

# 자동화된 추론을 이용한 퍼지 조건연산자의 비교 분석

## Comparison of Fuzzy Implication Operators using Automated Reasoning

김 용 기\*  
Yong-Gi Kim\*

※본 논문은 1994년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음

### 요 약

퍼지 정보 검색 기법을 이용한 가중치 선정 전략은, 주어진 입력 문제로 부터 가중치의 패턴과 가중치를 자동적으로 선정함으로써, 추론 시간과 공간을 줄인다. 퍼지 가중치 선정 전략은 퍼지 삼각 관계곱 및 퍼지 신속 관계 알고리즘을 사용하기 때문에, 선택하는 퍼지 조건연산자와  $\alpha$ -cut에 따라서 결론을 추출하는 시간과 추론 단계의 수가 틀린다. 본 논문에서는, 퍼지 정보 검색을 이용한 추론 전략의 조절 방법에 대하여 개괄적으로 보이고, 잘 알려진 블록 문제에 적용하여 주어진 영역에서 가장 적합한 퍼지 조건연산자와  $\alpha$ -cut을 선정한다.

### ABSTRACT

The weighting mechanism based on fuzzy information retrieval technique helps the user of the resolution based mechanical theorem prover decide the weighting pattern and the weights to reduce the deduction time and space automatically from input problem. The mechanism employs the triangle fuzzy relational products and fast fuzzy relational algorithm, and produces the answer in different length of time and number of deduction steps according to the fuzzy implication operators and  $\alpha$ -cuts chosen. In this paper, the method for control of inferential strategies of resolution based architectures is outlined and the method was applied to the well known block world problem to decide the best suitable implication operator and  $\alpha$ -cut for the specified domain.

### I. 서 론

퍼지 이론은 인공지능, 로봇공학, 제어, 데이터베이스 등 광범위한 분야에 응용되어, 종래의 방법으로 해결하기 힘들었던 벽을 넘거나 약점을 해결함으로써, 근래 정보 과학의 중요한 방법론으로 위치를 점하고 있다[1]. 특히, 전문가 시스템의 지식 표현이나 추론에 있어서, 퍼지 이론은 종래의 이진 집합(crisp set)으로는 해결하기 어려운 불확실성(uncertainty)의 처리에 중요한 역할을 하게 되었다.

---

\*경상대학교 컴퓨터과학과  
Dept. of Computer Science, Kyungsang National University

“if P then Q”를 표현하는 “ $P \rightarrow Q$ ”에서 조건연산자(implication operator) “ $\rightarrow$ ”는 특히 불확실성을 수반하는 퍼지 추론 또는 퍼지 값의 계산에서 적절한 결론을 얻기 위하여 주의 깊게 선택하여야 한다. 종래의 이진 집합의 불 논리(boolean logic)에서는 명제 P와 Q의 값에 따라서, 그 결과가 참 또는 거짓 만을 결정하며, 그의 증명역시 명약관화하여 이론의 여지가 없다. 그러나, 불확실성을 처리하는 퍼지 조건 연산자(fuzzy implication operator)의 경우, 그 필요성의 중대함에도 불구하고, 그 결과값을 결정 짓는 최선의 방법이 유동적이다. 그 이유는 boolean logic에서와는 달리 명제 P 또는 Q가 참, 거짓 이외에  $[0, 1]$ 의 어떠한 값도 가질 수 있는 관계로 그 결과 또한  $[0, 1]$  사이의 값으로 결정될 수 있기 때문이다. 이에 많은 연구자들에 의하여, 수많은 퍼지 조건 연산의 결과를 결정하는 방법[2, 3, 4]들이 개발되어 있으나, 그 결과들 또한 이론적으로 증명할 수 없는 관계로 인하여, 주어진 문제의 영역에 따라서, 경험적인 방법에 의하여 일부의 퍼지 조건 연산자들이 선택될 수밖에 없는 실정이다.

resolution에 근거한 자동화된 추론(automated reasoning)의 도구들인 ITP[5] 또는 OTTER[6]에 내재되어 있는 가중치 선정 전략(weighting strategy)[7]의 경험적인 면을 제거하기 위하여[8, 9, 10, 11] 응용된 퍼지 정보 검색 기법[12, 13, 14]의 경우에도, 퍼지 조건 연산자의 변화에 따라 그 효율성에 큰 영향을 받는다. 이에, 퍼지 정보 검색 기법을 적용하기 가능한 자동화된 추론의 몇 분야를 선택하여, 퍼지 조건 연산자의 변화에 따른 가중치 선정 전략의 효율성을 비교 분석함으로써, 주어진 영역에 가장 적합한 퍼지 조건 연산자를 선택한다.

본 연구에서의 문제 해결을 위한 논리적 추론의 가속화를 위한 기법의 적용 및 퍼지 조건연산자의 비교 분석을 위한 실험은 기계적 정리 증명기인 OTTER를 이용한다.

퍼지 정보 검색 기법과, 그 기법의 자동화된 추론에의 응용과 신속 퍼지 알고리즘(Fast fuzzy algorithm)[15]을 2장, 3장 및 4장에서 각각 설명 한후, 5장에서 그 퍼지 기법을 적용하여 퍼지 조건연산자를 비교 분석할 수 있는 응용 분야의 선택하고, 6장에서 퍼지 조건연산자를 비교 분석한다.

## II. 퍼지 정보 검색기법(Fuzzy Information Retrieval Technique)

Bandler와 Kohout에 의하여 고안된 퍼지 정보 검색 기법은 정보검색 분야[14, 16] 외에 의료 분야의 진단 및 환자 관리[17, 18, 19, 20], 수기 분류[21], 자동화된 추론[9, 11, 22] 등 그 범위를 넓히고 있다. 퍼지 정보 검색 모델은 퍼지 탐색 요구(Fuzzy Search Request) 및 퍼지 관계 요구(R-request)로 구성되며, 본 연구를 위하여 이용되는 것은 퍼지관계요구이다.

