

온톨로지 스키마 추론을 위한 향상된 개념 검색방법

(An Enhanced Concept Search Method for Ontology Schematic Reasoning)

권순현^{*} 박영택^{**}
(Soon Hyun Kwon) (Young Tack Park)

요약 온톨로지 스키마 추론은 개념의 정합성을 테스트하고 계층구조를 자동으로 생성해 준다. 이를 위해 개념의 검색과정은 반드시 수행하여야 한다. 온톨로지 스키마 추론은 온톨로지에 정의된 모든 개념들 간의 포함관계를 테스트함으로써 수행된다. 이러한 포함관계 테스트 결과는 개념의 완전 그래프 생성여부에 따라 결정되며 이는 추론의 전체적인 성능에 결정적인 영향을 준다. 일반적으로 완전 그래프 생성 과정은 복잡도가 높은 과정으로 알려져 있으며 이는 낮은 추론성능을 유발시키는 원인이 되고 있다. 본 논문에서는 개념들간의 포함관계 테스트시 개념의 향상된 검색방법을 통해 불필요한 포함관계 테스트를 생략함으로써 온톨로지 스키마 추론의 성능을 향상시키는 방법을 제시하고자 한다. 이 향상된 검색방법은 개념의 포함관계 테스트 결과를 정의상 연관된 다른 개념에 전파함으로써 이루어진다.

키워드 : 온톨로지, 계층구조 추론, 포함관계, 최적화

Abstract Ontology schema reasoning is used to maintain consistency of concepts and build concept hierarchy automatically. For the purpose, the search of concepts must be inevitably performed. Ontology schema reasoning performs the test of subsumption relationships of all the concepts delivered in the test set. The result of subsumption tests is determined based on the creation of complete graphs, which seriously weighs with the performance of reasoning. In general, the process of creating complete graph has been known as expressive procedure. This process is essential in improving the leading performance. In this paper, we propose a method enhancing the classification performance by identifying unnecessary subsumption test supported by optimized searching method on subsumption relationship test among concepts. It is achieved by propagating subsumption tests results into other concept.

Key words : Ontology, Classification, Subsumption Relationship, Optimization

1. 서론

OWL(Web Ontology Language)은 시맨틱웹상 온톨

로지를 소유하고 사용하기 위한 W3C(World Wide Web Consortium)의 표준 언어이다[1]. OWL은 모든 온톨로지를 생성하고 추론하는 시스템에서 도메인의 개념·계층적인 지식을 표현하기 위해 사용된다. 즉 도메인의 지식을 표현하기 위해 온톨로지는 개념을 정의하고 개념들 간의 관계를 표현한다. 이러한 개념과 개념들 간의 관계는 논리기반 표현형식에서 제공하는 연산자를 사용함으로써 복잡한 개념들을 확장하여 서술한다. 또한 추가적인 도메인의 사실에 대한 정보를 표현하기 위해 인스턴스를 사용한다.

스키마 수준의 개념과 개념들 간의 관계를 표현하고 인스턴스를 통한 도메인의 사실정보를 표현하는 시스템을 지식 표현 시스템(Knowledge Representation System)이라고 한다[2]. 그러나 지식 표현 시스템에서 표현된 온톨로지는 도메인의 정적인 지식을 표현할 뿐

* 본 연구는 송실대학교의 지원을 받았습니다.

† 학생회원 : 송실대학교 컴퓨터학과
kwonshzzang@naver.com

** 종신회원 : 송실대학교 컴퓨터학부 교수
park@ssu.ac.kr

논문접수 : 2009년 3월 2일

심사완료 : 2009년 8월 26일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 받고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제36권 제11호(2009.11)

지능적인 서비스를 제공하기 위해서는 추론이 필요하다. 추론이란 도메인 상 정의된 명시적인(Explicit) 지식으로부터 묵시적인(Implicit) 지식을 도출해 내는 과정이다[2].

이러한 추론의 한 부류인 온톨로지 스키마 추론은 온톨로지에 정의된 개념의 정합성 여부와 개념들 간의 상하 관계를 자동으로 만들어 준다. 이러한 온톨로지 스키마 추론은 정적인 지식을 동적이고 지능적인 지식으로 확장하는 역할을 수행한다.

