온톨로지 Open World 추론과 규칙 Closed World 추론의 통합

(Integration of Ontology Open-World and Rule Closed-World Reasoning)

최 정 화 † 박 영 택 ††
(Jung-Hwa Choi) (Young-Tack Park)

요 약 OWL 온톨로지는 실세계의 도메인 지식을 모델링 하는데 적합하다. 또한 명백하게 정의된 지식으로부터 암시적인 새로운 지식을 추론할 수 있다. 하지만 이 모델링된 지식은 완전할 수 없다. 사람이 가지고 있는 모든 상식을 모델링 할 수 없기 때문이다. 온톨로지는 완전한 지식표현을 위한 무결성 제약조건과 예외 처리와 같은 비단조 추론을 지원할 방법이 없다. 디폴트 규칙은 온톨로지 안의 특정 클래스에 대한 예외를 처리할 수 있다. 또한 무결성 제약은 온톨로지에 정의된 클래스의 제한조건(restriction)에 인스턴스가 일관되게 할 수 있다. 본 논문에서는 Open World Assumption(OWA) 기반의 온톨로지와 Closed World Assumption(CWA) 기반의 비단조 추론을 지원하는 규칙의 지식베이스를 통합하여 Open World와 Closed World 추론을 모두 지원하는 실질적인 추론 시스템을 제안한다. 이 시스템은 온톨로지에 정의된 불완전한 개념을 다룰 때 OWA기반이라서 발생하는 문제점을 ASP(Answer Set Programming)를 사용하여 해결방안을 제안한다. ASP는 논리 프로그래밍 언어로써 비단조 추론을 허용하며, 서술 논리 지식베이스에 CWA 기반의 질의를 가능하게 한다. 제안하는 시스템은 Protege에서 제공하는 Pizza 온톨로지를 예로써 비단조 추론이 필요한 경우를 보이고, 잘 알려진 온톨로지들로 성능 평가하여 본 시스템의 정당(sound)하고 완전(complete)함을 증명한다.

키워드 : CWA, OWA, 시맨틱 웹, OWL 온톨로지, ASP, 비단조 추론, 태블로 알고리즘

Abstract OWL is an ontology language for the Semantic Web, and suited to modelling the knowledge of a specific domain in the real-world. Ontology also can infer new implicit knowledge from the explicit knowledge. However, the modeled knowledge cannot be complete as the whole of the common-sense of the human cannot be represented totally. Ontology do not concern handling non-monotonic reasoning to detect incomplete modeling such as the integrity constraints and exceptions. A default rule can handle the exception about a specific class in ontology. Integrity constraint can be clear that restrictions on class define which and how many relationships the instances of that class must hold. In this paper, we propose a practical reasoning system for open and closed-world reasoning that supports a novel hybrid integration of ontology based on open world assumption (OWA) and non-monotonic rule based on closed-world assumption (CWA). The system utilizes a method to solve the problem which occurs when dealing with the incomplete knowledge under the OWA. The method uses the answer set programming (ASP) to find a solution. ASP is a logic-program, which can be seen as the computational embodiment of non-monotonic reasoning, and enables a query based on CWA to knowledge base (KB) of description logic. Our system not only finds practical cases from examples by the Protege, which require non-monotonic reasoning, but also estimates novel reasoning

· 이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-511-D00338)

† 학생회원 : 숭실대학교 컴퓨터학과
cjh7963@hotmail.com
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
park@ssu.ac.kr

논문접수 : 2009년 9월 23일
심사완료 : 2010년 1월 28일

정보과학회논문지: 소프트웨어 및 응용 제37권 제4호(2010.4)

results for the cases based on KB which realizes a transparent integration of rules and ontologies supported by some well-known projects.

Key words : Closed World Assumption, Open World Assumption, Semantic Web, OWL Ontology, Answer Set Programming, Non-monotonic Reasoning, Tableaux Algorithm

1. 서 론

시맨틱 웹은 웹의 정보를 컴퓨터 또는 에이전트가 이해할 수 있는 표준화된 기술로 현재의 웹을 확장하는 데 목적을 둔다. 현재의 웹 언어 표준 온톨로지 언어는 OWL(Web Ontology Language)이며, 이를 통해 에이전트가 읽을 수 있는 형식(format)으로 지식을 표현하고 이 지식으로부터 새로운 지식을 추론할 수 있다. 이러한 시맨틱 웹 기술은 W3C의 시맨틱 웹 계층도를 기반으로 한다[1]. 이 계층도는 2005년을 기준으로 온톨로지 위에 규칙이 존재하던 계층도에서 온톨로지와 규칙이 같은 레벨에 존재하게끔 변경된다. 현재, 변경된 계층도에 준하여 온톨로지와 규칙의 지식베이스(knowledge base: KB)를 결합하는 연구가 활성화되고 있다[1-3]. 하지만 이 두 KB를 결합하기에는 논리적으로 불일치하다. 온톨로지는 서술논리(Description Logic: DL)를 기반으로 하며, open world assumption(OWA)기반의 단조(monotonic) 추론을 제공한다. 반면에 규칙은 서술 논리 보다 개념이 큰 일차논리(First-Order Logic: FOL)를 기반으로 하며, closed-world assumption(CWA) 기반의 비단조(non-monotonic) 추론을 제공하기 때문이다. 사람들의 대화는 전형적으로 CWA를 기반으로 하며, 시맨틱 웹 기반 어플리케이션 서비스의 실현을 위해서는 이 두 개의 특징을 결합한 추론 시스템이 반드시 필요하다. 현재 많은 관련 연구들은 지식 모델링의 실제적인 시스템을 주장하면서 OWA와 CWA의 통합을 제안하지만 실제적으로 테스트 가능한 시스템은 없다.

본 논문에서는 온톨로지 구축 시 발생할 수 있는 불명확한 지식 표현을 개선하고자, 온톨로지와 규칙의 지식베이스를 결합하여 추론하는 방법을 제안한다. 제안하는 방법은 현재의 시맨틱 웹 계층도를 지향하며, 시맨틱 웹 기반의 비단조 추론을 지원하는 방법론의 실용성을 증명하기 위해 간단한 OWA+CWA 추론 엔진을 구현하였다. 이 엔진은 태블로 알고리즘(tableaux algorithm) 기반의 OWA 추론 엔진과 ASP(Answer Set Programming)[4]를 이용한 CWA 추론 엔진을 결합하여 OWL의 표현력(expressivity)을 확장한다.

통상적인 논리에서의 추론은 논리 시스템에 새로운 공리가 추가되더라도 이 시스템으로부터 생성될 수 있는 정리의 집합 크기가 줄어들지 않으므로 이런 의미에서 "단조롭다"고 한다. 즉, Δ'가 Δ를 포함한다면($\Delta \subseteq \Delta'$), $\Delta \vdash \omega$인 ω에 대하여 $\Delta' \vdash \omega$도 성립한다[5]. 그러나 사람의 상식적인 추론 중에는 단조롭지 않은 것들도 많으며, 단조 추론과는 달리 결과 공리에 자격요건(qualification)이 추가 되었을 때 제거 될 수 있는 디폴트 결론을 허용하는 비단조(nonmonotonic) 추론도 필요하다. 비단조 추론의 하나인 디폴트 추론(default reasoning)[5,6]은 어떤 지식을 추론하기 위한 관련 지식이 없는 경우 일단 그것이 옳다고 생각되는 것으로 추론을 하는데, 나중에 모순되는 새로운 지식을 얻게 되면 디폴트 추론 결과를 철회(retraction)한다. 이런 추론은 종종 어떤 상황에 대해 모델링할 지식이 많을 때 온톨로지 설계자에게 필요한 직관적인 지식이다.

많은 지식 모델링은 CWA와 관련이 있고, 일차논리로는 표현 할 수 없다. 디폴트 규칙은 예외(exception)를 모델링하는 것을 가능하게 한다. "채식주의자가 아니면 고기를 먹는다."를 예로 들 수 있다. 이 규칙은 채식주의자 인지를 각 사람에게 묻는 수고를 덜어준다. 디폴트 규칙은 도메인 지식으로 가정(conjecture)을 통합하는 의미를 가진다. 온톨로지 설계자가 만든 도메인 온톨로지에 특정 클래스 속성의 가정을 추가한다. 제안하는 OWA+CWA 추론 엔진은 OWA 기반으로 추론된 지식 중 비단조 추론이 요구되는 개념에 ASP를 이용하여 CWA 추론을 지원한다. ASP는 특정 개념의 제약(constraint) 사항을 서술적(declarative)으로 명시하고, 제약을 만족하는 인스턴스들을 해결 방법으로 제시하여 현실적으로 만족할 만한 수준의 추론 결과를 제공한다. 제안하는 시스템은 비단조 추론 중 디폴트 지식을 표현할 수 있는 시맨틱 웹을 위한 시스템을 제안하여, 불완전하고 불확실한 지식 베이스에서 결정 가능한 추론 결과를 유도한다. 실험으로는 Protege[2] 온톨로지로 SHOIN(D)의 표현력을 지원하는 Pizza 온톨로지에서 비단조 추론이 필요한 경우를 보이고, 잘 알려진 온톨로지들(OWL 가이드[7], AKT의 portal, NASA의 SWEET, Galen 등)로 실험하여 제안하는 시스템의 정당(sound)하고 완전(complete)함을 증명한다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 배경 지식과 기존의 OWA와 CWA의 결합을 논의한 연구들에 대해 설명하고 3장에서는 OWA와 CWA의 특징과 차이점 및 통합해야하는 이유에 대해서 논의한다. 4장

1) 최근의 시맨틱 웹 계층도 다이어그램(http://www.w3.org/2001/sw/)

2) http://protege.stanford.edu/

에서는 제안하는 OWA와 CWA 지식베이스의 통합 및 추론 방법론을 설명하고, 5장에서 제안한 방법의 타당성과 정확성을 검증하기 위해 실험한 결과를 기술한다. 마지막 6장에서는 결론을 맺고 향후 연구를 제시한다.

