[Symmetric]`	$R \equiv R^-$
`restriction(R minCardinality(1))`	$\ge 1R$	`[Functional]`	$\top \sqsubseteq \le 1R$
`restriction(R maxCardinality(0))`	$\le 0R$	`[InverseFunctional]`	$\top \sqsubseteq \le 1R^-$
`restriction(R maxCardinality(1))`	$\le 1R$	`[Transitive]`	$Trans(R)$
`restriction(R cardinality(0))`	$\le 0R \sqcap \ge 0R$	`domain(A1)...domain(Am)`	$\ge 1R \sqsubseteq A_i$
`restriction(R cardinality(1))`	$\le 1R \sqcap \ge 1R$	`range(A1')...range(Al'))`	$\top \sqsubseteq \forall R.A'_i$
個体の公理 A_o	DL $\mathcal{T}(A_o)$	`EquivalentProperties(`	
`Individual([o]`		`T1...Tn)`	$T_i \equiv T_{i+1}$
`type(A1)...type(An)`	$x\colon A_i$	`SubPropertyOf(T1 T2)`	$T_1 \sqsubseteq T_2$
`value(R1 I1)...value(Rn Im)`	$(x, y_i)\colon R_i,\ \mathcal{T}(I_i)$	`EquivalentProperties(`	
`value(T1 v1)...value(Tn vk))`	$(x, v_i)\colon T_i$	`R1...Rn)`	$R_i \equiv R_{i+1}$
`SameIndividual(o1...on)`	$o_1 = \cdots = o_n$	`SubPropertyOf(R1 R2)`	$R_1 \sqsubseteq R_2$
`DifferentIndividuals(o1...on)`	$o_i \neq o_j (i \neq j)$		

OWL オントロジー $\mathcal{O}_1$ のすべてのモデルが，OWL オントロジー $\mathcal{O}_2$ のモデルでもあるならば，$\mathcal{O}_1$ は $\mathcal{O}_2$ を含意する (entails) といい，$\mathcal{O}_1 \models \mathcal{O}_2$ で表す．

4.2 OWL Lite/DL の計算量

OWL Lite/DL におけるオントロジー含意の決定可能性を示す．OWL Lite/DL は同等な DL 言語への変換により，その DL と同じ計算量クラスに属すと言える．しかし DL の推論は概念や知識ベースの充足可能性判定なので，オントロジー含意をそれに還元する必要がある．表 1 と表 2 では，充足可能性を保ちながらオントロジー $\mathcal{O}$ を DL の知識ベースに変換している ($\mathcal{T}(\mathcal{O})$ で表す)．これに加えて，表 3 と表 4 は DL の知識ベース言明をその否定表現へ変換する．ここで述べる変換方法は，文献 [9] の定義に基づいており，その妥当性も保証されている．

例えば，表 3 の $C_1 \sqsubseteq C_2$ の否定は，$x\colon C_1 \sqcap \neg C_2$ となる．これは，$C_1 \sqsubseteq C_2$ が「C_1 のすべてのインスタンスは，C_2 のインスタンスである」を意味するので，その否定として $x\colon C_1 \sqcap \neg C_2$「ある x が存在して，C_1 かつ $\neg C_2$ のインスタンスとなる」へ変換される．表 4 の $x\colon C$ は「ある x が存在して，C のインスタンスとなる」なので，その否定の $\top \sqsubseteq \neg C$ は「すべてのインスタンスは，$\neg C$ のインスタンスである」を意味する．他の言明も同様に変換される．表 4 の $\overline{v}$ は，$\Delta^{\mathcal{I}}_{\mathrm{D}} - \{v^{\mathrm{D}}\}$ のデータ型を示す．

表 3 T-Box 言明の否定

T-Box 言明 s	変換 $\mathcal{G}(s)$
$C_1 \sqsubseteq C_2$	$x\colon C_1 \sqcap \neg C_2$
$R_1 \sqsubseteq R_2$	$x\colon A_0 \sqcap \exists R_1.(\forall (R_2)^-.\neg A_0)$
$T_1 \sqsubseteq T_2$	$x\colon \exists T_1.D_0 \sqcap \neg \exists T_2.D_0$
$Trans(R)$	$x\colon A_0 \sqcap \exists R.(\exists R.(\forall R^-.\neg A_0))$

これらの変換により，以下のようにオントロジー含意が DL の充足不可能性へ還元できる．

$$\begin{aligned}\mathcal{O}_1 \models \mathcal{O}_2 &\Leftrightarrow \mathcal{T}(\mathcal{O}_1) \models \mathcal{T}(\mathcal{O}_2)\\ &\Leftrightarrow \text{すべての } s \in \mathcal{T}(\mathcal{O}_2) \text{ に対して,}\\ &\qquad \mathcal{T}(\mathcal{O}_1) \cup \{\mathcal{G}(s)\} \text{ が充足不可能}\end{aligned}$$

