

가중 퍼지 페트리네트를 이용한 가중 퍼지 후진추론

Weighted Fuzzy Backward Reasoning Using Weighted Fuzzy Petri-Nets

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

요 약

본 논문에서는 가중 퍼지 페트리네트에 기반을 둔 규칙기반시스템을 위한 가중 퍼지 후진추론 알고리즘을 제안한다. 규칙기반시스템에 있는 퍼지 생성규칙은 가중 퍼지 페트리네트로 모형화된다. 여기에서 퍼지 생성규칙에 나타나는 퍼지 명제의 진리값과 규칙의 확신도는 퍼지 숫자로 표현한다. 그리고 규칙에 나타나는 퍼지 명제의 가중값도 퍼지 숫자로 표현한다. 제안한 가중 퍼지 후진추론 알고리즘은 목표노드에서 초기노드까지 후진추론 통로를 생성한 후 목표노드의 확신도를 계산한다. 우리가 제안한 알고리즘은 규칙기반시스템이 더 유연하고 사람과 같은 방법으로 가중 퍼지 후진추론을 하는 것을 가능하게 한다.

Abstract

This paper presents a weighted fuzzy backward reasoning algorithm for rule-based systems based on weighted fuzzy Petri nets. The fuzzy production rules in the knowledge base of a rule-based system are modeled by weighted fuzzy Petri nets, where the truth values of the propositions appearing in the fuzzy production rules and the certainty factors of the rules are represented by fuzzy numbers. Furthermore, the weights of the propositions appearing in the rules are also represented by fuzzy numbers. The proposed weighted fuzzy backward reasoning generates the backward reasoning path from the goal node to the initial nodes and then evaluates the certainty factor of the goal node. The algorithm we proposed can allow the rule-based systems to perform weighted fuzzy backward reasoning in more flexible and human-like manner.

Keyword : weighted fuzzy Petri nets, fuzzy backward reasoning, fuzzy numbers

1. 서 론

사람들이 사용하고 있는 애매한 지식을 표현하기 위한 방법으로 퍼지 페트리네트를 이용하는 방법이 많이 연구되고 있다[1, 2, 3, 4, 6, 8, 10, 13, 14]. 퍼지 생성규칙을 모형화하기 위해 사용하는 퍼지 페트리네트는 페트리네트[9, 11]에 퍼지 개념[12]을 표현할 수 있도록 확장한 네트이다. 퍼지 페트리네트는 정보의 흐름을 표현하기 위해

객체간의 입출력관계를 보여주고, 병렬성, 비동기성 그리고 분산성을 갖는 정보처리시스템을 기술하거나 연구하는 데 유용한 도구이다.

[1]에서는 퍼지 생성규칙의 확신도, 규칙에 나타나는 명제의 진리값과 규칙에 나타나는 명제의 중요도에 따라 부여하는 가중값을 퍼지 숫자로 주는 방법을 제안하였다. [3, 8, 10]에서는 규칙베이스에 있는 퍼지 생성규칙을 퍼지 페트리네트로 표현하는 방법과 퍼지 생성규칙의 확신도와 규칙에 나타나는 명제의 확신도를 퍼지 집합으로 표현하는 방법과 퍼지 추론 알고리즘을 제안하였다. [4]에서는 퍼지 페트리네트의 축소(reduction)를 이용하여 퍼지 지식베이스의 일관성을 검사하고, 정

* 정 회 원 : 청운대학교 인터넷컴퓨터학과 교수
sycho@mail.chungwoon.ac.kr(제 1저자)

** 정 회 원 : 청운대학교 인터넷컴퓨터학과 교수
delec@mail.chungwoon.ac.kr(공동저자)

리증명을 위한 반박(refutation)방법을 제안하였다. [6]에서는 퍼지 생성규칙에 나타나는 퍼지 명제의 불확실성을 추론하기 위해 구조화된 퍼지 추론엔진으로서 퍼지 페트리네트를 이용하는 방법을 제안하였다. 여기에서 사용하는 믿음갱신(belief-revision)은 기존의 확률적인 방법과는 달리 퍼지 집합을 사용하였다. [7]에서는 규칙기반 의사결정에 사용하기 위해 기존의 페트리네트를 퍼지 페트리네트로 확장하고, 여기에 사용하기위한 퍼지 추론 알고리즘은 퍼지 집합을 표현하는 퍼지 행렬을 기반으로 개발하였다. [13]에서는 퍼지 생성규칙의 믿음값과 규칙에 나타나는 명제의 믿음값을 퍼지 집합을 기반으로 하여 표현하고, 규칙의 전제부와 결론부에 있는 명제에 퍼지 개념의 유무에 따라 결론의 믿음값을 평가하는 방법을 제안하였다. 이 방법은 확신도만을 따로 계산하는 방법과 달리 규칙에 있는 명제의 퍼지 개념을 고려하여 결론의 확신도를 계산하였다.

