

OWL 기반 대용량 ABox 추론 시스템 평가

서은석^o 박영택

송실대학교 컴퓨터학과

erin214@ailab.ssu.ac.kr^o, park@computing.ssu.ac.kr

A Benchmarking of Large ABox Reasoners Based OWL

Eun-Seok Seo^o Young-Tack Park

Department of Computing, Soongsil University

요 약

시맨틱 웹에 대한 관심이 높아짐과 더불어 관련 기술인 온톨로지와 이를 이용한 추론 기술 역시 이슈가 되고 있다. RacerPro, Pellet 등 지금까지의 전형적인 추론 시스템들은 주로 Tableau Algorithm 기반의 추론 시스템으로 Tableau Algorithm의 특성상 대용량 ABox 추론에서 문제점을 나타낸다. 이를 해결하기 위한 연구로는 Tableau Algorithm 기반의 Instance Store와 Disjunctive Datalog Approach를 사용한 KAON2가 있다. 이러한 추론 기술에 대해서는 많은 연구가 진행되고 있지만 각 추론 시스템들에 대한 평가는 부족하다. 현재 추론 시스템들의 벤치마킹은 대부분 Tableau Algorithm 기반의 TBox 추론에 대한 것으로 ABox 추론 및 최근 이슈인 대용량 ABox 추론에 대한 평가는 특히 부족하다. 이에 본 논문에서는 각 추론 시스템들의 이론적 배경을 근간으로 지금까지의 전형적 추론 엔진들과 최근 이슈에 따른 대용량 ABox 추론을 위한 시스템들을 이론적 비교를 통해 살펴본다. 특히 대용량 ABox 추론을 위한 시스템인 Instance Store와 KAON2를 LUBM을 사용하여 평가함으로써 대용량 ABox 추론에 있어 사용자의 요구에 따른 적절한 시스템을 제시한다.

1. 서 론

최근 사람들은 사이버 공간이 아닌 실제 물리적 세계에서 서비스를 제공하는 유비쿼터스 환경과 이에 접목된 시맨틱 기술을 통하여 보다 동적이고 상호 운용적인 서비스를 제공받는다. 이러한 서비스를 제공받기 위해서는 기계가 정보자원의 의미를 정확히 인식해야하며 이를 기반으로 논리적 추론이 가능해야 한다. 이에 핵심적인 기술이 온톨로지[1]와 추론[2] 기술이다.

지금까지의 추론 시스템들은 주로 Tableau Algorithm[3] 기반의 Terminological(TBox) 추론 시스템들이었으나 Individual(ABox) 추론의 문제점을 개선하고자 최근 ABox 추론을 위한 시스템, 특히 대량의 ABox 추론을 위한 시스템에 관한 연구가 활발히 진행되고 있다. 실제 물리적 세계를 모델링한 온톨로지는 Taxonomy를 기반으로 실제 세계의 개체들을 Individuals로 포함한다. 이를 대상으로 한 추론에서는 TBox 추론 시스템과 더불어 ABox 추론이 더욱 강조되며, 특히 대량의 ABox 추론의 중요성이 강조된다. 즉 실제 물리적 세계를 기반으로 온톨로지를 구성할 경우 매우 많은 양의 Individuals이 생성되어지며 이를 위한 추론 시스템은 대량의 Individuals을 처리할 수 있어야 한다.

따라서 추론 시스템의 평가는 여러 가지 요구 사항들에 적합하도록 이루어져야 하는데 이는 매우 어렵다. 특히 범위유성(Scalability)의 조건에 대하여는 더욱 어렵다. 주된 이유는 일부 시맨틱 웹 데이터 셋의 크기가 매우 크며 사용되는 온톨로지(semantically rich ontology)의 표현범위 또한 크기 때문이다.

이에 본 논문에서는 LUBM을 기반으로 온톨로지의

TBox와 ABox를 구성하고 각 추론 시스템의 평가 표준과 온톨로지 표현의 적절히 범위를 규정하며, 다양한 양의 인스턴스와 그들의 관계를 구성하여 트리플(Triple)에 따른 추론 시스템의 양적, 의미적 부분의 성능을 평가한다. 지금까지의 추론 시스템 가운데 가장 많이 사용되며 높은 성능을 나타내는 Tableau 알고리즘 기반의 RacerPro 등을 비롯한 여러 추론 시스템을 비교하여 실제 물리적 세계의 대량의 ABox를 처리하기에 가장 적절한 추론 시스템을 제시한다.

