

다양한 OWL-DL 추론 엔진에서 대용량 ABox 추론에 대한 성능평가

(A Performance Analysis of Large ABox Reasoning in OWL-DL Reasoners)

서 은 석 [†] 박 영 택 ^{**}
(Eun-Seok Seo) (Young-Tack Park)

요 약 RacerPro, Pellet 등 지금까지의 전형적인 추론 시스템들은 주로 Tableaux Algorithm 기반의 추론 시스템으로 Tableaux Algorithm의 특성상 대용량 ABox 추론에서 문제점을 나타낸다. 이를 해결하기 위한 연구로 Tableaux Algorithm 기반에 DBMS를 함께 사용한 영국 Manchester 대학의 Instance Store와 Disjunctive Datalog Approach를 사용한 독일 Karlsruhe 대학의 KAON2가 있다. 현재 추론 시스템들에 대한 벤치마크 실험은 대부분 Tableaux Algorithm 기반의 TBox 추론 위주이며 ABox 추론에 대한 평가는 거의 진행되지 않았다. 특히 최근 이슈로 부각된 (대용량 ABox 추론을 위한 추론 시스템)의 특성별 벤치마크 실험은 거의 보고되지 않았다. 이에 본 논문에서는 각 추론엔진들의 이론적 배경을 근간으로 전형적 추론엔진들과 최근 이슈에 따른 대용량 ABox를 위한 추론엔진들을 상호 비교를 통해 살펴보고 특히, 대용량 ABox 처리를 위한 추론엔진인 Manchester 대학의 Instance Store와 Karlsruhe 대학의 KAON2를 LUBM을 통하여 분석 평가함으로써 사용자의 요구에 따른 대용량 ABox 추론엔진을 제시한다. 평가방법에서는 LUBM(Lehigh University BenchMark)에 대한 소개와 이를 이용한 벤치마크 실험 방법 및 평가 시스템에 대하여 소개한다. 본 논문은 결론을 통해 실험 결과와 각 추론엔진의 사용 Algorithm 특성을 기초로 다양한 환경에서의 대용량 ABox 처리에 적합한 추론엔진을 제시한다.

키워드 : 시맨틱 웹, 추론 시스템, 대용량 ABox, ABox 추론, 성능평가

Abstract Reasoners using typical Tableaux algorithm such as RacerPro, Pellet have a problem in Tableaux algorithm large ABox reasoning. Researches to solve these problems are dealt with Instance Store of University of Manchester which uses Tableaux algorithm based reasoner and DBMS and KAON2 of University of Karlsruhe using Disjunctive Datalog approach. An evaluation experiment for present reasoners is the experiment of TBox reasoning in most of Tableaux algorithm based one. The most of benchmarking tests in reasoning systems haven't done with ABox reasoning based Tableaux Algorithm but done with TBox reasoning based Tableaux Algorithm. Especially, rarely reported benchmarking tests in reasoners have been issued nowadays. Therefore, this thesis evaluates systems with theory of each reasoners for large ABox reasoning that becomes issues recently with typical reasoners. The large ABox reasoning engine will be analyzed using Instance Store and KAON2 of Manchester University for large ABox processing. At the analysing method, LUBM(Lehigh University BenchMark), benchmarking test method, and it's test system will be introduced. In conclusion, I recommend appropriate reasoner in various environment with experiment result and characteristic of algorithm used for each reasoner.

Key words : Semantic Web, Reasoner, Inference System, Large ABox, ABox reasoning, Performance Analysis

· 본 논문은 숭실대학교의 지원을 받았습니다.

[†] 학생회원 : 숭실대학교 컴퓨터학과
erin214@ailab.ssu.ac.kr

^{**} 종신회원 : 숭실대학교 컴퓨터학과 교수
park@computing.ssu.ac.kr

논문접수 : 2006년 7월 3일
심사완료 : 2007년 5월 14일

1. 서론

추론 기술은 이미 존재하는 몇 개의 증거를 바탕으로 하여 새로운 어떤 사실이 성립됨을 결론으로 이끌어 내는 기술이다[1]. 최근 OWL(Web Ontology Language)[2]

온톨로지를 사용한 추론은 서술논리(Description Logic) [3]를 기반으로 한다.

실제 물리적 세계의 다양한 의미 정보를 표현한 온톨로지 기반의 추론 시스템들은 다양한 방법으로 의미 정보를 처리한다. 현재의 추론 시스템은 대부분 Tableau 알고리즘 기반의 TBox(Terminological) 추론에 초점이 맞춰진 시스템들이다. 하지만 최근 많은 양의 Individual들이 포함된 ABox 추론[4,5]이 이슈가 되고 있다. 왜냐하면 실제 물리적 세계를 기반으로 한 유비쿼터스 환경을 위한 온톨로지를 구성할 경우 매우 많은 양의 Individual들이 생성되어진다. 즉, 이와 같은 환경에서 추론 시스템은 대량의 Individual들을 처리할 수 있어야 한다. 그러나 Tableau 알고리즘 기반의 추론 시스템들은 알고리즘 특성상 발생하는 ABox 추론의 문제점을 가진다. Tableau 알고리즘을 사용한 ABox 추론에서는 Individual을 모두 트리에 대입하며 탐색영역을 확장하므로 트리 구조의 탐색 영역이 너무 크게 확장되어 결국 Forest를 이룬다. 따라서 Tableau 알고리즘 기반의 추론 시스템은 대용량 ABox 추론을 위하여 매우 많은 시간 비용을 요구한다. 이러한 문제점을 개선하기 위한 연구로 기존 Tableau 알고리즘 기반의 추론 시스템에 DBMS을 적용한 영국 Manchester 대학의 Instance Store[6]와 비 Tableau 알고리즘 기반의 Disjunctive Datalog Approach를 사용한 독일 Karlsruhe 대학의 KAON2[7]가 있다.

대용량 ABox 추론을 위한 시스템은 두 가지 기본적 요구사항을 가진다. 첫 번째, 중요한 이슈인 대량의 데이터에서 발생하는 범위성(Scalability)과 능률성(Efficiency)의 문제 그리고 두 번째, 충분한 의미적 표현능력(Expression Power)의 문제이다. 이 두 가지 요구사항은 추론 능력에 관한 문제이다. 추론 능력의 문제는 처리 복잡도와 처리 시간 사이에서 Trade-Off 관계를 갖는다. 즉, 추론 능력의 향상은 응답 시간의 증가로 이어진다. 따라서 지금까지의 추론 시스템들에게 요구된 각기 상반된 요구 사항들에 대한 적절한 해결 방안을 고려해야한다. 특히 범위성에 대한 요구사항의 처리는 더욱 어렵다. 주된 이유는 일부 시맨틱 웹 데이터 셋의 크기가 매우 크며 사용되는 온톨로지의 표현 범위가 크기 때문이다.

본 논문에서는 범위성에 대한 평가를 위주로 효율성을 살펴보고, 각 추론 시스템의 온톨로지 표현에 대한 처리 능력을 알아본다. 평가는 LUBM[8]을 기반으로 온톨로지의 TBox와 ABox를 구성하여 각 시스템의 평가 표준과 온톨로지 표현의 범위를 규정한다. 그리고 다양한 크기의 ABox와 그들의 관계(relation)를 구성하여 트리플에 따른 양적, 의미적 내용으로 추론 시스템의 성능을 평가한다[9-11].

널리 사용되는 Tableau 알고리즘 기반의 RacerPro [12]와 같이 주로 TBox 추론에 사용되는 시스템도 ABox 추론을 위한 엔진을 자체적으로 가지고 있다. 따라서, 평가에서는 대용량 ABox 추론을 위해 연구된 Instance Store와 KAON2의 평가를 기준으로, 널리 사용되는 여러 추론 시스템을 비교하여 대량의 ABox를 처리하기에 가장 적절한 추론 시스템을 제시한다. 또한 추론 시스템에는 추론에 있어 서술논리의 표현에 대한 제약이 존재한다. 예를 들어 KAON2의 경우, 서술논리의 표현력 중 SHIQ(D)를 지원하며, O(nominal value)를 처리하지 못한다. 이와 같이 추론 시스템에서는 서술논리의 표현력에 있어 각기 다른 제약을 가진다. 따라서 다양한 추론 시스템들을 각 시스템의 추론 알고리즘, 제약사항, 보고된 장/단점 등의 요소를 고려하여 평가한다.

