

퍼지 Pr/T 네트를 기반으로 하는 술어논리 수준의 지식표현과 퍼지추론

Knowledge Representation and Fuzzy Reasoning in the Level of Predicate Logic based on Fuzzy Pr/T Nets

조 상 엽* 이 동 은**
Sang-Yeop Cho Dong-Eun Lee

요 약

본 논문에서는 지식기반 시스템의 퍼지 생성규칙을 술어논리 수준에서 표현할 수 있는 퍼지 Pr/T 네트를 제안한다. 퍼지 Pr/T 네트는 Pr/T 네트의 퍼지 확장이다. 퍼지 Pr/T 네트를 기반으로 퍼지추론 알고리즘을 제안한다. 퍼지추론 알고리즘은 퍼지 생성규칙에서 퍼지개념에 따라 적절한 믿음값 함수를 사용하므로 다른 방법보다 더 사람의 직관과 추론에 가깝다.

Abstract

This paper presents fuzzy Pr/T nets to represent the fuzzy production rules of a knowledge-based system in the level of first-order predicate logic. The fuzzy Pr/T nets are fuzzy extension of the Pr/T nets. Based on the fuzzy Pr/T net, we propose a fuzzy reasoning algorithm. This algorithm is much closer to human intuition and reasoning than other methods because of using the proper belief functions according to fuzzy concepts in fuzzy production rules.

1. 서 론

지식기반 시스템에서 지식을 표현할 때 주로 사용하는 것이 생성규칙이다[3]. 그러나 생성규칙은 사람이 가지고 있는 불확실한 지식을 표현하고 처리하는 데는 적절하지 못하다. 불확실한 지식을 처리하기 위한 이론적인 기반을 제공하는 것이 퍼지이론이다. 퍼지이론을 이용하여 생성규칙에 있는 불확실한 지식을 표현한 생성규칙을 퍼지 생성규칙 이라고 한다[8][10].

퍼지 생성규칙과 퍼지 추론과정을 모형화하는 도구로 많이 사용되는 것이 퍼지 페트리네트이다. 퍼지 페트리네트(fuzzy Petri net)는 페트리네트(Petri net)[13][15]에 불확실함을 처리할 수 있도록 퍼지

이론을 결합하여 확장한 정보흐름 분석을 위한 모형화 도구이다. 퍼지 페트리네트를 지식표현과 추론에 이용한 연구로는 [1][2][4][6][11][12][16] 등이 있다.

퍼지 페트리네트를 이용하여 퍼지 생성규칙을 표현하는 접근방법은 페트리네트에 대응되는 명제논리 수준의 표현법이다. 그러나 대부분의 지식기반 시스템에서 사용하는 규칙은 술어논리에 논리적 기반을 두고 있다. 본 논문에서는 술어논리 수준에 대응되는 Pr/T 네트[7]의 퍼지 확장인 퍼지 Pr/T 네트를 이용하여 퍼지 생성규칙을 표현하는 방법과 이 표현에서 사용할 수 있는 퍼지추론 알고리즘을 제안한다. 이 알고리즘은 페트리네트를 기반으로 개발된 알고리즘[9][14]을 퍼지 확장하였고 기존의 알고리즘[17] 보다 사람의 직관과 추론에 가까운 믿음값 평가함수를 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구와 연구배경을 기술한다. 3장에서는 퍼지

* 종신회원 : 청운대학교 인터넷컴퓨터학과 교수
sycho@cwunet.ac.kr

** 비 회 원 : 청운대학교 인터넷컴퓨터학과 교수
delec@cwunet.ac.kr

생성규칙의 분류와 퍼지 생성규칙에 대응하는 퍼지 PrT 네트 표현을 설명하고 믿음값 평가함수를 기술한다. 4장에서는 퍼지 PrT 네트에 사용할 퍼지추론 알고리즘을 기술한다. 5장에서는 본 논문의 결과를 기술한다.

2. 관련연구 및 연구배경

생성규칙을 모형화 할 때 그래프를 많이 사용한다. 여러 가지의 그래프 중에서 Petri 네트[13][15]가 규칙의 전제부와 결론부를 자연스럽게 표현할 수 있어 많이 사용된다. 이러한 Petri 네트는 명제논리 수준의 규칙을 모형화하는 데 유용하다. 그러나 실 세계에서 사용하는 규칙기반 시스템에서 사용하는 규칙은 술어논리에 그 기반을 두고 있다. 그러므로 이러한 규칙을 모형화할 때에는 Petri 네트를 술어논리 수준으로 확장한 PrT 네트[7]가 사용된다.

실세계의 불확실한 지식을 표현하기 위해서는 기존의 생성규칙으로는 적절하게 표현하기가 어렵다. 이러한 불확실함을 처리하는데 이론적인 기반을 제공하는 것이 퍼지이론이다. 퍼지개념이 포함된 규칙을 퍼지 생성규칙이라고 한다. 이러한 퍼지규칙을 모형화하기 위해서 사용하는 Petri 네트와 PrT 네트도 퍼지 확장하여야 한다. 이러한 확장의 결과가 퍼지 Petri 네트[11]와 퍼지 PrT 네트[17]이다.

본 논문에서는 기존의 연구와는 달리 규칙을 술어논리 수준의 퍼지 PrT 네트를 이용하여 모형화하고 이를 기반으로 추론 알고리즘을 제안한다. 비퍼지한 PrT 네트를 이용하여 규칙을 모형화할 경우에는 믿음값의 계산이 필요없다[9][14]. 퍼지 PrT 네트를 사용할 경우에는 믿음값을 계산하는 방법이 필요하다. 퍼지이론에서 사용하는 대표적인 방법인 min과 max를 사용하는 방법을 사용하는 Sheng-Ke Y. [17]가 있다. 그러나 이 방법은 계산하는 과정은 단순하고 간단하지만 사람이 사용하는 방법과는 차이가 있다. 본 논문에서는 규칙의 전제부와 결론부에 퍼지개념의 유무에 따라 유사도나 퍼지관계, 특히 R_{sg} 를 이용하여 믿음값

을 평가하기 때문에 기존의 방법보다 사람이 사용하는 추론과 직관에 더 유사하다[5][8].