퍼지 관계 요구는 관계정도구조(thesaurus)에 관련된 요구이다. 항목 사이의 계층 또는 포함 관계인 관계정도 구조를 구축하기 위해 이용되는 연산은 퍼지 삼각 논리곱과 퍼지 사각 논리곱이다[3][18]. A에서 B로의 퍼지 관계 R과 B에서 C로의 퍼지 관계 S에 대하여 A에서 C로의 삼각 논리곱과 사각 논리곱은 다음과 같이 정의되며 이를 이용하여 관계정도구조를 추출한다.

$$\begin{aligned} (1) (R \triangleleft S)_{ik} &= \mu_{R \triangleleft S}(a_i, c_k) = \pi(a_i R \subseteq S c_k) \\ &= 1 / |B| \sum (R_{ij} \rightarrow S_{jk}) \\ (2) (R \triangleright S)_{ik} &= \mu_{R \triangleright S}(a_i, c_k) = \pi(a_i R \supseteq S c_k) \\ &= 1 / |B| \sum (R_{ij} \leftarrow S_{jk}) \\ (3) (R \square S)_{ik} &= \mu_{R \square S}(a_i, c_k) = \pi(a_i R = S c_k) \\ &= 1 / |B| \sum (R_{ij} \leftrightarrow S_{jk}) \end{aligned}$$

여기서  $(R \triangleleft S)_{ik}$ 는  $a_i R$ 의 퍼지 afterset이  $S c_k$ 의 퍼지 foreset에 포함될 평균 등급을 나타낸다.

삼각 서브 관계 논리곱(triangle sub-product relation)인  $(R \triangleleft S)$ 는, 퍼지 관계 R은 A에서 B로의 관계이고 퍼지 관계 S는 B에서 C로의 관계라 할 때  $a_i$ 가  $b_j$ 로의 R관계 정도가  $b_j$ 가  $c_k$ 로의 관계 정도인 S를 포함(imply)하는 정도를 의미하는 것이다. 이 연산의 결과로 A에서 C로의 관련 정도를 알 수 있게 한다.

관계정도구조를 구축하기 위해서 논리곱으로 연산을 하며 연산은 퍼지 입력 행렬과 그 전치 행렬을 이용한다. R이 자료(document)에서 항목(term)으로의 관계( $D \rightarrow T$ )라면  $R^T$ 는 항목에서 자료로의 관계인 ( $T \rightarrow D$ )가 된다. 그러므로 ( $R^T \triangleleft R$ )은 항목에서 항목으로의 새로운 관계가 된다. 즉, 다음과 같이 요약할 수 있다.

$$D_{ik} = d_i \text{에 대한 } t_k \text{의 관계 정도} \\ = D^T_{ki}$$

$D^T$ 와  $D$ 에 대한 삼각논리곱과 사각논리곱의 각각의 의미는 다음과 같다.

$$(D^T \triangleleft D)_{ik} = 1 / N_j \sum (D^T_{ij} \rightarrow D_{jk}) \\ \text{: } t_i \text{가 } t_k \text{보다 더 구체적인 정도의 평균 등급}$$

$$(D^T \triangleright D)_{ik} = 1 / N_j \sum (D^T_{ij} \rightarrow D_{jk}) \\ \text{: } t_i \text{가 } t_k \text{보다 더 일반적인 정도의 평균 등급}$$

$$(D^T \square D)_{ik} = 1 / N_j \sum (D^T_{ij} \leftrightarrow D_{jk}) \\ \text{단, } a \leftrightarrow b = \text{MIN}(a \rightarrow b, a \leftarrow b) \\ \text{: } t_i \text{와 } t_k \text{가 유사한 정도의 평균 등급}$$

관계 요구의 출력은 위의 퍼지 관계곱을 이용하여 추출한, 각 항목 간의 포함 관계를 나타내는 관계 정도 구조로 부터 얻는다.

### III. 퍼지정보검색 기법을 이용한 자동화된 추론의 가중치 선정

Resolution에 근거한 자동화된 추론(Automated Reasoning)의 중요한 결점의 하나는 결론을 얻기에 필요하지 않은 중간 절들을 생산함으로써, 결론에 도달하는 길을 잃어 버린다는지, 또는 쉽게 컴퓨터의 메모리를 잠식함으로써 더 이상의 추론을 행하지 못하고 결론에 도달하지 못하는 점이었다[23]. 이에, 지원집합 전략(set of support strategy)[24] 또는 가중치 선정 전략(weighting strategy)[7] 등의 조절 방법들이 개발되어 이러한 결점들을 상당 부분 해결하게 되었다. 그러나, 자동 정리증명기(automated theorem prover) 들에 있어 이와 같이 조절 방법들의 중요성에도 불구하고, 가중치 선정 전략의 중요한 결점은 웨이팅을 주어진 입력 문제를 스스로 분석하여 결정하지 못하고, 전적으로 사용자의 경험 및 그 문제에 관한 지식에 의존하여 결정함으로써, 때로는 답에 도달하지 못하는 등의 시간과 기억 공간의 낭비가 있었다. 이러한 가중치 부여의 경험적인 면을 제거하고, 주어진 입력 문제를 스스로 분석하여 웨이팅을 자동적으로 결정하게 하는, 퍼지정보검색 기법을 이용한 가중치 선정 기법이 제안되었다[9, 11, 22].

기존의 정리 증명기 OTTER의 추론 구조에서, 정리 증명기가 추론을 위하여 선택하는 절의 우선순위는 OTTER의 사용자에 의해서 경험적으로 주어진 가중치에 의해서 결정된다. 제안된 구조에서는 이러한 경험적인 가중치를 퍼지 관계 알고리즘(fuzzy relational algorithm)(Bandler and Kohout[15])에 의해서 추출된 관계정도구조(thesaurus)로부터 결정된 가중치로 대체한다. 퍼지 정보 검색 기법의 자동화된 추론(automated reasoning)에의 적용은 근본적으로 다음의 세 가지 기본 요소로써 이루어진다.

1. 절(clauses :  $d_i$ )의 집합 D.
2. 항목(terms and literals :  $t_j$ )의 집합 T.
3. 절과 항목의 관계 R이며, 여기서 R은 퍼지 관계로서 다음의 형태로 표현된다.

$$R_{ij} \in R_f(D \rightarrow T)$$

그러면,  $R_{ij}$ 는 하나의 절( $d_i$ )이 어떤 항목( $t_j$ )에 관계 되는 정도를 나타낸다.

