

온톨로지 추론 기술 동향[†]

숭실대학교 최정화* · 박영택**

1. 서 론

온톨로지는 TBox와 ABox로 구분이 된다. TBox는 온톨로지의 스키마를 포함하고 있으며, ABox는 인스턴스로 구성된다. 초기의 온톨로지 추론은 온톨로지 스키마인 TBox의 무결성 테스트를 위해서 주로 이용되었다. 즉, 온톨로지를 설계하는 과정에서 스키마에 정의된 클래스(concept)와 속성(role)에 대한 검증을 위해서 온톨로지 추론이 활용되었고, 자연스럽게 온톨로지에 정의된 클래스사이의 암시적인 subsumption 구조를 명시적으로 추론하는 용도로 이용되었다. 그러나 온톨로지를 활용하는 측면에서 보면 ABox에 많은 인스턴스(individual)들이 구축되고 이러한 ABox에 대한 추론이 많이 연구되는 추세이다. OWL(Web Ontology Language)[1]이 온톨로지 언어로 활용되고 있고, OWL은 서술논리(Description Logic)[2]의 모델을 채택하고 있으므로 온톨로지 추론은 서술논리가 가지는 의미 모델에 기초를 두고 있다. 근래에는 온톨로지 규칙에 대한 추론의 필요성이 증대되어 SWRL(Semantic Web Rule Language)[3]에 대한 추론에 대한 연구도 진행되고 있다.

온톨로지 subsumption 추론은 sound하고 complete한 알고리즘[4]을 활용하여 클래스 사이의 subsumption 구조를 파악하는 방식이 주를 이루고 있다. 이러한 알고리즘은 트리 구조의 모델을 구축하는 방식과 FOL(First Order Logic)으로 변환하여 Datalog로 변환하는 방식[5]이 있다. 각각의 방식은 대용량 TBox, 대용량 ABox 추론에서 각기 장점을 보이고 있다. 이와 같은 방식과는 다르게, OWL이 가지고 있는 논리적 특성을 규칙(axiom)으로 표현하여 이를 고속으로 추론하는 방식을 취하는 연구 방식도 있다. 비록

이러한 방식은 complete하지 못한 단점이 있지만 여러 부분에서 성공적으로 활용되고 있다.

본 논문에서는 온톨로지 추론을 TBox, ABox, SWRL 추론으로 나누어 2장에서 기술하고, 온톨로지 추론 기술을 tableaux 알고리즘, disjunctive datalog, axiomatic 방식으로 나누어 3장에서 설명한다. 4장에서는 국내외에서 개발된 온톨로지 추론 엔진에 대한 소개를 한다. Subsumption 알고리즘을 활용하는 엔진들과 공리(axiom) 기반의 엔진들에 대해서 설명한다.

2. 온톨로지 추론 개요

OWL로 표현된 온톨로지를 추론하기 위해서는 서술 논리로 표현된 지식을 추론하는 기술의 이해가 필요하다. 서술 논리는 기본적으로 클래스, 속성, 인스턴스로 사람이 가지고 있는 지식을 표현한다[2]. 일반적으로 우리가 서술 논리로 사람이 가지고 있는 지식을 표현하는 경우 TBox(Terminological Box)와 ABox(Assertional Box)의 구조를 활용하여 지식을 표현한다. TBox에는 지식의 Terminological 부분이 선언되고 ABox에는 지식의 Assertional 부분이 선언된다. 예를 들어 TBox와 ABox에는 다음과 같은 내용을 포함하게 된다.

TBox:

```
PowerPerson ≡ FanClub-Person ⊓  
    ∃ love. ¬MovieStar ⊓ ∃ love. ¬Singer  
SoccerPlayer-Fan ⊆  
FanClub-Person ⊓ ∃ love.SoccerPlayer  
MovieStar ⊆ Star  
Singer ⊆ Star  
SportMan ⊆ Star
```

ABox:

```
FanClub-Person(person101)  
SoccerPlayer(player001)  
love(person101, player001)
```

서술 논리는 많은 종류가 연구되고 있으며 사람이

[†] 본 연구는 국가 IT 온톨로지 인프라 기술개발사업의 지원을 받았습니니다.

* 학생회원

** 종신회원

가지고 있는 지식을 표현하는 방식에 따라서 그 종류가 다르다. 가장 기본적인 서술 논리는 ALC 서술 논리이다. 이는 $\sqcap, \sqcup, \exists, \forall, \neg$ 만으로 개념을 표현한다. 서술 논리는 지식 표현 범위에 따라 다음과 같은 약어를 추가적으로 사용한다.

- S: ALC + transitivity
- H: for role inclusion axioms (role hierarchy, $R \sqsubseteq S$)
- O: for nominals (of form $\{i_n, \dots, i_n\}$)
- I: for inverse roles (of form R^{-})
- N: for number restrictions (of form $\geq n R, \leq n R$)
- Q: for qualified number restrictions (of form $\geq n R.C, \leq n R.C$)

예를 들어, ALC에 transitive role을 추가한 경우에 S 서술논리라고 표현한다. 그리고 현재 많이 활용되고 있는 OWL DL은 SHOIN(D) 서술 논리, 즉 ALC + R^+ + role hierarchy + nominal + inverse roles 형식의 서술 논리에 기반을 두고 있다. D는 concrete domain을 의미한다. 본 논문에서는 서술 논리의 추론을 세 가지로 나누어 살펴본다. 첫째, subsumption 관계를 추론하는 TBox 추론과 둘째, instantiation 관계를 추론하는 ABox 추론, 마지막으로 규칙을 SWRL로 작성해서 TBox와 ABox의 관계를 추론하는 SWRL추론으로 분류하여 기술한다.