본 논문은 실제 세계의 비즈니스 환경, 다양한 연구의 목적과 서술논리의 표현 범위 등 여러 가지 제약상황(환경)을 고려하여 각 사용에 적합한 추론 시스템 제시를 목적으로 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련연구로 다양한 추론 시스템을 평가하기 위하여 연구된 LUBM과 이를 이용한 추론 시스템 평가에 관하여 설명한다. 제3장에서는 서술논리에 대한 설명과 더불어 서술논리 기반의 온톨로지를 처리하기 위한 추론 시스템의 기본 알고리즘에 관하여 설명한다. 더불어 LUBM을 사용한 실험에 대하여 설명하며, 평가에 사용되는 추론 시스템인 RacerPro, Pellet[13], Instance Store, KAON2에 대한 설명과 각 시스템별 지원 가능한 서술논리의 표현 범위, 특성 그리고 제약사항 등을 설명한다. 제4장에서는 제시한 평가 방법에 따른 실험 내용과 결과를 기술한다. 마지막 제5장에서는 본 논문의 결론과 차후 연구 방향에 관하여 설명한다.

2. 관련 연구

본 장에서는 다양한 추론 시스템을 평가하기 위하여 연구된 LUBM과 이를 이용한 추론 시스템 평가에 관하여 알아본다. 1절에서는 추론 시스템을 평가하기 위하여 제안된 벤치마킹 방법인 LUBM에 대하여 설명하고, 2절에서는 추론 시스템 연구에서 LUBM을 사용하여 자체 평가한 내용에 관하여 기술한다.

2.1 LUBM(Lehigh University BenchMark)

LUBM은 다양한 추론 시스템에 대한 평가 방법을 제시한 것으로 추론 시스템의 평가에 있어 널리 사용되어 표준화된 평가 방법론이다. LUBM은 대학 도메인의 스키마 온톨로지 및 범위성 평가를 위한 임의의 크기의 OWL 데이터를 제공한다. 또한 다양한 프로퍼티를 포함

한 14개의 확장된 질의 패턴을 제공하며 질의는 SparQL[14]로 구성된다.

LUBM의 평가 온톨로지는 대학(University) 도메인으로 OWL의 서브셋 중 OWL-lite를 사용하여 총 43개의 클래스와 32개의 프로퍼티(25개의 오브젝트 프로퍼티와 7개의 데이터타입 프로퍼티)로 구성된다.

Unive-Bench 온톨로지에 대한 Individual을 생성하기 위하여 제공되는 생성기는 임의적이며 반복적으로 Individual을 생성한다. 생성되는 Individual은 각 대학 클래스별로 15에서 25개의 서브 파트를 가지며, undergraduate student와 faculty의 비율이 14 : 8로 구성된다. 또한 각 graduate student는 최소 1개에서 최대 3개의 course를 수강하도록 구성된다.

벤치마킹 구조도는 그림 1과 같다. 구조는 각 대상 추론 시스템에 객체화된 인터페이스를 만들고 평가 테스트 모듈은 이 인터페이스를 통하여 각 추론 시스템의 동작을 제어한다. 테스트 모듈은 각 추론 시스템 각각의 설정 내용과 질의 형태를 포함한다. 각 추론 시스템별 설정 사항들과 질의 형태를 각 시스템에 맞도록 미리 구성함으로써 온톨로지 입력시간 측정, 질의에 대한 결과 도출시간 측정 등 평가에서 정확성과 객관성을 유지할 수 있다.

LUBM은 추론 시스템의 평가 방법을 제시하면서 평가의 예로 4개의 추론 시스템을 평가하였는데 평가에 사용한 추론 시스템은 크게 메모리 기반 추론 시스템과 데이터베이스 기반 추론 시스템을 분류하여 메모리 기반 추론 시스템으로 Sesame-Memory와 OWLJessKB를 선택하고 데이터 베이스 기반 추론 시스템으로 DLDB-OWL와 Sesame-DB를 선택하였다. LUBM에서 예로 제시한 추론 시스템 평가에서는 그 평가의 목적이 메모리 기반 추론 시스템과 데이터베이스 기반 추론 시스템의 성능평가로 메모리 기반시스템과 데이터베이스 기반 시스템의 비교 평가를 목적으로 한다.

2.2 KAON2에서의 LUBM을 사용한 자체 성능 평가

KAON2는 기존 시스템들이 가지는 대용량 ABox 추론의 문제점을 해결하고자 Disjunctive Datalog Approach를 기반으로 연구된 추론 시스템이다. KAON2는 자체 알고리즘의 문제점을 확인하기 위하여 다른 서술논리 기반 추론 시스템과 KAON2의 성능을 평가하였다. 이에 온톨로지는 존재 한정(Existential quantifiers)과 논리 함(Disjunction)이 없는 VICODI[15], 매우 단순하지만 함수 규칙(Function role)을 포함한 SEMINTEC 그리고 확장된 질의를 포함하는 복잡한 형태의 LUBM을 사용한다. KAON2에서는 평가의 목적이 KAON2 알고리즘의 문제점을 확인하는 것으로 평가에 사용한 온톨로지도 부분적으로 사용하며 사용한 질의 역시 선택적으로 몇 가지만을 사용한다.

3. 서술논리 및 추론 시스템 알고리즘

본 논문은 많은 추론 시스템들에서 사용하고 있는 Tableaux 알고리즘에서 나타난 대용량 ABox 추론의 문제점의 이유를 이해하고 이를 해결하기 위한 연구인 Instance Store와 대용량 ABox 추론을 위하여 새로운 접근을 시도한 KAON2의 성능 위주로 기존 추론 시스템과 비교하여 평가한다. 먼저 서술논리를 지식 베이스와 함께 알아보고, 각 추론 시스템에서 사용하고 있는 추론 알고리즘을 알아본다[9-11].

3.1 서술논리

온톨로지 기반 추론에서 OWL-DL 온톨로지는 서술논리 기반으로 작성된다. 서술논리에 대한 연구는 1980년대 부터 진행되어 최근 OWL의 구조와 서술논리의 구조적 일치로 온톨로지를 추론하기 위한 배경 지식으로서 서술논리의 이해 필요성에 따라 많은 관심 속에 발전하였다. 일반적으로 우리가 서술논리로 지식을 표현할 경우 TBox와 ABox의 구조를 활용하여 지식을 표현한다. TBox는 지식 베이스의 스키마를 표현하며, ABox는 TBox의 스키

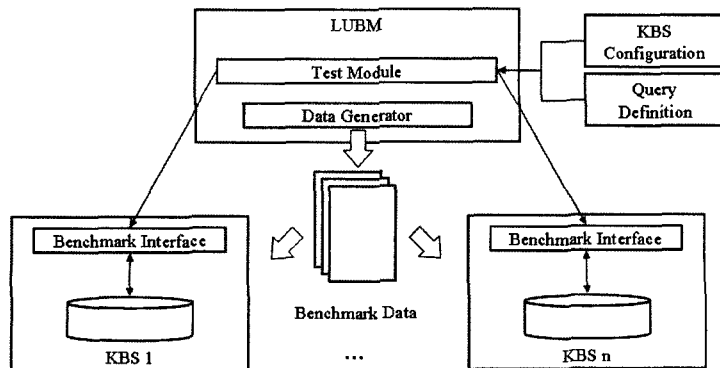


그림 1 벤치마킹 구조도

표 1 DL 지식 베이스의 TBox와 ABox

TBox	ABox
Mother =	Father(Peter)
Women \cap \exists hasChild.Person	Grandmother(Mary)
Parent = Mother \cup Father	hasChild(Mary, Peter)
Grandmother =	hasChild(Mary, Paul)
Mother \cap \exists hasChild.Mother	hasChild(Peter, Harry)

마에 대한 Assertion 데이터를 표현한다. TBox와 ABox에서 표현되는 지식은 다음의 표 1과 같다.

