

퍼지생성규칙을 위한 퍼지페트리네트표현에서 후진추론

조 상 업†

요 약

본 논문에서는 퍼지생성규칙을 표현한 퍼지페트리네트에서 사용할 수 있는 후진추론 알고리즘을 제안한다. 퍼지페트리네트표현은 단사추론 시스템을 모형화하거나 퍼지추론엔진을 구현하는데 사용할 수 있다.

본 논문이 제안한 알고리즘은 단순히 min과 max 계산만을 하는 기존의 알고리즘과는 달리 퍼지생성규칙의 전제부와 결론부에 퍼지개념의 유무에 따라 적절한 믿음값평가함수를 사용하여 보다 더 인간적인 추론을 한다. 후진추론 알고리즘은 목표노드에서 시작노드까지의 후진추론 통로를 구한 후에 믿음값평가함수를 이용하여 목표노드의 믿음값을 구한다.

Backward Reasoning in Fuzzy Petri-net Representation for Fuzzy Production Rules

Sang-Yeop Cho†

ABSTRACT

In this paper, we propose a backward reasoning algorithm which can be utilized in the fuzzy Petri-net representation representing fuzzy production rules. The fuzzy Petri-net representation can be used to model a approximate reasoning system and implement a fuzzy inference engine.

The proposed algorithm, which uses the proper belief evaluation functions according to fuzzy concepts in antecedentes and consequents of fuzzy production rules, is more closer to human intuition and reasoning than other methods. This algorithm generates the backward reasoning path from the goal to the initial nodes and evaluates the belief value of the goal node using belief evaluation functions.

1. 서 론

실세계에서 사람들이 사용하는 지식을 컴퓨터 상에 표현하기 위한 여러 가지 지식표현 방법이 연구되고 있다. 이러한 지식표현 방법 중에 생성규칙이 많이 사용

되고 있다. 그러나 생성규칙은 애매한 지식을 표현하기에는 많은 어려움이 있다. 생식규칙내의 애매함을 표현하기 위해 사용할 수 있는 이론적인 기반이 퍼지이론이다. 퍼지이론을 사용하여 애매함을 표현한 생성규칙을 퍼지생성규칙이라고 한다.

규칙기반시스템에서 지식을 표현하기 위해 사용하는 생성규칙의 모형화 도구로 사용할 수 있는 것이 페트리네트이다[12][13]. 페트리네트를 이용하여 애매함을 표현할 수 있도록 확장한 모형화 도구가 퍼지페트리네

* 본 논문의 연구는 충남산업대학교의 교내 연구비를 일부지원받아 수행됨.

† 상 화 원 : 충남산업대학교 인공지능학과
논문접수 : 1997년 11월 17일, 심사완료 : 1998년 3월 16일

트이다. 그러므로 애매한 지식을 포함하는 퍼지생성규칙을 모형화하는데 사용할 수 있는 적절한 도구는 퍼지 페트리네트가 된다[1][2][4][5][8].

지식표현 분야에서 애매한 지식을 표현하기 위해 퍼지페트리네트를 사용한 연구로는 [1], [2], [4], [6], [8], [10] 등이 있다. 이러한 연구에서는 애매한 지식을 포함하고 있는 퍼지생성규칙에 대한 퍼지페트리네트표현과 이 표현에서 사용할 수 있는 추론방법들을 기술하고 있다. 이러한 연구에서 제안한 추론 방법은 퍼지페트리네트가 본연적으로 가지고 있는 하향식 정보흐름을 이용하는 전진추론이다.

그러나 규칙기반시스템의 응용영역에 따라서는 후진 추론이 전진추론보다 더 효율적인 경우가 있다. 본 논문에서는 이러한 상황에서 사용할 수 있는 후진추론 알고리즘을 제안한다. 이 알고리즘은 목표노드에서 시작 노드로 후진추론 통로를 구한 뒤 목표노드의 믿음값을 구하는 후진추론 알고리즘을 제안한다.

2 장에서는 본 논문에서 사용하는 퍼지생성규칙의 네 가지 형을 소개한다. 3 장에서는 퍼지페트리네트표현에서 사용하는 퍼지추론규칙을 설명한다. 4 장에서는 퍼지생성규칙에 대한 퍼지페트리네트표현을 기술한다. 5 장에서는 본 논문에서 제안하는 후진추론 알고리즘을 설명한다. 6 장에서는 본 연구의 결론과 연구과제를 기술한다.

