

Jess를 이용한 OWL과 SWRL 통합추론에 관한 연구

Integration of OWL and SWRL Inference using Jess

이기철, 이지형

성균관대학교 컴퓨터공학과

Ki-Chul Lee, Jee-Hyong Lee

Department of Computer Engineering, Sungkyunkwan University

E-mail : lee77@skku.edu, jhlee@ece.skku.ac.kr

요 약

W3C에서는 온톨로지의 표준 언어로 OWL(Web Ontology Language)을 발표하였고 이를 활용한 온톨로지가 다양한 곳에 적용되어 구축이 되고 있다. 하지만, DL(Description Logic)기반인 OWL언어가 표현할 수 있는 규칙의 한계로 인하여 이를 확장하기 위한 연구가 활발히 진행되고 있다. 이러한 연구를 통하여 W3C에서는 OWL과 RuleML(Rule Markup Language)을 통합하여 규칙(Rule)에 대한 표현력이 더욱 향상된 SWRL(Semantic Web Rule Language) 언어를 제안하였다. 현재 이러한 연구는 OWL, SWRL 온톨로지 언어를 활용하고 Racer, Jess와 같은 엔진을 통하여 추론을 하는 형태로 활성화 되어 가고 있다. 하지만 이러한 형태로 온톨로지를 구축하는데 있어서 Racer를 이용한 DL추론, Jess를 이용한 Rule-base추론이 병행되고 있다. 이에 따라 본 논문에서는 온톨로지를 추론하기 위한 엔진으로 Racer와 Jess의 병행이 아닌, Jess를 이용하여 DL기반언어인 OWL온톨로지를 추론하는 것 뿐 만아니라 SWRL언어의 규칙 또한 추론할 수 있도록 한다. 이러한 시스템을 구축하기 위해 OWL을 Jess언어를 이용하여 추론할 수 있도록 개발된 OWLJessKB라는 틀과 SWRL언어를 추론하기 위해 Jess언어로 변환하여 이를 추론하는 SWRL Factory, 그리고 이를 이용하여 통합 추론하기 위한 세가지 통합 추론 플랫폼을 제안한다.

1. 서론

현재 W3C(World Wide Web Consortium)에서는 OWL과 Rule-ML을 이용하여 SWRL이라는 언어를 제안하였다. 이는 기존의 OWL만으로는 표현할 수 없었던 규칙을 포함 하여 이를 표현할 수 있도록 확장한 것이다 [1].

이러한 온톨로지 언어의 확장에 따라 이를 추론하기 위한 연구가 새로운 이슈가 될 것이라고 생각된다. 따라서 본 논문에서는 이러한 온톨로지 언어의 확장에 따라 이를 추론하기 위한 연구 중 하나로써 OWL언어와 규칙이 포함된 SWRL언어 모두를 Jess추론엔진을 이용하여 추론할 수 있도록 한다.

다음 2장에서는 온톨로지에 대해 살펴보겠다. 첫째로 온톨로지의 개념, 다음으로 온톨로지에서 사용하는 언어, 그리고 이를 추론하기 위한 연구에 대해 알아보고 마지막으로 온톨로지언어를 추론엔진에 적용하기 위하여 어떻게 변환하는지 알아보겠다. 3장에서는 본 논문에서 제안하는 OWL과 SWRL을 추론하기 위한 방안에 대해, 마지막 4장에서는 본 논문의 결론과 향후 연구에 대해서 논하도록 하겠다.

2. 관련 연구

2.1 온톨로지

온톨로지는 현재 W3C를 중심으로 세계 각국의 기관 및 기업에서 활발히 연구 중인 분야이다. 이는 수많은 웹상의 정보들 간의 관계를 이용하여 웹에 의미를 부여하고 이를 이용하고자 하는 시멘틱 웹을 목표로 하여 발전해 가고 있다. 이러한 연구는 XML(eXtensible Markup Language)기반인 RDF(Resource Description Framework), OWL, Rule-ML, SWRL와 같은 여러 가지 언어를 만들어 내었고 이러한 언어를 각종 규칙기반 추론엔진(Rulebase Inference Engine), 로직(Loginc)언어 등을 이용하여 추론 서비스를 제공하기 위한 여러 가지 연구가 진행되고 있다.

