

Fuzzy Petri Nets를 이용한 퍼지 추론 시스템의 모델링 및 추론기관의 구현

(A Model with an Inference Engine for a Fuzzy Production System Using Fuzzy Petri Nets)

全 命 根*, 卞 增 男*

(Myung Geun Chun and Zeung Nam Bien)

要 約

본 논문에서는 규칙기반 시스템의 일반적인 모델인 퍼지 추론 시스템을 모델링하는 방법과 그것의 추론기관을 구현하는 방법을 퍼지 페트리 넷이라는 개념을 이용하여 제시하고자 한다. 퍼지 페트리 넷이 퍼지 추론 시스템을 모델링하는데 사용되었으며, 부정확한 지식을 이용한 추론을 가능하게 하는 추론기관의 구현에도 사용되었다. 부정확한 지식에 대한 처리로, 추론 규칙속에 포함된 애매함을 나타내기 위해서는 퍼지논리가 사용되었으며, 전문가로부터 주어지는 규칙에 대한 불확실함의 정도를 확신도의 개념을 이용하여 나타내었다. 또한 실제적인 상황에서 요구되는 실시간 추론을 위하여 병렬적인 추론이 가능하도록, 퍼지 페트리 넷을 행렬로 변환하는 방법을 제시하고 이를 이용하여 Mamdani의 추론방법을 사용했을 경우에 간단한 대수식으로 표현되는 추론기관의 구현방법을 제시하였다.

Abstract

As a general model of rule-based systems, we propose a model for a fuzzy production system having chaining rules and an inference engine associated with the model. The concept of so-called 'fuzzy petri nets' is used to model the fuzzy production system and the inference engine is designed to be capable of handling inexact knowledge. The fuzzy logic is adopted to represent vagueness in the rules and the certainty factor is used to express uncertainty of each rules given by a human expert. Parallel inference schemes are devised by transforming Fuzzy Petri nets to matrix formula. Further, the inference engine mechanism under the Mamdani's implication method can be described by a simple algebraic formula, which makes real time inference possible.

I. 서 론

컴퓨터를 이용하여 전문가의 지식을 효율적으로 처리하기 위한 지식표현 방법으로 추론시스템(production system), 의미망(semantic nets), 프레임(frame), 명

제논리학(propositional calculus), 술어논리학(predicate calculus) 등의 여러가지 지식표현 방법들이 제시되었으며^[1, 2, 3] 이들 각각에 대한 특성 및 서로간의 비교는 [3]에 잘 정리되어 있는 바와 같다. 그렇지만 앞서 제시된 지식표현 방법들에서 더욱 더 요구되는 것은 인간의 지식속에 포함되어 있는 부정확함(inexactness)에 대한 표현방법이다. 여기서 부정확함(inexactness)이란 애매함(vagueness)과 불확실성(uncertainty)을 포함하는

* 正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)
接受日字 : 1992年 1月 21日

것으로, 애매함은 인간의 언어 속에 포함된 “조금 크다” 혹은 “매우 작다” 등과 같이 명확한 경계를 갖지 않는 애매한 양으로 표현되는 경우에 발생하며, 불확실성은 어떠한 규칙이나 사실에 대하여 절대적인 참값을 갖지 않는 경우에 발생하는 것으로 나누어 생각할 수 있다^[4].

불확실성을 나타내기 위한 방법으로는 Bayesian 방법, certainty factor방법, Dempster-Shafer 방법 등이 있다^[4, 5]. Bayesian 방법에서는 불확실한 조건에 대한 주관적인 척도로 확률이 사용되며, certainty factor방법에서는 사실이나 규칙에 대한 확신에 대한 척도를 0과 1사이의 값으로 나타내고 있으며, Dempster-Shafer 방법에서는 어떤 사실에 대한 증거(evidence) 및 무지(ignorance)의 정도를 표현함으로써 불확실성을 나타내고 이를 처리하고 있다. 이들 중 Bayesian 방법 및 Dempster-Shafer방법 등은 각각 조건 확률 분포함수(conditional probability density function)와 확신함수(belief function) 등을 구하는 것이 쉽지 않고 실제적용에 있어서 계산량이 많은 단점이 있으므로 간편하고 빠른 계산이 가능한 certainty factor방법이 많이 사용되고 있다. 반면에 인간의 지식을 표현하는데 많이 사용되는 언어변수(linguistic variable)등에 포함된 애매함을 나타내기 위해서는 퍼지논리가 널리 이용되고 있다^[6].

따라서, 본 논문에서 대상으로 하고 있는 퍼지 추론 시스템(fuzzy production system)에서는 일반적인 추론 시스템(production system)에서 “If left-hand-side(LHS), then right-hand-side(RHS)”과 같이 주어지는 추론 규칙(production rule)에서 LHS와 RHS에 퍼지 개념이 들어 있고 추론규칙기에 대한 확신도가 주어진 일반적인 형태의 퍼지 추론규칙(fuzzy production rule)을 다루고자 한다. 그렇지만, 이러한 추론규칙을 이용한 지식표현 방법은, Whalen^[15]이 지적했듯이 그것이 단순히 많은 추론규칙들이 모여 서로 연결되어 있는 형태이므로 각 규칙들간의 관계를 이해하기가 쉽지 않고 몇몇 규칙들이 수행되었을 때의 전체 시스템의 동작을 미리 예측할 수 없는 단점이 있다.

이런 점에서 좋은 지식표현 방법이 되기 위해서는 규칙간의 관계를 쉽게 이해할 수 있고, 지식 베이스의 수정이나 새로운 규칙의 추가가 용이하며, 주어진 사실과 지식 베이스로부터 새로운 추론 결과를 빨리 얻을 수 있어야 하며, 추론의 결과에 대한 추론 과정을 설명해 줄 수 있어야 한다^[3].

이러한 요구에 부응할 수 있는 지식표현 방법의 한 형태로 페트리 넷트라는 그래프에 기초를 둔 지식표현 방법이 있다^[7, 8, 9]. 페트리넷트는 동기 및 병렬 구조를 갖는 복잡한 시스템의 모델링에 유용하게 사용될 수 있는 구조를

가지고 있기 때문에 현재 여러 분야에서 널리 쓰이고 있고 또한 logic network을 표현하는데도 쓰이고 있다^[13]. 그렇지만 논리적 구조의 관점에서 보면 본래의 페트리 넷트는 0과 1만의 논리값만을 갖는 제한적인 구조를 갖고 있다. 따라서 이를 극복하고 좀더 일반적인 논리 구조를 갖는 페트리 넷트의 일종으로 퍼지 페트리 넷트(fuzzy Petri nets)가 제안되었다^[10].

Lipp^[10]에 의해서 처음으로 제안된 퍼지 페트리 넷트가 복잡한 산업공정의 모델링에 사용된 반면 [11][12]에서는 지식을 표현하기 위하여 사용되고 있다. 특히 [12]에서는 부정확한 지식표현을 위한 퍼지 페트리 넷트가 [11]에서 제시된 것보다 좀더 체계적인 방법으로 제시되었으며, 추론 과정을 행할 수 있는 알고리즘이 제시되어 있다. 그렇지만, 각각의 퍼지 추론 규칙에 있는 명제에 대한 진리값을 certainty factor에만 기초를 둔 0과 1사이의 실수로 나타내고 이를 이용하여 추론을 행함으로써, 소속함수(membership function)등을 통해 표현되는 애매성을 처리하는 데는 바로 적용할 수 없는 단점이 있다. 또한, 제시된 tree search 방법에 기초를 둔 추론 방법은 여러개의 규칙을 동시에 적용한다든지, 여러개의 명제에 대한 진리값을 모두 알아내고자 하는 경우에 있어서는 효과적이지 못한 단점이 있다.

본 논문에서는, [11] [12]에서 제시된 페트리 넷트가 부정확한 지식의 표현에만 적용가능한 제한적인 구조를 갖는 것을 극복하고자 가중치를 갖는 두개의 마더로 구성되어 있는 일반적인 방향성 그래프의 하나로 퍼지 페트리 넷트를 정의하고 그것을 행렬로 표현할 수 있는 방법을 근접행렬(adjacency matrix)에 기초하여 제시하고자 한다. 또한 이러한 것들을 이용하여 애매함과 각 규칙에 관한 불확실함이 포함된 퍼지 추론 시스템의 모델링 방법과 병렬처리가 가능하며 간단한 대수식으로 표현되는 추론기관의 구현 방법에 대하여 논의한다.

