

퍼지 객체 추론 모델의 정형화

(A Formal Specification of Fuzzy Object Inference Model)

양재동[†] 양형정^{**}
(Jae-Dong Yang) (Hyung-Jeong Yang)

요약 기존의 퍼지 규칙 기반 전문가 시스템 언어에는 크게, 세 가지 단점들이 있다. 첫째, 복합 객체 추론 기능이 없으며, 둘째, 의미적으로 이해하기 쉽고 개념적으로 사용하기 용이한 퍼지 추론을 지원하지 못할 뿐 아니라, 셋째, 지식 표현과 추론 방식이 기존의 데이터베이스 모델과 구문이나 의미에서 현격한 차이를 보이고 있기 때문에 서로 통합되기 어렵다. 본 논문에서는 이 세가지 단점들을 해결하기 위한 퍼지 객체 추론 모델의 정형화를 보이고, GIS 응용을 예로 들어 제시하는 모델이 데이터베이스내 GIS 복합 객체들을 자연스럽게 모델링하고, 이들 사이의 퍼지 추론을 성공적으로 수행함을 보인다.

Abstract There are three significant drawbacks in extant fuzzy rule-based expert system languages. First, they lack the functionality of composite object inference. Second, they do not support fuzzy reasoning semantically easy to understand and conceptually simple to use. Third, knowledge representation and reasoning style of their model have a great semantic gap with those of current database models. Therefore, it is very difficult for the two models to be seamlessly integrated with each other. This paper provides the formal specification of a fuzzy object inference model to solve the three drawbacks. GIS(Geographic Information System) application domain is used to demonstrate that our model naturally models complex GIS information in terms of composite objects and successfully performs fuzzy inference between them.

1. 서론

지난 수 십년 동안 여러 퍼지 규칙 기반 전문가 시스템 언어들이 개발되어 의료 진단, 정보 검색, 자연어 처리 등의 분야에 성공적으로 적용되었다[1][2][3]. 이 언어들은 특히, 퍼지 언어 용어(fuzzy linguistic term)를 처리하는 능력으로 이들 분야에서 다양한 불확실성을 모델링하는데 자연스런 해석을 제공해 왔다.

그러나, 최근 들어 전문가 시스템 언어들이 복잡한 공학적 응용 영역에까지 확장 적용되면서, 이들이 복합 객체들간의 추론 기능은 지원하지 못한다는 점이 점차 큰 단점으로 부각되었다[4][5]. NASA에서 개발된 Fuzzy CLIPS 6.0[4]은 객체 지향 패러다임과 연역 메커니즘을

퍼지 객체 추론 언어 모델로 통합하려는 한 예이다. 그러나 FuzzyCLIPS는 객체 추론 방법이 현재의 데이터베이스 시스템과 구문이나 의미 면에서 큰 차이를 보인다는 문제점을 가지고 있다.

한편 전문가 시스템에서 다루는 데이터의 양이 점차 방대해지고, 데이터베이스 시스템에 연역 기능을 추가하는 작업의 중요성이 증대됨에 따라, 전문가 시스템 언어와 데이터베이스를 통합하기 위한 많은 연구가 이루어졌다[6][7][8][9]. 그러나 이러한 통합은 인공 지능과 데이터베이스 영역에서 사용하는 데이터 모델이 원천적으로 서로 다른데서 오는 불일치(impedence mismatch) 문제를 야기시킨다. 예를 들어, FuzzyCLIPS 6.0을 데이터베이스와 통합시키는 경우, 프로그래머들은 FuzzyCLIPS 언어와 전혀 다른 데이터베이스 언어까지 알아야 응용 프로그램을 개발할 수 있게 된다. 이 문제를 해결하기 위한 한 방법으로, 데이터베이스 언어에 객체 추론 기능을 추가하는 많은 연구들이 진행되어왔다. K[6]와 Coral++[8], Adam[9] 등은 데이터베이스 위에 불일치 없이 객체 추론 기능을 구현한 언어들이다. 그러나 이 객체 추론

· 본 논문은 전북대학교 영상 정보 신기술 연구소의 지원으로 수행되었음.

† 정 회 원 : 전북대학교 컴퓨터과학과 교수
jdyang@cs.chonbuk.ac.kr

** 학생회원 : 전북대학교 컴퓨터과학과
hjyang@cs.chonbuk.ac.kr

논문접수 : 1999년 5월 19일

심사완료 : 2000년 3월 29일

은 지식베이스가 불확실성을 포함하고 있을 때는 적용할 수 없는 한계를 가지고 있다.

본 논문에서는 객체 추론과 퍼지 추론이 객체-관계형 데이터베이스와 호환성있는 일관된 틀로 병합되어 있는 퍼지 객체 추론 모델을 제안하고 이를 정형화한다. 제안 모델은 객체 지향 패러다임의 대부분이 관계형 구조로 표현되기 때문에, 의미적으로 이해하기 쉽고 개념적으로 사용하기 단순한 퍼지 추론을 지원한다. 또한, 퍼지 추론 방식도 데이터베이스에 적합한 구조로 설계되었기 때문에, 필요하다면 [8][10][11]에서 개발된 여러 방법들을 이용하여 데이터 지속성을 쉽게 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 본 논문에서 응용 사례로 보일 GIS(Geographic Information System) 노선 관리 문제를 설명한다. 4장에서는 퍼지 객체 클래스들을 포함하는 퍼지 객체 추론 모델과 클래스들을 다루는 규칙의 평가를 설명한다. 마지막으로, 5장에서는 결론과 앞으로의 연구 방향에 대하여 언급한다.

2. 관련 연구

FLOPS[21]와 Z-II [22]는 퍼지 추론을 지원하는 전문가 시스템 셸들로서, 이들은 퍼지 용어들을 처리할 수 있는 강력한 추론 능력을 제공함으로써 확률 이론을 적용하는 시스템들에 비해 불확실한 지식을 보다 자연스럽게 해석해 준다. 그러나 이 전문가 시스템 셸들의 퍼지 추론 방식과 시멘틱은 관계형 구조(relational framework)와 호환성이 없다는 중요한 결점을 가지고 있다. 반면에, OPS5[23]와 ECLPS[24]는 다루는 자료의 형식이 관계형 구조와 유사한 생성 규칙 기반 언어들이지만, FLOPS와 Z-II와 같이 강력한 퍼지 추론 능력은 제공하지는 못하고 있다.

NASA에서 개발된 FuzzyCLIPS [4]는 지식 기반 전문가 개발 도구인 CLIPS(C Language Production System)의 객체 지향 버전인 CLIPS 6.0x를 기반으로 퍼지 개념과 퍼지 추론 기능을 포함하도록 개발된 범용의 퍼지 전문가 시스템 셸이다. CLIPS는 ART를 모델로 하여 C 언어로 개발되었으며, 도구 자체의 강력한 기능 이외에 외부 시스템과의 병합의 용이성, 하드웨어 환경의 호환성, 빠른 실행 속도, 저렴한 가격, 보급의 편이성 등의 장점으로 널리 사용되고 있다. FuzzyCLIPS는 자체에서 제공하는 퍼지 용어를 이용하여 규칙을 작성하며, 퍼지 용어와 일반 용어와의 혼합 및 논리 제어의 수치 비교를 처리하고 규칙과 사실내의 불확실성을 허용한다. 그러나 FuzzyCLIPS의 시멘틱은 데이터베이스와 달라 데이터베

이스와의 병합시 불일치 문제가 발생하며, 일반화 계층과 복합 계층에서 발생하는 불확실성을 처리하지 못한다.

