객체 지향 질의 처리에서 의미적 재작성 규칙에 관한 연구

이 홍 로[†] 곽 훈 성[††] 류 근 호[†††]

요 약

객체 지향 데이타베이스 시스템은 복잡한 데이타 관리기능에 대한 응용을 제공하는 효과적인 해결책으로써 제안되어왔다. 질의 처리와 같은 문제점에 대한 연구와 이러한 요구를 입증하는 것은 형식적인 객체 지향 질의 모델이 없어서 진척되지 못하고 있다. 본 논문은 집단화 상속성에 기반한 질의 모델을 정의하며, 질의의 대수 표현에서 재작성 규칙을 보존하는 동등성에 적용할 수 있는 의미적 재작성 규칙을 개발한다. 이질의 모델을 의미적으로 분석하여 논리적으로 최적화하고, 질의의 대수식들은 등가 보존 재작성 규칙에 의하여 최적화될 수 있다.

Semantic Rewrite Rules at Object Oriented Query processing

Hong Ro Lee[†], Hoon Sung Kwak[††] and Keun Ho Ryu[†††]

ABSTRACT

Object-oriented database systems have been proposed as an effective solution for providing the data management facilities of complex applications. Proving this claim and the investigation of related issues such as query processing have been hampered by the absence of a formal object-oriented query model. In this paper, we not only define a query model based on aggregation inheritance but also develop semantic rewriting rules which are applied to equivalence preserving rewrite rules in algebraic expression of a query. Analyzing semantically the query model, the query model can be optimized to logically object-oriented query processing. And algebra expresstions of a query can be optimized by applying equivalence preserving rewrite rules.

1. 서 론

객체 지향 데이타베이스 시스템(object-oriented database system)은 CAD, CAM, OIS 및 다매체 시스템 등과 같은 차세대 데이타베이스 기술로서 대두되고 있다. 관계형 데이타 모델과 비교하여, 객체 지향 데이타모델은 일정한 기틀에 의하여 구조적이고 행위적 차원을 통합하여 다루기 때문에 매우 강력한 표현력을 제공한다. 그러나 이러한 객체 지향 기법은 효율성을 향상시키는 질의 처리 최적화에서 아직도 연구가 미비한 실정이며[1,2,3], 재사용성에 대한 관점에서 일반화 상속성(generalization inheritance)을 갖는 복합 객체를 기반으로 연구되었다[4,5]. Codd는 제 1 정규형 관계 데이타 모델의 문제점을 개선하기 위해서 더욱 강력한 기능을 가지는 관계형 모델을 제안하였고[6,7], Roth는 기본 관계 모델의 확장형인 내포형 관계 모델에 기반한 대수식(algebra expression)에 대한 연구를 하였으며[8], Smith등은 집단화(aggregation), 일반화(generalization)와 연관화(association) 등의 직교적 관계성을 가지는 의미적 데이타 모델(semantic data model)에 대한 연구를 하였다[9,10,11]. 복잡한 실세계 데이타의 모델을 위한 의미적 데이타 모델의 직교성을 가지는 복합 객체에 대한 연구는 O_2[12]와 NO_2[1]에서 제안

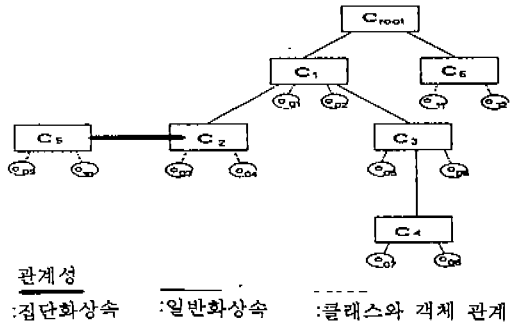
† 정 회 원 : 충북대학교 컴퓨터과학연구소 연구원
†† 종신회원 : 전북대학교 컴퓨터공학과 교수
††† 종신회원 : 충북대학교 컴퓨터과학과 부교수
논문접수 : 1995년 5월 3일, 심사완료 : 1995년 6월 12일