2.1 TBox 추론

TBox 추론이란 subsumption 관계를 추론하는 것을 의미하는데 이 과정에서는 클래스와 서브클래스의 관계를 추론하게 된다. Subsumption 추론은 하나의 클래스가 다른 클래스를 포함하는 것을 의미한다.

일반적으로 subsumption 관계는 $C \sqsubseteq D$ 와 같은 형식으로 표현된다. 이를 증명하기 위해서는 $\neg C \sqcup D$ 가 satisfiable[6]하다는 것을 보여주면 된다. 즉, $C \sqcap D$ 가 unsatisfiable 하다는 것을 증명하면 subsumption 관계가 추론된다. 이를 tractable하게 증명하기 위해서 많은 추론엔진이 연구되고 있으며, 본 논문에서는 가장 보편적으로 사용되는 온톨로지 저작 도구인 Protege[7]를 이용하여 온톨로지를 구축하고, tableaux[4]알고리즘 방식을 기반으로 한 RacerPro[8]를 이용하여 TBox 추론을 설명한다. 다음은 TBox에 저장된 지식을 서술 논리로 나타낸 것이다.

FanClub-Person
 PowerPerson \sqsubseteq FanClub-Person \sqcap

$\exists \text{love}.\neg \text{MovieStar} \sqcap \exists \text{love}.\neg \text{Singer}$

SoccerPlayer-Fan \sqsubseteq FanClub-Person \sqcap
 $\forall \text{love.SoccerPlayer} \sqcap$
 $\exists \text{love.SoccerPlayer}$

대부분의 온톨로지 저작도구와 추론 엔진은 일반적으로 DIG(Description Logic Interface Group)[9]인터페이스로 연결된다. Protege에 RacerPro를 연결하여 TBox의 subsumption 온톨로지 추론을 수행할 수 있다. 추론결과는 다음과 같다.

SoccerPlayer-Fan \sqsubseteq PowerPerson

이는 SoccerPlayer-Fan의 조건으로 closure axiom을 사용하여 한명 이상의 SoccerPlayer를 좋아하는데 Soccer-Player 만을 좋아해야한다고 정의하였고, 이 제약조건이 PowerPerson 조건에 subsume되었기 때문에 이와 같은 TBox 추론 결과를 보이게 된다.

하지만 closure axiom을 사용하지 않고 존재 정량자(existencial quantifier)만 사용하여 SoccerPlayer-Fan의 조건을 정의할 경우, 즉 한 명 이상의 SoccerPlayer를 가져야 한다고만 하면 SoccerPlayer 이외에 다른 Star도 좋아할 수 있다는 것을 가정할 수 있기 때문에 SoccerPlayer-Fan이 PowerPerson에 속하지 않게 된다. 이는 OWL 온톨로지 추론이 Open World Assumption(OWA)[10]에 기반을 두고 있기 때문이다.

반면에 데이터베이스와 기타 인공지능 시스템에서는 Closed World Assumption(CWA)[6]에 기반을 두고 있다. CWA는 질의가 들어왔을 때, 주어진 데이터베이스나 지식베이스의 데이터만을 가지고 추론하므로 원하는 정보가 없다면 False라고 판단하는 반면, OWA는 False라고 증명 되지 않은 모든 사실은 True가 될 수 있을 지도 모른다고 가정하게 된다. 이러한 특성 때문에 온톨로지를 구축하는 경우는 closure axiom을 이용하여 OWA하에서 원하는 정보를 표현하여야 우리의 생각대로 온톨로지가 추론되는 것을 확인할 수 있다.

2.2 ABox 추론

ABox 추론이란 인스턴스 관계를 추론하는 기능으로 이 과정에서는 하나의 인스턴스가 정의된 클래스에 포함되는지를 검사한다. ABox 추론은 realization이라고도 하며, 인스턴스 $C(a)$, 속성 $R(a, b)$, $a = b$, $a \neq b$ 와 같은 형식으로 표현된 인스턴스들이 어느 클래스에 subsume되는지 추론한다. ABox 추론의 예는

TBox 추론에서 살펴본 온톨로지 예제에 인스턴스를 추가하여 추론되는 과정을 설명한다.

```
FanClub-Person(person101)
love(person101, player001)
SoccerPlayer(player001)
```

이와 같은 예의 ABox 추론 결과는 다음과 같다.

```
PowerPerson(person101)
```

이와 같은 결과는 "FanClubPerson person101은 SoccerPlayer player001을 좋아한다."라는 사실(fact)이 존재하고 PowerPerson 클래스는 "영화배우와 가수를 좋아하지 않는 사람은 PowerPerson이다."라는 제약조건을 가지므로 person101이 PowerPerson이라는 것이 추론됨을 입증 할 수 있다.