図 1 は，OWL Lite/DL と同等な DL の知識ベースに関する充足可能性問題の計算量 [17] を示している．各言語はデータ型無しであるが，計算量に違いがないのでこれらの結果を利用できる．EXPTIME[†6] と

†6 EXPTIME は，指数関数時間で計算可能なクラスを示す．

表 2　OWL DL と $\mathcal{SHOIN}(D)$ との対応関係

クラスの公理 A_C	DL $\mathcal{T}(A_C)$	プロパティの公理 A_P	DL $\mathcal{T}(A_P)$
`Class(A partial C1...Cn)`	$A \sqsubseteq \sqcap_{1\le i\le n}\mathcal{T}(C_n)$	`DataProperty(T`	
`Class(A complete C1...Cn)`	$A \equiv \sqcap_{1\le i\le n}\mathcal{T}(C_n)$	`super(T1)...super(Tn)`	$T \sqsubseteq T_i$
`EnumeratedClass(A o1...on)`	$A \equiv \{o_1,\ldots,o_n\}$	`[Functional]`	$\top \sqsubseteq \le 1T$
`DisjointClasses(C1...Cn)`	$\mathcal{T}(C_i) \sqsubseteq \neg\mathcal{T}(C_j)$	`domain(D1)...domain(Dm)`	$\ge 1R \sqsubseteq \mathcal{T}(D_i)$
`EquivalentClasses(C1...Cn)`	$\mathcal{T}(C_i) \equiv \mathcal{T}(C_{i+1})$	`range(D1')...range(Dl'))`	$\top \sqsubseteq \forall T.\mathcal{T}(D'_i)$
`SubClassOf(C1 C2)`	$\mathcal{T}(C_1) \sqsubseteq \mathcal{T}(C_2)$	`ObjectProperty(R`	
Description c	DL $\mathcal{T}(c)$	`super(R1)...super(Rn)`	$R \sqsubseteq R_i$
`unionOf(C1...Cn)`	$\sqcup_{1\le i\le n}\mathcal{T}(C_n)$	`[InverseOf R0]`	$R \equiv (R_0)^-$
`intersectionOf(C1...Cn)`	$\sqcap_{1\le i\le n}\mathcal{T}(C_n)$	`[Symmetric]`	$R \equiv R^-$
`complementOf(C)`	$\neg\mathcal{T}(C)$	`[Functional]`	$\top \sqsubseteq \le 1R$
`oneOf(o1...on)`	$\{o_1,\ldots,o_n\}$	`[InverseFunctional]`	$\top \sqsubseteq \le 1R^-$
Restriction r	DL $\mathcal{T}(r)$	`[Transitive]`	$Trans(R)$
`restriction(R allValuesFrom(C))`	$\forall R.\mathcal{T}(C)$	`domain(C1)...domain(Cm)`	$\ge 1R \sqsubseteq \mathcal{T}(C_i)$
`restriction(R someValuesFrom(C))`	$\exists R.\mathcal{T}(C)$	`range(C1')...range(Cl'))`	$\top \sqsubseteq \forall R.\mathcal{T}(C'_i)$
`restriction(R value(o))`	$\exists R.\{o\}$	`EquivalentProperties(`	
`restriction(T allValuesFrom(D))`	$\forall T.\mathcal{T}(D)$	`T1...Tn)`	$T_i \equiv T_{i_1}$
`restriction(T someValuesFrom(D))`	$\exists T.\mathcal{T}(D)$	`SubPropertyOf(T1 T2)`	$T_1 \sqsubseteq T_2$
`restriction(T value(v))`	$\exists T.\{v\}$	`EquivalentProperties(`	
`restriction(R minCardinality(n))`	$\ge nR$	`R1...Rn)`	$R_i \equiv R_{i+1}$
`restriction(R maxCardinality(n))`	$\le nR$	`SubPropertyOf(R1 R2)`	$R_1 \sqsubseteq R_2$
`restriction(R cardinality(n))`	$\le nR \sqcap \ge nR$	データ域 D	$\mathcal{T}(D)$
個体の公理 A_o	DL $\mathcal{T}(A_o)$	`oneOf(v1...vn)`	$\{v_1,\ldots,v_n\}$
`Individual([o]`			
`type(C1)...type(Cn)`	$x\colon \mathcal{T}(C_i)$		
`value(R1 I1)...value(Rn Im)`	$(x, y_i)\colon R_i,\ \mathcal{T}(I_i)$		
`value(T1 v1)...value(Tn vk))`	$(x, v_i)\colon T_i$		
`SameIndividual(o1...on)`	$o_1 = \cdots = o_n$		
`DifferentIndividuals(o1...on)`	$o_i \neq o_j (i \neq j)$		