온톨로지 스키마 추론은 온톨로지에 포함된 각각의 개념들 간의 포함관계 테스트를 진행함으로써 수행되어진다[3]. 이러한 포함관계 테스트는 정해진 시간 내에 결과를 도출할 수 없는 이슈를 포함하고 있다. 이러한 문제를 해결하기 위해 많은 연구가 진행되고 있고 연구 결과 포함관계 테스트는 최소한 NP-Hard 또는 주어진 시간 내에 결과를 도출하지 못한다는 결론에 도달했다[3]. 하지만 포함관계 복잡도의 이슈를 해결하기 위한 많은 연구가 진행되고 있으며 많은 최적화 기법이 발표되었다. 예를 들면 정의정렬(Definition Order), 완전정의(Complete Define), 사전정의 포함자(Told Subsumer) 등이 그것이다[4]. 그러나 여전히 포함관계 테스트는 스키마 수준의 추론 복잡도를 증대시키고 성능을 저하시키는 요인으로 작용하고 있다.

본 논문에서는 온톨로지 스키마 수준의 추론에서 개념의 향상된 검색방법을 도입하여 추론의 성능을 향상시키는 최적화 검색기법을 제시한다. 복잡도가 높은 포함관계 테스트 횟수를 줄임으로써 전체적인 추론성능을 향상시키는 것이 목표이다. 즉, 향상된 개념 검색방법을 통해 개념들 간의 포함관계를 유추함으로써 테스트가 불필요한 개념에 대한 테스트의 생략을 가능하게 한다.

이를 위한 본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 포커스인 온톨로지 스키마 추론 중 계층구조 추론에 대해 논한다. 3장에서는 추론을 위해 온톨로지에 정의된 개념의 검색방법인 하-상향 검색방법에 대해 논한다. 4장에서는 3장에서 제시한 개념의 검색방법에서 향상된 검색방법을 제안한다. 5장에서는 4장에서 제시한 제안을 실제 시스템으로 구현하여 실험을 진행한다. 이를 위해 일반적으로 사용되는 온톨로지 테스트 케이스의 추론성능 비교를 통해 제안된 검색방법의 효율성에 대한 실험을 진행한다. 6장에서는 본 논문의 결론에 대해 논할 것이다.

2. 온톨로지 계층구조 추론

온톨로지 계층구조 추론은 온톨로지의 포함관계에 기초한 개념들의 집합에서 부분정렬을 만드는 과정이다[5]. 온톨로지서 모든 개념의 포함관계를 부분적으로

계산, 저장하는 것은 유용하다. 그러나 실제로 많은 온톨로지 추론엔진은 임의의 온톨로지에 관한 포함관계를 결정할 수 없는 포함관계 테스트 알고리즘을 사용한다. 지난 몇 년 사이에 대용량 온톨로지의 Sound, Complete한 알고리즘이 개발되었다[3]. 이러한 알고리즘 대부분은 만족성(Satisfiability) 체크 알고리즘으로 설계되었고 First Order Tableaux Calculus에 유사하다. 예를 들면 개념 A가 개념 B를 포함한다면, 즉 $B \sqsubseteq A$ 라고 하면 $B \sqcap \neg A$ 가 만족하지 말아야 한다. 즉 $B \sqcap \neg A$ 의 해석(Interpretation)이 비공집합(non empty set)인 해석이 존재하지 말아야 한다. 이러한 만족성 문제는 완전 그래프(Tableau) 생성을 통해 진행되어진다. 완전 그래프 생성과정은 그것이 실패하면(명백히 논리적 모순(Contradiction)이 발생하면) 만족하지 않고 그렇지 않으면 만족한다고 할 수 있다.

온톨로지 계층구조 추론에서 포함관계 테스트 횟수는 개념의 서술에 정의된 내용을 바탕으로 한 순서에 영향을 받는다. 온톨로지 계층구조 추론은 온톨로지의 개념을 정의한 순서, 즉 정의정렬(Definition Order)상 개념을 정렬하여 이 순서에 의해 포함관계를 테스트하는 것이다[4]. 예를 들어 개념 C가 $C \sqsubseteq D_1 \sqcap D_2$ 로 정의되어 있을 때 정의정렬상 개념 D_1 과 D_2 가 개념 C보다 먼저 포함관계 테스트가 진행되어야 한다.

