

## 추론을 위한 OWL-DL과 SWRL의 효율적 결합

서은석<sup>o</sup> 박준상 박영택

송실대학교 컴퓨터학과

erin214@ailab.ssu.ac.kr<sup>o</sup>, theartiest@ailab.ssu.ac.kr, park@computing.ssu.ac.kr

### Efficient Combination of OWL-DL and SWRL for Maintaining Decidability

Eun-Seok Seo<sup>o</sup> Jun-Sang Park Young-Tack Park

Department of Computing, Soongsil University

#### 요약

유비쿼터스 컴퓨팅 시대의 도래와 시맨틱 웹에 대한 관심이 높아짐에 따라 관련 기술인 온톨로지와 이를 이용한 추론 기술에 대한 요구가 증가하고 있다. 따라서, 추론이 가능한 시맨틱 웹 기반의 모델링과 추론에 대한 연구가 필요하다. 모델링을 위해 사용되는 OWL-DL과 임의의 사용자 규칙을 표현하는 SWRL은 각각 W3C의 표준안으로서, 유비쿼터스 컴퓨팅 환경에 효율적으로 자동적인 개인화 서비스[1][2]를 제공하는데 있어서 적합하다. 그러나 OWL-DL과 SWRL의 단순한 결합은 질의응답(Query Answering)에 대한 처리가 비결정 가능한(undecidable) 문제를 야기한다.

본 논문에서는, 비결정가능성 문제의 원인인 무한반복의 가능성을 제거하기 위한 블록(blocking) 방법을 제안한다. OWL-DL이 지닌 서술논리(Description Logic)의 표현력을 유지하고, 그에 따른 추론의 질적인 성능을 유지하는 범위에서 블록방법을 사용하여 결정 가능한 질의응답을 수행하는데 궁극적인 목적을 두고 있다. OWL-DL의 TBox에 위치하는 존재 정량자(Existential Quantifier)를 대체하고 ABox에 삽입하여, 무한반복의 가능성을 없애는 해결 방법을 제시한다. 실험은 비결정가능성 문제를 DL-Safe 규칙을 통해 해결한 KAON2와 비교하여 진행한다.

실현은 비결정가능성 문제를 DL-Safe 규칙을 통해 해결한 KAON2와 비교하여 진행한다.

#### 1. 서론

온톨로지 기반 추론에서 효과적인 추론과 온톨로지의 표현력을 효율적으로 처리하기 위해서는, OWL-DL과 규칙을 표현할 수 있는 SWRL[3]을 통한 확장이 필요하다. OWL-DL은 서술 논리의 SHOIN(D)으로 표현이 이루어질 수 있다. 즉, OWL-DL의 모든 개념과 속성들이 SHOIN(D)와의 완벽한 매핑이 가능하다. SHOIN(D)는 수학적으로 결정 가능한 알고리즘을 구성할 수 있음이 여타 관련 논문들에서 증명되었다. 즉, SHOIN(D)로 구성된 모델을 바탕으로 추론 시스템을 구축하여, 어플리케이션에 적용이 가능하다. SWRL은 혼 유사 규칙으로 구성되어 있다. 혼 유사 규칙은 일차술어논리의 부분집합 형태이고, 이는 표현력은 다소 떨어지지만 결정 가능한 성질을 갖는 논리이다. 이러한, 두 결정 가능한 논리의 결합은 유감스럽게도 비결정 가능한 결과를 야기한다. 다른 특성을 지닌 두 논리의 결합이기에, 복잡한 원인에서 문제가 발생한다.[4]

본 논문에서는 OWL-DL과 SWRL의 결합에 의해 발생하는 비결정가능성 문제를 해결함과 동시에 표현력을 잃지 않는 높은 성능의 추론을 수행하는데 목적이 있다. OWL-DL과 SWRL을 결합하기 위해서는 논리의 형태를 단일화 시켜주어야 한다. OWL-DL은 서술논리의 SHOIN(D)로 구성되며, SWRL은 혼 유사 절로 구성된다. SHOIN(D)는 classification 작업을 통해 혼 절로 변환이 가능하다. SHOIN(D)은 존재 정량자와 전체 정량자(Universal Quantifier)를 모두 지원하지만, 혼 절은 폐쇄

세계 가정(Closed World Assumption)에 기반으로 하여 전체 정량자만을 사용하여 논리 구문을 구성 가능하다. Skolemization 과정을 거쳐 혼 절에서 허용하지 않는 존재 정량자를 대체하여 몇 개의 절로 구분하여 표현할 수 있다. 그러한 변환과정을 거쳐 OWL-DL은 SWRL과 유기적인 동작을 할 수 있는 혼 절로 표현된다. 본 논문은 혼 절로의 변환은 가능하다는 전제 하에 진행되며, 비결정가능성 문제 해결에 대한 방법을 강구한다.