그렇지만 [1], [3], [4], [6-7], [10] 그리고 [13]에서 제안한 추론방법은 전진추론 알고리즘을 다루고 후진추론 알고리즘은 다루지 않았다. Arnould, T.는 [1]에서 후진추론에서는 목표(goal)를 증명하기 위해 이 목표와 관련된 정보를 고려하기 때문에, 특정한 목표를 증명하려고 할 때에는 증명에 필요한 정보의 양이 더 적은 후진추론이 전진추론보다 더 효과적이라고 주장하였다. 그리고 규칙의 확신도를 구간값 퍼지 집합에 기반을 둔 구간으로 표현하고, 일반화된 modus ponens에 기반을 둔 퍼지 후진추론방법을 제안하였다. [2]에서 Chen, S.와 [14]에서 조상엽은 후진추론을 하는 능력을 컴퓨터에 허용한다면 컴퓨터 시스템이 더 유연(flexible)하고 더 지적(intelligent)이게 된다고 지적하고, 퍼지 페트리네트를 사용하는 규칙기반 시스템을 위한 퍼지 후진 추론알고리즘을 제안하였다. [2]에서는 규칙의 확신도를 0과 1사이의 실수로 표현하고, 규칙의 확신도를 구하기 위해 단순히 min과 max 연산을 사용하는 방법을 제안하였다. [14]에서는 규칙의 믿음값을 0과 1사이의 실

수로 표현하고, 규칙의 믿음값을 구하기 위해, 규칙에 전제부와 결론부의 명제에 나타나는 퍼지 개념의 유무에 따라 믿음값을 믿음값 평가함수로 평가하는 방법을 제안하였다. 제안한 알고리즘은 목표노드에서 시작노드까지 후진추론 통로를 찾아낸 후 목표노드의 확신도를 계산한다.

본 논문에서는 퍼지 생성규칙을 모형화하기 위한 가중 퍼지 페트리네트를 정의하고, 퍼지 규칙의 전제부와 결론부에 있는 퍼지 명제의 중요도를 고려하여 부여한 가중값을 이용하는 가중 퍼지 후진추론 방법을 제안한다. 여기에서 퍼지 생성규칙의 확신도와 규칙에 나타나는 퍼지 명제의 진리값은 0과 1사이의 실수가 아닌 퍼지 숫자로 표현한다. 그리고 규칙에 나타나는 퍼지 명제의 중요도를 나타내는 가중값도 퍼지 숫자로 표현한다. [15]에서 제안한 후진 추론알고리즘은 실세계의 응용을 실행하는 데 많은 시간이 걸리므로 효율적인 퍼지 후진추론 알고리즘의 개발이 필요하다. 퍼지 후진 추론알고리즘을 개발하기 위해 [2]와 [14]의 알고리즘을 기반으로 하여 퍼지 페트리네트에서 후진추론 문제를 다룰 수 있는 가중 퍼지 후진추론 알고리즘을 제안한다. [16]에서는 퍼지 집합을 기반으로 하여 의료진단 문제를 다루는 기존의 연구[17]에서는 환자의 여러 가지 증상을 모두 동일한 중요도로 가정하였으나, 환자에 적절한 치료나 처방을 위해서는 환자의 여러 가지 증상이 다른 정도의 중요도를 갖는다고 하였다. 그러므로 환자의 증상과 대응하는 생성규칙의 전제부에 있는 명제에 이러한 중요도를 표현할 수 있는 가중값(weight)을 고려하는 추론알고리즘을 개발한다면 보다 더 유연하고 전문가가 하는 것과 같은 추론이 가능하다고 하였다. 제안한 추론 알고리즘은 퍼지 명제의 중요도를 고려한 추론을 실행하기 때문에 기존의 추론방식보다 더 유연하고 사람과 같은 퍼지 추론을 할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 퍼지 숫자를 간단하게 살펴본다. 3 장에서는 지식표현방법과 추론방법을 기술한다. 4 장에서는 가중

퍼지 페트리네트를 정의한다. 5 장에서는 가중 퍼지 후진추론 알고리즘을 제안하고 예를 보인다. 6 장에서는 결론을 기술한다.

2. 퍼지 숫자

퍼지 집합이론은 1965년 Zadeh에 의해 제안되었다[13]. 퍼지 집합은 퍼지 경계의 한 종류로 볼 수 있다. 전체집합(universe of discourse) U , $U=\{u_1, u_2, \dots, u_n\}$ 에 있는 퍼지 집합 \tilde{A} 는 소속함수 $\mu_{\tilde{A}}$, $\mu_{\tilde{A}}:U \rightarrow [0,1]$ 에 의해 다음과 같이 표현될 수 있다.

$$\tilde{A} = \mu_{\tilde{A}}(u_1)/u_1 + \mu_{\tilde{A}}(u_2)/u_2 + \dots + \mu_{\tilde{A}}(u_n)/u_n$$

여기에서 $\mu_{\tilde{A}}(u_i)$ 는 퍼지 집합 \tilde{A} 에 u_i 의 소속 정도를 가리킨다. 만일 $\exists u_i \in U$, $\mu_{\tilde{A}}(u_i)=1$ 이면 퍼지 집합 \tilde{A} 는 정상(normal)이다. 전체집합 U 에 있는 모든 u_1, u_2 에 대해서 다음을 만족하면 퍼지 집합 \tilde{A} 는 볼록(convex)이다.

$$\mu_{\tilde{A}}(\lambda u_1 + (1-\lambda)u_2) \geq \text{Min}(\mu_{\tilde{A}}(u_1) + \mu_{\tilde{A}}(u_2))$$

여기에서 $\lambda \in [0,1]$. 퍼지 숫자는 전체집합 U 에서 볼록하고 정상인 퍼지 집합이다.

삼각 퍼지 숫자 \tilde{A} 는 세 쌍의 파라미터 (a_1, a_2, a_3) 로 표현할 수 있다. 삼각 퍼지 숫자 \tilde{A} 는 다음과 같이 정의할 수 있다.

$$\mu_{\tilde{A}}(u) = \begin{cases} 0, & u < a_1 \\ (u - a_1)/(a_2 - a_1), & a_1 \leq u \leq a_2 \\ (a_3 - u)/(a_3 - a_2), & a_2 \leq u \leq a_3 \\ 0, & u \geq a_3 \end{cases}$$

\tilde{A} 와 \tilde{B} 를 세 쌍 (a_1, a_2, a_3) 와 (b_1, b_2, b_3) 로 각각 표현되는 삼각 퍼지 숫자라고 하자. 삼각 퍼지 숫자 \tilde{A} 와 \tilde{B} 사이의 산술연산은 다음과 같이 정의된다[1, 5].