퍼지 가중치 선정 기법은 항목(term and literal)과 절(clause)사이의 관련성을 분석한 후 항목의 계층 구조인 관계정도구조(thesaurus)를 추출한다. 계층 구조에서 가장 높은 계층의 term이나 literal이 모순으로써 결론을 유도할 가능성이 가장 높은 것이다. 그러므로 term이나 literal의 웨이트는 계층 구조의 역으로 결정된다. 관계정도구조(thesaurus)를 추출하기 위한 입력 행렬의 행과 열은 절(clauses)과 항목(term과 literal)으로 구성된다. 행렬의 관계 등급은 사용자로부터 입력된 절을 이용하여 자동적으로 결정된다.

제안된 시스템에서 항목의 계층 구조인 관계정도구조(thesaurus)를 구성하는 관계(relation)는 아래의 식에 의해서 만들어진다.

$$(T \Rightarrow D) @ (D \Rightarrow T)$$

여기서, T : 항목(terms and literals)

D : 절(clauses)

@ : relational product (triangle super-product, sub-product and square product)

입력 퍼지 행렬의 행의 요소인 항목은, 주의 깊게 고안된 일련의 실험이나, 적절한 이론적인 고려를 바탕으로 경험적으로 결정된다. 이러한 명시된 실험에 포함된 요소들의 관련있는 특징은 OTTER에 표현된 문제의 수학적 특징에 의존한다. 이 퍼지 관계 행렬은 신속 퍼지 알고리즘(Fast fuzzy algorithm)을 이용하여, 항목의 계층 구조인 관계정도구조(Thesaurus)를 추출하며, 자동적으로 추출된 관계정도구조가 정리증명기 OTTER의 추론 과정을 가속화한다(자세한 구조는 [11] 참조).

입력 퍼지 행렬의 하나의 절( $d_i$ )이 어떤 항목( $t_j$ ) 사이의 관계되는 정도( $R_{ij}$ )를 결정하기 위하여, 다음의 규칙들을 적용한다.

1. 항목( $j$ )이 절( $i$ )의 요소일 경우 1.
2. 항목( $j$ )이 절( $i$ )의 요소가 아닌 경우 0.
3. 항목( $j$ )이 절( $i$ )의 아래항목의 요소(subterm, 예 :  $g(a)$ )인 경우 0.5.
4. 항목( $j$ )이 절( $i$ )의 아래항목의 아래항목의 요소(예 :  $g(g(a))$ )인 경우  $0.5 * 0.5$
5. 항목( $j$ )이 절( $i$ )의 요소이고, 또한 아래항목의 요소(예 :  $P(a, b, g(a))$ )인 경우, 위 1부터 4까지의 규칙을 적용한 것 중 큰 값.

결정된 퍼지 입력 행렬로부터, 퍼지 정보 검색 기법의 관계 요구(relational request)[14][16]를 적용하여, 항목들간의 관계 정도를 나타내는 구조(thesaurus)를 추출함으로써, 웨이팅을 결정하게 된다. 관계정도구조(thesaurus)를 결정하기 위한 관계요구(r-request)는 아래의 퍼지 관계곱들을 이용하여 그 출력을 결정한다.

$$D_j = \text{절 } d_i \text{에 항목 } t_j \text{의 관련된 정도} \\ = D_{ji}^T \text{라고 할때,}$$

D와  $D^T$ 의 삼각논리곱(triangle sub-product, super-product) 및 사각논리곱(square-product)는 아래와 같은 의미를 갖게 된다.

$$(D^T \triangleleft D)_{ik} = \text{항목 } t_i \text{가 항목 } t_k \text{ 보다 empty clause에 덜 관련된 정도} \\ (D^T \triangleright D)_{ik} = \text{항목 } t_i \text{가 항목 } t_k \text{ 보다 empty clause에 더 관련된 정도} \\ (D^T \square D)_{ik} = \text{항목 } t_i \text{와 항목 } t_k \text{가 유사한 정도.}$$

계층 구조는 위의 복합 관계곱을 이용하여 구해진다. 위의 퍼지 삼각논리곱들을 적용한 결과는 또 다른 하나의

행렬의 형태를 이루며, 선택한  $\alpha$ -cut을 적용하여 그림 1과 같은 출력 행렬을 얻는다. 이 출력 행렬에 신속 퍼지 알고리즘(Fast fuzzy algorithm)을 적용하여 항과 항 사이의 포함 관계를 쉽게 알아 볼 수 있는 Hasse diagram을 구성하며, 이 Hasse diagram을 우리는 관계정도구조라 부른다(그림 2).

	A	B	C	SI	[]	Noop
A	1	0	0	1	0	0
B	0	1	0	1	0	0
C	0	0	1	1	0	0
SI	0	0	0	0	0	0
[]	1	1	1	1	1	1
Noop	1	1	1	1	1	1

그림 1. 출력행렬

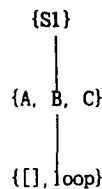


그림 2. 관계정도구조

#### IV. 신속 퍼지 관계 알고리즘(Fast Fuzzy Relational Algorithm)

계층 구조를 유도하기 위한 실제의 연산은 신속퍼지관계 알고리즘을 이용하여 구현한다. 신속퍼지관계 알고리즘은, 앞에서 설명한 관계곱에서 local preorder에 대한 정보를 이끌어 내며, local preorder로 부터 계층 구조를 추출하여 Hasse Diagram을 구성한다. 다음은 신속 퍼지 관계 알고리즘과 Hasse diagram을 유도하기 위한 fuzzy local preorder의 처리에 대한 개념이다(자세한 내용은 [15] 참조).

\* local preorder closure를 구하기 위한 신속 알고리즘 :

1. Form  $T_0 = locref\ clo\ R = E_R \cup R$ .
  2. Compute  $T_1 = T_0^2, T_2 = T_1^2, \dots, T_s = T_{s-1}^2$ , until  $T_s = T_{s-1}$  or  $2^s \geq n$ .
  3. Then  $locpre\ clo\ R = T_s$ .
- where, n is the cardinality of the universe X.

\* 퍼지 local preorder 를 구하기 위한 개념적 절차 :

1. Take  $\alpha$ -cut  $R_\alpha$ .
2. Form  $S = sym\ int\ R_\alpha$ .
3. Remove the zero-class  $C_0$ , consisting of those x's which are unrelated by S to any elements.
4. Take the equivalence classes of  $X \setminus C_0$  according to S.
5. The relation  $\bar{R}_\alpha = R_\alpha / S$  which is induced on the equivalent classes by  $R_\alpha$  is in order, given by

$$C_i \bar{R}_\alpha C_j \Leftrightarrow x_i R_\alpha x_j.$$

Prepare the Hasse matrix of this order, by dividing the classes into levels, while keeping track for each class of all immediate successors.