2. 퍼지생성규칙

애매한 지식을 표현하는데 사용하는 지식표현 방법으로는 퍼지생성규칙을 많이 사용한다. 퍼지생성규칙은 퍼지 명제간의 퍼지관계를 기술하는 방법이라고 할 수 있다. 이러한 퍼지생성규칙의 간단한 형식은 다음과 같다.

$$\text{Rule}_i: \text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \quad (\beta_i) \quad (1)$$

여기에서 Rule_i는 퍼지생성규칙의 이름이다. $1 \leq i \leq n$. n 은 퍼지생성규칙의 수이다. $x \text{ is } A$ 와 $y \text{ is } B$ 는 퍼지 변수를 갖는 퍼지명제이다. IF $x \text{ is } A$ 와 THEN $y \text{ is } B$ 는 각각 퍼지생성규칙의 전제부와 결론부이다. $\beta_i \in [0, 1]$ 인 퍼지생성규칙의 믿음값(belief)이다. (1)은 IF d_1 THEN d_r (β_i)와 같이 간단히 표기하기도 한다. 여기에서 d_1 와 d_r 는 퍼지명제이다.

퍼지생성규칙을 사용하는 분야로는 제어분야와 지식

공학분야로 나누어 볼 수 있다. 각 분야에서 사용하는 퍼지생성규칙의 분류는 [1], [2], [4], [6], [7], [14] 등에서 찾아 볼 수 있다. 본 논문에서 사용하는 지식공학분야의 퍼지생성규칙의 분류는 <표 1>과 같다.

<표 1> 퍼지생성규칙의 형
<Table 1> Types of fuzzy production rules

형	퍼지생성규칙
1형	IF d_1 THEN d_r (β_i)
2형	IF d_1 THEN $d_{r1} \wedge d_{r2} \wedge \dots \wedge d_{rn}$ (β_i)
3형	IF $d_{11} \wedge d_{12} \wedge \dots \wedge d_{1m}$ THEN d_r (β_i)
4형	IF $d_{11} \wedge d_{12} \wedge \dots \wedge d_{1m}$ THEN $d_{r1} \wedge d_{r2} \wedge \dots \wedge d_{rn}$ (β_i)

3. 퍼지추론

이 절에서는 퍼지생성규칙으로 표현된 지식에서 사용하는 퍼지추론방법인 합성추론규칙(compositional rule of inference)을 설명한다. 그리고 추론시에 필요한 결론의 믿음값을 계산하기 위한 믿음값평가함수(belief evaluation function), 합성퍼지생성규칙의 믿음값을 계산하는 믿음값 합성함수(belief composition function) 그리고 같은 결론의 믿음값을 계산할 때 사용하는 믿음값결합함수(belief combine function)를 설명한다[2][3][4][5][9].

3.1 믿음값평가함수

(1)을 이용한 퍼지추론은 합성추론규칙을 사용한다. 합성추론규칙의 형식은 다음과 같다.

$$\begin{aligned} \langle \text{규칙} \rangle : & \text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \quad (\beta_r) \\ \langle \text{사실} \rangle : & \quad x \text{ is } A' \quad (\beta_f) \end{aligned}$$

$$\langle \text{결론} \rangle : \quad y \text{ is } B' \quad (\beta_c) \quad (2)$$

(2)에서 A, A', B 그리고 B' 는 퍼지개념을 가지고 있는 퍼지변수 또는 퍼지집합이다. β_r, β_f 그리고 β_c 는 각각 <규칙>, <사실> 그리고 <결론>의 믿음값이다.

<결론>의 믿음값 β_c 는 믿음값평가함수 $\beta : (\beta_f, \beta_r) \rightarrow \beta_c$ 를 이용하여 계산한다. 퍼지생성규칙에 퍼지 변수의 유무에 따라 <표 2>의 믿음값평가함수를 이용하여 <결론>의 믿음값 β_c 를 계산한다. 첫째, 전제부

가 비퍼지이고 결론부는 퍼지 또는 비퍼지한 경우는 《사실》과 《규칙》의 믿음값을 곱하여 구한다. 둘째, 전제부와 결론부가 모두 퍼지한 경우는 규칙의 전제부와 결론부의 관계를 퍼지관계 R_{sg} 를 구성한 후 《사실》의 믿음값과 곱하여 구한다. R_{sg} 는 다음과 같다.