2.3 SWRL 추론

SWRL 추론이란 경험적 규칙을 SWRL로 작성하여 TBox와 ABox 관계를 추론하는 것을 의미하는데 이 경우는 규칙의 조건부에 정의한 TBox와 ABox 관계가 성립하는 경우 규칙의 결론을 유도하여 새로운 지식을 추론한다. SWRL은 Horn 논리와 비슷한 구문으로 규칙을 사용자가 표현하기 쉬운 상위 레벨의 구문으로 작성할 수 있도록 개발되었다. 모든 규칙은 OWL 개념들(즉 클래스, 속성, 인스턴스)의 term으로 표현된다. 다음은 TBox와 ABox에 저장된 지식과 SWRL 규칙의 예이다. ABox 추론 예에 다음과 같은 클래스와 인스턴스, 속성을 추가하였다.

```
TBox:
Mother ⊆ FanClub-Person
Trot-Singer ⊆ Singer
Singer ⊆ Star
```

```
ABox:
Trot-Singer(singer02)
Mother(mom99)
love(mom99, singer02)
```

```
SWRL rule-1:
(?x) ∧ Mother(?y) ∧ love(?y, ?x)
```

→ hasAuntFan(?x, ?y)

본 논문에서는 Protege에 axiomatic 방식의 Jess 추론엔진을 연결하여 SWRL 규칙의 추론 결과를 살펴본다. 다음은 위와 같이 TBox와 ABox에 지식이 존재 할 경우 SWRL rule-1에 의해 추론된 결과이다. rule-1의 조건에 맞는 지식은 모두 hasAuntFan이라는 관계가 추론되어 ABox에 저장된다. 이 예에서는 다음과 같은

하나의 hasAuntFan 관계가 추론됨을 볼 수 있다.

```
hasAuntFan(singer02, mom99)
```

3. 온톨로지 추론 기술

일반적으로 서술 논리로 표현된 지식을 추론하는 데는 sound하면서 complete하고 tractable한 추론 알고리즘이 존재해야한다. 온톨로지 추론에서의 'sound 하다'의 의미는 valid한 subsumption 관계만이 추론되어져야함을 말하고, 'complete 하다'는 모든 valid한 subsumption 관계들이 추론되어야함을 의미한다. 그리고 'tractable 하다'의 의미는 모든 valid한 subsumption 관계들이 polynomial 시간 내에 추론이 가능한 것을 말한다. 이러한 알고리즘은 세 가지 방식으로 구분 할 수 있다. 첫째, 트리 구조의 모델을 구축하는 대용량 TBox 추론에 적합한 tableaux 알고리즘 방식과 둘째, FOL로 변환하여 datalog로 변환하는 대용량 ABox 추론에 적합한 disjunctive datalog 방식, 셋째, axiomatic rule을 사용한 빠른 추론 속도에 중점을 둔 axiomatic 방식으로 나눌 수 있다.

3.1 Tableaux 알고리즘 방식

Tableaux 알고리즘은 서술논리로 표현된 지식을 추론하기 위한 방법으로 completion 트리를 생성하여 각 노드의 satisfiability를 증명한다. 이 알고리즘은 resolution refutation과 유사하며 sound하고 complete한 추론 엔진을 개발하기 위해 추론기능을 구현하였으며 대표적인 엔진으로 FaCT, FaCT++, Pellet, RacerPro[4,13,8] 등이 있다.

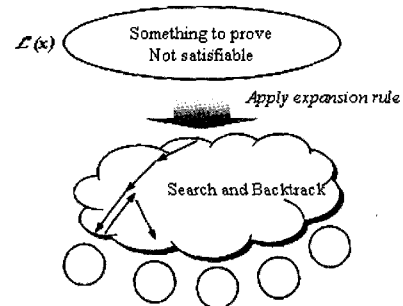


그림 1 Tableaux 알고리즘의 아이디어

Tableaux 알고리즘 방식은 증명하고자 하는 내용의 부정에 대해서 다양한 변환 규칙을 적용하여 satisfiable하지 않음을 보여 증명하고자 하는 내용이 satisfiable함을 증명하는 방식을 취하고 있다. 그림 1과 같이 모든 탐색공간을 검색하여 증명하고자하는 것의 부정이 satisfiable하지 않다는 것을 보여 줌으로써, satisfiable함을 추론하게 된다.

서술논리는 $\cap, \cup, \exists, \forall, \neg$ 등으로 표현되기 때문에 tableaux 알고리즘에서는 이에 대한 expansion 규칙을 반복적으로 적용하면서 탐색공간을 확장한다. 다음은 tableaux 알고리즘에서 활용하는 expansion 규칙의 일부를 보여준다.