3. 지식표현과 추론

3.1 퍼지 생성규칙

실세계의 불확실하고 애매한 지식은 퍼지 생성규칙을 이용하면 적절하게 기술할 수 있다. 퍼지 생성규칙은 두 개의 술어논리 공식 사이의 퍼지 관계를 기술한다고 생각할 수 있다. 가장 간단한 퍼지 생성규칙의 형식은 다음과 같다.

$$\text{Rule: } d_j \Rightarrow d_k \text{ (Bel} = \beta_j) \quad (\text{식 1})$$

여기에서 d_j 와 d_k 는 각각 퍼지변수를 포함할 수 있는 퍼지술어 $d_j(x_1, x_2, \dots, x_m)$ 와 $d_k(y_1, y_2, \dots, y_n)$ 를 표현한다. m 과 n 은 양의 정수이다. x_p 와 y_q 는 같을 수도 있고 다를 수도 있으며, 변수이거나 상수이다($p=1, 2, \dots, m, q=1, 2, \dots, n$). β_j 는 규칙의 믿음의 강도를 표시하는 믿음값이다($\beta_j \in [0,1]$).

λ 를 임계값이라고 하자. $\lambda \in [0,1]$. $d_j(x_1, x_2, \dots, x_m)$ 의 어떤 인스턴스의 믿음값이 $\beta_j \in [0,1]$ 이다. $\beta_j \geq \lambda$ 라면 식 1은 실행가능하고, $\beta_j < \lambda$ 라면 식 1은 실행되지 않는다.

퍼지 생성규칙을 많이 사용하는 분야로는 제어 분야와 지식공학 분야가 있다. 각 분야에서 사용하는 퍼지 생성규칙의 분류는 전명근 et al. [1], 조상엽 et al. [2], Chen et al. [4], Looney et al. [10] 등에서 찾아 볼 수 있다. [17]은 제어 분야에서 사용하는 규칙을 [10]에 따라 분류하여 PrT 네트로 표현하였고, 본 논문에서는 지식공학에서 사용하는 규칙을 논리적으로 분류하여 PrT 네트로 표현하였다. 본 논문에서는 지식공학 분야에서 사용하는 퍼지 생성규칙을 논리연결자의 유무에 따라 표 1과 같이 아홉 가지의 형으로 논리적인 구분을 하였다. 1형은 단순 퍼지 생성규칙이라고 하고, 2형에서 9형은 합성 퍼지 생성규칙이라고 한다.

(표 1) 퍼지 생성규칙의 분류

형	퍼지 생성규칙
1	$d_j \Rightarrow d_k (\beta_i)$
2	$d_j \Rightarrow d_{k1} \wedge d_{k2} \wedge \dots \wedge d_{kn} (\beta_i)$
3	$d_{j1} \wedge d_{j2} \wedge \dots \wedge d_{jm} \Rightarrow d_k (\beta_i)$
4	$d_{j1} \wedge d_{j2} \wedge \dots \wedge d_{jm} \Rightarrow d_{k1} \wedge d_{k2} \wedge \dots \wedge d_{kn} (\beta_i)$
5	$d_{j1} \vee d_{j2} \vee \dots \vee d_{jm} \Rightarrow d_k (\beta_i)$
6	$d_{j1} \vee d_{j2} \vee \dots \vee d_{jm} \Rightarrow d_{k1} \wedge d_{k2} \wedge \dots \wedge d_{kn} (\beta_i)$
7	$d_j \Rightarrow d_{k1} \vee d_{k2} \vee \dots \vee d_{kn} (\beta_i)$
8	$d_{j1} \wedge d_{j2} \wedge \dots \wedge d_{jm} \Rightarrow d_{k1} \vee d_{k2} \vee \dots \vee d_{kn} (\beta_i)$
9	$d_{j1} \vee d_{j2} \vee \dots \vee d_{jm} \Rightarrow d_{k1} \vee d_{k2} \vee \dots \vee d_{kn} (\beta_i)$

본 연구에서는 아홉 가지의 퍼지 생성규칙 중에서 1형에서 4형까지 만을 사용한다. 2형과 4형은 1형과 3형으로 축소할 수 있지만 퍼지 생성규칙의 표현성을 위해 그대로 사용한다. 5형과 6형은 퍼지 생성규칙을 다루기 쉽게 하기 위해 1형과 2형으로 각각 축소하여 표현한다. 7형, 8형 그리고 9형은 의미있는 연역추론을 할 수 없으므로 사용하지 않는다.

3.2 퍼지추론

퍼지 생성규칙을 이용한 퍼지추론을 할 때에는 퍼지 합성추론을 사용한다. 퍼지 합성추론이 적용될 때 《결론》의 믿음값을 계산하기 위한 함수들이 사용된다. [17]은 min과 max로만 믿음값 계산을 하나, 본 논문은 퍼지개념의 유무에 따라 유사도와 R_{sg} 를 이용하여 믿음값을 계산한다. 여기에서는 퍼지 합성추론과 믿음값 함수에 대하여 기술한다[3][5][8].

퍼지 합성추론 : 퍼지변수를 가지고 있는 퍼지 생성규칙을 이용하여 추론을 하기 위해 개발된 추론규칙이 퍼지 합성추론이다. 퍼지 합성추론의 형식은 다음과 같다.