$$R_{sg} = (A \times V \rightarrow U \times B) \cap (\neg A \times V \rightarrow U \times \neg B)$$

$$= \min_{U \times V} [\mu_A(u) \rightarrow_s \mu_B(v), 1 - \mu_A(u) \rightarrow_g \mu_B(v)] / (u, v).$$

$$\mu_A(u) \rightarrow_s \mu_B(v) = \begin{cases} 1 & \mu_A(u) \leq \mu_B(v) \\ 0 & \mu_A(u) > \mu_B(v). \end{cases}$$

$$\mu_A(u) \rightarrow_g \mu_B(v) = \begin{cases} 1 & \mu_A(u) \leq \mu_B(v) \\ \mu_B(v) & \mu_A(u) > \mu_B(v). \end{cases}$$

전제부와 결론부가 모두 퍼지한 경우에 퍼지관계를 R_{sg} 로 표현한다. R_{sg} 는 인간의 직관과 추론과정을 가장 적절하게 표현할 수 있는 퍼지관계이다[9]. 셋째, 전제부가 퍼지하고 결론부는 비퍼지한 경우는 《사실》과 《규칙》의 믿음값에 유사도(similarity) S를 곱하여 구한다. 유사도는 다음과 같다[5].

$$S = \begin{cases} P(A | A') & N(A | A') > 0.5 \\ (N(A | A') + 0.5) * P(A | A') & N(A | A') \leq 0.5. \end{cases}$$

$$P(A | A') = \max_{u \in U} (\min(\mu_A(u), \mu_{A'}(u))).$$

$$N(A | A') = 1 - P(\neg A | A').$$

〈표 2〉 믿음값평가함수
(Table 2) Belief evaluation function

		결론부	
		비퍼지	퍼지
전제부	비퍼지	$\beta_c = \beta(\beta_f, \beta_r) = \beta_f * \beta_r$	$\beta_c = \beta(\beta_f, \beta_r) = \beta_f * \beta_r$
	퍼지	$\beta_c = \beta(\beta_f, \beta_r) = \beta_f * \beta_r * S$	$\beta_c = \beta(\beta_f, \beta_r) = \beta_f * R_{sg}$

3.2 믿음값합성함수

전제부와 결론부가 두 개 이상의 퍼지명제로 구성된 합성퍼지생성규칙의 믿음값을 계산하기 위해서 다음과 같은 믿음값합성함수를 사용한다.

$$\beta_c = \beta_{comp}(\beta(\beta_f, \beta_r)) = \min_p (\max_q (\beta(\beta_{fp}, \beta_{rpq})))$$

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

전제부와 결론부에 존재하는 퍼지관계와 비퍼지관계를 모두 고려하므로 단순히 min만을 사용하는 [4]와 [11] 보다 인간이 사용하는 추론과 직관에 더 가깝다.

3.3 믿음값결합함수

규칙기반시스템에서는 추론을 하는 동안 서로 다른 추론 통로를 통해 같은 결론에 도달할 수 있다. 이와 같은 결론은 서로 다른 믿음값을 갖는 결론이므로 믿음값이 적절하게 다시 계산하여야 한다[3]. 이와 같이 믿음값을 다시 계산할 때 사용하는 함수가 믿음값결합함수이다. 믿음값결합함수는 다음과 같다.

$$\beta_c = \beta_{comb}(\beta_{c1}, \beta_{c2}) = \max(\beta_{c1}, \beta_{c2})$$

4. 퍼지페트리네트표현

이 장에서는 퍼지생성규칙을 표현하기 위한 퍼지페트리네트를 설명한다[1][2][4][6][10][11][14]. 퍼지페트리네트는 퍼지이론을 이용하여 애매함을 표현할 수 있도록 페트리네트를 확장하여 얻을 수 있다.

4.1 퍼지페트리네트구조

퍼지페트리네트(fuzzy Petri-net: FPN)의 정의는 다음과 같다.

정의 1: 퍼지페트리네트

$$FPN = (P, T, I, O, \tau, \beta, \lambda)$$

· $P = \{p_1, p_2, \dots, p_n\}$ 은 플레이스(place)의 유한 집합.