2.2 온톨로지 언어

현재 온톨로지를 개발하기 위한 언어로 W3C에서 제안하는 여러 가지 언어가 있다. 그중 XML기반의 언어인 OWL은 DAML+OIL(DARPA Agent Markup Language + Ontology Inference Language)라는 웹 온톨로지 언어에서 비롯되었으며, 인간에게 정보를 표

현하는데 그치지 않고 이러한 정보를 이용해서 새로운 정보를 추론해 낼 수 있도록 DL 언어를 토대로 하여 구성 되어 있다. 또한 OWL은 어휘의 의미(Semantic)와 그 관계(Relation)를 표현하는데 있어 매우 유용하게 쓰인다. [2]

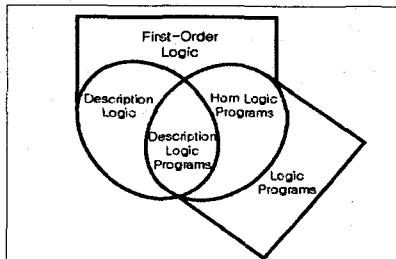
다음으로, SWRL은 OWL과 Rule-ML을 확장한 언어이며 OWL에 규칙에 대한 표현을 할 수 있도록 확장한 언어이다. 예를 들면, [그림1]과 같은 형태의 규칙을 SWRL언어로 표현할 수 있다. SWRL은 <swrl:Imp>태그를 이용하여 규칙을 선언하고 이는 <swrl:head>와 <swrl:body>을 이용하여 구분하며 각각의 절은 <swrl:AtomList>로 묶여서 표현하게 된다. 이러한 형태로 표현한 규칙과 추론엔진을 이용하여 온톨로지를 추론하는 데 활용할 수 있다.

```
Rule-3 :
hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) -> hasUncle(?x1,?x3)
```

[그림 1] SWRL Rule

2.3 온톨로지의 추론

온톨로지의 추론은 Tim Berners-Lee가 제안한 시맨틱 웹의 스택구조상에서 온톨로지의 바로 위 단계에 해당하며 이러한 온톨로지의 추론을 위한 로직의 표현 방법에 대한 각각의 표현 범위는 [그림2]과 같이 크게 LP(Logic Programs)와 FOL(First Order Logic)로 나눌 수 있다. 두 가지 언어 모두 Description Logic Program에 대한 표현은 가능하지만, First-Order Logic과 Logic Program은 각각의 표현의 범위를 갖고 있다. 이러한 이유로 로직 언어를 온톨로지 상에 적용할 수 있도록 온톨로지언어를 확장하는 연구가 활발히 진행 중이며 그 연구 결과의 하나로 SWRL과 같은 언어를 들 수 있을 것이다.



[그림 2] 논리 지식의 표현범위[3]

SWRL을 통하여 추론 기능이 확장된 온톨로지의 예를 들면, 다음 [그림 3]와 같은 Family 온톨로지를 예로 들 수 있다. 이 예는 OWL 온톨로지를 이용하여 DL-Rule을 정의하고 SWRL의 Rule을 정의한 것이다. 이렇게 구성된 DL과 Rule을 이용해서 온톨로지 상의 개체(Individual)에 대한 추론이 가능하다. 여기서는 DL 추론을 위하여 Racer를 이용하였고 SWRL의 추론을 위하여 Jess를 이용하였다. 이를 이용한 추론 결과는 [표 1]과 같다. 이 온톨로지는 Man과 Woman 클래스에 남자 10명(M1~M10)과 여자 10명(F1~F10)의 개체를 초기화 하였다. 표의 왼쪽 부분은 Racer를 이용한 추론 결과이다. 또한 Jess를 이용하여 SWRL Rule을 추론한 결과가 오른쪽에 표시되어 있다. 이와 같이 DL언어를 이용한 추론과 이에 SWRL 규칙을 포함하여 그 규칙에 대한 추론결과를 확인 할 수 있다 [4].

