

불명확 연역 데이터베이스를 위한 포괄적 및 배타적 or 해석

석 윤 영[†] · 전 종 훈^{††}

요 약

불명확절을 포함하고 있는 연역 데이터베이스에서 or 연산자를 해석하고 처리하기 위해서 Lassez의 강력한 모델의미해석(Lassez's Strong Model Semantics: LSMS)[9]을 적용하면, 단순하고 적절한 정의에 의해 배타적(exclusive) 의미해석과 포괄적(inclusive) 의미해석이 모두 가능하게 된다. 그러나, 기존의 의미해석 방식들은 배타적 의미해석과 포괄적 의미해석 중 한 가지만을 지원하거나, 두 가지를 다 지원하기 위해서는 복잡한 처리과정을 거쳐야 한다는 단점이 있었다. 따라서, 본 논문에서는 불명확 정보 표현의 장점과 문제점들을 연구하고, 이에 대한 해법으로서 LSMS가 어떻게 포괄적 or 해석과 배타적 or 해석을 모두 제공할 수 있는가를 보인다. 또한 LSMS의 특성을 다양하게 연구하고 기존의 의미해석들과 비교 분석하여, 그들과의 의미적 차이점들을 엄밀하게 비교하여 정리한다. 결론적으로 LSMS만이 포괄적 및 배타적 해석을 지원하는데 적절하면서도 강력한 의미해석 방식임을 보인다.

Inclusive and Exclusive or Interpretation for Indefinite Deductive Databases

Yoon-Young Sok[†] · Jong-Hoon Chun^{††}

ABSTRACT

In order to properly interpret and process or operators in deductive databases including indefinite clauses, we propose to use Lassez's Strong Model Semantics (LSMS)[9] which is reasonably simple yet powerful enough to support both exclusive and inclusive interpretations. Conventional semantics either fail to support both interpretations or simply too complex. Therefore, in this paper we study advantages and difficulties of representing indefinite information, and as for the solution to difficulties, we show how LSMS can be used to support both inclusive or and exclusive or interpretations. We also investigate and analyze its properties and show how it semantically differs from others. We believe that LSMS is the only "reasonably simple" semantics that supports both inclusive and exclusive interpretations.

키워드 : 연역 데이터베이스(deductive database), 논리 데이터베이스(logic database), 의미해석(semantics), 모델 의미해석(model semantics), 배타적 or(exclusive or), 포괄적 or(inclusive or), 해석(interpretation)

1. 서 론

연역 데이터베이스와 논리 프로그래밍을 위한 불명확 정보 표현의 장점과 문제점은 이미 오랜 기간동안 체계적으로 잘 연구되었고 다양한 해법이 제시되어 왔다[4, 23, 15]. 연역 데이터베이스와 논리 프로그래밍에 대한 선언적 의미해석을 정의하는 것은 일반적으로 최소성의 원칙(minimality principle)이 가장 적절한 것으로 알려져 있다[22]. 각 접근방법론적인 특징과 차이점을 살펴보면 다음과 같다.

1.1 배타적 or 해석

일반 논리 프로그래밍에서 유일한 최소 모델(unique minimal model)이 없을 때 데이터베이스의 의미를 정의하는데

혼란이 발생하는 문제를 해결하기 위한 방안의 하나로서, 어떤 최소 모델은 계층(stratified) 데이터베이스[24]에서 다른 모델들보다 우선적으로 적용되거나, 데이터베이스의 규칙을 기술할 때에 부정적으로 기술한 것에 우선순위를 부여해서 해결하기도 한다[1]. Przymusiński[13, 14]는 분리 절을 허용함으로써 계층을 확장한다. 또는 우선순위 없이 최소모델을 확장하는 방법으로는 GCWA(Generalized Closed World Assumption)[12]와 EGCWA(Extended Generalized Closed World Assumption)[27], WGCWA(Weak Generalized Closed World Assumption)[15]가 있으며 이들은 모두 CWA(Closed World Assumption)[16]의 개념을 확장한 것이다.

1.2 3차 논리(three-valued logic)의 이용

불명확 정보를 처리하기 위하여 많은 연구자들이 3차 논리(three-valued logic)를 이용하는 것을 제안하였다. Van Gelder는 3차 WFS(Well-Founded Semantics)[25, 26]를 제시하

* 본 연구는 명지대학교 신진교수 교내연구 장려금 지원사업의 결과임.

† 종신회원 : 수원여자대학 컴퓨터응용학부 교수

†† 정회원 : 명지대학교 컴퓨터학부 교수

논문접수 : 2001년 11월 5일, 심사완료 : 2001년 12월 26일

였고 모든 논리 프로그램에 적용됨을 보였다. Gelfond는 계층 프로그램을 위한 재귀적 정수 의미해석보다 더 일반화된 안정모델(*Stable Model*)[6]을 정의하였다. Buccafurri와 Leone [3]는 순서 논리를 위해서 안정모델을 보다 확장하였다. Marco-schaerf[11]와 You[28]에 의해서 다양한 의미해석들에 대한 비교연구가 되었으며, 이 연구들은 *배타적 불명확 절(clause)*을 다루는 *최소모델 접근법*으로 잘 알려져 있다.

그러나 *포괄적 or 해석*이 요구될 때 *최소 모델 접근법*은 논리적으로 심각한 결함이 있다. 예로써 수영을 즐기는 Minsu와 Mirwoo이라는 쌍둥이의 경우를 생각해보자. 이웃의 한 사람 A가 수영장에 갔다가 쌍둥이 중의 한 사람을 보았지만 누구를 보았는지 모른다고 하자. 이 경우 데이터베이스 절을 아래의 DB_1 과 같이 쓸 수 있다.

$$DB_1 = \{swimming(Minsu) \vee swimming(Mirwoo)\}$$

나중에 다른 이웃B가 수영장에 있는 사람이 Minsu이라는 것을 확인하면 이 사실이 DB_1 에 새로운 절로서 (*swimming(Minsu)*)이 추가된 DB_1' 로 데이터베이스의 상태가 전이한다.

$$DB_1' = \{swimming(Minsu), swimming(Minsu) \vee swimming(Mirwoo)\}$$

데이터베이스 DB_1' 에 의해서 *swimming(Minsu)* or *swimming(Mirwoo)* 인 것은 참이다. 또한 *swimming(Minsu)*는 확실하게 참이 된다. 그러나 *swimming(Mirwoo)*은 참인지 거짓인지 알 수 없다. 만약 이 예를 *최소 모델 방법*에 적용하면 데이터베이스의 *최소 모델*은 (*swimming(Minsu)*)이다. GCWA의 정의에 따르면 만약 *원자식(atom)*이 모든 *최소 모델*에 속한다면 참이며, 만약 *원자식*이 어떠한 *최소 모델*에도 속하지 않는다면 거짓이다. 그 외의 모든 *원자식*은 불명확이 된다. 그러므로 GCWA에 의하면 *swimming(Minsu)*은 참이 되고, *swimming(Mirwoo)*은 거짓으로 추정된다. 그러나 위의 예는 직관적으로 생각해 볼 때 $\neg swimming(Mirwoo)$ 라고 결정할 수 있는 근거가 전혀 없다는 문제가 있다.