Coral++[8]와 K 언어[6]는 데이터베이스와의 불일치 없이 객체 추론 기능을 구현하기 위한 객체 언어들이다. Coral++는 데이터베이스에 C++의 객체를 수용하며, C++의 객체들은 확장된 Coral 프로그램에 의해 접근될 수 있도록 함으로써 객체 지향 패러다임과의 병합을 이루고 있다. 그러나 C++의 구문은 Coral과 많이 달라서 C++의 많은 부분이 Coral의 규칙에 사용될 수 없고, Coral++의 객체 추론 기능은 지식베이스가 불확실성을 포함하고 있을 때는 적용될 수 없다. 한편, 고차원 지식 모델인 OSAM* [11]의 인터페이스 언어로 개발된 K 언어는 일반적인 프로그래밍뿐 만 아니라 고차원의 지식을 정의할 수 있고, 지식의 접근과 지식의 조작을 일관된 방식으로 수행할 수 있다. K 언어는 범용의 시스템 개발 언어로서 객체 지향, 병렬, 비 결정적, 그리고 규칙 기반 메커니즘을 지원한다. 그러나, K 언어는 여전히 관계형 데이터베이스나 객체-관계형 데이터베이스(object-relational database)와 같은 데이터베이스 시스템과 불일치 문제를 보이고 있으며, 또한 객체 추론시 불확실성도 처리하지 못하고 있다.

3. 응용도메인의 예제 스키마

뉴욕이나 런던, 서울과 같은 대도시에서는 버스나 전철이 주요 공공 교통 수단으로 이용되고 있다. 따라서 버스나 전철 노선을 효율적으로 관리할 수 있는 교통 노선망 관리 시스템은 교통 혼잡을 해결하는데 유용하게 사용될 수 있다. 본 논문에서는 구현 시스템의 응용 사례로 GIS(Geographic Information System) 분야의 노선 관리 문제를 사용한다[12]. 노선 망의 구성과 관리를 결정하는데 사용되는 도메인 지식은 실제 현장에서 전문가들로부터 습득된다.

그림 1은 버스 전철 노선 관리 지리 정보 시스템의 스키마중 일부를 보이고 있다. 하나의 클래스는 클래스 이름과 애트리뷰트를 포함하는 사각형으로 표현되고, 클래스 사이의 연결선은 두 객체간의 일반화 계층 구조와 복합 계층 구조를 표현하고 있는데, 일반화는 진한 선으로 나타내며, 복합 계층 구조는 곡선으로 나타낸다. 일반화는 한 클래스와 다른 클래스들과의 부클래스 관계를 의미하는 것으로 예를 들어, 'commuter_bus_route'와 'subway_route'는 'route'의 부클래스이며, 'route' 클래스의 애트리뷰트들을 상속받는다. 복합 계층 구조는 클래스와 객체(instance) 사이의 관계를 표현하는 것으로 애트리뷰트의 형(type)으로 클래스 형을 지정하고 애트리뷰

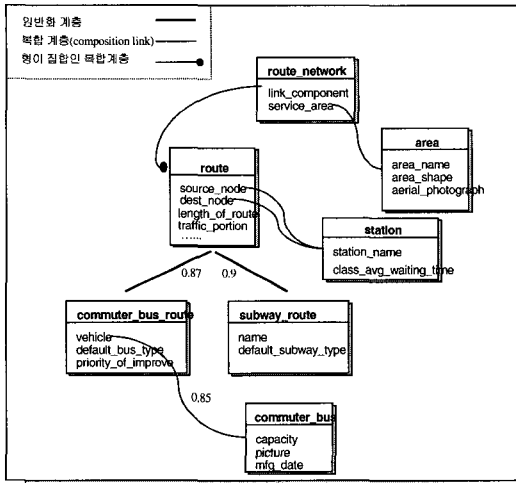


그림 1 노선 관리 시스템의 부분 스키마

트 값으로 다른 클래스의 한 객체, 또는 객체 집합이 할당된다. 예를 들어, 'commuter_bus_route' 클래스내의 'vehicle' 애트리뷰트와 'commuter_bus' 클래스는 복합 계층 구조를 보이고 있다.

클래스 정보를 포함한 두 계층 구조는 이 응용 예에서 정교하게 퍼지화 되어 있는데, 예를 들어, 'subway_route'는 철도라는 비교적 고정된 'route_network'를 형성하므로 'commuter_bus_route'보다 'route'의 클래스일 가능성이 높다. 또한 'commuter_bus_route' 클래스내의 'vehicle' 애트리뷰트 값으로 'commuter_bus' 클래스의 한 객체가 될 가능성이 불확실한 경우에 복합 계층 관계가 퍼지화 될 수 있다.

제안 모델에서 대부분의 퍼지 객체 추론은 클래스들 사이의 상호 작용을 제어하는 전역 규칙을 사용함으로써 이루어진다.

4. 퍼지 객체 추론 모델

제안 모델에서 규칙 기반 추론은 기본적으로 Fprolog [13]에서 채택된 개념을 사용하였지만 하이브리드 객체 지향 패러다임이라는 점에서는 Fprolog와 근본적으로 다르다. 본 장에서는 퍼지 객체 클래스들을 포함하는 퍼지 객체 추론 모델과 규칙의 평가를 설명한다.

4.1 퍼지 객체 클래스의 표현

객체의 구조는 관계형 구조와 불일치를 최소화하는 방향으로 설계되었다. 즉, 퍼지 정보를 포함하는 객체는 추상 데이터 형과 함께 관계형 테이블로 표현되고, 객체 지

향 패러다임 또한 관계형 구조로 표현된다. 먼저, 기본이 되는 퍼지 집합을 정의하자.

[정의 1] 전체 정규 집합 U 에서 정의되는 퍼지 집합 F 는 다음의 소속성 함수에 의해서 설정 된다.

$$\mu_F : U \rightarrow [0,1]$$

여기서, 각 $c \in U$ 에 대해 $\mu_F(c)$ 는 퍼지 집합 F 에 c 가 속하는 소속성 정도를 나타낸다.

F 는 퍼지 집합의 표기법에 따라, 다음의 두 가지 방법으로 쓰여질 수 있다.

$$F = \{ \mu_F(c_1)/c_1, \mu_F(c_2)/c_2, \dots, \mu_F(c_n)/c_n \}, \forall c_i \in U \text{에 대해, } F \text{가 이산 집합일 때,}$$

$$= \int \mu_F(c)/c, \forall c \in U \text{에 대해, } F \text{가 연속 집합일 때.}$$

정의 1에서 $\int \mu_F(c)/c$ 는 수학에서의 적분 표시는 아니고 소속성 함수가 연속성을 지닌 함수로 표현됨을 의미한다. 예를 들어, $long = \int \mu_{long}(c)/c$, $\mu_{long}(x) = (1 + 1/1000 \times |2000 - x|)^{-1}$ 에서 2050이 퍼지 집합 $long$ 의 원소일 가능성(또는 2050을 길다고 볼 가능성)은 $\mu_{long}(2050) = (1 + 1/1000 \times |2000 - 2050|)^{-1} = 0.95$ 이다.