스키마의 개념 정보를 저장하는 구조는 개념 C의 사전정의 포함자(Told Subsumer)를 계산하기 위해 사용된다. 예를 들어 개념 C가 $C \sqsubseteq C_1 \sqcap \dots \sqcap C_n$ 로 정의되어 있을 때 개념 $C_i, 1 \leq i \leq n$ 은 개념 C의 사전정의 포함자이다[4]. 이러한 사전정의 포함자는 개념들 간의 포함정보를 내포하는 것이기 때문에 추가적인 포함관계 테스트를 생략할 수 있고 개념 계층구조에 포함될 수 있는 위치를 추론 전 미리 판단할 수 있는 방법이다[5].

개념 계층구조 추론시 개념 검색방법을 위해 하향식(Top-Down Search)와 상향식(Bottom-Up Search)방법을 사용한다. 하향식 검색방법은 정의정렬 순번으로 진행되어지며 개념 계층구조의 Top에서 시작하여 계층구조에 추가하고자 하는 개념의 상위개념을 검색한다. 상향식 검색방법은 역시 정의정렬 순번으로 진행되며 개념 계층구조의 Bottom에서 시작하여 개념계층 구조에 추가하고자 하는 개념의 하위개념을 검색한다. 이때 개념 계층구조에 추가하고자 하는 개념은 자기 자신의 모든 사전정의 포함자가 추론되기 전까지는 자기 자신을 추론하지 말아야 한다[4]. 이는 모든 개념들이 정의정렬 상 순번으로 되어 있을 때 가능하다.

이러한 기본 개념을 바탕으로 본 논문에서는 앞에서 논한 포함관계에 의해 유도되는 온톨로지 개념 계층구조를 계산하기 위한 절차에 초점을 맞춘다. 즉, 이러한

작업은 전체적인 개념의 부분정렬을 계산하기 위한 과정이다. 앞에서도 언급하였듯이 포함관계 테스트 과정은 복잡도가 높다. 전체적인 추론의 성능을 높이기 위한 전략은 포함관계 테스트의 횟수를 줄이는 것이다.

3. 하향 · 상향식 검색

기본적으로 온톨로지 계층구조 추론은 개념들 간의 부분정렬을 반복적으로 만드는 과정이다. 이러한 정렬은 초기에 개념 Top(T)과 Bottom(B)의 부분정렬, 즉 $B < T$ 로 생성되어진다[5]. 새로운 개념 C가 개념 계층구조에 추가 되어질 때 모든 과정은 자기 자신의 직접적인 상위개념 집합과 직접적인 하위개념 집합을 결정한다. 이러한 과정은 재귀적으로 수행되어지며 이러한 두 개의 개념집합으로부터 추가된 개념 C의 개념상 위치를 결정한다. 상위개념 집합을 생성하는 과정은 하향식 검색방법으로 하위개념 집합은 상향식 검색방법으로 진행되어진다.

하향식 검색은 이미 정의된 개념 계층의 가장 상위개념 Top으로부터 진행되어진다. 각각의 개념 계층상 포함된 개념 X는 개념계층 X_i 에 포함되는 개념, 즉 $X \in X_i$ 이다. 이 때 추론하기 위한 개념 C가 개념 계층구조의 개념 X의 직접적인 하위개념 집합과 포함관계가 존재하는지를 테스트한다. 즉 개념 C와 개념 X의 부분정렬 ($C \leq X$)을 계산하게 된다. 만약 부분정렬 테스트 결과 하위개념이 존재하면 재귀적으로 개념 X의 하위개념에 대한 검색을 진행하게 되며 그렇지 않으면 개념 C를 개념 X와 동등 수준의 계층에 포함하게 된다.

개념계층 X_i 에 있는 각각의 개념 X의 방문횟수 카운트와 추론을 하기 위한 개념 C의 중복된 포함관계 테스트를 진행하는 것을 방지하기 위해 하향식 검색방법은 이미 테스트를 진행한 개념에 대해 "Visited"라고 하는 지시자를 둔다. 예를 들어 개념계층 X_i 에 있는 개념 X의 "Visited" 지시자가 "True"이면 이 개념은 이미 방문한 개념이므로 다음에 포함관계 테스트를 진행하지 않는다. 또한 개념 X의 지시자가 "Positive"이면 포함관계 테스트가 성공되었다는 것을 의미하고 "Negative"이면 포함관계 테스트의 결과가 실패하였다는 것을 의미한다. "Unknown"이면 아직 포함관계 테스트가 진행되지 않았음을 의미한다. 아래 그림 1은 하향식 검색방법에 대한 의미코드를 나타낸 것이다.