2. 관련 연구

2.1 Autoepistemic 논리

Autoepistemic 논리[8]는 가정(conjecture, K 연산자)을 기반으로 추가적인 결론을 유도한다. 이 결론에 의해 도메인 온톨로지의 특정 클래스에 대한 불완전(incomplete)한 지식을 다룰 수 있다. Autoepistemic 추론은 비단조 추론의 한 형태이다. 하지만 autoepistemic 질의 언어는 비단조 추론의 특성인 디폴트 부정(negation) 또는 예외(exception) 모델링을 제공하지 않는다[9]. 또한 non-disjunctive와 disjunctive ASP 보다 상당한 계산 복잡도(computational complexity)를 가진다[10]. epistemic 연산자로 구성되는 일차 논리(FOL: First-Order Logic)에서의 정리 증명(theorem proving)은 semi-decidable도 아니다. 관련 연구들은 질의에 대해 옳고(correct) 완전한(complete) 추론 기법을 찾으려 하지 않았다. 다만, 한 예제의 지식표현을 결정가능(decidable)하게 표현하였다[11]. 이에 반해 ASP는 결정 가능함이 증명되었다[3]. 본 연구에서는 ASP를 이용하여 비단조 추론을 지원한다.

2.2 관련연구들의 CWA와 OWA 결합

관련연구들[9, 12-14]은 온톨로지 지식 베이스 기반의 질의에 대해 '알 수 없음(unknown)'의 응답을 얻는 경우에 대해 yes 또는 no의 결정 가능한 응답을 얻도록 하였다. 기초되는 이론은 온톨로지에 CWA 기반의 디폴트 규칙과 무결성 조건을 추가하여 비단조 추론을 지원하는 방법이다.

Pellet[12]은 ALCK(ALC + K 연산자(autoepistemic 논리))를 사용하여 OWL기반의 비단조 추론 데모를 제공한다.[3] 이 연구는 연산자 K를 클래스나 속성에 적용한다. 하지만 존재 양화사(existential quantifier, '∃')와 전칭 양화사(universal quantifier, '∀') 제약에 K 연산자를 적용하는 것은 고려하지 않았다. 일차논리에서는 제약(constraint)이나 디폴트(default)의 형식화를 제공하지 않는다[12]. 하지만 이러한 비단조 추론을 위한 요소는 지식 베이스의 상태에 대한 추론을 위해서 필요하며, 본 논문에서는 필요(충분)조건(또는 description)에 존재 양화사를 포함하는 클래스의 일관성도 고려한다.

Boris Motik[9]은 MKNF(Minimal Knowledge & Negation-as-Failure) 규칙[15]을 사용하여 CWA와 OWA 지식의 통합을 제안한다. OWL의 단점을 논리 프로그래밍(LP: Logic Programing)으로 해결하려는 아이디어를 기반으로 SWRL의 비결정 문제를 DL-safe 규칙과 autoepistemic 논리로 확장시키는 접근법을 제안한다. MKNF는 ALC 표현력만을 지원하며 알고리즘만 공개되어 있다. 이 연구는 하나의 시나리오를 통해 제안한 방법론을 적용하는 예만 보여 주고 있다.

Stephan Grimm[13]은 Boris Motik[9]의 아이디어를 기반으로 온톨로지 에디터인 Protege의 예제 중, 피자 온톨로지(pizza.owl)에서 CWA 추론이 필요한 예를 제시하고 일관성을 유지할 수 있는 방법을 제안한다. 이 방법 역시 autoepistemic 논리의 K 연산자를 사용하지만 보여주는 예는 OWL의 공리(axioms)를 이용해서도 해결 할 수 있다.

C.V.Damasio[14]는 본 논문에서 사용한 ASP를 사용하여 질의의 답변을 '알 수 없음'으로 반환할 수 있는 술어(predicate), p(a)에 대해 Negation-as-Failure(NaF, ~)를 적용한 두 개의 규칙, 즉 "~¬p(a)이면 p(a)이다", "~p(a) 이면 ¬p(a)이다"를 추가하여 p(a)를 닫힌(closed) 술어로 만들고 yes 또는 no를 반환하게 한다. 이 방법은 가능성이 있는 모든 술어에 대해 닫힌 술어로 정의해야 하므로 저장소의 공간을 많이 차지하고, 온톨로지의 OWA 특징을 배제한 CWA기반 지식 베이스를 초래하는 문제점이 있다. 이 연구 또한 예제를 통해서만 OWA와 CWA 지식베이스의 통합을 보이고 있다. 본 논문에서는 OWA 추론 시 불일치하는 지식에 대해 CWA 지식으로 변환하는 인터페이스를 제공하고, ASP를 사용한 CWA와 DL 기반의 OWA 지식의 통합과 추론을 위한 일반적인 시스템을 제안한다.

2.3 서술논리 표현의 한계

OWL DL은 SHOIQ 수준의 풍부한 표현력을 지원하지만 사람이 생각하는 지식을 완전하게 표현하지 못한다. 예를 들어, 요즘 웰빙(well-being) 열풍이 불면서 사람들이 건강을 중요시하자, 피자 가게 주인이 100% 콩으로 만든 소시지를 피자 토핑으로 추가하고, 기존의 피자 온톨로지의 소시지 토핑의 하위분류로 콩 소시지 토핑을 추가한다. 하지만 추론엔진은 이 지식 표현이 불일치(inconsistency)하다는 결과를 반환한다. 그림 1은 Protege에서 예제로 제공하는 피자 온톨로지에 위의 시나리오를 추가하고 Pellet 추론엔진을 수행하였을 때, 에러가 발생한 화면이다. 사람의 사고로는 콩 소시지는 소시지 분류에 속하는 게 당연한데 왜 서술논리 기반 추론엔진은 이 지식을 일관성이 깨진다고 판단한 것일까? 온톨로지의 논리로는 불일치하는 지식을 어떻게 표현할 수 있을까? 본 연구에서는 이 문제의 해답을 찾기 위해 CWA기반의 휴리스틱 탐색 방법을 제안한다.

3) http://www.mindswap.org/2005/alck/alckdemo.shtml

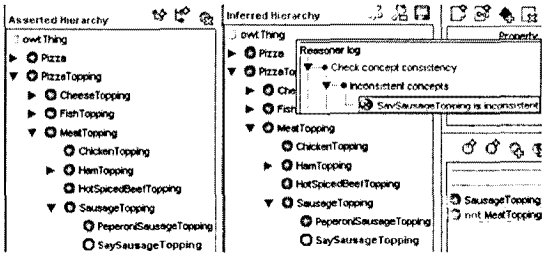

그림 1 비단조 추론이 필요한 예

2.4 서술논리 기반 규칙 언어, SWRL(Semantic Web Rule Language)

SWRL(Semantic Web Rule Language)[16]은 온톨로지에 모델링된 지식들의 규칙을 정의 할 수 있다. FOL 보다는 표현력이 다소 떨어지고, 비결정 가능한 알고리즘을 구성할 수 있다는 단점을 지닌다. 아래의 규칙을 예로 들 수 있다.

$$Person(?x) \wedge hasAge(?x, ?age) \wedge swrlb:add$$
$$(?newage, ?age, 1) \rightarrow hasAge(?x, ?newage)$$

SWRL 추론은(만약에 built-in 술어를 추가하였다면, 예: $swrlb:add$) 단조롭다. 그래서 위의 예제에서는 사람의 나이를 수작업으로 수정하는 대신에 규칙을 통해 자동으로 증가한다. 하지만 이 규칙은 무한으로 실행되며 추론의 종료를 보장하지 못한다. 즉, 비결정 가능한(non-decidable) 결과를 초래한다. 본 연구에서는 SWRL과 달리 결정가능성을 보장하는 ASP를 이용하여 규칙을 정의한다.

3. 지식베이스의 일관성(consistency)과 완전성(completeness)

지식베이스 KB에 정의된 명제 ω에 대해 ω, $\neg\omega$가 존재하지 않으면 'KB가 일관성 있다'고 한다. 이것은 또한 KB에 정의된 개념들의 계층 구조가 만족(satisfiable)됨을 의미한다. 또한 KB의 모든 명제 ω가 ω 또는 $\neg\omega$로 알려져(known)있으면, 이 KB는 완전(complete)하다고 말한다. 일반적으로 지식베이스는 불완전(incomplete)할 수 있다. 예를 들어, $KB = \{PizzaTopping \subseteq SpicyPizza \vee \neg SpicyPizza, PizzaTopping(Margarita)\}$와 같은 지식베이스에서 $SpicyPizza(Margarita)$라는 질의를 수행했을 때, 피자 인스턴스 $Margarita$는 $SpicyPizza(Margarita)$ 또는 $\neg SpicyPizza(Margarita)$의 명제를 수반(imply)하지 못한다. 따라서 '알 수 없다'는 대답을 반환한다. 온톨로지에서는 이와 같은 불완전한 지식베이스의 모델링을 생성 할 수 있다. 본 연구에서는 이러한 불완전한 지식을 완전하게 할 수 있는 closed-world 추론을 지원하여 완전한 지식 베이스 기반의 추론을 가능하게 한다.

3.1 서술논리 기반 온톨로지 표현 언어의 제약

제안하는 CWA 추론을 위한 휴리스틱 탐색 방법을 살펴보기 전에 2.3절에서 언급한 문제가 발생 할 수밖에 없는 서술논리의 특징을 살펴보자. 사람들의 사고와 대화는 전형적으로 CWA를 기반으로 한다[9,13]. 하지만 서술논리는 다음과 같은 특징을 가진다.

3.1.1 Open World Assumption(OWA)

서술논리 기반의 OWL 온톨로지는 OWA를 기반으로 한다. 지식베이스에 진실(ω) 혹은 거짓($\neg\omega$)으로 정의되지 않은 명제 ω에 대해 '알 수 없다(unknown)'고 처리한다. 즉, 찾을 수는 없지만 어딘가에 정의되어 있다고 본다. 이러한 논리는 시맨틱 웹 기반의 웹 서비스 프로그램에서는 비결정 가능한 결과를 초래한다. 이 문제는 완전한 지식베이스 기반의 결정 가능성이 보장되어야 하는 웹 서비스 어플리케이션에서는 신뢰성을 저하시킨다.