하였고, 합성 객체(composite object)는 Orion [2]에서 연구하였다. 또한 객체 지향 모델에서 재사용성에 대한 연구가 중요하다. 상속성은 속성과 행위를 재사용할 수 있는 기법으로서 일반화 관계성에 의한 클래스 계층 구조에서는 속성과 행위를 전체 다형성(universal polymorphism)에 의하여 상속하여, 집단화 관계성에 의한 클래스 합성 계층 구조에서는 임의의 다형성(adhoc polymorphism)에 의하여 상속한다. 클래스 합성 계층구조에서 노드는 클래스를 나타내고, 에지는 상속 관계를 나타내는 방향성 그래프로 표현할 수 있으며, 데이타베이스의 스키마 구조를 나타낸다. Smalltalk와 같은 초기 시스템은 속성 표현의 상속과 행위의 상속 사이를 구분하지 못하고, 단지 단일 상속성 그래프만 제시하였다[13]. Lalonde와 Zdonik은 상속성의 각 관계에 대하여 연구하였다[14]. 하나 이상의 상위 클래스를 상속할 것인가의 여부는 직교성 문제인데, 단일 상속성은 상속성 그래프가 엄격한 계층 구조에 따른 다중 상속성에 있어서, 클래스들의 방향성 비순환 그래프로 구성된 순수한 객체 지향 모델은 지나치게 유동적이거나 제한적이기 때문에 시스템 설계에 있어서 잘못된 설계 규칙들이 연구되어 왔다[16]. 그리고 모든 개념들이 클래스들로 구성되어야 하는 요구 조건은 지나치게 제한적일 수 있기 때문에, Borgida는 클래스들의 비제한적인 분류가 적절한 예외 실체들에 사용될 수 있는 방법을 연구하였다[17]. 질의는 데이타가 데이타베이스로부터 검색될 수 있도록 하는 기법이며, 질의의 최적화는 논리적으로 동가인 질의를 재작성하는 규칙을 생성하는 논리적 최적화와, 논리적으로 최적화된 질의를 실행하는 물리적 최적화가 있다. Jarke가 관계형 모델에서 이들 문제들과 상호 관련성에 대해서 연구하였다[18]. 논리적 최적화에서 질의 변환에 대한 두 가지 유형이 존재한다: 첫째, 안정성과 같은 특성들을 입증하기에 편리한 질의 표현을 생성하는 것이다. 정의역에 독립적인 술어 형식을 얻기 위해서 해석식을 재작성하는 방법이 제시되었다[19]. 정의역의 독립성은 질의가 참조하는 클래스만을 질의함으로써 정확한 응답을 보장하는 안정성을 추론하는 것이다[20]. 둘째, 변환이 비용 추산에 적합한 질의 표현식을 생성하는 것이다. 이것은 논리적 변환으로써 종종 언급하고 있다. Jarke[18]와 Talbot[21]는 관계형 질의 모델을 위해서 그와 같은 변환을 확인하였다. 그리고 Shaw는 객체 지향 대수 연산자를 위한 제한된 논리적 변환의 수를 확인하였다[5]. 논리적 변환의 특성과 전제 조건을 정확하게 정의할 능력은 규칙 기반 시스템들을 이용하는 응용으로부터 유도할 수 있다. 이 기법은 질의를 개선하고[21,22], 그들을 실행할 수 있는 접근 계획을 변환하기 위한 것이다[23]. 또한 데이타베이스 스키마의 여러 무결성 제약조건에서 클래스 계층 구조에 기반한 질의의 객체 대수식의 의미적 재작성 규칙에 대한 연구가 있어 왔으나[3], 클래스 합성 계층 구조를 분석하여 집단화 상속성을 추가한 의미적 재작성 규칙에 대한 연구는 미비한 실정이다. 그래서 본 연구는 일반화 상속성에 의한 포함 상속과 매개변수 상속성을 지원할 수 있게하고, 집단화 관계성과 연관화 관계성에 의해서 각각 합성 객체와 연관 객체로 구분하여 집단화 상속을 지원하도록 제시한다. 집단화 관계성은 속성 함수가 참조하는 클래스와 성분 관계를 유지하고 클래스 수준과 객체 수준에서는 종속 관계성을 맺고 있다. 연관화 관계성은 참조 관계성에 있어서 속성 함수와 참조되는 클래스가 독립적으로 객체 수준에서만 의미적 연관성을 유지한다. 또한 본 논문은 질의를 절차적으로 처리를 하기 위해서 질의식에서 등가 보존 변환을 이용하여 의미적 재작성 규칙을 규정하고, 규칙 기반 변환 시스템에 적합한 조건들을 가지는 질의식을 제시한다. 본 논문을 효율적으로 지원하기 위하여, 제 2 장에서는 객체 지향 질의 모델에 대해 기술한다. 그리고 제 3 장에서는 의미적 재작성 규칙의 외연성과 그 확장을 제안하고, 의미적 재작성 규칙의 이항 연산자들의 의미적 변환에 관하여 설명한다. 마지막으로, 제4장에서는 본 연구를 마무리하기 위해서 결론을 맺으며, 향후 연구에 대해 언급한다.

2. 객체 지향 질의 모델

본 논문에서 제안한 질의 모델은 객체 식별성과 메소드, 추상 데이타 타입, 캡슐화, 타입과 클

래스, 클래스 계층 구조와 클래스 상속 등과 같은 공통적으로 수용하는 객체 지향 개념을 지원한다[12, 2]. 추가로 한 클래스의 속성이 다른 클래스를 참조하는 속성-영역(attribute-domain)관계를 집단화 참조와 연관화 참조로 구분하여 사용하고, 집단화 상속 개념을 지반나화 참조 클래스 계층구조에 도입하며, 또한 전체 다형성과 임의 다형성을 지원하기 위해서 집단화 상속성과 일반화 상속성을 조합한다. 객체 지향 모델은 식별자의 유무에 의해서 값과 객체를 구분하며, 값은 원자 값과 구조화 값으로 구성된다. 객체는 원자 객체와 튜플과 집합 구성자의 직교성에 의하여 재귀적으로 이루어진다[4]. 클래스는 인터페이스와 메소드들과 그 요약들의 집합에 의해서 인스턴스들의 의미와 인터페이스를 정의한다. 클래스 계층 구조(class hierarchy)에서 부분 클래스는 "Is-A" 일반화 관계성(generalization relationship)을 가진다. 클래스 합성 계층 구조(class composition hierarchy)에서 속성과 영역에 대한 관계성은 연관화 관계성과 집단화 관계성으로 구분된다. 연관화 참조는 속성-영역 관계의 독립적인 두 개의 클래스가 "Has-A" 연관화 관계성(association relationship)을 가지며, 이 연관화 관계성은 반사적, 대칭적 및 추이적 특징의 관계를 가지는 순환 경로식을 운행한다. 집단화 참조는 속성-영역 관계의 종속적인 두 개의 클래스가 "Is-Part-Of" 집단화 관계성(aggregation relationship)을 가지며, 비대칭적이고 추이적 특징의 관계를 가지는 비순환 경로식을 운행한다. 또한 각 클래스는 일반화 관계성과 집단화 관계성에 의해서 상위 클래스들로부터 속성 및 메소드들을 상속할 수 있다. 그래서 클래스 합성 계층 구조는 상속성 반격자(semi-lattice) 구조를 가지며, 이 반격자 구조는 방향성 그래프 G = (V,E)로 표현할 수 있다. 여기서 V에 속하는 정점(vertex)들은 클래스들을 나타내고, 만일 vj가 vi의 상위 정점이라면, 에지(vi, vj)는 E에 존재한다. 만일 vi로부터 vj까지 경로(path)가 vi는 vj의 부분 클래스이다. 집단화와 일반화는 새로운 클래스를 추가하는 재사용에 있어서 다른 개념이다. 다형성 입장에서, 부분 클래스 상속성은 다형성 특징이 모든 환경에서 상속하고 강제성이 필요치 않은 전체 다형성을 의미하지만, 집단화 상속성은 다형성 특징이 클래스의 수준에서만 존재하고 강제성이 요구되는 임의 다형성을 의미한다[24]. 또한 일반화는 기존의 개념을 세분하기 위해 사용하며, 집단화는 더욱 강력한 개념의 복잡성을 증가시키기 위해 사용한다. 한 클래스 c에 속하는 객체들의 집합을 그 클래스의 외연(extent)이라 하고, 이것을 ext(c)로 표기한다. 루트 클래스가 존재하는 클래스 반격자 구조에서 클래스의 외연은 해당 클래스에 포함되는 객체들의 범위를 설정하고, 객체들을 참조하는 의미적 기능(semantic facility)을 부여한다. 또한 한 클래스 c에서, c의 깊은 외연(deep extent)은 ext*(c)로 표기한다. 이에 대한 의미는 클래스 c에 루트 클래스를 둔 상속성 반격자 구조에서 일반화 관계성을 가지는 부분 클래스들과 집단화 관계성을 가지는 부분 클래스들에 속하는 객체들을 조합한 모임이 해당 클래스 c의 깊은 외연이 된다. 세부적으로 분석하면, 일반화 상속성에 의한 부분 클래스들의 모임이 $C_s=(c_{s1}, c_{s2},..., c_{sm})$이라 하고, 집단화 상속성에 의한 속성-영역 관계에 있는 클래스들의 모임이 $C_{ad}=(c_{ad1}, c_{ad2},..., c_{adn})$이라 하자. 클래스 C에 대해서 상속 관계가 있는 전체 클래스의 집합은 일반화 상속 관계의 클래스들과 집단화 상속 관계의 클래스들을 합한 것이기 때문에, $C=C_s \cup C_{ad}=(c_{s1}, c_{s2},...; c_{sm}, c_{ad1}, c_{ad2},..., c_{adn})$이다. 그래서