- \cap -rule if 1. $(C \cap D) \in L(x)$
 2. $\{C, D\} \notin L(x)$
 then $L(x) \rightarrow L(x) \cup \{C, D\}$
- \cup -rule if 1. $(C \cup D) \in L(x)$
 2. $\{C, D\} \cap L(x) = \emptyset$
 then either $L(x) \rightarrow L(x) \cup \{C\}$
 or $L(x) \rightarrow L(x) \cup \{D\}$
- \exists -rule if 1. $\exists R.C \in L(x)$
 2. there is no y
 s.t. $L(\langle x, y \rangle) = R$ and $C \in L(y)$
 then create a new node y and edge $\langle x, y \rangle$
 with $L(y) = \{C\}$ and $L(\langle x, y \rangle) = R$
- \forall -rule if 1. $\forall R.C \in L(x)$
 2. there is some y
 s.t. $L(\langle x, y \rangle) = R$ and $C \notin L(y)$
 then $L(y) \rightarrow L(y) \cup \{C\}$

모든 탐색 공간의 단말 노드가 모순(contradiction)이라는 것을 보여줌으로써 증명하고자 하는 subsumption 관계의 부정이 unsatisfiable하다는 것을 증명하게 된다. tableaux 알고리즘의 과정은 탐색 공간을 검색하여 unsatisfiability를 확인하므로 많은 backtracking을 수행하게 된다. OWL 수준의 온톨로지 추론을 tractable하게 하기 위해서는 이러한 backtracking에서 발생할 수 있는 비효율적인 문제를 보다 현명하게 처리하는 최적화 방법이 필요하다. tableaux 알고리즘에서는 이를 위해 semantic branching, dependency directed backtracking, partial backjumping, satisfiability status caching, optimized blocking과 같은 다양한 기법을 활용하고 있다.

3.2 Disjunctive Datalog 방식

온톨로지를 표현하는 방법으로 서술 논리 기반의 표현 방식에 대해서 주로 다루고 있는데, 또 다른 연구 분야에서는 F-논리[10]를 기반으로 온톨로지를 표현하는 연구가 매우 활발하게 진행되고 있다. Disjunctive datalog 방식은 SWRL과 F-Logic에 대한 온톨로지를 직접 다룰 수 있는 기반을 제공한다. 대표적인 엔진으로는 KAON2가[5] 있으며, 이 추론엔진은 대용량의

ABox를 추론하기 위하여 개발 되었다. Disjunctive datalog 프로그램은 다음과 같은 형태의 규칙들의 유한한 집합이다.

$$A_1 \vee \dots \vee A_n \leftarrow B_1, \dots, B_m$$

$$head(r) = \{ A_i \mid 1 \leq i \leq n \}$$

$$body(r) = \{ B_i \mid 1 \leq i \leq m \}$$

Disjunctive datalog 방식은 온톨로지를 TBox와 ABox로 분류하여 분리된 ABox는 별도의 처리 없이 위와 같은 규칙들의 절의 형태, 즉, $A_1 \vee K \vee A_n \vee \neg B_1 \vee K \vee \neg B_m$ 와 같은 형태로 datalog에 입력된다. 그리고 TBox 중 존재 정량자를 가지는 지식들은 존재 정량자 핸들러를 통해 존재 정량자와 관계된 구문을 찾아서 정해진 규칙을 처리 한 후 ABox 형태로 변환되어 datalog에 입력되어 진다. 나머지 존재 정량자와 관계 없는 TBox 구문들은 clausification 과정에 의해서 datalog에서 처리할 수 있는 절의 형태로 변환된다. 즉 규칙의 조건부에 해당하는 모든 변수는 지식베이스에 명시된 사실 만이 값으로 할당 될 수 있다는 의미로 정의된다.

KAON2는 DL-Safe 규칙을 사용하여 비결정 가능한 절의의 응답 처리 문제를 해결 하였다. 아래는 DL-Safe 규칙처리의 예이다.

$$Person(John), Healthy(John), Person \sqsubseteq \exists father. Person$$

$$GoodSon(x) \leftarrow person(x) \wedge father(x, y) \wedge healthy(x)$$

$$\downarrow$$

$$GoodSon(x) \leftarrow person(x) \wedge father(x, y) \wedge healthy(x)$$

$$\wedge O(x), O(y)$$

이와 같은 규칙 처리 과정은 존재 정량자에서 발생할 수 있는 무한반복의 가능성을 규칙을 제한하는 방식의 블록을 적용하여 제거한다. 이 방법은 기존의 SWRL의 규칙에 $O(x)$ 와 $O(y)$ 를 추가하여 DL-Safe한 규칙으로 변환한다. 이러한 과정을 통하여 대용량 ABox의 추론을 결정 가능한 시간 안에 추론 할 수 있게 한다.

3.3 Axiomatic 방식

Axiomatic 방식은 수많은 규칙을 입력하여 서술 논리의 추론 기능을 구현한 추론 방식이다. Axiomatization이란 온톨로지를 FOL 언어로 변환하는 작업을 의미한다. 온톨로지 추론을 위한 axiom은 DAML (DARPA Agent Markup Language)부터 정의되어 활용되고 있다[11,12]. 이런 방식에서는 온톨로지 표현 언어의 시맨틱스를 axiom으로 표현하고 이를 실행함으로써 온톨로지 추론이 가능하다. Axiomatization

을 통해서 온톨로지를 FOL 언어로 변환하면 기존의 규칙 엔진이나 자동 증명기(automated theorem prover)들을 이용하여 추론 또는 증명 작업을 수행 할 수 있다. 이 방식은 추론엔진의 sound나 complete함 보다는 추론 속도에 중심을 둔 추론 엔진 개발에 이용 된다. 대표적인 엔진으로는 Jena2, 보쌈, JTP, OWLJessKB 등이 있다[15,17-19].