위의 그림에서 보듯이 하향식 검색방법은 두 개의 개념을 입력으로 한다. 현재 개념 c는 현재의 계층구조에 추가하고자 하는 개념이고 개념 x는 개념 계층구조 X_i 에서 현재 테스트 중인 개념이다. 이러한 개념 x에 대한 직접적인 하위개념을 대상으로 포함관계 테스트가 진행되어진다.

하향식 검색방법은 $x = T$ 으로 초기화된다[5]. x의 직접적인 하위개념 집합중 하나인 y가 추론하고자 하는 개념 c를 포함하는지, 즉 $c \leq y$ 에 대한 포함관계 테스트를 진행한다. 이러한 과정은 "simple-top-sub?"에서 진행되어진다. 개념계층이 트리형태일 필요가 없으므로 복잡도가 높은 포함관계 테스트 결과를 저장하여 중복된 포함관계 테스트 "simple-top-sub?" 모듈을 중복 호출하지 않게 한다.

```

Top-Search(c, x) =
  mark(x, "visited")
  for all y with y < x do
    if simple-top-sub?(y, c) then
      Pos_Succ ← Pos_Succ ∪ {y}
    fi
  od

if Pos_Succ is empty then
  Result ← {x}
else
  for all y ∈ Pos_Succ do
    if not marked?(y, "Visited") then
      Result ←
        Result ∪ Top-Search(c, y)
    fi
  od
fi
    
```

```

simple-top-sub?(y, c) =
  if marked(y, "Positive") then
    Result ← True
  elsif subs?(y, c) then
    mark(y, "Positive")
    Result ← True
  else
    mark(y, "Negative")
    Result ← False
  fi
    
```

그림 1 하향식 검색 알고리즘

포함관계 테스트 결과 “Positive”한 결과는 Pos_Succ에 저장하게 된다. 만약 이 곳에 포함되어진 개념이 없다면 현재 테스트를 진행중인 개념을 개념계층 구조에 포함하고 결과를 반환한다. 그렇지 않으면 Pos_Succ에 포함되어진 각 요소에 대해 이미 포함관계 테스트가 진행되지 않은 개념, 즉 개념의 “Marked” 지시자가 “Unknown”으로 되어 있는 개념이라면 재귀적으로 다시 하향식 검색모듈을 호출하게 된다.

상향식 검색방법은 하향식 검색방법과 상대적으로 행해진다. 상향식 검색방법은 개념계층 구조상 최하위개념인 Bottom에서 시작한다. 즉, $x = 1$ 으로 초기화된다 [5]. 개념계층 X_i 에 포함되어 있는 각각의 개념에 대해 직접적인 상위개념중의 하나인 y 가 현재 추론하고자 하는 개념 c 에 포함되는지를 검사한다. 즉, $y \leq c$ 가 만족하는지를 검사한다. 만약 포함관계 테스트가 만족한다면 그 개념의 상위개념에 대해 재귀적 검색을 진행한다. 그렇지 않으면 현재 테스트 중인 개념을 개념계층 X_i 에 결과를 저장한다. 아래 그림 2는 상향식 검색방법에 대한 의미코드를 나타낸다.

위의 그림에서 보듯이 개념계층 X_i 의 개념 x 의 직접적인 상위개념과 입력과 개념 c 에 대한 포함관계 테스트를 진행하며 이러한 과정의 “simple-bottom-sub?”에서 수행되어진다. 다른 일련의 과정은 하향식 검색방법과 유사하다.

4. 향상된 하향 · 상향식 검색

앞에서도 논했듯이 포함관계 테스트는 복잡도가 높은

과정이다. 개념의 계층구조 추론에서 전체적인 성능은 포함관계 테스트의 횟수와 비례한다. 본 논문은 개념의 하 · 상향식 검색방법에서 포함관계 테스트를 진행하지 않아도 개념들 간의 포함관계를 묵시적으로 확인할 수 있는 향상된 검색방법을 사용하여 불필요한 포함관계 테스트를 생략함으로써 전체적인 추론의 성능을 향상시키는데 있다. 이것을 위한 기본적인 아이디어는 다음과 같다.