2. LUBM(Lehigh University BenchMark)

LUBM(Lehigh University BenchMark)은 다양한 추론 시스템의 평가를 위한 방법을 제시한 것으로 추론 시스템의 평가에 있어 널리 사용되어 표준화된 평가 방법론이다. LUBM은 대학 도메인의 스키마 온톨로지와 확장성 평가를 위한 임의의 크기의 OWL 데이터를 생성할 수 있는 인스턴스 생성기를 제공한다. 또한 다양한 프로퍼티를 포함한 14개의 확장된 질의 패턴을 제공하며 질의 언어로는 SparQL을 사용한다.

LUBM의 Benchmark 온톨로지는 Unive-Bench.owl로 OWL의 서브셋 중 가장 단순한 OWL-lite를 사용하여 총 43개의 Class와 32개의 프로퍼티(25개의 오브젝트 프로퍼티와 7개의 데이터타입 프로퍼티)로 구성된다.

LUBM의 구조는 각 타겟 추론 시스템에 객체화된 인터페이스를 만들고 Benchmark 테스트 모듈은 이 인터페이스를 통하여 각 추론 시스템의 동작을 제어한다.

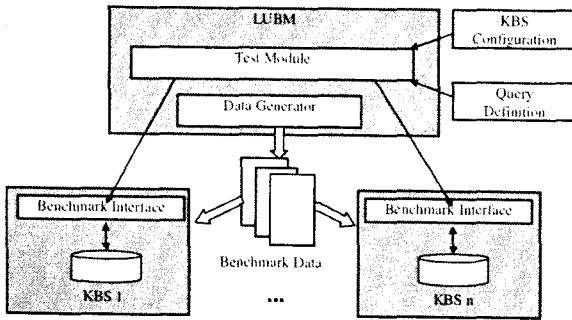


그림 1 LUBM 구조도

테스트 모듈은 각 추론 시스템 각각의 설정 내용과 질의 형태를 포함한다. 각 추론 시스템별 설정 사항들과 질의 형태를 각 시스템에 맞도록 미리 구성함으로써 평가에서 정확성과 객관성을 유지 할 수 있다.

3. DL과 추론 시스템의 기본구조

온톨로지 기반 추론에서 온톨로지는 (Description Logic)[5] 기반으로 작성된다. 서술논리에 대한 연구는 1980년대부터 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 스키마에 대한 Assertion 데이터를 표현한다. TBox와 ABox에서 표현되는 지식은 다음의 [표 1]과 같다.

표 1 DL 지식 베이스의 TBox와 ABox

TBox :	ABox :
Mother = Woman \sqcap hasChild.Person	Father(Peter)
Parent = Mother \sqcup Father	Grandmother(Mary)
Grandmother = Mother \sqcap hasChild.Mother	hasChild(Mary, Peter)
	hasChild(Mary, Paul)
	hasChild(Peter, Harry)

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

3.1 Tableau 알고리즘

Tableau 알고리즘의 기본 아이디어는 증명하고자 하는 어떤 것에 대하여 그에 대한 부정을 다양한 규칙을 적용하여 un-Satisfiable함을 증명하는 방식을 취한다. 일반적으로 지식 K에 대하여 포함관계를 표현할 때 C D으로 표현하는데 이를 Tableau 알고리즘에서는 $\neg C \sqcup D$ 가 un-Satisfiable함을 증명함으로 C D이 Satisfiable함을 증명하게 된다. 다음 [그림 2]는 Tableau 알고리

즘이 트리구조로 탐색영역을 확장해가며 탐색영역을 모두 검색하여 증명하고자 하는 것의 부정이 un-Satisfiable함을 보임으로 포함관계를 추론하게 된다.