[정의 2] 이름이 N 인 퍼지 클래스(또는 단순히 클래스)는 다음과 같이 정의된다.

$$C_N = \langle E_N, sch(C_N), T_N \rangle$$

- 여기서
- 1) E_N 은 퍼지 객체들의 집합,
 - 2) $sch(C_N)$ 은 애트리뷰트들의 집합,
 - 3) T_N 은 형(type)들의 집합.

$A_i \in sch(C_N)$, $i=1, \dots, n$ 일때 $\tau(A_i)$ 를 애트리뷰트 A_i 의 형이라하면 $\tau(A_i) \in T_N$ 은 퍼지 집합이나 퍼지 집합의 집합으로 정의된다. 퍼지 클래스가 관계-객체형 테이블과 대응됨을 강조하기 위해, 퍼지 객체는 퍼지 튜플(또는 단순히 튜플)과 구별없이 부르기로 한다.

[정의 3] E_N 은 $\tau(A_1) \times \tau(A_2) \times \dots \times \tau(A_n)$ 의 퍼지 부분 집합이고, $\tau(A_1) \times \tau(A_2) \times \dots \times \tau(A_n)$ 은 $U_1 \times U_2 \times \dots \times U_n$ 의 퍼지 부분집합이다. 여기서 \times 은 카르티션 곱을 나타내고, U_i , $i=1, \dots, n$ 은 해당 퍼지 집합들이 정의되는 정규 집합들이다.

[정의 4] 튜플 $t = \langle a_1, a_2, \dots, a_n \rangle$ 라 하면, $t \in E_N$ 의 정도는 $\mu_{E_N}(t)$ 또는 간단히 $\mu(t)$ 이며, 다음과 같은 제약 을 만족한다.

$$\mu_{E_N}(t) \leq \min(a_1/\tau(A_1), a_2/\tau(A_2), \dots, a_n/\tau(A_n)).$$

$\mu_{E_N}(t)$ 이 정의 4의 제약을 만족해야 하는 이유는 E_N 이 $\tau(A_1) \times \tau(A_2) \times \dots \times \tau(A_n)$ 의 퍼지 부분 집합이기 때문이다.

즉 $t = \langle a_1, a_2, \dots, a_n \rangle$ 이 E_N 에 속하기 위해서는 먼저 t 가 $\tau(A_1) \times \tau(A_2) \times \dots \times \tau(A_n)$ 에 속해야 하고, 이 속하는 정도는 퍼지 표준 연산자를 사용할 경우 $\min(a_i/\tau(A_i), a_2/\tau(A_2), \dots, a_n/\tau(A_n))$ 으로 정량화될 수 있다.

퍼지 튜플과 퍼지 클래스의 한 예를 들어 보면 다음과 같다.

[예 1] 'commuter_bus_route' 클래스는 다음과 같이 명시된다.

$E_{commuter_bus_route}$

tuple	length_of_route	traffic_portion	degree_of_congestion	over_traffic	vehicle	μ
e1	2500	high	0.7	crowd	c1	0.8
e2	long	0.8	low	200	c2	0.7
e3	1500	0.7	0.75	300	c3	0.8

$T_{commuter_bus_route}$

$$\begin{aligned}
 long &= \int \mu_{long}(C_i)/C_i \\
 crowd &= \int \mu_{crowd}(C_i)/C_i \\
 \mu_{long}(x) &= (1+1/1000 \times |2000-x|)^{-1} \\
 \mu_{crowd}(x) &= (1+1/1000 \times |400-x| \times 2)^{-1} \\
 \mu_{short}(x) &= 1 - \mu_{long}(x) \\
 \mu_{old}(x) &= 1 - (1+1/2 \times (98-x))^{-1}, x \leq 98 \\
 \mu_{high}(x) &= x \\
 \mu_{low}(x) &= 1 - \mu_{high}(x)
 \end{aligned}$$

여기서 $c_1, c_2, c_3 \in E_{commuter_bus}$

퍼지 데이터베이스 내 클래스들은 클래스 계층 구조로 구조화되는데, 다음 정의는 이 계층 구조를 명시하고 있다.

[정의 5] 퍼지 데이터베이스, FDB는 다음과 같이 POSET(Partial Ordering SET)으로 정의된다.

$FDB = \langle C, \leq \rangle$

여기서 C 는 퍼지 클래스들의 집합이고, $C_A, C_B \in C$ 에 대하여 C_A 가 다음의 조건을 만족할 때 $C_A \leq C_B$ 로 표기하며, C_A 를 C_B 의 부클래스라고 한다.

$$\mu_{E_A}(t) \leq \mu_{E_B}(t) \text{ 이고, 여기서 } t \text{는 } E_A \text{내의 튜플이다.}$$

개념적으로, C_A 가 α 정도로 C_B 의 부클래스임을 스키마로 명시할 경우, $\leq_n \leq_\alpha$ 로 나타낸다.

클래스와 객체내에 표현되는 퍼지 정보를 표현하기 위한 구문은 다음과 같다.

```

defclass <class_name> isa['/'<real>] <class_name>
'('(<attr_name> ':' <f_attr_type>)+ <possibility>)'
make [<var> ':' ] <class_name> '('(<attr_name> ':'
<fuzzy_attr_val> | (make <class_name> {<attr_name> ':'
<fuzzy_attr_val> }+ <possibility> ')')['/'<real>]'+
<possibility>)
    
```

<possibility> = 'poss : ' <real>

클래스의 정의는 'defclass' 구문을 사용하고, 슈퍼 클래스와 서브 클래스 관계는 일반화 관계에서 나타나는 불확실성과 함께 'isa['/'<real>]'로 표현할 수 있다. 예를 들어, [정의 5]에서 C_A 가 0.8의 가능성으로 C_B 의 부 클래스 일 경우 $C_A \text{ isa}/0.8 \text{ } C_B$ 로 나타낸다. 객체의 선언은 'make' 구문을 이용한다. 애트리뷰트 값에서 발생하는 불확실성을 표현하기 위해 퍼지형 애트리뷰트가 사용되며, 'crowd', 'long', 'high', 'old'와 같은 퍼지 집합(또는 퍼지 용어)을 허용한다. 애트리뷰트 'poss'가 튜플 t 의 $\mu(t)$ 를 나타내며, 클래스 복합 계층도 역시 퍼지화되어 복합 객체에 'real'로 명시된다.