4.1 기본 제언

- 부정정보 하향전파(Negative Information Down Propagation) : 개념 계층구조에 있는 개념 A에 대한 포함관계 테스트시 부정정보 일 때 이 정보는 하향식 검색에서 개념계층의 하위방향으로 전파한다.
- 긍정정보 상향전파(Positive Information Up Propagation) : 개념 계층구조에 있는 개념 A에 대한 포함관계 테스트시 긍정정보 일 때 이 정보는 하향식 검색에서 개념계층의 상위방향으로 전파한다.
- 부정정보 상향전파(Negative Information Up Propagation) : 개념 계층구조에 있는 개념 A에 대한 포함관계 테스트시 부정정보 일 때 이 정보는 상향식 검색에서 개념계층의 상위방향으로 전파한다.
- 긍정정보 하향전파(Positive Information Down Propagation) : 개념 계층구조에 있는 개념 A에 대한 포함관계 테스트시 긍정정보일 때 이 정보는 하향식 검색에서 개념계층의 하위방향으로 전파한다.
- 하향식 검색에서 긍정정보 추가(Positive Information gain for Top Down Search) : 개념 A가 개념 B에 포함된다면 개념 B를 포함하는 모든 개념 역시 개념

```

Bottom-Search(c, x) =
  mark(x, "visited")
  for all y with x <i y do
    if simple-bottom-sub?(y, c) then
      Pos_Succ ← Pos_Succ ∪ {y}
    fi
  od

if Pos_Succ is empty then
  Result ← {x}
else
  for all y ∈ Pos_Succ do
    if not marked?(y, "Visited") then
      Result ←
        Result ∪ Bottom-Search(c, y)
    fi
  od
fi
    
```

```

simple-bottom-sub?(y, c) =
  if marked(y, "Positive") then
    Result ← True
  elsif subs?(y,c) then
    mark(y, "Positive")
    Result ← True
  else
    mark(y, "Negative")
    Result ← False
  fi
    
```

그림 2 상향식 검색 알고리즘

A를 포함한다.

- 하향식 검색에서 부정정보 추가(Negative Information gain for Top Down Search) : 개념 A가 개념 B에 포함하지 않는다면 개념 B를 포함하는 모든 개념 역시 개념 A를 포함하지 않는다.
- 상향식 검색에서 긍정정보 추가(Positive Information gain for Bottom Up Search) : 개념 A가 개념 B에 포함된다면 개념 B에 포함되는 모든 개념들도 개념 A를 포함한다.
- 상향식 검색에서 부정정보 추가(Negative Information gain for Bottom Up Search) : 개념 A가 개념 B에 포함하지 않는다면 개념 B를 포함하는 모든 개념 역시 개념 A를 포함하지 않는다.

하향식 검색에서 추론하고자 하는 개념 C와 개념계층 X_i 의 개념 X의 포함관계 테스트 결과 부정정보로 되었다면 개념 X의 모든 하위개념으로 부정정보가 전파된다. 예를 들어 개념 C가 개념 D에 포함하지 않는다면, 즉 $C \not\subseteq D$ 이면 개념 D의 하위개념 D' ($D' \subseteq D$) 역시 개념 C를 포함하지 않는다($C \not\subseteq D'$)[4]. 이러한 정보는 개념계층 X_i 의 특정 개념 X의 모든 하위 개념들로 전파되며 이를 통해 하위 개념의 지시자 "Marked"가 모두 부정정보, "Negative"로 세팅된다. 이는 이후의 개념계층 구조에 추가하고자 하는 개념과의 포함관계 테스트 시 이미 해당 개념들은 "Marked" 정보가 부정정보로

세팅되었기 때문에 생략할 수 있다. 이 제안과 대칭적으로 하향식 검색에서 추론하고자 하는 개념 C와 개념계층 X_i 의 특정 개념 X의 포함관계 테스트시 긍정정보 즉, 개념 X의 "Marked" 지시자가 "Positive"로 되었다면 개념 X의 모든 상위개념으로 긍정정보가 전파된다. 긍정정보가 전파된 개념들은 이후의 포함관계 테스트시 생략이 가능하다. 이에 반하여 상향식 검색방법은 포함관계의 테스트 결과에 따라 긍정정보는 개념계층의 하위방향으로 부정정보는 개념계층의 상위방향으로 전파한다.