- $T = \{t_1, t_2, \dots, t_m\}$ 은 트랜지션(transition)의 유한집합
- $I : T \rightarrow P^\infty$ 은 트랜지션을 입력 플레이스에 사상시키는 입력함수
- $O : T \rightarrow P^\infty$ 은 트랜지션을 출력 플레이스에 사상시키는 출력함수
- $\tau : P \rightarrow [0,1]$ 는 플레이스 내에 있는 토큰을 0과 1사이의 실수로 사상시키는 토큰가중함수
- $\beta : T \rightarrow [0,1]$ 는 트랜지션을 0과 1사이의 실수로 사상시키는 트랜지션가중함수
- $\lambda : T \rightarrow [0,1]$ 는 트랜지션을 0과 1사이의 실수로 사상시키는 트랜지션임계함수
- $P \cap T = \emptyset$

4.2 퍼지페트리네트 그래프

퍼지페트리네트를 시각적으로 표현하는 퍼지페트리네트 그래프는 두 종류의 노드를 갖는 양분된 방향성 그래프(bipartite directed graph)이다. 퍼지페트리네트 그래프 G의 정의는 다음과 같다.

정의 2: 퍼지페트리네트 그래프
 $G = (V_p, V_t, A, \tau)$

- $V_p = \{V_{p1}, V_{p2}, \dots, V_{pn}\}$ 는 원, \bigcirc 으로 표현하는 플레이스 정점(vertex)의 집합
- $V_t = \{V_{t1}, V_{t2}, \dots, V_{tm}\}$ 는 수직선, $|$ 으로 표현하는 트랜지션 정점의 집합
- $A = \{a_1, a_2, \dots, a_k\}$ 는 $a_i \in (V_p \times V_t) \cup (V_t \times V_p)$ 인 아크(arc)의 집합
- τ 는 점, \bullet 으로 표현하는 플레이스 내부의 토큰
- $V_p \cap V_t = \emptyset$

$\tau_i = \tau(p_i)$ 는 플레이스 p_i 에 있는 토큰의 가중값이고, $\beta_j = \beta(t_j)$ 는 트랜지션 t_j 의 입력 플레이스와 출력플레이스의 관계의 강도를 나타내는 가중값이며, $\lambda_j = \lambda(t_j)$ 는 트랜지션 t_j 의 임계값이다.

만일 $\min\{\tau_i \mid \tau_i = \tau(p_i), p_i \in I(t_j)\} > \lambda_j$ 이면, 트랜지션 t_j 는 실행가능(enable)하다. 실행가능한 트랜지션 t_j 는 입력 플레이스에서는 토큰을 제거하고 출력 플레이스에는 한 개씩의 토큰을 출력하면서 실행(fire)하게 된다. 출력 플레이스에 나타나는 토큰의 가중값은 트랜지션의 가중값과 실행하기 전에 입력 플레이스에

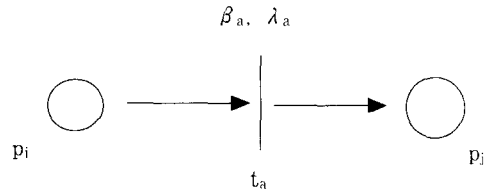
있는 토큰의 가중값을 이용하여 계산한다.

만일 $p_i \in I(t_a)$ 그리고 $p_j \in O(t_a)$ 이라면, p_j 는 p_i 에서 직접 도달가능하다(direct reachable). p_i 에서 직접 도달가능한 플레이스의 집합을 직접도달집합(direct reachability set)이라고 하고 $DRS(p_i)$ 로 표시한다. p_j 가 p_i 에서 직접도달가능하고, p_k 는 p_j 에서 직접도달가능하다면, p_k 는 p_i 에서 도달가능하다(reachable). 이러한 도달관계는 직접도달관계의 반사추이폐포(reflexive transitive closure)가 된다. p_i 에서 도달가능한 플레이스의 집합을 도달집합(reachability set)이라고 하고 $RS(p_i)$ 로 표시한다.

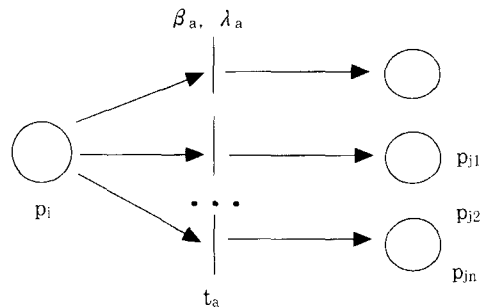
$p_i \in I(t_a)$ 이고 $p_j \in I(t_a)$ 이라면 p_j 를 t_a 에 대한 p_i 의 이웃 플레이스(neighbor place)라고 한다. 이러한 p_j 의 집합을 이웃 플레이스 집합(neighbor place set)이라고 하고 $NPS(p_i)$ 로 표시한다.