논문의 구성은 다음과 같다. 2장에서는 페트리 넷트에 대한 용어들을 간단히 요약한 후 입력연결 행렬을 분리하는 방법 및 프레이스 근접행렬을 정의하고 애매함을 다루기 위해 퍼지논리에 대한 몇몇 정의를 제시한다. 이어서, 3장에서는 퍼지 페트리 넷트를 정의하고 퍼지 추론 시스템의 모델링 방법 및 추론 기관의 구현 방법에 대해서 논하고 이의 수렴성을 보인 후 간단한 예제를 보인다.

II. 페트리넷트 및 ω -수준 퍼지 구간의 정의

1. 페트리 넷트의 개념 및 용어

페트리 넷트는 $C=(P, T, \alpha, \beta, \mu)$ 와 같이 다섯개의 구성요소로 이루어져 있다. 여기서, $P=\{p_1, p_2 \dots p_n\}$

($n \geq 1$)는 플레이스의 집합이며, $T = \{t_1, t_2, \dots, t_m\}$ ($m \geq 1$)는 트랜지션의 집합이고, $\alpha \subseteq \{P \times T\}$ 와 $\beta \subseteq \{T \times P\}$ 는 각각 방향성 아크의 집합이며, $\mu: P \rightarrow Q$, $Q = \{0, 1, 2, \dots\}$ 는 각 플레이스에 토큰을 할당하는 마킹(marking)을 나타낸다. $\mu(P_i) = \mu_i$ 로 플레이스 P_i 에 있는 토큰을 나타낸다면 $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$ 이다. 페트리 넷 그래프는 2가지의 마디와 아크들로 구성된 방향성 그래프(그림 1 참조)로 원은 (○)은 플레이스를 나타내며, 막대기 (|)는 트랜지션을, 플레이스내의 점 (●)은 토큰을 표시하며, 방향성 아크는 각 플레이스간의 관계를 나타낸다.

트랜지션 t_i 에 대한 입력 플레이스 집합 $I(t_i)$ 및 출력 플레이스 집합 $O(t_i)$ 는 다음과 같이 정의된다.

$$I(t_i) = \{p | (p, t_i) \in \alpha\} \quad (1)$$

$$O(t_i) = \{p | (t_i, p) \in \beta\} \quad (2)$$

또다른 형태로, 페트리 넷은 $C = (P, T, D^+, D^-, \mu)$ 와 같은 대수적인 형태로도 표현이 가능하다 [13]. 여기서, $D^+ = (d_{ij}^+)$ 와 $D^- = (d_{ij}^-)$ 는 각각 입력 연결행렬 및 출력 연결 행렬을 나타내는데, d_{ij}^+ 는 트랜지션 t_j 로부터 그것의 출력 플레이스 P_i 로의 아크 갯수이며, d_{ij}^- 는 플레이스 P_i 로부터 트랜지션 t_j 로의 아크 갯수이다. e_j 를 j 번째 요소를 제외한 모든 요소들이 모두 0인 m -차원의 벡터라 할때 트랜지션 t_j 를 m -차원의 벡터 e_j 로 나타낼 수 있다.

페트리 넷을 이용하여 추론 시스템을 모델링하는 대부분의 논문에서 [11, 12] 플레이스로 조건을 나타내며, 트랜지션으로는 추론 규칙을 나타냄으로서 추론 시스템을 체계적으로 모델링하고 있으므로 본 논문에서도 같은 방식을 따르고자 한다. 이런 경우 특정한 하나의 플레이스와 하나의 트랜지션이 여러개의 아크로 연결되는 경우는 논리적 의미가 없으므로 발생하지 않고 따라서 연결 행렬의 요소들이 0과 1로만 구성되게 된다. 따라서 추론 시스템을 모델링한 페트리 넷은 일종의 ordinary Petri net이다 [13]. 위와 같이 하면, 규칙을 나타내는 트랜지션에 해당하는 입력 플레이스 및 출력 플레이스들이 각각 그 규칙에 대한 전제 조건 및 전제후 조건을 나타내게 되는데, 어떤 플레이스가 토큰을 가지고 있으면 그 플레이스가 나타내는 조건이 참이 되었음을 뜻하게 된다. 이렇게 트랜지션 t_i 의 모든 입력 플레이스에 토큰이 있을 때 트랜지션 t_i 는 인에이블(enable) 되었다고 하며, 인에이블된 플레이스는 점화(firing)될 수 있다.

페트리 넷과 보통의 방향성 그래프(directed graph)와의 차이점은 다음과 같다. 첫째, 페트리 넷은 토큰이라는 요소를 포함하고 있음으로서 시스템의 동적

인 상태를 표현하기가 용이하다. 즉 토큰의 흐름으로 트랜지션의 발화에 기인한 추론시스템의 상태변화를 쉽게 나타낼 수 있다. 둘째 구조 자체가 두가지의 마디로 구성되어 있으므로 AND나 OR 같은 논리연산 등을 쉽게 나타낼 수 있는 점이다. 이러한 성질을 이용하면 현재 많이 사용되고 있는 AND-OR 그래프를 페트리 넷으로 표현이 가능하다.

위의 개념들을 정리하고자 다음의 예를 제시한다.

[예 제 1]: 다음과 같이 다섯개의 조건 C_i ($i=1, 2, \dots, 5$)와 세개의 규칙 R_i ($i=1, 2, \dots, 5$)로 조건으로 주어지는 추론 시스템을 생각해 보자.

R_1 : IF C_1 and C_2 are satisfied, then C_3 will be satisfied.

R_2 : IF C_3 is satisfied, then C_5 will be satisfied.

R_3 : IF C_4 is satisfied, then C_5 will be satisfied.

위와 같이 주어진 추론 시스템에 대한 페트리 넷 그래프 모델이 그림 1에 주어져 있다.

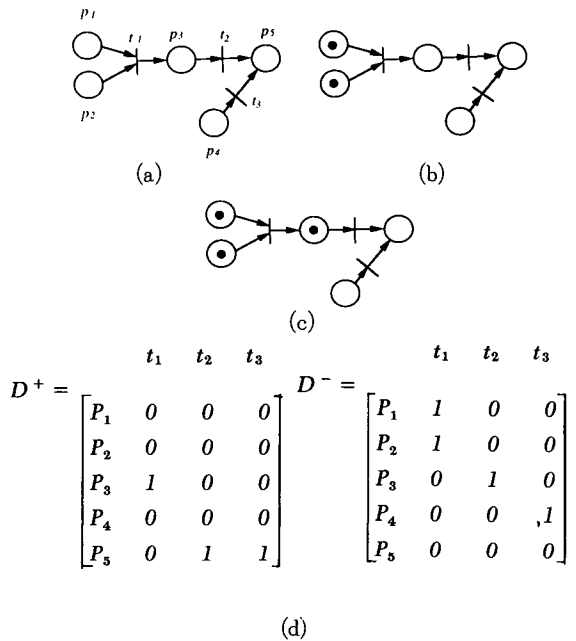


그림 1. 예제 1에 대한 페트리 넷모델

- (a) 주어진 추론 시스템에 대한 페트리 넷 모델
- (b) C_1 과 C_2 가 참일 때
- (c) 규칙 R_1 이 적용된 후의 상태
- (d) 연결 행렬 모델

Fig. 1. An example of a petri net.

- (a) A petti net model of a production system,
- (b) When C_1 and C_2 are satisfied,
- (c) The state after t_1 fires,
- (d) Incidence matrices model.

여기서 p_i 과 t_i 는 각각 C_i 와 R_i 에 해당되며, 구해진 페트리 네트에 대한 연결 행렬은 그림 1(d)에 나타내었다. 그림 1(b)는 C_1 과 C_2 가 모두 만족 되었을 때를 나타낸다. 그러면, 규칙 R_1 이 만족되고 C_3 가 참이 된다. 이때의 상황이 그림 1(c)에 표시되어 있다. 이와같이, 토큰의 흐름으로 이러한 상태 변화를 쉽게 나타낼 수 있으며, 연결행렬을 이용하면 페트리 네트가 쉽게 표현할 수 있음을 그림 1.(d)를 통해서 알 수 있다.