表 4　A-Box 言明の否定

A-Box 言明 s	変換 $\mathcal{G}(s)$
$o\colon C$	$o\colon \neg C$
$x\colon C$	$\top \sqsubseteq \neg C$
$(o_1, o_2)\colon R$	$o_2\colon A_0,\ o_1\colon \forall R.\neg A_0$
$(o, x)\colon R$	$o\colon A_0,\ \top \sqsubseteq \forall R^-.\neg A_0$
$(x, o)\colon R$	$o\colon A_0,\ \top \sqsubseteq \forall R.\neg A_0$
$(x, y)\colon R$	$\top \sqsubseteq \neg(\ge 1T)$
$(o, v)\colon T$	$o\colon \forall T.\overline{v}$
$(x, v)\colon T$	$\top \sqsubseteq \forall T.\overline{v}$
$o_1 = o_2$	$o_1 \neq o_2$
$o_1 \neq o_2$	$o_1 = o_2$

NEXPTIME[†7]は，DL の知識ベースに含まれるすべての言明の長さに関して，EXPTIME-完全および NEXPTIME-完全であることを示す．

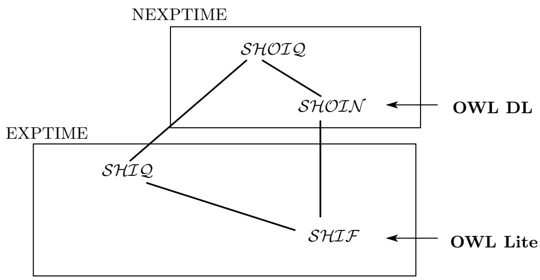

図 1　DL の計算量

OWL Lite は $\mathcal{SHIF}(D)$ に対応するので EXPTIME-完全である．注目すべきことに，より表現力の高い $\mathcal{SHIQ}(D)$ の計算量は指数関数時間に及ぶが，最適化を施すことで効率的な推論システム RACER [7] が実装されている．文献 [6] では，10 万個以上の概

†7 NEXPTIME は，非決定的チューリングマシンにより指数関数時間で計算可能なクラスを示す．

念を含む大規模データを使ってその推論システムの実用性が評価されている．従って最近の DL 研究では，EXPTIME でも実現可能な (tractable) 計算量クラスと判断される．一方，OWL DL は $\mathcal{SHOIN}(D)$ に対応して NEXPTIME-完全である．しかし NEXPTIME-完全の DL には，効率の良いアルゴリズムはまだ開発されておらず，OWL Lite の方が実装に適している．このような理由から，NEXPTIME は実現が困難な (intractable) 計算量クラスと認識される．加えて，図 1 の EXPTIME と NEXPTIME との境界が示すように，DL では逆ロール $\mathcal{I}$ と個体概念 $\mathcal{O}$ の組み合わせが，計算量の増大要因と指摘されている．

4.3 OWL とルールの決定不能性

ルール表現は，OWL のオントロジー記述には無い複雑で一般的な法則を記述するのに有効な手段である．そのため OWL のクラスやプロパティに関するルール表現を導入するために，OWL Lite/DL に RuleML(Rule Makeup Language) [2] の下位言語を取り込んだ SWRL(Semantic Web Rule Language) [10] が提案された．SWRL は OWL との互換性を保ち，以下に定義されるルール表現を含む．

```
Implies( Antecedent(a1...an)
         Consequent(a1'...am') )
```

`ai` と `aj'` はアトムと呼ばれ，`C(x)`，`d(y)`，`R(x1,x2)`，`T(x,y)`，`sameAs(x1,x2)`，`differentFrom(x1,x2)`，`builtin(r,x1,...,xn)` のいずれかである．最初の 4 つのアトムは，それぞれ Description `C`，データ型 `d`，オブジェクトプロパティ`R`，データ型プロパティ `T` のインスタンス言明であり，`x`,`x1`,`x2`,`xn` は個体名または個体変数，`y` はデータ値またはデータ変数を示す．`sameAs(x1,x2)` と `differentFrom(x1,x2)` の 2 つは個体変数を許した個体の公理であり，`builtin(r,x1,...,xn)` は `x1`, ... ,`xn` を引数とする組み込み述語 `r` である．このようなルールの直観的な意味は「アトム `a1` から `an` が成り立つならば，アトム `a1'` から `am'` が成り立つ」である．

ルール表現を使うと，次のプロパティによる規則が記述できる．

```
Implies( Antecedent(north(x,y) east(y,z))
         Consequent(diagonal(x,z)))
Implies( Antecedent(east(x,v) north(v,w))
         Consequent(diagonal(x,w)))
```

