

Jess를 이용한 OWL과 SWRL의 통합추론에 관한 연구

Integration of OWL and SWRL Inference using Jess

이기철 · 이지형*

Ki-Chul Lee and Jee-Hyong Lee

성균관대학교 컴퓨터공학과

요 약

W3C에서는 온톨로지의 표준언어로 OWL(Web Ontology Language)을 발표하였고 이를 활용한 온톨로지가 다양한 곳에 적용되어 구축되고 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. 따라서 최근에는 온톨로지 구축을 위해 OWL과 SWRL언어가 함께 사용되기 시작하였다. 그러나 이렇게 통합적으로 구성된 온톨로지의 추론을 위한 방법에 대한 연구는 시작단계에 있다. 현재의 일반적인 방법은 OWL로 기술된 부분은 Racer등을 이용한 DL추론을 수행하고 SWRL로 기술된 부분은 Jess등을 이용한 Rule-base 추론이 병렬적으로 사용되고 있다. 이에 따라 본 논문에서는 OWL과 SWRL로 기술된 온톨로지를 추론하기 위한 엔진으로 Racer와 Jess의 병행이 아닌, Jess를 이용한 통합추론엔진의 개발에 관하여 기술한다. 이러한 시스템을 구축하기 위해 OWL을 Jess언어를 이용하여 추론할 수 있도록 개발된 OWLJessKB와 SWRL언어를 Jess언어로 변환하는 SWRL Factory를 이용하는 세 가지 통합 추론 플랫폼을 제안한다.

Abstract

OWL(Web Ontology Language) is the Ontology Standard Language and the a lot of Ontologies are being constructed in OWL. But the research on the extension of OWL is also progressing because of the limit of representation power of in OWL language. The W3C suggests the SWRL(Semantic Web Rule Language) based on the combination of OWL and RuleML(Rule Markup Language), which is improved in the representation of rule. Thus, both OWL and SWRL are used for developing ontologies. However, research on inference of ontologies written in both languages is just begun. These day, for the inference of ontologies written in both languages, ontologies and divided in to two parts : The part written in OWL and written in SWRL. For the inference of the part written in OWL, Racer, a DL based inference engine, is used and for the other part Jess, a rule-based engine, is used. In this paper, we will propose three methods for integrated inference of the OWL part and the SWRL part of ontologies using Jess and some tools for ontology inference : OWLJessKB and SWRLFactory

Key Words : 온톨로지, OWL, SWRL, Jess

1. 서 론

현재 W3C(World Wide Web Consortium)에서는 OWL과 RuleML을 이용하여 SWRL이라는 언어를 제안하였다. 이는 기존의 OWL만으로는 표현할 수 없었던 규칙(Rule)을 표현할 수 있도록 확장한 것이다 [1].

이러한 온톨로지 언어의 확장에 따라 이를 추론하기 위한 추론엔진에 대한 연구가 활발히 진행되고 있다. 본 논문에서는 이러한 온톨로지 언어의 확장에 따라 이를 추론하기 위한 연구로써 OWL언어와 SWRL언어 모두를 Jess추론엔진을 이용하여 추론하기 위한 플랫폼을 제안한다. 그리하여 기존에 OWL을 추론하기 위해 사용하던 Racer를 제거하고

Rule-Based 전문가 시스템인 Jess를 이용함으로써 온톨로지 언어의 확장에 따른 추론 기능 향상을 도모 하고자 한다.

다음 2장에서는 온톨로지의 개념, 온톨로지에서 사용하는 언어, 이를 추론하기 위한 연구와 온톨로지언어를 추론엔진에 적용하기 위한 변환에 대하여 살펴본다. 3장에서는 본 논문에서 제안하는 OWL과 SWRL을 통합추론하기 위한 여러 가지 방안에 대해 논하고 이들을 비교 분석한 결과에 대해, 마지막 4장에서는 본 논문의 결론과 향후 연구에 대해서 논한다.

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 시맨틱 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 언어

접수일자 : 2005년 10월 21일

완료일자 : 2005년 12월 5일

감사의 글 : 본 연구는 정보통신부의 유비쿼터스컴퓨팅 및네트워크원천기술개발사업에 의하여 지원되었으며 이에 감사드립니다.