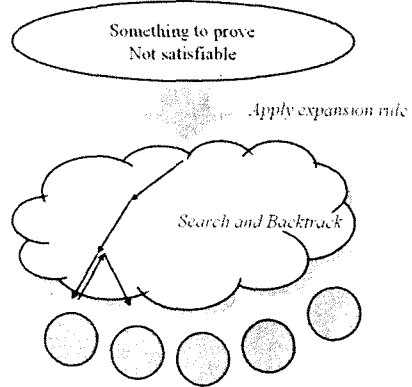


그림 2 Tableau 알고리즘의 아이디어

Tableau 알고리즘의 탐색영역은 Expansion 규칙이 적용되어 트리구조로 확장하게 된다. 그리고 모든 탐색 공간의 종단 노드가 모순(Contradiction)임을 보임으로 증명하고자 하는 것의 부정이 un-Satisfiable함을 증명하게 된다. 또한 증명을 위한 탐색영역 확장에서 트리구조의 종단 노드가 존재하지 않게 되는 문제를 Non-termination 문제라 하는데 이를 해결하기 위한 방법으로 Blocking 기법을 사용한다.[3]

Tableau 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 Individuals를 대입하여 탐색영역을 확장함으로써 트리의 규모가 매우 커지며 처리에 대하여 많은 소요비용이 발생하게 된다. 즉, 방대한 트리구조의 Forest를 이룸으로써 효율성이 떨어지게 된다.

3.2 Instance Store

Tableau 알고리즘이 대용량 ABox 추론을 위해 탐색영역을 확장하는 과정에서 Forest를 이루는 결과를 해결하기 위한 연구 중 하나가 Instance Store[6]이다.

Instance Store는 대용량 ABox 추론을 위하여 영국 Manchester 대학에서 연구한 추론 시스템으로 자체적인 추론 알고리즘을 가지지 않고 Tableau 알고리즘 기반 추론 시스템인 RacerPro, FaCT등을 DIG Interface를 통해 연동하여 사용한다. Instance Store는 추론 시스템을 통하여 온톨로지 클래스 레벨의 스키마를 정의하고 인스턴스는 관계형 데이터베이스 시스템을 통하여 데이터베이스에 정의한다. 즉 기존 추론 시스템과 관계형 데이터베이스 시스템을 함께 사용함으로써 대용량 ABox 추론의 어려움을 해결한다. Instance Store의 구조는 [그림 3]과 같다.

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

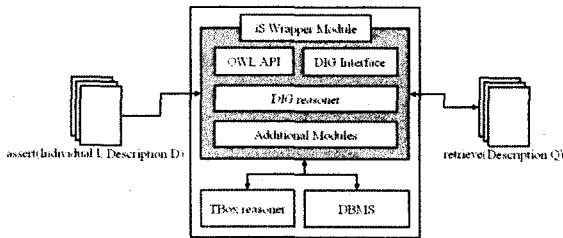


그림 3 Instance Store 구조도

Instance Store는 관계형 데이터베이스 시스템의 Description 테이블을 중심으로 질의에 응답한다. Description 테이블에는 인스턴스의 입력 과정에서 주어지는 인스턴스의 이름과 서술부를 매핑하기 위한 Description ID와 Description이 저장된다. Instance Store의 질의 형태는 aOWL(Abstract OWL)이며 질의가 주어질 경우 질의를 Description 테이블의 서술부와 비교하여 동일한 서술부 내용이 있을 때 이에 매핑되는 인스턴스의 이름을 반환한다. 이때 추론 시스템은 필요한 때에만 사용되며 추론 시스템을 통하여 추론된 스키마 정보를 미리 데이터베이스에 캐쉬하여 빠른 추론을 지원한다. 하지만 Description 테이블에 일치하는 결과가 없을 때 Instance Store는 최악의 경우로 TBox 추론 시스템을 통하여 추론을 수행한다. 이 경우 앞서 설명한 Tableau 알고리즘의 문제가 발생한다.

3.3 KAON2(Disjunctive Datalog Approach)

대용량 ABox 추론을 위해 연구된 독일 Karlsruhe대학의 KAON2[7]는 Tableau 알고리즘이 아닌 Disjunctive Datalog Approach를 사용한다. 선언적 데이터 로그(Disjunctive Datalog) Approach는 온톨로지를 자체 Reduction 알고리즘을 통하여 선언적 데이터 로그 형태의 Clause로 형 변환한다. KAON2의 Reduction 알고리즘의 내부 단계는 [그림 4]과 같다.