《규칙》 : $x \text{ is } A \Rightarrow y \text{ is } B \quad (\beta_r)$

《사실》 : $x \text{ is } A' \quad (\beta_f)$

《결론》 : $y \text{ is } B' \quad (\beta_c) \quad (\text{식 } 2)$

식 2에서 A, A', B 그리고 B'는 퍼지 집합이거나 비퍼지 집합이다. β_r , β_f 그리고 β_c 는 각각 《규칙》, 《사실》 그리고 《결론》의 믿음값이다. β_c 는 β_r 과 β_f 을 이용하여 구한다. β_c 를 구하기 위해 사용하는 함수가 믿음값 함수이다.

- 믿음값 평가함수 : 단순 퍼지 생성규칙의 믿음값을 계산할 때 사용하는 믿음값 함수 $\beta: (\beta_r, \beta_f) \rightarrow \beta_c$ 이다. 규칙의 전제부가 비퍼지하면 $\beta_c = \beta_f * \beta_r$ 를 사용하고, 전제부가 퍼지하고 결론부가 비퍼지하면 $\beta_c = \beta_f * \beta_r * S$ 를 사용한다. S는 유사도(similarity)이다. 전제부와 결론부가 모두 퍼지하면 $\beta_c = \beta_f * R_{\text{sg}}$ 를 사용한다. R_{sg} 는 인간의 추론과정을 가장 적절히 표현하는 퍼지관계이다[4][5].
- 믿음값 합성함수 : 퍼지 합성추론이 합성 퍼지 생성규칙에 적용될 때에는 믿음값 평가함수로 결론의 믿음값을 평가할 수 없다. 이때 믿음값을 계산하기 위해 다음과 같은 믿음값 합성함수를 사용한다.

$$\begin{aligned} \beta_c &= \beta_{\text{comp}}(\beta(\beta_r, \beta_f)) \\ &= \min(\max(\beta(\beta_{fp}, \beta_{rpq}))) \end{aligned}$$

여기에서 $p=1, 2, \dots, m$, $q = 1, 2, \dots, n$. m 과 n 은 각각 전제부와 결론부에 있는 퍼지명제의 수이다.

- 믿음값 결합함수 : 퍼지 생성규칙의 5형과 6형의 축소된 1형과 2형은 퍼지 추론시 서로 다른 추론통로를 통해 같은 노드, 즉 같은 결론에 도달할 수가 있다. 이러한 노드에서는 같은 결론이 두 개 이상의 서로 다른 믿음값을 갖게된다. 이러한 경우에 결론의 믿음값을 다시 계산하기 위해 사용하는 함수가 믿음값 결합함수이다.

$$\begin{aligned} \beta_c &= \beta_{\text{comb}}(\beta_c, \beta_c^{\text{old}}) \\ &= \max(\beta_c, \beta_c^{\text{old}}) \end{aligned}$$

여기에서 β_c^{old} 는 이미 추론통로를 통해 도달한

결론에 대한 믿음값이고, β_c 는 다른 추론통로를 통해서 도달한 또 다른 결론의 믿음값이다.

3.3 퍼지 Pr/T 네트

지식기반 시스템의 퍼지 생성규칙을 표현하기 위해 다음과 같은 형식을 갖는 FPN(fuzzy Pr/T net)을 정의한다.

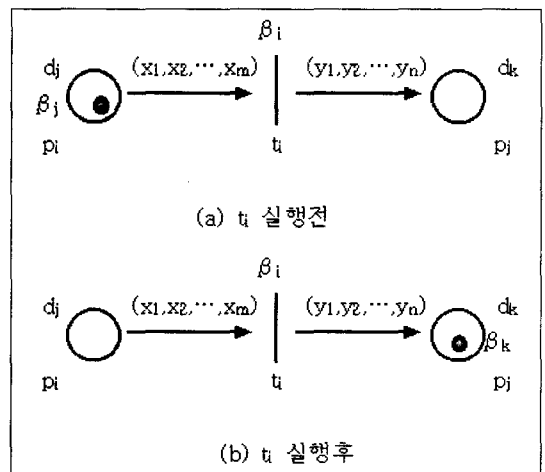
$$FPN = (P, T, F, D, V, \pi, A_P, A_T, A_F, f, fs, fb, \alpha, M_0)$$

여기에서 P와 T는 각각 플레이스와 트랜지션의 유한집합이다. $P \cup T \neq \emptyset, P \cap T = \emptyset$. F는 흐름관계의 유한집합이다. $F \subseteq (P \times T) \cup (T \times P)$, $dom(F) \cup ran(F) = P \cup T$; D는 개체(individual)집합이라고 불리는 공집합이 아닌 유한집합이다. V는 D 상의 변수의 집합이다. π 는 D 상의 술어의 집합이다. $A_P: P \rightarrow \pi$, A_P 는 전단사사상이다. $\forall p \in P$ 에 대해 만일 $A_P(p)$ 가 n-항 술어라면 p는 n-항 술어라고 부른다. $A_T: T \rightarrow f$, f 는 D 상의 공식집합이다. $\forall t \in T$ 에 대해 만일 $A_T(t)$ 는 유일한 술어가 되거나 연산자가 될 수 있다. $A_F: F \rightarrow fs$, fs 는 D 상의 기호합 집합(symbolic sum set). $\forall p \in P$ 에 대해 만일 $(t, p) \in F$ 또는 $(p, t) \in F$ 라면 $A_F(t, p)$ 또는 $A_F(p, t)$ 는 n-항 기호합(symbolic sum)이고, 그렇지 않으면 $A_F(t, p)$ 또는 $A_F(p, t)$ 는 널이다. $\forall t \in T$ 에 대해서 $A_T(t)$ 안의 자유변수(free variable)들은 방향성 아크의 양 끝 중 하나가 t를 갖는 이 아크에서 단지 한번만 발생한다. $f: T \rightarrow [0, 1]$ 는 결합함수이다. $\forall t \in T$ 에 대해 $f(t) = \beta_t$ 이다. 여기에서 β_t 는 대응되는 생성 규칙의 믿음강도이고 $\beta_t \in [0, 1]$ 이다. $\alpha: P \rightarrow [0, 1]$ 는 결합함수이다. $\forall p \in P$, $\alpha(p)$ 는 술어 p의 모든 인스턴스의 믿음값을 나타낸다. p 내의 어느 토큰 tok_k 에 대해, $\alpha'(p(tok_k)) \in \alpha(p)$ 이다. 여기에서 $\alpha'(p(tok_k))$ 는 p의 인스턴스 tok_k 의 믿음값을 표시한다. $M_0: p \rightarrow N$ 는 마킹함수이다. N은 음이 아닌 양