다음은 위의 구문을 이용한 스키마 정의문 및 객체 생성문의 예들을 보이고 있다.

```

defclass route_network isa object (link_component:route),
service_area:area, poss:real)
    
```

```

defclass route isa object (source_node:station,
dest_node:station, length_of_route:f_length_of_route,
over_traffic:f_over_traffic, traffic_portion:f_traffic_portion, ...,
degree_of_congestion: f_degree_of_congestion, avg_speed:
f_avg_speed, poss:real)
    
```

```

defclass commuter_bus_route isa/0.87 route
(priority_of_improve :f_priority_of_improve, vehicle:
commuter_bus, poss:real)
    
```

```

defclass subway_route isa/0.9 route (name:char,
default_subway_type:char_subway, poss:real)
    
```

```

defclass commuter_bus isa object (capacity:int,
mfg_date:int, poss:real)
    
```

```

make <E>:route (length_of_route:long, avg_speed : high,
..., poss:0.98)
    
```

```

make <E>: commuter_bus_route (source_node: <S1>,
dest_node: <S2>, length_of_route:1500, over_traffic:300, ...,
vehicle:(make <C1>:commuter_bus (mfg_date:92, ...,
poss:0.9))/0.85, poss:0.8)
    
```

'route' 클래스내에서 퍼지형 애트리뷰트는 'f_length_of_route', 'f_avg_speed' 와 같이 접미사 f를 붙여 명시한다. 'route' 클래스의 객체 생성문에서 이 애트리뷰트들은 값으로 'long', 'high' 등의 퍼지 용어를 포함한다. 'route' 객체 생성문에 나타나는 0.98은 이 객체가 'route' 클래스에 소속되는 정도를 나타낸다. 퍼지 복합 객체인 'commuter_bus_route'의 복합 애트리뷰트 'vehicle'의 복합 링크는 0.85 정도로 퍼지화 되고 있다. 객체 선언시 사용하는 <E>는 새롭게 생성된 'commuter_bus_route'의 객체, 즉 튜플의 식별자가 값으로

로 실례화(instantiation) 되는 튜플 변수이다.

4.2 규칙 평가 방법의 정형화

본 절에서는 퍼지 객체 추론을 위한 규칙 평가 프로시저를 설명한다. 다음 정의는 규칙의 기본이 되는 요소를 명시하고 있다.

[정의 6] 단일 변수 A를 갖는 패턴 필드(또는 간략하게 패턴 필드)는 규칙의 LHS (Left Hand Side)를 구성하는 기본 요소이며 다음과 같이 표현된다.

$$f = A : v,$$

여기서 v 는 일반 값(crisp value)이거나, 형 변수(type variable) $\langle X \rangle$, 또는 퍼지 집합이다. 형 변수는 관련된 형으로부터 값을 취하는 변수이다.

[정의 7] 클래스 C_N 의 패턴 필드들의 논리곱은 C_N 의 단일 패턴이라고 하며 다음과 같이 표현된다.

$$P_m^{C_N} = \langle E \rangle : N(f_1, f_2, \dots, f_p)$$

여기서 p 는 $P_m^{C_N}$ 내의 패턴 필드들의 수이며, $\langle E \rangle$ 는 튜플 변수이다.

[예 2] $f_1 = \text{traffic_portion} : \text{high}$ 이고 $f_2 = \text{degree_of_congestion} : \text{more_or_less_high}$ 라고 하면, $C_{\text{commuter_bus_route}}$ 의 한 단일 패턴은 다음과 같다.

$$P_{m_1}^{C_{\text{commuter_bus_route}}} = \langle E_1 \rangle : \text{commuter_bus_route}(f_1, f_2).$$

[정의 8] 단일 패턴들의 논리곱을 패턴 P 라 하고 다음과 같이 정의한다.

$$P = \bigwedge_{j=1}^n P_{m_j}^{C_{N_j}}$$

여기서 n 은 P 를 구성하는 단일 패턴들의 수이고, $C_{N_1}, C_{N_2}, \dots, C_{N_n}$ 는 서로 다를 필요는 없는 클래스들이다.

튜플 변수 $\langle E \rangle$ 는 형이 클래스인 형 변수의 일종이다. 따라서, 앞으로 튜플 변수의 객체들이 속하는 클래스를 명시할 필요가 있을 때는 $\text{type}(E)$ 로 표기한다.

클래스 명은 튜플 변수로 대치될 수 있으며, 이 방식은 클래스 복합 계층에서 튜플 변수 $\langle E \rangle$ 가 한 애트리뷰트의 값이 되어, $\text{type}(E)$ 내의 어떤 튜플을 참조하는 링크로 사용될 경우 필요하게 된다. 다음은 이 경우 $\langle E \rangle$ 의 사용 예를 보여주고 있다.

[예 3] 패턴 $P = P_{m_3}^{C_{\text{commuter_bus_route}}} \wedge P_{m_2}^{C_{\text{commuter_bus}}}$ 가 다음과 같이 표현된다고 하자.

$\langle E_1 \rangle : \text{commuter_bus_route} (\text{length_of_route} : \text{long}, \text{over_traffic} : \text{very_crowd}, \text{vehicle} : \langle E_2 \rangle)$

$\langle E_2 \rangle : (\text{mfg_date} : \text{old}).$

이때, $\langle E_2 \rangle$ 는 'commuter_bus_route'의 한 객체로부터 관련되는 'commuter_bus' 객체를 바로 참조하기 위해,

즉, 복합 계층 링크를 운행하기 위해 사용된다.

[정의 9] 패턴들의 논리합은 규칙의 LHS라 하며, LHS 인 L 은 다음과 같이 정의된다.

$$L = \bigvee_{i=1}^m P_i$$

여기서 $P_i, i=1, 2, \dots, m$ 는 L 을 구성하는 패턴들이다.

한 규칙은 기다리는 사건이 발생할 경우 실행 가능하여지며, 이때 LHS가 조건을 만족하면 실행된다. 규칙이 실행되면 규칙내의 튜플 변수들은 각 단일 패턴들을 만족하는 튜플들로 실례화되며, 규칙의 RHS(Right Hand Side)내 명령들을 그 실례화된 튜플들에 적용하고자 할 때, 튜플 변수가 사용된다. 앞으로, 규칙 실행의 결과로 이 명령들이 작동될 때, RHS 명령들이 수행된다고 말하기로 한다.

본 논문에서 다루는 규칙은 새로운 튜플의 삽입이나 갱신 등으로 FDB가 변경되는 순간을 기다리는 사건이 발생하는 것으로 가정하기 때문에, 사건 명세 부분을 별도로 포함하는 ECA(Event-Condition-Action) 규칙[14]은 아님을 명확히 할 필요가 있다. ECA 규칙은 능동적 시스템을 구현하는 강력한 도구로 평가되지만, 반면에 사건이 단순한 응용 환경에서는 사건 명세를 위한 확장이 오히려 프로그램의 이해도를 저하시킬 우려가 있다. 개념적 단순성을 위해, 본 논문에서는 ECA 규칙을 지원하기 위한 확장은 고려하지 않기로 한다.

L 은 패턴들의 논리합으로 이루어지며, 각 패턴들은 패턴 필드들로 구성되어 있다. 따라서 L 을 평가하기 위해서는 먼저, 이들 각 요소들과 FDB 내 튜플들과의 정합 정도를 체계적으로 산정하여야 한다. 다음은 이를 위한 정의들이다.

[정의 10] $\text{Poss}(f(t)) = \alpha, \alpha \in [0,1]$ 는 튜플 t 가 f 를 만족하는 가능성 정도를 나타낸다.

[예 4] f_3 가 $\text{length_of_route} : \text{long}$ 이라고 할 때, 튜플 $e_3 = \text{commuter_bus_route} (\text{length_of_route} : 1500, \text{over_traffic} : 300, \text{traffic_portion} : 0.7, \text{degree_of_congestion} : 0.75, \text{vehicle} : c_3, \text{poss} : 0.8)$ 에 대해 $\text{Poss}(f_3(e_3)) = \mu_{\text{long}}(1500) = 0.67$ 이다.