본 논문에서는 비결정가능성 문제를 임의의 대안 개체(Individual)를 생성하는 블록(blocking)방법을 제안하여 해결하고자 한다. TBox에서 발생할 수 있는 무한반복의 가능성을 제거하기 위하여, TBox의 존재 정량자를 ABox에 적합한 개체들로 대체 한다. 궁극적으로 높은 성능의 추론이 가능한 풍부한 표현력을 유지하면서, 결정 가능한 효율적인 질의응답이 가능한 온톨로지를 구성할 수 있게 된다. 시맨틱 웹을 활용하여 서비스 어플리케이션을 구현하는데 있어서, 효율적인 시간 내에 서비스의 제공이 이루어질 수 있다고 확신한다.

#### 2. 관련연구

##### 2.1 Racer

RACER(Renamed ABox and Concept Expression Reasoner Professional)[7]는 OWL 기반의 시맨틱 웹을 관리하는 시스템이다. OWL로 시맨틱 모델[6]을 생성하고, 수정하고, 추론할 수 있는 구조를 가지고 있다. 대용량의 TBox에 대한 클래스 분류(Classification)와 일관성

체크(Consistency Checking)를 사용자 요구에 적합한 합리적인 시간 내에 처리할 수 있는 장점을 지니고 있다. SWRL의 확장을 지원하며, 자체적인 문법을 사용한 nRQL(new Racer Query Language)를 사용하여 질의응답을 수행한다. ABox 추론은 Tableau 알고리즘[5]을 기반으로 이루어진다. TBox의 추론에서는 상당한 성능을 나타내지만, 대용량의 ABox에서는 처리 속도가 제한적인 문제점을 가지고 있다.

2.2 Pellet

Pellet[8]은 Tableau 알고리즘을 기반으로 개발된 OWL-DL 추론 엔진이다. Pellet은 Nominal을 포함한 OWL-DL의 모든 추론을 지원한다. SHOIQ을 위한 결정 과정(Decision Procedure)이 정당(sound)하고 완전(complete)함을 보장한다. SWRL 규칙을 지원하며, SPaRQL(Simple Protocol and RDF Query Language)을 통한 질의응답이 가능하다. Pellet의 두드러진 특징은 데이터타입(Datatype) 추론[9]을 완벽히 구현하였다는 것이다. [그림 1]에서 보듯이, Tableau 알고리즘 기반 추론엔진에 별도의 XSD 추론엔진을 추가하여 데이터타입에 대한 처리를 수행하고 있다.

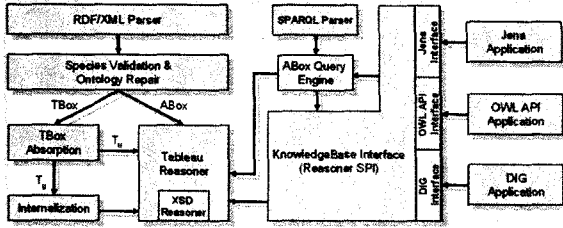


그림 1 Pellet의 시스템 구조

2.3 KAON2

KAON2는 독일의 Karlsruhe 대학에서 연구한 OWL-DL 추론 엔진이다. 기존 추론에 많이 사용되는 Tableau 알고리즘을 사용하지 않고, 선언적 데이터 로그(Disjunctive Datalog) 기법을 사용하여 추론을 수행한다. SWRL의 확장을 통한 추론을 지원하며, 질의응답은 SPaRQL을 채택하였다. 비결정 가능한 질의응답 처리 문제를 DL-Safe 규칙을 적용하여 해결하였다.[10]