삼각퍼지숫자 더하기 \oplus : $\tilde{A} \oplus \tilde{B} = (a_1, a_2, a_3) \oplus (b_1, b_2, b_3) = (a_1+b_1, a_2+b_2, a_3+b_3)$.

삼각퍼지숫자 빼기 \ominus : $\tilde{A} \ominus \tilde{B} = (a_1, a_2, a_3) \ominus (b_1,$

$b_2, b_3) = (a_1-b_3, a_2-b_2, a_3-b_1)$.

삼각퍼지숫자 곱하기 \otimes : $\tilde{A} \otimes \tilde{B} = (a_1, a_2, a_3) \otimes (b_1, b_2, b_3) = (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3)$.

삼각퍼지숫자 나누기 \oslash : $\tilde{A} \oslash \tilde{B} = (a_1, a_2, a_3) \oslash (b_1, b_2, b_3) = (a_1/b_3, a_2/b_2, a_3/b_1)$.

전체집합 U 에서 퍼지 집합 \tilde{A} 의 α -cut A_α 는 다음과 같이 정의된다.

$$A_\alpha = \{u_i \mid \mu_{\tilde{A}}(u_i) \geq \alpha, u_i \in U\}$$

여기에서 $\alpha \in [0,1]$. 삼각 퍼지 숫자 \tilde{A} 와 \tilde{B} 는 수준집합(level set; i.e., α -cut)으로 각각 나눌 수가 있다.

$$\tilde{A} = \int_0^1 \alpha A_\alpha$$

$$\tilde{B} = \int_0^1 \alpha B_\alpha$$

여기에서 $A_\alpha = [a_1^{(\alpha)}, a_2^{(\alpha)}]$, $B_\alpha = [b_1^{(\alpha)}, b_2^{(\alpha)}]$, $\alpha \in [0,1]$. 따라서 삼각 퍼지 숫자 \tilde{A} 와 \tilde{B} 의 OR 연산은 $\tilde{A} \oslash \tilde{B}$ 로 표기하고, 다음과 같은 전체집합내의 식으로 기술할 수 있다.

$$\tilde{A} \oslash \tilde{B} = \int_0^1 \alpha [a_1^{(\alpha)} \vee b_1^{(\alpha)}, a_2^{(\alpha)} \vee b_2^{(\alpha)}]$$

여기에서 \oslash 는 최대값 연산자이다.

3. 지식표현과 추론

퍼지 생성규칙을 이용하면 실세계에서 사람이 사용하는 불확실하고 애매한 지식을 표현할 수 있다. 퍼지 생성규칙은 두 개의 퍼지 명제사이의 퍼지 관계를 기술한다고 생각할 수 있다. 이 장에서는 가중 퍼지 논리(weighted fuzzy logic)에 기반을 둔 지식표현과 가중 퍼지 추론에 대하여 기술한다[1-3][8][10].

사례 1: 다음과 같은 규칙이 규칙베이스에 있다고 가정하자.

$$\text{Rule}_1: d_j \Rightarrow d_k \text{ (CF}=\tilde{\beta}_1\text{)}$$

여기에서 Rule₁는 규칙의 이름이고, d_j와 d_k는 각각 퍼지 명제이다. $\tilde{\beta}_1$ 는 전체집합(universe of discourse) [0,1]에서 정의된 퍼지 숫자로서 규칙 Rule₁의 확신도이다. 명제 d_j와 d_k의 가중값(weight)은 각각 $\tilde{\omega}_j$ 와 $\tilde{\omega}_k$ 로 표현하고, $\tilde{\omega}_j$ 와 $\tilde{\omega}_k$ 는 전체집합 [0,1]에서 정의되는 퍼지 숫자이다. 명제 d_j와 d_k의 퍼지 진리값(fuzzy truth value)을 각각 $\tilde{\tau}_j$ 와 $\tilde{\tau}_k$ 라고 가정하자. 여기에서 $\tilde{\tau}_j$ 와 $\tilde{\tau}_k$ 는 전체집합 [0,1]에서 정의되는 퍼지 숫자이다. 그러면 명제 d_k의 퍼지 진리값 $\tilde{\tau}_k = \tilde{\tau}_j \otimes \tilde{\beta}_1$ 로 평가할 수 있다. \otimes 는 퍼지 숫자의 곱하기 연산자이다.

사례 2: 다음과 같은 합성 퍼지 생성규칙이 규칙베이스에 있다고 가정한다.

$$\text{Rule}_1: d_{j_1} \wedge d_{j_2} \wedge \dots \wedge d_{j_m} \Rightarrow d_k \text{ (CF}=\tilde{\beta}_1\text{)}$$