4. 국내외 추론 엔진

온톨로지에 대한 관심이 증대되는 추세에 따라서 OWL 온톨로지 추론 엔진 기술에 대한 연구도 활발히 진행되고 있는 추세이며 상용 엔진도 출시되어 대용량 데이터 실시간 추론에 활용되고 있다. 본 장에서는 추론엔진과 관련된 최신 연구 결과들을 살펴봄으로써 현재 이 분야의 State of the Art를 알아보고 국내에서의 대응 방안 도출에 기반을 제공하고자 한다. 현재, 추론엔진들은 3장에서 언급한 온톨로지 추론 알고리즘 방식에 따라 표 1과 같이 나누어 볼 수 있다.

표 1 온톨로지 추론 방식에 따른 온톨로지 추론 엔진

온톨로지 추론 엔진		연구 기관	구현 언어 및 상업용
온톨로지 추론 방식	추론엔진		
Tableaux 알고리즘 방식	FaCT, FaCT++	Ian Horrocks, U. of Manchester	LISP, C++기반의 open source
	Pellet	U. of Maryland, MIND Lab.	Java 기반의 open source
	RacerPro	Racer/Franz Inc.	LISP기반의 상업용 추론 엔진
Disjunctive Datalog 방식	KAON2	U. of Karlsruhe/Ontoprise	Java 기반의 API 및 상업용 추론 엔진
Axiomatic 방식	Jena2	HP Lab.	Java 기반의 open source
	보쌈	한국, ETRI	Java 기반 바이너리 버전 공개 중
	JTP	Stanford Univ.	FOPC 기반 하이브리드 온톨로지 추론
	OWLJessKB	Drexel Univ.	Java 기반의 open source

온톨로지 추론 엔진들은 기본적으로 다음과 같은 서비스를 제공한다. 첫째, 클래스의 계층구조를 생성하기 위해 모든 클래스 사이의 subsumption 관계를 계산하여 클래스들을 분류한다. 둘째, 온톨로지가 satisfiability한지 검사한다. 셋째, 온톨로지를 표현하고 있는 클래스 descriptor의 consistency를 검사한다. 이 기능은 구축된 온톨로지의 무결성을 검사하는데 이용 될 수 있으며 모순되는 개념이 온톨로지에 정의 될 수 없음을 확인해 준다. 넷째, 인스턴스가 어느 클래스에 속하는지 realization 기능을 제공한다.

4.1 FaCT 및 FaCT++

FaCT++(Fast Classification of Terminology) [4]은 기존의 FaCT 시스템에 새로운 기능을 추가하고 최적화하여 확장한 SHOIQ 서술 논리 추론 엔진이다. FaCT++는 string과 integer를 포함하는 datatype의 추론도 지원하도록 개발된 tableaux 알고리즘에 기초한 시스템이다. 또한 DIG 인터페이스를 제공하여 온톨로지 저작 도구들과 연결되어지며, FaCT++은 다음과 같이 기본적으로 추론 시스템이 제공하는 기능의 최적화를 위해 개발 되었다.

- 온톨로지의 consistency 검사
- single concept과 concept group의 satisfiability 검사
- 두 concept 간의 subsumption 관계 추론
- concept의 제약조건에 의해 온톨로지 계층을 재분류

특히 전처리 단계의 absorption, model merging 과정과 TBox의 satisfiability 검사의 최적화를 위해 ordering heuristics와 taxonomic classification 기술을 새로이 연구 하였다. 다음은 FaCT++에서의 최적화 방법 중 전처리의 최적화를 위한 알고리즘이다.

- Lexical normalization과 simplification: normalization단계에서 clash(ABox의 inconsistency) 발생을 전처리 과정 초기에 탐지하도록 하였다. 이 아이디어는 모든 개념을 SNF(Simplified Normal Form)으로 변환하여 파싱하는 동안에 simplification을 하도록 하여 트리 탐색 시간을 단축하는 이점이 있다. SNF란 $\neg, \sqcap, \sqcup, \leq$ 만을 허용한 형태를 말한다. 이는 \sqcup, \exists 가 존재하는 term을 전처리 단계에서 SNF로 변환함으로써 트리 탐색공간을 줄여준다.
- Absorption: absorption 단계에서는 트리 탐색공간을 확장시키는 TBox에 포함된 세 가지 형태에 대해 최적화 알고리즘을 제안하였다. 첫째, GCI(General Concept Inclusion axioms), 즉 $C \sqsubseteq D$, (C, D 는 complex concept expressions) 형태를 제거하기 위한 알고리즘을 최적화 하였다. GCI form은 tableaux 기반의 satisfiability와 subsumption 검사 시 성능을 감소하므로 absorption 과정의 최적화는 중요하다. FaCT++은 CD-classification 최적화 방법을 사용하여 GCI를 제거하였다. 둘째, TBox에 cycle이 발생하는 클래스들의 cycle 제거 방법을 최적화하였다. FaCT++에서는 subsumer hierarchy라 하

는 방법을 제안하여 TBox에 cycle이 존재하는 경우에 발생할 수 있는 문제점을 해결하였다. 마지막으로 지식베이스에 unique한 개념만 존재하게 하기 위해 동의어($C \equiv D$)를 변환하는 방법을 최적화하였다. 이 알고리즘은 clash 탐지 성능을 향상하고, simplification의 가능성 확장을 위해 사용한다.