3.1.2 단조(monotonic) 추론

OWA는 단조 추론을 기반으로 한다. 통상적인 논리에서의 추론은 논리 시스템에 새로운 공리가 추가되더라도 이 시스템으로부터 생성될 수 있는 정리의 집합 크기가 줄어들지 않으므로 이런 의미에서 '단조롭다'고 한다. 즉, KB_2가 KB_1을 포함한다면($KB_1 \subseteq KB_2$), $KB_1 \models \omega$인 ω(명제, statement)에 대하여 $KB_2 \models \omega$도 성립한다. 그림 1의 예에서는 $SoySausage\,Topping$이 $Sausage\,Topping$에 속하고, $Meat\,Topping$이 아님이 사람의 상식으로는 진실이지만, 단조추론을 기반으로 하는 온톨로지 추론에서는 모순이 된다.

3.2 논리 프로그래밍 기반 규칙언어의 특징

3.2.1 Closed World Assumption(CWA)

규칙은 CWA를 기반으로 한다. CWA는 KB의 어떤 명제 ω에 대해 ω 또는 $\neg\omega$를 수반한다. 즉 완전한 지식베이스를 구축할 수 있다. KB의 모든 명제는 KB_1에서 확장된 KB_2에 의해 결정가능하다. 즉 KB_2의 모든 명제에 대해 진실(true) 혹은 부정(negative)을 반환한다. 시맨틱 웹 기반의 실질적인 서비스를 위해서는 도메인 지식에 따라서 CWA를 지원해야 하는 예외가 발생한다. 그림 1의 예제는 이를 보여주며, 본 연구에서는 사람의 상식 수준과 일치하는 완전한 추론을 제공하기 위하여 OWA기반 온톨로지에 CWA기반 규칙 지식베이스를 결합하는 추론을 제공한다.

3.2.2 비단조(non-monotonic) 추론

CWA는 비단조 추론을 기반으로 한다. 비단조 추론은 단조 추론과는 달리 KB_1에서 수반된 명제가 확장된 지식베이스 KB_2에서는 수반되지 않는 것을 의미한다. 즉, KB_2가 KB_1을 포함한다면($KB_1 \subseteq KB_2$), $KB_1 \models \omega$인 명제 ω에 대하여 $KB_2 \models \omega$는 성립하지 않는다. 예를 들

어, 디폴트 추론은 비단조 추론의 한 종류이며, 어떤 지식을 추론하기 위한 관련 지식이 없는 경우 일단 그것이 옳다고 생각되는 것으로 (KB_1에서) 추론을 한다. 그리고 나중에 모순되는 새로운 지식을 (KB_2에서) 얻게 되면, 디폴트 추론 결과를 철회(retraction) 한다. 이러한 추론이 필요한 예는 사람이 사는 세상에서 너무도 많이 발생하며, 시맨틱 웹 기반의 지식베이스 모델링 시에 반드시 다루어져야할 문제이다. 그림 1은 비단조 추론이 필요한 하나의 예이며, 비단조 추론의 방법인 디폴트 추론을 이용하면 이 문제를 해결할 수 있다. 이 문제의 해결방법은 4.1절에서 자세히 다룬다.

4. Open World 추론과 Closed World 추론의 통합

본 논문에서는 시나리오에 종속되지 않은 OWA+CWA 추론을 지원하는 실질적인 시스템을 제안한다. 제안하는 OWA와 CWA의 결합은 온톨로지 모델링 시에 사람의 상식으로는 정당한 지식이 불일치로 분류되는 지식에 CWA 기반의 추론을 지원한다. 이로 인해 서술논리 기반의 온톨로지와 논리 프로그램인 규칙 지식베이스의 통합이 가능하다.

그림 2는 OWA와 CWA 지식베이스의 통합 구조를 보여준다. 첫째, 온톨로지 설계자는 OWA 기반으로 추론된 지식베이스(O)를 검토하고, 상식적으로 합당한데 OWA의 특징 때문에 불일치로 나타나는 개념(C_n)을 디폴트 규칙 편집기를 통해 선택한다. 디폴트 규칙 생성 엔진은 선택된 개념(들)에 디폴트 규칙(D^+)을 생성한다. 둘째, OWA+CWA 추론 엔진은 사용자 질의가 들어왔을 때, OWA 지식과 CWA 지식을 기반으로 결정 가능한 답변을 반환한다. 무결성 제약 일관성 검사 엔진

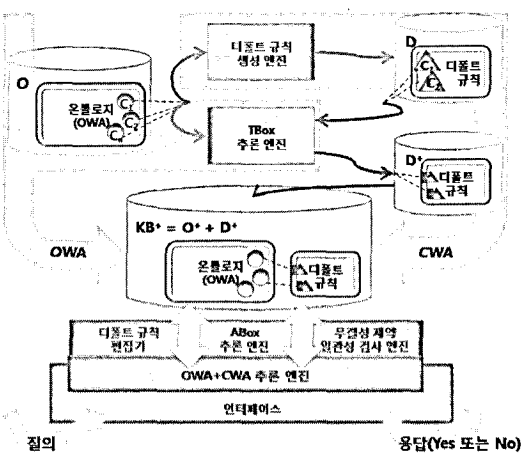

그림 2 OWA와 CWA 지식베이스의 통합 구조

은 사용자 질의 시에 OWA 지식의 불명확한 지식 표현을 개선(KB^+)한다. 본 장의 4.1절에서는 제안하는 시스템에서 적용한 CWA기반 추론에서 가능한 기술을 설명하고, 이 기능을 제공하기 위한 휴리스틱 추론 방법을 4.2절에서 설명한다.

4.1 Closed World 추론에서 가능한 기술

4.1.1 예외 모델링(Modeling Exceptions)

Closed-world 추론에서는 실세계 지식 중 일반화 할 수 없는 예외 지식의 모델링이 가능하다. 가장 유명한 예로써 "새는 난다", "펭귄은 새이다"라는 일반적인 지식을 서술논리 기반으로 모델링하면 예제 1의 KB_1과 같다. 하지만 상식적으로 펭귄은 날지 못하므로 KB_2의 새로운 지식을 추가해야 한다.

[예제 1]

$KB_1 = \{Fly \supseteq Bird, Bird \supseteq Penguin\}$

$KB_2 = KB_1 \cup \{\neg Fly \supseteq Penguin\}$

$KB_3 = \{Fly \supseteq Bird \cap \neg abnormal,\ Bird \supseteq Penguin,$
$abnormal \supseteq Penguin\}$

KB_2는 KB_1에 의해 추론되는 $Fly \supseteq Penguin$과 모순된다. $Penguin$은 $\neg Fly$와 Fly를 모두 수반하기 때문이다. 하지만 CWA에서는 날지 못하는 새, 즉 예외 모델링을 할 수가 있다. KB_3과 같이 펭귄을 날지 못하는 새로 분류하면 된다. 그림 1도 예외 모델링에 속하며 본 논문의 추론 엔진에서는 예외 처리 기술을 추가한 일관성 있는 모델링을 지원한다.

4.1.2 무결성 제약(Integrity Constraints)

무결성이란 데이터베이스 내의 데이터들이 결함이 없는 일관성을 유지하려는 성질을 말한다. RDF 스키마(schema), 즉 RDFS를 기반으로 하는 온톨로지 언어도 데이터를 저장하는 저장소로써 무결성(일관성)을 보장하여야 완전한 지식베이스를 구축할 수 있다. 무결성 제약은 새로운 사실(fact)을 유도하는 과정 없이 지식베이스의 상태를 검사하는 데 사용된다.

예를 들어, $KB = \{Pizza \subseteq \exists topping.T,\ Pizza(pizzabread)\}$, 이 지식베이스는 무결성이 깨진다. 이유는 "피자로 알려진 인스턴스는 하나의 토핑을 가져야 한다"는 피자의 존재 양화사 때문이다. 이 조건을 만족하지 않는 피자로 정의된 인스턴스는 일관성이 깨진다(inconsistency: 불일치). 하지만 OWA기반의 온톨로지 추론에서는 $pizzabread$ 인스턴스의 토핑이 어딘가에 정의되어 있다고 보고, $pizzabread$를 불일치하는 개념으로 보지 않는다. 하지만, 인터넷으로 피자 배달 서비스를 한다고 생각해 보자. 새로운 피자가 나와서 온톨로지에 등록하였는데, 토핑을 입력하지 않았다면 이 피자는 토핑을 검색하는 대상에서 제외 된다. 무결성은 데이터베이

스에서 뿐만 아니라 실세계 예제에서도 지식베이스의 완전성을 위해서 고려해야할 기능이다.

4.1.3 UNA(Unique Name Assumption)

OWA는 UNA를 지원하지 않는다. ABox 지식베이스에 *Pizza*(*Pizza*01), *SpicyTopping*(*Chilli*)와 *SpicyTopping*(*Pepper*)가 있을 때, 사람이 판독하기는 *Chilli*와 *Pepper*가 다르다고 인식하지만, OWA기반의 온톨로지 추론 시에는 두 개체를 다르다고 보지 못한다. 즉 *allDifferent*{*Chilli*, *Pepper*}라고 정의하지 않는 한 동일 개체로 인식한다. 예를 들어, SWRL로 $hasSpicyTopping(?p, ?t_1) \wedge hasSpicyTopping(?p, ?t_2) \wedge diffentFrom(?t_1, ?t_2) \rightarrow hasSimilarTasteTopping(?t_1, ?t_2)$라는 규칙을 정의하고 추론하면, 위의 지식베이스의 *SpicyTopping* 두 개를 같은 인스턴스로 취급하여 이 규칙의 결론을 도출하지 못한다. 반면에, Prolog와 같은 규칙 언어에서는 디폴트값으로 UNA를 지원하여 모든 인스턴스를 다르다고 간주하고 추론한다.

