

# セマンティック Web 推論と議論エージェント推論の統合

## Integration of Semantic Web Reasoning and Argument-based Reasoning

若木 利子

Toshiko Wakaki

芝浦工業大学 システム工学部

College of Systems Engineering, Shibaura Institute of Technology

twakaki@sic.shibaura-it.ac.jp, <http://www.ailab.se.shibaura-it.ac.jp/>

沢村 一

Hajime Sawamura

新潟大学工学部 情報工学科

Faculty of Engineering, Niigata University

sawamura@ie.niigata-u.ac.jp, <http://www.cs.ie.niigata-u.ac.jp/~sawamura/>

福本 太郎

Taro Fukumoto

新潟大学大学院 自然科学研究科

Graduate School of Science and Technology, Niigata University

fukumoto@cs.ie.niigata-u.ac.jp

向井 孝徳

Takanori Mukai

芝浦工業大学大学院 工学研究科

Graduate School of Engineering, Shibaura Institute of Technology

m106096@sic.shibaura-it.ac.jp

新田 克己

Katsumi Nitta

東京工業大学大学院 総合理工学研究科

Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology

nitta@dis.titech.ac.jp, <http://www.dis.titech.ac.jp/staff/nitta.html>

**keywords:** ontological reasoning, argument-based reasoning, multi-agents, description logics, Semantic Web

### Summary

Though many kinds of multi-agent systems based on argumentation have been proposed where only rule-based knowledge is taken into account, they have been unable to handle the ontological knowledge so far. In our daily life, however, there are a lot of human argumentation where both ontological and rule knowledges are used. For example, in e-commerce, a seller and a buyer usually use ontologies about products along with their respective strategic rules for buying and selling. Recent progress of the Semantic Web technology provides expressive ontology languages. In this paper, we demonstrate integration of the Semantic Web reasoning and argument-based reasoning. We have implemented the integrated system such that Logic of Multiple-valued Argumentation-based agent system (specialized to two values  $\{f, t\}$ ) can be accessible to the Semantic Web reasoning established as the description logic reasoning system, given ontologies expressed by OWL DL or its notational variant the DL *SHOIN(D)*. An interesting argumentation result using both ontologies and rules about the university curriculum is shown as an example executed by our system.

### 1. はじめに

これ迄、複数エージェントが各自のルールベースの知識を用いて対話的証明論に基づく議論 (argumentation) により互いにコミュニケーションを行ない、問題解決をするマルチエージェントシステムが提案・実装され、稼動している [Takahashi 04]。しかし、これらのエージェントは対象分野の概念や語彙、及び、それらの関係に関するオントロジー知識を持っていない。しかし一般に、Web上の電子商取引等では、売り手と買い手が、売買に関する戦略的ルール知識以外に、販売店や商品に関するオントロジー知識を用いて議論や交渉を行っている。

近年の技術動向として、2004年にセマンティック Web上にオントロジー表現のための Web オントロジー言語

OWL が W3C により勧告され、OWL のサブ言語である OWL DL、OWL Lite と記述論理 DL *SHOIN(D)*、DL *SHIF(D)* との各々の意味論的同値性も示された [Horrocks 03]。他方、2005年に記述論理の単調推論が、通常の Answer Set Programming (ASP) の非単調推論に縮約して推論ができるという Heymans らの方法が提案されている [Heymans 05]。

そこで本研究では、複数エージェントによる所与の議題の問題解決において、エージェントが各自の当該問題に関する戦略的ルールベースの知識と対象ドメインに関するオントロジー知識を用いて、議論・推論することにより問題解決を行なうマルチエージェントシステムの推論方式の提案と実現を研究の目的とする。ルールとオントロジーという異質の知識を用いる場合、ルールベース

知識を用いた対話的証明論に基づく議論は非単調推論であり、記述論理で表現されたオントロジーを用いた推論は単調推論である。この結果、このような推論の原理が異なる 2 つの推論系を如何に整合的に統合するかが技術的課題となる。

本研究では、各エージェントは所与の議題に関する各自固有の戦略的知識を EALP(Extended Annotated Logic Programs) の論理プログラム (ルール集合) で表現された知識ベースとして持ち、他方、セマンティック Web 上にオントロジー知識が OWL DL で表現され、各エージェントがそれを共有知識として参照できるものと想定する。特にオントロジー推論に ASP を用いた Heymans らの方法を採用する。そして、このような知識環境下で、各エージェントが他エージェントとの議論や交渉時にオントロジーに関する参照・推論が必要になる度に、随時セマンティック Web 推論系を介して Web 上のオントロジー知識に関する単調推論を行ないつつ、EALP のルールベースの知識を用いたエージェントの議論 (非単調推論) が可能となるような、2 つの推論システムの整合的な推論統合の方法を提案する。更に提案手法に基づいて実装した議論エージェントシステムの実行例や評価等を述べる。

本論文では、2 章でセマンティック Web 技術に関する準備、3 章でセマンティック Web 推論系と議論エージェントシステムの推論統合に関する提案、及び提案方法に基づく統合推論システムの試作と実行結果、4 章では構築したシステムの評価や今後の研究課題等に関して述べる。

## 2. 準備

### 2.1 セマンティック Web のオントロジーと OWL 言語

セマンティック Web におけるオントロジー記述言語として W3C により勧告された OWL 言語は、XML ベースの構文を持ち、オントロジーのセマンティクスが表現可能である。OWL には OWL FULL, OWL DL, OWL Lite の 3 種類のサブクラスがある。特に、OWL DL, OWL Lite に関する推論 (entailment) は、それぞれ記述論理  $SHOIN(D)$ ,  $SHIF(D)$  の充足不能性との同値性が示されている [Horrocks 03]。

### 2.2 記述論理 $SHOIN(D)$

記述論理  $SHOIN(D)$  の構文と意味論の概要を以下に示す。(詳細は [Baader 03, Eiter 04] 等の文献を参照)

#### 【定義 1】 (概念とロール)

$D$  をデータタイプの集合、各データタイプ  $d \in D$  のデータ値の集合 ( $d$  のドメイン) を  $dom(d)$  で、 $\bigcup_{d \in D} dom(d)$  を  $dom(D)$  で表す。 $A, D, R, U, I$  を原子的概念、データタイプ、抽象ロール、データタイプロール、個体、のそれぞれの集合とする。抽象ロール  $R_a \in R$  の逆ロール  $R_a^-$  の集合を  $R^-$  とする。ロール  $R$  は、 $R \cup U \cup R^-$  の

要素で定義される。概念は、 $C, C'$  を概念として以下のように帰納的に定義される。

$$C, C' \rightarrow A \mid \neg C \mid C \sqcap C' \mid C \sqcup C' \mid \{o_1, \dots, o_n\} \mid \forall R.C \mid \exists R.C \mid \geq nR \mid \leq nR \mid \forall U.D \mid \exists U.D \mid \geq nU \mid \leq nU$$

但し、 $A \in A, D \in D, R \in R \cup U \cup R^-, U \in U, o_i \in I$ 。

【例 1】 原子的概念 Male と Human より、 $Male \sqcap Human$  は「男の人」という概念を、原子的概念 Father と Mother より、 $Father \sqcup Mother$  は「両親」という概念を表す。抽象ロール hasChild を用いて、 $Human \sqcap (\leq 1 \text{ hasChild})$  は高々 1 人の子供をもつ人を、 $Human \sqcap (\geq 3 \text{ hasChild})$  は 3 人以上の子供をもつ人という概念を表す。データタイプに XML スキーマのデータ型 (例: integer, string) などがある。データタイプロール hasAge は、 $\exists \text{ hasAge.nonNegativeInteger}$  の hasAge のように個体  $o$  をデータ型 nonNegativeInteger の値に関係付けるロールである。

#### 【定義 2】 (知識ベース)

$SHOIN(D)$  の知識ベースは TBox と ABox で構成される。TBox は以下の (1)~(3), ABox は (4)~(7) の形式の公理からなる。ここで、 $\sqsubseteq$  の意味は次の定義 3 に従う。