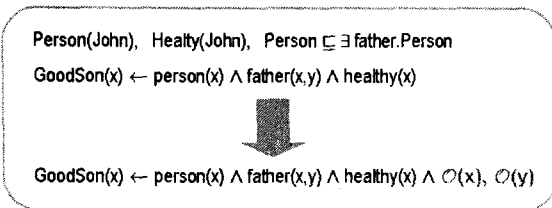


그림 2 DL-Safe 규칙의 예

[그림 2]는 DL-Safe 규칙처리의 예를 보이고 있다. 존재 정량자에서 발생할 수 있는 무한반복의 가능성을 규칙을 제한하는 방식의 블록을 적용하여 제거하고 있다. 기존 SWRL의 규칙에 O(x)와 O(y)를 추가하여 DL-Safe한 규칙으로 변환한다. 규칙의 조건부에 존재하는 모든 변수는 지식베이스에 명시된 사실만이 값으로 할당될 수 있다는 의미로 정의된다. 즉, DL-Safe한 규칙의 사용은 ABox 에 존재하는 사실만을 추론할 수 있는 단점이 남게 된다. 본 논문에서는 이러한 단점을 극복하고 결정 가능한 추론을 할 수 있는 방법을 제안하고자 한다.

3. 비결정 가능성 해결 기법

관련연구의 KAON2는 SWRL을 지원하지만, 다소 표현력을 잃어 좋은 추론을 수행하지 못한다. 본 연구에서는 OWL-DL과 SWRL 결합 시에 결정가능성을 보장하고, KAON2에서 해결하지 못한 Unsafe Rule을 지원함으로써 OWL-DL의 규칙적인 표현력과 SWRL의 풍부한 표현력을 모두 고려한 추론을 가능하도록 하였다. 이로써 결정 가능한 시간 내에 사용자 질의응답 또한 보장된다고 볼 수 있다.[11]

3.1 SWRL의 표현력

SWRL은 온톨로지 표현 언어인 OWL의 단점을 극복하고자 제안되었다. OWL에서 배제된 규칙의 정의를 SWRL을 추가한 확장을 통해 이를 수 있다. SWRL은 혼절[12]을 기반으로 하여 구성된다. 혼절을 기반으로 하기 때문에 전체 정량자만이 허용되며 존재 정량자는 표현하지 않는다. 혼절은 한 개 이하의 긍정적 논리구(Positive Literal)만을 갖는 절을 의미한다. 한 개 이하의 긍정적 논리구가 존재하고 나머지는 모두 부정적 논리구일 경우, 결의(resolution)에 의해 논리구문에 대한 만족여부가 결정되어 질 수 있게 된다. 결국 결정 가능한 알고리즘에 의해 처리되어질 수 있는 논리이다.

[그림 3]에서 논리의 표현력을 고려한 포함 관계를 나타내고 있다. 일차술어논리는 표현력이 풍부한 특성상 결정 가능한 알고리즘을 갖는다는 증명이 되지 않는다. 그에 포함된 논리인 서술논리와 혼 논리는 각각 일차술어논리에서 표현력을 제한하여 결정 가능한 알고리즘을 통해 처리가 가능한 속성을 지닌다.

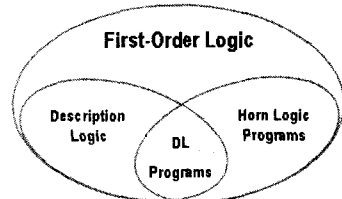


그림 3 논리의 표현력을 고려한 포함관계

SWRL은 논리구문에서 정의하는 변수를 존재 정량자

로 제한하지 못하므로, 지식베이스(Knowledge Base)에 명백히 존재하지 않는 사실에 대한 도출은 수행하지 못한다. 전체 정량자만이 허용되어, 계산 시에는 무한반복과 같은 문제가 발생할 여지가 없다는 장점이 있으나, 표현력이 부족하여 높은 질의 결과 도출이 힘들다는 단점을 지니고 있다.

표 1 SWRL 표현력 예제

지식 베이스 (Knowledge Base)	(1) person(john)
	(2) person(cain)
	(3) health(john)
	(4) father(cain, john)
규칙 (SWRL)	goodson(X) ← person(X) ∧ father(X,Y) ∧ healthy(X)