(그림 1) 상속성에 의한 반격자 구조에서 클래스의 외연성

(Fig. 1) An Extent of a Class at Semi-lattice by Inheritance

클래스 C에 대한 깊은 외연 ext*(C)은 일반화 상속성에 속하는 클래스들의 외연 ext*(C_s)과 집단화 상속성에 속하는 외연 ext*(C_{ad})의 합이다. 즉, ext*(C)=ext*(C_s)∪ext*(C_{ad})이다.

(그림 1)에 상속성 반격자 구조에 대한 의미적 관계성이 보여져 있다. 정점들은 클래스나 객체를 나타내며, 세부적으로 직사각형은 클래스를 의미하고 원은 객체를 지칭한다. 또한 에지들은 클래스 및 객체 사이의 관계성을 표현하기 위한 것으로서, 굵은 실선은 집단화 관계성을 나타내고, 가는 실선은 일반화 관계성을 나타내며, 점선은 해당 클래스와 객체들의 외연 관계를 의미한다. 예에 대한 예로써, 클래스 c_1에 대한 외연 ext(c_1)는 객체들 {o_1, o_2}이며, 클래스 c_1에 대한 깊은 외연 ext*(c_1)는 객체들 {o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}}이다.

복합 객체를 위한 데이타베이스 스키마의 구조가 C=(A_1...A_n)이라면, C는 클래스이고 A_i(단, $1 \le i \le n$)는 속성 이름이다. 속성-영역 관계성에서 속성 A_i가 원자 값에 대한 영역을 참조하는 것이라면 클래스 C에 대한 영차 속성 함수(Zero order function)는 Zfn(C)이고, 속성 Ai가 집합 구조와 튜플 구조의 값에 대한 영역을 참조하거나 원자 값으로된 식별자들의 영역을 참조하는 것이라면 클래스 C에 대한 고차 속성 함수(Higher order function)는 Hfn(C)이다. 속성-영역 관계성에서 속성들의 전체 집합을 클래스 C에 대한 좌변 속성 함수(Left hand side attribute function)는 Lfn(C)이라 한다. 그래서 좌변 속성 함수의 전체 집합은 영차 및 고차 속성 함수를 합한 전체 집합으로써 Lfn(C) = Zfn(C) + Hfn(C)이다. 또한 클래스 합성 계층 구조(C)에서 일반화 관계성과 속성-영역 관계성에 의한 전체 함수 공간(Function space)을 Fspace(C)이라 한다.

집단화 경로식은 집단화 상속성에 의해서 중간 속성에 무관하게 Path(A[C]) = C.A로 간략화 시켜 사용하며 비순환 질의 그래프를 운행한다. 연관화 경로식은 C.A = Ci.*self*로 규정하여 사용하며 순환 질의 그래프를 운행한다[25, 26].

일반화 상속성, 집단화 상속성과 연관화 참조를 가지는 클래스 합성 계층 구조의 스키마는 (그림 2)에 보여져 있다. 이 그림을 이용하여 일반화 관계성, 집단화 상속성과 연관화 참조에 의한 관계성을 SQL 유형의 구문[26]으로 작성하여 고찰하여 보자.

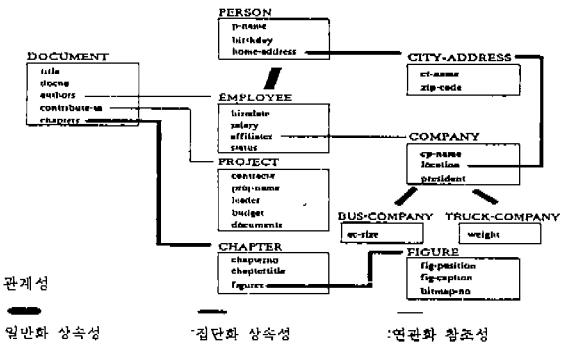
【예제 2.1】 "피고용인의 거주지가 청주이고 봉급이 100만원 이상인 피고인의 생일을 찾아라."

```
SELECT    e.birthday
FROM      EMPLOYEE*: e
WHERE     e.ct-name = "청주"
          AND e.salary > 1000000
```

예제 2.1의 범위절에 해당하는 클래스 EMPLOYEE는 상위 클래스 PERSON으로부터 속성들을 일반화 상속을 하기 때문에 클래스 계층 구조를 객체 변수의 영역으로 선언한 것이다. 외연 ext(PERSON)은 클래스 PERSON에 의해서 정의된 영역에 있는 객체들이고, 깊은 외연 ext*(PERSON)은 클래스 PERSON에 의해서 정의된 영역에 있는 객체들의 집합인, 일반화 상속성을 가지는 하위 클래스 EMPLOYEEON에 의해서 정의된 영역에 속하는 객체들, 그리고 집단화 상속성을 가지는 속성-영역 관계의 클래스 CITY-ADDRESS에 의해서 정의된 영역에 속하는 객체들을 합한 것이다.