- (1) 概念包含:  $C \sqsubseteq D$ , (2) ロール包含:  $R \sqsubseteq S, U \sqsubseteq V$ ,
- (3) ロール推移律:  $Trans(R)$ , (4) 概念の要素:  $C(a)$ ,
- (5) ロールの要素:  $R(a, b), U(a, v)$ , (6) 同一個体:  $a = b$ ,
- (7) 異なる個体:  $a \neq b$ 。但し、 $C, D$  は概念、 $R, S \in R, U, V \in U, a, b \in I, v \in dom(D)$  とする。

【定義 3】 ( $SHOIN(D)$  の意味論) 解釈  $I = (\Delta, \cdot^I)$  は、ドメイン  $\Delta$  (但し、 $\Delta \cap dom(D) = \emptyset$ , かつ  $\Delta \neq \emptyset$ ) と解釈関数  $\cdot^I$  からなる。 $\cdot^I$  は、原子的概念  $A \in A$  に  $\Delta$  の部分集合  $A^I$  を、 $o \in I$  に  $o^I \in \Delta$  を、ロール  $R \in R$  に  $\Delta \times \Delta$  の部分集合  $R^I$  を、ロール  $U \in U$  に  $\Delta \times dom(D)$  の部分集合  $U^I$  を割り当てる。記述論理の公理  $F$  の解釈  $I = (\Delta, \cdot^I)$  における充足可能性 ( $I \models F$ ) は以下で定義される。

- (1)  $I \models C \sqsubseteq D$  iff  $C^I \subseteq D^I$ ; (2)  $I \models R \sqsubseteq S$  iff  $R^I \subseteq S^I$ ;
- (3)  $I \models Trans(R)$  iff  $R^I$  が推移的である;
- (4)  $I \models C(a)$  iff  $a^I \in C^I$ ;
- (5)  $I \models R(a, b)$  iff  $(a^I, b^I) \in R^I$ ;
- (6)  $I \models U(a, v)$  iff  $(a^I, v) \in U^I$ ;
- (7)  $I \models a = b$  ( $a \neq b$ ) iff  $a^I = b^I$  ( $a^I \neq b^I$ )。

解釈  $I$  が公理  $F$  を充足する (i.e.  $I$  が  $F$  のモデルである) iff  $I \models F$ 。 $I$  が知識ベース  $\mathcal{K}$  を充足する (i.e.  $I \models \mathcal{K}$  iff 任意の  $F \in \mathcal{K}$  について、 $I \models F$ ); 知識ベース  $\mathcal{K}$  は充足可能 (充足不能) iff  $\mathcal{K}$  はモデルを持つ (持たない); 公理  $F$  が知識ベース  $\mathcal{K}$  の論理的帰結である (つまり  $\mathcal{K} \models F$ ) iff 全ての  $\mathcal{K}$  のモデルが  $F$  を充足する。

### 2.3 記述論理 $ALCHOQ(\sqcup, \sqcap)$

DL  $ALCHOQ(\sqcup, \sqcap)$  は、 $SHOIN(D)$  と類似の構文を持つ。 $ALCHOQ(\sqcup, \sqcap)$  の概念の構文を以下に示す。

$$C, C' \rightarrow A \mid \neg C \mid C \sqcap C' \mid C \sqcup C' \mid \{o_1, \dots\}$$

$$\forall R.C \mid \exists R.C \mid \geq nR.C \mid \leq nR.C,$$

【例 2】 Male, Female, Human を原子的概念, hasChild をロールとすると,  $\text{Human} \sqcap (\leq 1 \text{ hasChild.Male})$  は高々1人の息子をもつ人,  $\text{Human} \sqcap (\geq 3 \text{ hasChild.Female})$  は3人以上の娘をもつ人, という概念を表す.

$\mathcal{ALCHOQ}(\sqcup, \sqcap)$  の意味論は,  $\mathcal{SHOIN}(\text{D})$  と同様に与えられる. DL-safe ルールが付与された  $\mathcal{ALCHOQ}(\sqcup, \sqcap)$  の意味論を以下に示す [Heymans 05]. 知識ベース (但し, 概念包含とロール包含の公理からなる Tbox) を  $\Sigma$ ,  $\Sigma$  に出現する概念名とロール名の集合をそれぞれ  $N_C, N_R$  とし,  $N_P$  を  $N_C \cup N_R \subseteq N_P$  なる述語記号の集合とする. DL 原子式は,  $C(s)$ , または  $R(s, t)$  (但し,  $C \in N_C, R \in N_R$ ) の形の原子式, DL-safe ルールは,  $a \leftarrow b_1, \dots, b_n$  (但し,  $a, b_i$  は原子式) の形で, ルールに出現する全ての変数は必ずルールの本体の非 DL 原子式に出現するルールである. DL-safe プログラムは, DL-safe ルールの有限集合である.  $\text{cts}(\Sigma, P)$  を  $\Sigma$  の個体と  $P$  の定数の集合とする.

$\mathcal{ALCHOQ}(\sqcup, \sqcap)$  の知識ベース  $\Sigma$  と DL-safe プログラム  $P$  を統合した  $(\Sigma, P)$  の意味論は,  $\Sigma$  を一階理論  $\pi(\Sigma)$  (e.g. [Borgida 96]) で表現し,  $P$  の各ルール  $a \leftarrow b_1, \dots, b_n$  を節  $a \vee \neg b_1 \vee \dots \vee \neg b_n$  とした  $\pi(\Sigma) \cup P$  の一階理論の解釈により与えられる. 知識ベース (i.e. Tbox)  $\Sigma$  と DL-safe プログラム  $P$  の統合知識ベース  $(\Sigma, P)$  に関する質問応答 (query-answering) は, 基礎原子式  $\alpha$  が  $\pi(\Sigma) \cup P$  の全てのモデルで真である (以後,  $(\Sigma, P) \models \alpha$  で表す) か否かを判定することである.

#### 2.4 概念論理プログラム CLP と開解集合意味論

概念論理プログラム (Conceptual Logic Programs; 略して, CLP) について説明する [Heymans 04]. 原子式は, 単項  $q(s)$ , 又は 2 項  $f(s, t)$  (但し,  $q, f$  は述語記号,  $s, t$  は項) のものとし, リテラルは原子式  $A$ , 又は,  $\neg A$  (但し,  $\neg$  は論理否定), NAF リテラルは  $\text{not}L$  (但し,  $L$  はリテラル,  $\text{not}$  は negation as failure オペレータ), 拡張リテラルはリテラル 或は NAF リテラルである. 拡張リテラル集合  $\alpha$  について,  $\alpha^+, \alpha^-$  を  $\alpha^+ = \{\ell \mid \ell \in \alpha\}$ ,  $\alpha^- = \{\ell \mid \text{not } \ell \in \alpha\}$  で定義する. 選言論理プログラム DLP は,  $\alpha \leftarrow \beta$  (但し,  $|\alpha^+| \leq 1$ , 即ち頭部  $\alpha$  に高々1個のリテラルが出現) なるルールの集合である. DLP  $P$  に出現する個体定数の集合を  $\mathcal{H}_P$  とし,  $P$  のユニバース  $\mathcal{H}$  は  $\mathcal{H}_P \subseteq \mathcal{H}$  なる集合として定義される.

$P$  のユニバース  $\mathcal{H}$  により基礎化された基礎プログラム  $P_{\mathcal{H}}$  の全てのリテラル集合を  $\mathcal{L}_P$  とすると,  $\mathcal{L}_P$  の無矛盾な部分集合は  $P$  の解釈 (interpretation) である. simple な基礎プログラム  $P$  (即ち,  $P$  に NAF が出現しない) の解集合 (answer sets) は,  $P$  の”集合の包含関係”に関する極小モデルとして定義される.  $P$  が simple でない場合, 任意の解釈  $I$  に関する  $P$  の GL reduct  $P^I$ :

$$P^I = \{\alpha^+ \leftarrow \beta^+ \mid \alpha \leftarrow \beta \in P, \beta^- \cap I = \emptyset, \alpha^- \subseteq I\}$$

について, 基礎  $P$  の解釈  $M$  が  $P^M$  の解集合であるならば,  $M$  は  $P$  の (通常の) 解集合として定義される [Gelfond 91].