* 교신저자

를 각종 규칙기반 추론엔진(Rulebased 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 표현하는데 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 [그림2]와 같은 형태의 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 예제
Fig. 1. An Example of SWRL Rule

```
<swrl:Imp rdf:ID="Rule-3">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument1 rdf:resource="#x1"/>
          <swrl:propertyPredicate rdf:resource="#hasUncle"/>
          <swrl:argument2 rdf:resource="#x3"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest rdf:resource
        ="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil" />
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest rdf:resource
            ="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          <rdf:first>
            <swrl:IndividualPropertyAtom>
              <swrl:argument1 rdf:resource="#x2"/>
              <swrl:argument2 rdf:resource="#x3"/>
              <swrl:propertyPredicate rdf:resource="#hasBrother"/>
            </swrl:IndividualPropertyAtom>
          </rdf:first>
        </swrl:AtomList>
      </rdf:rest>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument1 rdf:resource="#x1"/>
          <swrl:propertyPredicate rdf:resource="#hasParent"/>
          <swrl:argument2 rdf:resource="#x2"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:body>
</swrl:Imp>
```

그림 2. SWRL 예제
Fig. 2. An example of SWRL

2.3 온톨로지의 추론

온톨로지의 추론은 Tim Berners-Lee가 제안한 시멘틱 웹의 스택구조상에서 온톨로지의 바로 위 단계에 해당하며 이러한 온톨로지의 추론을 위한 로직의 표현 방법에 대한 각각의 표현 범위는 [그림3]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. 다시 말하면 기존의 OWL언어는 DL기반의 언어이므로 DL이 표현할 수 없었던 LP(Logic Programs)의 일부분인 Horn Logic의 정보를 표현할 수 있도록 확장된 언어가 SWRL이라고 할 수 있겠다.

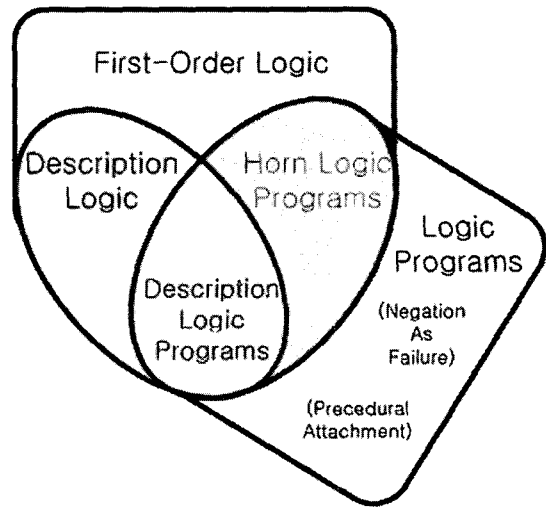


그림 3. 논리 지식의 표현범위[3]
Fig. 3. Representation Power of Logic Knowledge

SWRL을 통하여 추론 기능이 확장 된 온톨로지의 예를 들면, 다음 [그림 6]와 같은 Family온톨로지를 예로 들 수 있다. 이 예는 OWL온톨로지와 이를 이용한 SWRL Rule을 정의한 것이다. 이렇게 구성한 OWL온톨로지와 SWRL Rule을 이용해서 온톨로지 상의 개체(Individual)에 대한 추론이 가능하다. 여기서는 DL추론을 위하여 Racer를 이용하였고 SWRL의 추론을 위하여 Jess를 이용하였다. 이 온톨로지는 [그림4]과 [그림5]에서 나타내듯이 Man과 Woman클래스에 각각 남자 10명(M1~M10)과 여자 10명(F1~F10)의 개체를 [그림5]의 오른쪽 부분에 해당하는 관계(Relation)를 설정하여 초기화 하였다. [그림4]은 각 개체들 간의 관계이고 [그림 5]는 하나의 Person개체에서의 관계(Relation)가 그림 원편의 15개의 관계들을 갖게 되지만 개체를 초기화 할 때는 HasConsort, HasSex, HasChild만을 정의하여 나머지 관계에 대한 정보를 추론을 통하여 얻어낼 수 있도록 한다. 이를 이용한 추론 결과는 [표 1]과 같다. 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].