【예제 2.2】 "무량수전" 이라는 그림을 표제로 하는 문서를 찾아라.

```
SELECT    d
FROM      DOCUMENT: d
WHERE     d.figcaption = "무량수전"
```



(그림 2) 클래스 합성 계층 구조

(Fig. 2) A Class Composition Hierarchy

예제 2.2에서 범위절에 해당하는 클래스 DOCUMENT에 의한 집단화 상속 관계는 다음과 같다. 첫째, 속성 함수 chapters가 종속적으로 클래스 CHAPTER를 참조하고, 속성 함수 figures는 종속적으로 클래스 FIGURE를 참조한다. 그래서 속성-영역의 참조는 집단화 관계를 가지기 때문에 집단화 상속 관계가 성립한다. 외연 ext(DOCUMENT)는 클래스 DOCUMENT에 의해서 정의된 영역에 있는 객체들이고, 깊은 외연 ext*(PERSON)은 클래스 DOCUMENT에 의해서 정의된 영역에 있는 객체들의 집합 ext(DOCUMENT), 집단화 상속성을 가지는 속성-영역 관계의 클래스 CHAPTER와 클래스 FIGURE 각각에 의해서 정의된 영역에 속하는 객체들 ext(CHAPTER)와 ext(FIGURE)를 합한 것이다.

【예제 2.3】 "한 회사의 사장이면서 피고용인을 찾아라."

```
SELECT    e
FROM      EMPLOYEE: e, COMPANY: c
WHERE     e.self = c.president
```

예제 2.3은 일반화 상속성 관계에 의한 속성 함수만을 이용하고 있다. 술어절에서 클래스 EMPLOYEE와 클래스 COMPANY는 독립적 연관 관계를 가지는 것이기 때문에 상속성을 가지지 않는다. 그래서 "*self*"라는 자기 자신의 클래스에 대한 식별자를 지칭하는 키워드를 사용하여 두 클래스를 연관시켰다.

본 논문에서 제시한 상속성의 주요한 장점은 일반화 상속성과 집단화 상속성을 조합하여 스키마를 규정함으로써 속성 함수 및 메소드의 재사용성을 증가시킨다. 또한 집단화 상속성에 의하여 경로식을 간략하게 표기할 수 있고, 복합 객체의 스키마 변경(schema evolution)에 이상(anomaly)을 발생시키지 않는다.

3. 의미적 재작성 규칙

의미적 재작성 규칙은 구문적 재작성 규칙과 유사하나 클래스의 "Is-A"와 "A-Part-Of" 상속 반격자 구조의 데이타베이스 스키마의 의미에 의해 생성된다. 세부적으로 질의의 최적화는 두 단계로 나누어진다. 첫째, 상위 단계에서는 논리적으로 등가인 질의들의 재작성 규칙을 생성한다. 둘째, 하위 단계에서는 상위 단계에서 생성된 질의의 접근 계획을 물리적으로 대응하는 접근 비용을 최소화하면서, 비용 모델, 접근 경로의 지식과 데이타베이스 통계에 기반한 질의를 최소 비용의 접근 계획을 실행한다[3]. 논리적 최적화는 대수식을 구문적으로 등가인 규칙들을 재작성하는 것과 클래스의 반격자 구조에 대한 데이타베이스 스키마의 의미적으로 등가인 재작성 규칙을 생성하는 것이다. 본 절에서는 논리적 질의 최적화에서 의미적 재작성 규칙에 중점을 두겠다.

3.1 깊은 외연성

본 논문은 클래스 상속성 반격자 구조에 의한 의미적 정보를 이용하는 객체 대수식에 대한 변환 규칙을 제시한다. 각 데이타베이스는 클래스들에 따른 객체들의 분포를 가지는 그 자신의 클래스 격자 구조를 가지기 때문에 의미적 재작성 규칙은 데이타베이스의 상태에 종속한다. 내포성은 데이타베이스의 클래스와 속성 함수에 대한 통계적 정보를 최적으로 접근하기 위한 기법이다. 그래서, 이 절은 클래스 계층 및 클래스 합성 계층 구조의 의미적 정보를 이용하여 클래스 외연성에 관한 재작성 규칙 및 클래스 일치성에 종속하는 재작성 규칙을 제시하고자 한다. 앞으로, ci는 하나의 클래스를 나타내며, Ci와 Ci*는 각각 ci의 외연과 깊은 외연을 나타낸다.

【정리 3.1】 깊은 외연성 확장

객체 대수식에서 한 클래스 C에 대한 깊은 외연 ext*(C)는 클래스 C의 외연 ext(C), 일반화 상속에 의한 클래스 C의 외연 ext*(c_s), 그리고 집단화 상속에 의한 클래스 C의 외연 ext(c_{ad})의 조합이다. 즉, 클래스 C에 대한 깊은 외연은 ext*(C) = ext(C) ∪ ext*(c_s) ∪ ext(c_{ad})이다. 여기서 일반화 상속에 의한 부분 클래스들은 c_s = {c_{s1}, ⋯ ,c_{nm}}이며, 집단화 상속에 의한 속성-영역 관계에 의한 클래스들은 c_{ad} = {c_{ad1}, ...,$c_{ad\,n}$}이다.