非基礎なる DLP  $P$  について, 開解釈 (open interpretation) は, 対  $(\mathcal{H}, I)$  である (但し,  $\mathcal{H}$  は  $P$  のユニバース,  $I$  は  $P_{\mathcal{H}}$  の解釈).  $P$  の開解集合 (open answer set) は,  $M$  が  $P_{\mathcal{H}}$  の解集合であるような開解釈  $(\mathcal{H}, M)$  で定義される. プログラム  $P$  は開解集合を持てば, 無矛盾 (consistent) である.

DLP  $P$  に出現する単項述語記号  $p$  について, ある  $x \in \mathcal{H}$  に対して  $p(x) \in M$  なる  $P$  の開解集合  $(\mathcal{H}, M)$  が存在するならば,  $p$  が  $P$  に関して充足可能 (satisfiable) である.

開解集合意味論 (Open Answer Set Semantics) のもとで充足可能性チェックは非決定性である. 一般に Open Answer Set Programming の推論も非決定性 (undecidable) になるが, forest model property という性質を保持させつつ DLP の構文に制限を加えた local conceptual logic programs (略して, local CLP) のクラスの DLP については, 有限個の匿名定数を導入することにより, 通常の有限ドメインで決定性推論である ASP に縮約可能になるという後述の定理が得られている [Heymans 04].

【定義 4】 CLP (概念論理プログラム) は以下の形式のルールからなる DLP である.

- free rules:  $l \vee \text{not } l \leftarrow$ , (但し,  $l$  はリテラル)
- unary rules:
  - $a(s) \leftarrow \alpha(s), \cup_m \gamma_m(s, t_m), \cup_m \beta_m(t_m), t_i \neq t_j$ ,
  - 但し,  $\gamma_m \neq \emptyset$  ならば  $\gamma_m^+ \neq \emptyset$  である. また  $t_m$  が変数の場合,  $\beta_m \neq \emptyset$  ならば  $\gamma_m \neq \emptyset$  である.
- binary rules:  $f(s, t) \leftarrow \alpha(s), \gamma(s, t), \beta(t)$ ,
- 但し,  $t$  が変数ならば,  $\gamma^+ \neq \emptyset$
- constraints:  $\leftarrow a(s)$ .

特に, unary と binary rules が次の locality の条件:

「 $\forall b \in \beta_{(m)}^+$  について  $b(t_{(m)}) \vee \text{not } b(t_{(m)}) \leftarrow \in P$ , 或はルール  $r: b(s) \leftarrow \text{body}(r)$  の全てについて  $\text{body}(r)^+ = \emptyset$ 」を満たす CLP を, local CLP (local な概念論理プログラム) という.

CLP は forest model property を持つ. つまり CLP  $P$  の開解集合  $(\mathcal{H}, M)$  の  $M$  は,  $M$  に出現する個体定数を節 (ノード), 節のラベルは単項述語の集合, 節から節への有向辺 (枝) のラベルには 2 項述語が付与された高々  $(|\mathcal{H}_P| + 1)$  個の (無限木も可能な) 木の集合 (forest, 森) で表現されることが証明されている (文献 [Heymans 04] の Theorem 1). これに上記の locality の条件が加わることにより, local CLP  $P$  の充足可能性判定に関して, 無限木の節の重ね合わせによる枝刈りが適用された有限節数の有限木集合 (森) で  $M$  が縮約表現可能になり, 以下の定理が得られる.

【定理 1】 [Heymans 04]  $P$  を local CLP とする.  $P$

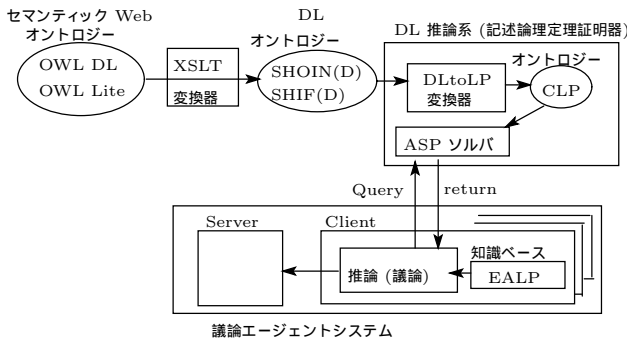


図 1 システム構成

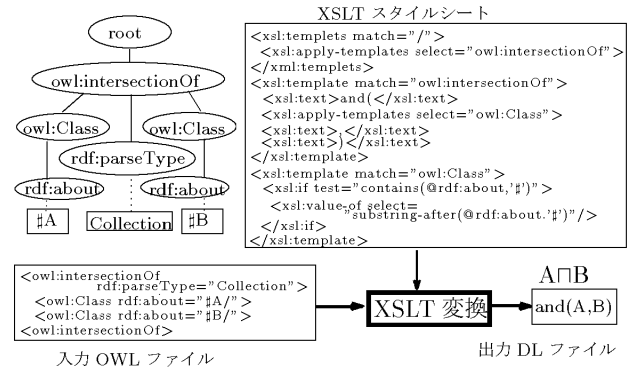


図 2 conjunction の変換例

に出現する単項述語  $p$  が  $P$  に関して充足可能 (*satisfiable*) であるならば、そしてその時に限り、 $p(x_i)$  ( $1 \leq i \leq k_P$ ) を含む  $\psi(P)$  の解集合  $M$  が存在する。

但し、 $\psi(P) = P \cup \{cte(x_i) \leftarrow |1 \leq i \leq k_P\}$ . (1)

ここで  $cte$  は匿名定数  $x_i$  を導入するための  $P$  に出現しない述語記号、 $k_P$  は  $P$  により定まる正の整数である。

基礎リテラル  $\alpha$  について、DLP  $P$  の全ての開解集合  $(\mathcal{H}, M)$  に対して  $\alpha \in M$  ならば、 $P \models \alpha$  で表す。  $P \models \alpha$  を質問応答 (query answering) と称する。質問応答は、次式の矛盾性判定の問題に縮約される。

$$P \models \alpha \text{ iff } P \cup \{not \alpha \leftarrow\} \text{ is inconsistent. } \quad (2)$$

### 3. システム概要

マルチエージェントの共有知識として、セマンティック Web 上に、オントロジーが OWL DL で表現され、他方、各議論エージェントは、他エージェントと議論、論争、交渉するための各自の戦略的知識を、EALP (Extended Annotated Logic Programs, 拡張注釈付き論理プログラム) [Takahashi 04] で表現された知識ベースを持つものとする。本研究では、他エージェントとの議論時にオントロジーに関する参照・推論が必要になると、随時セマンティック Web 推論系を介して Web 上のオントロジー知識に関する単調推論を行ないつつ、非単調な対話的証明論に基づく議論を可能とする 2 つの推論システムの推論統合の方法を提案する。図 1 にシステム構成図を示す。本システムは、

- (1) OWL DL から SHOIN(D) へのオントロジー表現変換器、
  - (2) Heymans 法に基づく記述論理定理証明器 (DL 推論系)、
  - (3) 議論エージェントシステム ([Takahashi 04])、
- で構成される [Sawamura 06, Wakaki 06]。

#### 3.1 OWL DL/ SHOIN(D) オントロジー表現変換器

セマンティック Web 上に OWL DL で表現されたオントロジーは、変換規則がテンプレートルールに記述された XSLT のオントロジー表現変換器により、意味論的に同値な記述論理: DL SHOIN(D) に変換される ([Horrocks

```
<rdf:RDF>
  <owl:Class rdf:ID="university_curriculum">
    <owl:equivalentClass rdf:resource="#course"/>
  </owl:Class>
  <owl:Class rdf:ID="course">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#faculty_e"/>
      <owl:Class rdf:about="#faculty_s"/>
      <owl:Class rdf:about="#faculty_h"/>
    </owl:unionOf>
  </owl:Class>
</rdf:RDF>
```

図 3 OWL によるカリキュラム・オントロジー記述

03])。図 2 に、“A かつ B” という概念の OWL DL 表現から DL SHOIN(D) 表現:  $A \sqcap B$  への XSLT による変換の例を図 2 に示す。

この図で左に示したものが変換元の OWL DL とその XML 木構造表現、右に示したものが変換後の記述論理 (DL SHOIN(D)) 表現とそのファイル表現、真ん中に、XSLT で用いるテンプレートルール (変換規則) が記述されたスタイルシートが示されている。図 3 に OWL DL で記述されたカリキュラム・オントロジー (一部) の例を示す。