1.3 포괄적 or 해석

Ross[17]는 분리절의 역을 추론할 때 분리절의 본래 의미가 포괄적 해석을 강력하게 요구하므로 주의해야 한다는 것을 제안하였다. 또한, *배타적 or 해석*의 문제를 해결하기 위해 *포괄적 or 해석*을 지원하는 많은 의미해석들이 제안되었다. Sakama[19, 20]는 PMS를 제안하였다. Kong[7, 8]은 불명확 절의 포괄적 해석을 지원하는 ICWA를 제안하였다. 불행하게도 이러한 의미해석들은 대단히 복잡하고 어떤 상황에서는 엉뚱한 결론을 내리기도 한다. Ross[18]는 명확한 의미 해석을 위하여 약한 or 연산자는 분리할 것을 제안하였고, SMS(*Succinct Model Semantics*)[5]는 포괄적 or 해석의 취급을 제안하였지만 두 의미해석 모두 배타적 or 해석을 지원하지는 못한다.

1.4 포괄적 및 배타적 or 해석

포괄적 및 배타적 or 해석을 모두 지원하는 모델 의미해석 방법이 필요하다. 또 다른 데이터베이스의 예를 살펴보자. Daehee가 중학교에 다니거나 고등학교에 다닌다면 다음과 같이 DB_2 를 만들 수 있다.

$$DB_2 = \{middle_school(Daehee) \vee high_school(Daehee)\}$$

나중에 Daehee가 중학생인 것을 알게 되었다면 이 사실을 데이터베이스에 추가하여 다음과 같이 DB_2' 를 만들 수 있다.

$$DB_2' = \{middle_school(Daehee) \vee high_school(Daehee), middle_school(Daehee)\}$$

이런 경우 Daehee가 중학생이면서 동시에 고등학생일 수 없으므로 *high_school(Daehee)*은 거짓이다. 이 예는 배타적 or 해석이 적용되는 것이다.

위의 두 개의 예를 살펴보면 불명확 절을 포함하는 데이터베이스에서는 포괄적 or 해석과 배타적 or 해석이 필요하다는 것을 알 수 있다. 그러나 포괄적 or 해석은 무조건 포괄적으로 해석하며 불명확 정보가 늘어난다는 문제점이 있다. 또한 배타적 or 해석을 위해서는 무결성 제약 조건절을 사용하는 것이 필수적이다. 따라서 포괄적 or 해석과 배타적 or 해석이 모두 가능한 해석이 필요하다. 기존의 해석 방법으로서 PMS가 포괄적 or 해석과 배타적 or 해석을 모두 지원하지만, 이러한 해석을 위해서 PMS는 복잡한 분할 처리 과정이 필요하다. 따라서 포괄적 or 해석과 배타적 or 해석이 모두 가능하면서 복잡한 처리과정이 필요 없는 해석 방법이 필요하게 된다. LSMS는 복잡한 처리과정이 필요없고 기본 정의를 이용하면 포괄적 or 해석과 배타적 or 해석이 모두 가능하므로 매우 효율적인 방법이다.

그러나 LSMS의 우수한 의미해석 기법에 대한 정의 및 다른 의미해석과 비교분석한 연구가 거의 없는 실정이다. 따라서 본 논문에서는 LSMS의 명확한 정의를 나타내고, 다른 의미해석과의 비교 분석을 통해서 LSMS가 어떻게 포괄적 or 해석과 배타적 or 해석이 모두 가능한지 살펴보고, 또한 LSMS 의미해석 방법이 다른 해석보다 간단하고 적절하며 해석방법이 어떻게 다른지 고찰하기로 한다.

본 논문의 구성은 다음과 같다. 2장에서 기본개념과 논문에서 사용된 용어를 설명하고, 또한 관련 연구에 대해 간단하게 소개한다. 3장에서는 다른 의미해석과 LSMS가 어떻게 다른지 예를 통해 보여주고, 사례를 이용하여 LSMS를 정의한다. 또한 LSMS의 특성을 분석한다. 4장에서는 LSMS와 다른 의미해석을 이론적인 측면에서 비교해서 설명한다. 그리고 5장에서는 결론과 앞으로의 연구과제를 제시하겠다.

2. 관련 연구

이 장에서는 본 논문에서 사용된 기본 정의와 용어를 설명

하고 관련연구에 대해 간략하게 설명한다.

2.1 용어

항목(*term*)은 상수 혹은 변수이다. 그라운드(*ground*) 항목은 상수다. 원자식은 항목과 서술(*predicate*)로 정의된다. t_1, \dots, t_n 이 원자식이고 P 가 서술 심볼이면 $P(t_1, \dots, t_n)$ 은 원자식이다. 그라운드 원자식은 모든 항목이 그라운드 항목인 원자식이다. 리터랄(*literal*)은 긍정적(*positive*) 원자식이거나 부정적(*negative*) 원자식이다. 그라운드 리터랄은 원자식이 그라운드 원자식인 리터랄이다. 리터랄을 분리한 것을 절이라고 한다. 즉,

$$L_1 \vee \dots \vee L_n \quad (n \geq 1)$$

위의 절은 사실(*fact*)이다. 만일 $n=1$ 이면 명확 사실이고 $n>1$ 이면 불명확 사실이다. 분리 절(*disjunction*)은 명확한 원자식의 유한한 분리 절이다. 분리 절은 변수를 포함하지 않는 그라운드이다. 절은 다음과 같이 쓸 수 있다. 모든 리터랄 P_i 와 Q_j 에 대해서

$$P_1 \vee \dots \vee P_n \vee \neg Q_1 \vee \dots \vee \neg Q_m \quad (n \geq 1, m \geq 1)$$

이것은 다음과 같이 쓸 수 있다.

$$P_1 \vee \dots \vee P_n \vee \leftarrow Q_1 \wedge \dots \wedge Q_m$$

분리 절은 $n>1$ 인 절이고 이런 것은 불명확 절이라 한다. $n=1$ 인 호른(*horn*) 절은 명확 절이라 한다. 호른 데이터베이스¹⁾는 명확 절로 이루어진 논리 데이터베이스이다. 비호른(*non-horn*) 데이터베이스²⁾는 최소한 하나 이상의 불명확 절로 이루어진 논리 데이터베이스로서, 의미적으로 불명확 데이터베이스(*indefinite database*)와 같은 개념으로 사용된다.