증명 : 객체 대수 연산자에서 피연산자는 객체들의 집합이다. 질의 트리의 단말 노드들에서 이들의 객체 집합들은 클래스 외연이나 클래스 깊은 외연이다. 이 데이타 모델은 클래스의 깊은 외연 ext*(C)이 C에서 루트 노드인 클래스 반격자의 부분 트리에 있는 모든 클래스의 외연들의 합을 나타낸다. 따라서, 아래 식은 의미적으로 등가이다.

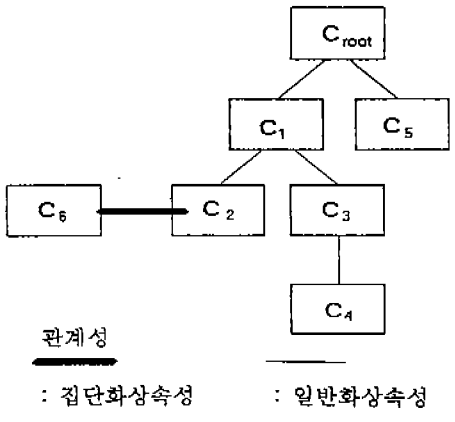
$$\begin{aligned} ext^*(C) &\equiv ext(C) \cup ext^*(C_s) \cup ext(C_{ad}) \\ &\equiv ext(C) \cup ext(c_{s1}) \cup \ldots \cup ext(c_{sn}) \\ &\quad \cup ext(c_{ad1}) \cup \ldots \cup ext(c_{adn}) \end{aligned}$$

【예제 3.1】 (그림 3)과 같은 클래스 반격자 구조가 주어진다면, 다음 식이 성립한다.

$$c_1{}^*\sigma_F< > \Leftrightarrow (C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_6)\sigma_F< >$$

이 예는 깊은 외연에 있는 클래스들의 완전한 신장을 구현한 것이며, $\sigma_F< >$는 술어 조건 F를 만족하는 선택 대수 연산자이다. 물론, 깊은 외연을 구별하는 합의 부분 집합은 또한 깊은 외연에 의해 대치될 수 있다. 따라서, 다음 식은 등가식이다.

$$C_1{}^*\sigma_F< > \Leftrightarrow (C_1 \cup C_2 \cup C_3{}^* \cup C_6)\sigma_F< >$$



(그림 3) 간단한 클래스 반-격자
(Fig. 3) A Simple Class Semi-lattice

3.2 이항 연산자들의 의미적 변환

클래스 반-격자에 대한 지식은 이항 연산의 특별한 경우들을 구별하고 감소시키는데 이용한다. 일관성 있는 방법으로 이들의 특별한 경우들을 거론하기 위해서, 클래스 상속 그래프에 대한 정보를 조작하는 이항 관계는 다음처럼 정의할 수 있다.

【정의 3.1】 깊은 외연 중첩성: 표기 $c_1 \nabla c_2$는 $\{\exists c_i \mid c_i \le c_1 \wedge c_i \le c_2\}$를 나타낸다. 즉, $C_1{}^*$과 $C_2{}^*$는 공통적인 객체들을 가진다. 관계 ∇는 일반화 상속성에 의해서 반사적, 대칭적 및 비추이적 특성을 가지며, 집단화 상속성에 의해서 비반사적, 비대칭적 및 추이적 특성을 가진다.

【정리 3.2】 이항 연산자의 특별한 경우는 다음처럼 구분된다.

1. $C_1 \cap C_2 = \emptyset$ when $c_1 \neq c_2$
증명 : 이 데이타 모델은 객체가 단지 한 클래스의 일원임을 나타낸다. 즉, 임의의 두 개의 다른 클래스가 공유하는 객체를 가지지 않는다.

2. $C_1 - C_2 = C_1$ when $c_1 \neq c_2$
증명 : 이 증명은 위의 경우 1과 같다.

3. $C_1 \cap C_s{}^* \cap C_{ad}{}^* = C_1$ when $c_1 \le c_s \wedge c_1 \le c_{ad}$
증명 : $\le$와 ∇ 연산자의 정의에 의해서 c_s는 일반화 상속성에 속하는 클래스이고, c_{ad}는 집단화 상속성에 속하는 클래스라는 조건하에 $c_1 \le c_s \wedge c_1 \le c_{ad}$이면, C_1에 있는 모든 객체들은 $C_s{}^*$와 $C_{ad}{}^*$에도 있다. c_1의 어느 객체들도 다른 클래스의 객체들이 될 수 없으므로 이 재작성 규칙은 성립한다.

4. $(C_1 \cap C_s{}^*) \cup (C_1 \cap C_{ad}{}^*) = \emptyset$ when $c_1 \not\nabla c_s \wedge c_1 \not\nabla c_{ad}$
증명 : 정의 3.1에 의해서, $C_1{}^*$(C_1을 포함)과 일반화 상속에 의한 클래스 $C_s{}^*$는 $c_1 \not\nabla c_s$이고 집단화 상속에 의한 클래스 $C_{ad}{}^*$는 $c_1 \not\nabla c_{ad}$일 때 공유하는 객체들을 가지지 않는다.

5. $C_1 \cup C_s{}^* = C_s{}^*$ when $c_1 \le c_s$
증명 : 정의 4.2에 의해서, $c_s{}^*$는 c_s의 일반화 상속에 의한 부분 클래스인 각 클래스의 외연에 있는 모든 객체들을 포함한다.

6. $C_1 \cup C_{ad}{}^* = C_{ad}{}^*$ when $c_1 \le c_{ad}$

증명 : 정의 4.2에 의해서, c_{ad}^*는 c_{ad}의 집단화 상속에 의한 부분 클래스인 각 클래스의 외연에 있는 모든 객체들을 포함한다.

7. $C_1 - C_s^* = C_1$ when $c_1 \neq c_s \wedge c_1 \not\leq c_s$
증명 : 조건 $c_1 \neq c_s$는 c_1과 c_s가 공유하는 객체를 가지지 않음을 의미한다. 즉, $C_1 \cap C_s^* = \emptyset$는 일반화 상속에 의한 부분 클래스가 공유하는 객체들을 가지지 않는다. 따라서, C_1과 C_s^*는 공유하는 객체를 가지지 않는다.