図 3 の OWL DL で記述されたオントロジーは以下の記述論理 SHOIN(D) の論理式に表現変換される。変換規則については付録を参照されたい。

$$\begin{aligned} university\_curriculum &\equiv course \\ course &\equiv faculty\_e \sqcup faculty\_s \sqcup faculty\_h \end{aligned}$$

なお  $\equiv$  は、概念  $C, D$  に対して以下で定義される。

$$(C \equiv D) \stackrel{def}{=} (C \sqsubseteq D) \text{ かつ } (D \sqsubseteq C)$$

本研究で実装したオントロジー表現変換器は “OWL DL から DL SHOIN(D) ”と “DL SHOIN(D) から OWL DL ”への双方向変換が可能であり、これらの変換も本研究の成果の一つである (文献 [宮下 06] を参照)。

【例 3】 OWL DL で記述した大学のカリキュラム・オントロジーを付録に示す。この OWL DL 表現のカリキュラム・オントロジーをオントロジー表現変換器に入力す

<p>&lt;TBox&gt;</p> <pre> uni_curriculum ≡ course course ≡ faculty_e ⊔ faculty_s ⊔ faculty_h cs_dept ⊆ faculty_e ee_dept ⊆ faculty_e math_dept ⊆ faculty_s philo_dept ⊆ faculty_h we_logic ≡ western_logic ⊔ eastern_logic we_logic ⊆ logic mp_logic ≡ math_logic ⊔ philo_logic mp_logic ⊆ logic easy_get_credit ≡ ∃reg.(¬pass) hard_get_credit ≡ ∀reg.pass </pre>
<p>&lt;ABox&gt;</p> <pre> hard_get_credit(philosophy) reg(ai, st_0) cs_dept(l1) math_dept(l2) philo_dept(l3) cs_dept(ai) cs_dept(c-programming) cs_dept(prolog-programming) cs_dept(technical_english) math_logic(l2) philo_logic(l1) ¬pass(st_0) </pre>

図 4 カリキュラム・オントロジー:  $\mathcal{K}_1$ 

ると DL *SHOIN*(D) 表現のカリキュラム・オントロジー:  $\mathcal{K}_1$  (図 4) が生成される。本オントロジー表現変換器は構造エディター機能を有し、図 4 の DL *SHOIN*(D) 表現を入力して、OWL DL 表現のカリキュラム・オントロジーを生成することも可能である (付録参照)。

### 3.2 記述論理の推論系

DL *SHOIN*(D) 表現に変換されたオントロジーの推論については、S. Heymans *et al* [Heymans 04, Heymans 05] らにより提案された Answer Set Programming (以後 ASP) [Gelfond 91] で記述論理推論 (以後、DL 推論と称する) をシミュレートする方法 (以後、Heymans 法と称する) を採用した。

Heymans 法では、*SHOIN*(D) と類似の *ALCHOQ* ( $\sqcup, \sqcap$ ) を対象としている。両者の相異点として、データタイプ、データタイプルール、逆ルール、ルール推移律が *SHOIN*(D) に存在するが *ALCHOQ* ( $\sqcup, \sqcap$ ) に無く、他方、制限付き数量限定は *ALCHOQ* ( $\sqcup, \sqcap$ ) に存在するが *SHOIN*(D) に無く、代わりに制限なし数量限定が *SHOIN*(D) に存在する。本研究では OWL DL、即ち、*SHOIN*(D) 表現のオントロジーを扱うため、

- [1] データタイプ、データタイプルールはサポートしない。
- [2] ロール  $R \in \mathbf{R}$  の逆ルール  $R^-$  については、

$$R^-(Y, X) \leftarrow R(X, Y), \quad (3)$$

$$R(Y, X) \leftarrow R^-(X, Y), \quad (4)$$

のルールを生成する。

- [3] ロール推移律: *Trans*( $R$ ) の公理については、

$$R(X, Z) \leftarrow R(X, Y), R(Y, Z), \quad (5)$$

のルールを生成する。DL-safe ルールとして処理する。

- [4] 制限なし数量限定:  $\geq nR, \leq nR$  については、制限付き数量限定:  $\geq nR.\top, \leq nR.\top$  として計算する。但し、任意の概念  $C$  について、 $\top = C \sqcup \neg C$ 。

*ALCHOQ* ( $\sqcup, \sqcap$ ) で記述されたオントロジー知識は、CLP (*Conceptual Logic Programs*) と称される選言論理プログラム (DLP) のクラスに変換される。無限ドメインの DLP では一般に非決定性推論となるが、Heymans 法では、*ALCHOQ* ( $\sqcup, \sqcap$ ) のオントロジー (Tbox) を構文的に制限された DLP (即ち、local CLP) に変換し、有限個の匿名定数の導入と単一名公理を仮定して、DL 推論が有限ドメインにおける DLP の (通常の) ASP 推論に縮約可能となるという以下の定理が得られている。下記の定理で、Tbox の公理集合は  $\Sigma$ 、Abox の公理は基礎原子式、つまり  $P$  の head のみのルールとして表現され、 $(\Sigma, P) \models \alpha$  は、質問  $\alpha$  が一階理論  $\pi(\Sigma) \cup P$  の論理的帰結である事を表す。

[定理 2] [Heymans 05] *ALCHOQ* ( $\sqcup, \sqcap$ ) の知識ベース (Tbox) を  $\Sigma$ 、DL-safe プログラムを  $P$  とする。基礎原子式  $\alpha$  に対して、次式が成り立つ。

$$(\Sigma, P) \models \alpha \text{ iff } \Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P) \models \alpha. \quad (6)$$

但し、

1.  $\Phi_1(\Sigma, P)$  は、以下の CLP のルールで構成される。

- (i)  $C_1 \sqsubseteq C_2, R_1 \sqsubseteq R_2 \in \Sigma$  について、以下の一貫性制約を生成。

$$\leftarrow C_1(X), \text{not } C_2(X),$$

$$\leftarrow R_1(X, Y), \text{not } R_2(X, Y).$$

- (ii) a. (i) の body 部に出現する概念  $C_1, C_2$  が原子的概念でなければ、その概念を head とする CLP のルールを表 1 に従い生成。

- b. a. で生成したルールの body 部に出現する概念が原子的概念でなければ、その概念を head とする CLP のルールを表 1 に従い生成。body 部が全て原子的概念になるまで、これを繰り返す。

- (iii)  $\Sigma$  に出現する原子的概念  $A$  とロール  $R$  について、以下のルール (i.e. free rules) を生成。

$$A(X) \vee \text{not } A(X) \leftarrow,$$

$$R(X, Y) \vee \text{not } R(X, Y) \leftarrow.$$

2.  $\Phi_2(\Sigma, P)$  は、以下の基礎 DLP のルール集合として構成される。ここで  $cts(\Sigma, P)$  を  $\Sigma$  の個体と  $P$  の個体定数からなる集合とする。

- (i)  $P$  の各ルールを  $cts(\Sigma, P)$  の定数で基礎化した基礎ルールを構成。以後、この基礎ルールの集合を基礎  $P_{cts(\Sigma, P)}$  で表す。

(ii)  $P_{cts(\Sigma, P)}$  の各ルール  $r$  毎に次のルール：

$$head(r) \vee not\ head(r) \leftarrow,$$

を構成．但し  $head(r)$  はルール  $r$  の  $head$  のリテラルを表す．

表 1 は，DL  $ALCHQ(\sqcup, \sqcap)$  の概念と Heymans 法による概念の CLP 表現，及び，DL 推論系におけるそれらのファイル出力仕様を示す．