4.2 완전한 온톨로지를 위한 CWA기반 휴리스틱(Heuristic) 추론 방법

본 연구에서 제안하는 추론시스템은 서술논리 기반 온톨로지 추론에서 지원하지 못하는 불완전한 명제(클래스)의 처리를 위해 휴리스틱을 사용하여 OWL 온톨로지 모델링시에 지원할 수 있는 최대의 표현력을 지원한다. 제안하는 휴리스틱 방법은 OWL 온톨로지로 정의된 지식 중에 CWA 처리를 요구하는 지식들을 ASP를 이용하여 비단조 추론을 지원하는 방법이다. 본 휴리스틱 방법을 OWL-DL기반 추론과 결합하면 한정된 시간 내에 현실적으로 만족할 만한 수준의 추론 결과를 제공할 수 있다.

4.2.1 ASP(Answer Set Programming)

ASP는 Answer sets(또는 stable models)[4]의 논리(logic) 프로그램으로 지식을 표현함으로써 문제(problem)를 해결하기 위한 프로그래밍 언어이다. 특정 개념의 제약(constraint)사항을 서술적(declarative)으로 명시하고, 제약을 만족하는 인스턴스들을 해결책(solution)으로 제시한다. 이 방법은 디폴트 논리를 구현하는 데 적합하며, 사람이 생각하는 직관적인 지식의 가정을 정의할 수 있다. ASP는 Prolog 프로그램 언어 기반의 문법을 제공하며, Answer set이란 Prolog 형태의 규칙을 만족하는 리터럴(literal; p 또는 $\neg p$)의 최소 집합을 의미한다. ASP의 정규 표현식은 (1)과 같다.

$$A \leftarrow B_1, ..., B_m,\ not\ C_1, ..., not\ C_n \qquad (1)$$

본 논문에서 closed-world 추론을 위해 ASP를 선택한 이유는 다음과 같다.

- Answer set은 결정 가능(decidable)한 프로그램을 가능하게 한다. 즉, answer set이 없으면 그 논리 프로그램의 solution은 존재하지 않는 것이다.
- 비결정 가능한 지식베이스 추론 시에 추론의 복잡도(complexity)를 감소한다.
- ASP는 대표적인 규칙 언어인 Prolog와 다음과 같은 차이점을 가진다.
- 규칙의 head 부분에 이접적인(disjunctive) 논리 프로그램 형식을 제공한다.

$$A_0 \text{ or } ... \text{ or } A_1 \leftarrow B_1, ..., B_m, not\ C_1, ..., not\ C_n \qquad (2)$$

- 특정 명제의 제약사항을 정의한다. 즉 결론(conclusion)부를 비워두어 거짓이 되는 명제를 표현한다.

$$\leftarrow A, not\ B \qquad (3)$$

식 (3)은 A를 만족하는 상수 x가 B를 만족하지 않으면 A는 지식베이스에 불일치하는 명제이다. 다시 말해서 A는 모두 B를 만족해야하는 제약사항을 정의한다. 마지막으로 ASP와 서술논리 기반 온톨로지와의 부정(negation) 표현의 차이점은 다음과 같다.

- 온톨로지에서 부정은 classical negation을 의미한다. $\neg p$는 지식베이스에서 p가 거짓임을 나타내고, 질의 $\neg p$는 지식베이스에 $\neg p$가 존재하면 진실, 존재하지 않으면 거짓이다.
- ASP 또는 Prolog에서의 부정은 온톨로지에서의 부정인 classical negation과 NaF도 지원한다. NaF는 명제 앞에 '*not*'을 명시하고, *not* p는 지식베이스에 p가 없으면 진실, 있으면 거짓이 된다.

4.2.2 ASP의 Reduct 알고리즘을 이용한 Answer Set (Stable Model) 도출 방법

Reduct 알고리즘은 answer set X가 지식베이스에 만족(satisfies)하는지를 검사하는데 사용된다. 따라서 이 과정을 반복하여 answer set을 구한다.

① 지식베이스에 표현된 formula에서 answer set X에 만족하지 않는 maximal subformula를 구한다.
② ①에서 구한 formula를 $\perp$ (false)로 대체한다.
③ 남은 formula가 X에 만족하는지 검사한다. 만족하면 answer set에 포함된다.
④ 남은 formula가 X의 subset에 만족하는지 검사한다. 만족하면 answer set에서 제외된다.

[예제 2]

$F = p \wedge ((p \wedge q) \rightarrow r)$, $X = \{p\}$

① $p \wedge q, r$

② $p \wedge (\perp \rightarrow \perp)$, simplify하면 p

③ p는 X에 만족한다.

④ p는 p의 subset $\varnothing$에 만족하지 않는다.

그러므로, p는 위의 formula F의 answer set에 포함된다.

4.2.3 디폴트 규칙(Default Rules)을 이용한 예외 처리

디폴트 규칙은 도메인 온톨로지의 특정한 개념(또는

클래스)에 대한 불완전한 지식을 다루기 위하여 가정을 기반으로 추가적인 결론을 유도한다. 디폴트 규칙은 OWA 기반 추론 중에 알 수 없는(unknown), 즉 불완전한 지식을 완전하게 할 수 있다. [예제 3]은 그림 1에서 설명한 불완전한 지식 베이스이다. KB_{OWA}에서 '*SoySausage Topping*은 *Meat Topping*이다'는 명제에 대해 참(*Meat Topping*)과 거짓($\neg$*Meat Topping*)이 모두 존재하므로 *SoySausage Topping*은 KB_{OWA}에 일관성이 깨지는 개념이다.

[예제 3]

KB_{OWA} = { *Meat Topping* $\supseteq$ *Sausage Topping*,
Sausage Topping $\supseteq$ *SoySausage Topping*,
$\neg$*Meat Topping* $\supseteq$ *SoySausage Topping*}

본 연구에서는 사람의 상식으로는 정당하지만, 온톨로지 모델링 시에 일관성이 깨지는 지식에 디폴트 규칙을 추가한다. 이를 위해 특정 속성에 대한 불확실한 Answer Set 모델을 제거한 Answer Set을 추출하는 방법을 사용한다. 디폴트 규칙의 형태는 $<\alpha:\beta/\gamma>$이고, 이 의미는 "α가 참(true)이고, β가 참이라는 가정(assume)이 일관(consistent)되면, 결론적으로 γ가 참이다." 디폴트 규칙(DR)은 $\gamma \leftarrow \alpha$로 형식화한다. [예제 3]을 위한 디폴트 규칙은 5.1.2절에서 기술한다.

4.2.3.1 CWA 기반 디폴트 규칙 생성 알고리즘

그림 3은 사용자가 예외로 선택한 클래스에 대해 자동으로 디폴트 규칙을 생성하는 알고리즘이다. 입력은 예외로 선택한 클래스 C이고, 출력은 디폴트 규칙을 정의한 파일이다. 생성과정은 다음의 (1)~(8)과 같다.

(1) 클래스 C의 상위 클래스들의 집합 D를 구한다. 그림 4를 예로 들어 보자. 클래스 C는 *Penguin*, D는 {*Bird*, *Fly*, $\neg$*Fly*}이다.

(2) 변수 i를 0으로 초기화한다.

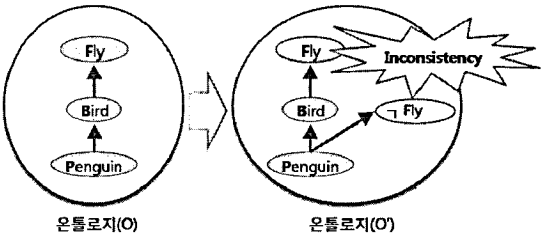

그림 4 Open World 추론에서 모순이 발생하는 예

(3) 집합 D의 원소 개수만큼 (3)에서 (9)를 수행한다.

(4) 원소 d_i의 부정(negation)이 집합 D에 속하지 않으면, (5)를 수행한다. 그림 3에서, d_i가 *Bird*라면 집합 D에 *Bird*의 부정인 $\neg$*Bird*가 존재하지 않으므로 (5)를 수행한다. $\neg$*Bird*의 부정은 $\neg\neg$*Bird*, 즉 *Bird* 이다.

(5) 디폴트 규칙 파일(outFile)에 (5)의 출력문에 대입되는 문자열을 출력한다. 그림 4에서는 *Bird*의 하위 클래스가 *Penguin*이므로 *Bird* :– *Penguin*을 출력한다.

(6) 원소 d_i의 부정이 집합 D에 속하면, 원소 d_i의 부정이 집합 D에 존재하여 충돌(즉, inconsistency)이 발생한 것이며, (7)과 (8)을 수행한다. (7)과 (8)은 디폴트 규칙을 생성한다. d_i가 *Fly*인 경우가 여기에 해당한다.

(7) C가 불일치하는 원인이 클래스 d_i 때문이므로, d_i가 유도되는 조건에 $\neg$abnormal을 추가한다. 즉, abnormal을 제외한 나머지 클래스들만 d_i로 수반된다. d_i가 *Fly*인 경우, *Fly* :– *Bird*, $\neg$*abnormal*1 문장이 생성된다.

(8) 클래스 C가 예외 클래스이므로 abnormal로 추가한다. C가 *Penguin*이므로 *abnormal*1 :– *Penguin* 문장이 생성된다.

입력 : 클래스 C
출력 : *outFile*

(1) *get D*, $C \sqsubseteq D$
$D = \{ d_1, d_2, d_3, ..., d_{k-1}, d_k \}$
(2) *int* $i = 0$
(3) *while*($i <$ *D.size*()) {
(4) if(*negationOf*(d_i) $\notin D$)
{
(5) *outFile.write*(d_i + " :– " + *subClassOf*(d_i)) }
(6) *else* {
(7) *outFile.write*(d_i + " :– " + *subClassOf*(d_i) + ", " + "$\neg$*abnormal*" + i)
(8) *outFile.write*("*abnormal*" + i + " :– " + C)
}
(9) $i = i+1$
}

그림 3 CWA기반 디폴트 규칙 생성 알고리즘

4.2.4 클래스의 무결성 제약(Integrity Constraints)

온톨로지에서의 무결성이 깨지는 클래스의 인스턴스는 검색 시에 고려대상에서 제외되고, 클래스는 해당 인스턴스를 답으로 반환하지 못한다. 이 역시 온톨로지 지식베이스가 불완전하기 때문에 발생하는 문제이다. [예제 4] 온톨로지를 예로 들어 보자.