[표 1]은 SWRL의 부족한 표현력을 예로 보이고 있다. 지식 베이스에는 "john" 이 "person" 타입이고, "john" 은 "health" 타입이라는 인스턴스(instance)가 삽입이 되어 있다. 즉, "john은 사람이고, 건강하다."를 의미하는 정보가 지식베이스에 기술되어 있다. 규칙에는 "person" 타입의 변수 "X"가 있고, 변수 "X"는 "father"의 속성에 의해 변수 "Y"를 가지고, 변수 "X"가 "health" 타입의 개체이기도 하면, 변수 "X"는 "goodson" 타입의 변수를 가진다는 것이 기술되어 있다. 즉, "건강한 아들은 좋은 아들이다."를 의미하는 정보가 규칙으로 표현되어 있다. 지식베이스와 규칙을 엔진에서 구동하면 "john이 goodson이다"라는 사실을 도출해 내지 못한다. 왜냐하면 "john이 아버지를 가진다."라는 사실이 지식베이스에 명백히 명시되어 있지 않기 때문이다. 사람이려면 누구나 아버지를 가지지만 SWRL만으로는 이와 같은 상황을 표현하기 힘들고, 이러한 추론에 약점을 지니게 된다. 약점을 보완하기 위해서는 OWL-DL에서 표현 가능한 존재 정량자를 활용해야만 한다.

3.2 OWL-DL의 표현력

OWL-DL은 SHION(D)으로 표현된다. SHION(D)는 [그림 4]와 같이 트리형태의 모델로 구성이 가능하다. DL은 변수가 명시 되지 않지만, 논리를 처리할 때에는 오른쪽 트리와 같이 변수가 있음을 가정하여 모델을 구성한다. 그리고 DL의 특성상 그러한 변수들은 노드(node)로 표현되며 트리의 모습과 유사한 방식으로만 역할(role)에 의해 연결이 되어 있기 때문에 다음과 같이 트리형태의 모델을 지니게 된다.

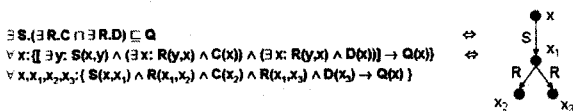


그림 4 트리형태의 모델로 표현 가능한 DL

트리형태의 모델은 유한한 트리 모델(Finite Tree

Model)과 무한한 트리 모델(Infinite Tree Model)로 구분할 수 있다. 트리형태의 모델은 노드들을 연결하는 역할의 속성에 의해 종료가 되지 않는 무한한 트리 모델을 구성할 수 있다. 노드의 역할이 추이적인(Transitive) 속성을 지닐 경우 그러하다. 무한한 트리 모델이 발생하면 블록(blocking)을 사용하여 유한한 트리 모델로 변환할 수 있다. 트리 형태라는 규칙적이고 제한적인 성격을 고려하여, 반복 가능한 역할에 대한 제제를 가하여 무한한 반복을 사전에 차단하는 기능을 구성할 수 있다.

표 2 OWL-DL의 무한반복 가능성 예제

SHOIN(D)	의미(Semantics)
person(john)	john은 사람이다.
person ⊆ ∃ father.person	사람은 사람인 father를 가진다.
∃ father.( ∃ father.person) ⊆ grandchild	사람인 father의 father를 갖는 것은 grandchild이다.

[표 2]는 종료가 되지 않는 문제를 발생할 수 있는 예제를 보이고 있다. 두 번째 SHOIN(D) 구문을 보면, "person이면 적어도 한명 이상의 person을 아버지로 갖는다."와 같은 의미를 지닌 문장이 있다. DL의 특성상 존재 정량자를 허용하여 표현 가능한 문장이다. 이는 굉장히 뛰어난 추론을 할 수 있는 요소이자 동시에 여러 문제를 야기한다. 도메인(Domain)과 레인지(Range)가 같은 타입의 "person" 클래스 이고, 속성인 "father"는 존재 정량자로 제한이 되어 있으므로, 무한한 반복의 가능성이 남아 있게 된다. 이와 같이, 프로퍼티가 추이적인 속성을 지닐 때, 이와 같은 문제가 발생하게 된다.

3.3 비결정 가능성의 원인

OWL-DL에서 발생할 수 있는 무한반복은 단순 블록 방법에 의해 해결할 수 있다. 하지만 이러한 블록은 모델이 트리형태를 유지할 때만이 가능하다. 트리형태가 깨지게 되면, 규칙성이 존재하지 않으므로 또 다른 무한 반복의 여지가 생기게 되고 단순한 블록이외의 다른 해결책이 제시되어야만 한다.