(6) 式右辺： $\Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P) \models \alpha$ において， $\Phi_1(\Sigma, P)$  は構文的に local CLP の条件を満たさないが充足可能性を保持する local CLP のプログラムに変換できる [Heymans 04]．他方，local CLP と基礎 DLP の和集合として定義される local ECLP の論理プログラム ([Heymans 05]) についても，local CLP  $P$  に関する定理 1 の (1) 式が  $P$  を local ECLP に拡張しても同様に成りたつことが文献 [Heymans 05] の Theorem 3 で示されている．従って， $\Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P)$  は local ECLP と見做され， $\Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P) \models \alpha$  が推論されるか否か？ (即ち，質問応答の (2) 式) は，local ECLP に対する有限ユニバースにおける充足可能性判定の (1) 式に基づき，有限ユニバースでの充足不能性判定，つまり  $\Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P) \cup \{cte(x_i) \leftarrow\} \cup \{not\ \alpha \leftarrow\}$  (7) が矛盾か否か？を判定して計算する．我々の場合， $\Phi_1(\Sigma, P) \cup \Phi_2(\Sigma, P)$  に (3),(4),(5) のルールが追加される．(7) 式は，匿名定数  $x_i$  ( $1 \leq i \leq k$ ) が導入された論理否定 (classical negation) が出現しない通常の ASP の論理プログラムであり，そのユニバースは個体定数の有限集合：

$$cts(\Sigma, P) \cup \{x_i \mid cte(x_i) \leftarrow\ for\ 1 \leq i \leq k\}$$

となる．従って ASP ソルバーを用いて，容易に (7) 式の矛盾性判定は計算できる．本研究の DL 推論系は，ASP ソルバー DLV[Eiter 97] と C++ を用いて実装し，(7) 式を評価している．

〔例 4〕オントロジー  $\mathcal{K}_1$  (図 4) の Tbox の公理，

$$easy\_get\_credit \equiv \exists reg. (\neg pass),$$

について定理 2 の 1(i) より 2 個の一貫性制約が生成され，

$$\leftarrow easy\_get\_credit(X), not\ \{\exists reg. (\neg pass)\}(X),$$

$$\leftarrow \{\exists reg. (\neg pass)\}(X), not\ easy\_get\_credit(X).$$

(ii) では表 1 の変換に従い，次のルールが生成される．

$$\{\exists reg. (\neg pass)\}(X) \leftarrow reg(X, Y), \{\neg pass\}(X),$$

$$\{\neg pass\}(X) \leftarrow not\ pass(X).$$

また 1 の (iii) で，次の 3 個の free rules が生成される．

$$easy\_get\_credit(X) \vee not\ easy\_get\_credit(X) \leftarrow,$$

$$pass(X) \vee not\ pass(X) \leftarrow,$$

$$reg(X, Y) \vee not\ reg(X, Y) \leftarrow.$$

他方，Abox (図 4) の以下の公理，

$$cs\_dept(ai), \quad \neg pass(st\_0).$$

については，定理 2 の 2(i) で，

$$cs\_dept(ai) \leftarrow, \quad \{\neg pass\}(st\_0) \leftarrow.$$

のルールが生成される．2(ii) では，

$$cs\_dept(ai) \vee not\ cs\_dept(ai) \leftarrow,$$

$$\{\neg pass\}(st\_0) \vee not\ \{\neg pass\}(st\_0) \leftarrow.$$

が生成される．上例のように range-restricted でない CLP 表現のルールが存在する場合，我々の DLtoLP 変換器では range-restricted な形式\*1 にしてルールを出力する (表 1 参照)．

### 3.3 議論エージェントシステム

議論エージェントシステムとは，議論の制御を行う仲介エージェントと，議論するエージェントで構成されるマルチエージェントシステムである．議論するエージェントは EALP (Extended Annotated Logic Programs, 拡張注釈付き論理プログラム) [Takahashi 04] のルール集合で表現された知識ベースと，LMA (a Logic of Multiple-valued Argumentation, 多値議論の論理) という推論エンジンをもち，論証の交換によって議論し，対話的証明論によって我々人間の行う議論をシミュレートするものである．さて，本研究の議論エージェントは，DL 推論系との推論インターフェースを持つ．ここで  $SHOIN(D)$  等で表現されたオントロジー知識ベース  $\mathcal{K}$  が無矛盾と想定し， $\mathcal{K}$  の (任意の) 概念の集合， $\mathcal{K}$  に出現するロールの集合，個体の集合をそれぞれ  $C_{\mathcal{K}}$ ,  $R_{\mathcal{K}}$ ,  $cts(\mathcal{K})$  として， $Lit_{\mathcal{K}}$  を以下に定義する．

$$Lit_{\mathcal{K}} \stackrel{def}{=} \{C(a) \mid C \in C_{\mathcal{K}}, a \in cts(\mathcal{K})\}$$

$$\cup \{R(a, b) \mid R \in R_{\mathcal{K}}, a, b \in cts(\mathcal{K})\}$$

以後， $Lit_{\mathcal{K}}$  の要素を基礎 DL アトム (略して，DL アトム) と称する．例えばオントロジー  $\mathcal{K}_1$  (図 4) において， $\neg pass \in C_{\mathcal{K}_1}$ ,  $st\_0 \in cts(\mathcal{K}_1)$  であるから， $\neg pass(st\_0)$  は  $Lit_{\mathcal{K}_1}$  の要素である．

各議論エージェント  $A_i$  ( $1 \leq i \leq n$ ) の各自の知識ベースを  $KB_i$  とすると，本研究では，各議論エージェント  $A_i$  は，ideals-based semantics [Kifer 92] のモデル論をもつ完備束  $\mathcal{T} = (\{t, f\}, \leq)$  (但し， $f \leq t$ ) 上の EALP で表現された以下の知識ベース  $P_i$  を仮想的に持つと考える．

$$P_i = KB_i \cup \{l : t \leftarrow \mid \mathcal{K} \models l\ for\ l \in Lit_{\mathcal{K}}\} \quad (8)$$

ここで  $t$  は注釈， $l$  は  $C(a)$  又は  $R(a, b)$  なる DL アトムであるが，EALP のルール  $l : t \leftarrow$ ，つまり  $C(a) : t \leftarrow$  又は  $R(a, b) : t \leftarrow$  における  $C$  や  $R$  は，それぞれ概念  $C$  やロール  $R$  を表現する一つの述語記号を表す．例えば  $\mathcal{K}_1$  の概念  $\neg pass$  については， $\neg pass(st\_0) : t \leftarrow$  における  $\neg pass$  は例 4 と同様に一つの述語記号である．ここで記述論理の補概念を表すための記号  $\neg$  は，述語記号  $\neg pass$  を構成する記号となっていることに注意されたい．

$KB_i$  は以下の形式のルール：

$$H \leftarrow L_1, \dots, L_k, not\ L_{k+1}, \dots, not\ L_m$$

の集合である． $H$ ,  $L_j$  は注釈付きリテラルで， $H$  をルールの結論， $L_1 \dots L_k$  をルールの前提， $L_{k+1} \dots L_m$  をルールの仮定という．ルールの結論には DL アトムが出現しないが，本体の  $L_j$  ( $1 \leq i \leq m$ ) に，注釈付き DL アト

\*1 DLP のルール  $r: \alpha \leftarrow \beta$  (但し， $|\alpha^-| = \emptyset$ ) について， $r$  に出現する任意の変数が  $r$  の本体の  $\beta^+$  のリテラルであるとき，そのような  $r$  を「range-restricted な形式」のルールという．

表 1 Heymans 法における  $\Phi_1(\Sigma, P)$  への CLP 変換

DL 表現	CLP 表現	DLtoLP 変換器の出力形式
$\neg C$	$\neg C(X) \leftarrow \text{not } C(X)$	$n\_c\_d(X) : - \text{dom}_d(X), \text{not } c_d(X).$
$C_1 \sqcap C_2$	$C_1 \sqcap C_2(X) \leftarrow C_1(X), C_2(X)$	$c1\_and\_c2\_d(X) : - c1\_d(X), c2\_d(X).$
$C_1 \sqcup C_2$	$C_1 \sqcup C_2(X) \leftarrow C_1(X) \leftarrow C_2(X)$	$c1\_or\_c2\_d(X) : - c1\_d(X). \quad c1\_or\_c2\_d(X) : - c2\_d(X).$
$\exists R.C$	$\exists R.C(X) \leftarrow R(X, Y), C(Y)$	$\text{exists}_r\_c\_d(X) : - r_d(X, Y), c_d(Y).$
$\forall R.C$	$\forall R.C(X) \leftarrow \text{not } \exists R.\neg C(X)$	$\text{all}_r\_c\_d(X) : - \text{dom}_d(X), \text{not } \text{exists}_r\_n\_c\_d(X). \quad n\_c\_d(X) : - \text{dom}_d(X), \text{not } c_d(X).$
$\leq n R.C$	$\leq n R.C(X) \leftarrow \text{not } (\geq n+1 R.C)(X)$	$\text{atmost}_n\_r\_c\_d(X) : - \text{dom}_d(X), \text{not } \text{atleast}_n\_r\_c\_d(X). \text{ (但し } m=n+1)$
$\geq n R.C$	$\geq n R.C(X) \leftarrow R(X, Y_1), \dots, R(X, Y_n), C(Y_1), \dots, C(Y_n), Y_1 \neq Y_2, \dots$	$\text{atleast}_n\_r\_c\_d(X) : - r_d(X, Y_1), \dots, r_d(X, Y_n), c_d(Y_1), \dots, c_d(Y_n), Y_1 \neq Y_2, \dots$