4.3 퍼지페트리네트 표현

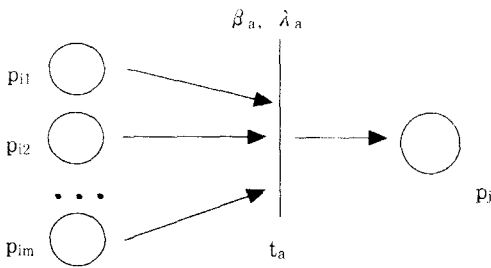
이 절에서는 퍼지생성규칙의 네 가지 형에 대한 퍼지페트리네트 표현을 기술한다. (그림 1), (그림 2), (그림 3) 그리고 (그림 4)는 각각 1형, 2형, 3형 그리고 4형의 퍼지생성규칙을 퍼지페트리네트로 표현한 그



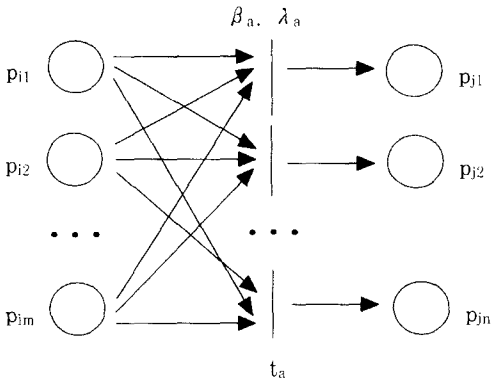
(그림 1) 1형 퍼지페트리네트 표현
 (Fig. 1) Fuzzy Petri-net representation of type 1



(그림 2) 2형 퍼지페트리네트 표현
 (Fig. 2) Fuzzy Petri-net representation of type 2



(그림 3) 3형 퍼지페트리네트표현
(Fig. 3) Fuzzy Petri-net representation of type 3



(그림 4) 4형 퍼지페트리네트표현
(Fig. 4) Fuzzy Petri-net representation of type 4

립이다.

네 가지 형에 대한 퍼지페트리네트표현에서 모든 입력 플레이스에 토큰이 존재하고, $\min\{\tau_i \mid \tau_i = \tau(p_i), p_i \in I(t_a)\} \geq \lambda_a$ 이라는 조건을 만족하면 실행가능해진다. 실행가능한 트랜지션 t_a 가 실행되면 입력 플레이스에서는 토큰을 제거하고 각각의 출력 플레이스에 한개씩의 토큰이 나타나게 된다. 출력 플레이스에 나타난 토큰의 믿음값은 믿음값합성함수 $\tau_j = \beta_{comp}(\beta(\tau_i, \beta_a))$ 를 이용하여 계산한다.

5. 후진추론 알고리즘

본 논문에서 제안하는 추론 알고리즘은 퍼지추론 시스템을 모형화하거나 또는 퍼지추론엔진을 구현하기 위한 퍼지페트리네트표현에서 후진추론 알고리즘으로 사용하기 위한 것이다. 이 알고리즘은 목표노드에서 시작

노드까지의 추론통로를 찾은 후 목표노드의 믿음값을 구한다.

후진추론 알고리즘은 Find_Path와 Compute_Belief로 구성된다. Find_Path 알고리즘은 목표노드에서 시작노드까지의 후진추론 통로를 찾아주는 알고리즘이다. Compute_Belief 알고리즘은 목표노드의 믿음값을 계산하는 알고리즘이다.

5.1 Find_Path 알고리즘

Find_Path 알고리즘은 주어진 퍼지페트리네트 FPN을 이용하여 목표노드에서 시작노드까지 후진추론 통로를 찾아준다. 즉, 원래 주어진 FPN에서 후진추론 경로만을 갖는 축소된 FPN을 출력한다.

이 알고리즘에서 FPN은 퍼지생성규칙을 표현한 퍼지페트리네트이고, GOAL은 목표노드이다. DRS는 퍼지페트리네트에 존재하는 플레이스들의 직접도달집합이며, N은 FPN내의 플레이스의 수이다.