6. Draw the Hasse diagram.

본 연구의 실험을 위하여, 위의 신속 퍼지 관계 알고리즘은 파스칼 프로그래밍 언어를 사용하여 구현하였다. 구현한 신속 퍼지 관계 알고리즘은 관계 정도 구조를 추출하며, 이것은 입력 axiom에 나타나는 항(term)이나 변수의 트리 구조이다. 트리의 높은 계층의 항이나 변수는 낮은 계층보다 결론을 유도할 가능성이 높다는 것을 나타낸다. 정리증명기 OTTER는 추론의 과정에서, 낮은 가중치(weight)를 가진 절(clause)을 먼저 선택하여 추론하기 때문에, 높은 계층의 절에 보다 낮은 가중치가 부여된다.

실제로, 상기의 알고리즘을 블록 문제에 적용한 결과는 그 효율성을 잘 보여 주고 있다. 주어진 블록 문제는 초기상태가 블록 A와 블록 C가 테이블위에 있고 블록 B가 블록 C위에 있는 상태이고 목표상태는 블록 B가 테이블 위에 있고 블록 A가 블록 C위에 있는 상태이다. resolution에 근거한 계획(plan) 과정인 그린의 방법[25,26]을 이용하여, 추론규칙으로써 hyper-resolution과 UR-resolution을 사용한 실험에서, 넓이 우선 탐색시에 OTTER는 1445초, 37381 생성 절, 12455 추론 단계에 결론을 유도하였고, 디폴트 웨이트 시에는 456초, 18595 생성 절, 10716 추론 단계에 결론을 유도하였으며, 위의 알고리즘을 적용한 새로운 퍼지 가중치 부여 기법은 27초, 3251 생성 절, 1446 추론 단계에 결론에 도달하였다.

## V. 퍼지가중치 부여 및 신속 퍼지 관계 알고리즘의 자동화된 추론에의 적용

현재까지 제안된 수많은 퍼지 조건연산자[2, 3, 4]들은 각기 주어진 영역에서 우수한 결과를 보이고 있다. 그 중에서도, 6장에서 보이는 것들이 가장 널리 쓰여지는 대표적인 퍼지 조건 연산자이며, 본 연구에서는 퍼지 신속 알고리즘을 구현한 파스칼 프로그램에 그 대표적인 것들을 내재 시켜, 자동화된 추론 분야의 가중치 부여 기법 (weighting strategy)을 이용하여 그 효율성을 비교 분석한다. 위에서 설명한 퍼지 가중치 선정 기법과 신속 퍼지 알고리즘을 적용하여 추출한 관계정도구조로부터 결정한 가중치를 이용한 기법의 효율성을 증명하고, 또한 다양한 퍼지 조건 연산자(fuzzy implication operators)의 성능을 비교 분석하기 위하여 선정된 자동화된 추론의 적용 분야는 잘 알려진 인공지능의 응용 분야인 블록 문제의 네가지 경우이다(그림 3)[27].

- a. 블록문제 1: 초기상태는 블록 A는 블록 B위에 있고, 블록 B는 블록 C위에 있는 상태이며, 목표 상태는 블록 A가 테이블 위에 놓여 있는 상태이다.
- b. 블록문제 2: 초기상태는 블록 A는 블록 B위에 있고, 블록 B는 블록 C위에 있는 상태이며, 목표 상태는 블록 A와 C 모두 테이블 위에 놓여 있는 상태이다.
- c. 블록문제 3: 초기상태는 블록 A는 블록 B위에 있고, 블록 B는 블록 C위에 있는 상태이며, 목표 상태는 블록 B가 테이블 위에 놓여 있는 상태이다.
- d. 블록문제 4: 초기상태는 블록 A, B, C가 테이블위에 놓여 있는 상태이며, 목표 상태는 블록 A가 블록 B 위에 있고 블록 B가 블록 C 위에 있는 상태이다.

그림 3의 입력 문제를 정리증명기 OTTER를 이용하였을 때, 그 해결의 가속화를 위한 가중치는, 3장과 4장의 퍼지 가중치 부여 기법 및 신속 퍼지 알고리즘의 적용 후에 추출된 아래 그림 4의 관계정도구조로부터 결정한다. 관계정도구조의 높은 곳에 위치하는 항은 문제의 결론에 도달하기에 보다 더 기여할 수 있는 항으로 간주한다.