여기에서 $\tilde{\beta}_1$ 는 전체집합 [0,1]에서 정의된 퍼지 숫자로서 규칙 Rule₁의 확신도이다. 명제 d_{j₁}, d_{j₂}, ..., d_{j_m}의 가중값은 각각 전체집합 [0,1]에서 정의되는 퍼지 숫자 $\tilde{\omega}_{j_1}$, $\tilde{\omega}_{j_2}$, ..., $\tilde{\omega}_{j_m}$ 이다. 그리고 명제 d_{j₁}, d_{j₂}, ..., d_{j_m}의 퍼지 진리값은 전체집합 [0,1]에서 정의되는 퍼지 숫자 $\tilde{\tau}_{j_1}$, $\tilde{\tau}_{j_2}$, ..., $\tilde{\tau}_{j_m}$ 이다. 그러면 명제 d_k의 퍼지 진리값 $\tilde{\tau}_k = [(\tilde{\tau}_{j_1} \otimes \tilde{\omega}_{j_1} \oplus \tilde{\tau}_{j_2} \otimes \tilde{\omega}_{j_2} \oplus \dots \oplus \tilde{\tau}_{j_m} \otimes \tilde{\omega}_{j_m}) \oslash (\tilde{\omega}_{j_1} \oplus \tilde{\omega}_{j_2} \oplus \dots \oplus \tilde{\omega}_{j_m})] \otimes \tilde{\beta}_1$ 로 평가할 수 있다. 여기에서 \oplus , \otimes , \oslash 는 퍼지 숫자의 더하기, 곱하기, 나누기 연산자이다.

사례 3: 다음과 같은 합성 퍼지 생성규칙이 규칙베이스에 있다고 가정한다.

$$\text{Rule}_1: d_j \Rightarrow d_{k_1} \wedge d_{k_2} \wedge \dots \wedge d_{k_n} \text{ (CF}=\tilde{\beta}_1\text{)}$$

여기에서 $\tilde{\beta}_1$ 는 전체집합 [0,1]에서 정의된 퍼지 숫자로서 규칙 Rule₁의 확신도이다. 명제 d_j, d_{k₁},

d_{k₂}, ..., d_{k_n}의 가중값은 각각 전체집합 [0,1]에서 정의되는 퍼지 숫자 $\tilde{\omega}_j$, $\tilde{\omega}_{k_1}$, $\tilde{\omega}_{k_2}$, ..., $\tilde{\omega}_{k_n}$ 이고, $\tilde{\omega}_j$ 는 j=1이다. 명제 d_j의 퍼지 진리값은 전체집합 [0,1]에서 정의되는 퍼지 숫자 $\tilde{\tau}_j$ 라고 가정한다. d_{k₁}, d_{k₂}, ..., d_{k_n}의 퍼지 진리값 $\tilde{\tau}_{k_s} = \tilde{\tau}_j \otimes \tilde{\beta}_1$ 이다. s=1, 2, ..., n. 여기에서 \otimes 는 퍼지숫자 곱하기 연산자이다.

사례 4: 다음과 같은 합성 퍼지 생성규칙이 규칙베이스에 있다고 가정한다.

$$\text{Rule}_1: d_{j_1} \vee d_{j_2} \vee \dots \vee d_{j_m} \Rightarrow d_k \text{ (CF}=\tilde{\beta}_1\text{)}$$

여기에서 $\tilde{\beta}_1$ 는 전체집합 [0,1]에서 정의된 퍼지 숫자로서 규칙 Rule₁의 확신도이다. 명제 d_{j₁}, d_{j₂}, ..., d_{j_m}의 가중값은 각각 전체집합 [0,1]에서 정의되는 퍼지 숫자 $\tilde{\omega}_{j_1}$, $\tilde{\omega}_{j_2}$, ..., $\tilde{\omega}_{j_m}$ 이다. 그리고 명제 d_{j₁}, d_{j₂}, ..., d_{j_m}의 퍼지 진리값은 각각 전체집합 [0,1]에서 정의되는 퍼지숫자 $\tilde{\tau}_{j_1}$, $\tilde{\tau}_{j_2}$, ..., $\tilde{\tau}_{j_m}$ 이다. 그러면 명제 d_k의 퍼지 진리값 $\tilde{\tau}_{k_s} = [\bigvee_s (\tilde{\tau}_{j_s} \otimes \tilde{\omega}_{j_s}) \oslash (\tilde{\omega}_{j_1} \oplus \tilde{\omega}_{j_2} \oplus \dots \oplus \tilde{\omega}_{j_m})] \otimes \tilde{\beta}_1$ 로 평가할 수 있다. 여기에서 s= 1, 2, ..., m이고 \otimes , \oslash , \bigvee_s 는 퍼지 숫자의 곱하기, 나누기, 최대값 연산자이다.

4. 가중 퍼지 페트리네트

규칙기반시스템의 가중 퍼지 후진추론을 모형화하기 위한 가중 퍼지 페트리네트(Weighted Fuzzy Petri Net: WFPN)를 정의한다[1-3][10].

$$\text{WFPN} = (P, T, D, I, O, \tau, \alpha, \beta, W)$$

여기에서 P={p₁, p₂, ..., p_n}은 플레이스(place)의 유한집합이고, T={t₁, t₂, ..., t_m}은 트랜지션(transition)의 유한집합이며, D={d₁, d₂, ..., d_n}은 명제의 유한집합이다. P ∩ T = ∅, P ∪ T ≠ ∅. I: T → P[∞]은 트랜지션을 입력 플레이스에 사상시키는 입력 함수이고, O: T → P[∞]은 트랜지션을 출력 플레이스에 사상시키는 출력함수이다. τ: P → {0, 1}사이의 퍼지 숫자는 플레이스에 있는 토큰을 퍼지 숫자

로 사상시키는 토큰함수이다. $\alpha: P \rightarrow D$ 는 플레이스를 명제에 사상시키는 전단사함수이다. $\beta: T \rightarrow [0, 1]$ 사이의 퍼지 숫자는 트랜지션을 퍼지 숫자로 사상시키는 트랜지션함수이다. $W: P \rightarrow [0, 1]$ 사이의 퍼지 숫자는 플레이스를 퍼지 숫자에 사상시키는 가중함수이다.