OPEN은 부모노드를 찾기 위해 현재 처리중인 플레이스, 즉 노드의 리스트이다. 현재 처리중인 노드의 부모노드를 찾아서 저장하기 위한 리스트가 PARENTS와 PARENTHSHIP이다. PLACES와 TRANSITIONS는 FPN내의 플레이스와 트랜지션들의 리스트이다. Find_Path 알고리즘이 종료된후에 PLACES와 TRANSITIONS에는 후진통로상에 있는 플레이스와 트랜지션들만 남는다. p_i 와 p_k 는 플레이스이고, t_{ik} 는 플레이스 p_i 와 p_k 사이에 있는 트랜지션을 나타낸다.

Find_Path 알고리즘

입력: FPN, GOAL, DRS, N

출력: 목표노드에서 시작노드까지의 경로

1. OPEN = GOAL
2. repeat until (OPEN == \emptyset)
 - 2.1 PARENTS = \emptyset
 - 2.2 do for each p_k in OPEN
 - 2.2.1 OPEN -= $\{p_k\}$
 - 2.2.2 PLACES $\cup = \{p_k\}$
 - 2.2.3 PARENTHSHIP = \emptyset
 - 2.2.4 do for i from 1 to N
 - 2.2.4.1 if ($p_k \in DSR(p_i)$)
 then PARENTHSHIP $\cup = \{p_i\}$
 TRANSITIONS $\cup = \{t_{ik}\}$

2.2.5 PARENTS U = PARENTHSHIP

2.3 OPEN = PARENTS

3. return PLACES and TRANSITIONS

5.2 Compute_Belief 알고리즘

Compute_Belief 알고리즘은 Find_Path 알고리즘이 찾아낸 축소된 퍼지페트리네트를 이용하여 목표노드의 믿음값을 계산한다. 이렇게 계산된 목표노드의 믿음값 τ_j 가 출력이 된다.

이 알고리즘에서 STARTS는 축소된 퍼지페트리네트 내에 있는 시작노드의 리스트이고, TRANSITIONS는 Find_Path 알고리즘이 찾아낸 축소된 퍼지페트리네트 내에 있는 트랜지션의 리스트이다.

TOKENPLACES는 축소된 퍼지페트리네트 내에서 토른을 가지고 있는 플레이스들의 리스트이다. ENABLES는 현재 실행가능한 트랜지션의 리스트이다. p_i 와 p_j 는 플레이스이고, t_a 는 트랜지션이며 τ_j 는 플레이스 p_j 의 믿음값이다. τ_j^{old} 는 다른 통로를 통해서 먼저 계산되어 TOKENPLACES내에 이미 존재하는 플레이스 p_j 의 믿음값이다.

Compute_Belief 알고리즘

입력: STARTS, TRANSITIONS

출력: 목표노드의 믿음값 τ_j

1. TOKENPLACES = STARTS
 2. repeat until (TRANSITIONS == \emptyset)
 - 2.1 do for each t_a in TRANSITIONS
 - 2.1.1 if $I(t_a) \subseteq$ TOKENPLACES

then ENABLES U = { t_a }

TRANSITIONS -= { t_a }
 - 2.2 do for each t_a in ENABLES
 - 2.2.1 if $\min\{\tau_i \mid \tau_i = \tau(p_i), p_i \in I(t_a)\} > \lambda_a$

then do for each p_j in $O(t_a)$
 - 2.2.1.1 $\tau_j = \beta_{comp}(\beta(\tau_i, \beta_a))$
 - 2.2.1.2 if $p_j \in$ TOKENPLACES

then $\tau_j = \beta_{comb}(\tau_j, \tau_j^{old})$
 - 2.2.1.3 TOKENPLACES U = { p_j }
 - 2.2.2 ENABLES -= { t_a }
3. return τ_j

5.3 예

$d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9$ 그리고 d_{10} 은 명제이고, 임계값 $\lambda_a = 0.2$ 이고, d_1 과 d_2 의 믿음값은 각각 0.9이며 목표노드는 d_9 라고 가정하자. 그리고 퍼지생성규칙은 다음과 같다고 가정한다.

Rule₁: IF $d_1 \wedge d_3$ THEN $d_5 \wedge d_6$ ($\beta=0.9$)

Rule₂: IF d_2 THEN $d_3 \wedge d_4$ ($\beta=0.8$)

Rule₃: IF $d_4 \wedge d_8$ THEN d_{10} ($\beta=0.9$)

Rule₄: IF d_5 THEN d_7 ($\beta=0.7$)

Rule₅: IF d_6 THEN d_7 ($\beta=0.8$)

Rule₆: IF d_7 THEN d_9 ($\beta=0.9$)

퍼지생성규칙을 퍼지페트리네트로 표현하면 (그림 5)와 같이 된다. (그림 5)에 대한 직접도달집합은 <표 3>과 같다.