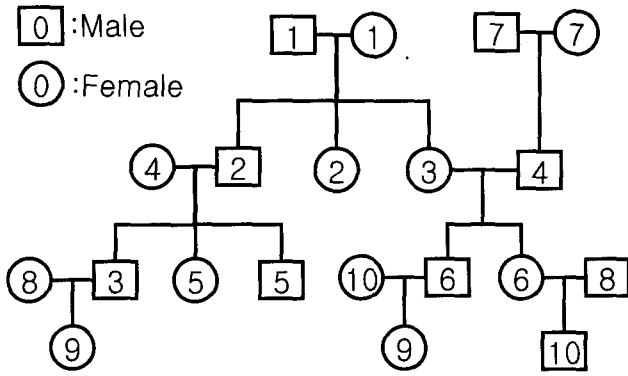


그림 4. 초기화 개체들 간의 관계도

Fig. 4. Relations between Family Ontology's individual

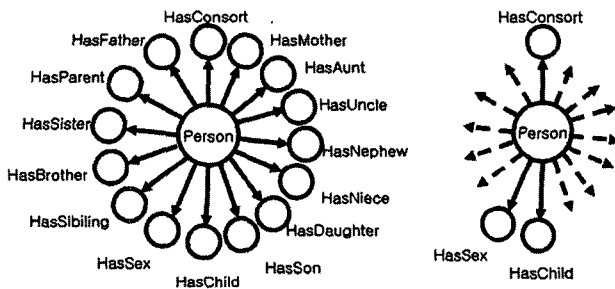


그림 5. Person의 Relations
Fig. 5. Relations of a Person

OWL ontology

- 1- Person := Man \cup Woman
- 2- Parent := Person \cap hasChild ≥ 1
Child := Person \cap hasParent ≥ 1
(hasChild)⁻¹ = hasParent
- 3- Father:=Parent \cap Man Mother:=Parent \cap Woman
- 4- Son:=Child \cap Man Daughter:=Child \cap Woman
- 5- Brother:=Sibling \cap Man Sister:=Sibling \cap Woman
- 6- Nephew := Man \cap (hasUncle $\geq 1 \cup$ hasAunt ≥ 1)
- 7- Relative := Child \cup Parent \cup Aunt \cup Nephew
 \cup Niece \cup Uncle \cup Sibling

SWRL rules

Initial rule base

- 8- hasParent(?x1,?x2) \wedge hasConsort(?x2,?x3)
 \Rightarrow hasParent(?x1,?x3)
- 9- hasParent(?x1,?x2) \wedge hasSister(?x2,?x3)
 \Rightarrow hasAunt(?x1,?x3)
- 10- hasParent(?x1,?x2) \wedge hasParent(?x3,?x2)
 \wedge differentFrom(?x1,?x3) \Rightarrow hasSibling(?x1,?x3)
- 11- hasSibling(?x1,?x2) \wedge hasDaughter(?x2,?x3)
 \Rightarrow hasNiece(?x1,?x3)

Rules mirroring the ontology knowledge

- MR1- hasSibling(?x1,?x2) \wedge Man(?x2)
 \Rightarrow hasBrother(?x1,?x2)
- MR2- hasSibling(?x1,?x2) \wedge Woman(?x2)
 \Rightarrow hasSister(?x1,?x2)
- MR3- hasParent(?x1,?x2) \wedge Man(?x2)
 \Rightarrow hasFater(?x1,?x2)

[그림 6] Family OWL 온톨로지와 SWRL Rule
Fig. 6. A Family OWL Ontology and SWRL Rule

[표 1] RACER와 Jess를 통한 추론 결과

Table 1 A result of RACER and Jess Inference

| 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,M6F2,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] |

2.4 추론을 위한 온톨로지 언어의 변환

마지막으로 위와 같이 OWL과 SWRL을 추론하는데 있어서 SWRL을 Jess언어로 변환하는 방법에 관해 논하겠다. Jess는 새로운 Fact들이 들어올 경우 이미 정의해 놓은 Rule에 의해 추론이 되는 Foward-Chaining 방식 추론이 가능한 전문가 시스템이다. 이러한 Jess를 온톨로지 상에서 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) ^ swrlb:greaterThan(?age2,?age1)
-> hasOlderBrother(?x, ?y)
    