$\tau_i = \tau(p_i)$ 는 플레이스 p_i 에 있는 토큰의 진리값이고, $\tilde{\beta}_i = \beta(t_i)$ 는 트랜지션 t_i 의 입력 플레이스와 출력 플레이스의 관계의 정도를 나타내는 확신도이다. $\tilde{\omega}_i = W(p_i)$ 는 플레이어의 중요도를 나타내는 가중값이다. 실행가능한(enable) t_i 는 입력 플레이스에서는 토큰을 제거하고 출력 플레이스에는 토큰을 출력하면서 실행(fire)하게 된다. 출력 플레이스에 나타나는 토큰의 확신도는 트랜지션의 확신도와 실행전의 입력 플레이스에 있는 토큰의 확신도를 이용하여 계산한다.

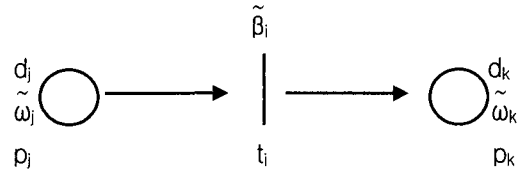
만일 $p_i \in I(t_i)$ 이고 $p_j \in O(t_i)$ 이라면, p_i 는 p_j 에서 직접 후진 도달가능하다(direct backward reachable). p_j 에서 직접 후진 도달 가능한 플레이어의 집합 $DBRS(p_j)$ 을 직접후진도달집합(direct backward reachability set)이라고 한다. p_i 가 p_j 에서 직접 후진 도달 가능하고, p_j 는 p_k 에서 직접 후진 도달 가능하다면, p_i 는 p_k 에서 후진 도달가능하다(backward reachable). 이러한 플레이어의 집합을 후진 도달집합(backward reachability set) $BRS(p_i)$ 라고 한다. 후진 도달관계는 직접 후진 도달관계의 반사추이폐포(reflexive transitive closure)가 된다.

$p_i \in I(t_i)$ 이고 $p_j \in I(t_i)$ 이라면 p_j 를 t_i 에 대한 p_i 의 이웃플레이스(neighbor place)라고 한다. 이러한 p_j 의 집합을 이웃플레이스 집합(neighbor place set)이라고 하고 $NPS(p_i)$ 로 표시한다. 그리고 $|NPS(p_i)|$ 는 원소의 수이다.

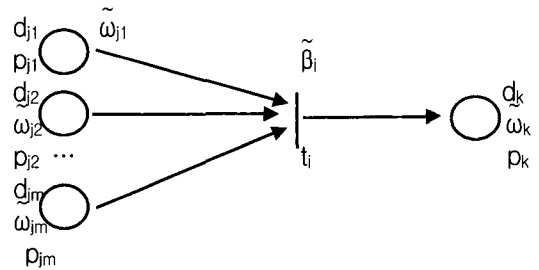
그림 1-4는 각각 사례 1-4의 퍼지 생성규칙의 가중 퍼지 페트리네트 표현이다.

5. 가중 퍼지 후진추론 알고리즘

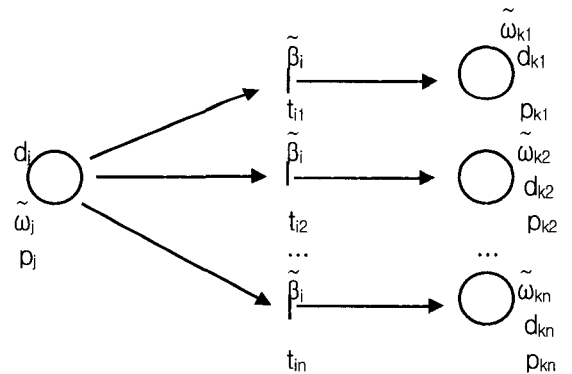
이 장에서는 가중 퍼지 페트리네트에서 사용하



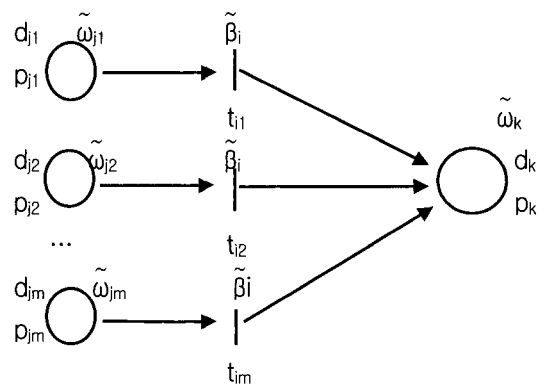
〈그림 1〉 사례 1에 대한 FPN



〈그림 2〉 사례 2에 대한 FPN



〈그림 3〉 사례 3에 대한 FPN



〈그림 4〉 사례 4에 대한 FPN

기 위한 가중 퍼지 후진추론 알고리즘을 제안한다. 이 알고리즘은 목표노드에서 시작노드까지의 후진추론 통로를 찾아낸 후, 목표노드의 확신도를 계산한다.

GOAL은 목표노드의 리스트이다. OPEN은 부모노드를 찾기 위해 현재 처리중인 플레이스, 즉 현재 노드의 리스트이다. PARENTS와 PARENTHSHIP은 현재 처리중인 노드의 부모노드를 저장하는 리스트이다. PLACES와 TRANSITIONS는 목표노드에서 시작노드사이의 후진추론 통로 상에 있는 플레이스와 트랜지션을 저장하는 리스트이다. STARTS는 부모노드가 더 이상 없는, 즉 시작노드를 저장하는 리스트이다. TOKENPLACES는 현재 토큰을 가지고 있는 플레이스를 저장하는 리스트이다. ENABLES는 현재 실행 가능한 트랜지션의 리스트이다. t_{ik} 는 플레이스 p_i 과 p_k 사이에 있는 트랜지션이다. $I(t_a)$ 는 트랜지션 t_a 에 대한 입력플레이스의 수이다.