그림 1에서 알 수 있듯이 입력 플레이스사이의 AND나 OR같은 접속사가 입력연결 행렬에 표현되어 있다. 즉, D^- 의 첫번째 컬럼에서 0이 아닌 d_{11} 과 d_{21} 은 p_1 과 p_2 이 AND로 연결되어 있음을 나타낸다. 이는 C_1 과 C_2 이 AND로 연결되어 있음을 나타낸다. 이러한 사실에 기초하여, 입력 연결행렬에 내포된 AND와 OR을 나누기 위하여 두개의 연결 행렬로 나누는 방법을 제시하고자 한다. 그 하나를 conjunctive input incidence matrix (CIIM) 그리고 또 다른 행렬을 disjunctive input incidence matrix(DIM)라고 하고 CIIM은 주어진 입력 연결 행렬 D^- 에 대하여 두개 또는 그 이상의 입력 플레이스를 갖는 트랜지션에 대한 컬럼들은 그대로 두고 나머지 컬럼들은 모두 0으로 만든 행렬이고, DIIM은 주어진 입력 연결 행렬 D^- 에 대하여 하나의 입력 플레이스를 갖는 트랜지션에 대한 컬럼은 그대로 두고 나머지는 모두 0으로 둔 행렬로 정의하면 두 행렬은 다음과 같이 구해진다.

[제안 1]: 주어진 입력 연결행렬 $D^- = (d_{ij}^-)$ 에 대하여 CIIM을 $D_c^- = ((d_{c_{ij}}^-))$, DIIM을 $D_a^- = ((d_{a_{ij}}^-))$ 라 놓고 $x > \theta$ 이면 $f(x - \theta) = 0$ 인 함수 $f(\cdot)$ 를 정의하면,

$$D_c^- = ((d_{c_{ij}}^-)) = (d_{ij}^- \cdot d_j^+)$$
 (3)

$$D_a^- = ((d_{a_{ij}}^-)) = (d_{ij}^-) - ((d_{c_{ij}}^-))$$
 (4)

여기서, $d_j^+ = f(\sum_i d_{ij}^- - 1)$.

(증명) 위에서 $\sum_i d_{ij}^-$ 이 j 번째 트랜지션에 연결된 입력

플레이스의 갯수를 나타내므로

$f(\sum_i d_{ij}^- - 1)$ 를 이용하여 그것이 2이상 인가를 알아낼 수

있으므로 (3)에서 쉽게 CIIM을 구할 수 있으며, (3)과 $D^- = D_c^- + D_a^-$ 의 관계식으로 (4)가 얻어진다.

페트리 네트가 2개의 서로 다른 마디로 연결된 방향성 그래프임을 감안하면, 그래프이론에서 정의된 근접행렬(adjacency matrix)이 두가지로 정의 될 수 있음을 알 수 있다. 하나는 플레이스 마디에 관한 것이고 다른 하나는 트랜지션 마디에 관한것이다. 근접행렬이란 [16], n 개의 마디를 갖는 방향성 그래프에서 $n \times n$ 의 1과 0으로 이루어진

어린 행렬 $A = (a_{ij})$ 로 만약 j 번째 마디로 부터 i 번째 마디로 아크가 연결되어 있을 경우는 $a_{ij} = 1$ 이고 그렇지 않을 경우, $a_{ij} = 0$ 이다. 이를 이용하여 페트리 네트에서의 플레이스 근접행렬(place adjacency matrix) $D^a = (d_{ij}^a)$ 를 플레이스 P_j 와 플레이스 P_i 사이에서 적어도 하나의 트랜지션 t_k 가 존재하여 P_j 에서 t_k 로 다시 t_k 에서 P_i 로 아크가 연결되었을 때 $d_{ij}^a = 1$ 이며, 그렇지 않은 경우는 $d_{ij}^a = 0$ 인 행렬로 정의하면 플레이스 근접행렬 (place adjacency matrix)는 다음과 같이 입력 연결행렬을 이용하여 구할 수 있다.

[제안 2]: 주어진 페트리 네트 $C = (P, T, D^+, D^-, \mu)$ 에 대하여 $D^+ = (d_{ij}^+)$, $D^- = (d_{ij}^-)$ 라 두고 $D^a = (d_{ij}^a)$ 를 플레이스 연결 행렬이라 두면,

$$D^a = D^+ \cdot (D^-)^T$$
 (5)

여기서, 윗 첨자 T 는 행렬의 transpose를 표시하며, 행렬은 곱은 Boolean이다. 즉, 곱셈은 논리연산 AND로 그리고 덧셈은 논리연산 OR로 대체된 것이다.

$$(증명) (5)식에서, d_{ij}^a = \sum_k d_{ik}^+ \cdot d_{jk}^- = \sum_k d_{jk}^- \cdot d_{ik}^+$$

$= 1$ 인 항이 적어도 하나 존재할 때 즉, 어떤 k 에 대하여, $d_{jk}^- = d_{ik}^+ = 1$ 일 때 이는 연결행렬의 정의로 부터 플레이스 p_j 로 부터 트랜지션 t_k 로 아크가 연결되어 있으며, 트랜지션 t_k 로 부터 플레이스 p_i 로 아크가 연결되어 있음을 나타낸다. 정리하면, 플레이스 p_j 와 p_i 가 트랜지션 t_k 를 통하여 근접해 있음을 알 수 있다. 유사하게, $d_{ij}^a = \sum_k d_{ik}^+ \cdot d_{jk}^- = 0$ 은 p_j 와 p_i 사이에 하나의 트랜지션을 통한

근접성이 없음을 나타낸다.

결과적으로, (5)는 페트리 네트에서 하나의 트랜지션을 통한 플레이스 상호간의 근접성을 나타내는 플레이스 근접행렬이다.

2. ω -수준 퍼지 구간 및 퍼지 진리값의 저의

퍼지논리를 이용하여 실제적인 문제를 푸는 경우 퍼지수(fuzzy number)를 효과적이고 체계적으로 기술할 필요가 있다. 이러한 이유로 본 논문에서는 여러 논문에서 많이 사용되고 있는, LR형태의 퍼지수 기술 방식을 따르고자 한다 [14]. 이러한 형태의 퍼지수를 이용하면 계산이 용이할 뿐만 아니라 체계적인 기술을 가능하게 하는 장점이 있다. 이러한 장점들은 후에 설명될 추론기관을 구현하는데 유용하게 사용된다. 그럼, 먼저 평평한 LR형태의 퍼지수(LR-type flat fuzzy number)에 대하여 살펴 보자.

[정의 1]: [14] L (또는 R)을 (1) $L(-x)=L(x)$, (2) $L(0)=1$, (3) L은 $[0, +\infty)$ 에서 단조감소인 성질을 갖는 기준함수(reference function)라 할때 평평한 LR형태의 퍼지수 F는 기준함수 L과 R 그리고 4개의 매개변수 $(m_1, m_2) \in \mathbb{R}^2 \cup \{-\infty, +\infty\}$, α, β , 그리고 다음과 같은 소속함수로 정의된다.

$$\begin{aligned} \mu(x) &= L\left(\frac{m_1-x}{\alpha}\right) \text{ for } x \leq m_1, \alpha > 0 \\ &= 1 \text{ for } m_1 \leq x \leq m_2 \\ &= R\left(\frac{x-m_2}{\beta}\right) \text{ for } x \geq m_2, \beta > 0 \end{aligned} \quad (6)$$

F를 $F=(m_1, m_2, \alpha, \beta)_{LR}$ 로 표기하고자 한다. 만약 $m_1=m_2$ 이면 F는 LR형태의 퍼지수가 된다.

주어진 평평한 퍼지수 F에 대하여 다음과 같이 ω -수준 퍼지구간 LF를 정의한다.