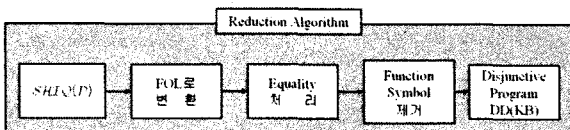


그림 4 KAON2의 Reduction 알고리즘

온톨로지는 Mapping Axiom을 사용하여 FOL로 변환

하고 이때 발생 할 수 있는 Equality의 처리를 위하여 Resolution을 확장한 Basic Superposition을 사용한다. 그리고 변환 과정에서 존재 한정자(Exist Quantifier)를 통해 발생할 수 있는 Function Symbol을 제거함으로써 최종적으로 선언적 데이터 로그로 변환된다.

지식베이스에 존재하는 지식들에 대하여 형 변환 후 각 지식들에 의해 도출될 수 있는 모든 지식들을 도출한다. 이렇게 도출된 결과는 처음 지식베이스의 지식들과 의미적으로 동일하다. 이를 매직셋이라 하며 도출된 결과는 처음 지식베이스의 의미와 동일성을 유지함으로써 추론에 그대로 적용할 수 있다.

선언적 데이터 로그 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 선언적 데이터 로그 방식은 효율적 최적화 방법을 적용하여 더욱 향상된 응답성을 지원한다..[8]

4. LUBM 평가 방법에 따른 실험 및 결과

LUBM[4] 평가 방법을 사용하여 대용량 ABox 추론 시스템인 Instance Store와 KAON2를 기준으로 전형적 추론 시스템 2개와 각각 비교를 통해 평가하였다.[9]

4.1 평가 대상 추론 시스템

실험에서는 각 추론 시스템에 대하여 확장성에 대한 평가와 온톨로지의 규모가 커짐에 따른 추론 성능에 대한 평가를 진행한다. 추론 시스템은 Tableau 알고리즘을 기반으로 현재 가장 많이 사용되는 추론 시스템과 서술논리 기반 추론에서 가장 높은 표현을 지원하는 추론 시스템을 선택하였고 Tableau 알고리즘 기반 추론 시스템들에서 나타난 대용량 ABox 추론의 문제의 대안 시스템으로 Tableau 알고리즘 기반 시스템과 non-Tableau 알고리즘 기반 시스템을 선택하였다.

첫 번째 시스템으로 RacerPro[10]는 서술논리 기반 추론 엔진으로 현재 많은 곳에서 활용되고 있으며 현재 RDF, DAML+OIL 그리고 OWL까지 지원한다. 구동에 있어서는 Server와 Client로 구분되며, Server는 Client와 상호 DIG 인터페이스를 통하여 HTTP 또는 TCP 프로토콜 통신으로 연결된다. RacerPro는 질의 언어로 nRQL을 사용하며 이를 통하여 LUBM 14개 질의를 모두 표현할 수 있다.

두 번째 시스템으로 Pellet[11]은 Tableau 알고리즘 기반의 추론 시스템 중 가장 높은 서술논리 표현(SHOIN)을 지원한다. 시스템의 핵심은 Tableau 추론기(지식 베이스의 일관성 검사)이며 이는 XML Schema 단순 데이터타입의 논리곱 일관성(Conjunction consistency)을 검사할 수 있는 데이터타입 오라클(Datatype oracle)과 함께 말할 수 있다. 질의 언어로는 SparQL을 사용한다.

세 번째 시스템으로 Instance Store는 Tableau 알고

리즘을 사용하는 추론 시스템인 FaCT, FaCT++, RacerPro등과 DIG 인터페이스를 사용하여 연동하여 동작한다[6]. 즉 기존의 Tableau 알고리즘 기반 추론 시스템에서 나타난 대용량 ABox 문제점을 그대로 계승하게 되는데 이를 해결하기 위하여 MySQL, Oracle, 그리고 순수 Java 기반의 HyperSonic 등의 DBMS를 함께 사용한다. 사용되는 온톨로지는 스키마와 인스턴스로 구분하여 입력되는데 인스턴스는 aOWL의 형태로 입력된다. 질의 언어는 aOWL을 사용한다.

네 번째 시스템으로 KAON2는 온톨로지 관리 하부구조(Infrastructure)로 불리는 시스템이다. 이는 Tableau 알고리즘을 사용하지 않고 선언적 데이터 로그 언어를 통하여 DBMS에서 추론을 모두 처리할 수 있도록 설계된 추론 시스템이다[7]. KAON2는 질의 언어로 SparQL을 사용한다.