위의 표 1에서 알 수 있듯이 지식의 스키마가 TBox에 표현되고, ABox는 TBox 형식에 따른 각 Individual들에 대한 지식이 표현된다.

이와 같이 서술논리로 구성된 온톨로지를 사용한 전형적 추론 기법으로는 Tableaux 알고리즘을 사용한 방법이 있다. 다음 절에서는 Talbeaux 알고리즘에서 대용량 ABox 추론에 나타난 문제점의 이유와 이를 개선하기 위하여 Tableaux 알고리즘에 DBMS를 적용한 시스템을 살펴본다.

3.2 Tableaux 알고리즘

Tableaux 알고리즘의 기본 아이디어는 증명하고자 하는 어떤 것에 대하여 그에 대한 부정이 un-Satisfiable함을 증명하는 방식을 취한다. 일반적으로 지식 K에 대하여 포함관계를 표현할 때 C \sqsubseteq D로 표현하는데 이를 Tableaux 알고리즘에서는 \neg C \sqcup D가 un-Satisfiable함을 증명함으로 C \sqsubseteq D가 Satisfiable함을 증명하게 된다. 다음 그림 2는 Tableaux 알고리즘이 트리구조로 탐색영역을 확장해 가며 탐색영역을 모두 검색하여 증명하고자 하는 것의 부정이 un-Satisfiable함을 보임으로 포함관계를 추론하게 된다.

Tableaux 알고리즘의 탐색영역은 Expansion 규칙이 적용되어 트리구조로 확장하게 된다. 그리고 모든 탐색 공간의 중단 노드가 모순(Contradiction)임을 보임으로 증명하고자 하는 것의 부정이 un-Satisfiable함을 증명

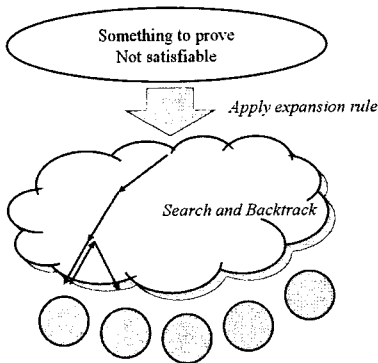


그림 2 Tableaux 알고리즘의 아이디어

하게 된다. 또한 증명을 위한 탐색영역 확장에서 트리구조의 중단 노드가 존재하지 않게 되는 문제를 Non-termination 문제라 하는데 이를 해결하기 위한 방법으로 Blocking 기법을 사용한다[16].

Tableaux 알고리즘은 위와 같은 방법으로 트리형태로 탐색영역을 확장하는 구조를 갖는데 이는 TBox 대상의 추론을 할 경우 다양한 최적화 기법들을 사용한 좋은 성능을 나타낸다. 이에 비해 대량의 ABox 추론을 할 경우 트리형태의 탐색영역 확장에 있어 모든 ABox의 Individuals을 대입하여 탐색영역을 확장함으로써 트리의 규모가 매우 커지며 처리에 대하여 많은 소모비용이 발생하게 된다. 즉, 방대한 트리구조의 Forest를 이룸으로써 효율성이 떨어지게 된다.

4. 대용량 ABox 추론을 위한 시스템

4.1 Instance Store

대부분의 추론 시스템이 사용하고 있는 Tableaux 알고리즘은 앞에서 언급한 바와 같이 대용량 ABox 추론을 위해 탐색영역을 확장하는 과정에서 트리가 매우 복잡해지며, 결국 Forest를 이루어 대용량 ABox 추론에는 부적합한 결과를 보여준다. 이를 해결하기 위한 연구 가운데 다른 추론 알고리즘을 사용하지 않고 Tableaux 알고리즘 기반으로 문제점을 개선하고자 접근한 연구가 Instance Store이다.

Instance Store는 대용량 ABox 추론을 위하여 영국 Manchester 대학에서 연구한 추론 시스템으로 Tableaux 알고리즘 기반 추론 시스템인 RacerPro, FaCT, FaCT++등을 DIG Interface를 통해 연동하여 사용한다. Instance Store는 연동하여 사용하는 추론 시스템을 통하여 온톨로지 클래스 레벨의 스키마를 정의하고 Individual들은 관계형 데이터베이스 시스템을 통하여 데이터베이스에 정의한다. 즉 기존 추론 시스템과 관계형 데이터베이스 시스템을 함께 사용함으로써 Tableaux 알고리즘의 대용량 ABox 추론에서 발생한 문제를 해결한다. Instance Store의 구조는 그림 3과 같다.

Instance Store는 추론 시스템을 사용하여 정의한 온톨로지의 스키마를 클래스(Concept) 이름과 서술부(Description)로 분리하여 데이터베이스에 저장한다. 이때 서술부는 별도의 테이블에 저장되며 이를 매핑하기 위한 ID 필드를 생성한다. 그리고 Individual들의 입력은 Individual 이름과 서술부를 분리하여 클래스와 동일한 방법으로 데이터베이스에 저장된다.

Instance Store는 관계형 데이터베이스 시스템의 Description 테이블을 중심으로 질의에 응답한다. Description 테이블에는 Individual들의 입력 과정에서 주어지는 Individual의 이름과 해당하는 서술부를 매핑하기 위

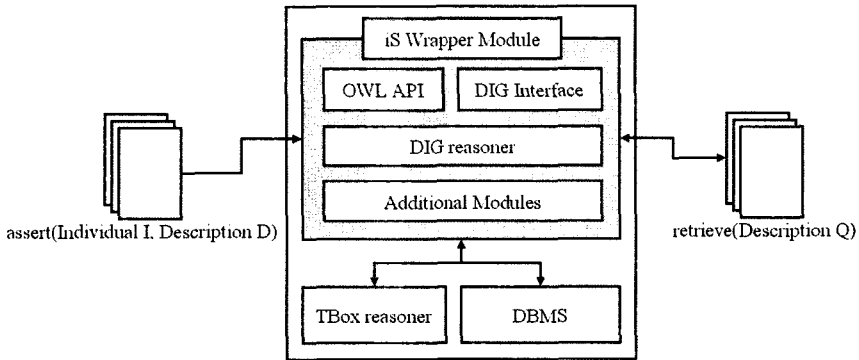


그림 3 Instance Store 구조도

한 ID와 서술부가 저장된다. Instance Store의 질의 형태는 aOWL(Abstract OWL)이며, 질의가 주어질 경우 질의를 Description 테이블의 서술부와 비교하여 동일한 서술부가 있을 때 이에 매칭되는 Individual의 이름을 반환한다. 이때 추론 시스템은 엄격히 필요한 때에만 사용되며 추론 시스템을 통하여 추론된 스키마 정보를 데이터베이스에 캐쉬하여 빠른 추론을 지원한다. 하지만 Description 테이블에 일치하는 결과가 없을 경우 Instance Store는 최악의 경우로 TBox 추론 시스템을 통하여 추론을 수행한다. 이 경우 앞서 설명한 Tableau 알고리즘의 특성으로 대량의 ABox 추론을 수행하기 위하여 Forest를 만들게 되어 매우 많은 시간이 소요는 문제가 발생한다.

4.2 KAON2(Disjunctive Datalog Approach)

독일 Karlsruhe대학의 KAON2는 대용량 ABox 추론에 초점을 맞추어 진행한 연구로 Disjunctive Datalog Approach를 사용한다. 이 방법론은 온톨로지를 자체 Reduction 알고리즘을 통하여 Datalog형태의 Clause로 변환한다. KAON2의 Reduction 알고리즘의 내부 단계는 그림 4와 같다.