[정의 2]: 주어진 LR형태의 평평한 퍼지수 $F=(m_1, m_2, \alpha, \beta)_{LR}$ 과 0과 1사이의 실수 ω 에 대하여, ω -수준 퍼지구간 LF는 다음과 같이 두개의 매개변수 $a=m_1 - \alpha L^{-1}(\omega)$, $b=m_2 + \beta R^{-1}(\omega)$ 와 소속함수

$$\begin{aligned} \mu_{\omega}(x) &= L\left(\frac{m_1-x}{\alpha}\right) \text{ for } x \leq a, \alpha > 0 \\ &= \omega \text{ for } a \leq x \leq b \\ &= R\left(\frac{x-m_2}{\beta}\right) \text{ for } x \geq b, \beta > 0 \end{aligned} \quad (7)$$

로 정의된다. 앞으로 $LF=(m_1, m_2, \alpha, \beta)_{LF}$ 로 표기하도록 한다. 쉽게 알 수 있듯이, LF는 $\omega=1$ 일 경우 LR형태의 퍼지수가 된다.

그림 2는 평평한 LR형태의 퍼지수와 그에 대응되는 ω -수준의 퍼지구간을 LF를 보여준다.

[정의 3]: L 과 \tilde{L} 을 전체집합 U에 대하여 u_L 와 $u_{\tilde{L}}$ 의 소속함수를 갖는 두개의 퍼지 부분집합이라 하고 다음과 같은 퍼지명제를 생각해보자.

$$x \text{ is } L \quad (8)$$

여기에 대하여 다음과 같은 사실이 들어 왔을때

$$x \text{ is } \tilde{L} \quad (9)$$

(8)에 대한 (9)에서의 퍼지진리값 $\omega \in [0, 1]$ 를 가능성 척도 [18]를 이용하여 다음과 같이 정의한다.

$$\omega = \max_u \min \{ \mu_L(u), \mu_{\tilde{L}}(u) \} \quad (10)$$

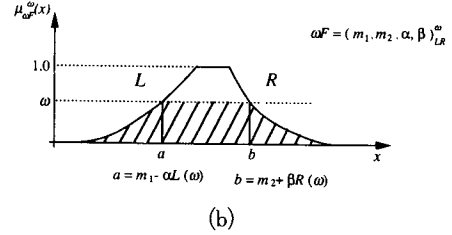
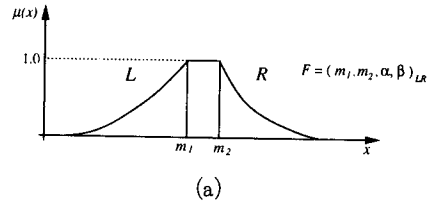


그림 2. 평평한 LR 형태의 퍼지수와 ω -수준 퍼지 구간
(a) 평평한 퍼지수
(b) ω -수준 퍼지 구간

Fig. 2. An LR-type flat fuzzy number and its ω -level fuzzy interval.
(a) An LR-type flat fuzzy number,
(b) An ω -level fuzzy interval

III. 퍼지 추론 시스템과 퍼지 페트리 넷

1. 퍼지 추론 시스템

일반적으로 그림 3과 같이 주어지는 규칙 기반시스템(rule-based system)에 있어서의 추론과정을 살펴보자.

추론기관은 먼저 I/O 부분으로부터 주어진 사실들을 working memory의 초기값으로 삼아 추론을 시작하게 된다. 이를 이용하여 추론기관은 working memory의 내용과 주어진 추론규칙들로부터 각 규칙의 전제조건이 만족되는지의 여부를 점검하게 된다. 만약 주어진 사실이 어떤 규칙의 전제조건을 모두 만족하게 되면 그 규칙을 적용한 후 그 규칙의 전제후조건(post-conditions)들이 새롭게 working memory에 추가되게 된다. 이런식으로 또 다른 규칙들에 대해서 위의 과정을 수행하면서 working memory에 더이상 새로운 내용이 추가되지 않을 때까지 위의 과정을 반복하게 된다. 이렇게 하여 최종적으로 얻어진 working memory의 내용이 추론의 결과로서 I/O 부분으로 보내져서 규칙기반 시스템의 목적에 맞는 어떠한 결정을 내리든지(decision making) 혹은 적절한 제어 출력을 만들게 된다.

본 논문에서 다루고자 하는 시스템을 그림 3과 연관지어 볼때, 지식베이스를 이루는 추론 규칙의 전제부 및 후건부가 평평한 LR-형태의 퍼지수를 취하는 언어변수(linguistic variable)가 포함된 명제들로서 구성되어 있고 각각의 추론규칙은 전문가로부터 지식을 추출할때 주

어진 확신도 $c \in [0, 1]$ 를 가지고 있는 경우이다. 여기에 쓰인 확신도는 그 값이 클때 그 규칙에 대한 확신도가 높음을 나타낸다. 그리고 각각의 퍼지수에 대한 정보, 즉 기준함수나 다른 매개변수에 대한 정보가 I/O 부분에 저장되어 있어서 추론기관에서 주어지는 결과가 제시된 ω -수준의 퍼지 구간의 개념을 이용하여 최종적인 추론 결과로 표현된다.

본 논문에서 사용된 퍼지 추론 시스템을 정리하면 다음과 같이 정의할 수 있다.

[정의 4]: X 를 U_i 상에서 소속함수가 $u_{L_j^i} = ((m_1)_j^i, ((m_2)_j^i, \alpha_j^i, \beta_j^i)_{L_j^i R_j^i} (1 \leq i \leq n, 1 \leq j \leq l_i))$ 로 표현되는 l_i 개의 평평한 LR-형태의 퍼지수 L_j^i 를 갖는 언어변수 $x_i (1 \leq i \leq n)$ 로 이루어진 집합이라 하자. " x_i is L_j^i " 같이 주어지는 퍼지 명제를 d_j^i 로 나타내고 그때의 퍼지 진리값을 ω_j^i 로 놓으면, 퍼지 추론 시스템은 다음과 같이 표현된다.

$$\Psi = (X, \Delta, C, \Phi)$$

여기서, $\Phi = [\omega_{i_1}^1, \dots, \omega_{i_{n-1}}^n, \omega_{i_n}^n]^T$, $\Delta = \bigcup_{i=1}^m \delta_i$ 는 δ_i^1 또는 δ_i^2 의 형태로 주어지는 퍼지 명제 δ_i 의 집합이다.

$$\begin{aligned} \delta_i^1 &= IF d_{j_1}^{i_1} \text{ and } d_{j_2}^{i_2} \dots \text{ and } d_{j_r}^{i_r}, \\ THEN d_{k_2}^{i_2} \dots \text{ and } d_{k_q}^{i_q}; \end{aligned} \quad (11)$$

$$\begin{aligned} \delta_i^2 &= IF d_{j_1}^{i_1} \text{ or } d_{j_2}^{i_2} \dots \text{ or } d_{j_r}^{i_r}, \\ THEN d_{k_2}^{i_2} \dots \text{ and } d_{k_q}^{i_q} \end{aligned} \quad (12)$$

여기서, (i_1, i_2, \dots, i_r) 와 (k_1, k_2, \dots, k_r) 는 $(1, 2, \dots, n)$ 의 subsequence이며, $C = [c_1, c_2 \dots c_m]^T$ 는 퍼지규칙 δ_i 에 대한 주관적인 확신도를 나타내는 $c_i \in [0, 1]$ 로 이루어진 벡터이다.

본 논문에서는 후건부의 퍼지 명제들이 OR로 연결되어 있는 경우에 대해서는 생각하지 않았다. 왜냐하면, 그러한 형태의 규칙에 대해서는 특정사실에 대한 추론의 결과를 얻을 수 없기 때문이다 [11].

2. 퍼지 페트리 네트를 이용한 퍼지 추론 시스템의 모델링

[정의 5]: 퍼지 페트리 네트 FPN는 6개의 구성요소로 이루어져 있다.

$$FPN = (P, T, \alpha, \beta, W, \Omega)$$

여기서, $P = \{p_1, \dots, p_n\} (n \geq 1)$ 는 플레이스의 집합이

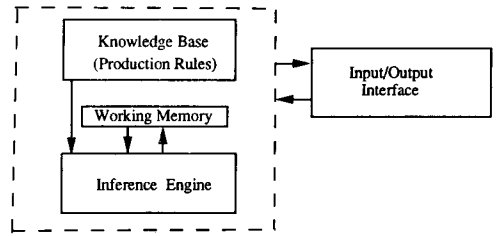


그림 3. 규칙기반 전문가 시스템의 구성요소

Fig.3. Major components of a rule-based expert system.