[정의 11] $P_m^{C_N} = N(f_1, f_2, \dots, f_p)$ 이 C_N 의 단순 패턴 이라면, E_N 에 속하는 t 가 $P_m^{C_N}$ 을 만족할 가능성 정도는 다음과 같다.

$$\text{Poss}(P_m^{C_N}(t)) = \min(\text{Poss}(f_1(t)), \text{Poss}(f_2(t)), \dots, \text{Poss}(f_p(t)), \mu_{E_N}(t)).$$

퍼지 and/or 연산의 결과는 응용 도메인의 특성에 따라 달리 얻어져야 하는 것이 보통이기 때문에, 퍼지 min/max 표준 연산자를 포함하여, 확률합/확률적, 격렬합

/격렬적 등 수많은 연산자들이 제안되었다[20]. 본 논문에서는 이들 중 min/max 퍼지 표준 연산자가 개념적으로 이해하기 쉽고 또 일반적이기 때문에, 앞으로 이 연산자를 사용하기로 한다.

[예 5] $f_1 = \text{over_traffic} : \text{very_crowd}$ 이고, $P_{m_3}^{C_{\text{commuter-bus-route}}}$ = $\langle E_1 \rangle : \text{commuter_bus_route}(f_3, f_4, \text{vehicle} : \langle E_2 \rangle)$ 라면, $\text{Poss}(P_{m_3}^{C_{\text{commuter-bus-route}}}(e_3)) = \min(\text{Poss}(f_3(e_3)), \text{Poss}(f_4(e_3)), 0.85, 0.8)$
 $= \min(\mu_{\text{long}}(1500), (\mu_{\text{crowd}}(300))^2, 0.85, 0.8)$
 $= \min(0.67, 0.69, 0.85, 0.8) = 0.67$

$\text{Poss}(P_{m_n}^{C_N}(t))$ 는 $P_{m_n}^{C_N}$ 내의 튜플 변수 $\langle E \rangle$ 가 객체로 t 를 취할 가능성 정도로 볼 수 있다. 여기서 퍼지화된 복합 링크 값 $\mu_{\langle E \rangle}(c_3)$, 즉 0.85 는 $\mu_{\langle E \rangle}(c_3) \geq \mu(e_3) = 0.8$ 을 만족하고 있음을 유념하자. 그렇지 않다면 $\mu(e_3) = 0.8$ 는 불가능하게 된다(정의 4 참조).

[정의 12] $P = (P_{m_1}^{C_{N_1}}) \wedge (P_{m_2}^{C_{N_2}}) \wedge \dots \wedge (P_{m_n}^{C_{N_n}})$ 을 클래스 $C_{N_1}, C_{N_2}, \dots, C_{N_n}$ 의 패턴이라고 하면 $t_1 \in E_{N_1}, t_2 \in E_{N_2}, \dots, t_n \in E_{N_n}$ 이 P 를 만족할 가능성 정도는 다음과 같다.

$$\text{Poss}(P(t_1, t_2, \dots, t_n)) = \min(\text{Poss}(P_{m_1}^{C_{N_1}}(t_1)), \text{Poss}(P_{m_2}^{C_{N_2}}(t_2)), \dots, \text{Poss}(P_{m_n}^{C_{N_n}}(t_n))).$$

여기서 t_1, t_2, \dots, t_n 은 서로 다를 필요는 없는 튜플들이다.

[예 6] 튜플 $c_3 = \text{commuter_bus}(\text{mfg_date} : 92, \text{poss} : 0.9)$ 이고, $f_5 = \text{mfg_date} : \text{old}$, $P_1 = (P_{m_3}^{C_{\text{commuter-bus-route}}}) \wedge P_{m_2}^{C_{\text{commuter-bus}}}$ 라면, $\text{Poss}((P_{m_3}^{C_{\text{commuter-bus-route}}}(e_3))) = 0.67$ (예 5),

$$\text{Poss}(P_{m_2}^{C_{\text{commuter-bus}}}(c_3)) = \min(\text{Poss}(f_5(c_3)), 0.9) = \min(0.75, 0.9) = 0.75 \text{ 이다.}$$

따라서, $\text{Poss}(P_1(e_3, c_3)) = \min(\text{Poss}(P_{m_3}^{C_{\text{commuter-bus-route}}}(e_3)), \text{Poss}(P_{m_2}^{C_{\text{commuter-bus}}}(c_3))) = 0.67$.

이제, 다음의 두가지 부수적인 정의들을 이용해 LHS를 평가하도록 한다. 두 정의들은 LHS 내 튜플 변수들을 체계적으로 표기하기 위하여 필요하다.

[정의 13] 튜플 리스트 E 는 다음과 같이 정의된다.

1. 튜플 변수 $\langle E \rangle$ 는 튜플 리스트 E 이다.
2. $\langle t_1, t_2, \dots, t_n \rangle \in E_{N_1} \times E_{N_2} \times \dots \times E_{N_n}$ 은 튜플 리스트 E 이다.
3. 만약 E_1, E_2, \dots, E_n 이 튜플 리스트이면, $\langle E_1, E_2, \dots, E_n \rangle$ 도 튜플 리스트이다.

4. 그 외에는 튜플 리스트가 될 수 없다.

[정의 14] E 와 E' 를 튜플 리스트라 하자. 만약 E' 내의 모든 튜플 변수가 E 에도 나타난다면 E' 는 E 의 부리스트라고 하며, $E' \subseteq E$ 로 나타낸다.

규칙의 평가는 LHS 부분을 평가하는 것으로 볼 수 있다. 다음은 이 LHS 평가를 위한 정의이다.

[정의 15] $L = \bigvee_{i=1}^m P_i$ 를 LHS라 하자. 만약 $E = \langle E_1, E_2, \dots, E_m \rangle$ 이 L 의 튜플 리스트라면 $\text{Poss}(L(E)) = \max(\text{Poss}(P_1(E_1)), \text{Poss}(P_2(E_2)), \dots, \text{Poss}(P_m(E_m)))$. 여기서 E_k 는 $P_k, k=1, \dots, m$ 의 튜플 리스트이다.

[예 7] P_2 를 $P_{m_1}^{C_{\text{commuter-bus-route}}} = \langle E_1 \rangle : \text{commuter_bus_route}(f_1, f_2)$ 라 하고, $L = P_1 \vee P_2$ 라 하면, $\text{Poss}(P_1(e_3, c_3)) = 0.67$ (예 6) 이고, $\text{Poss}(P_2(e_3)) = \min(\text{Poss}(f_1(e_3)), \text{Poss}(f_2(e_3)), 0.8) = \min(0.7, 0.75, 0.8) = 0.7$.

따라서, $\text{Poss}(L(\langle \langle e_3, c_3 \rangle, \langle e_3 \rangle \rangle)) = \max(\text{Poss}(P_1(\langle e_3, c_3 \rangle)), \text{Poss}(P_2(\langle e_3 \rangle))) = 0.7$.