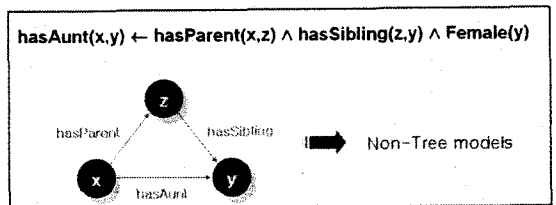


그림 5 hasAunt를 정의하는 규칙의 모델

SWRL은 트리형태의 모델로 구성된다고 보장되지 않는다. 트리형태의 모델은 루트(Root) 노드와 리프(Leaf) 노드가 존재하고, 루트 노드와 리프 노드는 직접적으로 연결되어서는 안 되는 특징을 가진다. [그림 5]를 보면

알 수 있다. "hasAunt"를 정의 하는 규칙은 닫힌 삼각형 태의 모델로 표현된다. 그로 인해, OWL-DL에 SWRL을 확장하게 되면, 트리형태의 모델을 잃게 된다. 즉, 보편적인 방법의 블록으로는 무한반복을 제거할 수 없게 된다는 의미이다. 무한반복을 제거할 수 없으므로 두 논리의 결합은 질의응답에 대한 결정 가능한 결과를 도출해 낼 수 없는 문제를 야기하게 된다.

3.4 무한반복의 제거를 위한 블록(Blocking) 방법

본 연구에서는 Unsafe 규칙을 지원하면서, 결정 가능한 질의응답이 가능하도록 하는 블록 방법을 제안한다. KAON2에서 제안한 DL-Safe 규칙과는 달리, ABox에 명백히 존재하지 않는 사실에 대한 추론을 가능하게 하여 높은 질의 추론을 수행할 수 있는 블록 방법을 사용한다. 존재 정량자의 의미를 살린 추론이 가능하다. 존재 정량자가 Classification 과정에서 검색되면, 임의의 상수를 대입하여 대안 개체를 생성하는 기법을 사용한다. 즉, DL의 표현력을 손절에서 유지할 수 있게 된다.

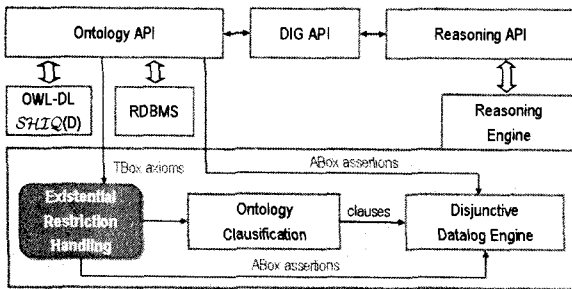


그림 6 블록 모델이 포함된 시스템 구조

[그림 6]은 본 논문에서 제안하는 시스템의 구조를 보이고 있다. 온톨로지를 구성하는 OWL-DL을 온톨로지 API를 통해 파싱(Parsing)하여 TBox와 ABox로 구분한다. 구분된 지식베이스는 각각 존재 정량자 핸들러(Existential Restriction Handler)와 데이터로그 엔진(Disjunctive Datalog Engine)으로 입력된다. ABox는 별도의 처리 없이 데이터로그에 질의 형태로 입력되어진다. TBox는 존재 정량자 핸들러를 통해 존재 정량자와 관련된 구문을 찾아서 정해진 규칙에 따라 처리되어 ABox 형태로 변환된다. 변환된 ABox는 데이터로그 엔진에 입력된다. 존재 정량자와 관계없는 다른 TBox의 구문들은 다음 단계인 온톨로지 Classification 과정에서 데이터로그에서 처리할 수 있는 질의 형태로 변환되어진다.

3.4.1 DL에서 일차술어논리의 매핑(Mapping)

OWL-DL과 SWRL의 논리의 형을 일치시키기 위해서는, Classification 과정이 필요하다. [그림 7]을 보면, DL의 클래스 컨스트럭터(Class Constructors)가 손절로 매핑 되어짐을 알 수 있다. 다른 컨스트럭터들은 의미의 정확성을 유지하면서 변환이 가능하지만, 존재 정량자는

완벽한 변환이 힘들다. 조건절에 존재 정량자가 위치할 경우에 그러하다. 반대로 결론절에 전체 정량자가 위치하게 될 경우에도 DL에서 일차술어논리로의 완벽한 변환이 가능하지 않다.