며, $T = \{t_1, \dots, t_m\} (m \geq 1)$ 는 트랜지션의 집합 $\alpha \in \{P \times T\}$ 와 $\beta \in \{T \times P\}$ 는 방향성 아크의 집합이다. $W: T \rightarrow [0, 1]$ 는 가중함수(weight function)이고 $\Omega: P \rightarrow [0, 1]$ 는 0과 1 사이의 값을 갖는 토른을 마킹하는 것이다.

앞에서 정의된 퍼지 페트리 네트는 보통의 페트리 네트와 다음과 같은 두 가지 점에서 다르다.

- (1) 각각의 트랜지션들은 W 에 의해서 0과 1 사이의 값으로 가중치가 주어져 있다.
- (2) 퍼지 페트리 네트는 Ω 에 의해서 0과 1 사이의 값을 갖는 토른을 갖는다.

위의 차이점들이 퍼지 추론 시스템을 모델링하는 유용하게 이용될 수 있다. 만약, $W=1$ 이고 $\Omega: P \rightarrow \{0, 1\}$ 이면 제안된 퍼지 페트리 네트는 보통의 페트리 네트가 됨을 알 수 있다.

위와 같은 차이점 이외에도 퍼지 페트리 네트의 동적인 성질 즉, 트랜지션의 발화규칙 및 어떠한 트랜지션이 발화된 후의 네트의 상태변화에 대한 상태 방정식에 대해 생각할 필요가 있다. Looney에 따르면 [11], "페트리 네트가 논리시스템을 모델링할 경우 보통의 발화 규칙과는 달리, 발화되는 트랜지션은 그에 해당되는 모든 출력 플레이스에 토른을 전달해 주어야 하며, 마찬가지로 플레이스에서는 그것에서 출발되는 모든 아크에 토른을 보급하게 해야한다. 따라서 트랜지션의 발화 후에도 그것의 입력 플레이스에 있는 토른은 소멸되지 않는다"라고 지적되고 있다. 뒤에, 우리는 이러한 Looney의 지적을 만족시키는 퍼지 페트리 네트의 상태 방정식을 퍼지추론 시스템의 추론기관으로서 제안한다.

이제 정의 4와 같이 주어진 퍼지 추론 시스템을 제안된 퍼지 페트리 네트로 모델링하는 방법을 제시하고자 한다.

[알고리즘 1]: 퍼지 페트리 네트를 이용한 퍼지 추론 시스템의 모델링

주어진 퍼지 추론 시스템 $\Psi=(X, \Delta, C, \Phi)$ 에 대하여 그것에 대응되는 퍼지 페트리 넷 FPN= $(P, T, \alpha, \beta, W, \Omega)$ 는 다음과 같이 얻어진다.

Step 1: 먼저 플레이스 집합은 다음과 같이 정의한다. $P=\{p_j^i; 1 \leq i \leq n, 1 \leq j \leq l_i\}$. 여기서 P_j^i 는 퍼지 명제 d_j^i 즉, "x is L_j^i "에 대응한다.

Step 2: 주어진 퍼지 추론 규칙 $\delta_i (1 \leq i \leq m)$ 에 대하여 트랜지션 집합과 각 트랜지션에 대한 가중치는 다음과 같이 정의된다.

Step 2-1: 퍼지추론 규칙 δ_i 이 (11)과 같이 주어진 경우, $T_i=\{t_i\}$ 으로 하고 $w_i=c_i$ 으로 한다.

Step 2-2: 퍼지추론 규칙 δ_i 이 (12)와 같이 주어지고, 퍼지추론 규칙의 전건부와 후건부에 있는 퍼지 명제가 각각 k_i, h_i 개 일 경우, $k_i \times h_i$ 개의 트랜지션을 이용하여 $T_i=\{t_j^i \dots t_{k_i h_i}^i\}$ 로 놓고 $W_{pg}^i=C_i (1 \leq p \leq k_i, 1 \leq q \leq h_i)$

$$\text{그러면 } T = \bigcup_i^m T_i$$

Step 3: α 는 다음과 같이 정의한다.

Step 3-1: 퍼지추론 규칙이 (11)과 같이 주어진 경우, δ_i 의 전건부에 관계되는 플레이스들로부터 트랜지션 t_i 로 아크를 만든다.

Step 3-2: 퍼지추론 규칙이 (12)과 같이 주어진 경우, δ_i 의 전건부에 있는 $l (1 \leq l \leq k_i)$ 번째 플레이스로 부터 트랜지션 $t_j^i (1 \leq q \leq h_i)$ 로 아크를 만든다.

Step 4: β 를 다음과 같이 정의한다.

Step 4-1: 퍼지추론 규칙이 (11)과 같이 주어진 경우, 트랜지션 t_i 로부터 δ_i 의 후건부에 관계되는 플레이스로 아크를 만든다.

Step 4-2: 퍼지추론 규칙이 (12)와 같이 주어진 경우, 트랜지션 $t_j^i (1 \leq l \leq k_i, 1 \leq q \leq h_i)$ 로부터 δ_i 의 후건부에 관계되는 플레이스로 아크를 만든다.

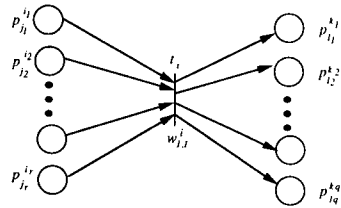
Step 5: 플레이스에 대한 마킹은 다음과 같이 된다. $\Omega = \Phi$ 즉,

$$\Omega = [\omega_1^1, \omega_2^1, \dots, \omega_{l_1}^1; \dots; \omega_1^n, \omega_2^n, \dots, \omega_{l_n}^n]^T$$

여기서 $\omega_j^i \in [0, 1]$ 는 p_j^i 에 잇는 토큰이 지닌 값이다. 그림 4에 (11)과 (12)와 같이 주어지는 퍼지 규칙에 대한 퍼지 페트리 넷 모델을 나타내었다. (11)과 (12)가 규범형에 속하므로 보다 복잡한 경우도 그림 4에 나타나 있는 개념을 이용하면 쉽게 나타낼수 있다.

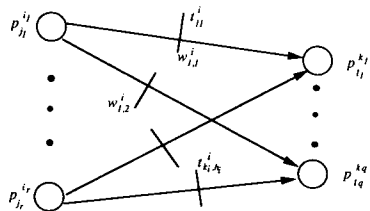
그러면, 제안된 퍼지 페트리 넷을 행렬로 표현하는 방법에 대하여 생각해보자. 기존의 페트리 넷은 입력 연결행렬 및 출력 연결행렬로 표현됨으로서 보다 간결한

$$\delta_1^i = \text{If } d_{j_1}^{i_1} \text{ and } d_{j_2}^{i_2} \dots \text{ and } d_{j_r}^{i_r} \text{ then } d_{i_1}^{k_1} \text{ and } d_{i_2}^{k_2} \dots \text{ and } d_{i_q}^{k_q} (CF = c_i)$$



(a)

$$\delta_2^i = \text{If } d_{j_1}^{i_1} \text{ or } d_{j_2}^{i_2} \dots \text{ or } d_{j_r}^{i_r} \text{ then } d_{i_1}^{k_1} \text{ and } d_{i_2}^{k_2} \dots \text{ and } d_{i_q}^{k_q} (CF = c_i)$$



(b)

그림 4. 퍼지추론 규칙의 퍼지 페트리 넷 모델

- (a) 논리곱으로 연결된 경우
- (b) 논리합으로 연결된 경우

Fig. 4. A fuzzy Petri net model of a fuzzy production rule.

- (a) With conjunctive connective,
- (b) With disjunctive connective.

형태로 페트리 넷을 표현할 수 있었다. 그렇지만 퍼지 페트리 넷이 가중치를 갖는 구조를 가지고 있고, 트랜지션의 발화규칙도 페트리 넷과는 다른 점에 착안하여 근접 연결행렬에 기반을 둔 표현방법 및 발화규칙을 제시하고자 한다. 이를 위하여 먼저 제안 1을 이용하여 논리합으로 연결된 플레이스 근접행렬(disjunctively connected place adjacency matrix; DCPAM)과 논리곱으로 연결된 플레이스 근접행렬(conjunctively connected place adjacency matrix; CCPAM)을 다음과 같이 정의한다.