8. $C_1 - C_{ad}^* = C_1$ when $c_1 \neq c_{ad} \wedge c_1 \not\leq c_{ad}$
증명 : 조건 $c_1 \neq c_{ad}$는 c_1과 c_{ad}가 공유하는 객체를 가지지 않음을 의미한다. 즉, $c_1 \cap c_{ad}^* = \emptyset$는 집단화 상속에 의한 부분 클래스가 공유하는 객체들을 가지지 않는다. 따라서, c_1과 c_{ad}^*는 공유하는 객체를 가지지 않는다.

9. $C_s^* - C_2 = C_s^*$ when $c_s \not\nabla c_2$
증명 : 일반화 상속에 속하는 클래스에 의해서, $c_s \not\nabla c_2$는 $C_s^* \cap C_2^* = \emptyset$임을 의미한다. 또한, $C_2 \subseteq C_2^*$, $C_s^* \cap C_2 = \emptyset$임을 의미한다.

10. $C_{ad}^* - C_2 = C_{ad}^*$ when $c_{ad} \not\nabla c_2$
증명 : 집단화 상속에 속하는 클래스에 의해서, $c_{ad} \not\nabla c_2$는 $C_{ad}^* \cap C_2^* = \emptyset$임을 의미한다. 또한, $C_2 \subseteq C_2^*$, $C_{ad}^* \cap C_2 = \emptyset$임을 의미한다.

11. $C_1^* \cap C_2^* = \emptyset$ when $c_1 \not\nabla c_2$
증명 : 정의 3.1에 의해서 $c_1 \not\nabla c_2$는 c와 c_2가 같은 루트 클래스에 있지 않고 공유하는 객체들을 가지지 않는다.

12. $C_1^* \cap C_2^* = \cup C_i$, $c_i \leq c_1 \wedge c_i \leq c_2$ when $c_1 \nabla c_2$
증명 : $c_{11}, \cdots, c_{1n}$은 c_1에서 루트를 둔 클래스 상속 그래프의 부분 트리에 있는 모든 클래스들이라 하자.
정리 3.1을 적용하여, $C_1^* \cap C_2^*$는 다음처럼 재작성 된다.

$$(C_{11} \cup \cdots \cup C_{1n}) \cap C_2^*$$

집합의 분배 법칙에 의해서, 위식은 다음과 같이 된다.

$$(C_{11} \cap C_2^*) \cup \cdots \cup (C_{1n} \cap C_2^*)$$

위의 3의 경우에, 각 항$(C_{1i} \cap C_2^*) = C_{1i}$ when $c_{1i} \leq c_2$.

13. $C_s^* \cap C_{ad}^* = C$, $c_s \leq C \wedge c_{ad} \leq C$ when $c_s \nabla c_{ad}$
증명 : 일반화 상속에 의한 부분 클래스 c_s와 집단화 상속에 의한 부분 클래스 c_{ad}가 $c_s \leq C \wedge c_{ad} \leq C$ 조건하에 $c_s \nabla c_{ad}$이 성립하면, c_s와 c_{ad}는 같은 루트 클래스 C에서 공통 클래스를 가진다.

14. $C_s^* \cap C_{ad}^* = \emptyset$ when $c_s \not\nabla c_{ad}$
증명 : 일반화 상속에 의한 부분 클래스 c_s와 집단화 상속에 의한 부분 클래스 c_{ad}가 $c_s \not\nabla c_{ad}$ 관계에 있으면, c_s와 c_{ad}는 다른 루트 클래스를 가진다.

15. $C_{s1}^* \cup C_{s2}^* = C_{s2}^*$ when $c_{s1} \leq c_{s2}$
증명 : 정의 3.1에 의해서 c_{s1}과 c_{s2}가 일반화 상속성에 의한 부분 클래스들이라면, $c_{s1} \leq c_{s2} \Rightarrow C_{s1}^* \subseteq C_{s2}^*$

16. $C_{ad1}^* \cup C_{ad2}^* = C_{ad2}^*$ when $c_{ad1} \leq c_{ad2}$
증명 : 정의 3.1에 의해서 c_{ad1}과 c_{ad2}가 집단화 상속성에 의한 부분 클래스들이라면, $c_{ad1} \leq c_{ad2} \Rightarrow C_{ad1}^* \subseteq C_{ad2}^*$

17. $C_s^* \cup C_{ad}^* = C^*$ $c_s \leq C \wedge c_{ad} \leq C$ when $c_s \nabla c_{ad}$
증명 : 일반화 상속에 의한 부분 클래스 c_s와 집단화 상속에 의한 부분 클래스 c_{ad}가 $c_s \leq C \wedge c_{ad} \leq C$ 조건하에 $c_s \nabla c_{ad}$이 성립하면, c_s와 c_{ad}는 같은 루트 클래스 C의 깊은 외연 C^*를 가진다.

18. $C_s^* \cup C_{ad}^* = \{C_s^*, C_{ad}^*\}$ when $c_s \not\nabla c_{ad}$
증명 : 일반화 상속에 의한 부분 클래스 c_s와 집단화 상속에 의한 부분 클래스 c_{ad}가 $c_s \not\nabla c_{ad}$ 관계에 있으면, c_s와 c_{ad}는 다른 루트 클래스를 가진다.

19. $C_s^* - C_{ad}^* = C_s^*$ when $c_s \not\nabla c_{ad}$
증명 : 정의 3.1에 의해서 c_s와 c_{ad}가 각각 일반화 상속성과 집단화 상속성에 의한 부분 클래스들이라면, $c_s \not\nabla c_{ad} \Rightarrow C_s^* \cap C_{ad}^* = \emptyset$이다. 단, C_s^*와 C_{ad}^*가 같은 루트 클래스일 필요는 없다.