```

그림 7. SWRL Rule 예제
Fig. 7. An example of 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 8. Jess Rule로 변환된 SWRL Rule 예제
Fig. 8. An example of converted Jess Rule from SWRL Rule

위 [그림 7]은 SWRL을 이용하여 정의한 Rule이다. 이러한 SWRL의 Rule을 Jess언어의 Rule형태로 변환한 것이 [그림 8]이다. 이러한 변환을 위하여 논문[5]에서는 작성된 SWRL Rule을 Jess를 이용하여 추론하고 그 결과를 조회하기 위해 SWRL Factory라는 컴포넌트를 개발하였다. SWRL Factory는 SWRL의 규칙을 다루기 쉽도록 Java API를 제공한다.

3. 통합 추론

앞에서 논한 내용과 같이 SWRL언어를 이용하여 기존에 OWL언어로 표현할 수 없었던 규칙들을 표현 할 수 있게 되었다. 따라서 OWL과 SWRL을 모두 사용하여 기술된 온톨로지를 추론하기 위해서는 OWL과 SWRL을 통합 추론하는 방법이 필요하다. 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 접근 방법을 제안한다. 첫 번째 방법은 OWL을 추론하기 위한 도구인 OWLJessKB[8]를 OWL을 변환하기 위해 사용하고 SWRL을 변환하기 위한 부분을 추가하는 방법, 두 번째 방법은 SWRL의 변환을 위해 SWRL Factory를 이용하고 이에 OWL을 변환하기 위한 부분을 추가하는 방법이며 마지막은 OWLJessKB와 SWRL Factory모두를 사용하는 방법이다.

3.1 OWLJessKB를 이용한 통합추론

첫 번째로 기존에 OWL을 Jess를 이용하여 추론하도록 구현된 OWLJessKB를 OWL과 SWRL의 통합추론을 위하여 이용하는 방법이다. OWLJessKB는 OWL의 개체들을 모두 RDF의 Triple 형태로 변환한 후, 이를 Jess의 Fact로 넣어 이를 추론하게 된다. [그림 9]에 이를 좀 더 자세히 나타내었다. 첫 번째로 ①은 OWLJessKB의 객체를 생성하면 OWL Role을 TripleCollector를 이용하여 Triple형태로 변환하고 이를 Jess의 Fact와 Rule로 삽입한다. 다음으로 ②은 TripleCollector가 OWL로 기술된 온톨로지를 읽어 들여 OWL의 Concept, Role, Individual을 Triple형태로 변환하여 Jess에 Fact와 Rule로 입력하게 된다. 마지막으로 ③은 이렇게 입력된 Fact와 Rule을 이용하여 Jess를 통한 추론을 수행하게 된다. 이렇게 추론된 결과는 Jess-Query를 이용하여 조회 할 수 있다.

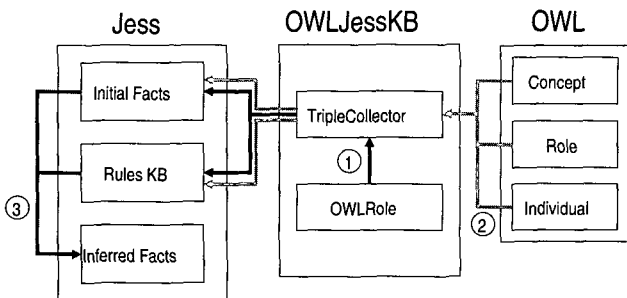


그림 9. OWLJessKB의 추론과정
Fig. 9. The inferencing process of OWLJessKB

이와 같은 OWLJessKB를 이용하면 OWL온톨로지를 Jess로 변환할 수 있으므로 이에 SWRL을 Jess의 Rule로 변환하기 위한 부분을 추가하기 위하여 [그림 7]과 같은 HotJess를 설계하였다. 이는 OWLJessKB를 통하여 변환된 OWL언어와 SWRL Rule을 SWRLConver를 이용, Jess언어로 변환하여 OWL과 SWRL모두 통합 추론을 수행할 수 있도록 한다.