2.2 기본 정의

데이터베이스 DB 의 HU (*Herbrand Universe*)는 DB 의 상수와 함수 심볼로부터 구성될 수 있는 모든 그라운드 항목의 집합이다. 만일 데이터베이스 DB 에 상수 심볼이 없으면 임의의 상수라도 $HU(DB)$ 에 있을 수 있다. 데이터베이스 DB 의 HB (*herbrand base*)인 $HB(DB)$ 는 모든 그라운드 원자식의 집합으로 정의되며, 그라운드 원자식은 DB 에 포함되어 있는 서술로서 HU 에 포함되어 있는 항목을 인자(*argument*)로서 대입해서 넣는다.

정의 1 : 데이터베이스 DB 를 위한 3차 해석(*three-valued interpretation*) $I = \langle T; F \rangle$ 는 그라운드 원자식 집

합의 쌍이다. 주어진 그라운드 원자식 a 는 I 에 대하여 아래와 같이 평가된다.

1. 만약 $a \in T$ 면 I 에 대하여 a 는 참이다.
2. 만약 $a \in F$ 면 I 에 대하여 a 는 거짓이다.
3. 만약 $a \in T$ 나 $a \in F$ 도 아니면 I 에 대하여 a 는 불명확이다.
4. $T \cap F = \emptyset$

우리는 T, F, U 를 각기 참, 거짓, 불명확으로 평가되는 그라운드 원자식의 집합을 표현하는데 사용하기로 한다. 데이터베이스 DB 를 위한 3차 해석 $I = \langle T; F \rangle$ 가 $TUF = HB(DB)$ 인 경우는 불명확이 배제되므로 T 와 F 의 두 개 값 해석으로 줄일 수 있다.

정의 2 : 데이터베이스 DB 의 모델 M_I 는 DB 의 모든 절을 참으로 만드는 DB 의 3차 해석 I 이다.

정의 3 : 모델 비교 연산자 \leq 는 $T \subseteq T'$ 이고 $F \supseteq F'$ 면 $\langle T; F \rangle \leq \langle T'; F' \rangle$ 와 같이 정의된다.

정의 4 : 3차 해석 I 에 대하여 데이터베이스 DB 의 모델 MM_I 가 $M_I \leq MM_I$ 인 모델 M_I 가 존재하지 않으면 최소라고 한다.

2.3 관련 연구

Kong[8]이 $GCWA$ 에 3차 해석을 적용한 내용은 다음의 정의 4와 같이 정리할 수 있다.

정의 5 : 3차 논리기반 $GCWA$ 에서 $HB(DB)$ 의 진리 대입은 다음과 같다.

1. 그라운드 원자식 a 가 모든 최소 모델의 참 집합 T 에 나타난다면 a 는 참이다.
2. 그라운드 원자식 a 가 어느 최소 모델의 참 집합 T 에서 나타나지 않는다면 a 는 거짓이다.
3. 그외의 경우 a 는 불명확이다.

아래의 예제는 정의 5를 명확하게 보여준다.

예제 1 : 단일 불명확 절인 데이터베이스를 고려하자.

$$DB = \{P(a) \vee P(b)\}$$

$P(a)$ 와 $P(b)$ 는 둘 다 불명확이다. 합집합 연산자로 표현하여 $DB' = DB \cup P(a)$ 라고 하자

$$DB' = \{P(a) \vee P(b), P(a)\}$$

그렇다면 $P(b)$ 는 모든 최소 모델의 참 집합 T 에 나타나지 않으므로 2차 논리로는 $P(a)$ 는 참, $P(b)$ 는 거짓이 된다. 그러므로 $GCWA$ 는 논리연산자 *or*의 배타적 해석만을 제공한다. 그러나 $P(b)$ 가 참이 될 가능성은 변하지 않았다. 따라서

1) 명확 데이터베이스(*definite database*)와 같은 개념으로서 데이터베이스를 구성하는 모든 절이 명확절인 데이터베이스
 2) 불명확 데이터베이스(*indefinite database*)와 같은 개념으로서 하나 이상의 불명확 절을 포함하는 데이터베이스

무조건 모든 *or* 를 배타적으로 해석하는 것은 많은 문제가 있다.

*PMS*와 *ICWA*는 포괄적 *or* 연산자의 해석을 제안하였다. 포괄적 해석을 지원하기 위해서 *PMS*와 *ICWA* 모두 불명확 데이터베이스를 여러개의 명확한 하위 데이터베이스로 분할한다. 그러나 이 방법은 데이터베이스를 분할하는 작업이 복잡하고 분할된 하위 데이터베이스의 숫자가 급격하게 증가한다는 단점이 있다.

정의 6 : *ICWA*의 *HB(DB)*의 진리 대입은 아래와 같다[7, 8]: *I-model IM*은 *ICWA*의 분할처리에 의해 생성된 각 하위 데이터베이스의 최소 모델이다.

1. 그라운드 원자식 *a*가 모든 *I-model*의 참 집합 *T*에 나타난다면 *a*는 참이다.
2. 그라운드 원자식 *a*가 어떤 *I-model*의 참 집합 *T*에 나타나지 않는다면 *a*는 거짓이다.
3. 그 외의 경우는 *a*가 모두 불명확이다.