4.2 향상된 하·상향식 검색방법

온톨로지 계층구조 추론을 위한 하·상향식 검색을 진행을 할 때 포함관계 테스트의 결과를 개념 계층에서 상·하 방향으로 전파하는 것에 집중한다. 아래 그림 3은 앞의 기본 아이디어를 바탕으로 재구성한 향상된 하향식 검색 알고리즘이다.

하향식 검색방법에서 개념계층구조에 포함하고자 하는 개념 c와 개념 y의 직접적인 상위 개념 z에 대한 포함관계의 테스트 결과가 부정정보라고 하면, 즉 $c \not\subseteq z$ 라고 하면 복잡한 포함관계 테스트 과정 없이 $c \not\subseteq y$ 라는 결론을 도출할 수 있다[3]. 이러한 효과를 최대한 증대시키기 위해 개념 y의 모든 상위개념은 y의 포함관계 테스트 전에 모두 테스트 되어져야 한다. 또한 하향식 검색에서 개념계층구조에 포함하고자 하는 개념 c와 개

```

Top-Search(c, x) =
  mark(x, "visited")
  for all y with x <i y do
    if enhanced_top_subs?(y, c) then
      Pos_Succ ← Pos_Succ ∪ {y}
      Propagate_Info("Positive", "Up", y)
    else
      Propagate_Info("Negative", "Down", y)
    fi
  od

  if Pos_Succ is empty then
    Result ← {x}
  else
    for all y ∈ Pos_Succ do
      if not marked?(y, "Visited") then
        Result ←
          Result ∪ Top-Search(c, y)
      fi
    od
  fi
  
```

```

enhanced-top-subs?(y, c) =
  if marked(y, "Positive") then
    Result ← True
  elsif marked(y, "Negative") then
    Result ← False
  elsif exists z with z <i y and
    marked?("Positive", z) then
    mark(y, "Positive")
    Result ← True
  elsif
    for all z with z <i y do
      if enhanced_top_subs?(z, c) and
        subs?(y, c) then
        mark(y, "Positive")
        Result ← True
      else
        mark(y, "Negative")
        Result ← False
      fi
    od
  fi
  
```

그림 3 향상된 하향식 검색 알고리즘

```

Bottom-Search(c, x) =
  mark(x, "visited")
  for all y with y < x do
    if enhanced_bottom_subs?(y, c) then
      Pos_Succ ← Pos_Succ ∪ {y}
      Propagate_Info("Positive", "Down", y)
    else
      Propagate_Info("Negative", "Up", y)
  fi
od

if Pos_Succ is empty then
  Result ← {x}
else
  for all y ∈ Pos_Succ do
    if not marked?(y, "Visited") then
      Result ←
        Result ∪ Bottom-Search(c, y)
    fi
  od
fi

enhanced-bottom-sub?(y, c) =
  if marked(y, "Positive") then
    Result ← True
  elsif marked(y, "Negative") then
    Result ← False
  elsif exists z with z < y and
    marked?("Negative", z) then
    mark(y, "Negative")
    Result ← False
  elsif
    for all z with z < y do
      if enhanced_bottom_subs?(z, c) and
        subs?(y, c) then
        mark(y, "Positive")
        Result ← True
      else
        mark(y, "Negative")
        Result ← False
      fi
    od
  fi

```

그림 4 향상된 상향식 검색 알고리즘

념 y 의 직접적인 하위개념 z 에 대한 포함관계 테스트 결과가 긍정정보라고 하면, 즉 $c < z$ 라고 하면 역시 복잡한 포함관계 테스트 과정 없이 $c < y$ 라는 결론을 도출할 수 있다. 이러한 이점을 최대한 살리기 위해 y 의 모든 하위개념은 y 가 수행되기 전에 테스트 되어져야 한다.

개념계층구조를 통해 부정정보의 하향전파, 긍정정보의 상향전파를 진행하는 것은 복잡한 포함관계 테스트의 횟수를 줄이는 효율적인 방법을 제시한다. 이것은 일반적인 하향식 검색방법의 Top-Search 메소드의 수정을 통해 쉽게 이루어질 수 있다.

아래 그림 4는 앞의 기본 아이디어를 바탕으로 재구성한 향상된 상향식 검색 알고리즘이다.