ムが出現可能である．なお本システムでは実装上の都合から，DL アトム  $\alpha$  に注釈  $t$  が付与された注釈付き DL アトム  $\alpha:t$  を，DL 式と称する  $DL[\alpha]$  で表現する．まず表 2 を用いて知識表現と議論の概要を述べ，その後議論エージェントシステムについて述べる．

### §1 知識表現

表 2 は，Lecturer エージェントと Student エージェントの知識ベースを表す．EALP は次のような形の規則の集合である．但し，ここでは本体のリテラルの区切り記号”,” の代わりに  $\&$  を用いる．

$\sim \text{take\_credit\_in}(ai, st\_1, second):t \leftarrow$

$DL[\text{course}](ai) \& DL[\text{logic}](l1) \& \text{not } \text{study}(st\_1, l1, first):t$   
 本論文が対象とするオントロジーおよび記述論理は 2 値であるため，本論文では真理値としての注釈を 2 値  $t, f$  に特殊化した EALP を用いる． $\therefore$  の右隣りにある  $t$  が注釈であり  $\text{take\_credit\_in}(ai, st\_1, second)$  について真であるという認識が存在していることを表している． $\sim$  は存在論的な否定（つまり ASP の論理否定  $\neg$  [Gelfond 91]）であり， $\sim \text{take\_credit\_in}(ai, st\_1, second):t$  は，「生徒  $st\_1$  が第二期に AI の単位を取ることは認められない」と読む． $\text{not } \text{study}(st\_1, l1, first):t$  は「第一期に生徒  $st\_1$  が  $l1$  という単位を取ったという事実が今のところ存在しない」ことを意味する．全体的には「専攻が AI であり，第一期に  $l1$  の単位を取っていない生徒は第二期に AI の単位を取ることは認められない」と読む．上記の  $DL[\text{course}](ai)$ ， $DL[\text{logic}](l1)$  は DL 式であり，前述のように例えば  $DL[\text{course}](ai)$  は，DL アトム  $\text{course}(ai)$  に注釈  $t$  が付いた注釈付き DL アトム  $\text{course}(ai):t$  を表す．

### §2 マルチエージェントの議論フレームワーク

論証 ( $Arg$ ) とは規則の有限列である．議論エージェント  $A_i$  の (仮想的) 知識ベース  $P_i = KB_i \cup \{l : t \leftarrow | \mathcal{K} \models l\}$  の全ての論証集合を  $Args_{P_i}$  で表すと，マルチ議論エージェントの知識ベース集合  $MAS = \{P_1, \dots, P_n\}$  の論証集合  $Args_{MAS}$  は

$$Args_{MAS} = Args_{P_1} \cup \dots \cup Args_{P_n} \quad (9)$$

で定義する [Takahashi 04]．例えば表 2 の Lecturer エー

ジェントの知識ベース  $KB_{Lecturer}$  の最初のルールを含む論証の場合，DL アトム  $\text{course}(ai)$ ， $\text{logic}(l1)$  がいずれもカリキュラムオントロジー  $\mathcal{K}_1$  の論理的帰結，

$$\mathcal{K}_1 \models \text{course}(ai), \quad \mathcal{K}_1 \models \text{logic}(l1) \quad (10)$$

ならば，(8) 式に対応して動的に

$$\text{course}(ai):t \leftarrow, \quad \text{logic}(l1):t \leftarrow$$

が生成され，以下の論証  $Arg \in Args_{P_{Lecturer}}$  :

$$Arg = [\sim \text{take\_credit\_in}(ai, st\_1, second):t \leftarrow \text{course}(ai):t \& \text{logic}(l1):t \& \text{not } \text{study}(st\_1, l1, first):t, \text{course}(ai):t \leftarrow, \text{logic}(l1):t \leftarrow]$$

が作成される．これは生徒  $st\_1$  が AI の単位を取れないことを主張するための論証である． $Arg$  を構成する規則の結論を  $Arg$  の結論と呼び，その仮定を  $Arg$  の仮定と呼ぶ． $Arg$  の結論の集合を  $\text{concl}(Arg)$ ，仮定の集合を  $\text{assm}(Arg)$  と書く．

なお (10) 式の評価は， $\text{course}(ai)$ ， $\text{logic}(l1)$  のそれぞれを質問  $\alpha$  として，(6) 式の ASP 論理プログラムの矛盾性判定により計算される．その際，(6) 式の  $\Phi_1(\Sigma, P)$  はカリキュラムオントロジー  $\mathcal{K}_1$  の Tbox (図 4) を  $\Sigma$  と見做して定理 2 の 1 に従って生成されたルール集合， $\Phi_2(\Sigma, P)$  は  $\mathcal{K}_1$  の Abox を  $P$  と見做して定理 2 の 2 に従って生成されたルール集合である．

### §3 論証の正当化：対話的証明論と不動点意味論

DL アトムが出現する (9) 式の論証集合  $Args_{MAS}$  が与えられると，文献 [Takahashi 04] と同様にして，正当化された論証や却下される論証は，対話木を構築する対話的証明論により計算可能である．他方，対話的証明論で証明される正当化された論証集合は，単調関数  $F_{Args_{MAS}}$  の最小不動点  $\text{lfp}(F_{Args_{MAS}})$  として意味論が与えられる．本研究では， $Args_{MAS}$  にアトムが出現し，論証に含まれる EALP のルールに DL アトムが出現すると，前述の例のように，動的に当該 DL アトム  $\alpha$  が DL 推論系の論理的帰結 ( $\mathcal{K} \models \alpha$ ) か否かを判定し，そうであるならば  $\alpha:t \leftarrow$  の知識を用いて動的に論証を作成する．従って対話木構築アルゴリズムで動的に DL 推論系の呼出しが発生する点が先行研究 [Takahashi 04] と異なる．紙面の都合上，(8) 式のように静的に全ての DL 推論系の論理的

帰結が得られていると想定し、議論と対話木を以下に説明する [Takahashi 04] .

議論とは、次の条件を満たす提議  $move_i = (Player_i, Arg_i)$  の空でない有限列である。但し、 $Arg_i \in Arg_{SMAS}, i \geq 1$  .

1.  $Player_i =$  提案者  $\iff i$  は奇数である;  
 また  $Player_i =$  反対者  $\iff i$  は偶数である .
2.  $Player_i = Player_j (i \neq j)$  ならば  $Arg_i \neq Arg_j$  .
3.  $(Arg_i, Arg_{i-1}) \in$  打破 .

条件 1 は、議論が提案者から始まり、提案者と反対者が交互に提議することを表している。条件 2 は議論が循環することを防いでいる。条件 3 は議論で用いる攻撃関係である。議論には、典型的な攻撃関係である反論、無効化以外にも様々な攻撃関係を考えることができる [Takahashi 04] . 反論と無効化は多値議論では以下のように定義される .

- $Arg_\alpha$  が  $Arg_\beta$  を反論する  $\iff \mu_1 \geq \mu_2$  である  $A: \mu_1 \in concl(Arg_\alpha)$  と  $\sim A: \mu_2 \in concl(Arg_\beta)$  が存在するか、 $\mu_1 \leq \mu_2$  である  $\sim A: \mu_1 \in concl(Arg_\alpha)$  と  $A: \mu_2 \in concl(Arg_\beta)$  が存在する .
- $Arg_1$  が  $Arg_2$  を無効化する  $\iff \mu_1 \geq \mu_2$  である  $A: \mu_1 \in concl(Arg_1)$  と  $\text{not } A: \mu_2 \in assem(Arg_2)$  が存在するか、 $\mu_1 \leq \mu_2$  である  $\sim A: \mu_1 \in concl(Arg_1)$  と  $\text{not } \sim A: \mu_2 \in assem(Arg_2)$  が存在する .