온톨로지는 Mapping Axiom을 사용하여 FOL로 변환하고 이때 발생 할 수 있는 Equality의 처리를 위하여 Resolution을 확장한 Basic Superposition을 사용한다. 그리고 변환 과정에서 존재 한정자를 통해 발생 할 수 있는 Function Symbol을 제거함으로써 최종적으로 Disjunctive Datalog 지식 베이스로 변환된다. Reduc-

$KB = \{A \subseteq \exists R.B, B \subseteq C, \exists R.C \subseteq D\}$		
(1)	$\neg A(x) \vee R(x, f(x))$	
(2)	$\neg A(x) \vee B(f(x))$	
(3)	$\neg B(x) \vee C(x)$	
(4)	$D(x) \vee \neg R(x, y) \vee \neg C(y)$	
(5)	$\neg A(x) \vee D(x) \vee \neg C(f(x))$	$R((1), (4))$
(6)	$\neg A(x) \vee D(x) \vee \neg B(f(x))$	$R((5), (3))$
(7)	$\neg A(x) \vee D(x)$	$R((6), (2))$
(8)	$C(x) \rightarrow B(x)$	$R(3)$
(9)	$D(x) \rightarrow R(x, y), C(y)$	$R(4)$
(10)	$D(x) \rightarrow A(x)$	$R(7)$

그림 5 Reduction 알고리즘을 통한 변환

tion 알고리즘을 통한 온톨로지의 변환 예는 그림 5와 같다.

지식베이스에 존재하는 지식들에 대하여 형 변환 후 각 지식들에 의해 도출될 수 있는 모든 지식들을 도출한다. 이 과정에서 생성될 수 있는 Function Symbol들을 모두 제거하여 최종적으로 그림 5의 (8), (9), (10)의 결과를 얻게 된다. 이렇게 도출된 (8), (9), (10)은 처음 지식베이스의 지식들과 의미적으로 동일하다. 이를 매직셋이라 하며, 이 같은 이유로 도출된 결과를 추론에 그대로 적용할 수 있다.

Disjunctive Datalog Approach를 사용한 KAON2는 원패스 Bottom-up 방식으로 질의에 대한 결과를 미리 계산하고 적절한 ABox의 Facts와 변환된 매직셋을 바인딩하여 질의에 대해 Tableau 알고리즘을 사용한 추론 시스템보다 빠른 응답을 지원한다. 더불어 Disjunc-

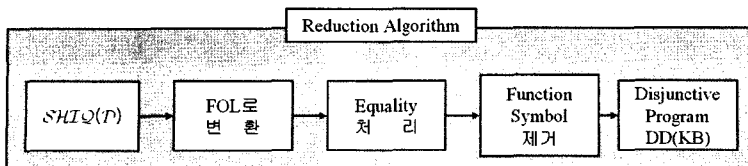


그림 4 KAON2의 Reduction 알고리즘

tive Datalog 방식은 효율적 최적화 방법을 적용하여 더욱 빠른 응답을 보인다.

5. LUBM 평가 방법에 따른 실험 및 결과

본 논문에서는 LUBM을 사용하여 Instance Store와 KAON2의 성능 평가를 목적으로 RacerPro와 서술논리의 가장 많은 표현력을 지원하는 Pellet을 평가하였다. 이에 본 장에서는 각 추론 시스템에 대한 실험 및 결과를 설명한다. 성능 평가의 주된 목적은 대용량 ABox 추론에 가장 적합한 현존하는 알고리즘 및 엔진을 제시로 한다. 따라서 추론 시스템별 알고리즘의 범위성(Scalability)을 위주로 평가한다. 이것은 많은 양의 온톨로지 데이터에 대한 의존성을 평가하는 것으로 질의에 대한 응답 시간을 확인한다.

5.1 평가 대상 추론 시스템

실험에서는 각 추론 시스템에 대하여 범위성에 대한 평가로 온톨로지의 ABox 규모가 커짐에 따른 추론 성능에 대한 평가를 진행한다. 대상 시스템은 Tableaux

기반 시스템과 non-Tableaux 기반 시스템을 각각 선택하였으며, 현재 가장 많이 사용되는 추론 시스템과 서술논리기반 추론에서 가장 높은 표현을 지원하는 추론 시스템을 선택하였다. 표 2는 각 추론 시스템별 특징을 나타낸다.

RacerPro는 Concordia University와 Hamburg University of Technology에서 개발한 LISP 기반의 엔진이다. 본 논문에서 사용한 RacerPro는 Version 1.9.0을 사용하였으며, 평가 시스템과 JRacer Library를 사용하여 연동하였다. 현재 RacerPro는 Tableaux 알고리즘을 사용한 상용 시스템으로 가장 많이 사용되는 추론 시스템이며, 서술논리의 SHIQ의 표현력을 가진다. 제공하는 질의 언어로는 nRQL을 지원한다.

Pellet은 University of Maryland에서 개발한 시스템으로 OWL-DL의 표현을 완벽히 지원하는 첫 번째 시스템이다. 사용한 시스템은 Version 1.3을 사용했으며, 제공되는 API를 사용하여 평가 시스템과 연동하였다. Pellet은 자바로 구현된 시스템으로 현재 Open Source

표 2 추론 시스템별 특징

시스템	항 목	내 용
Instance Store	지원언어	RDF, OWL
	표현력	SHIQ
	특징	- RacerPro, FaCT, FaCT++등의 추론 시스템과 DIG Interface를 통한 동작 - DBMS(MySQL, Oracle, HyperSonic등)와 연동 - Ontology가 TBox/ABox로 구분되어 입력
	질의언어	Abstract OWL
KAON2	지원언어	OWL
	표현력	SHIQ(D)
	특징	- Ontology 관리를 위한 하부구조(Infrastructure) - Novel Algorithm 기반 Disjunctive Datalog를 사용 (non-Tableaux 알고리즘) - File System 및 DBMS 사용 - Magic sets / Join-order Optimizations 기법 사용 - SWRL 지원
	질의언어	SparQL
RacerPro	지원언어	RDF, DAML+OIL, OWL
	표현력	SHIQ
	특징	- Server/Client 구성 - DIG Interface 사용한 HTTP/TCP Protocol 통신 - SWRL 지원
	질의언어	nRQL
Pellet	지원언어	OWL
	표현력	SHOIN(Q)
	특징	- DIG Interface 사용한 HTTP/TCP Protocol 통신 - Consistency: ABox의 consistency 검사 - Satisfiability: TBox의 satisfiability 검사 - Realization: Individual의 concept subsumption 검사 - Classification: TBox와 ABox의 restriction을 적용한 계층화
	질의언어	SparQL

로 제공된다. 제공하는 질의 언어로는 SparQL을 지원한다.

RacerPro와 Pellet 두 시스템 모두 첫 질의에 대한 응답 시 응답 이전에 ABox의 Consistency 검사를 수행한다[16].

5.2 실험 환경 구성

실험환경은 LUBM에서 제안하는 평가 방법에 의하여 구성되었으며 추론 시스템은 각기 발표된 최종 버전을 사용하고, Instance Store와 KAON2를 위한 DBMS로 MySQL(v5.0.1)을 사용한다.

평가에 있어 질의는 LUBM의 총 14개의 질의를 각 추론 시스템의 질의 언어로 변환하여 적용하며, 데이터셋은 LUBM에서 기본적으로 제공하는 Schema 온톨로지와 Instance 생성기를 통한 온톨로지 Instance의 조합을 사용한다. Instance 생성기를 사용하면 작은 크기의 다수의 ABox를 담은 파일이 생성된다. 따라서 평가를 위하여 생각된 각 파일을 통합하여 하나의 온톨로지 파일로 구성한다. 데이터셋은 각기 1~10개의 대학 클래스를 포함하며, 각 대학 클래스는 15~25개의 서브파트로 구성된다. 총 트리플의 개수는 1,052,895개이며 처음은 20,659개의 Individual에 대한 82,415개의 트리플로 실험을 시작한다.