[그림 10]에서 ②과정과 SWRLConver토는 본 연구를 통하여 구현해야 할 부분에 해당한다. 이 부분은 Jess가 OWLJessKB와 OWL로부터 변환한 결과와 함께 기존의 OWL정보를 이용하여 SWRL의 변환을 수행 할 수 있도록 SWRL Rule을 Jess언어에 맞게 변환하도록 한다. OWL JessKB는 OWL 언어를 RDF의 Triple형태로 변환하고 이를 다시 Jess언어로 변환하여 추론을 수행한다. 때문에, Jess를 통한 SWRL의 추론을 위해서 OWL이 변환되는 형태인 Triple 형태와 함께 추론할 수 있도록 변환해야 한다.

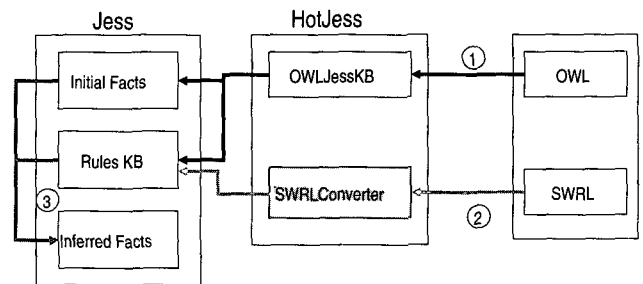


그림 10. OWLJessKB를 이용한 통합 추론 설계
Fig. 10. An inferencing architecture using OWLJessKB

3.2 SWRL Factory를 이용한 통합추론

다음은 SWRL Factory를 이용하는 방법이다. SWRL Factory는 SWRL의 Rule을 Jess를 이용하여 추론하고 그 결과를 Protégé를 통해 확인 할 수 있다. 다음 [그림 11]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.

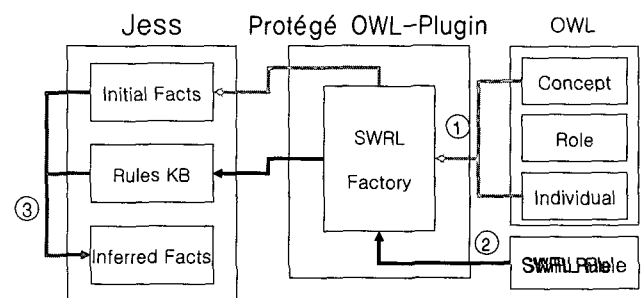


그림 11. SWRL Factory의 추론과정
Fig. 11. The inferencing process of SWRLFactory

SWRL Factory는 OWLJessKB와는 달리 OWL을 Triple로 변환하여 수행하지 않고, 단지 Concept을 Jess의 Template, 그리고 Individual을 Fact로 변환하게 된다. 즉, SWRL Factory는 SWRL Rule의 변환을 위해 OWL의 Concept과 Individual에 대한 변환만을 수행한다. 때문에, SWRL Factory를 이용하여 통합 추론을 하기 위해서는

OWL온톨로지를 Jess로 변환하는 부분이 추가적으로 필요하다.

[그림 12]은 SWRL Factory를 이용하여 통합 추론하기 위한 흐름을 나타낸다. ①과 OWLConver는 본 연구를 통하여 구현해야 할 부분으로, SWRL Factory가 변환 작업을 수행할 수 있도록 OWLConvert를 구성하여 SWRL Factory에서 사용하는 변환 방식과 같은 방법으로 OWL을 변환하여 추론을 수행하도록 한다. 그 후, ②는 SWRL Factory를 이용하여 SWRL을 변환하고 이에 대한 추론을 수행하게 된다. 이와 같이 구성 할 경우 OWL의 정보들은 OWLConver에 의해 변환 되기 때문에 SWRL Factory에서 OWL을 읽어 오기 위한 부분 없이 SWRL Rule 만을 Jess Rule로 변환하여 이를 추론하게 된다.

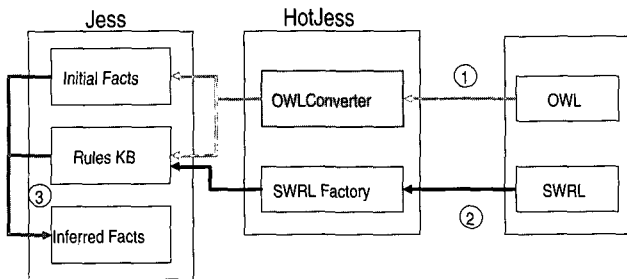


그림 12 SWRL Factory를 이용한 통합 추론 설계
Fig. 12. An inferencing architecture using SWRLFactory