OWL ontology

- 1- Person := Man ∪ Woman
- 2- Parent := Person ∩ hasChild ≥ 1
Child := Person ∩ hasParent ≥ 1
(hasChild)⁺ = hasParent
- 3- Father := Parent ∩ Man Mother := Parent ∩ Woman
- 4- Son := Child ∩ Man Daughter := Child ∩ Woman
- 5- Brother := Sibling ∩ Man Sister := Sibling ∩ Woman
- 6- Nephew := Man ∩ (hasUncle ≥ 1 ∪ hasAunt ≥ 1)
- 7- Relative := Child ∪ Parent ∪ Aunt ∪ Nephew ∪ Niece ∪ Uncle ∪ Sibling

SWRL rules

Initial rule base

- 8- hasParent(?x1,?x2) ^ hasConsort(?x2,?x3) => hasParent(?x1,?x3)
- 9- hasParent(?x1,?x2) ^ hasSister(?x2,?x3) => hasAunt(?x1,?x3)
- 10- hasParent(?x1,?x2) ^ hasParent(?x3,?x2) ^ differentFrom(?x1,?x3) => hasSibling(?x1,?x3)
- 11- hasSibling(?x1,?x2) ^ hasDaughter(?x2,?x3) => hasNiece(?x1,?x3)

Rules mirroring the ontology knowledge

- MR1- hasSibling(?x1,?x2) ^ Man(?x2) => hasBrother(?x1,?x2)
- MR2- hasSibling(?x1,?x2) ^ Woman(?x2) => hasSister(?x1,?x2)
- MR3- hasParent(?x1,?x2) ^ Man(?x2) => hasFater(?x1,?x2)

[그림 3] Family OWL 온톨로지와 SWRL Rule

Class	Facts after RACER	After RACER and Jess
		Initial SWRL rule base
1. Person	M1->M10, F1->F10 [20]	M1->M10, F1->F10 [20]
2. Man	M1->M10 [10]	M1->M10 [10]
3. Woman	F1->F10 [10]	F1->F10 [10]
4. Parent	M1, M2, M3, M4, M6, M7, M8 [7]	M1, M2, M3, M4, M6, M7, M8, F1, F3, F4, F6, F7, F8, F10 [14]
5. Father	M1, M2, M3, M4, M6, M7, M8 [7]	M1, M2, M3, M4, M5, M6, M7, M8 [7]
6. Mother		F1, F3, F4, F6, F7, F8, F10 [7]
7. Child	M2, M3, M4, M5, M6, M9, M10, F2, F3, F5, F6, F9 [12]	M2, M3, M4, M5, M6, M9, M10, F2, F3, F5, F6, F9 [12]
8. Son	M2, M3, M4, M5, M6, M9, M10 [7]	M2, M3, M4, M5, M6, M9, M10 [7]
9. Daughter	F2, F3, F5, F6, F9 [5]	F2, F3, F5, F6, F9 [5]
10. Sibling		M2, M3, M5, M6, F2, F3, F5, F6 [8]
11. Brother		M2, M3, M5, M6 [4]
12. Sister		F2, F3, F5, F6 [4]
13. Relative	M1->M10, F2, F3, F5, F6, F9 [15]	M1->M10, F1->F10 [20]
		Additional mapping rules
14. Uncle		M2, M5, M6 [3]
15. Aunt		F2, F3, F5, F6 [4]
16. Nephew		M3, M5, M6, M9, M10 [5]
17. Niece		F5, F6, F9 [3]

[표 1] RACER and Jess Inferences

2.4 추론을 위한 온톨로지 언어의 변환

마지막으로 위와 같이 OWL과 SWRL을 추론하는데 있어서 SWRL을 Jess언어로 변환하는 방법에 관해 논 하겠다. Jess는 새로운 Fact들이 들어올 경우 이미 정의해 놓은 Rule에 의해 추론이 되는 Foward-Chaining 방식 뿐 만 아니라 반대의 경우인 Backward-Chaining 추론이 모두 가능한 전문가 시스템이다. 이러한 Jess를 온톨로지 상에서 이용하여 추론이 가능 하도록 하기 위해서는 온톨로지 언어를 Jess에서 사용할 수 있도록 변환하여 Fact와 Rule을 구성해야 한다.

```
Person(?x) ^ Man(?y) ^ hasSibling(?x, ?y) ^ hasAge(?y, ?age2)
^ swrl:greaterThan(?age2, ?age1)
-> hasOlderBrother(?x, ?y)
```