[예제 4]

KB^1_{OWA} = {$Pizza \subseteq \exists topping.T$,
$Chili \subseteq \neg Cheese \cap \neg Tomato$,
$Margarita \equiv \exists topping.Tomato \cap \exists topping.Mozzarella \cap \forall topping.(Tomato \cup Mozzarella)$,
$MildChili \subseteq \neg SpicyDish$,
$NormalChili \subseteq \exists topping.Chili$,
$NonSpicyDish \equiv \neg SpicyDish$}
$ABox^1$ = {$Margarita(Margarita01)$,
$NormalChili(NormalChili03)$,
$MildChili(MildChili07)$}

수많은 인스턴스가 제약조건에 맞는지 사람이 눈으로 확인하기란 불가능하다. 온톨로지 편집기로 많이 사용되는 Protege에서도 마찬가지이다. 피자 배달 웹 서비스에서 고객이 *SpicyDish*가 아닌 피자를 주문하려한다. 온톨로지 추론에서는 $\neg SpicyDish$를 검색하였을 때, 명백히 $\neg SpicyDish$로 추론되는 *MildChili*07만을 결과로 도출한다. 하지만, 사람이라면, *Margarita*피자는 *SpicyTopping*을 포함하지 않으므로 *SpicyDish*가 아닌 피자로 추천할 것이다. 또한 *SpicyDish* 역시 명백하게 명시되어 있지는 않지만, *NormalChili*가 *Chili* 토핑을 포함하므로 *SpicyDish*로 분류할 것이다. 이와 같은 시나리오에서 질의는 *SpicyDish*와 *SpicyDish*가 아닌 인스턴스를 구분하려 한다. *Spicy*하면서 *Spicy*하지 않는 음식은 없기 때문이다. 하지만, 온톨로지 지식베이스는 사람이 완전하게 모델링 할 수 없으므로 모든 인스턴스에 대해 결정 가능한 결과를 반환하지 못한다. 따라서 무결성 제약, KB = {$Pizza \subseteq Spicy \cup \neg Spicy$}에서 *Margarita*와 *NormalChili* 피자는 무결성이 깨진다.

4.2.4.1 ASP를 이용한 무결성 제약 조건 생성 방법

본 연구에서 제안하는 방법은 온톨로지 설계자가 질의응답 시에 무결성 제약이 필요한 클래스에 그 클래스에 포함되는 인스턴스가 분류될 조건을 입력하게 하여 무결성이 깨지는 인스턴스들의 일관성을 유지하도록 한다. 즉, [예제 4]의 KB^1_{OWA}에 [예제 5]의 KB^1_{CWA}를 입력하여, 피자이면 *SpicyDish* 또는 *NonSpicyDish*로 분류되게 할 수 있다. 이 표현식은 4.2.1절에서 언급한대로 ASP의 장점이다. 또한 제안하는 시스템은 무결성 제약 조건의 head에 정의된 클래스들(*SpicyDish*, *NonSpicyDish*)의 예외를 사용자가 디폴트 규칙 편집기를 이용해서 쉽게 편집할 수 있게 한다(예: DR^1_{CWA}). 만약, 이 표현식을 온톨로지 모델링시에 추가하게 한다면 예외가 추가될 때마다 온톨로지 편집기를 통해 스키마를 변경할 것이다. 즉, 서비스가 제공된 후에도 설계부분에 해당하는 온톨로지 편집이 계속 발생하게 된다. 또한 OWL지식추론에 사용되는 트리 기반의 태블로 알고리즘은 각 클래스의 예외가 추가될 때마다 이접조건(트리의 가지(branch))이 늘어나서 계산 복잡도를 무한으로 증가하는 단점을 가진다.

[예제 5]

KB^1_{CWA} = {$spicyDish(P) \mid nonSpicyDish(P) :- pizza(P).$}
DR^1_{CWA} = {$nonSpicyDish(P) :- topping(P, X), not\ abnormal(X).$
$abnormal(X) :- chili(X).$
$spicyDish(P) :- topping(P, X), not\ abnormal2(X).$
$abnormal2(X) :- tomato(X).$
$abnormal2(X) :- cheese(X).$}

4.2.4.2 OWA기반 온톨로지 공리(axiom)를 이용한 디폴트 규칙의 확장

제안하는 CWA기반 디폴트 규칙은 디폴트 규칙에 포함된 예외에 대해 온톨로지에 정의된 description을 고려하여 규칙을 확장한다. 즉, [예제 5]에서 abnormal로 정의된 *cheese*는 온톨로지에 하위 개념으로 *mozzarella*를 포함한다(즉, $Mozzarella \subseteq Cheese$). 따라서 디폴트 규칙 처리 시 *mozzarella* 토핑을 가진 *Margherita*피자는 *SpicyDish*에서 제거된다. 이는 온톨로지의 표현력을 유지하기 위한 방법이다.

4.2.5 존재 양화사(existential quantifier)의 무결성 제약 조건

본 논문은 또한 온톨로지 지식베이스의 무결성을 보장하기 위해 OWL-DL의 존재 양화사의 완전성을 고려한다. 예를 들어, [예제 6]과 같이 모델링된 온톨로지가 있다고 보자.

[예제 6]

KB^2_{OWA} = {$CheesePizza \subseteq Pizza$, $CheesePizza \subseteq \exists topping.CheeseTopping$,
$CheesePizza(CheesePizza01)$,
$SetMenu(SetMenu1)$,
$hasPizza(SetMenu1, CheesePizza01)$}

피자 가게 주인은 금요일에 할인해 주는 피자를 홍보하기 위해 해당 조건을 피자 서비스에 추가하기를 원한다. 시스템 관리자는 SWRL 언어로 다음과 같이 쉽게

정의할 수 있다.

$FridayDiscount(?x) \leftarrow Pizza(?x) \wedge hasTopping(?x, ?y) \wedge CheeseTopping(?y) \wedge hasPizza(SetMenu1, ?x)$

하지만, 지식베이스 KB^2_{OWA}의 $CheesePizza01$은 이 규칙의 조건에 일치함에도 불구하고, 결과로 도출되지 않는다. 이유는 $CheesePizza01$은 치즈 토핑이 명시되어 있지 않기 때문이다. 하지만 치즈 피자라면 치즈 토핑을 가짐이 분명하고, 위의 지식베이스에도 적어도 하나의 치즈 토핑을 포함한다고 조건이 명시되어 있기 때문에 $CheesePizza01$을 금요일 할인 피자로 추천해야 한다. SWRL의 이와 같은 단점을 해결하기 위하여 본 연구에서는 온톨로지 클래스의 description으로 정의된 존재 양화사의 무결성을 유지하기 위한 방법을 제안한다.

4.2.5.1 존재 양화사의 무결성 유지 방법

본 연구에서는 존재 양화사의 조건을 보유한 클래스 중, 무결성이 깨져서 결과로 도출되지 못하는 개념에 대해 다음과 같은 과정을 통해 임의의 대안 개체를 자동으로 생성한다. 이 과정을 통해 [예제 6]의 질의는 $FridayDiscount(CheesePizza01)$을 결과로 수반할 수 있다.

① KB^2_{OWA}에서 문제가 되는 개념 추출

KB^{2+}_{OWA} =

$\{ CheesePizza \subseteq \exists topping.CheeseTopping,$ ····· (1)

$CheesePizza(CheesePizza01) \}$ ····· (2)

② 서술논리기반 존재 양화사를 일차논리로 변환

$topping(X, Y), CheeseTopping(Y)$ ····· (1′)

③ 임의의 대안 개체 생성

$topping(CheesePizza01, anon01),$

$CheeseTopping(anon01)$ ····· (1″)

본 연구에서의 OWA와 CWA 지식베이스를 통합한 추론은 OWL-DL 온톨로지와 규칙의 지식베이스의 표현력을 잃지 않는 범위에서 통합 지식베이스 기반 추론을 수행한다. 태블로 알고리즘을 기반으로 한 OWL-DL 추론 결과에서 상식적으로 정당하지만 불일치하게 나오는 개념들에 대해 규칙 기반의 추론을 수행한다. 규칙 기반의 추론 시에도 OWL-DL의 표현력이 필요한 추론이 나오게 되면 OWL-DL 지식베이스를 참조하며, 이 두 개의 OWA와 CWA의 지식베이스를 서로 상호작용하여 추론 결과를 수반한다. 이 방법론은 2005년을 기준으로 변경된 W3C의 시맨틱 웹 계층도를 지향한 것이다.

5. 실험 및 평가

5.1 OWA와 CWA 통합 지식베이스를 이용한 추론 실험

그림 5는 본 연구에서 제안하는 OWA와 CWA의 지식베이스를 통합한 실질적인 추론 시스템을 보여 준다. 이 시스템은 클래스들의 포함관계에서 모순이 발생하는 경우와 인스턴스와 연관이 있는 질의응답에서 모순이 발생하는 경우를 다룬다. 이 두 기능을 [Class Reasoning]과 [Instance Reasoning] 탭으로 각각 구분하여 제공한다.

5.1.1 실행 과정

첫째로 본 절에서는 [Class Reasoning]에서 제공하는 기능을 살펴본다(그림 5). 지식베이스를 OWA 기반과 OWA+CWA 기반으로 구분하고, 다음과 같은 기능을 제공한다.

① 상단의 [File] 메뉴를 통해서 구축한 온톨로지 파일을 열면 OWA KB의 [Class Hierarchy]영역에 설계자가 정의한 클래스 계층도가 보인다. 다음으로 상단 메뉴의 [Reasoning] → [OWA Reasoning] → [Classification]을 선택하면, 설계자가 정의한 클래스의 description을 반영하여 서술논리 기반으로 추론된 클래스 간의 계층도가 [Classification] 영역에 보인다. OWA KB부분은 온톨로지 편집기인 Protege에서의 추론과 동일하다.

② 하단의 [Diagnosis Result of OWA Reasoning]에는 온톨로지 추론 결과로 일관성이 깨지는 클래스들(unsatisfied classes)을 출력하고, 이유를 [Conflict Superclasses]에 출력한다. 온톨로지 설계자는 일관성이 깨지는 클래스(들)를 살펴보고, 예외로 처리할 클래스를 식별하여 [Exception Concept] 필드의 체크박스를 선택한다.