정의 7 : *PMS*에서 *HB(DB)*의 진리 대입은 아래와 같다 [19, 20]:

분할 데이터베이스는 *PMS*의 분할 처리에 의해 생성된 하위 데이터베이스이다. 각 분할 데이터베이스를 이루는 최소 모델은 가능한 *PM* 모델이다. *PMS*에서 그라운드 원자식 *a*가 참이라는 것은 데이터베이스를 이루는 모든 *PM*에 포함되어 있다는 것과 같다. 만일 그라운드 원자식 *a*가 어느 *PM*의 참 집합 *T*에 나타나지 않았다면 *a*는 거짓이다. 그 외의 그라운드 원자식 *a*는 불명확이다.

3. LSMS(Lassez의 강한 모델 의미해석)

이 장에는 *LSMS*의 정의를 나타내고, 이 의미해석과 기존의 다른 의미해석의 차이점을 비교 분석한다.

3.1 동기

연역 데이터베이스에서 불명확 절을 사용하는 것은 불명확한 지식을 표현하려는 것에서 비롯되었다. 아래의 데이터베이스를 통하여 배타적 정보의 표현에 대해서 살펴보자.

$$DB_3 = \{female(Yejin) \vee male(Yejin)\}$$

데이터베이스 *DB₃*의 상태는 성별이므로 *Yejin*은 *male*이나 *female* 둘 중의 하나의 상태만 가능하다. 이것은 상식적으로 *DB₃*의 *or* 연산자가 배타적 처리를 필요로 한다는 것을 알 수 있다. 다음은 불명확 데이터베이스의 경우를 살펴보자.

$$DB_4 = \{like(Yejin, Chanho) \vee like(Yejin, sunmi)\}$$

*DB₄*의 *or* 연산자는 *Yejin*이 *Chanho*와 *sunmi*를 둘 다 좋아할 수 있으므로 포괄적 처리를 필요로 한다. 결과적으로 *or* 연산자는 데이터베이스의 의미해석에 따라 어떤 때는 배타적

으로, 어떤 때는 포괄적으로 처리해야 한다. 그래서 Ross[18]는 *or* 연산자를 포괄적 *or* 연산자와 배타적 *or* 연산자로 나누는 것을 제안하였다. 그러나 *or* 연산자를 배타적으로 처리할 때와 포괄적으로 처리할 때를 구분할 수 있다면 *or* 연산자를 두 가지로 구분해서 사용할 필요가 없다.

*DB₃*은 배타적 성격을 갖고 있으므로 *female(Yejin) ∨ male(Yejin)*에 대한 무결성 제약 절인 $\neg(female(Yejin) \wedge male(Yejin))$ 을 추가할 수 있다. 즉, *DB₃'*와 같이 *female(Yejin)*과 *male(Yejin)*에 대해서 하나가 참이면 다른 하나는 거짓이 되도록 상호 배타적인 표현을 할 수 있다.

$$DB_3' = \{female(Yejin) \vee male(Yejin), \neg(female(Yejin) \wedge male(Yejin))\}$$

이것은 *or* 연산자의 배타성을 강화하는 방법이며, *or* 연산자를 포괄적으로 해석할 것인가의 여부를 확인하는데 필수적이다.

위에서 살펴보았듯이 의미해석 방법을 측정할 수 있는 척도로서 불명확 정보를 줄일 수 있는 정도를 기준으로 한다면 *GCWA*를 사용한 의미해석 방법이 강력하다. *GCWA*가 강력한 이유는 배타적 논리를 사용하여 불명확 정보를 줄일 수 있으므로 강력한 것이다. 그러나 *GCWA*가 배타적 *or* 해석을 하기 위해서는 무결성제약 조건절을 사용해야 한다. 또한 불명확 정보를 잘못 해석하는 오류가 발생할 수 있다는 단점이 있다. 이에 반하여 Lassez가 제안한 강력한 의미해석은 그라운드 원자식의 진리 할당을 이용하므로 방법론적으로 아주 간단하다. 그리고 포괄적 *or* 해석과 배타적 *or* 해석 처리를 모두 다 수행할 수 있다. 따라서 *LSMS*가 가장 이상적인 의미해석 방법이라고 할 수 있다.

3.2 LSMS

정의 8 : 불명확 데이터베이스 *LSMS*에 속한 *HB(DB)*에 그라운드 원자식의 진리 할당은 아래와 같이 정의된다.

1. 만일 그라운드 원자식 *a*가 모든 모델의 참 집합 *T*에 나타날 때, *a*는 참이다.
2. 만일 그라운드 원자식 *a*가 모든 모델의 참 집합 *T*에 나타나지 않으면, *a*는 거짓이다.
3. 그외의 *a*는 불명확이다.

모든 모델의 참 집합 *T*는 모든 최소 모델의 참 집합 *T*와 동일하다. 모든 모델의 거짓 집합 *F*는 최소 모델의 거짓 집합 *F*의 부분집합이다. *LSMS*의 거짓집합 *F*는 *GCWA*의 거짓 집합 *F*의 부분집합이므로, *LSMS*의 불명확 집합은 *GCWA*의 불명확 집합을 포함하는 수퍼-집합(super-set)이다.

정의 9 : 데이터베이스 *DB*를 위한 $M_{LSMS}(DB) = \langle T ; F ; U \rangle$ 는 다음과 같은 그라운드 원자식 집합중의 하나이다.

1. 만일 DB의 그라운드 원자식 a 가 참이면, $a \in T$.
2. 만일 DB의 그라운드 원자식 a 가 거짓이면, $a \in F$.
3. 만일 DB의 그라운드 원자식 a 가 불명확이면, $a \in U$.

$M_{LSMS}(DB)$ 를 데이터베이스 DB의 Laissez의 강한 모델이라고 한다.

포괄적 or 해석과 배타적 or 해석을 모두 포함하는 다른 예를 살펴보자.

예 2 : 다음과 같은 예제 DB를 고려하자.

$DB = \{like(Soyoung, coffee), like(Soyoung, coffee) \vee (like(Soyoung, tea), pregnant(Soyoung, boy), pregnant(Soyoung, boy) \vee pregnant(Soyoung, girl), \neg(pregnant(Soyoung, boy) \wedge pregnant(Soyoung, girl))\}$

이 DB는 다음의 두 모델 M_1 와 M_2 가 있다.