사용된 컴퓨터 시스템은 운영체제로 WindowsXP 프로페셔널 서비스팩2가 적용되었으며 3.0MHz의 펜티엄4 프로세서와 1GByte의 메인 메모리, 그리고 200GByte의 HDD로 구성된다. 자바 개발 환경은 JDK 1.5.0이며, 각 추론 시스템은 Heap 메모리 영역을 기본 512MB, 최대 1GB로 구성한다.

5.3 실험 및 결과

본 논문의 실험에서는 LUBM을 기준으로 평가를 수행한다. 평가에 있어서 각 시스템에 있어 공정성을 잃지 않는 것이 중요하다. 하지만 각기 다른 특징을 갖는 시스템간의 평가의 공정성을 유지하기는 어려운 일이다. 이에 본 논문에서는 많은 평가 연구에서 사용한 LUBM을 사용하여 그 평가의 공정성을 유지하고자 하였다. 또한 사용한 실험 데이터는 LUBM에서 제공하는 데이터를 동일하게 사용함으로써 각 시스템 간의 형평성을 유지하고자 하였다. 평가에 사용된 LUBM 온톨로지는 Disjunction과 Number Restriction을 포함하고 있지 않다. 그러나 존재한정(Existential Quantifier)을 포함하여 추론의 복잡도가 높아진다. 평가 시스템은 그림 6의 인터페이스를 가지며, 그림 7의 흐름으로 동작한다.

시스템은 LUBM에서 제공되는 Data Generator를 통하여 데이터를 생성하고 Test Module 내부에서 생성된 데이터를 하나의 온톨로지로 통합하는 과정을 수행한다. 그리고 Test Module은 LUBM의 총 14개의 질의를 각

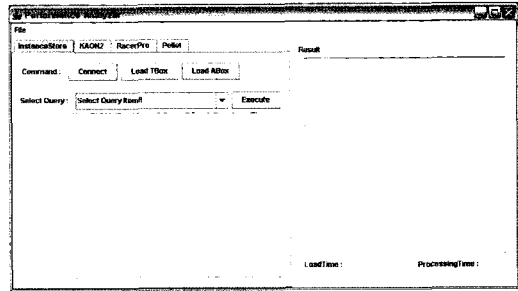


그림 6 평가 시스템 UI

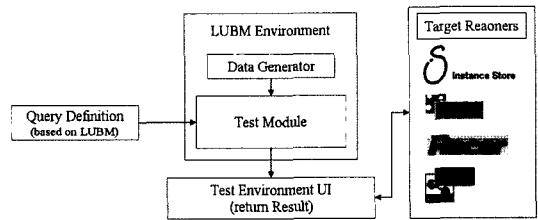


그림 7 시스템 수행 흐름도

추론 엔진의 질의 형식에 맞도록 정의하여 받아들인다. 이후 사용자 UI를 통하여 각 추론 엔진에게 각 질의가 입력되며, 처리 결과를 반환받는다.

각 추론 시스템에 있어 트리플의 수에 따른 평균 데이터 로드시간, 평균 질의응답시간은 다음과 같다.

표 3은 평가에 사용된 모든 추론 시스템들의 평균 데이터 로드시간 및 평균 질의 응답시간을 나타내며, 이에 대해 트리플의 수가 증가함에 따른 시간 증가를 그림 8에서 보여준다. 트리플의 수는 그 특성상 Individual의 수와 관계(Property)의 수에 밀접한 관련이 있다. 특히 관계의 수는 일반적으로 Individual의 수에 비하여 적다. 따라서 한정된 수의 관계가 존재할 때 Individual의 수에 따라 트리플의 수는 매우 민감히 변화한다. 그림 8에서와 같이 본 논문에서 평가하고자 했던 대용량 ABox 처리에 대하여 기존의 추론 시스템과 대용량 ABox 추론을 위한 시스템과는 현저한 차이를 나타낸다. 대용량 ABox에 대하여 유연히 대응한다는 것은 시스템이 입력 데이터에 대한 범위성이 뛰어나다는 것을 의미 하는데 위 결과를 통해 Instance Store와 KAON2는 범위성에 대한 유연성을 갖고 있음을 확인할 수 있다. 이는 입력 데이터에 관한 가장 기본적인 범위성이지만 유비쿼터스 환경 등에서 실세계를 모델링하여 온톨로지를 구성한다고 가정할 때, 발생 할 수 있는 매우 많은 양의 Individual을 받아들여야 하므로 결코 간과할 수 없는 조건이 된다. DBMS를 사용하지 않는 Tableau 알고리즘 기반 추론 시스템들에서는 메모리 Heap 크기를 최대 1GB까지 설정을 하여 평가를 수행하였다.

표 3 추론 시스템별 평균 데이터 로드시간 및 질의응답시간

평균 데이터 로드시간(s)		추론 시스템			
Triple 수	iS	KAON2	Pellet	Racer	
82,415	14.52	5.09	4.21	18.70	
189,553	30.94	10.94	9.88	57.69	
278,230	57.28	17.28	86.90	-	
394,584	88.87	28.87	229.96	-	
597,479	151.87	51.87	-	-	
731,041	474.16	736.59	-	-	
828,256	1957.60	1758.60	-	-	
930,343	4283.26	2154.68	-	-	
1,052,895	9245.70	3335.70	-	-	

평균 질의응답시간(s)		추론 시스템			
Triple 수	iS	KAON2	Pellet	Racer	
82,415	0.49	0.28	3.80	2.57	
189,553	0.64	0.58	10.64	5.56	
278,230	0.88	0.95	57.62	-	
394,584	1.26	1.41	160.94	-	
597,479	2.76	2.40	-	-	
731,041	6.98	23.03	-	-	
828,256	17.80	31.61	-	-	
930,343	56.52	50.33	-	-	
1,052,895	99.01	95.62	-	-	

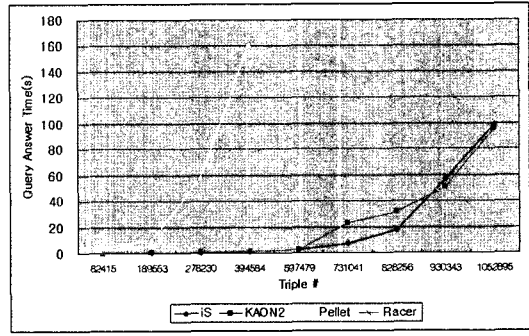
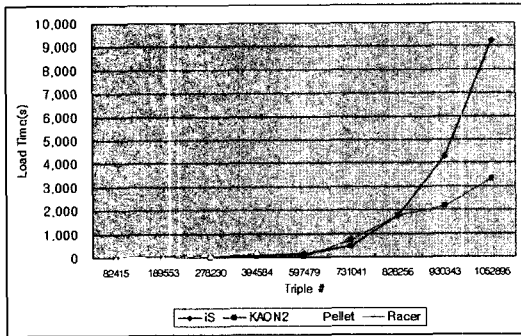


그림 8 추론 시스템 별 평균 데이터 로드시간 및 질의응답시간

그러나 메모리 부족의 발생으로 RacerPro는 189,553개의 트리플을 처리 후 중단되었고, Pellet은 394,594개의 트리플을 처리 후 중단되었다. 이는 Tableau 알고리즘의 수행 방식의 문제점을 나타낸다. 또한 평균 질의 응답시간을 통하여 소량의 ABox 추론에서는 평가에 사용된 모든 추론 시스템이 비슷한 응답 시간을 보였다. 그러나 ABox의 크기가 증가함에 따라 메모리 부족 발생으로 인한 중단과 더불어 질의 응답시간의 곡선은 매우 가파른 기울기를 그리며 급격히 증가 한다. Pellet의 경우 ABox의 크기가 증가함에 따른 질의 응답시간은 지수 곡선 형태의 그래프를 나타내므로 성능 측면에서 매우 좋지 않은 양상을 나타낸다. 이에 반해 Instance Store와 KAON2는 트리플의 증가에 있어 완만한 기울기의 그래프를 그리며 증가하는 비교적 유연한 모습을 나타낸다.