[정의 6]: 트랜지션에 대한 가중치가 $C=[c_1, c_2, \dots, c_m]^T$ 로 표시되는 퍼지 페트리 넷에 대하여 제안 1에 의해 D_c^- 와 D_d^- 를 구할 수 있으며, CCPAM, D_c^+ 와 DCPAM, D_d^+ 는 다음과 같이 정의된다.

$$D_c^+ = D^+ \cdot \text{Diag}(C) \cdot (D_c^-)^T \tag{13}$$

$$D_d^+ = D^+ \cdot \text{Diag}(C) \cdot (D_d^-)^T \tag{14}$$

여기서, $\text{Diag}(C)$ 는 C 의 대각행렬(diagonal matrix)이

다. 즉, $Diag(C)$ 의 대각열에 있어서 i 번째 요소는 C 의 i 번째 요소와 같고 다른 요소들은 모두 0인 행렬이다. C 가 1만으로 이루어진 벡타의 경우에는 $D^c + D^d$ 는 앞에서 정의되었던 플레이스 근접행렬 D^a 와 같음을 알 수 있다.

3. 퍼지 페트리 네트를 이용한 추론기관의 구현

앞에서 퍼지 페트리 네트를 이용하여 퍼지 추론 시스템을 모델링함에 있어서 논리곱이나 논리합 그리고 추론연산자(implication operator)에 대하여 특정한 제한을 두지 않았다. 이는 이들로 이루어진 임의의 퍼지 추론 시스템의 모델링에 이를 적용하려 함이었다. 그렇지만 퍼지 추론 시스템에 있어서 특정한 사실에 대하여 지식베이스로부터 결론을 추론하기 위해서는 논리곱, 논리합 그리고 추론연산자에 해당되는 것들을 t-norm, t-conorm, 추론연산자 중에서 특정한 것을 선택해야 한다. 본 논문은 여러 t-norm과 t-conorm 중에서 가장 널리 쓰이는 방식을 방법으로 논리곱은 min을 논리합은 max를 사용하는 방식을 사용하였다. 또한 추론연산자에 있어서는 Mamdani^[17]가 제시한 방법을 확신도를 고려하여 사용하고자 한다.

Whalen과 Schott이 지적 했듯이^[15], 추론시스템은 완전한 순차적인 구조도 아니며 그렇다고 완전한 parallel한 프로그램 구조를 갖는 것도 아니다. 따라서 하나의 추론 규칙이 수행되었을 때 이는 working memory의 내용을 변화시키고 다음번에 수행될 추론규칙이 어떤 것이 되는지에 영향을 미친다. 이런 문제를 일컬어 conflict resolution이라 하는데, 이는 working memory의 내용이 여러개의 규칙의 전제조건을 동시에 만족할때 어떤 규칙을 먼저 적용할지를 결정 할때 발생한다. 대부분의 추론 시스템을 이용한 성공적인 프로그램들은 이 문제를 각기 나름대로의 방법으로 해결해 나가고 있는데 이러한 방법들이 실제적인 상황에서 중요한 의미를 갖는 것은 아니므로 처음부터 이런 문제를 배제하는 것이 바람직하다.

이런 문제들은 퍼지 추론 시스템에 있어서는 min과 max 등을 적용하게 되면 발생되지 않는다. 왜냐하면, min과 max의 commutative하고 associative한 성질로 말미암아 규칙의 적용순서에 무관하게 같은 추론 결과를 얻을 수 있으며 모든 규칙을 동시에 적용하더라도 역시 동일한 결과를 얻을 수 있다. 관점에서 많은 추론 규칙 중에서 다음과 같은 Mamdani의 추론 방법이 실제적인 문제에 많이 쓰이고 있음을 알 수 있다^[17].

규 칙 : IF x_1 is L_1^1 and x_2 is L_1^2 THEN x_3 is L_1^3
($CF=c_1$)

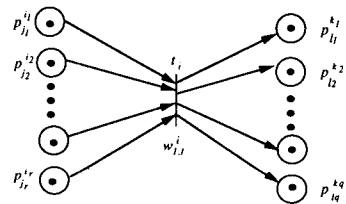
사 실 : x_1 is \tilde{L}_1^1 and x_2 is \tilde{L}_1^2

결 과 : x_3 is \tilde{L}_1^3

앞에서 제안된 ω -수준 퍼지 퍼지구간과 퍼지 진리값의 정의를 이용하면 각각의 소속함수가 $u_{L_i^1} = ((m_1)_i^1, (m_2)_i^1, \alpha_i^1, \beta_i^1)_{L_i^1 R_i^1}$ ($i=1, 2, 3$)라고 할때 추론의 결과는 다음과 같이 표현 될 수 있다. 즉, $u_{L_1^3} = ((m_1)_1^3, (m_2)_1^3, \alpha_1^3, \beta_1^3)_{L_1^3 R_1^3}$ 여기서 $\omega_1^3 = c_1 \times \min \{ \max_u \min \{ u_{L_1^1}(u), u_{L_1^2}(u) \}, \max_v \min \{ u_{L_1^2}(v), u_{L_1^2}(v) \} \}$.

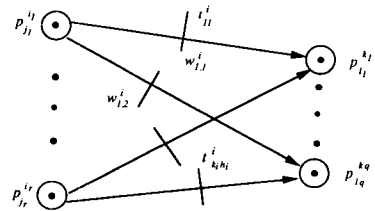
제시된 추론 방식에 대한 페트리 네트에서의 모델이 그림 5에 표시되어 있다.

$\delta_1^1 = \text{If } d_{11}^1 \text{ and } d_{12}^2 \dots \text{ and } d_{1r}^r \text{ then } d_{11}^{k1} \text{ and } d_{12}^{k2} \dots \text{ and } d_{1q}^{kq} (CF = c_1)$



(a)

$\delta_1^2 = \text{If } d_{11}^1 \text{ or } d_{12}^2 \dots \text{ or } d_{1r}^r \text{ then } d_{11}^{k1} \text{ and } d_{12}^{k2} \dots \text{ and } d_{1q}^{kq} (CF = c_1)$



(b)

그림 5. 퍼지 페트리 네트에서의 추론과정
(a) 논리곱으로 연결된 경우
(b) 논리합으로 연결된 경우

Fig. 5. Inference process in a fuzzy petri net.
(a) With conjunctive connective,
(b) With disjunctive connective.

제시된 ω -수준의 퍼지구간 및 퍼지 진리값의 개념을 사용하면 추론 과정을 체계적으로 기술할 수 있음에 착안하여 다음과 같은 추론기관을 제시한다.

[알고리즘 2]: 퍼지 페트리 네트를 이용한 추론 기관의 구현

퍼지 추론 시스템이 chaining된 규칙으로 이루어진 경우 몇개의 추론 단계를 거쳐야만 최종적인 추론 결과를 얻을 수 있는데 이때의 추론 단계를 k로 나타내기로 하자. 먼저, I/O부로 부터 k=0인 단계에서 working memory로 주어지는 최초의 퍼지 진리값을 Ω^0 두고, k 단계에서의 퍼지 진리값을 Ω^k 로 두자. 이는 퍼지 페트리 네트에 있어서는 단계 k에서의 마킹에 해당된다. 각 단계마다 Ω^k 가 Ω^{k+1} 와 같아질 때 까지, 즉 working memory에 더 이상의 새로운 정보의 추가가 없을 때까지 아래 (15)식을 적용하여 트랜지션을 점화시킨다.

$$\Omega^{k+1} = \Omega^k V(D^d \oplus \Omega^k) V(D^c \otimes \Omega^k) \quad (15)$$

여기서, V와 \oplus 는 각각 max연산과 max-product 연산을 나타낸다. 그리고 \otimes 는 다음과 같이 정의된다. 먼저, 다음과 같은 성질을 갖는 기호 σ 를 정의하자. 모든 $x \in [0, 1]$ 에 대하여 $\min(\sigma, x) = \max(\sigma, x) = x$ 그리고 $x_1, x_2 \in [0, 1]$ 에 대하여 다음과 같은 이항연산을 정의하자.

$$x_1 \cdot x_2 = \sigma \quad \text{if } x_1 = 0 \\ x_1 \cdot x_2 \quad \text{otherwise}$$

그러면

$$D^c \otimes \Omega^k = \bigwedge_i \{(d^c)_{ij} \cdot w_i^k\}$$

(15)식의 두번째 항은 논리합으로 이루어진 플레이스에 대한 max 연산을 하는 것을 기술하며, 세번째 항은 논리 곱으로 연결된 플레이스에 대하여 min 연산을 나타낸다.