3.3 OWLJessKB + SWRLFactory를 이용한 통합추론

마지막 세 번째 방법으로는 OWLJessKB와 SWRL Factory를 모두 이용하는 것이다.

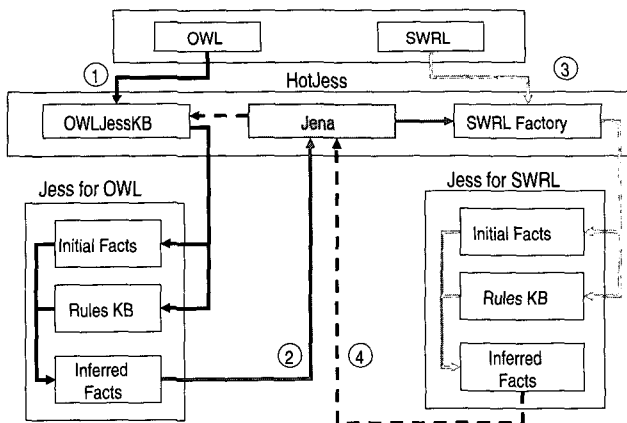


그림 13. OWLJessKB와 SWRL Factory이용 추론
Fig. 13. The inference using OWLJessKB and SWRLFactory

[그림 13]에서 보듯이 OWLJessKB와 SWRL Factory를 모두 이용한다. 이는 ① OWLJessKB가 OWL을 읽어 들여 추론을 수행하고 ② 수행한 결과를 Jena[9]를 통해 SWRL Factory에 설정한다. 다음으로 ③ SWRL Factory는 OWL JessKB에 의해 수행된 추론 결과와 SWRL Rule을 이용하여 추론을 수행하게 된다. 마지막 ④에서는 OWL과 SWRL에 의해 추론되어 생성된 결과(Fact)들을 Jena를 통하여 다시 OWLJessKB언어로 변환하게 한 후 이를 다시 추론하도록 한다. 이와 같이 구성 할 경우 OWLJessKB와 SWRL Factory에서 OWL의 Concept와 Role, Individual 을

Jess 언어로 변환하는 방식이 다르기 때문에 OWLJessKB를 이용하여 추론 후에 이에 대한 추론 결과와 SWRL로 부터 또 다른 Jess엔진을 이용하여 추론을 수행해야만 한다. 또한, 기존의 첫 번째와 두 번째 방식과는 다르게 점선으로 나타내어진 ④은 SWRL에 의해 추론된 결과를 다시 OWL의 정보를 통한 추론을 수행하게끔 한다. 이는 SWRL을 추론한 결과를 다시 OWL에 적용하여 추론하여 또 다른 새로운 사실(Fact)들을 얻어내기 위함이다. 예를 들면, [그림 14]의 경우 "C"라는 사실(Fact)이 Jess로 들어왔을 경우 두 개의 추론엔진을 통해서 추론되는 결과는 만족스럽지 못하다. 때문에 [그림 15]와 같이 이 결과를 포함하여 다시 추론을 수행하게 되면 원하는 추론 결과를 얻어낼 수 있다. 이처럼 SWRL의 추론 결과를 다시 OWL을 통해 추론하고 새로 추론되는 결과(Inferred Fact)가 없을 때 까지 이를 반복하지 않는다면, 원하는 추론 결과가 나오지 않을 수 있기 때문이다.

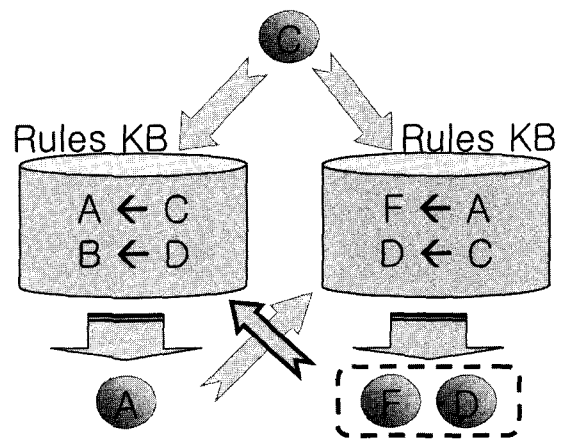


그림 14. 1차 추론 결과
Fig. 14. The result of first inference

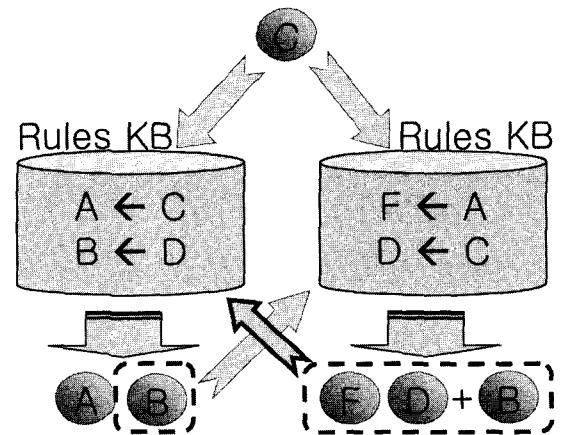


그림 15. 2차 추론 결과
Fig. 15. The result of second inference

3.4 비교분석

앞에서 논한 세 가지 통합 추론의 특징다음과 같다. 첫 번째 방법은 OWLJessKB를 이용하고 SWRL을 위한 SWRL Converter를 구현하여 OWL과 SWRL을 통합한 추론이 가능하도록 한다. 두 번째는 SWRL Factory를 이용하고 OWLConverter를 구현하는 방법이고, 마지막으로 OWLJessKB와 SWRL Factory를 모두 이용하는 방법이다.