위의 향상된 상향식 검색방법은 기본 제안에 따라 구성되었으며 완전히 하향식 검색방법과 대칭적이라고 할 수 있다.

아래 그림 5는 일반 하향 검색방법과 향상된 하향 검색방법을 통해 개념 C에 대한 개념계층추론을 진행할 때 수행되어지는 포함관계 테스트의 횟수를 비교한 그림이다.

위의 그림에서 보듯이 일반 하향 검색방법과 향상된 하향 검색방법은 Top의 직접적인 하위 개념인 X1, X2, X3와의 포함관계 테스트를 통해 진행된다. 개념 X3에서 향상된 하향식 검색방법은 포함관계 테스트의 결과 부

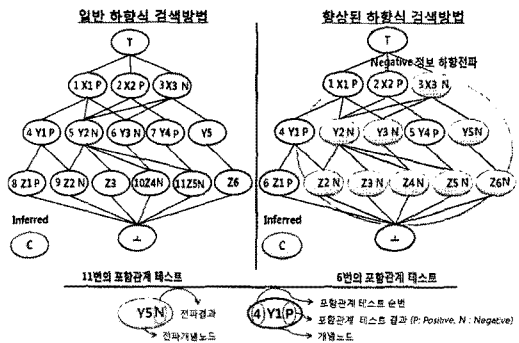


그림 5 하향식 검색에 대한 포함관계 테스트 횟수 비교

정정보를 X3의 모든 하위 개념으로 하향 전파한다. 이로써 개념 X3의 하위개념 Y2, Y3, Y5, Z2, Z3, Z4, Z5, Z6의 모든 개념에 대한 포함관계 테스트를 생략할 수 있다. 그 개념들은 명시적으로 부정으로 결정할 수 있다. 결과적으로 개념 C를 계층구조에 포함시키기 위해서 일반 하향 검색은 11번의 포함관계 테스트를 진행한 반면 향상된 하향 검색방법은 6번의 포함관계 테스트를 진행하였다.

5. 성능평가

본 실험은 온톨로지 계층구조 추론에서 일반 하향 검색방법과 향상된 하향식 검색방법을 비교한다. 서론에서

논했듯이 계층 구조의 추론은 복잡도가 높은 포함관계 테스트 횟수에 의존한다고 할 수 있다. 따라서 본 실험은 온톨로지 계층구조 추론에 있어 포함관계 테스트 횟수와 전체적인 개념계층구조 추론의 속도를 비교한다. 본 실험은 실제 추론엔진 모듈을 이용하여 구현하였고 테스트 케이스로는 Swoogle(<http://swoogle.umbc.edu/>)에서 등록된 온톨로지를 기준으로 테스트를 진행하였다. 이는 Swoogle 온톨로지는 시맨틱 웹 환경에서 현재 많은 참조와 활용을 하는 온톨로지이기 때문이다. 본 실험은 CPU(Intel(R) Core2 660@ 2.4GHZ)와 메인메모리(4G)의 시스템에서 진행되었다.

아래 표 1은 하향식 검색에 있어서 각 검색방법에 대한 테스트 결과표이고 그림 6은 성능 비교 도표이다.

본 실험 결과 koala 온톨로지는 개념수가 20개인 스펙의 온톨로지이다. 일반 하향식 검색과 향상된 하향식 검색의 테스트 결과 포함관계 테스트 횟수는 각각 172번과 130번이다. 추론속도도 포함관계 테스트 횟수와 비례하여 각각 3,013ms와 2,840ms이다. 이것은 위의 그림 5의 계층 구조 추론과정에서 보듯이 추론하고자 하는 개념과 계층구조의 개념과의 한번의 포함관계 테스트는 부정정보는 하향전파, 긍정정보는 상향 전파를 통해 테

스트 결과의 전파를 통해 불필요한 포함관계 테스트를 생략할 수 있기에 가능한 것이다.

6. 결론

우리는 온톨로지에 속한 개념에 대한 계층구조를 자동으로 계산해 주기 위한 계층구조 추론에서 향상된 기법에 대해 논했다. 이를 위한 향상된 검색 방법을 적용하였고 성능평가를 통해 입증하였다. 결론적으로 향상된 기법은 복잡하고 비용이 큰 포함관계 테스트의 횟수를 줄임으로써 진행된다. 성능평가를 통해 향상된 검색방법은 온톨로지의 스펙이 크고 개념 계층구조의 복잡도가 높을수록 향상율이 높다는 것을 알 수 있다. 도메인의 지식표현을 위해 사용되는 온톨로지는 개념의 명세를 통해 모델링되어진다. 잘 정의된 모델링 방법은 계층구조 추론에서 복잡한 구조를 산출하게 된다. 따라서 이러한 온톨로지에서는 본 논문의 제안은 최적의 성능을 발휘한다고 할 수 있다.