$M_1 = \langle \{like(Soyoung, coffee), pregnant(Soyoung, boy)\}; \{like(Soyoung, tea), pregnant(Soyoung, girl)\} \rangle$
 $M_2 = \langle \{like(Soyoung, coffee), like(Soyoung, tea), pregnant(Soyoung, boy)\}; \{pregnant(Soyoung, girl)\} \rangle$

M_1 와 M_2 로부터 $M_{LSMS}(DB)$ 는

$M_{LSMS}(DB) = \langle \{like(Soyoung, coffee), pregnant(Soyoung, boy)\}; \{pregnant(Soyoung, girl)\}; \{like(Soyoung, tea)\}; \rangle$

LSMS 해석을 적용하면 $like(Soyoung, coffee)$ 와 $pregnant(Soyoung, boy)$ 는 모두 참이다. $like(Soyoung, tea)$ 는 불명확이다. $pregnant(Soyoung, girl)$ 는 거짓이다.

3.3 LSMS의 속성

GCWA가 배타적 or 해석을 사용하는 것에 반해 LSMS는 포괄적 or 해석도 사용하므로 LSMS는 GCWA와 다르다. ICWA와 PMS는 포괄적 or 해석을 사용하지만, 이들 구문 해석의 참, 거짓, 불명확 집합의 속성은 LSMS와 다르다. 이 장에서는 LSMS의 특성에 대해서 설명한다.

정의 10 : $F(LSMS)$

데이터베이스 DB와 $HB(DB)$ 의 그라운드 원자식 a 가 있다 하자. LSMS를 이용한 데이터베이스로부터 추론 가능한 그라운드 부정적 리터럴 집합인 $F(LSMS)$ 는 아래와 같다.

$F(LSMS) = \{ \neg a \mid a \in HB(DB), a \text{는 DB의 어느 모델에도 속하지 않는다} \}$

$F(LSMS)$ 는 DB로부터 LSMS를 이용해 추론된 그라운드 부정적 리터럴 집합이다.

DB가 불명확 데이터베이스와 일치하고, a 가 그라운드 원자식이면 다음의 명제가 성립된다.

명제 1 : 일관성. $DB \cup F(LSMS)$ 는 일관성이 있다.

$DB \cup F(LSMS)$ 가 일관성이 없다면, $DB \cup F(LSMS) \vdash a$ 이고 $DB \cup F(LSMS) \not\vdash \neg a$ 이 동시에 성립해야 한다. 만약 $DB \cup F(LSMS) \vdash a$ 이면 a 를 참으로 만드는 모델이 존재한다. 정의 9로부터 $F(LSMS)$ 의 모든 멤버는 DB의 어느 모델에도 속하지 않으므로 $DB \vdash a$ 이다.

만약 $DB \cup F(LSMS) \not\vdash \neg a$ 이라면 a 를 참으로 만드는 모델이 존재하지 않으므로 $a \in F(LSMS)$ 이고 $DB \not\vdash a$ 이다. 만일 $DB \cup F(LSMS)$ 가 일관성이 없다면 DB의 일관성이 없도록 만드는 것은 $DB \vdash a$ 와 $DB \not\vdash a$ 이다. 그러나 DB는 일관성이 있으므로, $DB \cup F(LSMS) \vdash a$ 이면서 $DB \cup F(LSMS) \not\vdash \neg a$ 는 발생할 수 없다. 따라서 $DB \cup F(LSMS)$ 는 일관성이 있다.

$T(GCWA)$, $F(GCWA)$, $U(GCWA)$ 를 GCWA에 의한 참, 거짓, 불명확 집합을 표현한 것이라 하자. 또한 $T(LSMS)$, $F(LSMS)$, $U(LSMS)$ 를 LSMS에 의한 참, 거짓, 불명확 집합을 표현한 것이라 하자.

명제 2 : 약함(weakness). $F(GCWA) \supseteq F(LSMS)$

$F(GCWA) \subset F(LSMS)$ 라 가정하자. 그러면 $a \in F(LSMS)$ 이고 $a \notin F(GCWA)$ 인 원자식 a 가 존재한다. $a \in F(LSMS)$ 이므로, a 는 DB의 모든 모델에서 거짓이다. 만약 a 가 DB의 모든 모델에서 거짓이라면 모든 최소 모델은 모델의 부분집합이므로 a 가 DB의 모든 최소 모델에서 거짓이다. 그러므로 GCWA의 정의에 의해서 $a \notin F(GCWA)$ 는 참이 아니다. 따라서 $a \in F(LSMS)$ 이고 $a \notin F(GCWA)$ 인 원자식 a 가 존재하지 않는다. 그러므로 $F(GCWA) \subseteq F(LSMS)$ 이고 결과적으로 $F(GCWA) \supseteq F(LSMS)$ 이다.

4. LSMS와 다른 의미해석과의 비교

이 장에서는 LSMS와 다른 의미해석들을 비교한다. 비교는 LSMS가 포괄적 or 해석을 지원하며 배타적 or 해석도 지원하고 있음을 보여준다.

4.1 LSMS와 GCWA

$T(LSMS)$ 의 참 집합은 $T(GCWA)$ 의 참 집합과 같다. $T(LSMS)$ 와 $T(GCWA)$ 의 진리값은 두 의미해석의 최소모델의 참 집합 T 를 따르므로, $T(LSMS)$ 는 데이터베이스의 의도하는 바를 포함한다. 다음의 정리는 LSMS 해석이 참 집합을 위한 최소 모델 해석과 같음을 보여준다. 그러므로 두 해석은 동일한 참 집합을 갖는다. 다만 유일한 차이점은 최소 모델 해석이 배타적 or 연산자를 사용하지만 LSMS 해석은 포괄적 or 연산자를 사용한다는 것이다.

정리 1 : 데이터베이스의 원자식이 LSMS 해석에서 참인 것은 GCWA에서 참인 것과 동치관계이다.

정리 1을 증명하려면 $T(LSMS) = T(GCWA)$ 임을 증명하는 것이 필요하다. $T(LSMS) \supset T(GCWA)$ 라고 가정하면, a 는 원자식이고 $a \in T(LSMS)$ 이며 $a \notin T(GCWA)$ 이다. 그러면 원자식 a 는 DB의 모든 모델에서 참이다. 원자식 a 가 DB의 모든 모델에서 참이고 최소 모델이 DB의 모델의 부분집합이므로 $a \in T(LSMS)$ 이며 $a \notin T(GCWA)$ 인 원자식 a 는 존재하지 않는다. 따라서 $T(LSMS) \supset T(GCWA)$ 는 참이 아니다.