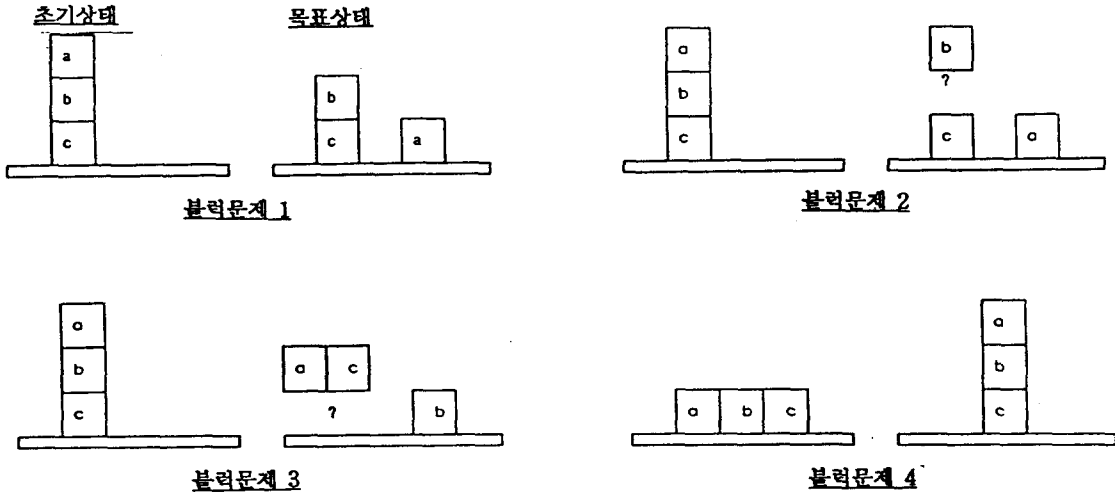
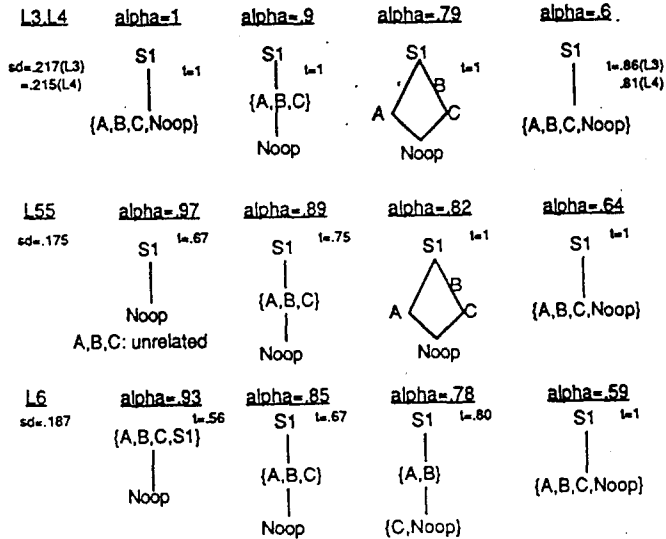
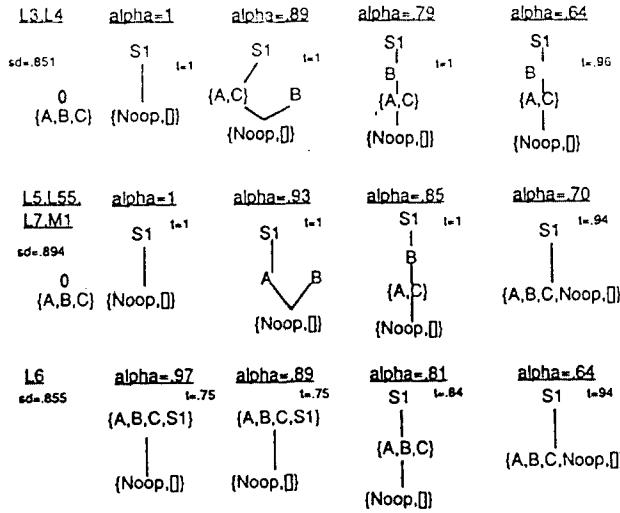


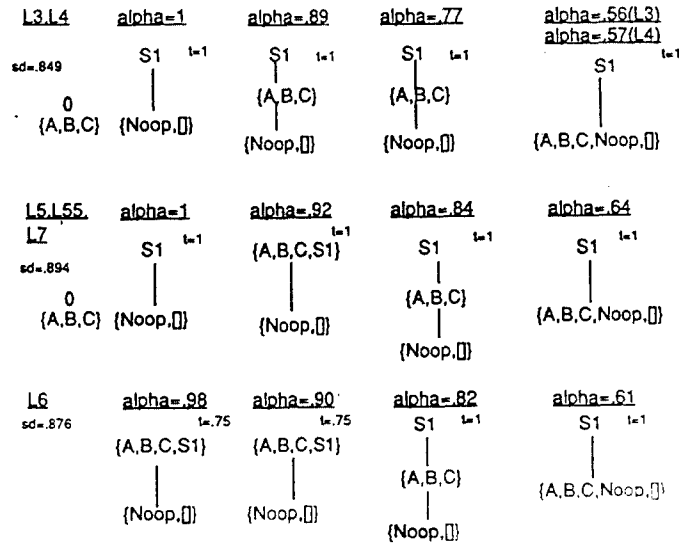
그림 3. 블럭 문제



블럭문제 1 및 2의 관계정도구조



블럭문제 3의 관계정도구조



블럭문제 4의 관계정도구조

그림 4. 블럭 문제의 관계정도구조

위 그림 4의 Hasse diagram을 추출하기 위하여 사용된 퍼지 조건 연산자는 Lukasiewicz(L5)이며, 신속 퍼지 관계 알고리즘을 구현한 파스칼 프로그램에 내재된, 널리 쓰이는 10 여종의 서로 다른 연산자들 중에서 실험의 결과로써 선택되었다. 6장에서, 퍼지 가중치 부여 기법의 효율성을 다양한 퍼지 조건 연산자를 사용하여 비교한다.



## VI. 퍼지 조건연산자(fuzzy implication operators)의 결과 비교

신속 퍼지 관계 알고리즘에서 관계성을 구하기 위해 사용된 퍼지 조건 연산자와  $\alpha$ -cut의 선택은 정리증명기의 결과에 영향을 미친다. 각 문제들은 각기 다른 조건 연산자와  $\alpha$ -cut에서 가장 우수한 결과가 나타난다. 여기서, 특정 영역에 대하여 다양한 조건 연산자와  $\alpha$ -cut을 이용한 결과를 비교하여 주어진 영역에서 가장 우수한 결과를 나타내는 퍼지 조건연산자 및  $\alpha$ -cut을 선정한다.

신속 퍼지 관계 알고리즘을 구현한 파스칼 프로그램에 내재되어, 관계 논리곱을 연산하기 위해 사용된 조건 연산자들은, L1(S# Standard Sharp), L2(S Standard Strict), L3(S\* Standard Star), L4(G43 Gaines 43), L5(L Lukasiewicz), L55(KDL Kleene-Dienes Lukasiewicz), L6(KD Kleene-Dienes), L7(EZ Early Zadeh), L8(W Willmott) 등이다.

L1 : S# Standard Sharp

$$a \rightarrow_1 b = \begin{cases} 1 & \text{iff } a \neq 1 \text{ or } b = 1 \\ 0 & \text{otherwise.} \end{cases}$$

L2 : S Standard Strict

$$a \rightarrow_2 b = \begin{cases} 1 & \text{iff } a \leq b \\ 0 & \text{otherwise.} \end{cases}$$

L3 : S\* Standard Star

$$a \rightarrow_3 b = \begin{cases} 1 & \text{iff } a \neq 1 \text{ or } b = 1 \\ b & \text{otherwise.} \end{cases}$$