그림 9를 통하여 Tableau 알고리즘 기반의 추론 시스템인 RacerPro와 Pellet은 Heap 메모리의 부족으로 인하여 트리플 입력이 도중에 중단된 것을 알 수 있으며 ABox 추론에 있어 ABox의 크기가 증가함에 따라 질의응답에 대한 소요시간도 매우 많이 소비 되는 것을 알 수 있다.

RacerPro는 트리플 20,000개 이하 영역에서는 동작을 하였고 20,000 이상 40,000 이하의 트리플 수의 영역에서는 메모리 문제를 발생하여 중단되었다. 1개의 대학을 포함하고 있는 트리플 20,000개 영역에서 질의에 대한

응답은 다른 추론 시스템에 비하여 뒤지지 않는 좋은 결과를 보였으며, 특히 LUBM 질의 5에서는 Instance Store보다 좋은 결과를 나타낸다. LUBM 질의 5번은 응답으로서 매우 깊고 넓은 범위의 검색을 요구하는 특징을 가진 질의로써 하나의 클래스 뿐 아닌 그 하위 클래스까지 그 검색 범위를 갖는다. 더불어 프로퍼티에 있어서도 그 하위 프로퍼티를 통한 넓은 범위의 검색을 요구하는 질의이다. 하지만 ABox의 크기가 증가함에 따라 메모리 사용의 유연성을 보이지 못하며 다른 추론 시스템에 비해 빠른 Heap 메모리 부족을 발생시켰다.

Pellet은 RacerPro에 비하여 유연한 메모리 사용으로 LUBM(4, 0) 데이터 셋까지 진행하여 트리플 20,000 이상 40,000 이하 구간까지 동작하였다. 하지만 역시 알고리즘의 문제로 인하여 대용량 ABox 추론에는 부적합한 모습을 나타내었다. LUBM 질의 1, 6, 9에서는 RacerPro에 비하여 좋은 결과를 보였다. LUBM 질의 1번은 대량의 입력 데이터 가운데 높은 선택을 요구하는 질의로 단지 하나의 클래스와 하나의 프로퍼티에 대한 검색 영역을 갖으며, 그 어떠한 추론이나 스키마 구조를 통한 검색이 요구되지 않는다. LUBM 질의 6번은 단지 하나의 클래스만을 검색 대상으로 하지만 추론이 필요한 질의로서 명시적으로 OWL Axiom인 subClassOf로 두 개의 클래스에 대한 검색이 요구되며 암묵적인 두 클래스 사이에서의 검색이 요구된다. 더불어 6번 질의는 대량의 입력 데이터에 대한 낮은 선택이 요구되는 질의이