4.2 Pellet

Pellet[13]은 SHOIQ 서술 논리의 추론을 위해 개발된 tableaux 알고리즘에 기초한 OWL-DL 추론시스템이다. 또한 taxonomy 분류의 확인을 위한 complete 문법을 가진다. Pellet은 OWL Lite와 OWL DL을 위한 sound하고 complete하며, decidable하고 일반적으로 유효한 추론 서비스를 제공한다. 그리고 불명확한 node를 처리하기 위해 heuristic을 사용함으로써 OWL Full 온톨로지를 처리한다. SWRL 추론을 지원하며, SPARQL(Simple Protocol and RDF Query Language)을 통한 질의응답이 가능하다.

Pellet의 두드러진 특징은 데이터타입(Datatype) 추론을 완벽하게 구현하였다는 것이다. 그림 2에서 보여주는 바와 같이 Tableaux 알고리즘 방식의 추론엔진에 별도의 XSD 추론엔진을 추가하여 데이터타입에 대한 처리를 수행하고 있다. 또한 DIG 인터페이스를 지원하여 다른 저작도구들과 통신할 수 있다.

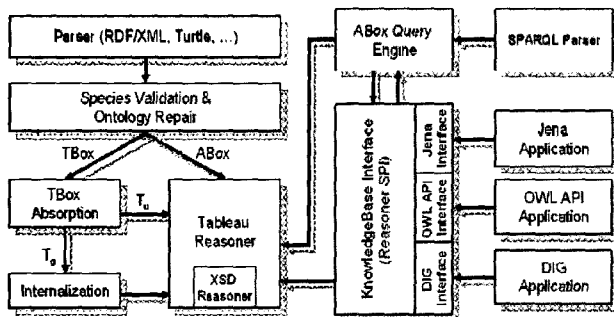


그림 2 Pellet 추론 엔진의 주요 구성 요소

Pellet은 다음과 같은 주요한 기능을 제공한다.

- 온톨로지 species validation 및 온톨로지 repair 기능
- Datatype 추론 지원
- Consistency, concept satisfiability 검사
- Classification, realization 지원
- Conjunctive ABox 질의 처리

이와 같은 기능의 최적화를 위해서 consistency, satisfiability, classification, realization, 질의 처리 전에 전처리 과정을 수행한다. 전처리 과정은 ta-

bleaux 알고리즘을 이용하여 split, TBox absorption, internalization, simplification, role hierarchy 등의 방법을 사용한다.

4.3 RacerPro

RacerPro(Remained ABox and Concept Expression Reasoner Professional)[14]는 ABox 계산의 실용적인 수행을 위해 만들어진 상업용 추론 시스템이다. RacerPro는 FacT에 내장된 모든 최적화 기술과 추가적으로 number restriction과 ABox를 다루는 기술을 지원한다.

ABox 추론은 tableaux 알고리즘을 기반으로 이루어진다. 그리고 대용량의 TBox에 대한 classification과 consistency 검사를 decidable한 시간 내에 처리할 수 있는 장점을 지니고 있다. 즉, TBox 추론에서는 상당한 성능을 나타내지만, 대용량의 ABox에서는 처리 속도가 제한적인 문제점을 가지고 있다. 또한 SWRL의 확장을 지원하며, 자체적인 문법을 사용한 nRQL(new Racer Query Language)를 사용하여 질의응답을 수행한다.

RacerPro는 다음과 같은 주요한 기능을 제공한다.

- 복수의 TBox와 ABox 추론을 지원
- ABox로 부터의 assertion 취소기능을 지원
- 질의가 처리된 후 ABox에 assertion을 추가 가능

4.4 KAON2

KAON2[5]는 기존의 추론 엔진에 많이 사용되는 tableaux 알고리즘을 사용하지 않고, 자체적인 알고리즘을 사용하여 추론을 수행한다. 기존의 tableau 알고리즘이 효과적인 TBox 추론을 위해 연구 되었듯이, KAON2는 대용량의 ABox 추론을 위해 효과적인 알고리즘을 개발하였다. KAON2는 이전의 KAON을 OWL에 맞게 수정, 보완한 추론 엔진이다. 이 엔진은 OWL을 비롯하여 SWRL과 F-Logic에 대한 온톨로지를 직접 다룰 수 있는 기반을 제공한다. 이는 추론 엔진으로서의 기능과 함께 온톨로지 구성된 지식 베이스에 새로운 지식을 추가, 수정 및 삭제할 수 있는 관리자로서의 기능을 포함한다. KAON2의 주요한 기능은 다음과 같다.

- DL 수준의 OWL과 SWRL을 프로그래밍 할 수 있는 Java API
- 원격 추론을 위한 독립형 서버 프로그램
- 질의를 수행할 수 있는 추론 엔진
- 외부 응용에서 네트워크를 통해 이용할 수 있는 DIG 인터페이스

- 일반 관계형 데이터베이스에서 데이터를 이용할 수 있는 어댑터

4.5 Jena2

Jena2[15]는 시맨틱 웹 기반의 자바 프레임워크로서 RDF/RDFS, OWL을 모두 지원하며 특히 데이터베이스에 효율적으로 저장하기 위한 스키마 생성 방법을 영구적인 모델로 제공한다. 또한 논리 처리를 위한 규칙 기반 추론 기능을 지원해 주며 statement의 triple 저장 구조를 가지고 있다. 또한 DIG 인터페이스를 지원하고 OWL 처리를 지원해주는 장점을 가지고 있으며 질의 언어로서 RDQL [16]을 지원한다. Jena에서 사용하는 질의의 형태는 그래프에 매칭되는 triple 패턴들을 찾는 것이다. 즉 질의의 한 형태는 다음과 같다.