L4 : G43 Gaines 43

$$a \rightarrow_4 b = \min \left( 1, \frac{b}{a} \right).$$

4'. G43' Modified Gaines 43

$$a \rightarrow_{4'} b = \min \left( 1, \frac{b}{a}, \frac{1-a}{1-b} \right).$$

L5 : L Lukasiewicz

$$a \rightarrow_5 b = \min(1, 1 - a + b)$$

L5.5 : KDL Kleene-Dienes Lukasiewicz

$$a \rightarrow_{5.5} b = 1 - a + ab.$$

L6 : KD Kleene-Dienes

$$a \rightarrow_7 b = (1-a) \vee b.$$

L7: EZ Early Zadeh

$$\begin{aligned} a \rightarrow_7 b &= (a \wedge b) \vee (1-a) \\ &= (a \rightarrow_6 b) \wedge ka. \end{aligned}$$

L8: W Willmott

$$\begin{aligned} a \rightarrow_8 b &= ((1-a) \vee b) \wedge (a \wedge (1-b) \vee (b \wedge (1-a))) \\ &= (a \rightarrow_7 b) \wedge kb \\ &= (a \rightarrow_6 b) \wedge ka \wedge kb \end{aligned}$$

여기서,  $a \wedge b = \min(a, b)$ ,  
 $a \vee b = \max(a, b)$ ,  
 $ka = \max(a, 1-a)$ .

아래의 표에서 보인 것은 생성된 절들의 수와 결론이 유도되기 까지의 kept clause들의 수, 그리고 괄호 속의 수는 정리증명을 완료하는데 소요된 CPU 시간을 나타낸다. 이 CPU시간은 정리증명기 OTTER의 실험이 수행된 UNIX하의 SUN sparc station에서 측정된 시간이다.

입력 문제들을 위한 관계 정도 구조는 앞 장에 나타나 있으며, 그 문제들에 대한 절 형태로 변환된 axiom 리스트는 부록에 첨부한다.

- 1. 블럭 문제 1-초기 상태: 블럭 A는 블럭 B 위에, 블럭 B는 블럭 C 위에 놓여 있다.  
 목표 상태: 블럭 A가 테이블 위에 놓여 있다.

$\alpha$ -cut	L2	L3	L4	L5	L55	L6	L7	L8	M1	M2
height 1	105	105	105	105	56	56	56	92	105	92
	70	70	70	70	41	41	41	61	70	61
	(.48)	(.48)	(.48)	(.48)	(.36)	(.36)	(.36)	(.48)	(.48)	(.48)
half-upper 0.93	69	69	69	69	69	69	56	92	69	167
	50	50	50	50	50	50	41	61	50	108
	(.42)	(.42)	(.42)	(.42)	(.42)	(.42)	(.36)	(.48)	(.42)	(.88)
mean 0.87	69	95	95	69	95	73	43	105	69	105
	50	68	68	50	68	52	32	70	50	70
	(.44)	(.60)	(.60)	(.44)	(.60)	(.42)	(.38)	(.48)	(.44)	(.48)
half-lower 0.70	105	105	105	105	105	105	105	105	105	56
	70	70	70	70	70	70	70	70	70	70
	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.36)

추론 규칙 hyper-resolution과 UR-resolution이 동시에 사용되었다. 비교하면, 디폴트 웨이팅에서는 생성된 절이 92, 선택된 절의 수는 61, CPU시간은 .48 초만에 결론을 얻었는데 반해서 넓이 우선 탐색에서는 생성된 절이 85, 선택된 절은 62, CUP 시간은 .54초에 도달하였다. 입력 절의 수는 20이다.

2. 블럭 문제 2-초기 상태 :블럭 A는 블럭 B 위에, 블럭 B는 블럭 C 위에 놓여 있다.  
 목표 상태 :블럭 A와 C 모두 테이블 위에 놓여 있다.

$\alpha$ -cut	L2	L3	L4	L5	L55	L6	L7	L8	M1	M2
height 1	105	105	105	105	69	69	69	105	105	105
	70	70	70	70	50	50	50	70	70	70
	(.48)	(.48)	(.48)	(.48)	(.40)	(.40)	(.40)	(.50)	(.48)	(.48)
half-upper 0.93	69	69	69	69	69	69	69	105	69	167
	50	50	50	50	50	50	50	70	50	108
	(.42)	(.42)	(.42)	(.42)	(.42)	(.42)	(.42)	(.50)	(.42)	(.90)
mean 0.87	72	95	95	72	95	76	76	105	72	105
	51	68	68	51	68	53	53	70	51	70
	(.50)	(.62)	(.62)	(.50)	(.62)	(.44)	(.46)	(.48)	(.50)	(.48)
half-lower 0.70	105	105	105	105	105	105	105	105	105	59
	70	70	70	70	70	70	70	70	70	42
	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.48)	(.40)

추론 규칙 hyper-resolution과 UR-resolution이 동시에 사용되었다. 비교하면, 디폴트 웨이팅에서는 생성된 절이 105, 선택된 절의 수는 70, CPU시간은 .50 초만에 결론을 얻었는데 반해서 넓이 우선 탐색에서는 생성된 절이 119, 선택된 절은 88, CUP 시간은 .72 초에 도달하였다. 입력 절의 수는 20이다.

3. 블럭 문제 3-초기 상태 :블럭 A는 블럭 B 위에, 블럭 B는 블럭 C 위에 놓여 있다.  
 목표 상태 :블럭 B가 테이블 위에 놓여 있다.

$\alpha$ -cut	L2	L3	L4	L5	L55	L6	L7	L8	M1	M2
height 1	1597	1597	1597	1597	1597	356	1597	356	1597	
	726	726	726	726	726	125	726	125	726	
	(7.72)	(7.72)	(7.72)	(7.72)	(7.72)	(1.48)	(7.72)	(1.48)	(7.72)	
half-upper 0.93	389	389	389	389	389	356	389	356	389	9305
	132	132	132	132	132	125	132	125	132	4086
	(1.60)	(1.60)	(1.60)	(1.60)	(1.60)	(1.48)	(1.60)	(1.46)	(1.60)	(100)
mean 0.85	573	573	573	573	573	389	573	1958	573	2367
	173	173	173	173	173	132	173	854	173	1051
	(2.58)	(2.58)	(2.58)	(2.58)	(2.58)	(1.60)	(2.58)	(9.90)	(2.58)	(12)
half-lower 0.70	1597	659	659	1597	1597	1597	1597	1597	1597	356
	726	194	194	726	726	726	726	726	726	125
	(7.72)	(2.70)	(2.70)	(7.72)	(7.72)	(7.72)	(7.72)	(7.72)	(7.72)	(1.48)

추론 규칙 hyper-resolution과 UR-resolution이 동시에 사용되었다. 비교하면, 디폴트 웨이팅에서는 생성된 절이 1478, 선택된 절의 수는 671, CPU시간은 6.42 초만에 결론을 얻었는데 반해서 넓이 우선 탐색에서는 생성된 절이 1393, 선택된 절은 637, CUP 시간은 7.44 초에 도달하였다. 입력 절의 수는 24이다.