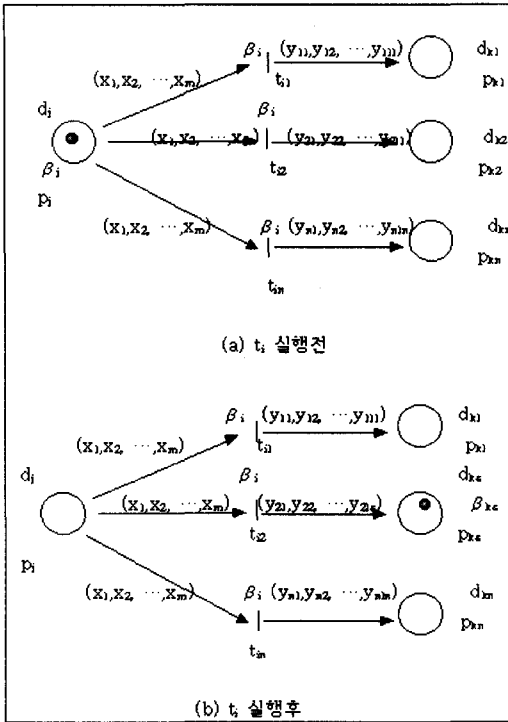
의 정수이다[17].

- 트랜지션의 실행 : 퍼지 Pr/T 네트에서 트랜지션의 실행 조건은 다음과 같다. (1) 만일 $\forall p \in \bullet t, M(p) \geq 1$ 이라면 트랜지션 t는 M에서 실행 가능하다(enable). (2) 만일 트랜지션 t가 M에서 실행 가능하다면 t의 실행은 M을 새로운 마킹 M'으로 변환시킨다. 즉, M'은 M에서 도달 가능하다(reachable). $\forall p \in P$ 에 대해 만일 $p \in \bullet t$ 그리고 $p \neq t$ 이라면 $M'(p) = M(p) + 1$; 만일 $p \in \bullet t$ 그리고 $p = t$ 이라면 $M'(p) = M(p) - 1$; 그렇지 않으면 $M'(p) = M(p)$ 이다. 이 규칙은 퍼지 Pr/T 네트의 동적인 동작을 정의한다. $\bullet t$ 와 $t \bullet$ 는 t의 모든 입력 플레이스와 모든 출력 플레이스의 집합을 각각 의미한다.
- 투사행렬 : FPN이 m개의 플레이스와 n개의 트랜지션을 갖는 Pr/T 네트라고 하자. FPN의 투사행렬(Incidence matrix)은 $n \times m$ 정수행렬 $C = [c_{ij}]$. $C = [c_{ij}]$ 에 대해 FPN의 투사행렬은 $c_{ij} = w(t_i, p_j) - w(p_i, t_j) = (A_F(t_i, p_j))\beta_i - A_F(p_i, t_j)$ 이다. 여기에서 t_i 와 p_j 는 각각 트랜지션과 플레이스이며 $f(t_i) = \beta_i$ 이다.

퍼지 생성규칙을 퍼지 Pr/T 네트로 표현하면 그림과 같다. 그림 1은 1형 퍼지 생성규칙에 대한 퍼지 Pr/T 네트 표현이다. $\beta_k = \beta(\beta_i, \beta_j)$.



(그림 1) 1형 Pr/T 네트



(그림 2) 2형 Pr/T 네트

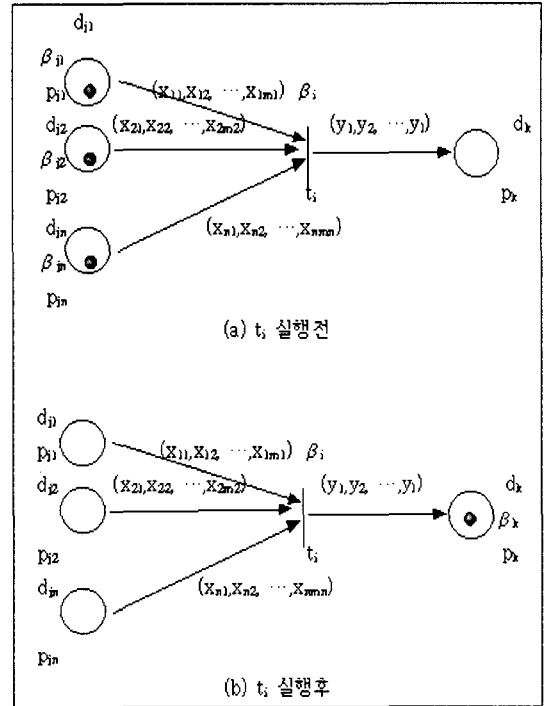
그림 2는 2형 퍼지 생성규칙에 대한 퍼지 Pr/T 네트 표현이다. p_i 에 있는 토큰은 p_k 중 오직 한 곳에만 나타난다. $\beta_k = \beta(\beta_i, \beta_j) = \beta_{\text{comp}}(\beta(\beta_{ip}, \beta_{jq})) = \min(\max(\beta(\beta_{ip}, \beta_{jq})))$.