4.2 실험 환경 구성

실험환경은 LUBM에서 제안하는 평가 방법에 의하여 구성되었으며 추론 시스템은 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 데이터 셋은 LUBM에서 기본적으로 제공하는 스키마 온톨로지 및 인스턴스 생성기를 통한 온톨로지 인스턴스의 조합을 사용한다. 데이터 셋은 각기 1~10개의 대학을 포함하며, 각 대학은 15~25개의 파트로 구성된다. 총 Triple의 개수는 1,052,895개이며 처음은 20,659개의 Individual에 대한 82,415개의 Triple로 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.

4.3 실험 결과

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

표 2 추론 시스템별 평균 데이터 로드시간

평균 데이터 로드시간(s)	추론 시스템				
	Triple 수	IS	KAON2	Pellet	Racer
82,415	14.82	5.05	4.21	13.77	-
189,552	30.94	10.94	9.85	57.89	-
279,230	57.26	17.26	59.30	-	-
394,594	88.87	23.57	229.70	-	-
597,479	151.87	51.87	-	-	-
791,041	474.13	796.59	-	-	-
829,256	1957.80	1759.60	-	-	-
930,543	4133.20	3192.60	-	-	-
1,052,895	9245.70	2335.70	-	-	-

표 3 추론 시스템별 평균 질의응답시간

평균 질의응답시간(s)	추론 시스템				
	Triple 수	IS	KAON2	Pellet	Racer
52.415	0.29	0.26	0.69	0.77	-
189,552	0.64	0.52	10.64	5.76	-
279,230	0.63	0.95	57.62	-	-
394,594	1.26	1.41	160.94	-	-
597,479	2.76	1.41	-	-	-
791,041	0.96	23.03	-	-	-
829,256	17.60	31.61	-	-	-
930,543	56.52	86.32	-	-	-
1,052,895	99.61	85.62	-	-	-

[표 2][표 3]은 평가에 사용된 모든 추론 시스템들의 평균 데이터 로드시간 및 평균 질의 응답시간을 나타내며 이에 대해 Triple의 수가 증가함에 따른 시간 증가를 [그림 5]에서 나타낸다. 평가 결과를 통해 추론 시스템의 확장성의 중요성이 부각된다. DBMS를 사용하지 않는 Tableau 알고리즘 기반 추론 시스템에서는 메모리 Heap 크기를 최대 1GB까지 설정을 하여 평가를 수행하였으나 메모리 문제로 RacerPro는 189,553개의 Triple을 처리 후 중단되었으며 Pellet은 394,594개의 Triple을 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 보였으나 ABox의 크기가 증가함에 따라 메모리 문제의 발생으로 인한 중단과 더불어 질의 응답시간의 급격한 증가를 확인하였다. Pellet의 경우 ABox의 크기가 증가함에 따른 질의 응답시간은 지수 곡선의 형태를 나타내므로 성능 측면에서 매우 좋지 않은 양상을 나타낸다. 이에 반해 Instance Store와 KAON2는 Triple의 증가에 있어 비교적 유연한 모습을 나타낸다.

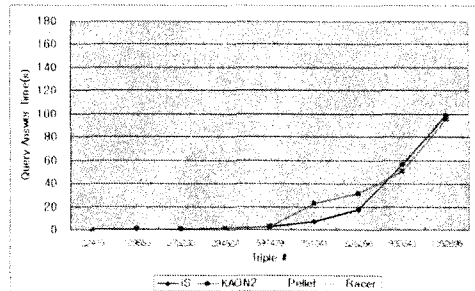
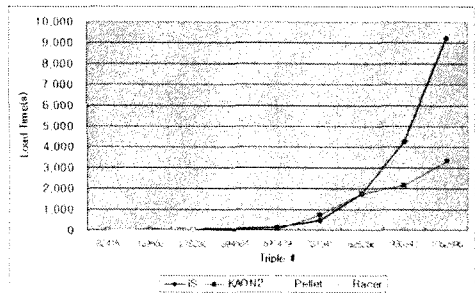


그림 5 추론 시스템 별 평균 데이터 로드시간 및 질의응답시간

[표 2]와[표 3] 그리고 [그림 5]에서와 마찬가지로 [그림 6]에서도 Tableaux 알고리즘 기반의 추론 시스템인 RacerPro와 Pellet은 Heap 메모리의 부족으로 인하여 중단된 것을 알 수 있으며 ABox 추론에 있어 ABox의 크기가 증가함에 따라 질의 응답에 대한 소요시간도 매우 많이 소비 되는 것을 알 수 있다.