4. 블럭 문제 4-초기 상태 :블럭 A, B, C는 table상에 있다.  
 목표 상태 :블럭 B는 블럭 C위에 있고, 블럭 A는 블럭 B위에 있다.

추론 규칙 hyper-resolution과 UR-resolution이 동시에 사용되었다. 비교하면, 디폴트 웨이팅에서는 생성된 절이 18595, 선택된 절의 수는 10716, CPU시간은 456 초만에 결론을 얻었는데 반해서 넓이 우선 탐색에서는 생성된 절이 37381, 선택된 절은 12455, CUP 시간은 1445 초에 도달하였다. 입력 절의 수는 30이다.

$\alpha$ -cut	L2	L3	L4	L5	L55	L6	L7	L8	M1	M2
height 1	27865	27865	27865	27865	27865	3251	27865	18595	27865	18959
	16178	16178	16178	16178	16178	1446	16178	10716	16178	10716
	(984)	(984)	(984)	(984)	(984)	(27)	(984)	(456)	(984)	(984)
half-upper 0.92	6011	6011	6011	3251	3251	3251	3251	18595	3251	18595
	2696	2696	2696	1446	1446	1446	1446	10716	1446	10716
	(64)	(64)	(94)	(27)	(27)	(27)	(27)	(456)	(27)	(984)
mean 0.84	6011	6011	6011	6011	6011	6011	6011	3251	6011	27865
	2696	2696	2696	2696	2696	2696	2696	1446	2696	16178
	(64)	(64)	(64)	(64)	(64)	(64)	(64)	(27)	(64)	(984)
half-lower 0.64	27865	27865	27865	27865	27865	27865	27865	27865	27865	3251
	16178	16178	16178	16178	16178	16178	16178	16178	16178	1446
	(984)	(984)	(984)	(984)	(984)	(984)	(984)	(984)	(984)	(27)

## Ⅶ. 결 론

퍼지 정보 검색 기법이 자동화된 추론(Automated Reasoning)의 가중치 선정에 응용되어, 블록 문제에서 추론 시간 및 공간을 감소시키는 예를 보였다. 퍼지 신속 관계 알고리즘의 파스칼 프로그래밍 언어로의 구현은, 퍼지 가중치 부여 기법으로써 주어진 입력 문제로 부터 분석되어 추출된 출력 행렬로부터, 가중치의 우선 순위를 나타내는 관계 정도구조를 자동적으로 구성하였다.

특정 분야에 대한 가장 적절한 퍼지 조건 연산자와  $\alpha$ -cut의 선택을 위하여서는 많은 실험이 행해져야 한다. 특정 분야에 대하여 실험을 거쳐 선택된 퍼지 조건 연산자와  $\alpha$ -cut은 그 분야에 대한 퍼지 조건 연산자와  $\alpha$ -cut의 선택에 대한 비용을 절감할 수 있게 한다. 최적의 퍼지 조건 연산자와  $\alpha$ -cut의 선택에 대한 실험에서 퍼지 조건 연산자 Lukasiewicz(L5), KDL Kleene-Dienes Lukasiewicz(L5.5), KD Kleene-Dienes(L6)와  $\alpha$ -cut은 upper-half 영역에서 다른 것에 비해 비교적 우수한 결과를 산출했다. 그러므로, 완전히 새로운 영역에 대한 실험에서 퍼지 조건 연산자를 선택할때는 조건 연산자 L5, L5.5, L6에 높은 우선 순위를 주어 실험하는 것이 유리하다.

퍼지 정보 검색 기법을 이용한 가중치 선정 전략을 자동화된 추론 분야에 적용하여, 퍼지 조건 연산자의 효율성을 비교하였다. 실제의 퍼지 조건연산자는 주어진 영역에 따라 각기 다른 우수성을 보이고 있으므로, 다른 분야에 의 퍼지 기법의 적용은 수 많은 실험적 연구를 통하여 선정하여야 한다.

## 부 록

1. 블록 문제, 초기 상태 : 블록 A, B, C가 테이블위에 있는 상태  
 목표 상태 : 블록 A가 블록 B 위에 있고 블록 B가 블록 C 위에 있는 상태

```
weight_list(pick_and_purge).
weight(S1, 1).
weight(A, 1).
weight(B, 1).
```

weight(C, 1).  
 weight(Noop, 20).  
 weight([ ], 20).  
 end\_of\_list.

list(axioms).  
 - T(On(x, y), z) | -T(Clear(x), z) | T(Table(x), Do(U(x, y), z)).  
 - T(On(x, y), z) | -T(Clear(x), z) | T(Clear(y), Do(U(x, y), z)).  
 - T(Table(x), z) | -T(Clear(x), z) | -T(Clear(y), z) | (x=y) | T(On(x, y), Do(S(x, y), z)).  
 - T(Table(u), z) | T(Table(u), Do(U(x, y), z)).  
 - T(Clear(u), z) | T(Clear(u), Do(U(x, y), z)).  
 - T(On(u, v), z) | (u=x) | T(On(u, v), Do(U(x, y), z)).  
 - T(Table(u), z) | (u=x) | T(Table(u), Do(S(x, y), z)).  
 - T(Clear(u), z) | (u=y) | T(Clear(u), Do(S(x, y), z)).  
 - T(On(u, v), z) | T(On(u, v), Do(S(x, y), z)).  
 - T(x, z) | T(x, Do(Noop, z)).  
 - T(x, z) | T(x, Do([ ], z)).  
 - T(x, Do(w, Do(y, z))) | T(x, Do([y|w], z)).  
 - T(Table(x), z) | -T(On(x, y), z).  
 - T(Clear(y), z) | -T(On(x, y), z).  
 - T(On(x, y), z) | -(y!=w) | -T(On(x, w), z).  
 - T(On(A, B), x) | -T(On(B, C), x) | Goal(x).  
 - Goal(Do(x, y)) | Ans(x).  
 - (A=B).  
 - (B=C).  
 - (A=C).  
 - (B=A).  
 - (C=A).  
 - (C=B).  
 end\_of\_list.

list(sos).  
 T(Table(A), S1).  
 T(Table(B), S1).  
 T(Table(C), S1).  
 T(Clear(A), S1).  
 T(Clear(B), S1).  
 T(Clear(C), S1).  
 - Ans([S(B, C)|S(A, B)]).  
 end\_of\_list.