알고리즘의 I 과 II는 목표노드에서 시작노드까지 후진추론 통로를 찾아준다. III은 시작노드의 확신도를 사용자로부터 입력받고, IV, V 그리고 VI은 목표노드의 확신도를 계산한다.

가중퍼지 후진추론 알고리즘

입력: GOAL
출력: 목표노드의 확신도 $\tilde{\tau}_j$

```

I. OPEN = GOAL
II. repeat until (OPEN == ∅)
    1 PARENTS = ∅
    2 do for each  $p_k$  in OPEN
        2.1 OPEN -= { $p_k$ }
        2.2 PLACES U= { $p_k$ }
        2.3 PARENTHSHIP = ∅
        2.4 if (DBRS( $p_k$ ) != ∅)
            then do for each  $p_i$  in DBRS( $p_k$ )
                PARENTHSHIP U= { $p_i$ }
    
```

```

TRANSITIONS U= { $t_{ik}$ }
else do for each  $p_i$  in DBRS( $p_k$ )
    STARTS U= { $p_i$ }
2.5 PARENTS U= PARENTHSHIP
3 OPEN = PARENTS
III. Input  $\forall p_i \in$  STARTS
IV. TOKENPLACES = STARTS
V. repeat until (TRANSITIONS == ∅)
    1 do for each  $t_a$  in TRANSITIONS
        1.1 if  $I(t_a) \subseteq$  TOKENPLACES
            then ENABLES U= { $t_a$ }
            TRANSITIONS -= { $t_a$ }
        2 do for each  $t_a$  in ENABLES;  $p_j=O(t_a)$ 
            2.1 If  $|I(t_a)| \geq 2$ 
                then  $\tilde{\tau}_j = [(\tilde{\tau}_{i1} \otimes \tilde{\omega}_{i1} \oplus \tilde{\tau}_{i2} \otimes \tilde{\omega}_{i2} \oplus \dots \oplus \tilde{\tau}_{in} \otimes \tilde{\omega}_{in}) \otimes (\tilde{\omega}_{i1} \oplus \tilde{\omega}_{i2} \oplus \dots \oplus \tilde{\omega}_{in})] \otimes \tilde{\beta}_a$ ,
                    where  $\tilde{\tau}_i = \tau(p_i)$ ,  $\tilde{\omega}_i = W(p_i)$ ,  $\forall p_i \in I(t_a)$ ,  $\tilde{\beta}_a = \beta(t_a)$ 
                else  $\tilde{\tau}_j = \tilde{\tau}_i \otimes \tilde{\omega}_i$ ,
                    TEMPVAUES U=  $\tilde{\tau}_j$ ,
                     $\tilde{\omega}_s \oplus = \tilde{\omega}_i$ ,
                    If  $p_j \in$  TOKENPLACES
                        then [ $\bigotimes \{ \tilde{\tau}_j \otimes \tilde{\omega}_s \mid I \quad \forall \tilde{\tau}_j \in$ 
TEMPVAUES}]  $\otimes \tilde{\beta}_a$ 
                    else TOKENPLACES U= { $p_j$ }
            2.2 ENABLES -= { $t_a$ }
VI. return  $\tilde{\tau}_j$ 
    
```

5.1 예([1-3]의 예를 기반으로)

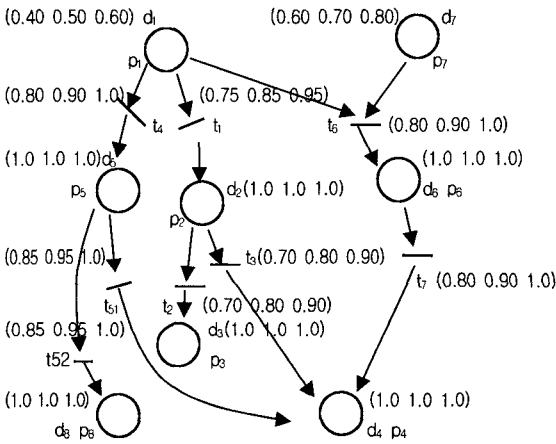
다음과 같은 퍼지 생성규칙이 지식베이스에 있다고 가정하자.

- Rule₁: IF d_1 THEN d_2 ($\tilde{\beta}_1=(0.75 \ 0.85 \ 0.95)$)
- Rule₂: IF d_2 THEN d_3 ($\tilde{\beta}_2=(0.70 \ 0.80 \ 0.90)$)
- Rule₃: IF d_2 THEN d_4 ($\tilde{\beta}_3=(0.70 \ 0.80 \ 0.90)$)
- Rule₄: IF d_1 THEN d_5 ($\tilde{\beta}_4=(0.80 \ 0.90 \ 1.00)$)
- Rule₅: IF d_5 THEN $d_4 \wedge d_8$ ($\tilde{\beta}_5=(0.85 \ 0.95 \ 1.00)$)

Rule₆: IF $d_1 \wedge d_7$ THEN d_6 ($\tilde{\beta}_6=(0.80 \ 0.90 \ 1.00)$)

Rule₇: IF d_6 THEN d_4 ($\tilde{\beta}_7=(0.80 \ 0.90 \ 1.00)$)

여기에서 $d_1, d_2, d_3, d_4, d_5, d_6, d_7$, 그리고 d_8 을 퍼지 명제라고 하자. 이 명제들의 가중값은 각각 (0.45 0.50 0.60), (1.0 1.0 1.0), (1.0 1.0 1.0), (1.0 1.0 1.0), (1.0 1.0 1.0), (1.0 1.0 1.0), (1.0 1.0 1.0) 그리고 (0.60 0.70 0.80)이라고 가정한다. 이 규칙들의 퍼지 페트리네트표현은 그림 5와 같다.