これらのルールの意味は「`x` の北が `y` かつ `y` の東が `z` ならば，`x` の斜めは `z` である」と「`x` の東が `v` かつ `v` の北が `w` ならば，`x` の斜めは `w` である」である．しかし文献 [8] では，このルールが決定不能性を導く例として紹介されている．

従って，クラスとプロパティのルール表現を制限しないで OWL DL/Lite に追加すると，決定可能性を失う．W3C の仕様書でも SWRL は決定不能と明記されている．一つの試みとして，Eiter ら [5] はクラスとプロパティの代わりに各々に対応する単項/二項述語 (入力述語と呼び，元のクラス/プロパティに近い名前で表す) をルールに導入することで OWL DL/Lite との融合を実現し，決定可能性 (計算量) を保たせた．しかし OWL オントロジーをルールに反映させる記述が少々複雑で，十分に自然なルール表現を許しながら決定可能性を保つには課題が残る．これに関連して，バーナーズ・リーの基調講演 (国際会議 WWW2005) で示された新しいセマンティック Web の技術層 [1] では，OWL 層とルール層が並列に並べられその下に OWL とルールの共通部分として決定可能な DLP(Description Logic Programs) 層が設けられている．

5 おわりに

本解説では，アルゴリズムや計算量が徹底的に研究されている DL の成果を用いて，OWL における推論の決定可能性について解説した．OWL Full の高い表現力に制限を与えることで OWL DL/Lite が得られ，DL 言語としての議論ができる．また実装を想定した場合，OWL Lite の推論は EXPTIME となるが効率の良い推論システムが存在することを述べた．

しかしながら，セマンティック Web の実現において OWL DL/Lite では表現できないオントロジー情報も考えられる．OWL DL/Lite は決定可能である反面，高階な表現など Web 情報を扱うために便利な記述を失う．その際に OWL Full を使うが，高い表

現力が決定不能性をもたらし実装でのマイナス面が大きい．個人的な意見としては，DL 研究者主導で設計された OWL DL だけでなく，セマンティック Web やオントロジーの研究者による視点で言語表現が選別され，その上で OWL の決定可能な枠組みが提案されることを期待したい．

謝辞

本解説の改善にあたり，編集者および査読者からご指摘をいただきました．ここに感謝致します．

参考文献

[1] Berners-Lee, T.: Web for real people. http://www.w3.org/2005/Talks/0511-keynote-tbl/.

[2] http://www.ruleml.org/.

[3] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P.(eds.): *The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge*, 2003.

[4] Dean, M. and Schreiber, G.: OWL Web Ontology Language Reference, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[5] Eiter, T., Faber, W., Fink, M., Pfeifer, G., and Woltran, S.: Complexity of Model Checking and Bounded Predicate Arities for Non-ground Answer Set Programming., *KR*, 2004, pp. 377–387.

[6] Haarslev, V. and Möller, R.: High Performance Reasoning with Very Large Knowledge Bases: A Practical Case Study., *IJCAI*, 2001, pp. 161–168.

[7] Haarslev, V. and Möller, R.: RACER System Description., *IJCAR*, 2001, pp. 701–706.

[8] Horrocks, I. and Patel-Schneider, P. F.: A Proposal for an OWL Rules Language, *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, ACM, 2004, pp. 723–731.

[9] Horrocks, I. and Patel-Schneider, P. F.: Reducing OWL entailment to description logic satisfiability, *J. of Web Semantics*, (2004). To Appear.

[10] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Recommendation, http://www.w3.org/Submission/SWRL/.

[11] Horrocks, I. and Sattler, U.: Ontology Reasoning in the SHOQ(D) Description Logic, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[12] Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language, *J. of Web Semantics*, Vol. 1,No. 1(2003), pp. 7–26.

[13] Horrocks, I., Sattler, U., and Tobies, S.: Practical Reasoning for Expressive Description Logics, *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR-99)*, LNAI, Vol. 1705, Springer, 1999, pp. 161–180.

[14] Patel-Schneider, P. F., Hayes, P., and Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-semantics-20040210/.

[15] Schmidt-Schauss, M. and Smolka, G.: Attributive concept descriptions with complements, *Artificial Intelligence*, Vol. 48(1991), pp. 1–26.

[16] Smith, M. K., Welty, C., and McGuinness, D. L.: OWL Web Ontology Language Guide, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[17] Tobies, S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation, 2001.

[18] 兼岩憲, 佐藤健: レクチャーシリーズ「哲学と AI における対象世界モデリング」〔第 6 回〕DL: Description Logics, **人工知能学会誌**, Vol. 18,No. 1(2003), pp. 73 – 82.