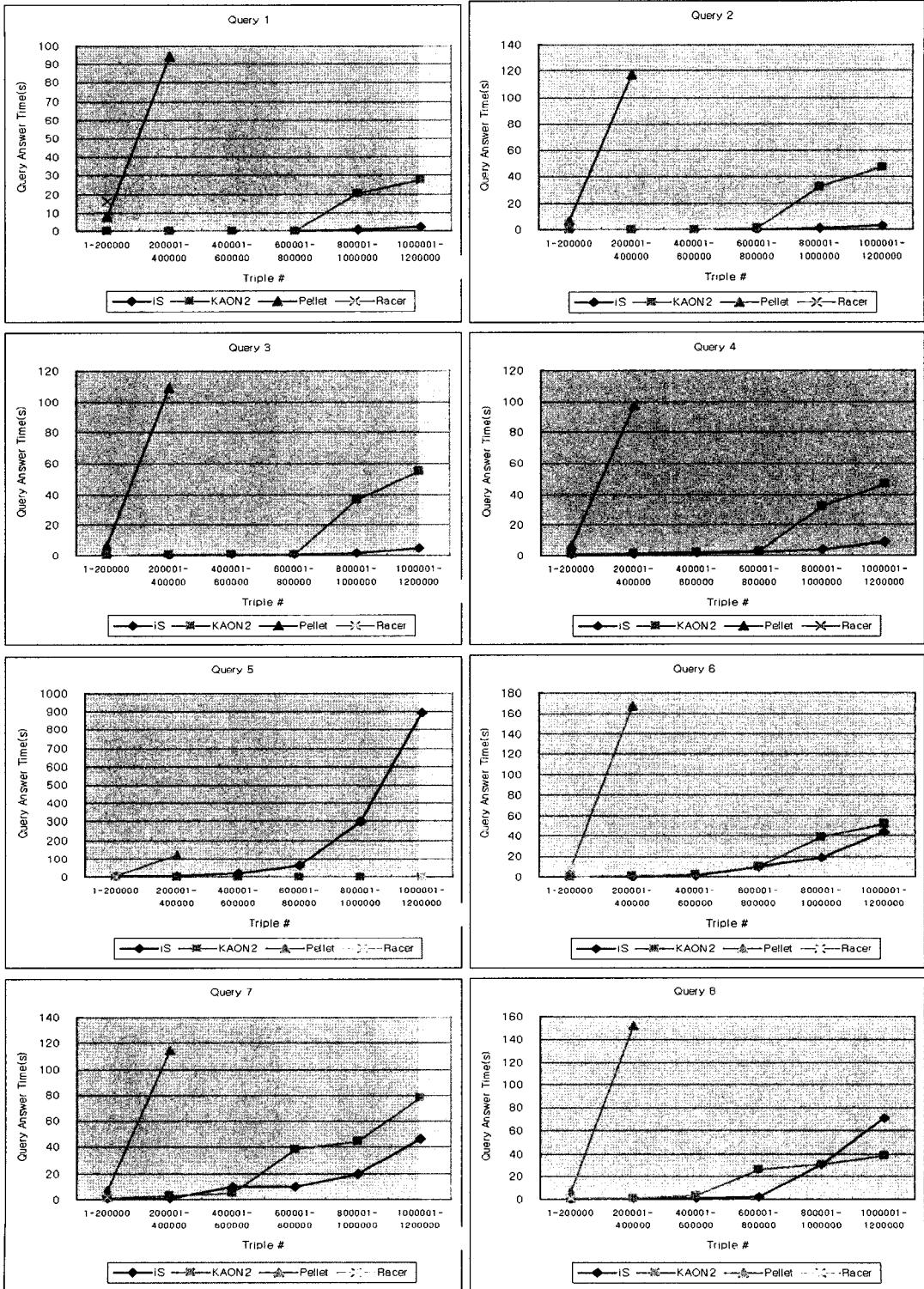


그림 9 각 추론 시스템의 질의 별 평균 응답시간

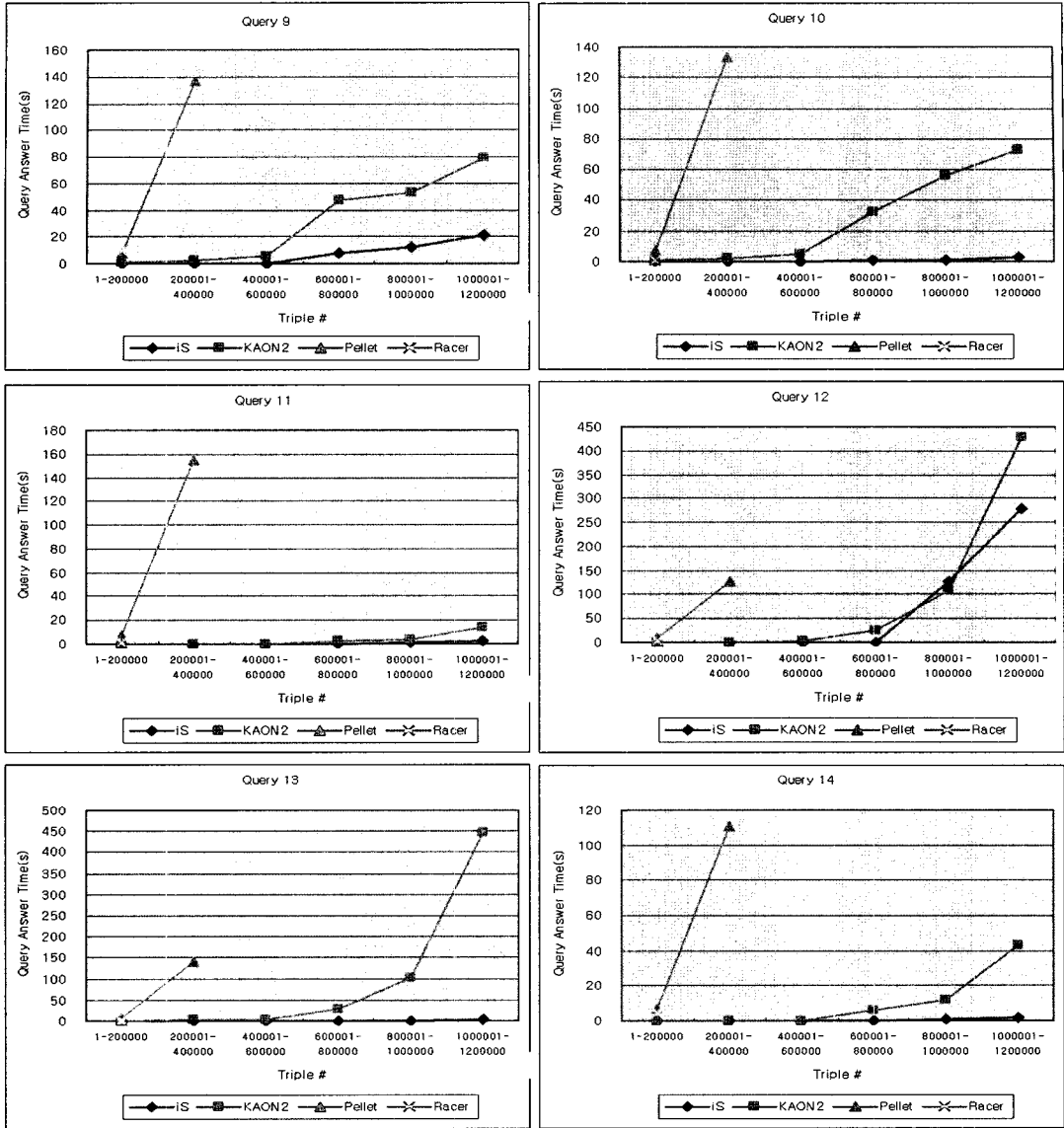


그림 9 각 추론 시스템의 질의 별 평균 응답시간 (계속)

다. LUBM 질의 9번은 트리아앵글 관계패턴을 갖는 질 의로써 넓은 범위의 구조적 검색이 요구되며, 많은 수의 클래스와 프로퍼티가 사용된다.

Instance Store는 DBMS를 함께 사용하여 기존 시스템의 문제점을 보완한 시스템으로 LUBM 질의에 대하여 전체적으로 우수한 결과를 보였다. 다만 질의 5의 경우 질의의 내용이 DBMS에 저장되어있는 서술부와 일치되는 내용이 존재하지 않으므로 Tableaux기반 추론 시스템을 통하여 추론을 시도하고 이에 따라 매우 많은 시간비용을 소비하게 되는 결과를 가져왔다.

KAON2는 ABox의 크기가 증가함에 유연히 대응하

였으며 전체 LUBM 질의에 대하여 고르면서 우수한 결과를 보였다. 특히 LUBM 질의 5, 8에서는 함께 평가한 다른 추론 시스템에 비하여 가장 좋은 결과를 보였다. 단 Instance Store에 비하여 데이터의 증가에 따라 다소 높은 응답시간의 변화를 가져왔다.

각 추론 시스템은 LUBM 질의에 따른 추론 결과 도출에 있어서 그림 10과 같이 “올바른 결과 수”를 “도출된 결과 수”로 나누어 질의에 대한 결과 완성도로서 표현하는데, 이는 각 추론 시스템의 결과가 얼마나 높은 정확도를 가지는가를 평가하기 위한 기준이 된다. 그림 10의 결과를 통하여 Instance Store, KAON2, Pellet의

System\Query	1	2	3	4	5	6	7	8	9	10	11	12	13	14
InstanceStore	1	1	1	1	n/a	1	1	1	1	1	1	1	1	1
KAON2	1	1	1	1	n/a	1	1	1	1	1	1	1	1	1
RacerPro	1	1	1	n/a	1	0.76	1	n/a	1	1	1	1	1	1
Pellet	1	1	1	1	n/a	1	1	1	1	1	1	1	1	1

그림 10 각 추론 시스템의 질의 별 결과 완성도

정확도가 RacerPro에 비하여 다소 높음을 알 수 있다. 실험을 정리하면 다음과 같다. Tableaux 알고리즘은 TBox 추론을 목적으로 설계되고 개발된 알고리즘으로 TBox 추론에 대해 매우 빠르고 효율적이다. 그러나 ABox에 대해서는 추론은 가능하지만 규모가 커질 경우 ABox의 각 Individual을 모두 탐색 영역에 대입하여 탐색 공간이 크게 확장되어 탐색 트리의 규모가 매우 커지는 문제점이 있다. 이에 반하여 Disjunctive Datalog Approach는 Resolution 프레임워크를 사용하며, Deductive Database 기법을 적용하여 OnePass Bottom-up 방식으로 Query answer를 미리 계산하여 빠른 응답을 제공한다. 그러나 상대적으로 TBox 추론에 있어서 기존의 Tableaux 알고리즘 기반 추론 시스템이 Disjunctive Datalog Approach를 사용한 KAON2에 비하여 더 좋은 결과를 나타낸다. 이는 Tableaux 알고리즘과 Disjunctive Datalog Approach가 각기 다른 목적으로 최적화 되어져 있기 때문이다. Tableaux 알고리즘은 제약이 없는 일반적 TBox를 기준으로 서술논리 표현력 중 ALC의 처리부터 시작 이후 서술논리의 표현력을 더욱 폭넓게 처리할 수 있도록 연구가 진행되어 ALC와 더불어 Inverse role, qualifying number restriction 등을 처리할 수 있도록 확장되었다. 또한 ABox 일관성 검사 등으로 처리 범위가 확장되었다. 그리고 Tableaux 알고리즘이 발표된 이후 10여 년간 지속적으로 TBox 추론에 있어서 발생할 수 있는 순환을 찾아 블록 하는 기법 등 TBox에 최적화 알고리즘들이 발표되었다. Disjunctive Datalog Approach의 KAON2는 대용량 ABox 추론을 위하여 연구된 것으로 TBox 추론에 비하여 ABox 추론에 더욱 최적화된 시스템이다. 추론 데이터베이스 기술에 질의응답을 위한 최적화 기법을 적용하면 높은 성능의 추론이 가능할 것이라는 가정을 기반으로 대용량 ABox를 처리하기 위해 연구 되어 현재 qualifying number restriction에 있어 제한적인 SHIQ-의 표현을 처리한다. 그리고 Tableaux 알고리즘 기반 추론 시스템과 DBMS를 연동한 Instance Store는 DBMS의 적절한 사용으로 빠른 응답을 제공한다. 그러나 질의를 위한 Abstract OWL Description이 DBMS의 Description 테이블의 Description과 매치되지 않을

경우 바로 결과를 반환하지 못한다. 이를 해결하기 위하여 TBox 추론 시스템에 의존하여 추론을 수행하는 문제점을 가진다. TBox 추론 시스템에 대한 의존이라는 것은 연동되어 사용되는 추론 시스템을 통하여 TBox를 처리하여 데이터베이스를 구성하고 ABox는 ID와 서술부를 구분하여 데이터베이스에 저장하여 질의 서술에 대한 데이터베이스 내의 서술과 직접 비교를 통하여 결과를 반환한다. 이때 정확히 일치 되는 서술이 존재하지 않는다면 Instance Store는 그 질의를 연동되어있는 추론 시스템으로 DIG Interface를 통하여 전달한다. 그러면 추론 시스템은 그 질의에 대한 추론 프로세스를 진행하는데 이때의 질의가 ABox 추론에 해당하는 질의일 때, Tableaux 알고리즘의 특성상 매우 넓은 탐색영역을 만들게 되어 많은 응답시간 비용을 필요하게 된다.