[그림 4] SWRL Rule

```
(defrule aRule (Person (name ?x)) (Man (name ?y))
(hasSibling ?x ?y) (hasAge ?x ?a1)
(hasAge ?y ?a2) (test (> ?age2 ?age1))
=> (assert (hasOlderBrother ?x ?y)))
```

[그림 5] Jess Rule

위 [그림 4]은 SWRL을 이용하여 정의한 Rule이다. 이러한 SWRL의 Rule을 Jess언어의 Rule형태로 변환한 것이 [그림 5]이다. 이는 Racer를 이용하여 DL기반 OWL을 추론한 후, SWRL언어로 정의한 Rule을 Jess를 이용하여 추론하기 위한 변환방법에 해당한다. 이는 Protégé[6]의 OWL Plug-in[7]에서 SWRL을 작성하고 이를 Jess를 이용하여 추론하고 그 결과를 조회하는 기능을 위해 [5]에서는 SWRL Factory라는 컴포넌트를

제작하였다. SWRL Factory는 OWL기반의 SWRL을 개발하기 쉽도록 Java API를 제공한다[5].

[그림 3]와 [표 1]그리고 위 [그림 5], [그림 6]과 같이 OWL의 추론이 적용되어 새로운 Fact가 생성되고 이후에 SWRL을 Jess에 적용하여 추론하기 위해 사용되는 방법에 해당한다. 다시 말하면, 이와 같이 SWRL언어를 변환 하는 SWRL추론의 경우 OWL에 대한 추론이 이루어진 후에나 가능하다는 것이다.

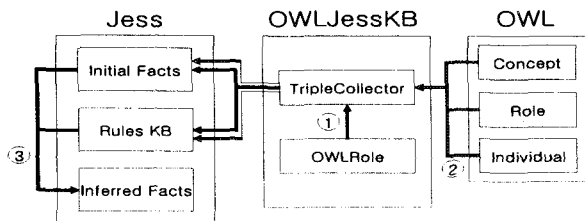
3. 통합 추론

앞에서 논한 내용과 같이 기존의 OWL온톨로지로 부터 이를 확장하여 SWRL언어를 이용하여 기존에 표현할 수 없었던 규칙들을 표현 할 수 있게 되었다. 이에 따라 이러한 온톨로지를 추론하기 위한 연구로 OWL과 SWRL을 통합 추론하기 위한 연구를 시작하였다. 즉, OWL만을 구성한 상태에서 RACER와 같은 DL기반 추론엔진을 사용하여 추론하거나, OWL의 추론과정 수행 후 SWRL규칙을 추가하여 Jess와 같은 전문가 시스템을 이용하여 추론을 수행하는 기존의 방법 대신에 OWL과 SWRL이 존재하는 상태에 모두 Jess를 이용하여 추론하는 것이다.

본 논문에서는 통합 추론을 위해 크게 세 가지 방법으로 연구를 진행하였다. 한 가지는 OWL을 추론하기 위한 틀인 OWLJessKB[8]를 OWL을 추론하기 위해 사용하고 SWRL을 추론하기 위한 부분을 추가하는 방법, SWRL의 추론을 위해 SWRL Factory를 이용하고 이에 OWL을 추론하기 위한 부분을 추가하는 방법. 마지막으로 OWLJessKB와 SWRL Factory모두를 사용하는 방법이다. 이 세 가지 방법에 대하여 논해보자.