첫 번째 방법을 살펴보면, OWLJessKB가 OWL을 추론하기 위해서는 Triple의 개념을 사용한 변환이 이루어지기 때문에 SWRL을 추론하기 위해서 또한 OWL의 Triple 형태를 추론할 수 있도록 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 때문에 OWL을 위한 Jess, 그리고 SWRL을 위한 Jess를 따로 구성하여 OWL에 대한 추론 결과를 Jena를 통하여 SWRL추론시 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 있을 뿐만 아니라, 각각의 추론결과(Inferred Fact)를 서로 피드백하여 다시 변환하고 추론하는 형태로 수행되기 때문에 추론의 수행속도를 보장할 수 없다는 문제가 존재할 수 있다.

OWL과 SWRL을 통합하여 추론하기 위해서 추가적으로 수행해야 할 내용은 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.

4. 결 론

본 논문에서는 온톨로지의 언어와 이의 추론에 대하여 논하였고 OWL과 SWRL을 통합 추론하기 위해 어떠한 틀을 사용하고 이를 활용하여 통합 추론을 수행할 지에 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.

참 고 문 헌

- [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, I.Horrocks, R.Volz, S.Decker "Description Logic Programs: Combining Logic Programs with Description Logic," International World Wide Web Conference Proceeding of the 12th International Conference, Budapest, Hungary, p.48~57, 2003
- [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/index.html>)
- [8] OWLJessKB : A Semantic Web Reasoning Tool (<http://edge.cs.drexel.edu/assemblies/software/owljesskb/>)
- [9] Jena2 - A Semantic Web(<http://www.hpl.semweb/jena2.htm>)

저 자 소 개



이기철(Ki-Chul Lee)

2003년 : 명지대 산업공학 학사
 2003년 : 명지대 컴퓨터공학 학사
 2005년 : (주)MECA XML팀
 2005년~현재 : 성균관대 컴퓨터공학과 석사과정

관심분야 : 온톨로지, 추론시스템, 논리언어

Phone : +82-31-290-7987

E-mail : lee77@skku.edu



이지형(Jee-Hyoung Lee)

1993년 : 한국과학기술원 전산학과 학사
 1995년 : 한국과학기술원 전산학과 석사
 1999년 : 한국과학기술원 전산학과 박사
 2000년 : 한국과학기술원 첨단정보기술 연구센터 위촉 연구원

2002년 : 미국 SRI International, International Fellow

2002년~현재 : 성균관대학교 정보통신공학부 조교수

관심분야 : 지능시스템, 기계학습, 온톨로지

Phone : +82-31-290-7154

E-mail : jhlee@ece.skku.ac.kr