③ [OWA+CWA KB] 영역에는 예외로 선택한 클래스들을 CWA 기반으로 예외 처리하여 추론한 결과를 보여준다. 상단 메뉴의 [Knowledge] → [Integrate OWA&CWA]를 선택하면, 예외로 선택한 클래스들의 디폴트 규칙을 4.2.3.1절에서 설명한 알고리즘을 이용하여 생성하여 [Default Rules for Exceptions] 영역에 출력한다. 그리고 [ASP] 영역에는 온톨로지를 ASP 표현 형식으로 변환한 표현식에 디폴트 규칙이 삽입되어 출력된다. 마지막으로 상단 메뉴의 [Reasoning] → [CWA Reasoning] → [Classification]를 선택하면 오른쪽 [Classification] 영역에 디폴트 규칙이 적용된 클래스 계층도가 출력된다. [ASP]와 [Classification]에는 OWA와 CWA의 지식베이스가 결합된 형태를 보여준다.

5.1.2 실험 결과

2.3절에서 서술논리 표현의 한계를 살펴보고, 그림 1의 문제점을 4.2.3절의 [예제 3](KB_{OWA})에서 살펴보았다. 그림 5의 시스템은 이 한계의 해결책을 제시한다.

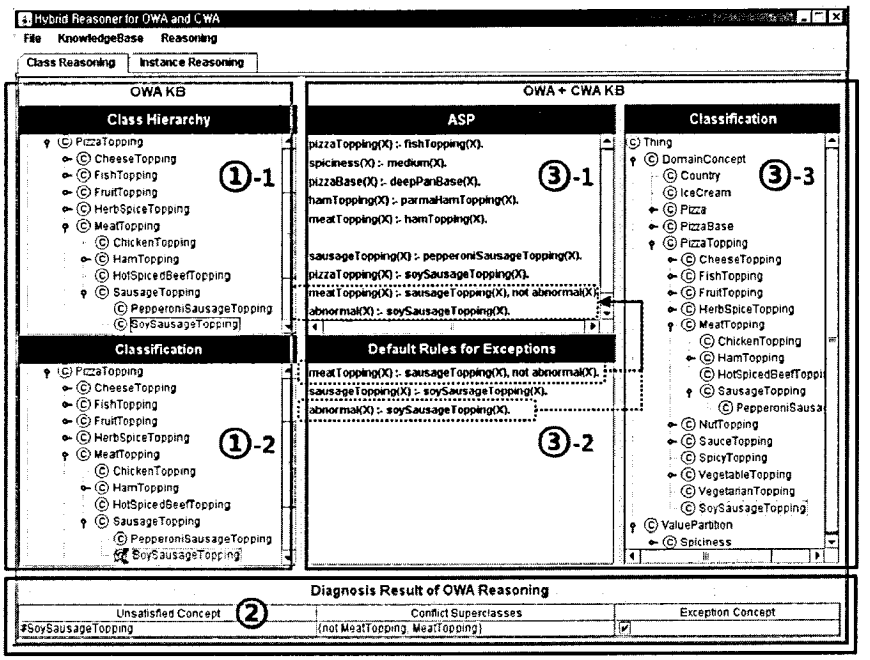

그림 5 OWA+CWA 지식베이스 통합 실험 결과: 온톨로지의 클래스명은 대문자로 시작하고 ASP의 atom과 규칙의 첫 문자는 소문자로 정의된다.

①-2는 이 문제를 발견한 것이다. 그리고 이 문제를 해결하기 위해 5.1.1절의 ②에 의해 콩 소시지를 예외로 처리하고 추론(③)하면, 디폴트 규칙 DR_{CWA}이 자동 생성되고 KB_{OWA}와 통합되어 KB^{+}_{OWA}가 된다. 이 통합 지식베이스는 ASP로 변환되어 그림 5의 ③-1에 출력된다. 이 때 "$sausage\,Topping(X)$:– $soySausage\,Topping(X)$." 문장은 KB_{OWA}에 이미 포함되었던 지식으로 KB^{+}_{OWA}에 중복 추가되지 않는다.

DR_{CWA} = {$meat\,Topping(X)$:– $sausage\,Topping(X)$, $not\ abnormal(X)$.

$abnormal(X)$:– $soySausage\,Topping(X)$.

$sausage\,Topping(X)$:– $soySausage\,Topping(X)$.}

KB^{+}_{OWA} = KB_{OWA} ∪ DR_{CWA}

그림 5의 ③-3은 OWA와 CWA의 지식베이스가 통합된 후의 클래스 Classification 결과이다. OWA기반 추론에서 unsatisfied 클래스로 추출된 *SoySausage Topping*이 충돌의 원인이었던 *meat Topping*의 상위 개념인 *PizzaTopping*의 하위로 위치하여 일관성을 유지한 것을 볼 수 있다.

KB^{+}_{OWA} 지식베이스를 ASP 프로그램으로 실행하면 Stable Model이 '참'인 일관성 있는 지식베이스로 판정된다. 이 지식베이스는 완전한 지식 모델이고, *ABox*와 같은 인스턴스가 입력되면, 지식베이스에서 참인 명제들의 결과로 Stable Model, *SM*를 반환한다.

ABox = {$soySausage\,Topping(redSoySausage)$.
$sausage\,Topping(currySausage)$.}

SM = {$soySausage\,Topping(redSoySausage)$
$sausage\,Topping(currySausage)$
$sausage\,Topping(redSoySausage)$
$abnormal(redSoySausage)$
$meat\,Topping(currySausage)$}

이 OWA와 CWA의 결합은 $KB_{OWA} \subseteq KB^{+}_{OWA}$가 성립하고, KB_{OWA}에서 몰랐던 $redSoySausage$가 $soy\,Sausage\,Topping$이라는 새로운 사실을 알게 되어 KB^{+}_{OWA}에 입력되면, $redSoySausage$가 $meat\,Topping$이라는 사실을 더 이상 믿지 않는 비단조 추론이 성립한다. OWA와 CWA 지식베이스의 결합은 인스턴스(KB의 ABox)의 질의응답에서 결정가능성을 유지한다. 예를 들어, 위의 예제에서 고기 토핑을 검색하면 stable model, *SM*의 결과를 참조하여 $currySausage$만을 반환한다. 이와 같은 인스턴스 추론 실험은 5.2절에서 살펴본다.

5.2 OWA와 CWA 기반 질의응답(Query Answering) 실험

본 절에서 살펴볼 실험은 OWA와 CWA가 통합된 완전한 지식베이스를 기반으로 결정 가능한 결과를 보장하는 질의응답 실험이다. 질의는 온톨로지에 정의된

클래스이며, 이 실험은 제안한 클래스의 무결성 유지 방법을 검증한다. 이 방법은 두 가지로 나뉜다. 질의 클래스가 무결성이 깨져서 결과가 완전하지 않을 때와 질의 클래스의 필요조건 중 존재 양화사의 무결성이 깨져서 결과가 완전하지 않을 때의 해결 방법이다.

5.2.1 실행 과정

본 절에서는 그림 5의 [Instance Reasoning] 탭에서 제공하는 기능을 살펴본다. 인스턴스 추론은 OWA와 CWA가 통합된 지식베이스를 기반으로 추론되며 그림 6에서 이를 위한 편집기를 보여준다.

① 왼쪽의 [Class Hierarchy] 영역은 [Class Reasoning] 탭의 ③-3과 동일하다.

② 중간의 [ASP] 영역의 ②-1은 [Class Reasoning] 탭의 ③-1과 동일하다. ②-2~5는 온톨로지의 무결성을 유지하기 위한 옵션이다. 우선, ②-2과 ②-3은 4.2.4절에서 살펴본 특정 클래스의 무결성을 유지하기 위한 규칙 편집기이다. ②-2에서는 무결성 유지의 대상이 되는 클래스를 편집할 수 있고, ②-3에서는 그 클래스의 무결성 제약 조건의 결론(head)에 해당하는 클래스들의 디폴트 규칙을 편집할 수 있다. 이 편집창들은 규칙 템플릿을 제공하고, 상단의 [Class]와 [Property] 버튼을 통해서 온톨로지에 정의된 클래스와 속성을 찾아서 입력할 수 있다. 그리고 [Variable]와 [Abnormal] 버튼을 통해서 규칙에 들어가는 변수와 abnormal을 쉽게 편집할 수 있다. 규칙의 편집이 끝나고, 오른쪽 체크 박스를 선택하여 상단의 [↑ ASSERT ↑] 버튼을 선택하면 OWA+CWA 지식베이스에 추가되어 ②-1에 출력된다.

다음으로 ②-4와 ②-5는 4.2.5절에서 언급한 클래스에 정의된 존재 양화사의 무결성 유지를 위한 편집기이다. ②-4는 SWRL과 같은 규칙 편집기로써 서비스 사용자가 정의하는 새로운 규칙을 편집할 수 있다. ②-5는 새로운 규칙의 조건절에 포함되는 속성과 속성의 range 클래스를 필요조건으로 가지는 클래스들의 무결성을 유지하기 위한 선택 옵션을 제공한다. 마찬가지로 두 제약을 편집하고 오른쪽 체크박스를 선택하여 [↑ ASSERT ↑] 버튼을 누르면 OWA+CWA 통합 지식베이스에 추가된다.

③ 질의응답은 OWA와 CWA기반의 통합된 지식베이스를 기반으로 제공한다. 즉, SHIQ 수준의 표현력을 지닌 온톨로지 스키마 추론 결과를 반영하면서, 특정 클래스의 무결성 유지를 위한 closed-world 추론도 지원한다. Closed-world 추론은 4.2.2에서 언급한 ASP 알고리즘을 이용하여 stable model을 구하고, 그 중 질의 클래스를 predicate로 가지는 상수, 즉 인스턴스(들)를 추출하여 결과로 반환한다.