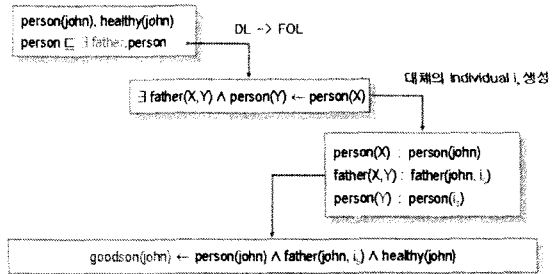
Class Constructors	Horn	problem
Conjunction (DL $\sqcap$ )	$C_1 \sqcap C_2 \sqsubseteq D \equiv D(x) \leftrightarrow C_1(x) \wedge C_2(x)$	
Disjunction (DL $\sqcup$ )	$C_1 \sqcup C_2 \sqsubseteq D \equiv D(x) \leftrightarrow C_1(x) \vee C_2(x)$	
Universal Restriction (DL $\forall$ )	$C \sqsubseteq \forall P.D \equiv D(y) \leftarrow C(x) \wedge P(x,y)$	l. h. s
Negation (DL $\neg$ )	$\neg \exists P.C \sqsubseteq \forall P. \neg C$	r. h. s
Cardinality (DL $\leq, \geq$ )	$\neg \geq nP \sqsubseteq \leq (n-1)P$	

그림 7 DL에서 일차 술어논리로 변환

온톨로지 Classification 과정을 위해 [그림 7]과 같은 변환 규칙이 필요하다. 존재 정량자 핸들러에서 비결정가능성의 원인이 되는 존재 정량자와 관련된 논리 구문을 탐색하기 이전에, [그림 3-7]과 같은 논리의 변환 규칙에 관한 연구가 선행된다.

3.4.2 임의의 대안개체 생성 기법

Unsafe 규칙을 지원하고, 결정 가능한 알고리즘을 적용할 수 있도록 돕는 블록 방법을 설명한다. [그림 8]과 같이 단계별로 블록 방법이 진행되어진다. DL의 TBox에 존재하는 존재 정량자는 임의의 대안 개체 생성 기법에 의해 ABox형태로 변환되는 기본 구조를 가진다.



[그림 8] 임의의 대안 개체 생성 기법

우선적으로 논리의 형을 통일화 시켜주어야 하므로, DL 형식의 OWL-DL을 일차술어논리 형식의 논리로 변환해준다. 변환과정 중에 비결정가능성의 원인이 되는 구문이 탐지되고, 관련된 구문은 ABox의 참조를 통해 새로운 ABox를 생성하게 된다. 즉, 비결정가능성이 존재하는 TBox의 논리적 구문은 추론에 활용될 수 있도록 특성을 유지하여 ABox로 변환되는 것이다. 이로써, OWL-DL과 SWRL의 결합과정에서 발생하는 비결정가능성을 제거할 수 있다.

[그림 9]는 종료가 결정되지 않아 무한히 반복할 수 있는 가능성을 보이고 있다. 이 문제는 임의의 임계값(Threshold)값을 설정하여, 표현력의 한계가 없는 범위에서 블록을 적용한다. 무한반복이 가능한 추이적인 속성

을 가진 프로퍼티에 적용된다.[13]

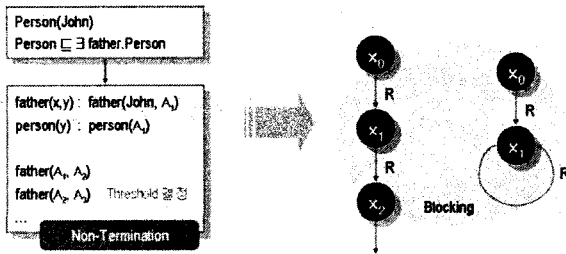


그림 9 Termination 문제 연구

4. 구현 및 실험

현재까지 SWRL을 지원하고, 결정 가능성을 고려한 OWL-DL 추론엔진은 그 수가 많지가 않다. KAON2는 이러한 연구를 수행한 대표적인 추론엔진이다. 하지만 결정 가능성을 유지하기 위해 표현력을 다소 잃는 해결 기법을 사용하여 문제점이 완전히 해결되지 않았다. 본 논문에서는 대안개체 생성 기법을 사용하여 표현력을 그대로 유지하면서 결정 가능성도 유지할 수 있는 방법을 제안하고 프로그램을 구현하였다.