〈그림 5〉 퍼지 페트리네트 표현

〈표 1〉 직접후진도달집합

플레이스	DBRS(p_i)
p_1	\emptyset
p_2	$\{p_1\}$
p_3	$\{p_2\}$
p_4	$\{p_2, p_5, p_6\}$
p_5	$\{p_1\}$
p_6	$\{p_1, p_7\}$
p_7	\emptyset
p_8	$\{p_5\}$

플레이스에 대한 직접후진도달집합 DBRS(p_i)는 표 1과 같다. 각 플레이스에 대한 이웃플레이스집합은 표 2와 같다.

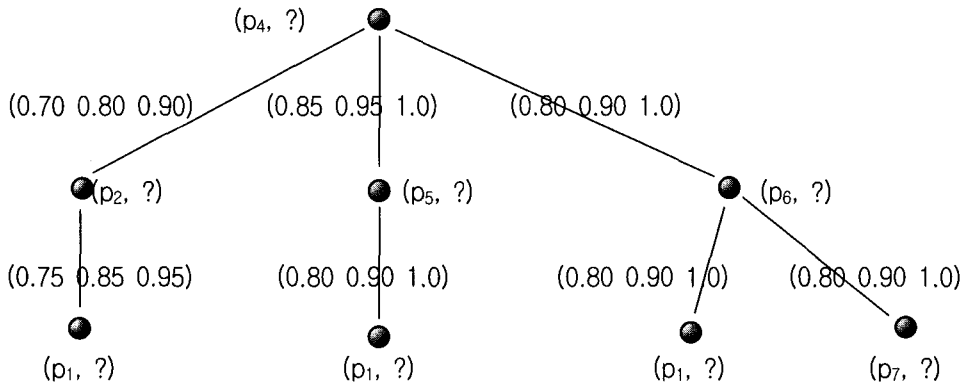
〈표 2〉 이웃플레이스 집합

플레이스 p_i	플레이스 p_k	NPS _{ik}
p_1	p_2	\emptyset
	p_5	\emptyset
	p_6	$\{p_7\}$
p_2	p_3	\emptyset
	p_4	\emptyset
p_5	p_4	\emptyset
	p_8	\emptyset
p_6	p_4	\emptyset
p_7	p_6	$\{p_1\}$

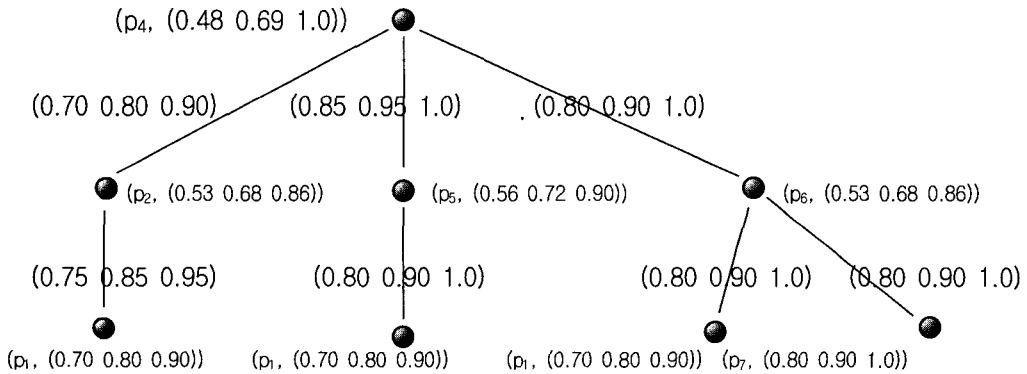
그림 6은 후진추론 알고리즘의 I 과 II를 실행하면 얻어지는 AND-OR 그래프이다. ?는 퍼지 명제의 확신도를 알지 못한다는 것을 표시한다. 그림 7은 알고리즘 III에서 STARTS에 있는 시작노드 d_1 과 d_7 의 확신도를 사용자가 각각 (0.70 0.80 0.90)와 (0.80 0.90 1.00)로 제공하는 것으로 가정한 후, 각 노드의 확신도를 계산한 AND-OR 그래프이다. p_6 의 확신도는 p_1 과 p_7 의 확신도, 진리값 그리고 가중값을 고려하여 평가한다. $\tilde{\tau}_6=[(\tilde{\tau}_1 \otimes \tilde{\omega}_1 \oplus \tilde{\tau}_7 \otimes \tilde{\omega}_7) \odot (\omega_1 \oplus \omega_7)] \otimes \tilde{\beta}_6=[((0.70 \ 0.80 \ 0.90) \otimes (0.4 \ 0.5 \ 0.6) \oplus (0.80 \ 0.90 \ 1.00) \otimes (0.6 \ 0.7 \ 0.8))] \odot ((0.4 \ 0.5 \ 0.6) \oplus (0.6 \ 0.7 \ 0.8))] \otimes (0.80 \ 0.90 \ 1.00)=(0.54 \ 0.86 \ 1.34) \otimes (0.80 \ 0.90 \ 1.00)=(0.43 \ 0.77 \ 1.34)$. 목표노드 p_4 의 확신도는 세 개의 서로 다른 추론 통로를 통해 ($p_4, (0.37 \ 0.54 \ 0.77)$), ($p_4, (0.48 \ 0.68 \ 0.90)$), ($p_4, (0.34 \ 0.69 \ 1.34)$)를 얻는다. p_4 의 확신도는 $(0.37 \ 0.54 \ 0.77) \odot (0.48 \ 0.68 \ 0.90) \odot (0.34 \ 0.69 \ 1.34)=(0.48 \ 0.69 \ 1.00)$ 이 된다. 퍼지 숫자의 세 번째 값은 1.34이지만 최대값이 1.0을 넘을 수 없기 때문에 세 번째 값은 1.0이 된다.