$$p=1, q=1, 2, \dots, n.$$

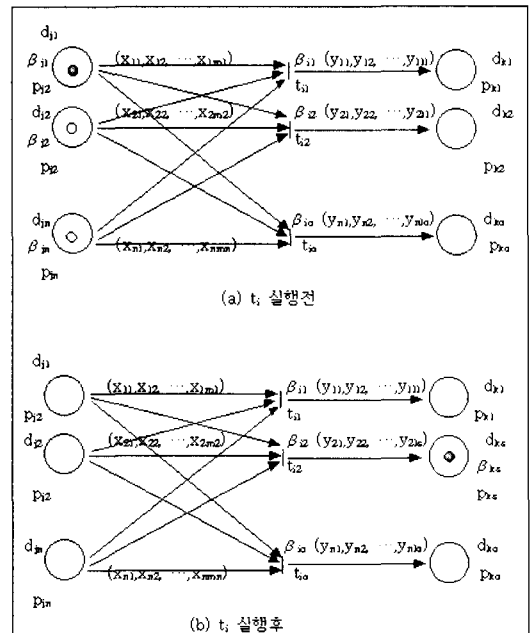
그림 3은 3형 퍼지 생성규칙에 대한 퍼지 Pr/T 네트 표현이다. $\beta_k = \beta(\beta_i, \beta_j) = \beta_{\text{comp}}(\beta(\beta_{ip}, \beta_{jq})) = \min(\max(\beta(\beta_{ip}, \beta_{jq})))$.

$$p=1, 2, \dots, n, q=1.$$

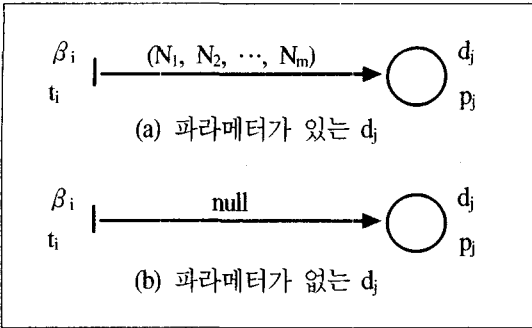
그림 4는 4형 퍼지 생성규칙에 대한 퍼지 Pr/T 네트 표현이다. $\beta_k = \beta(\beta_i, \beta_j) = \beta_{\text{comp}}(\beta(\beta_{ip}, \beta_{jq})) = \min(\max(\beta(\beta_{ip}, \beta_{jq})))$. $p=1, 2, \dots, n, q=1, 2, \dots, o$.



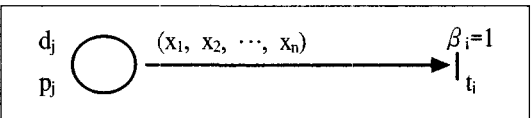
(그림 3) 3형 Pr/T 네트



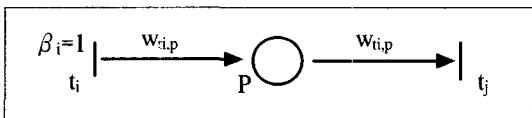
(그림 4) 4형 Pr/T 네트



(그림 5) 초기사실의 Pr/T 네트



(그림 6) 질의어의 Pr/T 네트



(그림 7) 단일화 가능관계의 Pr/T 네트

그림 5의 (a)와 (b)는 각각 파라미터를 가지고 있거나 없는 술어 d_j 의 인스턴스인 경우이다. 여기에서 t_i 는 근원(source) 트랜지션이 된다. 근원 트랜지션은 입력 플레이스가 없는 트랜지션으로, 언제나 실행 가능하고 출력 플레이스에 토큰을 출력한다. 그림 6은 질의어를 표현한다. 이 질의어의 부정형은 증명되어야 하고 목표 문장의 역할을 한다. 여기에서 t_i 는 배출(sink) 트랜지션이다. 배출 트랜지션은 출력 플레이스를 가지지 않는다. 그림 7은 단일화(unification)가 가능한 관계를 표현한 Pr/T 네트이다.

4. 퍼지 추론 알고리즘

4.1 알고리즘

FPN에 적당한 퍼지 추론 알고리즘을 개발하기 위해 혼절 추론[9][14]을 기반으로 하여 퍼지 확

장하였다. 단일화(unification)는 일차 술어논리의 추론에 필수적이다. 단일화 가능 관계를 명확히 표현하기 위해 각 플레이스 p 에 대한 단일화 가능 관계집합 $U(p)$ 가 필요하다. $\forall u(\theta) \in U(p)$, $u(\theta) = (w_{i,p}, w_{p,j}, \theta v)$ 는 mgu (most general unifier) θ 를 갖는 단일화가 가능한 쌍 $(w_{i,p}, w_{p,j})$ 를 표현한다. 여기에서 $w_{i,p}$ 는 t_i 에서 p 로 들어오는 아크에 관한 기호함이고, $w_{p,i}$ 는 p 에서 t_i 로 나가는 아크의 기호함이다. v 는 단일화 인자(unifier)의 믿음강도이다. 도출(resolution)을 사용하기 위해 단일화는 보수기호를 갖는 쌍(complementary signed occurrences)에서만 실행된다.

Lin et al. [9]의 알고리즘을 확장한 퍼지추론 알고리즘을 제안한다. Sheng-Ke Y. [17]과 본 논문에서 제안한 알고리즘은 Pr/T 네트를 기반으로 하는 Lin et al. [9]의 알고리즘을 기반으로 한다. Sheng-Ke Y. [17]은 Lin et al. [9]의 알고리즘에 \min 과 \max 를 이용하여 믿음값을 계산하는 부분이 본 논문에서는 3.2에서 제시한 믿음값 함수를 이용하여 믿음값을 계산하는 부분이 각각 제안하였다.