이번에는 $T(LSMS) \subset T(GCWA)$ 라고 가정하자. a 는 원자식이고 $a \notin T(LSMS)$ 이며 $a \in T(GCWA)$ 이다. 원자식 a 는 GCWA에서 참이므로 a 는 DB의 모든 최소 모델에 속한다. 또한 $a \notin T(LSMS)$ 은 a 가 DB의 모든 모델에 속하지 않는다는 것을 의미한다. a 가 DB의 모든 모델에 속하지 않으므로 a 가 거짓인 모델이 존재하고, 따라서 a 가 거짓인 최소 모델이 존재해야 한다. 그러나 a 가 DB의 모든 최소 모델에 속하므로 이런 경우는 존재할 수 없고, $T(LSMS) \subset T(GCWA)$ 는 거짓이다. 그러므로 $T(LSMS) = T(GCWA)$ 이다.

LSMS 해석의 진리 집합과 GCWA 해석의 진리 집합이 같으므로 데이터베이스의 원자식이 LSMS 해석에서 참인 것과 최소 모델 해석에서 참인 것은 동치관계이다.

정리 2 : 만약 원자식 a 가 LSMS에서 거짓이면 GCWA에서도 거짓이다.

LSMS의 정의에서 원자식 a 가 LSMS에서 거짓이면 DB의 모든 모델에서도 거짓이다. 당연히 DB의 모든 최소 모델에서도 거짓이다. 결과적으로 GCWA에서는 거짓이다. 예 2는 $MM = \langle \{like(Soyoung, coffee), pregnant(Soyoung, boy)\}; \{like(Soyoung, tea), pregnant(Soyoung, girl)\} \rangle$ 인 최소 모델을 보여준다. GCWA에서 $like(Soyoung, tea)$ 는 LSMS에 속한 불명확이므로 거짓이다. 예제 2는 $F(GCWA) \not\subseteq F(LSMS)$ 임을 보여준다.

정리 3 : 만약 원자식 a 가 GCWA에서 불명확이면 LSMS에서도 불명확이다.

정리 1과 정리 2로부터 $T(LSMS) = T(GCWA)$ 이고 $F(LSMS) \subseteq F(GCWA)$ 이다. 결과적으로 $U(LSMS) \supseteq U(GCWA)$ 이다.

4.2 LSMS와 ICWA

$T(ICWA)$, $F(ICWA)$, $U(ICWA)$ 를 ICWA에 속한 참, 거짓, 불명확 집합들의 표현이라 하자.

정리 4 : LSMS에서 원자식이 참이라는 것은 ICWA에서 참이라는 것과 동치관계이다.

ICWA와 GCWA의 참 집합은 *고유(identical)*하다. 정리 1로부터 ICWA와 LSMS의 참 집합은 같다.

4.3 LSMS와 PMS

$T(PMS)$, $F(PMS)$, $U(PMS)$ 를 PMS에 속한 참, 거짓, 불명확 집합들의 표현이라 하자.

정리 5 : 원자식이 LSMS에서 참이라는 것은 PMS에서 참이라는 것과 동치관계이다.

Sakama의 제안 2.5로부터, PMS로부터 모든 최소 원소의 집합은 데이터베이스의 최소모델과 일치한다. 정의 6에 따라 모든 최소 원소의 집합은 PMS의 참 집합이다. 따라서 PMS의 참 집합은 DB의 최소 모델을 포함한다. 정리 5는 $T(PMS) = T(LSMS)$ 임을 보여준다.

정리 6 : 만약 원소가 LSMS에서 거짓이면 PMS에서도 거짓이다.

모든 가능한 모델은 모델이다. 만약 LSMS의 원자식이 거짓이면 모든 모델의 원자식이 거짓이고, 모든 가능 모델의 원자식이 거짓이다. 그러므로 PMS의 원자식은 거짓이다.

정리 7 : 만약 원자식이 PMS에서 불명확이면 LSMS에서도 불명확이다.

정리 5와 정리 6으로부터 $T(PMS) = T(LSMS)$ 이고 $F(PMS) \supseteq F(LSMS)$ 이다. 따라서 $U(PMS) \subseteq U(LSMS)$ 이다.

5. 결론과 향후 연구과제

LSMS는 *or* 연산자의 포괄적 및 배타적 해석을 모두 표현하기 위하여 연구되었다. Loveland는 불명확 데이터베이스를 하위의 명확 데이터베이스들로 분할할 수 있고 질의에 적합한 답을 할 수 있음을 보여주었다. Sakama의 PMS와 Kong의 ICWA는 불명확 데이터베이스를 하위의 명확한 데이터베이스들로 분할하는 방식에서 서로 유사하다. 그러나 이 방법들은 당연히 하위의 명확 데이터베이스들의 수가 매우 커질 수 있으므로 방법적으로 복잡하다는 단점이 있다.