3.1 OWLJessKB를 이용한 통합 추론

첫 번째로 기존에 OWL을 Jess를 이용하여 추론하도록 구현된 OWLJessKB를 수행하여 보았다. OWLJessKB는 DAMLJessKB에서 비롯되었으며, 이는 DAML언어를 Jess로 추론하기 위해서 개발된 틀이다. OWLJessKB는 OWL의 개체들을 모두 RDF에서 출발한 Triple 형태로 변환한 후, 이를 Jess의 Fact로 넣어 이를 추론하게 된다. [그림 6]에 이를 좀 더 자세히 나타내었다. ①은 OWLJessKB의 객체를 생성하면 OWL Role을 TripleCollector가 Triple형태로 변환하여 이를 Jess의 Fact와 Rule로 삽입한다. 다음으로 ②은 TripleCollector가 OWL을 읽어 들여 OWL의 Concept, Role, Individual을 Triple형태로 변환하여 Jess에 Fact와 Rule로 입력하게 된다. 마지막으로 ③은 이렇게 입력된 Fact와 Rule을 이용하여 Jess는 이를 추론하게 된다. 이렇게 추론된 결과는 Jess-Query를 이용하여 조회 할 수 있다.

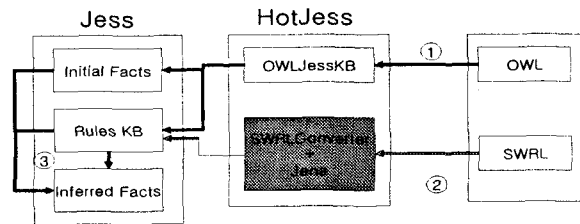


[그림 6] OWLJessKB의 추론과정

이와 같은 OWLJessKB를 이용하고 이에 SWRL을 Jess의 Rule로 변환하기 위한 부분을 추가하기 위하여 [그림 7]와 같은 HotJess를 설계하였고 이는 OWLJessKB를 통하여 변환된 OWL언어를 Jess를 이용하여 추론을 수행하고 SWRL Rule을 Jena[9]를 통

해 읽어 들여 이를 Jess언어에 맞게 Convert하여 SWRL Rule에 대한 추론을 수행하도록 한다.

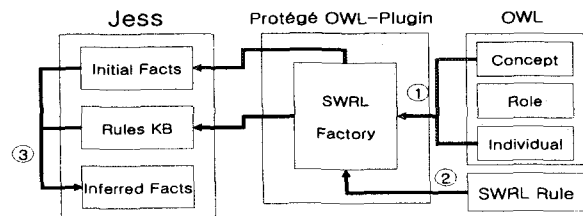
[그림 7]에서 회색으로 표시한 ②과정과 SWRLConvertor+Jena는 본 연구를 통하여 구현해야 할 부분에 해당한다. 이 부분은 Jess가 OWLJessKB와 OWL로부터 추론한 결과와 함께 기존의 OWL정보를 이용하여 SWRL의 추론을 수행 할 수 있도록 SWRL Rule을 Jess언어에 맞게 변환 하도록 한다. 또한 OWLJessKB는 OWL 언어를 RDF에서 사용하는 Triple형태로 변환하고 이를 다시 Jess언어로 변환하여 추론을 수행한다. 때문에, Jess를 통한 SWRL의 추론을 위해서 이 또한 Triple의 형태를 취해하도록 변환해야 한다.



[그림 7] OWLJessKB를 이용한 통합 추론 설계

3.2 SWRL Factory를 이용한 통합추론

다음으로 SWRL Factory를 이용하는 방법이다. SWRL Factory는 Protégé라는 온톨로지 모델링 툴의 Plug-in인 OWL-Plugin에서 RuleML의 확장 언어인 SWRL의 Rule을 정의하고 이를 Jess를 이용하여 추론하고 그 결과를 Protégé를 통해 확인 할 수 있도록 구현되어 있다. 다음 [그림 8]은 이러한 SWRL Factory가 Jess를 통해 추론을 수행하는 과정이다. 그림에서 보듯이 OWL의 Concept과 Individual, 그리고 SWRL의 Rule을 읽어 들여 각각 Jess의 Fact와 Rule로써 변환 후 이를 이용하여 새로운 결과를 추론하게 된다.