6. 결론 및 향후 연구

본 논문은 대용량 ABox 추론에 있어 다양한 환경에 따른 적합한 추론 시스템 제시를 목적으로 시스템의 범용성을 기준으로 평가를 수행한다. 따라서 지금까지의 전형적 추론 시스템에서 사용하고 있는 Tableaux 알고리즘의 이론적 내용과 그의 문제점 그리고 그 문제점을 해결하기 위한 방안으로 연구된 Instance Store와 KAON2를 비교 평가 하였다.

지금까지의 평가 연구에서는 Tableaux 기반 시스템 간의 TBox 추론에 대한 평가가 대부분이었으며, 자체 시스템을 평가하기 위한 목적으로 평가가 수행되었다. 그러나 본 논문에서는 대용량 ABox 추론을 위하여 연구된 시스템의 평가를 목적으로 하며, 각 시스템에서 사용되는 알고리즘의 특성 및 장/단점을 비교하였다. 이를 통하여 각 환경(도메인)이 가지는 조건에서 추론 시스템의 적합성을 제시하고자 하였다.

실험 결과를 통해 도메인 환경을 온톨로지 모델링 했을 때, TBox의 규모는 작으나 대량의 ABox가 구성 되어질 경우 KAON2의 사용이 적합하며, 대용량 ABox 추론과 더불어 TBox 추론을 병행해야 하는 경우 Instance Store 사용이 적합하다는 것을 알 수 있다. 이 결론은 실험 결과 중 표 3의 내용을 통해 간단히 알 수 있다. Individual의 증가를 통한 트리플의 증가에 따라

Tableaux 기반 시스템의 응답 속도가 현저히 떨어지는 것을 알 수 있다. 더불어 질의에 대한 결과의 완성도를 고려해야 하는데 이는 그림 10에서 나타나듯 평가된 각 시스템은 각기 유사한 결과 완성도를 가진다. 따라서 일반적인 환경에서 결과 완성도를 통해 어떠한 시스템을 사용할 것인지의 평가는 큰 의미가 없다.

Instance Store와 KAON2, 두 시스템은 초기 데이터 로드시간이 많이 소요되는 문제점을 가진다. Instance Store는 초기에 연동된 추론 시스템을 통해 TBox를 구성한 후, ABox가 로드될 때 Individual ID와 서술부를 분리하여 데이터베이스에 입력한다. 이때 많은 데이터베이스 처리 시간이 필요하다. 또한 KAON2는 온톨로지를 로드하여 서술논리기반의 지식베이스를 선언적 데이터로그(Disjunctive Datalog) 지식베이스로 형태를 변환할 때 많은 시간이 필요하다. 이는 Resolution 과정과 Function symbol을 제거하는 과정을 거치면서 지식베이스의 형태를 변환하는데, 이 과정은 결과적으로 질의에 대한 응답을 위하여 미리 결과 세트를 구성하기 때문이다. 이는 평가 결과인 표 3과 그림 8을 통하여 확인할 수 있다. 이것은 추론 환경이 초기 온톨로지를 구성하여 이후 온톨로지의 변화가 거의 없으며, 지속적인 질의응답을 요구하는 환경에 적합하다는 것을 의미한다. 즉, 초기 온톨로지가 지속적으로 업데이트 되며, 로드한 온톨로지에 대하여 지속적인 업데이트가 이루어지는 환경의 경우 적용이 어렵다. 따라서 대용량 ABox 추론환경을 위한 추론 시스템에서는 초기 로드시간을 줄여야 하며, 현재 Instance Store와 KAON2 모두 Nominal 처리가 불가능하고 Number restriction 역시 제한적으로 처리가 가능한데 이러한 서술논리기반의 표현력에 대한 처리 능력은 증가시켜야 한다.

본 실험을 통해 나타난 바와 같이 대용량 ABox 추론에 있어서 Tableau 알고리즘만을 사용하는 것은 어렵다. 따라서 DBMS의 적용에 대한 효율적인 방법론 등의 대체 방법 또는 최적화 알고리즘이 필요하며, 추론 데이터베이스 기술을 응용한 새로운 기법들이 요구된다. 향후 연구로는 여러 추론 시스템들에서 보여준 문제점들을 기반으로 KAON2 기반 최적화 기법에 대하여 연구를 진행하며, Nominal Value의 처리를 위한 연구와 더불어 서술논리의 표현력 처리에 대한 연구를 진행한다. 이를 통하여 대용량 ABox 처리를 위한 보다 최적화된 추론 방법론을 제시하고자 한다.

참 고 문 헌

- [1] Genesereth, M., Nilsson, N. Logical Foundations of Artificial Intelligence, Morgan-Kaufman, 1987.
[2] D.L. McGuinness, F.V. Harmelen : OWL Web Onto-

logy Language Overview. W3C recommendation 10 Feb, 2004.

- [3] F. Baader, W. Nutt : Basic Description Logics. The Description Logic Handbook. Cambridge University Press, 2003.
[4] I. Horrocks, U. Sattler and S. Tobies : Reasoning with individuals for description logic SHIQ. Proc. of the 17th Int. Conf. On Automated Deduction(CADE 2000): 482-496.
[5] J. Galinski, A. Kaya, and R. Möller : Development of a server to support the formal semantic web query language OWL-QL. In I. Horrocks, U. Sattler, and F. Wolter, editors, Proc. International Workshop on Description Logics, 2005.
[6] I. Horrocks, L. Li, D. Turi : The Instance Store: Description Logic Reasoning with Large Numbers of Individuals. DL2004.
[7] B. Motik, Ulrike Sattler : Practical DL Reasoning over Large ABoxes with KAON2.
[8] Y. Guo, Z. Pan, and J. Heflin : LUBM: Benchmark for OWL Knowledge Base Systems. Lehigh University, ISWC2004.
[9] Y. Guo, Z. Pan, and J. Heflin : Choosing the Best Knowledge Base System for Large Semantic Web Applications. In Proc. of the 13th International World Wide Web Conference(WWW2004) - Alternate Track Papers & Posters, 2004.
[10] Y. Guo, Z. Pan, and J. Heflin : An Evaluation of Knowledge Base Systems for Large OWL Datasets. In Proc. of the 3rd International Semantic Web Conference (ISWC2004), 2004.
[11] I. Horrocks and P. Patel-Schneider : DL Systems Comparison. In Proc. of DL'98, 1998.
[12] RacerPro Homepage, <http://www.racer-systems.com/>
[13] Pellet Homepage, <http://pellet.owldl.com/>
[14] SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>
[15] VICODI, <http://www.vicodi.org/>
[16] I. Horrocks : Optimising Tableau Decision Procedures for Description Logics. PhD thesis, University of Manchester, 1997.
[17] B. Motik and U. Sattler. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. Proc. of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2006), Phnom Penh, Cambodia, November, 2006.

서 은 석

정보과학회논문지 : 소프트웨어 및 응용
제 34 권 제 6 호 참조

박 영 택

정보과학회논문지 : 소프트웨어 및 응용
제 34 권 제 6 호 참조