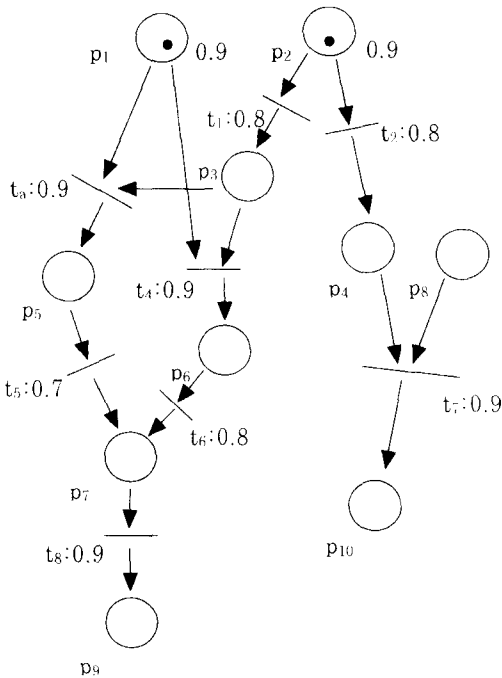
Find_Path 알고리즘은 (그림 5)의 퍼지페트리네트와 목표노드를 입력으로 받아, <표 3>의 직접도달집합을 이용하여 목표노드에서 시작노드까지 후진방향으로 부모관계를 갖는 플레이스를 반복적으로 찾아 후진추론 통로를 찾아낸다. 그러므로 원래 퍼지페트리네트에서 후진추론통로만을 분리하여 얻은 더 작은 차원의 축소된 퍼지페트리네트가 Find_Path의 출력이 된다.

(그림 5)의 퍼지페트리네트에서 목표노드는 p_9 이다. p_9 의 부모노드는 p_7 이고, 이들 사이에 있는 트랜지션은 t_8 이 된다. p_7 의 부모노드는 p_5 와 p_6 이 되고, 트랜지션은 t_5 와 t_6 이 된다. 이와 같은 방법으로 계속 알고리즘을 적용하면, 플레이스 p_4, p_8, p_{10} 그리고 트랜지션 t_2, t_7 을 제거한 축소된 퍼지페트리네트가 Find_Path의 실행결과가 된다. 축소된 퍼지페트리네트에서의 입력과 출력 플레이스를 보면 <표 4>와 같다.

Compute_Belief 알고리즘은 Find_Path의 실행결과인 축소된 퍼지페트리네트와 토른이 주어진 시작노드들을 입력으로 받는다. Compute_Belief는 축소된 퍼지페트리네트 상에서 <표 4>의 축소된 퍼지페트리네트의 입출력 플레이스의 집합을 이용하여 실행가능한 트랜지션을 모두 찾아 순차적으로 실행시킨다. 그리고 추론통로상에 나타나는 모든 플레이스의 믿음값을 믿음값 합성함수와 믿음값결합함수를 이용하여 순차적으로 계산한다. 이렇게 반복실행하여 목표노드의 믿음값을 이 알고리즘의 실행결과로서 출력한다.

축소된 퍼지페트리네트에서 믿음값의 계산은 시작노

스에서 시작한다. 믿음값 계산은 편의상 규칙의 전제부와 결론부가 모두 비퍼지하다고 가정한다. 플레이스 p_1 에 토큰이 존재하므로 트랜지션 t_1 이 실행가능해진다. 실행가능한 t_1 이 실행되면 p_3 의 믿음값은 0.72가 된다. 다음으로 트랜지션은 t_3 와 t_4 실행가능하므로 p_5 와 p_6 의 믿음값은 각각 0.81이 된다. 다음으로 t_5 와 t_6 이 실행가능하게된다. t_5 를 통한 p_7 의 믿음값은 0.57이고 t_6 을 통한 p_7 의 믿음값은 0.65가 되므로 p_7 의 믿음값은 믿음값합성함수에 의해 0.65가 된다. 그리고 목표노드 p_9 의 믿음값은 0.59가 된다.