알고리즘

C: m 플레이스와 n 트랜지션을 갖는 FPN의 투사행렬; F_n : 믿음강도행렬; $\lambda \in [0,1]$: 임계값.

(1) $A := C_n \times m$, $D := F_n$, F_n 은 $n \times n$ 행렬.

$v_i = f(t_i)$ 는 퍼지 생성규칙의 믿음강도($i=1, 2, \dots, n$).

$$F_n = \begin{matrix} & \begin{matrix} t_1 & t_2 & \dots & t_{n-1} & t_n \end{matrix} \\ \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_{n-1} \\ t_n \end{matrix} & \begin{bmatrix} v_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & v_2 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & v_{n-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & v_n \end{bmatrix} \end{matrix}$$

(2) Repeat for $i=1$ until $i=n$

만일 $v_i < \lambda$ 이면 $[D | A]$ 의 i 번째 행에 있는 각

요소를 0으로 한다.

(3) Repeat for $i=1$ until $i=m$;

3.1 i 번째 열을 제거한 $[D | A]$ 에서 단일화가 가능한 행의 쌍들에 대해 음이 아닌 선형조합(nonegative linear combination)의 결과인 모든 행을 $[D | A]$ 에 추가한다. 이때 여기에 대응하는 단일화 인자와 단일화 인자의 믿음값 v 도 각 행의 쌍에 붙인다.

만일 $v < \lambda$ 이면 추가되는 각 행의 요소를 0으로 치환한다.

3.2 $[D | A]$ 의 A의 i 번째 열이 null이 아닌 행을 $[D | A]$ 에서 제거한다.

믿음값 평가 함수를 이용하여 믿음값 v 를 계산한다.

$n \times n$ 믿음강도 행렬과 $n \times m$ 투사행렬로 구성된 $n \times (n+m)$ 행렬을 가지고 알고리즘을 실행한다. 알고리즘은 플레이스의 수인 m 단계로 구성된다. 각 단계에서 투사행렬에서 하나의 열이 음이 아닌 선형조합을 실행하여 제거된다.

4.2 예

P와 A는 술어 Parent와 Ancestor를 표시한다. D, J 그리고 M은 David, John 그리고 Michael을 각각 표시한다. 예에서 사용하는 생성규칙은 R_1 과 R_2 이다.

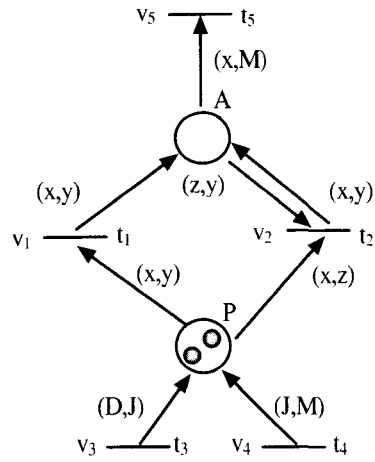
$$R = \{R_1, R_2\}$$

$$R_1: P(x,y) \rightarrow A(x,y) \text{ (Bel} = v_1 = 0.90)$$

$$R_2: P(x,z) \wedge A(z,y) \rightarrow A(x,y) \text{ (Bel} = v_2 = 0.80)$$

λ 는 임계값으로 $\lambda = 0.20$ 이다. $P(D,J)$ 과 $P(J,M)$ 의 믿음값은 각각 $v_3 = 0.90$ 과 $v_4 = 0.80$ 인 사실이다. 질의어는 “Who is/are the ancestor(s) of Michael?”. 즉, $A(x,M)$ 이다. 여기에서 믿음값은 $v_5 = 1.00$ 이다.

이 예의 FPN 표현과 투사행렬은 그림 8의 (a)와 (b)에 있다. 여기에서는 믿음값 계산의 편의상 규칙의 전제부와 결론부에 퍼지 개념이 없는 것으로 가정한다.