[정의 16] E 가 LHS의 튜플 리스트라면

$$\text{FDB} \vdash_a L(E) \Leftrightarrow \text{Poss}(L(E)) = a$$

[예 8] 예 7의 L 에 대하여

$$\text{FDB} \vdash_a L(\langle \langle e_3, c_3 \rangle, \langle e_3 \rangle \rangle) \text{이며, 이때 } a = \max(0.67, 0.7) = 0.7 \text{이다.}$$

정의 15에서 LHS가 논리합의 형태를 취하고 있는 이유를 살펴보자. 다음은 동일한 규칙 명 γ 를 갖는 두 개의 규칙이다.

$$P_1(E_1) \Rightarrow \text{ACT}(E')$$

$$P_2(E_2) \Rightarrow \text{ACT}(E')$$

여기서 ACT는 γ 의 수행부이고, $E' \subseteq E_1, E' \subseteq E_2$ 이다. 이때, $(\text{not } P_1(E_1) \vee \text{ACT}(E')) \wedge (\text{not } P_2(E_2) \vee \text{ACT}(E')) \Leftrightarrow \text{not } (P_1(E_1) \vee P_2(E_2)) \vee \text{ACT}(E')$ 이므로, 두 규칙은 $P_1(E_1) \vee P_2(E_2) \Rightarrow \text{ACT}(E')$ 이 된다.

규칙 γ 의 RHS는 수행들의 논리곱으로 구성되며, 수행부는 LHS가 FDB에 의해 만족될 때 수행된다. 수행의 결과는 FDB를 변경시킬 수도 있기 때문에, 수행부는 또 다른 규칙의 실행을 야기할 수 있다. 규칙 γ 이 실행될 때, γ 의 모든 튜플 리스트는 FDB의 튜플들에 의해 실행되다. 다음 정의 17은 RHS를 포함하는 규칙을 정의하고 있다.

[정의 17] L 과 R 을 각각 LHS와 RHS라 하면, 규칙 γ 은 다음과 같다.

$$L(E) \Rightarrow R(E'), \text{ 그리고 } R(E') = \bigwedge_{i=1}^l \text{ACT}_i(E' : P_{m_i})$$

여기서 ACT_i 는 한 수행부이고, E'_i 는 튜플 변수이며, P_{m_i} 는 클래스 이름을 갖지 않는 $\text{type}(E'_i), i = 1, \dots, l$ 의

패턴이다. 패턴들은 필요할 때만 명시된다. 또, E 와 E' 는 $E' \subseteq E$, $E' = \langle E'_1, E'_2, \dots, E'_n \rangle$ 을 만족하는 튜플 리스트이다.

다음은 규칙 γ 의 명세 구문과 예를 보이고 있다.

```
defrule <rule_name>
'('{<var>|not ' [ <class_name>] '(' [!']
<attr_name>:'<attr_val>'+ ')'+
[ ('with:' <alpha>)]
=>
{<command> [ <var> ':' <class_name> '('
{<attr_name>:'<attr_val>'+ ')'+}'
<attr_val> = <fuzzy_attr_val>|<var>
<command> = modify | make
```

[예 9] 다음은 우선적으로 개선해야 할 노선을 찾는 퍼지 규칙이다.

```
defrule find_route_of_prior_improve
(<E1>:commuter_bus_route (over_traffic:very_crowd,
length_of_route:long, vehicle:<E2>))
<E2>:( mfg_date:old )
=>
modify <E1>: ( priority_of_improve : high ))
defrule find_route_of_prior_improve
(<E>:commuter_bus_route ( traffic_portion:high,
degree_of_congestion:high)
=>
modify <E> : (priority_of_improve : high ))
```

개념적으로 L 은 하나의 중첩된 클래스로 볼 수 있고, L 의 튜플 리스트 E 는 가능성 $\mu_L(E)$ 로 L 에 정합되는 중첩된 튜플로 볼 수 있다. 일반적으로 $FDB \vdash_{\alpha_1} L(E)$ 이고 $\mu_L(E) = \alpha_2$ 라면 $\alpha_1 \neq \alpha_2$ 인데, 이것은 전자는 E 가 FDB 에 포함되는 가능성을 고려한 반면, 후자는 이를 고려하지 않기 때문이다.

규칙 γ 의 성공적인 평가를 위해서는 L 을 만족하는 가능성 정도를 R 에 반영하는 γ 의 함축(implication) 변환 규칙이 필수적이다. 퍼지 추론에서 함축 문제를 해결하기 위한 많은 연구들이 수행되어왔다[15][16][17]. 이들 중 가장 최근의 연구로, Raju[15]는 퍼지 관계형 데이터베이스에서 Rescher-Gaines 변환 규칙을 퍼지 함축으로 채택하였다. 즉, 규칙 γ 을 $L(E) \Rightarrow ACT(E')$ 로 두었을 때, γ 은 일종의 E 와 E' 간의 퍼지 관계이므로 $\mu_\gamma(E, E')$ 가 정의될 수 있고, 그 정도 값은 식 (3.1)에 의해 할당된다. 직관적으로 (3.1)은 γ 을 참으로 만들기 위해서는 E 가 L 에 부합되는 정도보다 더 높은 가능성 정도로 ACT 가 E' 를 $type(E')$ 의 멤버로 만들어야함을 의미한다.

$$\mu_\gamma(E, E') = 1, \quad \mu_L(E) \leq \mu_{type(E')}(E') \text{ 일 때,} \\ = 0, \quad \text{그외.} \quad (3.1)$$

그러나 변환 규칙 (3.1)은 L 에서 일어나는 정합의 정도가 ACT 에 반영되지 못한다는 문제점이 있다. 가령, L 에서 정합이 극히 적은 정도로 발생할 경우에도 ACT 는 아주 높은 가능성으로 수행될 수 있다. 따라서 본 논문에서는 새로운 변환 규칙을 제시하고 이를 (3.1)과 비교함으로써 본 방식의 장점을 보다 명확하게 하도록 한다.

[정의 18] $\alpha \in [0, 1]$ 이고 γ 을 $L(E) \Rightarrow ACT(E')$ 라고 하면,

$$\mu_\gamma(E, E') = \alpha, \quad \mu_L(E) \leq \mu_{type(E')}(E') \text{ 일 때} \\ = 0, \quad \text{그외}$$

여기서 α 는 보조정리 1에 의해 주어진다.

[보조정리 1] $\mu_\gamma(E, E')$ 가 정의 18에 의해 주어졌다고 가정하면,

$$\mu_\gamma(E, E') = \mu_{type(E')}(E'), \quad \mu_L(E) \geq 0.5 \text{ 이고 } \mu_L(E) \leq \mu_{type(E')}(E') \text{ 일 때} \\ = 0, \quad \text{그외.} \quad (3.2)$$

증명: $\gamma = \text{not } L(E) \vee ACT(E')$ 이기 때문에 $\mu_\gamma(E, E') = \max(1 - \mu_L(E), \mu_{type(E')}(E'))$ 이다. 따라서 $\mu_L(E) \leq \mu_{type(E')}(E')$ 이고 $\mu_L(E) \geq 0.5$ 라면, $\mu_{type(E')}(E') \geq (1 - \mu_L(E))$ 이고 $\mu_\gamma(E, E') = \mu_{type(E')}(E')$ 이다. 만약 그렇지 않다면 정의 18에 의해 $\mu_\gamma(E, E') = 0$ 가 된다.

[18]에 따르면, 이 변환 규칙은 modus ponens와 부울 함축(boolean implication)도 만족한다.