本論文ではもっとも直観的に理解しやすいため提案者、反対者ともに同じ攻撃関係、打破 [Takahashi 04] を用いる。多値議論では打破は次のように定義される。  $Arg_1$  が  $Arg_2$  を打破する  $\iff Arg_1$  が  $Arg_2$  を無効化するか、  $Arg_1$  が  $Arg_2$  を反論し、  $Arg_2$  が  $Arg_1$  を無効化しない。議論エージェントシステムは、論証の集合から正当化された論証を対話的に決定する。正当化の理解を容易にするために議論木 (an argumentation tree) を導入する。議論木とは、すべての枝 (branch) が議論となる木である。ただし、 $i$  が奇数、つまり  $Player_i$  が提案者の時、提議  $move_i = (Player_i, Arg_i)$  の子は  $Arg_i$  を打破するすべての (反対者の) 論証である。論証 Arg を根とする対話木のすべての枝の終端が提案者であったとき論証 Arg は証明論的に正当化されたという。図 5 は、生徒  $st\_1$  が第二期において AI の単位をとれるかどうかという議論を実際に展開し、議論木で表現したものである。図の中のそれぞれのフレームは論証を表す。defeat は打破の攻撃関係を表す。この議論ではすべての枝の終端が提案者なので根である論証  $Arg_1$  は正当化され、生徒  $st\_1$  は第二期において AI の単位の取得を認められる。

〔例 5〕 単位取得に関する教員と学生の議論  
 カリキュラム・オントロジー (図 4) を参照する教員と学生の各エージェントのルール知識を表 2 に、DL 推論系を介してカリキュラム・オントロジーを参照しながら議論するエージェントシステムの実行例を図 6 に示す。図の対話木より議題  $take\_credit\_in(ai, st\_1, second)$  が正当化されていることがわかる。

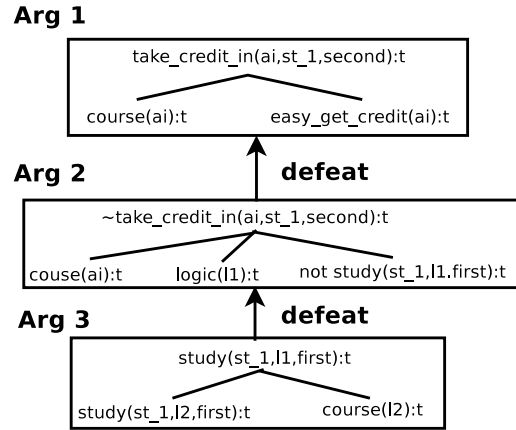


図 5 議論木

表 2 オントロジー  $\mathcal{K}_1$  を参照するエージェントの知識

$KB_{Lecturer} = \{$ $\sim take\_credit\_in(ai, st\_1, second):t \leftarrow$ $DL[course](ai) \& DL[logic](l1)$ $\& \text{not } study(st\_1, l1, first):t. \}$
$KB_{Student} = \{$ $take\_credit\_in(ai, st\_1, second):t \leftarrow$ $DL[course](ai) \& DL[easy\_get\_credit](ai).$ $study(st\_1, l1, first):t.$ $\leftarrow study(st\_1, l2, first):t \& DL[course](l2).$ $study(st\_1, l2, first):t.$ $study(st\_1, c\_programming, first):t. \}$

#### 4. おわりに

オントロジー知識を扱う DL 推論系の開発、および議論エージェントシステムとの推論統合システムを実現し、昨年 5 月に函館で開催された AAMAS 2006 国際会議 (The 5th International Joint Conference on Autonomous Agents and MultiAgent Systems) にて、本システムのデモ発表と実演を成功裏に終えた [Wakaki 06] .

本論文については、議論エージェントシステムのエージェントの知識は EALP のルール集合を想定しているが、エージェントが ELP (Extended Logic Programs, 拡張論理プログラム) の知識ベースを持つ議論エージェントシステムも既に開発し稼働している [野沢 07] . 後者の ELP 版議論エージェントシステムでは、本論文で述べた ASP ソルバ - を用いる DL 推論系と、ツ - ル RACER [Haarslev01] を用いて実装されたセマンティック Web 推論系の、両方のインターフェースを介してオントロジー知識に関する推論が可能である。本研究で実現した Heymans 法に基づく DL 推論系では、local CLP のル - ル集合に非単調なル - ルを追加して、オントロジーを参照する非単調推論も可能である [Heymans 05] . 他方、RACER を用いた推論系では、セマンティック Web 上の OWL DL で記述されたオントロジーを直接入力して効率的にオン



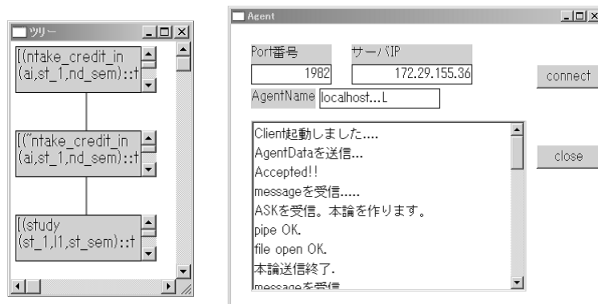


図 6 実験 議題 "take\_credit\_in(ai, st\_1, second)::t" の実行画面

トログ - 推論が可能である。従って対象ドメインのオントロジー知識がそれほど多くなく、かつオントロジーと非単調ル - ルによる推論結果がエージェントの議論で必要となるような場合に本論文で述べた DL 推論系が、他方、必要とするセマンティック Web 上のオントロジー知識がかなりの規模の場合、より効率的にオントロジー推論を行える RACER を用いたインターフェースの方が適しており、いずれの場合も、我々の議論エージェントシステムで対応可能である。

本研究ではカリキュラムオントロジーを用いた小さな例題で、オントロジー推論と議論エージェント推論の統合を目的とした提案手法に関する feasibility study とシステムの検証を行なった。今後、現実の電子商取引や法的推論問題等の具体的事例を用いて、オントロジーに関する DL 推論系と議論エージェント推論の統合に関する有効性の評価を行なう予定である。なお実用システム構築にむけて大規模セマンティック Web 知識ベースの計算時間等の問題は、今後の性能面での技術的課題である。

他方、法的推論等の分野ではルール間のプリファレンス (優先度)、動的プリファレンス、階層的プリファレンス等を用いた議論が必要になるが、これらをも扱うことを可能とする、或は、エージェントの議論による学習等を可能にする新たな議論の枠組の提案、及び、それらに対話的証明論等の手続きに如何に整合的に実現し、より高度の問題解決能力を持つ議論エージェントシステムとして構築するかが、今後の機能上の研究課題である。

## 謝 辞

新潟大学大学院沢村研究室の松永久美子さん (修士課程 2 年) には、議論エージェントシステムに関する技術的支援を頂き、感謝します。本研究は文部科学省科学研究費助成金 (課題番号 17500098) の一環として実施された。

## ◇ 参 考 文 献 ◇

- [Baader 03] F. Baader et al: The Description Logic Handbook, Cambridge University Press (2003)
- [Borgida 96] A. Borgida: On the Relative Expressiveness of Description Logics and Predicate Logics, *Artificial Intelligence* 82, pp.353-367 (1996)
- [Eiter 97] T. Eiter, N. Leone, C. Mateis, G. Pfeifer and F. Scarcello: A deductive system for nonmonotonic rea-

soning, *Proc. of LPNMR 1997, Lecture Notes in Computer Science* 1265, pp. 364-375, Springer (1997) URL <http://www.dbai.tuwien.ac.at/proj/dlv/>