(a) FPN 표현

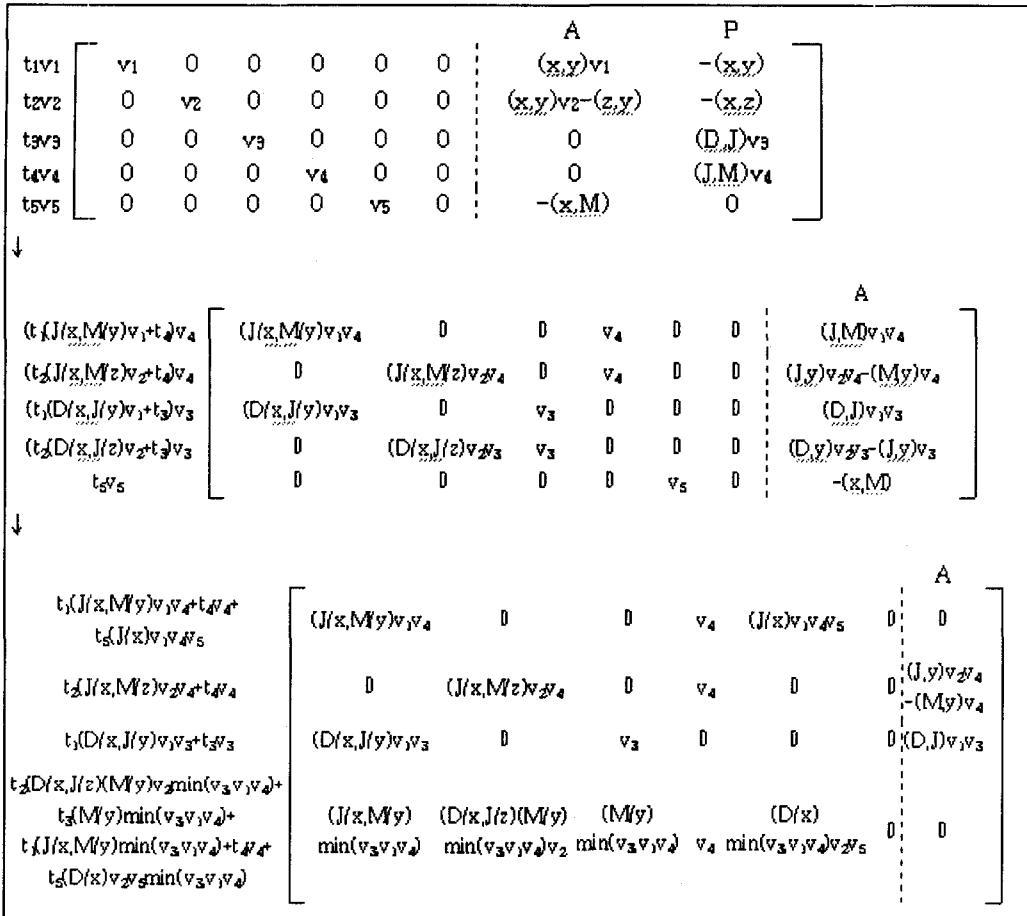
$$C = \begin{matrix} & & A & & P & & \\ t_1 & & (x,y)v_1 & & -(x,y) & & \\ t_2 & & (x,y)v_2 - (z,y) & & -(x,z) & & \\ t_3 & & 0 & & (D,J)v_3 & & \\ t_4 & & 0 & & (J,M)v_4 & & \\ t_5 & & -(x,M) & & 0 & & \end{matrix}$$

(b) 투사행렬

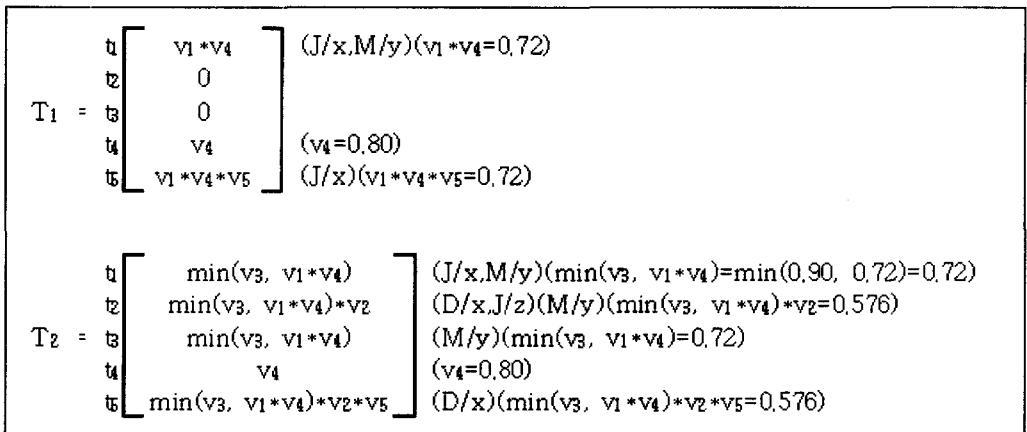
(그림 8) 예의 FPN 표현과 투사행렬

예에 대한 단일화가 가능한 관계집합은

- (1) $U(\text{Parent}) = \{u(\theta_1) = ((D,J), (x,y), (D/x, J/y)v_3), u(\theta_2) = ((D,J), (x,z), (D/x, J/z)v_3), u(\theta_3) = ((J,M), (x,y), (J/x, M/y)v_4), u(\theta_4) = ((J,M), (x,z), (J/x, M/z)v_4)\}$
- (2) $U(\text{Ancestor}) = \{u(\theta_5) = ((x,y), (x,M), (M/y)v_4), u(\theta_6) = ((x,y), (x,M), ((x,y), (z,y)), (M/y, z/x)v_6)\}$



(그림 9) 투사행렬을 이용한 알고리즘



(그림 10) 목표 트랜지션을 포함한 식

여기에서 v_a 와 v_b 는 t_1 과 t_2 가 실행될 때 각각 계산되는 사실의 믿음값이다.

주어진 예에 대한 알고리즘의 실행은 투사행렬을 이용하여 그림 9와 같이 표현할 수 있다. 목표 트랜지션 t_5 를 포함한 그림 10과 같은 식을 얻는다.

$T_1(t_5)$ 와 $T_2(t_5)$ 에서 $(J/x)0.72$ 와 $(D/x)0.576$ 을 각각 얻는다. $\lambda=0.20$ 이고 $0.72 > \lambda$, $0.576 > \lambda$ 이므로 질의어 $A(x,M)$ 에 대한 답은 $A(J,M)(0.72)$ 와 $A(D,M)(0.576)$ 이다. 이들은 서로 다른 통로를 통해 결론에 도달하였으므로 믿음값 결합함수에 의해 최종적인 결론의 믿음값은 다음과 같이 평가된다.

$$\begin{aligned} \beta_c &= \beta_{comb}(A(J,M)(0.72), A(D,M)(0.576)) \\ &= \max(0.72, 0.576) \\ &= 0.72 \end{aligned}$$

Jack과 David는 Michael의 조상일 수 있고, Jack이 보다 더 Michael의 조상인 것 같다. 이때의 믿음값은 0.72가 된다.

믿음값을 min과 max를 사용하여 계산하는 과정은 단순하고 간단하지만 사람이 사용하는 방법과는 차이가 있다. 본 논문에서는 규칙의 전제부와 결론부에 퍼지 개념의 유무에 따라 유사도나 퍼지관계, 특히 R_{sg} 를 이용하여 믿음값을 평가하기 때문에 기존의 방법보다 사람이 사용하는 추론과 직관에 더 유사하다.