5.2.2 클래스의 무결성 유지를 위한 실험 결과

4.2.4절에서 온톨로지에 정의된 클래스의 무결성이 깨지는 경우를 살펴보았다. 그림 6은 4.2.4절 [예제 4]의 *Pizza* 개념에 무결성을 유지한 지식베이스를 보여준다. 4.2.4.1절에서 언급한 방법대로 무결성 제약 조건(KB^1_{CWA} : 피자는 *SpicyDish*이거나 *NonSpicyDish*이다)과 그에 따

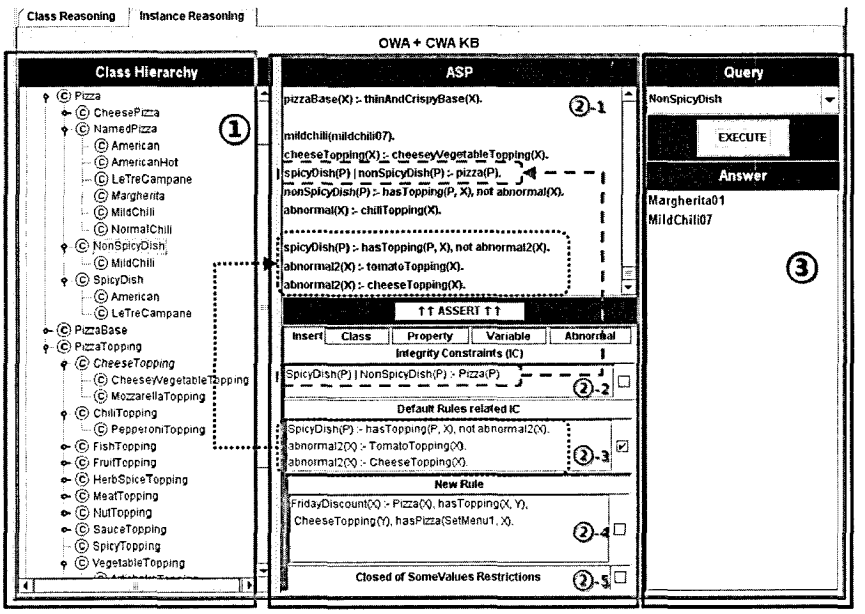

그림 6 질의응답 실험 결과 1

른 디폴트 규칙(DR^1_{CWA})을 ②-2와 ②-3을 통해 편집하고, 오른쪽의 체크 박스를 선택하여 ↑ ASSERT ↑ 하면 OWA+CWA 지식베이스에 추가되어 ②-1을 통해 출력된다.

무결성 제약의 추가 전과 후를 비교해보자. 4.2.4절의 KB^1_{OWA} 기반의 온톨로지 추론 결과로는 *MildChili*에 속하는 인스턴스만을 *NonSpicyDish*로 추론(즉, ①의 "Class Hierarchy"의 *NonSpicyDish* 클래스는 *MildChili*만을 하위 클래스로 추론)한다. 반면에 ③의 질의에서는 *Margherita* 클래스의 인스턴스인 *Margherita*01도 결과로 수반한다. 이유는 4.2.4절 KB^1_{OWA}의 *Margherita* 피자의 description을 보면, *Margherita* 피자는 *Tomato*와 *MozzarellarTopping*을 하나씩은 가져야 한다고 정의되어 있다. 그리고 ②-1에 추가된 *NonSpicyDish*의 디폴트 규칙(DR^1_{CWA})에 *ChiliTopping*만 아니면 된다고 정의되어 있으므로, *Margherita* 피자는 *NonSpicyDish*로 추론된다.

$$DR^1_{CWA} = \{nonSpicyDish(P) :- hasTopping(P, X), not\ abnormal(X).$$

$$abnormal(X) :- chiliTopping(X). \quad \cdots\cdots (1)$$

$$spicyDish(P) :- hasTopping(P, X), not\ abnormal2(X).$$

$$abnormal2(X) :- tomatoTopping(X). \quad \cdots\cdots (2)$$

$$abnormal2(X) :- mozzarellaTopping(X).\} \cdots\cdots (3)$$

본 연구는 4.2.4.2절에서 언급한대로 CWA 기반의 디폴트 규칙 생성 시, OWA기반의 온톨로지의 표현력도 유지시킨다. 즉, 디폴트 규칙으로 정의되면, *abnormal* 클래스의 (정의 하였거나 추론된) 하위 클래스도 자동으로 *abnormal*로 처리한다. 예를 들어, 그림 6의 ①의 *ChiliTopping*의 하위 개념인 *PepperoniTopping*도 *abnormal*로 처리하여, *PepperoniTopping*을 갖는 피자는 *SpicyDish*로 추론된다. 이는 서술논리 표현력을 유지한 것이다.

5.2.3 클래스의 필요조건 내 존재 양화사(Existential Quantifier)의 무결성 유지 실험 결과

4.2.5절에서 존재 양화사의 무결성 유지 방법을 살펴보았다. 그림 7은 4.2.5절에서 살펴본 [예제 6]의 실험 결과를 보여준다. ②-4의 규칙 편집기에서 4.2.5절의 SWRL 규칙을 편집하고 ②-5의 옵션을 선택하여 삽입하면, ②-1 지식베이스에 규칙과 규칙에 관계된 존재 양화사 조건을 가지는 클래스들의 무결성을 유지시키는 규칙이 추가된다. *CheesePizza*와 서술논리 특징에 의해 추론된 *American* 피자의 인스턴스들이다. 만약에 ②-5의 옵션을 선택하지 않는다면 온톨로지 기반의 SWRL 규칙 수행 시와 같은 결과가 반환된다.

다음으로 ③에서 규칙의 결론 클래스(*FridayDiscount*)를 질의로 입력하면, ②-5에 의해 유도된 규칙에 의해 *CheesePizza*01이 결과로 수반된 것을 볼 수 있다. 그리고 그림 7의 ①의 온톨로지 계층 구조를 참조해보면, 서

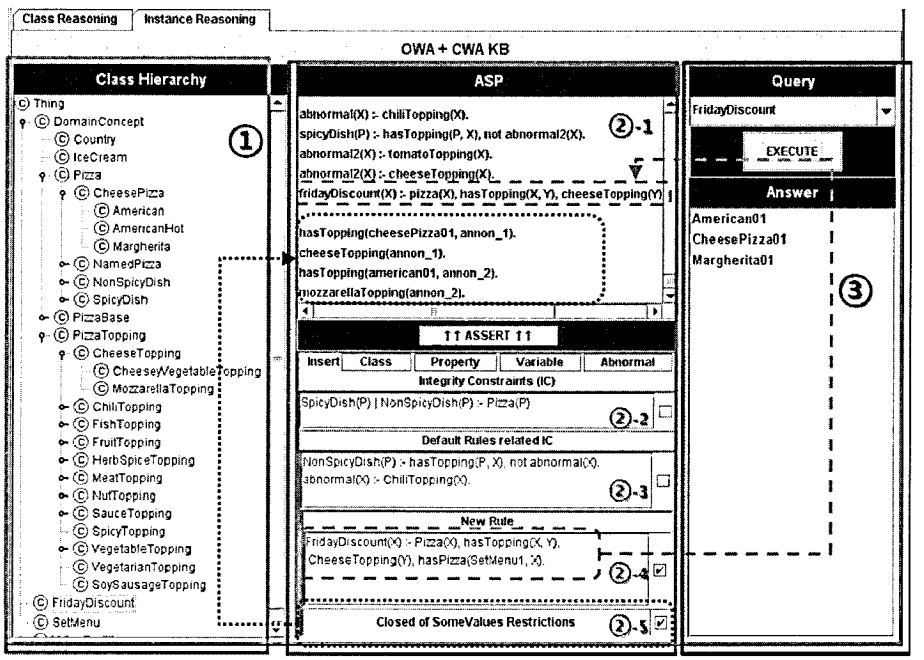

그림 7 질의응답 실험 결과 2

술논리의 특징에 의해 *CheeseTopping*의 하위 클래스인 *MozzarellaTopping*의 인스턴스를 가지는 *Margherita01* 피자도 결과로 수반된 것을 볼 수 있다. 또한 *American01*은 *CheeseFizza*의 하위 클래스로 추론되고 *MozzarellaTopping*을 가지므로 결과로 수반된다. 이와 같이 온톨로지 추론 기술은 명백하게 정의된 지식으로부터 암시적인 새로운 지식을 추론할 수 있다. 이는 서술논리의 큰 장점이며 OWA와 CWA기반 지식베이스 통합 시에 유지되는 OWA의 특징이다.

5.3 성능 평가

제안한 방법의 실현 가능성을 검증하기 위해 본 연구에서는 OWA 기반 온톨로지 지식베이스에 CWA 기반 규칙을 통합하여 open-world와 closed-world 추론을 결합한 실질적인 시스템을 구현하였다. OWA 추론에 사용한 엔진은 본 논문의 선행 연구로써 구현된 Minerva [17]이다. 본 실험은 2GB 메모리를 장착한 2.4GHz Pentium PC의 윈도우 환경에서 실행하였다. 실험에 적용한 데이터는 잘 알려진 프로젝트에서 공개한 온톨로지들이며, 이 중 closed-world 추론이 결합되었을 시에 제안하는 시스템의 성능 검증에 변별력이 있는 예제들을 표 1에 나타내었다. 표 1에 포함된 예제는 Protege의 Pizza, OWL 가이드[7]에 나오는 Food와 Wine, Mindswap의 Galen, buggy_sweet, terrorism, 그리고 NASA의 SWEET set의 phenomena 온톨로지이다.

표 1은 실험 온톨로지와 CWA기반 지식베이스를 기반으로 질의응답 실험을 한 결과이다. CWA 추론이 필요한 질의가 들어오면, 디폴트 규칙을 고려하여 ASP 프로그램을 수행한다. 그리고 첫 번째 stable model에서 응답 predicate를 찾아서 그림 6(또는 7)의 ③에 결과로 출력하는 시간(초, second)을 보고한 표이다. 이 결과는 제안하는 open과 closed-world 추론의 통합 방법이 다양한 서술논리 표현력을 가지는 온톨로지와 CWA기반 규칙을 결합한 지식베이스에 따라 영향을 받지 않고 합당한 성능을 보여줌을 입증한다. 표 1은 OWL 온톨로지별로 크기, 클래스/클래스들의 포함 관계/인스턴스 수, ASP 프로그램 수행 시 grounded program(즉, 변수가 없는)에서의 atom과 규칙(괄호 안은 디폴트 규칙의 수) 수, 응답까지 소요된 시간(time), 그리고 질의응답 분석 결과이다. time은 각 atom(클래스, 인스턴스, 인스턴스 간의 관계 등)들이 결정 가능한 결과(yes 또는 no)를 반환한 합리적인 시간을 보여준다. 질의응답 분석 결과는 디폴트 규칙 중 head에 있는 클래스를 질의로 하였을 때(그림 6과 7의 ③)의 정답 set을 만들고, 시스템이 도출한 결과와 결과 중 맞은 개수를 비교한다.