- [Eiter 04] T. Eiter, T. Lukasiewicz, R. Schindlauer and H. Tompits: Combining Answer Set Programming with Description Logics for the Semantic Web, *Proceedings of KR 2004*, pp. 141-151 (2004)
- [Gelfond 91] M. Gelfond, and V. Lifschitz: Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing* 9, pp. 365-385 (1991)
- [Haarslev01] V. Haarslev, R. Moller: RACER System Description, *Proceedings of IJCAR 2001, Lecture Notes in Computer Science* 2083, pp.701-706, Springer (2001) <http://www.racer-systems.com/>
- [Heymans 04] S. Heymans, D. V. Nieuwenborgh, and D. Vermeir: Semantic Web Reasoning with Conceptual Logic Programs, *Proceedings of RuleML 2004, Lecture Notes in Computer Science* 3323, pp.113-127, Springer (2004)
- [Heymans 05] S. Heymans, D. V. Nieuwenborgh, and D. Vermeir: Nonmonotonic Ontological and Rule-Based Reasoning with Extended Conceptual Logic Programs, *Proceedings of ESWC 2005*, pp.392-407 (2005)
- [Horrocks 03] I. Horrocks and P. F. Patel-Schneider: Reducing OWL entailment to description logic satisfiability, *Proceedings of ISWC-2003, Lecture Notes in Computer Science* 2870, pp. 17-29, Springer (2003)
- [Kifer 92] M. Kifer and V. S. Subrahmanian: Theory of generalized annotated logic programming and its applications, *Journal of Logic Programming*, Vol.12 (4), pp. 335-367 (1992)
- [宮下 06] 宮下, 若木: セマンティック Web ツールの試作 -OWL オントロジー変換器と構造エディタ-, 電子情報通信学会 2006 年総合大会 情報・システムソサエティ 総合大会特別号, ISS-P-38, p. 56 (2006)
- [野沢 07] 野沢, 若木: 動的セマンティック Web 推論呼び出しを行う議論エージェントシステム, 平成 18 年度電子情報通信学会東京支部学生会研究発表会講演論文集, p.96 (2007)
- [Sawamura 06] H. Sawamura, T. Wakaki and K. Nitta: The Logic of Multiple-Valued Argumentation and its Applications to Web Technology, *Computational Models of Argument*, IOS Press, *Proceedings of COMMA 2006*, pp.291-296 (2006)
- [Takahashi 04] T. Takahashi, H. Sawamura: A Logic of Multiple-Valued Argumentation, *Proceedings of AAMAS 2004*, pp.789-805 (2004)
- [Wakaki 06] T. Wakaki, T. Mukai, H. Miyashita, H. Sawamura, K. Matsunaga, T. Fukumoto and K. Nitta: Automated Argument System based on Logic of MultipleValued Argumentation, *Proceedings of AAMAS 2006*, pp.1461-1462 (2006)

〔担当委員：桜井 成一郎〕

2006 年 11 月 29 日 受理

## ◇ 付 録 ◇

### A. OWL DL から DL *SHOIN(D)* への変換規則の例

カリキュラム・オントロジー (図 3) の OWL DL 表現から DL *SHOIN(D)* 表現への変換に使用された変換規則を以下に示す。

- (1) ● OWL DL 表現:
 

```
<owl:Class rdf:ID="C">
  <owl:equivalentClass rdf:resource="#D">
</owl:Class>
```

  - *SHOIN(D)* 表現:  $C \equiv D$
- (2) ● OWL DL 表現:

```

<owl:Class rdf:ID="C">
  <owl:unionOf rdf:parseType="Collection">
    <概念 1>
    <概念 2>
    :
    <概念 n>
  </owl:unionOf>
</owl:Class>

```

- $SHOIN(D)$  表現:  $C \equiv \langle \text{概念 } 1 \rangle \sqcup \langle \text{概念 } 2 \rangle \sqcup \dots \sqcup \langle \text{概念 } n \rangle$

## B. カリキュラムオントロジー $\mathcal{K}_1$ の OWL DL 表現

オントロジー表現変換器により生成されたカリキュラムオントロジー  $\mathcal{K}_1$  の OWL DL 表現を以下に示す。

```

<?xml version="1.0" encoding="Shift-JIS"?>
<!DOCTYPE owl \[<![!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"]>\]
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Class rdf:ID="university_curriculum">
    <owl:equivalentClass rdf:resource="#course"/>
  </owl:Class>
  <owl:Class rdf:ID="course">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#faculty_e"/>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#faculty_s"/>
        <owl:Class rdf:about="#faculty_h"/>
      </owl:unionOf>
    </owl:unionOf>
  </owl:Class>
  <owl:Class rdf:ID="cs_dept">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#faculty_e"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ee_dept">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#faculty_e"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="math_dept">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#faculty_s"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="philo_dept">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#faculty_h"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="we_logic">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#western_logic"/>
      <owl:Class rdf:about="#eastern_logic"/>
    </owl:unionOf>
  </owl:Class>
  <owl:Class rdf:ID="we_logic">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#logic"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="mp_logic">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#math_logic"/>
      <owl:Class rdf:about="#philo_logic"/>
    </owl:unionOf>
  </owl:Class>
  <owl:Class rdf:ID="mp_logic">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#logic"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="easy_get_credit">
    <owl:Restriction>
      <owl:onProperty rdf:resource="#reg"/>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#pass"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:Class>
  <owl:Class rdf:ID="hard_get_credit">
    <owl:Restriction>
      <owl:onProperty rdf:resource="#reg"/>
      <owl:someValuesFrom>
        <owl:complementOf>
          <owl:Class rdf:about="#pass"/>
        </owl:complementOf>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:Class>
  <rdfs:Description rdf:ID="philosophy">
    <rdf:type rdf:resource="#hard_get_credit"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="philosophy">

```

```

    <reg rdf:resource="#st_0"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="11">
    <rdf:type rdf:resource="#cs_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="12">
    <rdf:type rdf:resource="#math_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="13">
    <rdf:type rdf:resource="#philo_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="ai">
    <rdf:type rdf:resource="#cs_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="c_programming">
    <rdf:type rdf:resource="#cs_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="prolog_programming">
    <rdf:type rdf:resource="#cs_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="technical_english">
    <rdf:type rdf:resource="#cs_dept"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="12">
    <rdf:type rdf:resource="#math_logic"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="13">
    <rdf:type rdf:resource="#philo_logic"/>
  </rdfs:Description>
  <rdfs:Description rdf:ID="ai">
    <rdf:type rdf:resource="#not(pass)"/>
  </rdfs:Description>
</rdfs:RDF>

```

## 著者紹介



### 若木利子 (正会員)

1969 年名古屋大学理学部物理学科卒業。京都大学理学研究科博士課程単位取得退学。博士(理学)。(株)富士通研究所、(株)富士通国際情報社会科学研究所、(株)富士通を経て、1999 年より芝浦工業大学システム工学部電子情報システム学科教授。非単調推論と論理プログラミング、議論エージェントに興味をもつ。人工知能学会、情報処理学会、ソフトウェア科学会、電子情報通信学会各会員。



### 沢村 一 (正会員)

1978 年北海道大学大学院工学研究科情報工学専攻単位取得退学。博士(工学)。1980 年-1996 年(株)富士通国際情報社会科学研究所、(株)富士通研究所で室長、主管研究員などを歴任。オーストラリア国立大学、ニュージーランドヴィクトリア大学の Visiting Fellow を歴任。1996 年-現在 新潟大学工学部情報工学科助教授。計算論理学、ソフトウェア基礎論、人工知能分野に興味をもつ。情報処理学会、ソフトウェア科学会、日本科学哲学学会の各会員。



### 福本 太郎 (正会員)

1980 年生。2003 年新潟大学工学部福祉人間工学科卒業。2005 年同大学大学院自然科学研究科情報・計算機工学専攻修士課程修了。現在同大学大学院自然科学研究科情報工学専攻博士課程に在学中。エージェントテクノロジー、機械学習に興味を持つ。



### 向井 孝徳

2006 年芝浦工業大学システム工学部電子情報システム学科卒業。現在、芝浦工業大学大学院工学研究科電子電気情報工学専攻修士課程に在学。



### 新田克己 (正会員)

1971 年東工大工学部電子工学科卒業。1980 年東工大大学院理工学研究科博士課程修了。工学博士。同年電子技術総合研究所入所。1989 年から 1994 年まで(財)新世代コンピュータ技術開発機構に赴向。1996 年東工大大学院総合理工学研究科。現在に至る。法的推論システム、交渉支援エージェント、マルチモーダルインタフェースなどの研究に従事。人工知能学会、情報処理学会、電子情報通信学会、情報ネットワーク法学会、IEEE 各会員。