20. $C_{ad}^* - C_s^* = C_{ad}^*$ when $c_s \not\nabla c_{ad}$
증명 : 정의 3.1에 의해서 c_s와 c_{ad}가 각각 일반화 상속성과 집단화 상속성에 의한 부분 클래스들이

라면, $c_s \not\subset c_{sd} \Rightarrow C_s' \cap C_{sd}' = \emptyset$이다. 단, C_s'와 C_{sd}'가 같은 루트 클래스일 필요는 없다.

이항 연산자들의 의미적 변환 규칙에 의한 적용 예를 다음과 같다.

【예제 3.2】 "청주에 살고 있으면서 피고용인을 제외한 사람을 찾아라"

```
SELECT  pe
FROM    PERSON' DIFFERENCE EMPLOYEE: pe
WHERE   pe.ct-name="청주" AND e.salary>1000000
```

예제 3.2는 일반화 상속성을 클래스 PERSON의 깊은 외연으로부터 클래스 EMPLOYEE의 외연을 제외한 범위를 객체변수로 선언한 것이다. 이에 대한 의미적 변환 규칙은 정리 3.2의 규칙 2와 규칙 9의 예이다.

【예제 3.3】 "청주에 지점을 두고 있는 버스 회사나 트럭 회사를 찾아라."

```
SELECT  at
FROM    BUSCOMPANY UNION TRUCKCOMPANY: at
WHERE   at.city = "청주"
```

예제 3.3은 범위절에서 일반화 상속성에 의하여 클래스 COMPANY로부터 속성함수들을 상속하는 클래스 BUSCOMPANY의 외연과 클래스 TRUCKCOMPANY의 외연의 합을 객체변수 at로 선언한 것이다. 또한 한정절에서 집단화 상속성에 의하여 루트 클래스 COMPANY의 속성 location이 종속적으로 영역인 클래스 CITY-ADDRESS를 참조하기 때문에 집단화 상속한다.

그래서 의미적으로 루트 클래스 COMPANY의 깊은 외연은 클래스 BUSCOMPANY의 외연, 클래스 TRUCKCOMPANY의 외연과 클래스 CITY-ADDRESS의 외연의 합이 전체 외연이 된다. 이에 대한 의미적 변환 규칙은 정리 3.2의 규칙 5, 규칙 15 그리고 규칙 17을 적용한 것이다.

정리 3.2의 의미적 변환 규칙들 중에서 아직 적용하지 않은 교집합 연산 및 그 외의 연산을 일반화 상속성과 집단화 상속성이 집단화 상속성이 추가된 질의들도 예제 3.2와 예제 3.3과 같은 방법으로 입증할 수 있다.

지금까지 본 논문은 객체 지향 모델에서 의미적 재작성 규칙들을 제시하였다. 더욱 복잡한 재작성 규칙들이 발견될 수 있다 할지라도, 제시한 재작성 규칙들의 합성에 의해서 유도하여 표현할 수 있다.

패턴 매칭에 의한 구문적 재작성 규칙과 같지 않게 의미적 재작성 규칙은 클래스, 부분 클래스 상속성, 집단화 클래스 상속성, 부분 클래스 계층 구조 및 무결성 제한 조건 등과 같은 객체 지향 모델의 지식에 의존한다. 이 객체 지향 모델의 지식은 효율적으로 객체 대수식에 의한 구문적 재작성 규칙을 개선하는데 도움을 준다.

의미적 재작성 규칙에 의하여 한 클래스의 깊은 외연은 부분 상속성을 갖는 각 클래스들의 합과 의미적으로 등가이다. 이항 연산자들에 대한 의미적 변환 규칙은 클래스의 상속성을 만족하는 외연성에 의해서 집합 연산자들의 의미적 등가성을 가진다.

그래서 Straube[3]가 제안한 일반화 상속성을 가지는 의미적 재작성 규칙은 간단하나, 본 논문에서 제안한 재작성 규칙은 일반화 상속성에 집단화 상속성을 새롭게 추가하여 의미적 재작성 규칙을 확장하였다.

결국, 의미적 재작성 규칙은 데이타베이스에 존재하는 객체나 값을 저장하고 접근하기 위한 접근 계획을 생성할 때, 질의 연산에 있어서 집합 연산자의 사용 범위에 대한 규칙을 정하고 값싼 비용 추산을 할 수 있는 규칙들을 구축할 수 있다.

5. 결 론

본 논문은 객체 지향 데이타베이스의 질의 처리를 위해서 질의 모델을 기반으로한 의미적 재작성 규칙을 제안하였다. 이 제안된 재작성 기법은 객체 지향 질의 모델에 관한 다양한 질의의 효율적 접근 계획을 생성하기 위한 방법으로써 본 논문은 집단화 상속성을 갖는 객체 지향 질의 모델을 기반으로 논리적 최적화를 수행할 수 있음을 정리, 증명과 예로써 보였다.

이 의미적 재작성 규칙에 의하여 한 클래스의 깊은 외연은 부분 상속성을 갖는 각 클래스들의

합과 의미적으로 등가이다. 이항 연산자들에 대한 의미적 변환 규칙은 클래스의 상속성을 만족하는 외연성에 의해서 집합 연산자들의 의미적 등가성을 가진다. 따라서 본 논문에서 제시한 질의 모델은 질의 처리에 있어서 암시적으로 참조되는 객체들의 영역만을 질의하기 때문에 질의의 연산 비용을 줄일 수 있고, 의미적 재작성 규칙은 데이타베이스에 존재하는 객체나 값을 저장하고 접근하기 위한 접근 계획을 생성할 때, 연산자들의 사용 범위에 대한 규칙을 정하고 값싼 비용 추산을 할 수 있는 규칙들을 구축할 수 있다.

향후 연구 방향으로는 제시한 상속성에 의한 의미적 재작성 규칙에 구문적 재작성 규칙을 추가하여 대수식을 표현하고, 효율적 질의 처리를 위한 접근 계획 생성 알고리즘에 제안한 재작성 규칙을 적용하고자 한다.

참 고 문 헌

[1] Demuth, B., Geppert, A., and Gorchs, T., "Algebraic Query Optimization in the CoOMS Structurally Object-Oriented Database System," Query Processing for Advanced Database Systems, ed. Freytag, J. C., et al., MORGAN KAUFMANN PUBLISHERS, San Mateo, California, pp. 121-142, 1994.