($?x P ?y$) ($?y Q ?z$) 이 예는 그래프에서 매칭되는 $?x$, $?y$ 그리고 $?z$ 를 찾으려는 질의이다. 이러한 질의 형태를 지원하는 것이 RDQL이다. RDQL은 Jena 모델에서의 RDF를 지원하기 위한 질의 언어로써 triple 패턴을 찾아서 결과를 반영한다.

4.6 보쌘

보쌘(Bossam)[17]은 온톨로지 규칙 추론기능이 내장된 axiomatic 방식의 추론 시스템이다. 국내에서 처음으로 개발된 보쌘 추론엔진은 W3C의 시맨틱 웹 표준을 모두 지원할 수 있도록 개발되었으며, 소프트웨어 개발자가 쉽게 온톨로지와 규칙을 정의, 다양한 응용서비스를 개발할 수 있도록 한 것이 특징이다. 또한 시간 데이터 표현 및 처리를 지원하고, 산술식을 술어의 항으로 적을 수 있게 되어 있다.

보쌘 추론 엔진은 규칙 기반 추론 시스템으로써 프로그램 크기가 작고 가벼우며, 규칙과 사실의 매칭에는 RETE 알고리즘을 이용하여 추론 속도가 빠르다. ETRI측은 이 기술로 인해 향후 기존 웹 서비스에 시맨틱 정보를 연결, 사용자 환경에 맞는 서비스를 로봇이 직접, 탐색 판단해 처리하는 것이 가능해 졌다고 밝혔으며 보쌘 추론 엔진의 지식 표현력은 다음과 같다.

- 확장된 Horn Logic
- Frame[6] 지식 표현
- 규칙의 우선순위 설정
- 높은 레벨의 논리 술어 표현
- NAF 및 classical negation을 지원

보쌘 추론엔진의 추론 기능은 다음과 같다.

- RDFS, OWL, RuleML, SWRL 등 주요 웹 온

- 톨로지 언어 및 웹 규칙 언어를 지원
- 온톨로지와 규칙을 결합하여 활용 가능
- 질의 처리 (부칭개(Buchingae) 질의만 지원)

보쌘 추론엔진은 OWL의 모든 어휘를 지원하고 Jena처럼 지원 어휘 축소를 통해 성능 향상을 가져왔으며, 국가 산업 활성화 및 국가 기술 경쟁력을 확보하는데 크게 기여할 것으로 예상된다.

4.7 JTP

JTP(Java Theorem Prover)[18]는 Axiomatic 추론 방식의 객체 지향 추론 시스템이다. JTP는 DAML+OIL로 만들어진 메타 데이터를 KIF(Knowledge Interchange Format) 형식으로 읽어 들여서 전방향 추론을 할 수 있도록 개발되었다. Stanford KSL 연구팀은 DAML+OIL 지식을 autonomous하게 추론하기 위해서 추론 기반 질의 응답기를 구축하고, 소프트웨어 에이전트를 지식 서버로 보고 DAML로 표현된 지식 베이스를 효과적으로 추론하여 질의 응답을 수행하고자 JTP를 활용하고 있다.

JTP는 여러 개의 추론엔진으로 구성되어 있으며, 이 추론엔진들은 plug-and-play 통합 모듈 시스템 구조로 연결되어 있다. 이렇게 모듈화된 구조는 새로운 추론 모듈을 추가하여 시스템을 확장시키거나, 기존의 추론엔진을 재구성하는 것을 쉽게 한다. 특히 JTP는 DAML+OIL을 기반으로 하는 추론엔진으로서 적합한 구조를 가지고 있는데, 현재는 이 구조를 확장하여 OWL 지식베이스까지 지원한다. JTP는 온톨로지를 읽어 triple로 변환하여 지식베이스를 구성하고 온톨로지 언어의 axiomatic 시맨틱을 추론하는 내장된 규칙을 이용하여 새로운 지식을 추론한다.

4.8 OWLJessKB

OWLJessKB[19] 서술 논리 추론을 위한 메모리 기반 추론 엔진으로 OWL-Lite까지 지원한다. OWL 어휘를 Jess 엔진용 규칙 언어로 번역하여 추론 또는 증명 작업을 수행한다. 규칙을 기술할 때 웹 온톨로지 요소(클래스, 속성 등)를 규칙에 기술할 수 있도록 함으로써 추론 실행 시 온톨로지 기술된 지식을 참조할 수 있도록 한다. 이 때, 온톨로지 안에 기술된 axiom을 추론 엔진이 직접 해석할 수도 있고, 규칙 엔진을 이용하여 간접적으로 해석할 수도 있다. 또한 형식적인 접근보다는 실용적인 접근이며, 따라서 추론 알고리즘의 soundness와 completeness에 대한 증명을 우선시하지는 않는다.