(3.1)의 단점은 보조정리 1의 (3.2)와 비교하면 더욱 명확해진다. $\mu_L(E) \leq \mu_{type(E')}(E')$ 이고 $\mu_L(E) \geq 0.5$ 라 가정하자. 만약 $\mu_\gamma(E, E') = \mu_{type(E')}(E')$ 대신 $\mu_\gamma(E, E') = 1$ 을 선택한다면 $\mu_{type(E')}(E') = 1$ 이 될 것이다. 이것은 만약 $\mu_L(E) \leq \mu_{type(E')}(E')$ 이고 $\mu_L(E) \geq 0.5$ 라면 $ACT(E')$ 이 완전한 가능성으로 수행됨을 의미한다. 반면에, (3.2)는 이 경우에, $ACT(E')$ 가 $\mu_\gamma(E, E')$ 의 가능성으로 수행됨을 의미한다.

구현의 관점에서, $\mu_\gamma(E, E')$ 는 $ACT(E')$ 의 수행 효과 정도를 나타내는 것으로 볼 수 있다. 더 나아가, $\mu_\gamma(E, E')$ 는 E 가 L 에 정합되는 정도 $\mu_L(E)$ 로 해석될 수도 있는데, 그 이유는 수행의 효과가 L 에서 이루어진 정합으로부터 야기되었고, 그 수행의 정도는 정합의 정도에 비례하는 것으로 볼 수 있기 때문이다. 이러한 해석은 다음 보조 정리에 의해 검증될 수 있다.

[보조정리 2] $E \subseteq E'$ 일 때 γ 을 $L(E) \Rightarrow ACT(E')$ 라 하자. 그러면, $\mu_L(E) \geq 0.5$ 이고 $\mu_L(E) \leq \mu_{type(E')}(E')$ 일 때, $\mu_{type(E')}(E') = \mu_L(E)$ 이다.

증명: [19]에 따르면,

$\mu_{\text{type}(E')}(E') = \min(\mu_r(E, E'), \mu_L(E))$ (3.3)
 이고, 보조정리 1을 (3.3)에 적용시키면 $\mu_L(E) \geq 0.5$ 이고
 $\mu_L(E) \leq \mu_{\text{type}(E')}(E')$ 라면, $\mu_{\text{type}(E')}(E') = \min(\mu_{\text{type}(E')}(E'), \mu_L(E))$ 이 된다. 따라서, $\mu_{\text{type}(E')}(E') = \mu_L(E)$ 이다.

규칙의 LHS가 만족되면 추론 엔진은 RHS를 수행한다. 그러나 퍼지 정합의 정도가 일정한 기준치보다 낮을 경우 RHS가 수행되지 않도록 하기 위해서는 임계값(threshold value)을 규칙의 LHS에 명시하면 된다. 임계값은 α -cut 값이라 하며, LHS의 정합 정도가 α' 보다 크다면 규칙의 LHS가 만족된 것으로 간주하여 RHS가 수행된다. 다음 정리는 규칙의 실행을 위한 조건과 수행 효과를 명시하고 있다.

[정리 1] 규칙 $r, L(E) \Rightarrow R(E')$ 을 수행될 규칙이라고 하고, $0 < \alpha' \leq 1$ 는 임계 값, $E \subseteq E'$ 일 때 $\beta = \text{Poss}(FDB(E))$ 라고 하자. 여기서 $\text{Poss}(FDB(E))$ 는 FDB내의 튜플들이 E 에 실재화될 가능성을 나타낸다. 이때, 만약 다음의 조건이 만족되면 R 은 α 의 가능성으로 수행된다.

$$FDB \vdash_{\alpha} L(E) \text{ 이고 } \alpha = \min(\mu_L(E), \beta) \geq \alpha'$$

증명: 이 정리의 증명은 보조정리 1과 보조정리 2에 의해 바로 증명된다.

이제, 임계값이 0.6일 때, 객체 e_3 에 대해 예 9의 규칙이 평가되는 과정을 살펴보자. 첫 번째 규칙의 LHS에 위의 객체가 정합될 가능성은 0.67이고, 두 번째 규칙의 LHS에 정합될 가능성은 0.7이다. 몇 개의 규칙이 논리합(or) 관계일 때, 가장 높은 가능성이 선택되므로 이 예의 최종 가능성 값은 $\max(0.67, 0.7) = 0.7$ 이다. 이 값은 규칙의 임계 값보다 크므로 규칙이 실행되고 객체가 생성된다. 0.7은 수정된 객체가 RHS 부분에서 명시된 클래스에

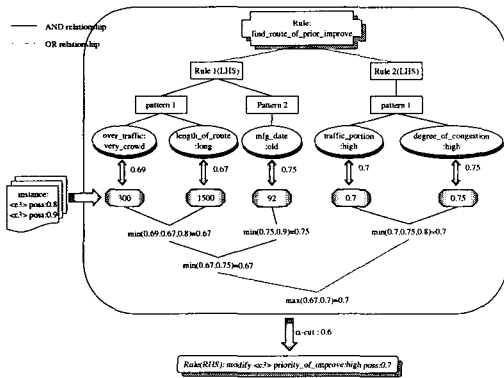


그림 2 퍼지 규칙 평가 과정 예

속할 소속성 정도로 해석된다. 그림 2는 이 규칙의 평가 과정을 보여주고 있다. 단, 여기서 복합 계층 구조의 퍼지화 정도, 즉 $\mu_{c(c_3)}=0.85$ 는 그림의 이해도를 돕기 위해 생략했다.

다음으로 퍼지 isa 관계의 실행시 시멘틱을 살펴보자. 이 시멘틱을 설명하기 위해 다음의 클래스와 객체 규칙을 고려한다.

```

defclass subway_route isa/0.9 route (name:char,
default_subway_type:char_subway, poss:real)
make <SR>: subway_route (name:line1,
no_of_passengers:2000, traffic_portion:0.7,
max_no_of_pssengers:1000, poss:0.85)
defrule rushhour_over_traffic
(<R>:route (no_of_passengers:<X>, traffic_portion:<Y>,
max_no_of_passengers:<Z>))
(with:0.8)
=>
modify <R>: ( over_traffic:<X>*<Y>-<Z>))
    
```

'subway_route'는 'route'의 부클래스이므로, 'route'를 위해 정의된 'rushhour_over_traffic' 규칙은 'subway_route'에 적용될 수 있다. 추론 엔진은 먼저 'subway_route'의 객체가 규칙의 LHS 부분에 정합될 가능성을 평가한다. <SR>의 값으로 할당되는 객체, sr 이 'subway_route' 클래스에 소속될 정도는 $\mu_{\text{subway_route}}(\text{sr})=0.85$ 이고, $C_{\text{route}} \geq 0.85 C_{\text{subway_route}}$ 이다. 따라서, $\mu_{\text{route}}(\text{sr})$ 을 sr이 'route' 클래스에 소속될 정도라고 하면, 규칙의 LHS 부분을 sr이 만족할 가능성은 $\min(0.9, \mu_{\text{route}}(\text{sr}))$ 로 정량화될 수 있다. 그러나, $\mu_{\text{route}}(\text{sr})$ 값은 명

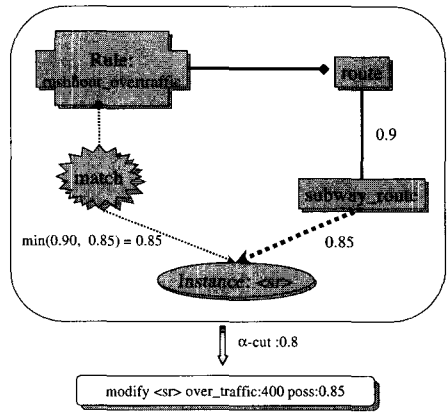


그림 3 일반화 계층에서 퍼지 추론

시되지 않기 때문에, sr이 'subway_route'의 객체가 되기 위해서는 먼저 'route'의 객체가 되어야 한다는 정의 5의 가정을 사용할 경우, $\mu_{route}(sr) \geq \mu_{subway_route}(sr) = 0.85$ 로 볼 수 있다. 따라서 이 가정하에서, $\min(0.9, \mu_{route}(sr)) \geq \alpha = 0.8$ 이므로 이 규칙은 수행된다.