표 1의 결과는 온톨로지의 표현력과는 상관없이, 디폴트 규칙에 영향을 받아 추가되는 ground atom과 규칙의 수는 일정한 비율로 증가하지만 수행 시간은 그들이 증가하는 비율에 비해 근소한 차이를 보임을 분석 할 수 있다. phenomena, terrorism의 온톨로지는 atom과 규칙이 디폴트 규칙에 의해 두 배 증가하였지만, 수행시간은 그 보다 적게 증가했다. 또한 근소한 차이로 증가한 debug_sweet와 Galen은 거의 수행 시간차이가 없다. Pizza 온톨로지의 경우도 디폴트 규칙이 0개에서 2개로 증가하여 atom과 규칙이 2배 이상 증가하였지만 수행시간은 그에 비해 비교적 적게 증가했다. 0개에서 4개로의 디폴트 규칙에서는 일정한 비율로 증가하였지만,

표 1 질의응답 성능 평가

OWL 파일	크기	class / subClass / instance 수	ASP		time	질의	
			atom 수	규칙 수 (디폴트 규칙 수)		정답	맞은수 /결과
Pizza	155K	113 / 266 / 103	861	1,056(4)	0.19	21	21/21
			854	1,049(2)	0.14	32	32/32
			389	388(0)	0.09	-	-
Food	51K	65 / 102 / 67	440	457(2)	0.09	12	12/12
			378	377(0)	0.04	-	-
Wine	102K	81 / 126 / 171	833	862(3)	0.19	20	20/20
			807	806(0)	0.14	-	-
Galen	2.3M	2,749 / 3,238 / 108	7,097	7,292(4)	6.9	23	23/23
			6,625	6,624(0)	6.4	-	-
buggy_sweet	520K	1,537 / 1,922 / 350	4,709	4,898(2)	3.4	53	53/53
			4,694	4,693(0)	3.4	-	-
terrorism	45K	32 / 38 / 245	1,831	2,310(2)	0.23	no	no
			863	862(0)	0.18	-	-
phenomena	125K	2,364 / 3,102 / 621	4,488	5,403(2)	2.3	7	7/7
			2,603	2,602(0)	1.8	-	-

실험 결과는 평균적으로 Log함수의 그래프 형태를 나타낸다. 이는 온톨로지 추론 결과, 즉 온톨로지 표현력은 유지하고 CWA기반 규칙만 추가하여 추론하는 방법의 장점이다. 즉, 디폴트 규칙의 개수 보다는 디폴트 규칙에 적용되는 인스턴스 수에 따라 atom과 규칙의 수만 증가하여 합리적인 시간에 결과를 도출하는 것을 알 수 있다. 이 성능 평가 실험을 통해서 OWA와 CWA 기반 KB를 위한 질의에 대한 답변을 찾는 과정이 정당하고 완전함을 보장할 수 있었다. 질의응답 분석 결과에서 질의에 대한 정답 set과 시스템 결과를 비교하여 오차범위 없이 완벽한 set을 도출하였고, 이 결과로써 제안한 추론규칙의 완전하고 정당함을 증명할 수 있었다.

본 연구는 사람의 상식에 가까운 높은 성능의 추론이 가능한 풍부한 표현력을 유지하면서, 결정 가능한 효율적인 질의응답이 가능한 온톨로지를 구성할 수 있도록 지원한다. 이 연구는 시맨틱 웹 기반 실질적인 시스템이 요구되는 이 시점에서, 시맨틱 웹을 활용한 서비스 어플리케이션을 구현하는데 있어서 효율적인 시간 내에 서비스 제공을 현실화시키는 방법으로 확신한다.

6. 결론 및 향후 연구

본 논문은 온톨로지 기반 시맨틱 웹 서비스에서 발생할 수 있는 예외(exception) 처리와 질의응답을 위한 완전한 지식 모델링을 지원하기 위해 open world와 closed-world 추론을 통합한 실질적인 시스템을 제안한다. 제안한 시스템은 구축된 OWL 온톨로지의 OWA 기반 추론 시 서술논리가 가지는 의미(semantics)와 사람의 생각 차이로 인해 일관성이 깨지는 개념(들)을 예외로 처리한다. 예외는 논리 프로그램 언어인 ASP를 이용하여 디폴트 규칙으로 생성하고, 온톨로지 지식 베이스와 결합하여 무결성 유지와 결정가능성을 보장한 추론을 가능하게 한다. 현재까지 OWA와 CWA 지식베이스를 통합한 추론을 제안하는 연구들은 많이 있었지만, 하나의 시나리오에만 국한되어 실질적인 시스템을 제안하지 못 했다. 하지만 본 논문은 온톨로지의 표현력을 유지하면서 예외처리와 완전한 온톨로지 모델링을 지원하여 도메인이 정해진 웹서비스에 합당한 시맨틱 검색을 가능하게 한다. 향후 연구로는 온톨로지 편집기능(클래스, 속성, 인스턴스, 클래스 제한조건)을 추가하여 OWA 지식베이스를 쉽게 편집할 수 있게 하고, closed-world 추론을 위한 다양한 기능을 추가하여 실세계의 지식표현의 범위를 넓히는 것이다. 본 시스템은 시맨틱 웹 기반의 응용 서비스를 구축하는 온톨로지 설계자라면 반드시 부딪히는 문제를 풀기 위한 새로운 해결책이 될 것이다.

참 고 문 헌

[1] G. Antonious, C.V. Damasio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, P.F. Patel-Schneider, "Combining rules and ontologies," *A survery. Technical Report IST506779/Linkoping/I3-D3/D/PU/a1, Linkoping University*, 2005.

[2] J.Z. Pan, E. Franconi, S. Tessaris, G. Stamou, V. Tzouvaras, L. Serafini, I. Horrocks, B. Glimm, "Specification of coordination of rule and ontology languages," *Project Deliverable D2.5.1, Knowledge-Web NoE*, 2004.

[3] T. Eiter, G. Ianni, A. Polleres, R. Schindlauer, H. Tompits, "Reasoning with Rules and Ontologies," *In Springer LNCS 4126*, pp.93-127, 2006.

[4] T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits, "Combining Answer Set Programming with Description Logics for the Semantic Web," *Proc. of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR2004)*, pp.141-151, 2004.

[5] R.J. Brachman, H. J. Levesque, "Knowledge Representation and Reasoning," *Morgan Kaufmann*, 1991.

[6] D. Poole, "A logical framework for default reasoning," *Artificial Intelligence*, vol.36, pp.27-47, 1988.

[7] M. Smith, C. Welty, D. McGuiness, "OWL Web Ontology Language Guide," *W3C Recommendation http://www.w3.org/TR/owl-guide/*, 2004.

[8] V. Lifschitz, "Nonmonotonic Databases and Epistemic Queries," *Proc. of the 12th IJCAI*, pp. 381-386, Aug. 1991.

[9] B. Motik, I. Horrocks, R. Rosati, U. Sattler, "Can OWL and Logic Programming Live Together Happily Ever After?," *Proc. of the 5th International Semantic Web Conference (ISWC 2006), vol.4273 of LNCS*, pp.501-514. Springer, 2006.

[10] Y. Zhang, "Epistemic Reasoning in Logic Programs," *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp.647-652, 2007.

[11] U. Hustadt, "Do we need the closed-world assumption in knowledge representation?," *Proc. of the 1st Workshop KRDB'94*, pp.123-124, 1994.

[12] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, Y. Katz, "Pellet: a practical owl-dl reasoner," *Proc. of the 3rd International Semantic Web Conference (ISWC2004)*, 2004.

[13] S. Grimm, B. Motik, "Closed-World Reasoning in the Semantic Web through Epistemic Operators," *Proc. of the OWL Experiences and Directions Workshop*, 2005.

[14] C.V. Damasio, A. Analyti, G. Antoniou, G. Wagner, "Supporting Open and Closed World Reasoning on the Web," *Proc. of the Principles and Practice of Semantic Web Reasoning (PPSWR'06), vol.4187 of*

LNCS, pp.149-163, Springer, 2006.
[15] F.M. Donini, D. Nardi, R. Rosati, "Description Logics of Minimal Knowledge and Negation as Failure," *ACM Trans. on Computational Logic*, 3(2), pp.177-225, 2002.
[16] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," *W3C Member Submission*, 21 May 2004. Available at http://www.w3.org/Submission/SWRL/. 2004.
[17] J.-M. Kim, S.-H. Kwon, J.-H. Choi, Y.-T. Park, "Tableaux Algorithm based OWL Ontology Reasoner," *Proc. of the 35th KIISE Fall Conference*, vol.35, no.1(A), pp.0102-0103, 2008. (in Korean)

최 정 화

2004년 2월 숭실대학교 정보과학대학 컴퓨터학부 졸업(학사). 2006년 2월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2006년 3월~현재 숭실대학교대학원 컴퓨터학과 박사수료(논문과정). 관심분야는 유비쿼터스 컴퓨팅, 시맨틱 웹, 온톨로지 추론, 시맨틱 Annotation, 다중 에이전트 시스템, Planning, Android 등

박 영 택

1978년 서울대학교 전자공학과 졸업 1980년 KAIST 전산학 석사 학위 취득 1992년 University of Illinois at Urbana-Champaign 박사 학위 취득. 1981년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 시맨틱웹, 온톨로지 추론, 유비쿼터스 컴퓨팅, 개인화 에이전트, 기계학습, 전문가 시스템, Conceptual Clustering 등