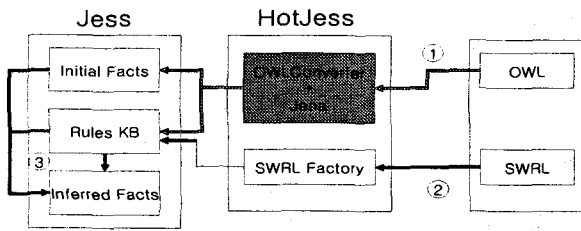


[그림 8] SWRL Factory의 추론과정

SWRL Factory는 OWLJessKB와는 달리 OWL을 Triple로 변환하여 수행하지 않고, 단지 Concept을 Jess의 Template, 그리고 Individual을 그에 대한 Fact로 변환하게 된다. 즉, SWRL Rule에 대한 추론을 위해 OWL의 Concept과 Individual에 대한 변환만을 수행한다. 때문에, SWRL Factory를 이용하여 통합 추론을 하기 위해서는 OWL에 대한 추론을 수행하는 부분이 필요하다. 또한 SWRL의 추론을 위해서는 SWRLFactory가 추론을 수행할 수 있도록 OWL을 변환해야 한다.

[그림 9]은 SWRL Factory를 이용하여 통합 추론하기 위한 흐름을 나타낸다. ①은 본 연구를 통하여 구현해야 할 부분으로, SWRL Factory가 변환 작업을 수행할 수 있도록 OWLConvert를 구성하여 SWRL Factory와 같은 방법으로 OWL을 변환하여 추론을 수행한다. 그 후, ②는 SWRL Factory를 이용하여 SWRL을 변환하고 이에 대한 추론을 수행하게 된다. 이와 같이 구성할 경우 OWL에 의해 추론된 결과가 이미 Jess의 Fact로 저장 되어 있기 때문에 SWRL Factory에서 OWL을 읽어 오기 위한 부분 없이 SWRL Rule 만을 Jess Rule

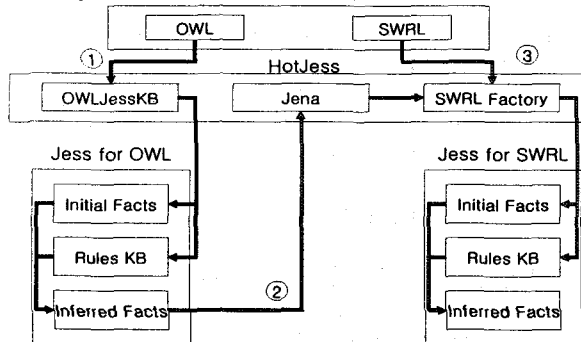
로 변환하여 이를 추론하게 된다.



[그림 9] SWRL Factory를 이용한 통합 추론 설계

3.3 OWLJessKB + SWRLFactory를 이용한 통합추론

마지막 세 번째 방법으로는 OWLJessKB와 SWRL Factory를 모두 이용하는 것이다.



[그림 10] OWLJessKB와 SWRL Factory이용 추론

[그림 10]에서 보듯이 OWLJessKB와 SWRL Factory를 모두 이용한다. 이는 ① OWLJessKB가 OWL을 읽어 들어 추론을 수행하고 ② 수행한 결과를 Jena를 통해 SWRL Factory에 설정한다. 다음으로 ③ SWRL Factory는 OWLJessKB에 의해 수행된 추론 결과와 SWRL Rule을 이용하여 추론을 수행하게 된다. 이와 같이 할 경우 OWLJessKB와 SWRL Factory에서 OWL의 Concept와 Role, Individual을 Jess 언어로 변환하는 방식이 다르기 때문에 OWLJessKB를 이용하여 추론 후에 이에 대한 추론 결과와 SWRL로 부터 또 다른 Jess엔진을 이용하여 추론을 수행해야만 한다.

3.4 비교분석

앞에서 논한 세 가지 통합 추론의 특징은 다음과 같다. 첫 번째 방법은 OWLJessKB를 이용하고 SWRL을 위한 SWRLConverter를 구현하여 OWL과 SWRL을 통합 한 추론이 가능하도록 한다. 두 번째 방법으로 SWRL Factory를 이용하고 OWLConverter를 구현하는 방법, 마지막으로 OWLJessKB와 SWRL Factory를 모두 이용하는 방법이다.

첫 번째 방법은 OWLJessKB가 추론하는 Triple의 개념을 사용하기 때문에 SWRL을 추론하기 위해서 또한 Triple의 개념을 도입하여 SWRLConvert를 구현해야 할 것이다. 그렇게 함으로써 SWRL을 추론하는데 필요한 OWL의 Concept, Individual에 대한 내용을 가져올 수 있고 또한 추론 결과를 기존의 Fact(OWL의 Fact)와 같은 형태로 삽입(Assert)할 수 있다.