5. 국내외 추론 기술 동향 및 향후 발전 방향

미국과 유럽을 중심으로 서술 논리 기반의 실시간 추론을 위한 연구와 DIG을 지원하는 인터페이스의 지원 및 활용 연구가 활발히 진행되고 있으며, 상용 엔진도 출시되어 대용량 데이터 실시간 추론에 활용되는 추세이다. 현재의 온톨로지 추론 기술의 세계적 개발 동향은 두 가지로 나누어 볼 수 있다. 첫째는 sound 하면서 complete한 추론 엔진을 개발하는 연구이다. 이 연구는 서술 논리의 구조를 이해하고 이를 기반으로 추론이 가능한 알고리즘을 활용하여 추론 기능을 구현한다. 최근에는 미국의 Pellet, RacerPro 및 EU의 KAON2의 활용이 점차 증가되고 있는 추세이다. 둘째는 sound나 complete 하지 않지만 추론 속도에 중점을 둔 추론 엔진을 개발하는 연구이다. 이 추론 엔진은 수많은 규칙을 입력하여 서술 논리의 추론 기능을 구현하는 방법을 사용한다. 미국의 Jena, OWLJessKB, 한국의 보쌈 추론 엔진 등이 많이 이용되고 있다. 이러한 서술 논리 및 F-논리를 활용하는 엔진 이외에도 보다 FOL에 가까운 방식을 이용하는 JTP나 Hoolet과 같은 추론 엔진 기술도 연구되고 있는 추세이다. 또한 온톨로지 추론에 기반이 되는 상위 온톨로지들을 표준화하는 연구도 활발히 진행되어 DAML-Time과 같은 시간 추론에 적합한 온톨로지를 처리할 수 있는 axiom을 처리하는 기능도 온톨로지 추론 엔진에 추가하여 연구되는 추세이다.

참고문헌

[1] Grigoris, A., Frank van Harmelen, A Semantic Web Primer, The MIT Press, 2004.

[2] Baader F., et al. The Description Logic Handbook, Cambridge, 2003.

[3] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosz, Mike Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C, 2004.
<http://www.w3.org/Submission/SWRL/>

[4] Ian Horrocks, Optimising Tableau Decision Procedures for Description Logics. PhD thesis, University of Manchester, 1997.

[5] KAON2, <http://kaon2.semanticweb.org/>

[6] Genesereth, M., and Nilsson, N. Logical

Foundations of Artificial Intelligence, Morgan Kaufman Publishers, 1987.

[7] Protege Homepage, <http://protege.stanford.edu/>

[8] RacerPro Homepage, <http://www.racer-systems.com/>

[9] <http://dig.sourceforge.net/>

[10] Steffen Staab and Rudi Studer (Eds.): Handbook on Ontologies: International Handbooks on Information Systems, Springer, 2004.

[11] <http://www.w3.org/TR/daml+oil-axioms>

[12] Jing Mei, Elena Paslaru Bontas, Zuoquan Lin, "OWL2Jess: A Transformational Implementation of the OWL Semantics. Proc. of ISPA, 2005, LNCS 3759: 599-608, 2005.

[13] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. "Pellet: a practical OWL-DL reasoner." Submitted for publication to Journal of Web Semantics, 2006.

[14] V. Haarslev and R. Moller. "Description of the RACER system and its applications," In DL2001 Workshop on Description Logics, Stanford, CA, 2001.

[15] K. Wilkinson, C. Sayers, H. Kuno, D. Reynolds, "Efficient RDF Storage and Retrieval in Jena2," Proceedings of SWDB'03.

[16] RDQL - A Query Language for RDF, <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>.

[17] Minsu Jang, Joo-chan Sohn, "Bossam: an extended rule engine for the web," Proceedings of RuleML 2004 (LNCS Vol. 3323), 2004.

[18] Fikes, Richard, Jessica Jenkins, and Gleb Frank. "JTP: A System Architecture and Component Library for Hybrid Reasoning." Proceedings of the Seventh World Multiconference on Systemics, Cybernetics, and Informatics, 2003.

[19] OWLJessKB: A Semantic Web Reasoning Tool. <http://edge.cs.drexel.edu/assemblies/software/owljesskb/>

최 정 화



2004. 2 숭실대학교 정보과학대학
컴퓨터학부(학사)
2006. 2 숭실대학교 컴퓨터학과(석사)
2006. 3~현재 숭실대학교 컴퓨터학과
박사과정
관심분야: 인공지능, 시맨틱 웹, 온톨로지
추론, Semantic Annotation
E-mail : cjh79@ailab.ssu.ac.kr

박 영 택



1978 서울대학교 전자공학과(학사)
1980 KAIST 전산학(석사)
1992 University of Illinois at
Urbana-Champaign(박사)
1981~현재 숭실대학교 컴퓨터학부 교수
관심분야: 인공지능, 시맨틱웹, 온톨로지,
지능형 에이전트
E-mail : park@comp.ssu.ac.kr

• 2007 한국컴퓨터종합학술대회(KCC 2007) •

- 일 자 : 2007년 6월 25~27일
- 장 소 : 무주리조트
- 내 용 : 논문발표 등
- 주 최 : 학회주관
- 상세안내 : http://www.kiss.or.kr/events/conf_intro.asp