5. 결 론

대부분의 지식기반 시스템에서 사용하는 규칙은 술어논리에 그 기반을 두고 있으므로, Pr/T 네트를 이용하는 방법은 명제논리를 기반으로 하는 페트리네트 방식보다 발전된 방식이다. 본 논문에서는 술어논리 수준의 지식표현을 하는 Pr/T 네트를 퍼지 확장한 퍼지 Pr/T 네트를 이용하여 퍼지 지식표현과 퍼지 추론방법을 제시하였다.

알고리즘에서 사용하는 믿음값 함수들은 기존의 연구처럼 단순히 max와 min 연산만을 사용하지

않고, 규칙의 전제부나 결론부에 퍼지 개념의 유무에 따라 유사도와 R_{sg} 를 이용하여 계산하는 믿음값 함수를 사용하기 때문에 사람들이 문제해결에 사용하는 추론과 직관에 보다 더 유사하다.

본 논문이 제안한 알고리즘은 단일 프로세서상에서 개발되었으므로 Pr/T 네트의 장점 중의 하나인 병렬성을 이용할 수 있는 병렬 추론 알고리즘의 개발이 필요하다.

참 고 문 헌

- [1] 전명근, 변증남, "Fuzzy Petri Nets를 이용한 퍼지추론 시스템의 모델링 및 추론기관의 구현," 전자공학회논문지, 제 29 권, 제 7 호, pp508-519, 1992, 7.
- [2] 조상엽, 김기태, "퍼지페트리네트를 이용한 퍼지생성규칙의 표현," 한국정보과학회논문지, 제 21 권, 제 2 호, pp298-306, 1994, 2.
- [3] Buchanan, B. G., and Shortliffe, E. H., Rule-based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project, Readings, MA: Addison-Wesely, 1984.
- [4] Chen, S., Ke, J., and Chang, J., "Knowledge Representation Using Fuzzy Petri-nets," IEEE Trans. on KDE, Vol. 2, No. 3, Sep., pp.311-319, 1990.
- [5] Fukami, S., Mizumoto, M., and Tanaka, K., "Some Considerations on Fuzzy Conditional Inference," Fuzzy Sets and Systems, Vol. 4, pp.243-273, 1980.
- [6] Garg, M. L., Ahson, S. I., and Gupta, D. V., "A Fuzzy Petri-nets for Knowledge Representation and Reasoning," Information Processing Letters, 39, pp165-171, 1992.
- [7] Genrich, H. J., and Lautenbach, K., "System Modelling with High-level Petri Nets," Theoretical Computer Science, 13, pp.109-136, 1981.
- [8] Leung, K. S., and Lam, W., "Fuzzy Concepts

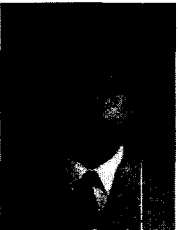
- in Expert Systems,” IEEE Computer, Sep., pp.43-56, 1988.
- [9] Lin, C., et al., “Logical Inference of Horn Clauses in Petri Net Models,” Knowledge and Data Engineering, 5, pp416-425, 1993.
- [10] Looney, G. C., and Alfize, A. A., “Logical Controls via Boolean Rule Matrix Transformation,” IEEE Trans. on SMC, Vol. 17, No. 6, Nov./Dec., pp1077-1082, 1987.
- [11] Looney, G. C., “Fuzzy Petri Nets for Rule-based Decision Making,” IEEE Trans. on SMC, Vol. 18, No. 1, Jan./Feb., 1988.
- [12] Manoj, T. V., Leena J., and Soney, R. B., “Knowledge Representation Using Fuzzy Petri Nets-Revisited,” IEEE Trans. on KDE, Vol. 10, No. 4, Jul./Aug., pp666-667, 1998.
- [13] Murata, T., “Petri Nets: Properties, Analysis and Applications,” Proceedings of the IEEE, Vol. 77, No. 4, April, pp541-580, 1989.
- [14] Peterka, G., and Murata, M., “Proof Procedure and Answer Extraction in Petri Net Model of Logic Programs,” IEEE Trans. Software Engineering, 15, pp209-217, 1989.
- [15] Peterson, J. L., Petri Net Theory and the Modeling of Systems, Prentice-hall, 1981.
- [16] Sheng-Ke Yu, “Comments on Knowledge Representation Using Fuzzy Petri Nets,” IEEE Trans. on KDE, Vol. 7, No. 1, Feb., pp190-192, 1995.
- [17] Sheng-Ke Yu, “Knowledge Representation and Reasoning Using Fuzzy Pr/T net-systems,” Fuzzy Sets and Systems, 75, pp33-45, 1995.

● 저자 소개 ●



조 상 업

1986년 한남대학교 전자계산학과(공학사)
 1988년 중앙대학교 대학원 전자계산학과(이학석사)
 1993년 중앙대학교 대학원 전자계산학과(공학박사)
 1993~1994 중앙대학교 컴퓨터소프트웨어 연구소 객원연구원
 1995~현재 청운대학교 인터넷컴퓨터학과 교수
 관심분야 : 인공지능, 퍼지이론, 페트리네트 응용
 E-mail : sycho@cwunet.ac.kr



이 동 은

1991년 전북대학교 컴퓨터공학과(공학사)
 1996년 전북대학교 대학원 컴퓨터공학과(공학석사)
 2000년 전북대학교 대학원 컴퓨터공학과(공학박사)
 2000~현재 청운대학교 인터넷컴퓨터학과 교수
 관심분야 : B-ISDN, 무선 ATM망
 E-mail : delee@cwunet.ac.kr