두 번째 방법은 SWRL Factory를 이용한 추론이다. 이는 OWL을 추론하기 위해 OWLConverter를 구현해야 한다. 이는 SWRL Factory에서 추론하기 위해 필요한 OWL의 Concept과 Individual을 변환하는 방식과 동일하게 OWL의 Role또한 변환하여 이를 추론할 수 있어야 한다.

마지막으로 OWLJessKB와 SWRL Factory를 모두 이용하는 방법은 이들 모두 각각의 변환 방법이 존재하기 때

문에 OWL을 위한 Jess, 그리고 SWRL을 위한 Jess를 따로 구성하여 OWL에 대한 추론 결과를 Jena를 통하여 SWRL추론 시 필요한 Concept과 Individual을 넘겨주면, SWRL Factory는 이를 SWRL을 위한 추론엔진을 이용하여 추론하게 된다. 이는 앞에서 말한 두 방법에 비하여 구현은 다소 쉬운 편이나, OWL을 위한 Jess, SWRL을 위한 Jess를 만들어야 하는 불편함과, 자원의 낭비가 있다.

OWL과 SWRL을 통합하여 추론하기 위해서 수행해야 할 연구는 다음과 같다. 첫째로 이러한 통합 추론에 있어서 OWL을 추론할 경우 OWL에 SWRL의 내용을 포함하고 있으면 OWL의 추론이 수행되지 않는다. 둘째, OWL과 SWRL을 통합하여 추론하더라도 RACER를 통하여 수행할 수 있는 consistency check, subsumption check 등과 같은 기능 또한 포함되어야 할 것이다. 마지막으로, OWL과 SWRL언어를 모두 추론할 수 있도록 Jess언어의 형태로 변환하는 과정에 대한 연구가 중요하다고 할 수 있겠다. OWL과 SWRL언어 모두 XML기반의 언어이지만 각각 표현할 수 있는 규칙이 다르고 이를 추론하는 방식 또한 다르기 때문이다. 다음으로는 이에 대한 구현이다. 이러한 통합 과정을 구현하지 않는다면 증명되지 않은 이론에 불과할 것이다.

4. 결론

본 논문에서는 온톨로지의 언어와 이의 추론에 대하여 논하였고 OWL과 SWRL을 통합 추론하기 위해 어떠한 틀을 사용하고 이를 활용하여 통합 추론을 수행할지에 대해 논하였다. 각각의 방법 모두 온톨로지 언어를 Jess언어에 맞게끔 변환하는 방법이 다르고 또한 추론 방법이 다르기 때문에, 이러한 통합 추론 방법을 좀 더 비교 분석 하고 이에 대한 연구가 더욱 필요할 것이다.

5. 참고문헌

- [1] SWRL: A Semantic Web Rule Language Combining OWL and SWRL (<http://www.w3.org/Submission/SWRL/>)
- [2] OWL Web Ontology Language Guide (<http://www.w3.org/TR/owl-guide/>)
- [3] B.Groszof and M.Dean, "Semantic Web Rules with Ontologies, and their E-Business Applications," ISWC2004-Tutorial (http://sweetrules.projects.semwebcentral.org/iswc_tutorial.pdf)
- [4] C.Golbreich, "Combining Rule and Ontology Reasoners for the Semantic Web," Rules and RuleMarkup Languages for the Semantic Web (RuleML2004), p.6~22, Nov. 2004.
- [5] M.O'Connor, H.Knblach, S.Tu, M.Musen, "Writing Rules for the Semantic Web Using SWRL and Jess," Protege with Rules Workshop, Madrid, 2005.
- [6] Protege (<http://protege.stanford.edu/>)
- [7] OWL Plug-in (<http://protege.stanford.edu/plugins/owl/index.html>)
- [8] OWLJessKB : A Semantic Web Reasoning Tool (<http://edge.cs.drexel.edu/assemblies/software/owljesskb/>)
- [9] Jena2 - A Semantic Web Framework (<http://www.hpl.hp.com/semweb/jena2.htm>)