참고 문헌

- [1] T.R.Gruber : November 1995, "Toward Principles for the design of ontologies used for knowledge

표 1 일반 하향식 검색과 향상된 하향식 검색방법의 추론성능 비교표

온톨로지	개념수	일반 하향식 검색		향상된 하향식 검색	
		포함관계횟수	추론속도(ms)	포함관계횟수	추론속도(ms)
koala.owl	20	172	3,013	130	2,840
pizza.owl	95	3,518	3,830	3,063	3,250
NMR.owl	300	8,686	3,356	8,321	3,306
miniEconomy.owl	337	13,484	3,656	13,245	3,180
miniTransport.owl	444	33,281	5,321	28,652	4,781
dice.owl	527	15,966	3,609	13,245	3,180

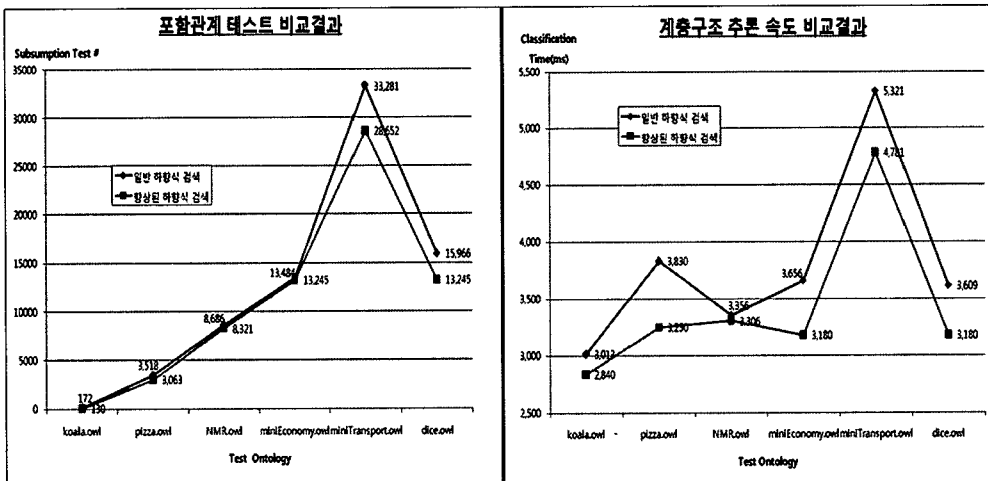


그림 6 일반 하향식 검색과 향상된 하향식 검색방법의 추론성능 비교 도표

- sharing," *International Journal of Human Computer Studies*, vol.43 Issues 4~5.
- [2] Daniel Nardi, Ronald J. Brachman : 2003, "An Introduction to Description Logics," *Description Logics HandBook*, pp.5-6.
- [3] Manfred Schmidt-Schauß and Gert Smolka : February 1991, "Attributive Concept Description with Complements," *Artificial Intelligent*, 48:1~26.
- [4] Dmitry Tsarkov, Ian Horrocks and Peter F. Patel-Schneider : March 2007, "Optimising Terminological Reasoning for Expressive Description Logics," *University of Manchester, UK and Bell Labs Research Lucent Technologies, USA*.
- [5] Franz Badler, Bernhard Hollunder, Bernhard Nebel, Hans-Jurgen Profitlich : October 1992, "An Empirical Analysis of Optimising Techniques for Terminological Representation System, Principles of Knowledge Representation and Reasoning," *Proceeding of the 3th International Conference, Cambridge, MA*.



권 순 현

2000년 9월 현대정보기술 근무. 2003년 2월 숭실대학교 정보과학대학원 컴퓨터학과(석사). 2006년 2월~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 인공지능, 시맨틱웹, 온톨로지추론, 지능형 에이전트

박 영 택

정보과학회논문지 : 소프트웨어 및 응용
제 36 권 제 5 호 참조