이러한 문제점을 해결하기 위하여 본 논문은 불명확 데이터베이스를 위한 *or* 연산자의 포괄적 해석과 배타적 해석을 모두 지원하고, 의미해석을 위해 불명확 데이터베이스를 명확 데이터베이스로 만들기 위해 몇 개의 하위 데이터베이스로 분리하지 않아도 되도록 LSMS를 이용하는 방법을 제안하였다. 본 논문은 GCWA와 PMS에서 제안한 거짓 집합과 불명확 집합이 다른 것임을 보였다. 또한 PMS와 LSMS가 *or* 연산자를 분리하지 않고도 포괄적 *or* 해석과 배타적 *or* 해석을 지원하는 것을 증명하였다. 향후 연구과제로는 PMS와 LSMS를 체계적으로 비교하는 연구가 필요하다. 또한 포괄적 *or* 해석과 배타적 *or* 해석을 모두 사용하면서도 불명확 정보를 줄일 수 있는 의미해석 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] K. R. Apt, et al., Towards a Theory of Declarative Knowledge, In J. Minker, ed., "Foundation of Deductive Databases and Logic Programming," pp.193-216, Los Altos, CA : Morgan Kaufman, 1988.
- [2] P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone, "Database Systems - Concepts, Languages and Architectures," McGraw-Hill Book Company 1999.
- [3] F. Buccafurri, et al., "Stable Models and Their Computation for Logic Programming with Inheritance and true Negation," Journal of Logic Programming, 27(1) : 5-44, 1996.
- [4] M. Cadoli, and M. Schaerf, "A survey and Complexity Results for Non-Monotonic Logics," Journal of Logic Programming, 17 : 127-160, 1993.
- [5] S. Chun, et al., Succinct Model Semantics : "A Simple Model for inclusive interpretations. Symposium of Applied Computing," pp.39-44, 1998.
- [6] M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming," In Proceedings, 5th International Conference and Symposium on Logic Programming, pp. 1070-1080, 1988.
- [7] Q. Kong and M. H. Williams, "Evaluating Different Strategies for Handling Incomplete Information in a Logic Database," In T. Dodds, Owens and S. Torrence, eds., Expanding the Horizons Workshop, pp.124-144, Intellect book, Oxford, 1990.
- [8] Q. Kong, et al., "The Indefinite Closed World Assumption," Data & Knowledge Engineering, 12 : 297-311, 1994.
- [9] J. L. Lassez and M. J. Maher, "Optimal Fixedpoints of Logic Programs," Theoretical Computer Science, 39 : 15-25, 1985.
- [10] Jeong-Oog Lee, Doo-Kwon Baik, "Semantic Integration of Databases Using Linguistic Knowledge," Australian Joint Conference on Artificial Intelligence, 72-83, 1999.
- [11] Marco-schaerf, "Negation and Minimality in Disjunctive Databases," Journal of Logic Programming, 23(1) : 63-86, 1995.
- [12] J. Minker, "On Indefinite Databases and the Closed World Assumption," In Proceedings of 6th Conference on Automated Deduction, pp.292-308, Springer-Verlag, 1982.
- [13] T. C. przymusinski, "On the Declarative Semantics of Deductive Databases and Logic Programs," In J. Minker, ed., Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann, Los Altos, California, pp. 193-216, 1988.
- [14] T. C. przymusinski, "Perfect Model Semantics," In Proceedings of the 5th International Conference and Symposium on Logic Programming(Seattle, USA), pp.1081-1096, 1988.
- [15] R. Ramakrishnan and J. D. Ullman, "A Survey of Deductive Database Systems," Journal of Logic Programming, 23(2) : 125-150, 1995.
- [16] R. Reiter, On Closed World Databases, "In Gallaire and Minker," eds., Logic and Databases, pp.55-76, Plenum Press, New York, 1978.
- [17] K. Ross, "Well Founded Semantics for Disjunctive Logic Databases," In Proc. Int. Conf. on Object Oriented and Deductive Databases, pp.352-369, 1989.
- [18] K. Ross, "The Well-founded Semantics for Disjunctive Logic Programs," In W. Kim, J. M. Nicolas, and S. Nishio eds., Deductive and Object-Oriented Databases, pp.385-402, Elsevier Science Publishers, B.V., North Holland, 1990.
- [19] C. Sakama, "Possible Model Semantics for Disjunctive Databases," In Proc. Int. Conf. On Object Oriented and Deductive Databases, pp.369-383, 1989.
- [20] C. Sakama and K. Inoue, "An alternative approach to the Semantics of Disjunctive Logic Programs and Deductive Databases," Journal of Automated Reasoning, 13 : 145-172, 1994.
- [21] C. Sakama, K. Inoue, "Paraconsistent Stable Semantics for Extended Disjunctive Programs," Journal of Logic and Computation 5(3) : pp.265-285, 1995.
- [22] J. S. Schlipf, "Formalizing a Logic for Logic Programming," Annals Mathematics and Artificial Intelligence, 5 : 279-302, 1992.
- [23] J. S. Schlipf, "Complexity and Undecidability results in Logic Programming," Annals Mathematics and Artificial Intelligence, 15 : 257-299, 1995.
- [24] H. Seki and H. Itoh, "A Query Evaluation Method for Stratified Programs under the Extended CWA," Proc. 5th Conf. and Symp. On Logic Programming, (Kowalski and Bowen, eds.), MIT Press, 1988.
- [25] A. Van Gelder, et al., "Unfounded Sets and Well-founded Semantics for General Logic Programs," In Proceedings of 7th ACM Symposium on Principles of Database Systems, pp.221-230, 1988.
- [26] A. Van Gelder, et al., "The Well-Founded Semantics for General Logic Programs," Journal of the ACM, 38(3) : 620-650, 1991.
- [27] A. Yahya, and L. J. Henschen, "Deduction in Non-horn Databases," Journal of Automated Reasoning, 1 : 141-160, 1985.
- [28] J. You and L. Yuan, "On the Equivalence of Semantics for Normal Logic Programs," Journal of Logic Programming, 22(3) : pp.211-222, 1995.

리즘의 데이터 입력으로 사용할 수 있다는 것이다. 즉 시계열 데이터를 잘 가공하여 트랜잭션 데이터화한다면 시계열 데이터에 대한 새로운 알고리즘 개발 없이 많은 정보를 기존의 알고리즘을 사용하여 얻을 수 있다. 본 논문에서는 시계열 데이터로부터 가상 트랜잭션을 생성할 때 시간윈도우 기법과 이벤트윈도우 기법을 이용한다. 이렇게 생성된 가상 트랜잭션을 연관규칙 알고리즘들[1, 2, 14, 17]을 적용하여 유용한 패턴을 찾는다. 그리고 가상 트랜잭션들의 발생시간을 트랜잭션에 속한 이벤트들의 평균발생시간으로 사용함으로써 패턴들의 주기를 발견할 수 있는 주기패턴 알고리즘[4, 7, 8, 12, 13]과 패턴들의 순서를 발견할 수 있는 순차패턴 알고리즘[5, 9-11]을 적용할 수 있다.

2. 관련 연구

시간 기반의 데이터에 대한 연구가 많이 진행되었고 현재도 계속 연구되고 있다. 이런 연구들은 시계열 데이터에 대한 연구와 시간 기반의 트랜잭션 데이터에 대한 연구로 진행되고 있다. 시계열 데이터는 이벤트발생시간과 이벤트 및 기타 정보를 한 개의 레코드로 구성되어 있으며 그 항목들 중 시간과 이벤트가 관심 항목이다. 반면에, 트랜잭션 데이터는 트랜잭션 발생시간과 항목들의 집합으로 구성되어 있고 시간과 항목집합이 관심 필드이다. 시계열 데이터에 관련된 연구와 트랜잭션 데이터에 관련된 연구는 같은 결과를 얻기 위하여 각기 다른 알고리즘을 사용한다. 그 이유는 위에서 언급했듯이 데이터 소스가 다르기 때문이다. 데이터 마이닝에서는 트랜잭션 데이터를 데이터소스로 이용한 많은 알고리즘들이 존재한다.