RacerPro는 Triple 20,000개 이하 영역에서만 동작을 하였고 20,000 이상 40,000 이하의 Triple 수의 영역에서는 메모리 문제를 발생하여 중단되었다. 1개의 대학을 포함하고 있는 Triple 20,000개 영역에서 질의에 대한 응답은 다른 추론 시스템에 비하여 뒤지지 않는 좋은 결과를 보였으며, 특히 LUBM 질의 5에서는 Instance Store보다 좋은 결과를 나타낸다. 하지만 Tableaux 알고리즘 기반의 추론 시스템으로써 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 Triple 20,000 이상 40,000 이하 구간까지 동작하였다. 하지만 역시 Tableaux 알고리즘을 사용하는 추론 시스템으로서 대용량 ABox 추론에는 부적합한 모습을 나타내었다. LUBM 질의 1, 6, 9에서는 RacerPro에 비하여 좋은 결과를 보였다.

Instance Store는 Tableaux 알고리즘 기반 추론 시스템과 DBMS를 함께 사용하여 Tableaux 알고리즘 기반 추론 시스템에서의 문제점을 개선한 시스템으로 전체 질의에 대하여 우수한 결과를 보였으나 LUBM 질의 5의 경우 질의 Description과 일치되는 내용이 DBMS 내에 존재하지 않으므로 Tableaux 기반 추론 시스템을 통하여 추론을 시도하고 이에 따라 매우 많은 시간비용을 소비하게 되는 결과를 가져왔다.

KAON2는 ABox의 크기가 증가함에 유연히 대응하였으며 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.

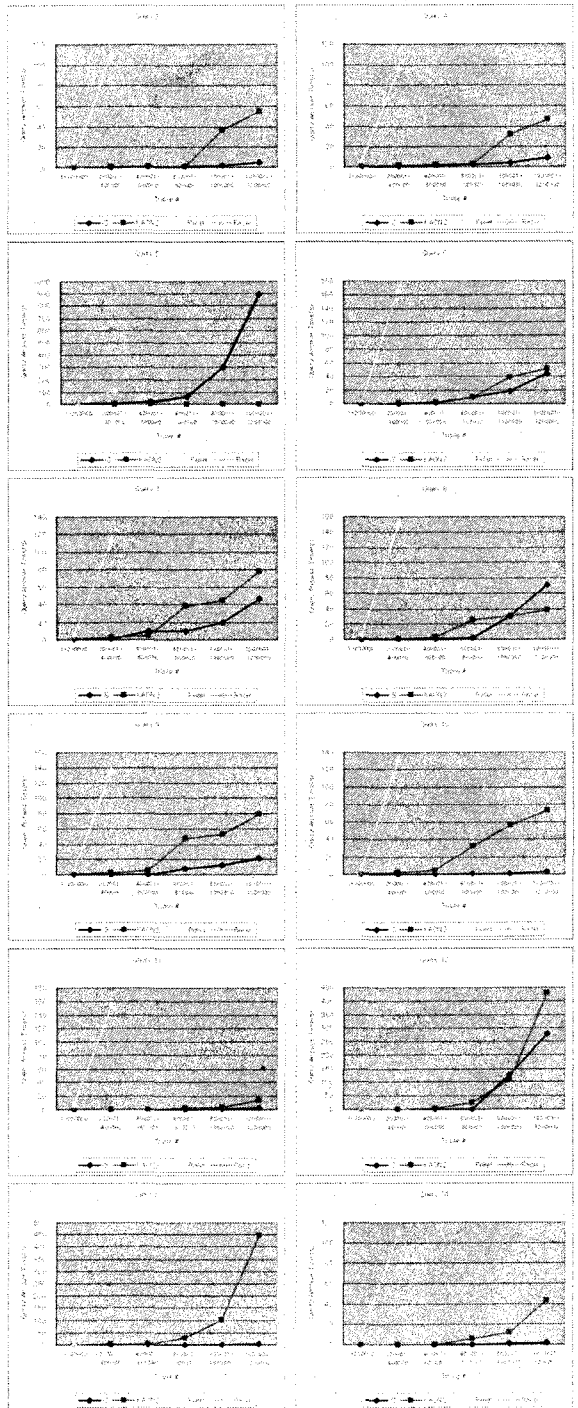
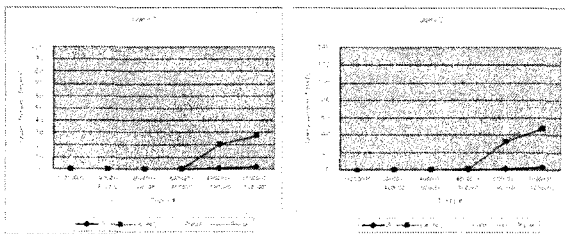