6. 결론

본 논문에서는 지식베이스에 있는 퍼지 생성규



〈그림 6〉 생성된 AND-OR 그래프



〈그림 7〉 후진추론과정의 AND-OR 그래프

칙을 가중 퍼지 페트리네트로 표현하고 여기에서 사용하기 위한 가중 퍼지 후진추론 알고리즘을 제안하였다. 규칙기반시스템의 지식베이스에 있는 규칙을 가중 퍼지 페트리네트로 모형화할 때 퍼지 생성규칙의 확신도와 규칙에 나타나는 퍼지 명제의 퍼지 진리값을 기존의 퍼지 집합이 아닌 퍼지 숫자로 표현한다. 그리고 규칙에 나타나는 퍼지 명제의 중요도를 나타내는 가중값도 퍼지 숫자로 표현한다. 확신도와 퍼지 진리값을 표현할 때 퍼지 숫자를 사용하기 때문에 단순히 실수로 표현하는 기존의 방법보다 더 유연성을 가지게

된다. 가중 퍼지 후진추론 알고리즘에서는 문제 해결에 사용하는 지식의 중요도인 가중값을 퍼지 숫자로 표현할 수 있으므로 더 유연하고 지적인 방법으로 추론이 가능하다.

참고 문헌

[1] Chen, Shyi-Ming, "Weighted Fuzzy Reasoning using Weighted Fuzzy Petri Nets." IEEE Trans. KED, Vol. 14, No. 2, March/april, pp. 386-397, 2002.

- [2] Chen, Shyi-Ming, "Fuzzy Backward Reasoning Using Fuzzy Petri Nets." IEEE Trans. on SMC-Part B: Cybernetics, Vol. 30, No. 6, pp. 864-856, 2000.
- [3] Chen, Shyi-Ming, Ke, J., and Chang, J., "Knowledge Representation Using Fuzzy Petri-nets," IEEE Trans. on KDE, Vol. 2, No. 3, Sep., pp. 311-319, 1990.
- [4] Garg, M. L., Ahson, S. I., and Gupta, D. V., "A Fuzzy Petri-nets for Knowledge Representation and Reasoning," Information Processing Letters, 39, pp. 165-171, 1992.
- [5] Kaufmann, A., and Gupta, M. M., Fuzzy mathematical Models in Engineering and management Sciences, New York: Elsevier, 1988.
- [6] Konar, A., and Mandel, A. K., "Uncertainty Management in Expert Systems Using Fuzzy Petri Nets," IEEE Trans. on KDE, Vol. 8, No. 1, pp. 96-105, 1996.
- [7] Looney, G. C., "Fuzzy Petri Nets for Rule-based Decision Making," IEEE Trans. on SMC, Vol. 18, No. 1, Jan./Feb., 1988.
- [8] 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., pp. 666-667, 1998.
- [9] Murata, T., "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, Vol. 77, No. 4, April, pp. 541-580, 1989.
- [10] Sheng-Ke Yu, "Comments on Knowledge Representation Using Fuzzy Petri Nets," IEEE Trans. on KDE, Vol. 7, No. 1, Feb., pp. 190-192, 1995.
- [11] Peterson, J. L., Petri Net Theory and the Modeling of Systems, Prentice-hall, 1981.
- [12] Zadeh, L. A., "Fuzzy Sets," Information and Control 8, pp. 338-353, 1965.
- [13] 조상엽, 김기태, "퍼지 페트리네트를 이용한 퍼지 생성규칙의 표현," 한국정보과학회 논문지, 제21권, 제2호, pp. 298-306, 1994, 2.
- [14] 조상엽, "퍼지 생성규칙을 위한 퍼지 페트리네트표현에서 후진추론," 한국정보처리학회 논문지, 제5권 제4호, pp. 1047-1054, 1998.
- [15] Arnould, T., and Tano, S., "Interval-valued Fuzzy Backward Reasoning," IEEE Trans. Fuzzy Systems, Vol. 3, pp. 425-437, 1995.
- [16] Shyi-Ming Chen, "A Weighted Fuzzy Reasoning Algorithm for Medical Diagnosis," Decision Support Systems 11, pp. 37-43, 1994.
- [17] Shyi-Ming Chen, "A New Approach to Handling fuzzy Decision making Problems," IEEE Trans. SMC., Vol. 18, No. 6, pp. 1012-1016, 1988.

◎ 저 자 소 개 ◎



조 상 업

1986년 한남대학교 전자계산학과(공학사)
1988년 중앙대학교 대학원 전자계산학과(이학석사)
1993년 중앙대학교 대학원 전자계산학과(공학박사)
1995~현재 청운대학교 인터넷컴퓨터학과 교수
관심분야 : 인공지능, 퍼지이론, 페트리네트 응용
E-mail : sycho@mail.chungwoon.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 : 광대역통신, 멀티미디어통신, 인터넷보안
E-mail : delee@mail.chungwoon.ac.kr