5. 결 론

본 논문에서는 규칙 기반 추론과 객체 지향 모델링, 퍼지 추론이 체계적으로 병합되어 있는 퍼지 객체 추론 개념 모델을 제안하고 정형화하였다. 퍼지 클래스, 일반화 계층, 복합 계층을 포함하는 퍼지화된 데이터베이스 구조가 기술되었으며, 이들의 표현 방식과 추론 메카니즘이 정형화되었다. 제안 모델은 두가지 주된 장점을 제공하는데, 첫째, 객체 지향 방식의 병합으로 기존의 규칙 기반 전문가 시스템의 모델링 능력을 향상 시켰으며, 둘째, 복잡한 공학 응용 도메인 모델링시 나타나는 불확실성을 객체 지향 방식에 따라 자연스럽게 표현할 수 있도록 하였다. 보다 정교한 시스템의 개발에 이 모델이 이용되기 위해서는 몇가지 보완점이 필요한데, 예를 들어, 전칭 한정사를 도입하고, ECA 규칙 지원 등 동적 메카니즘 구현에 대한 연구들이 부수적으로 필요하다.

참 고 문 헌

- [1] Adlassnig, K. P. Fuzzy set theory in medical diagnosis. IEEE Transactions on Systems Man, and Cybernetics, SMC-16, 1986, 260-265.
- [2] Zemankova, M. FILIP: A Fuzzy Intelligent Information System with Learning Capabilities. Information System, 14(6), 1989, 473-486.
- [3] Binaghi, E. A fuzzy logic inference model for a rule-based system in medical diagnosis. Expert Systems, 7(3), 1990, 134-141.
- [4] Orchard, R. A. FuzzyCLIPS, a users guide. Version 6.02, Knowledge Systems Laboratory: Institute for Information Technology, National Research Council, Canada, September, 1994.
- [5] Sheng, O. R. L. and Wei, C. P. Object-oriented modeling and design of coupled knowledge-base/database systems. In: N. J. Cercone, eds. Proceedings of the International Conference on Data Engineering, IEEE, Tempe, Arizona, USA, 1992, 98-105.
- [6] Shyy, Y. M. and Su, S. Y. W. K: A high-level knowledge base programming language for advanced database applications. In: James Clifford, Roger King, eds. ACM SIGMOD, 1991, Denver, Colorado, USA, May, 1991, 29-31.
- [7] Dechamboux, P. and Roncancio, C. Peplomd : an object-oriented database programming language extended with deductive capabilities, In: D. Karagiannis, eds. Proceeding of Database and Expert Systems Applications, Athens, Greece, 1994, 2-14.
- [8] Srivastava, D., Ramakrishnan, R., Seshadri, P. and Sudarshan, S. Coral ++ : adding object-orientation to a logic database language. In: R. Agrawals, eds. Proceedings of the 19th VLDB Conference, Dublin, Ireland, 1993, 158-170.
- [9] Diaz, O., Paton, N., and Gray, P. Rule Management in Object-Oriented Databases: A Uniform Approach. In G. M. Lohman, eds. Proceedings of the 17th International Conference on Very Large Data Bases, Barcelona. Sept. 1991, 317-326.
- [10] Alashqur, A. M., Su, S. Y. W. and LAM, H. A rule-based language for deductive object-oriented databases. In: J. Urban, eds. Proceedings International Conference on Data Engineering, Los Angeles, California, USA, 1990, 58-67.
- [11] Su, S. Y. W. and Lam, H. X. An object-oriented knowledge base management for supporting advanced applications. In: Proceeding of 4th International HK Computer Society Database Workshop, 1992.
- [12] Yang, H. J., Yang, J. D. and Kim, Y. H., ICOT: An Integrated C-Object Tool for Knowledge-Based Programming, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(3), 1998, 273-293.
- [13] Martin, T. P., Baldwin, J. F. and Pilsworth, B. W. The implementation of Fprolog -a fuzzy prolog interpreter. Fuzzy Sets and Systems, 23(1), 1987, 119-129.
- [14] 양재동, 정선호, Chakravarthy, S., 능동 데이터베이스 상에서 뒤부분 기반으로 복합 사건을 탐지하기 위한 기본 구조, 정보과학회 논문지, 25(10), 1998, 1466-1477.
- [15] Raju, K. V. S. V. N. and Majumdar, A. K. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. ACM TODS, 13(2), June 1988, 129-166.
- [16] Fukami, S., Mizumoto, M., and Tanaka, K. Some Considerations on Fuzzy Conditional Inferences, Fuzzy sets and Systems, 4, 1980, 243-273.
- [17] Zadeh, L. A. A Theory of approximate reasoning. In: J. hayes, et al., eds. machine intelligence 9, ellis horwood ltd., sussex, uk, 1979, 149-194.
- [18] Yang, J. D. and Lee, D., Incorporating Concept-based Match into Fuzzy Production Rules. Information Sciences, 104(3/4), 1997.
- [19] Zadeh, L. A. The role of fuzzy logic in the

- management of uncertainty in expert systems. *Fuzzy Sets and Systems*, 11(3), 1983, 199-227.
- [20] COX, Earl, *The Fuzzy Systems Handbook: a Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems*, Academic press, 1994.
- [21] Buckley, J.J. Siler, W and Tucker, D. FLOPS, A fuzzy expert system: application and perspectives, In *Fuzzy Logics in Knowledge Engineering*, C.V. Negoita and H. Prade, Eds., Verlag TUV Rheinland, Germany, 1986.
- [22] Leung, K.S. and Lan, W, Fuzzy concepts in expert system, *IEEE computer*, Sept., 1988, 43-56.
- [23] Forgy, C.L., OPS5 user's manual. Technical Report CMU-CS-81-135, Carnegie-Mellon University, Pittsburg, PA 15213, 1981.
- [24] IBM, Enhanced Common Lisp Production System User's Guide and Reference, First Edition, International Business Machine corporation, 1988.

양 재 동

정보과학회논문지: 데이터베이스
제 27 권 제 1 호 참조



양 형 정

1991년 2월 전북대학교 전산통계학과(학사). 1993년 2월 전북대학교 전산통계학과(석사). 1998년 8월 전북대학교 전산통계학과(박사). 2000년 3월 ~ 현재 동신대학교 컴퓨터 응용학과군 전임강사. 관심분야는 OODBs, 이미지/ 비디오 정보

검색,