[2] Kim,W., "A Model of Queries for Object Oriented Database," in Proc. VLDB, pp. 423-432, 1989.

[3] Straube,D., "Queries and Query Processing in Object-Oriented Database Systems," PhD thesis, the Univ. of Albert at Edmonton in Canada 1991.

[4] Bancilhon, F. and Khoshafian, S., "A Calculus for Complex Objects," In Proc. PODS, pp. 53-59, 1986.

[5] Shaw,G. and Zdonik.S.,"An Object-Oriented Query Algebra," in Proc. DBPL, Salisham Lodge, Oregon, 1989.

[6] Codd,E. F., "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol.13, No.6, pp. 377-387, Jun. 1970.

[7] Codd, E.F.,"Extending the Database Relational Model to Capture More Meaning," ACM TODS, Vol.4, No.4, pp. 397-434, Dec. 1979.

[8] Roth, M. A., Korth, H. F., and Silberschatz, A., "Extended Algebra and Calculus for Non-1NF Relational databases," ACM TODS, Vol.13, No.4, pp. 389-417, 1988.

[9] Abiteboul, S. and Hull, R., "IFO: A Formal Semantic Database Model," in Proc. ACM PODS, pp. 119-132, 1984.

[10] Chen,P. P., "The Entity-Relationship Model-Toward a Unified View of Data," ACM TODS, Vol. 1, No. 1, pp. 9-36, May 1976.

[11] Smith, J.M. and Smith, D.C., "Database Abstractions: Aggregation and Generalization," ACM TODS, Vol. 2, No. 2, pp. 105-133, Jun. 1977.

[12] Deux, O., "The Story of O2," IEEE TKDE, pp. 91-108, Mar., 1990.

[13] Goldberg,A. and Robson, D., SMALLTALK-80: The Language and its Implementation, Addison Wesley, 1985.

[14] LaLonde, W.R., Thomas, D.A., and Pugh, J.R., "An Exemplar Based Smalltalk," In Proc. OOPSLA, pp. 322-330, 1986.

[15] Zdonik,S., "Why Properties are Objects of Some Refinements of "is-a"," In ACM/IEEE Fall Joint Comp. Conf., pp. 41-47, Nov.1986.

[16] Halbert, O. and O'Brien, P., "Using Types and Inheritance in Object-Oriented Programming," IEEE Software, pp. 71-79, Sep. 1987.

[17] Borgida, A., Class Hierarchies in Information Systems: Sets, Types, or Prototypes, In Atkinson, M., Buneman, P.,

and Morrison, R., ed., Data Types and Persistence, Chap.10, Springer Verlag, pp. 137-154, 1988.

[18] Jarke,M. and Koch, J., "Query Optimization in Database Systems," ACM Computer Surv., Vol. 16, No. 2, pp. 112-152, Jun. 1984.

[19] Gelder, A. and Topoer, R.W., "Safety and Correct Translation of Relational Calculus Formalas," In Proc. PODS, pp. 313-327, 1987.

[20] Ozsoyglu, Z. and Yuan. L., "A New Normal Form for Nested Relations," ACM TODS, Vol. 12, No. 1, pp. 111-136, Mar. 1987.

[21] Graefe, G. and Maier, D., "Qurey Optimization in Object-Oriented Database Management Systems with Encapsulated Behavior," Tech. Rep., Oregon Graduate Center, Portland, Oregon, 1989.

[22] Talbot, S., "An Investigation into Logical Optimization of Relational Query Languages," the Computer Journal,27, pp. 301-309, 1984.

[23] Hasan,W. and Pirahesh, H., "Query Rewrite Optimization in Starburst," Tech. Rep., TR RJ 6367, IBM Almaden Research Center, Aug.1988.

[24] Lohman,G., "Grammar-like Functional Rules for Representing Query Optimization Alternatives," In Proc. ACM SIGMOD, pp. 18-27, 1988.

[25] Cardelli, L. and Wegner, P., "On Understanding Types, Data Abstraction, and Polymorphism," ACM Computer Surv., Vol. 17, No. 4, pp. 471-522, Dec. 1985.

[26] Lee, H.R, "A Logical Optimization of Queries in Object Oriented Database System," PhD thesis, the Chonbuk National Univ. at Chonbuk in Korea 1994.

[27] Ling,L., "A Recursive Object Algebra Based on Aggregation for Manipulating Complex Objects," Data & Knowledge Engineering, Vol. 11, North-Holland, pp. 21-60, 1993.

이 홍 로

1984년 전북대학교 전기공학과 졸업(학사)
1986년 전북대학교 대학원 전자계산기 전공(공학석사)
1994년 전북대학교 대학원 전산응용공학 전공(공학박사)
1994년~현재 충북대학교 컴퓨터과학연구소 연구원
관심분야 : 객체지향 언어, 객체지향 데이타베이스, 질의처리

곽 훈 성

1971년 전북대학교 졸업
1973년 전북대학교 대학원 전자공학 전공(석사)
1978년 전북대학교 대학원 전자공학 전공(박사)
1981년~82년 Univ. of Texas 객원교수
1982~95년 전북대학교 전자계산소장
1978년~현재 전북대학교 컴퓨터공학과 교수.
관심분야 : 영상처리, HDTV, 패턴인식, 인공지능, 멀티미디어 및 객체지향 시스템

류 근 호

1976년 숭실대학교 전산학과 졸업
1980년 연세대 산업대학원 전산전공(공학석사)
1988년 연세대 대학원 전산전공(공학박사)
1976년~86년 육군군수 지원사 전산실(ROTC장교), 한국전자통신연구소(연구원), 한국방송통신대 전산학과(조교수) 근무
1986년~현재 충북대학교 컴퓨터과학과 부교수겸 컴퓨터과학연구소장
관심분야 : 시간지원 데이타베이스, 시공간 데이타베이스, DBMS 및 OS, 객체 및 지식베이스 시스템