4. 추론 알고리즘의 수렴성 증명

(15)식의 수렴성을 보임으로서 본 논문에서 제시한 추론 알고리즘이 유한한 단계에서 종료되며, 어떤 특정한 결론을 유도함을 보이고자 한다.

[정리 1]: 식 (15)에 대하여, Ω^0 가 주어졌을 때, 다음과 같은 \tilde{k} 이 존재하여 $\tilde{k} \geq k$ 에 대하여 $\Omega^{\tilde{k}} = \text{constant}$ 가 된다.

(증명) D^a 와 D^c 의 모든 요소들이 0과 1사이의 값으로 구성되어 있고, Ω^0 의 모든 요소 또한 0과 1 사이의 값으로 구성되어 있다. 또한 사용되어진 연산이 min과 max 이므로

$$\Omega^k \leq \Omega^k \leq 1; \quad k \geq 0 \quad (16)$$

임을 알 수 있다. 또한

$$\Omega^k \leq \Omega^{\tilde{k}} \quad \text{if } k \geq \tilde{k} \quad (17)$$

임을 알 수 있고, \tilde{k} 에서 $\Omega^{\tilde{k}}$ 와 $\Omega^{\tilde{k}+1}$ 이 같은 값을 갖는다면 $\Omega^{\tilde{k}} = \Omega^{\tilde{k}+1}$ 이 됨을 쉽게 알 수 있다. 결과적으로 위의 성질들로 부터 Ω 가 어떠한 일정한 값에 수렴함을 알 수 있다.

5. 전체 알고리즘의 구성 및 흐름

먼저 특정한 분야에 대한 전문가의 부정확한 지식을 퍼지 추론 규칙의 형태로 체취한다. 이때 각 규칙에 대한 불확실성을 확신도를 이용하여 각 규칙에 부여하고 규칙속에 포함된 애매함은 적당한 퍼지수를 이용하여 표현하게 된다. 만약, 주어진 문제가 센서나 전문가로 부터 시스템의 상태가 입력되면 구해진 추론 규칙들로 부터 실시간으로 적절한 판단을 내려야 하는 decision making 문제라 하자. 이러한 문제는 실제로 원자력 발전소에 있어서의 지식기반 고장판단 및 고장처리에서 발생하는 문제이다. 이러한 문제를 본 논문에서 제안된 방법에 따르면 다음과 같다.

1. [알고리즘 1]에서 제시된 방법에 따라 퍼지 추론 규칙들이 퍼지 페트리 네트로 변환된다.
2. 구해진 퍼지 페트리 네트를 식 (13)과 (14)에서 정의된 두개의 행렬, DCPAM과 CCPAM으로 표현한다.
3. 위와 같은 행렬로 표현된 퍼지 페트리 네트에 의해 (15)과 같은 추론식이 얻어진다.

위의 과정들을 거치면, 주어진 decision making 문제가 간단한 대수식으로 표현되고 이는 쉽게 하드웨어로도 구현이 가능한 형태로 된다. 이후에 센서로 부터 시스템의 상태가 입력되는 경우에는 [정리 1]에서 보여진 바와 같이 유한한 단계안에 원하는 추론 결과를 얻을 수 있다. 이러한 문제를 기존의 인공지능 분야에서 많이 쓰이는 search방법을 쓰는 경우에 search space가 증대하며, 추론의 결과를 얻는데 많은 시간을 소모하며 하드웨어 구현도 용이하지 않아서 실제적인 실시간 decision making 시스템의 구현에 한계가 있다.

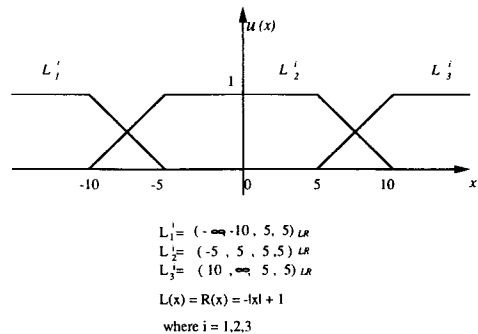


그림 6. 사용된 평평한 LR-형태의 퍼지수
Fig. 6. Adopted LR-type flat fuzzy number.

6. 예제

본 논문에서 제시된 방법들을 설명하기 위하여 각각이 그림 6과 같이 주어지는 평평한 LR-형태의 퍼지수를 갖는 3개의 언어변수로 이루어지며 I/O부분에서 부터 초기 퍼지 진리값 Ω^0 가 그림 7.(b)와 같이 주어진 다음과 같은 퍼지 추론시스템에 대한 퍼지 페트리 네트에 의한 모델링 및 추론기관에서의 추론과정 π 대하여 알아보자.

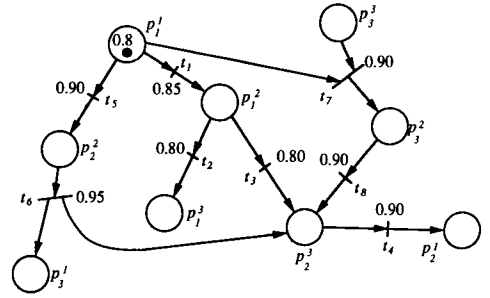


그림 8. 퍼지 네트리 네트 모델
Fig.8. A fuzzy petri net model.

$$C = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \\ 0.85 & 0.8 & 0.8 & 0.9 & 0.9 & 0.95 & 0.9 & 0.9 \end{bmatrix}$$

$$\Omega^0 = \begin{bmatrix} \omega_1^1 & \omega_2^1 & \omega_3^1 & \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_1^3 & \omega_2^3 & \omega_3^3 \\ 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

그림 7. 확신도와 초기 진리값

Fig.7. Certainty factors and initial fuzzy truth values.

- $R_1 : IF x_1 \text{ is } L_1^1 \text{ THEN } x_2 \text{ is } L_1^2 (c_1=0.85)$
- $R_2 : IF x_2 \text{ is } L_1^2 \text{ THEN } x_3 \text{ is } L_1^3 (c_2=0.80)$
- $R_3 : IF x_2 \text{ is } L_2^2 \text{ THEN } x_3 \text{ is } L_2^3 (c_3=0.80)$
- $R_4 : IF x_3 \text{ is } L_2^3 \text{ THEN } x_1 \text{ is } L_2^1 (c_4=0.90)$
- $R_5 : IF x_1 \text{ is } L_1^1 \text{ THEN } x_2 \text{ is } L_2^2 (c_5=0.90)$
- $R_6 : IF x_2 \text{ is } L_2^2 \text{ THEN } x_3 \text{ is } L_3^3 \text{ and } x_1 \text{ is } L_3^1 (c_6=0.95)$
- $R_7 : IF x_1 \text{ is } L_1^1 \text{ THEN } x_3 \text{ is } L_3^3 \text{ THEN } x_2 \text{ is } L_3^2 (c_7=0.90)$
- $R_8 : IF x_1 \text{ is } L_3^3 \text{ THEN } x_3 \text{ is } L_2^2 (c_8=0.90)$

위의 퍼지 추론 시스템에 대한 퍼지페트리 네트 모델을 주어진 방법에 따라 구하면 그림 8과 같이 되고 이를 다시 제안된 행렬에 의해 표현하면 그림 9와 같이 된다. 이들을 이용, (15)식에 따라 각 k단계에서의 working memory의 내용 즉 중간 추론 값들이 생성되는 과정을 그림 10 (b)에 나타내었는데 k=3에서 최종 추론 값을 얻을 수 있음을 알 수 있다. 이때 퍼지 페트리 네트리 네트에서 각 플레이스의 최종마킹 즉, 각 퍼지명칭에 대한 퍼지진리값을 그림 10 (a)에 나타내었다. 이렇게 언어진 최종의 퍼지 진리값은 I/O부분으로 보내진 후 w-수준 퍼지구간의 개념을 이용하여 비퍼지화(defuzzification)과정을 거쳐 최종적인 결론을 얻는데 이 부분에 대한 상세한 설명은 여기서는 생략하고자 한다.