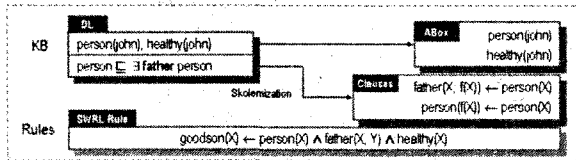


그림 10 지식베이스와 규칙의 형 일치

[그림 10]은 OWL-DL과 SWRL의 결합을 위해 형을 일치시키는 과정을 보이고 있다. OWL-DL은 DL을 근간으로 하므로, 일차술어논리 형식으로 변환이 가능하다. SWRL은 혼 논리를 근간으로 표현되기 때문에, 일차술어논리 형식이라 할 수 있다. 두 형식의 논리는 일차술어논리 형식으로 형이 일치됨을 알 수 있다.

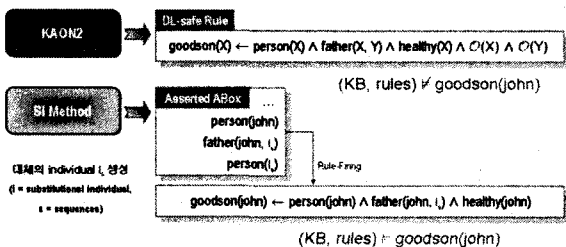


그림 11 KAON2와 SI 기법의 차이점

DL-Safe 규칙을 사용하고 있는 KAON2는 [그림 11]과 같이 조건절에 존재하는 모든 변수가 ABox에 명백히 명시되어 있는 경우만 그 규칙의 수행이 성공하게 된다. 즉, DL의 존재 정량자에 관한 표현을 잃을 수 있다는 단

점이 있다. 이를 보완하여 대안개체 생성 기법에서는 존재 정량자를 미리 일련의 변환 과정을 거쳐 ABox 형식으로 변환을 시켜놓는다. 결국 [그림 11]의 예제와 같이 같은 지식베이스와 규칙을 이용하여, KAON2는 goodson(john)이라는 사실을 도출하지 못하는 반면, 본 논문에서 제안한 대안개체 생성 기법을 사용하면 goodson(john)을 도출해 낼 수 있음을 알 수 있다. 물론 두 가지 기법 모두 결정가능성을 유지가 가능하다.

4.1 표현력 비교 실험

실험은 KAON2와의 표현력 비교실험을 수행하였다. 구체적인 온톨로지와 사용자 규칙을 제시하고 각각 KAON2와 본 논문에서 제안한 방식으로 실험을 실시하였다.

[그림 12]는 OWL-DL과 SWRL로 표현된 지식베이스를 나타내고 있다. 두 형식의 논리가 일차술어논리 형식으로 변환되어 형이 통일화된 논리적 구문도 볼 수 있다.

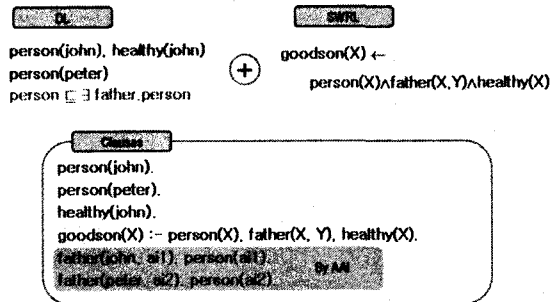


그림 12 실험 예제

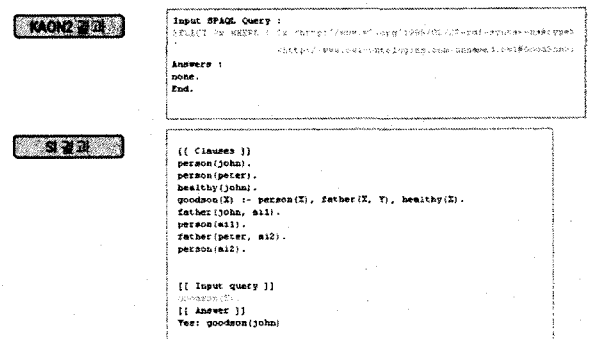


그림 13 실험예제의 결과

[그림 12]는 person(X), healthy(X), father(X,Y)이면 goodson(X)임을 SWRL 규칙을 이용하여 나타내고 있다. 즉, '사람이 건강하고 아버지를 가지고 있으면 좋은 아들이다.'라는 의미를 규칙으로 표현한 것이다. DL의 ABox와 TBox에는 다음과 같은 사실이 명시되어 있다. 'john이라는 사람이 있다.', 'john은 건강하다.', 'peter라는 사람이 있다.'를 정의한 ABox가 있고, '사람은 누구나 한명

이상의 사람인 아버지를 가진다.'라는 TBox가 명시되어 있다. KAON2에서는 무한반복을 피하기 위해서 TBox에 있는 사실을 제거한다. 그래서 [그림 13]과 같이 'john이 goodson이다.'라는 결론을 도출하지 못한다. 본 논문의 대안개체 방식을 사용하면 TBox의 무한반복 가능성을 ABox로 대체함으로써 무한반복도 제거하고 표현력도 유지할 수 있게 된다. 'john이 goodson이다.'라는 결론을 도출함을 알 수 있다.