그림 6 각 추론 시스템의 질의 별 평균 응답시간

5. 결론 및 향후 연구

Tableaux 알고리즘은 추론에 있어 매우 좋은 알고리즘이다. 하지만 이는 TBox 추론에 한하며 ABox 추론에서는 탐색영역 확장에 대한 탐색트리가 Forest화되는 문제점이 있다. 이에 KAON2의 선언적 데이터 로그 Approach는 Resolution 프레임워크를 사용하며 OnePass Bottom-up 방식으로 Query answer를 미리 계산하여 동으로써 빠른 응답을 제공한다. 그러나 TBox 추론에서 TBox의 규모가 커졌을 경우 더딘 결과 반환의 문제점이 발생한다. 그리고 기존의 Tableaux 알고리즘을 사용한 추론 시스템을 사용하지만 DBMS를 함께 결합하여 기존 Tableaux 기반 추론 시스템의 문제점을 해결한 Instance Store는 DBMS의 적절한 사용으로 빠른 응답을 제공하나 질의의 Description이 DBMS의 Description 테이블의 Description과 매치되지 않을 경우 최악의 경우로 TBox 추론 시스템에 의존하여 추론을 수행하는 문제점을 가지고 있다.

대용량 ABox 추론에 있어서는 기존의 Tableaux 알고리즘 기반의 추론 시스템이 아닌 대용량 ABox 추론을 위하여 연구된 KAON2와 Instance Store가 더욱 최적화되어 있으나, KAON2는 TBox 추론이 병행 되었을 경우에는 적합하지 못하며, Instance Store는 대용량 ABox 추론을 위한 온톨로지 Assertion Time이 매우 많이 소비되지만 이후 대부분의 경우에서 만족할 만한 좋은 성능을 보여준다. 따라서 대용량 ABox 추론에서 TBox의 크기가 작고 TBox 기반 추론이 적을 경우 KAON2 사용이 적합하며, 대용량 ABox 추론과 더불어 TBox 추론을 병행해야 하는 경우 온톨로지 Assertion Time이 많이 소비되는 단점이 있으나 Instance Store 사용이 적합하다.

향후 연구로는 Tableaux 알고리즘 기반 추론 시스템과 KAON2 그리고 Instance Store에서 보여준 문제점들의 개선이 요구되며 이러한 내용을 바탕으로 보다 최적화된 추론 시스템에 대한 연구가 필요하다.

참고문헌

[1] D.L. McGuinness, F.V.Harmelen : OWL Web Ontology Language Overview. W3C recommendation 10 Feb, 2004.
 [2] Genesereth, M., Nilsson, N. Logical Foundations of Artificial Intelligence, Morgan-Kaufman, 1987.
 [3] I. Horrocks : Optimising Tableaux Decision Procedures for Description Logics. PhD thesis, University of Manchester, 1997.
 [4] Y. Guo, Z. Pan, and J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems. Journal of Web Semantics 3(2), 2005, pp158-182.
 [5] F. Baader, W. Nutt : Basic Description Logics .The Description Logic Handbook. Cambridge University Press, 2003
 [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] KAON2 : The Karlsruhe ONtology and Semantic Web tool suite. <http://kaon.semanticweb.org/>
 [9] I. Horrocks and P.Patel-Schneider : DL Systems Comparison. In Proc. of DL'98, 1998.
 [10] RacerPro Homepage. <http://www.racer-systems.com/>
 [11] Pellet - OWL DL Reasoner. <http://www.mindswap.org/2003/pellet>.

Acknowledgement

본 논문은 정동부 및 정보통신연구진흥원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.