참 고 문 헌

1. Zimmermann, H.J. *Fuzzy Set Theory-and Its Applications, 2nd ed.* Kluwer Academic, 1991.
2. Mizumoto, M., and Zimmermann, H.J. Comparison of Fuzzy Reasoning Methods. *Fuzzy Sets and Systems*, 8, 253-283, 1982.
3. Bandler, W., and Kohout, L.J. Fuzzy Power Sets and Fuzzy Implication Operators. *Fuzzy Sets and Systems*, 4, 13-30, 1980.
4. Cao, Z., and Kandel, A. Applicability of some Fuzzy Implication Operators. *Fuzzy Sets and Systems*, 31, 151-186, 1989.
5. Lusk, E. L., and Overbeek, R. *The Automated Reasoning System ITP*. Argonne National Laboratory, Argonne, IL, 1984.
6. McCune, W.W. OTTER 2.0 Users Guide *In Technical Report ANL-90/9*, Argonne National Laboratory, Argonne, IL, March 1990.
7. McCharen, J., Overbeek R. and Wos, L. Complexity and related enhancements for automated theorem proving programs. *In Computers and Mathematics with Applications 2*, pages 1-16, 1976.
8. Kohout, L.J. and Kim, Yong-Gi. Generating control strategies for resolution-based theorem provers by means of fuzzy relational products and relational closures. In B. Lowen and M. Roubens, editors, *Fuzzy Logic: State of the Art*. Kluwer Academic, Boston and Dordrecht, 1993.
9. Kim, Yong-Gi and Kohout, L.J. Use of Fuzzy Relational Products and Algorithms for generating Control strategies in resolution based Automated Reasoning, *Proceedings of the fourth International Fuzzy System Association(IFSA) world congress*, (Brussels, Belgium), July 7-12, 1991.
10. Kim, Yong-Gi and Kohout, L.J. An Improvement of Weighting Strategy in Resolution Based Automated Reasoning, *Proceeding of the sixth International Symposium on Methodologies for Intelligent Systems (ISMIS'91)* Published by Oak Ridge National Laboratory(12 pages), Charlotte, NC, October 16-19, 1991.
11. 김용기. 퍼지기법을 이용한 자동화된 추론의 가중치 부여기법 개선. 정보과학회논문지, 제21권 제10호, page 1947-1957, 1994.
12. Bandler, W. and Kohout, L.J. Fuzzy relational products as a tool for analysis and synthesis of the behaviour of complex natural and artificial system. In : Wang P P, Chang S K (eds.) *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*. Plenum, New York, pp. 341-67.
13. Bandler, W. and Kohout, L.J. Semantics of implication operators and fuzzy relational products *Int. J. Man-Machine Studies*, 12, 89-116, 1980.
14. Kohout, L.J., Keravnou E. and Bandler, W. Automatic documentary information retrieval by means of fuzzy relational products. In Gaines, B.R., Zadeh L.A. and Zimmermann, H.-J., editors, *Fuzzy Sets in Decision Analysis*, pages 308-404, North-Holland, Amsterdam, 1984.
15. Bandler, W. and Kohout, L.J. Special properties, closures and interiors of crisp and fuzzy relations. *In Fuzzy Sets and Systems*, pages 26(3) : 317-332, June 1988.
16. Kohout, L.J. and Bandler, W. The use of information retrieval technique in construction of multi-center knowledge based systems. In Bouchon, B. and Yager, R.R., editors, *Uncertainty in Knowledge-Based Systems*, Springer-Verlag, Berlin, 1987.
17. Bandler, W. and Kohout, L.J. Fuzzy power sets and fuzzy implication operators. In Wang, P.P. and Chang, S.K., editors, *Fuzzy sets and systems*, pages 13-30, plenum press, New York and London, 1980(4).
18. Bandler, W. and Kohout, L.J. Hierachies in symptoms and patients through computation of fuzzy tri-

- angle products and closures. In Parslow, editor, *BST'81 Information technology for the eighties*, Heyden & Son LTD., 1981.
19. Bandler, Wyllis and Kohout, Ladislav J. A survey of fuzzy relational products in their applicability to medicine and clinical psychology. In Kohout, L.J. and Bandler, W., editors, *Knowledge Representation in Medicine and Clinical Behavioural Science*, Abacus Press, Cambridge MA, USA, and Tunbridge Wells, UK., 1986.
  20. Bandler, W. and Kohout, L.J. Special properties, closures and interiors of crisp and fuzzy relations. In *Fuzzy Sets and Systems*, pages 26(3) : 317-332, June 1988.
  21. Kohout, L. J. and Kallala, M. Evaluator of neurological patients' dexterity based on relational fuzzy products. In proc. of Second Expert Systems International Conference(London, October 1986), pages 1-12. Learned Information Inc., NJ, USA and Oxford, UK, 1986.
  22. Kohout, L.J. and Kim, Yong-Gi. Use of fuzzy information retrieval techniques for generating control strategies in resolution-based automated reasoning. In *FLAIRS-90 Proceedings*, pages 140-144, The Florida Artificial Intelligence Research Society, 1990.
  23. Wos, L., Overbeek R. Lusk E. and Boyle, J. *Automated Reasoning: Introduction and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1894.
  24. Wos, L., Carson, D., Robinson, G. Efficiency and Completeness of the set-of-support strategy in theorem proving. In *Journal of Assoc. for Comput. Mach.*, 12, pages 536-541, 1965.
  25. Green, C. Theorem-proving by resolution as a basic for question-answering systems. In Meltzer and Michie, editors, *Mah. Intell. 4*, Edinburgh University Press, pages 183-205, Edinburgh, 1969.
  26. Green, C., and Raphael, B. The use of theorem-proving techniques in question-answering systems. In *Proc. of the 23rd National Conf. of ACM*, pages 169-181, Brandon System Press, Princeton, 1968.
  27. Genesereth, M., and Nilsson N. *Logical Foundations of Artificial intelligence*, Morgan Kaufmann Publisher, Inc., Los Altos, CA, 1987.