### 5. 결론

OWL-DL과 SWRL의 결합 시 발생하는 비결정가능성 문제를 임의의 대안 개체를 생성하는 블록방법을 제안하여 해결하고자 하였다. TBox에서 발생할 수 있는 무한반복의 가능성을 제거하기 위하여, TBox의 존재 정량자를 ABox에 적합한 개체들로 대체 하였다. 논리적인 가독성이 좋은 TRIPLE 구조를 이용하여 OWL-DL과 SWRL로부터 형 일치된 일차적 술어논리를 탐색한다.

공극적으로 높은 성능의 추론이 가능한 풍부한 표현력을 유지하면서, 결정 가능한 효율적인 질의응답이 가능한 온톨로지를 구성할 수 있게 되었다. 시맨틱 웹을 활용하여 서비스 어플리케이션을 구현하는데 있어서, 효율적인 시간 내에 서비스의 제공이 이루어 질 수 있다고 확신한다.

하지만 논리의 특성을 고려할 때, 대용량의 ABox에서는 대안개체 생성 기법이 효율적으로 낮은 성능을 보일 가능성도 존재한다. 유비쿼터스 컴퓨팅 환경의 수많은 개체들과 그들 간의 관계를 ABox 형태의 객체로 표현하고 관리하게 되면, 논리적 구문도 복잡해지고 고려하지 못하는 논리적 오류의 가능성 또한 존재할 수 있다. 온톨로지 모델링의 경험을 통해 이러한 문제점들을 발견하고 논리적 오류의 특성을 파악하여 보다 완벽한 결정 가능성이 유지된 추론이 가능할 것으로 예상된다.

### 참고문헌

- [1] Harry Chen, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems", University of Maryland, 2004.
- [2] L. Bachmair and H. Ganzinger. Resolution Theorem Proving. In A. Robinson and A. Voronkov, editors, Handbook of Automated Reasoning, volume 1, chapter 2, pages 19-99. Elsevier Science, 2001.
- [3] Ian Horrocks, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", 2004.
- [4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. "The Description Logic Handbook" Cambridge University Press, 2002.
- [5] U. Hustadt, B. Motik, U. Sattler. "Reducing SHIQ Description Logic to Disjunctive Datalog Programs." Proc. of the 9th International Conference on Knowledge Representation and Reasoning (KR2004), June 2004, Whistler, Canada, pp. 152-162
- [6] M. Vardi. "Why is modal logic so robustly decidable?" In N. Immerman and P. Kolaitis, editors, Descriptive Complexity and Finite Models, volume 31 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 149-184. AMS, 1997.
- [7] V. Haarslev and R. Moller. "Description of the RACER system and its applications." In DL2001 Workshop on Description Logics, Stanford, CA, 2001.
- [8] 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.
- [9] U. Hustadt, B. Motik, U. Sattler. "Reasoning in Description Logics with a Concrete Domain in the Framework of Resolution." Proc. of the 16th European Conference on Artificial Intelligence (ECAI 2004), August, 2004, Valencia, Spain, pp. 353-357
- [10] B. Motik, U. Sattler, R. Studer. "Query Answering for OWL-DL with Rules." Proc. of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, November, 2004, pp. 549-563
- [11] U. Hustadt, B. Motik, U. Sattler. "A Decomposition Rule for Decision Procedures by Resolution-Based Calculi." Proc. of the 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004), Montevideo, Uruguay, March, 2005, pp. 21-35
- [12] A. Y. Levy and M.-C. Rousset. "Combining Horn rules and description logics in CARIN." Artificial Intelligence, 104(1-2):165-209, 1998.
- [13] S. Tobies. "Complexity Results and Practical Algorithms for Logics in Knowledge Representation." PhD thesis, RWTH Aachen, Germany, 2001.

### Acknowledgement

본 논문은 정보통신부 및 정보통신연구진흥원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.