위의 예제에서 알 수 있듯이 본 논문에서 구현된 추론 기관은 working memory의 전체내용을 동시에 갱신되게 하는 병렬 구조를 갖고 있어 빠른 추론이 가능하고 추

	p_1^1	p_2^1	p_3^1	p_1^2	p_2^2	p_3^2	p_1^3	p_2^3	p_3^3
$D^d = p_1^1$	0	0	0	0	0	0	0	0	0
p_2^1	0	0	0	0	0	0	0	0.9	0
p_3^1	0	0	0	0	0	0	0	0	0
p_1^2	0.85	0	0	0	0	0	0	0	0
p_2^2	0.9	0	0	0	0	0	0	0	0
p_3^2	0	0	0	0	0	0	0	0	0
p_1^3	0	0	0	0.8	0	0	0	0	0
p_2^3	0	0	0	0.8	0.95	0.9	0	0	0
p_3^3	0	0	0	0	0	0	0	0	0

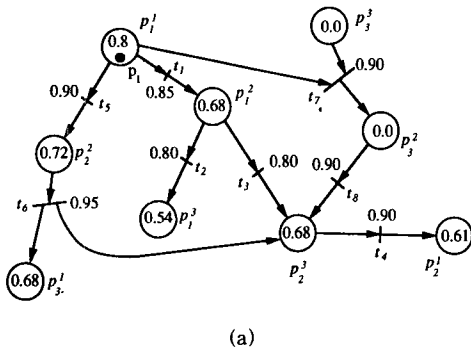
(a)

	p_1^1	p_2^1	p_3^1	p_1^2	p_2^2	p_3^2	p_1^3	p_2^3	p_3^3
$D^c = p_1^1$	0	0	0	0	0	0	0	0	0
p_2^1	0	0	0	0	0	0	0	0	0
p_3^1	0	0	0	0	0	0	0	0	0
p_1^2	0	0	0	0	0	0	0	0	0
p_2^2	0	0	0	0	0	0	0	0	0
p_3^2	0.9	0	0	0	0	0	0	0	0.9
p_1^3	0	0	0	0	0	0	0	0	0
p_2^3	0	0	0	0	0	0	0	0	0
p_3^3	0	0	0	0	0	0	0	0	0

(b)

그림 9. 행렬 표현
Fig. 9. Matrix representation.

론과정을 퍼지 페트리 네트에서의 토큰의 흐름으로 쉽게 알 수 있으므로 추론 결과에 대한 중간 과정의 설명이 용이함을 알 수 있다.



(a)

Ω^k	k				
	0	1	2	3	4
ω_1^1	0.8	0.8	0.8	0.8	0.8
ω_2^1	0	0	0	0.61	0.61
ω_3^1	0	0	0.68	0.68	0.68
ω_1^2	0	0.68	0.68	0.68	0.68
ω_2^2	0	0.72	0.72	0.72	0.72
ω_3^2	0	0	0	0	0
ω_1^3	0	0	0.54	0.54	0.54
ω_2^3	0	0	0.68	0.68	0.68
ω_3^3	0	0	0	0	0

(b)

그림 10. 추론기관의 출력

(a) 추론후의 퍼지 페트리 넷

(b) 추론과정 및 최종출력

Fig. 10. Output of inference engine.

(a) The final fuzzy petri net,

(b) Inference process.

IV.결 론

본 논문에서는 퍼지 페트리 넷을 좀더 일반적인 형태로 다시 정의하고 이를 이용하여 퍼지 추론 시스템을 모델링하는 방법을 제시하였다. 또한 퍼지 추론 시스템의 추론기관이 퍼지 페트리 넷의 상태 방정식으로 표현됨을 보였다. 제안된 퍼지 추론 시스템의 모델링 방법이 그래프적인 방법에 기초하므로 각 추론 규칙간의 관계를 쉽게 알 수 있을 뿐더러 전문가 시스템에서 요구되는 추론 과정에 대한 설명기능이 쉽게 구현 될 수 있는 장점이 있다. 더우기, 실제 산업용의 지식기반 시스템에서는 실시간 추론 기능이 요구됨을 감안하면 본 논문에서 제안된, 간단한 대수식으로 표현된 추론기관이 유용하게 이용될 수 있으리라 생각되어 진다.

본 논문에서 제시된 개념들을 이용하여 IBM-PC에서 MATLAB을 사용하여 FPNIENG(fuzzy Petri net inference engine)이라는 일종의 전문가 시스템용 shell을 구현하였다. FPNIENG은 에디터와 시뮬레이터로 구성되어 있는데, 에디터는 소속함수의 정의 및 수정 그리고 입력 연결행렬 및 출력 연결행렬의 수정과 확신도의 입력을 가능하게 하며, 시뮬레이터는 주어진 연결행렬로부터 추론기관의 구현에 필요한 접속행렬을 만들어 낸다. 또한 시뮬레이터에 본 논문에서 제안된 방법에 따라 구현된 추론기관은 주어진 초기 퍼지 진리 값에 따라 추론을 행하여 최종적인 추론 결과를 내게 된다.

參 考 文 獻

- [1] C.V. Negoita, *Expert Systems and Fuzzy Systems*, The Benjamin/Chumming Company, Reading, Mass. 1985.
- [2] H.J. Zimmermann, *Fuzzy Sets, Decision Making, and Expert Systems*, Kluwer Academic Publishers, Boston, 1987.
- [3] P. Jackson, *Introduction to Expert Systems*, Addison-Wesley, Reading, Mass., 1990.
- [4] R.K. Bhatnagar and L.N. Kanal, "Handling uncertainty information: A review of numeric and non-numeric Methods," in: L.N. Kanal and J.F. Lemmer, Ed., *Uncertainty in Artificial Intelligence*, North-Holland, Amsterdam, pp. 3-26, 1986.
- [5] K.C. Ng and B. Abramson, "Uncertainty management in expert systems," *IEEE Expert*, vol. 5, pp. 29-48, 1990.
- [6] L.A. Zadeh, "Knowledge representation in fuzzy logic," *IEEE Tr. on Knowledge and Data Engineering*, vol. 1, pp. 89-100, 1989.
- [7] A. Giordana and L. Saitta, "Modeling production rules by means of predicate transition networks," *Information Science*, vol. 35, pp. 1-45. 1985.
- [8] T. Murata and D. Zhang, "A predicate-transition net model for parallel interpretation of logic programs," *IEEE Tr. on Software Engineering*, vol. 14, pp. 481-497, 1988.
- [9] Y. Deng and S. -K. Chang, "A-net model for knowledge representation and reasoning," *IEEE Tr. on Knowledge and Data Engineering*, vol. 2, pp. 295-310, 1990.
- [10] H.-P. Lipp, "The application at a fuzzy petri net for controlling complex industrial process," *Proc. IFAC Sump. on Fuzzy Information*,

- Knowledge, Representation and Decision Analysis*, Marseille, pp. 471-477, 1983.
- [11] C. G. Looney, "Fuzzy petri nets for rule-based decisionmaking," *IEEE Tr. on System, Man, and Cybernetics*, vol. 15, pp. 178-183., 1988.
- [12] S.-M. Chen and J.-S. Ke and J.-F. Chang, "Knowledge representation using fuzzy petri nets," *IEEE Tr. on Knowledge and Data engineering*, vol. 2, pp. 311-319, 1990.
- [13] J.L. Peterson, *Petri Nets Theory and the Modeling of System*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
- [14] D. Dubois and H. Prade, *Fuzzy Sets and Systems-Theory and Applications*, Academic Press, New York, 1980.
- [15] T. Whalen and B. Schott, "Decision support with fuzzy production systems," in: P.P. Wang, Ed., *Advance in Fuzzy Sets, Possibility Theory, And Applications*, Plenum Press, New York, pp. 199-216, 1983.
- [16] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Englewood Cliff, N.J., 1974.
- [17] M. Mizumoto, "Fuzzy controls under various fuzzy reasoning methods," *Information Science*, vol. 45, pp. 129-151, 1988.
- [18] L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3-28, 1978.

 著 者 紹 介

全 命 根 (正會員) 第28卷 B編 第11號 參照
 현재 한국과학기술원 전기및
 전자공학과 박사과정 재학중.

卞 增 男 (正會員) 第28卷 B編 第11號 參照
 현재 한국과학기술원 전기및
 전자공학과 교수