시계열 데이터에 대한 대표적 연구로서는 통신회사에서 발생하는 시간적인 순서를 가지는 네트워크 이벤트 로그로부터 각 이벤트들 사이의 연관성을 찾는 빈발 에피소드 기법이 있다. 이 알고리즘에서는 시간정보를 담고있는 전체 이벤트집합이 주어질 때, 이벤트 집합 간격 $[t_1, t_2]$ 는 시간 t_1 부터 t_2 까지 발생한 이벤트 집합을 말하며 간격의 크기는 $t_2 - t_1$ 으로 정의된다. 이렇게 정의된 이벤트의 집합을 에피소드라 한다. 이러한 에피소드의 발생회수를 저장하여 알고리즘의 설정 값인 min-fr (최소지지도) 보다 큰 것들만 빈발 에피소드로 결정해낸다. 이렇게 결정된 빈발 에피소드는 자주 함께 발생하는 이벤트의 모음으로 정의하며 이런 빈발 에피소드를 사용하여 이벤트간의 연관 규칙을 찾는다. 기본적인 알고리즘은 첫째, 주어진 시간 이벤트 집합으로부터 두번 이상 발생한 모든 에피소드를 찾는다. 둘째, 주어진 최소지지도를 사용하여 빈발 에피소드를 찾는다[3]. 이렇게 찾아진 빈발에피소드는 이벤트들간의 순차패턴과 일치한다. 이 알고리즘은 현재 침입탐지시스템의 비정상행위 탐지모델에도 적용되고 있다. 이벤트기반의 시계열 데이터에는 주

기를 갖는 패턴도 중요하지만 주기에 어긋나는 패턴이 중요한 관심사항이 될 수 있다. 예를 들어 병원에서 어떤 환자에게 약을 1일 3회 투여하고 있다. 그런데 어느 날 환자의 상태가 악화되어 약을 좀 빨리 투여하였을 경우, 주기패턴 분석에 의하면 하루에 9시, 13시, 19시의 주기를 찾을 수 있다. 하지만 환자의 상태에 따라 빨리 투여 된 약의 경우는 분석이 안 된다. 이런 경우의 이벤트 패턴을 찾는 방법이 제안되었다[4]. 트랜잭션 데이터 관련 연구는 연관규칙을 찾는 기법[1, 2, 12]은 항목 집합으로 구성된 데이터 트랜잭션들로부터 각 항목간의 연관성을 반영하는 규칙을 찾는 기법이다. 지지도와 신뢰도를 설정 값으로 트랜잭션 내에 존재하는 항목 집합들간의 연관규칙을 찾는다. 지지도란 전체 트랜잭션에서 해당 연관규칙이 차지하는 확률이다. 예를 들어 $X, Y \rightarrow Z$ (X 이고 Y 이면 Z 이다)는 X, Y, Z 간의 연관규칙이 있다. 이 연관규칙의 지지도가 10%라면 전체 트랜잭션 중에서 이 규칙을 따르는 트랜잭션이 10%를 차지한다는 것이다. 그리고 신뢰도가 50% 라는 것은 X, Y 를 포함한 트랜잭션 중 50%는 Z 항목을 포함한다는 것이다. 트랜잭션 데이터들 속에서 순차패턴을 찾는 알고리즘[9-11]은 순서정보를 가지고 있는 트랜잭션 데이터 베이스로부터 트랜잭션 내의 데이터에 공통적으로 나타나는 순차적인 패턴을 찾는 기술이다. 고객의 트랜잭션을 담고 있는 대용량 데이터베이스를 입력 데이터로 한다. 여기서 트랜잭션은 고객 ID와 트랜잭션 시간정보를 담고 있어야 한다. 한 개의 트랜잭션에는 한번에 구매하는 아이템들이 담겨져 있다. 예를 들어 비디오 대여점에서 고객들은 '쉬리'를 빌려보는 사람들은 'JSA'를 빌려본다라는 순차패턴을 찾음으로써 '쉬리'를 본 고객들에게 'JSA'를 추천할 수 있다. 어떤 이벤트가 발생했을 때 얼마만큼의 시간이 지난 후에는 어떤 이벤트가 발생할 것인가를 예측하는 방법들이 많이 연구되고 있다. 이러한 예측은 주식시장에서 주가를 예측하는 곳에 사용되고 있다[8]. 대부분의 패턴을 찾는 알고리즘은 짧은 길이의 패턴을 찾는 데 효과적이다. 그러나 실제 데이터베이스에는 긴 길이의 패턴이 많이 포함되어 있다. 긴 패턴을 효과적으로 찾을 수 없는 기존의 Apriori 관련 알고리즘을 개선한 MAX-Miner 알고리즘을 고안했다[12].

3. 가상트랜잭션

본 논문의 궁극적인 목적은 시계열데이터를 트랜잭션화하여 트랜잭션을 데이터 소스로 사용하는 많은 알고리즘에 적용시킬 수 있도록 하는 것이다. 여기서는 가상 트랜잭션 개념을 소개한다. 가상 트랜잭션(Virtual Transaction)은 시계열데이터를 타임 윈도우 기법과 이벤트 윈도우 기법을 사용하여 만든 이벤트의 집합이다. 시계열데이터에는 시간 컬럼과 이벤트 컬럼이 존재한다. 이러한 시계열데이터를 시

석 윤 영

e-mail : sok@suwon-c.ac.kr

1986년 한남대학교 전자계산학과(학사)

1992년 건국대학교 컴퓨터공학과(석사)

2000년 명지대학교 컴퓨터공학과(박사수료)

1986년~1991년 대한통운(주) 전산실 계장

1992년~현재 수원여자대학 컴퓨터응용학부

조교수

관심분야 : 데이터베이스, 정보검색

전 종 훈

e-mail : jchun@mju.ac.kr

1986년 덴버대학교 전산과학(학사)

1988년 노스웨스턴대학교 전산과학(석사)

1992년 노스웨스턴대학교 전산과학(박사)

1992년~1995년 센트럴 오클라호마대학교 조

교수

1995년~현재 명지대학교 컴퓨터학부 부교수

관심분야 : 데이터베이스, 정보검색, 의료정보