(그림 5) 퍼지페트리네트표현
(Fig. 5) Fuzzy Petri-net representation

<표 3> (그림 5)의 직접도달집합
<Table 3> Direct reachability sets for (Fig. 5)

플레이스 p_i	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
$DRS(p_i)$	p_5	p_3	p_5	p_{10}	p_7	p_7	p_9	p_{10}	\emptyset	\emptyset

<표 4> 축소된 퍼지페트리네트의입출력 플레이스의 집합
<Table 4> Sets of input and output places in the reduced fuzzy Petri-net

트랜지션 t_i	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
$I(t_i)$	p_2	X	p_1	p_1	p_5	p_6	X	p_7
$O(t_i)$	p_3	X	p_5	p_6	p_7	p_7	X	p_9

6. 결 론

본 논문에서는 퍼지페트리네트표현에서 사용할 수 있는 후진추론 알고리즘을 제안하였다. 제안한 후진추론 알고리즘은 응용영역의 지식을 퍼지페트리네트로 표현하고, 문제해결을 위한 추론방향이 전진추론방향보다 후진추론방향이 보다 더 적합한 응용영역에서 효과적으로 사용할 수 있다. 그리고 후진추론 알고리즘의 추론 시간은 축소된 퍼지페트리네트에 나타나는 플레이스와 트랜지션의 수에 비례한다.

후진추론 알고리즘에서 사용하는 믿음값을 계산하는 함수들은 기존의 연구처럼 단순히 min연산만을 사용하지 않고, 규칙의 전제부나 결론부에 퍼지개념의 유무에 따른 적절한 계산함수를 적용하므로 사람들이 문제해결을 해결하기 위해 사용하는 추론과 보다 더 유사하다.

본 논문이 제안한 후진추론 알고리즘은 단일 프레임 서상에서 개발되었으므로 퍼지페트리네트의 장점 중의 하나인 병렬성을 이용할 수 있는 병렬 추론 알고리즘의 개발이 필요하다. 그리고 지식표현에 사용한 퍼지명제의 제한된 표현성을 극복하기 위해 술어논리 등과 같은 보다 유연한 지식표현에 대한 퍼지페트리네트표현과 추론 메카니즘의 연구가 필요하다.

참 고 문 헌

- [1] 전명근, 변중남, "Fuzzy Petri Nets를 이용한 퍼지추론 시스템의 모델링 및 추론기관의 구현," 전자공학회논문지, 제29권, 제7호, 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., pp311-319, 1990.

[5] Leung, K. S., and Lam, W., "Fuzzy Concepts in Expert Systems." IEEE Computer, Sep., pp43-56, 1988.

[6] Liu, N. K., and Dillon, T. S., "Modeling Uncertainty in Expert Sytems", PRICAI-92, pp610-616, 1992.

[7] 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.

[8] Looney, G. C., "Fuzzy Petri Nets for Rule-based Decision Making." IEEE Trans. on SMC, Vol. 18, No. 1, Jen./Feb., 1988.

[9] Fukami, S., Mizumoto, M., and Tanaka, K., "Some Considerations on Fuzzy Conditional Inference," Fuzzy Sets and Systems, Vol. 4, pp243-273, 1980.

[10] 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.

[11] Konar, A., and Mandal, A. K., "Uncertainty Management in Expert Systems Using Fuzzy Petri Nets," IEEE Trans. on KDE, Vol. 8, No. 1, pp96-105, 1996.

[12] Murata, T., "Petri Nets: Properties, Analysis and Applications." Proceedings of the IEEE, Vol. 77, No. 4, April, pp541-580, 1989.

[13] Peterson, J. L., Petri Net Theory and the

Modeling of Systems, Prentice-hall, 1981.

[14] Sheng-Ke Yu, "Comments on Knowledge Representation Using Fuzzy Petri Nets," IEEE Trans. on KDE, Vol. 7, No. 1, Feb., pp190-192, 1995.

조 상 업

1986년 한남대학교 전자계산학 졸업(공학사)
 1988년 중앙대학교 대학원 전자계산학과(이학석사)
 1993년 중앙대학교 대학원 전자계산학과(공학박사)
 1993년~1995년 중앙대학교 컴퓨터소프트웨어 연구소
 객원 연구원
 1995년~1997년 충남산업대학교 전자계산학과 전임강사
 1998년~현재 충남산업대학교 인공지능학과 조교수
 관심분야 : 인공지능, 전문가시스템, 퍼지이론