관계 해석 (Relational Calculus)

Chapter 4, Part B

관계 해석

- 종류 : 투플 관계 해석(*Tuple relational calculus* : TRC) 과 도메인 관계 해석(*Domain relational calculus* : DRC).
- 변수, 상수, 비교 연산자, 논리 접속사, 정량자로 구성.
  - *TRC*: *투플*에 대한 (속박되는) 변수.
  - *DRC*: *도메인 요소*(=필드값)에 대한 변수.
  - TRC와 DRC 모두 1차 논리(first-order logic)의 부분 집합.
- 해석의 식은 *formula*(식)라고 한다. 주어진 식을 *참*으로 하는 상수를 변수의 값으로 하는 투플이 결과가 된다.

도메인 관계 해석

- *질의* 형식:
  {<x1, x2,…,xn> | *p*[<x1, x2,…,xn>]}
- *결과*는 식 *p*[<x1,x2,…,xn>]를 참으로 하는 모든 투플 <x1, x2,…,xn> 의 집합
- *식*은 간단한 *원자식*(atomic formula)이거나, 식을 *논리 접속사*로 연결한 것으로 순환적으로 정의된다.

DRC 식

- *원자식:*
  - 〈x1, x2,…,xn〉 ∈ *Rname*, 또는 X *op* Y, 또는 X *op* 상수
  - *op* 는 <, >, =, ≤, ≥, ≠
- *식:*
  - 원자식, 또는
  - ¬*p*, *p*∧*q*, *p*∨*q* (*p*와 *q*는 식), 또는
  - ∃*X*(*p*(*X*)), *X*는 *p*(*X*) 내에서 자유변수, 또는
  - ∀*X*(*p*(*X*)), *X*는 *p*(*X*) 내에서 자유변수

자유 변수와 속박 변수

- 정량자 ∃*X*와 ∀*X*가 붙으면 *X*는 *속박*된다고 함.
  - 속박되지 않은 변수는 자유변수
- 질의 형태 :
  {<x1, x2,…,xn> | *p*[<x1, x2,…,xn>]}
- '|'의 좌측에 오는 변수 x1, x2,…,xn 들은 식 p(…) 내에서 모두 자유 변수이어야 한다.

등급이 7을 넘어서는 모든 뱃사람을 구하라

{<*I*, *N*, *T*, *A*> | <*I*, *N*, *T*, *A*> ∈ 뱃사람 ∧ *T* > 7}

- 조건 <*I*, *N*, *T*, *A*> ∈ 뱃사람 때문에 도메인 변수 *I*, *N*, *T*, *A* 가 동일한 뱃사람 투플의 각 필드에 속박됨.
- '|' 좌측의 <*I*, *N*, *T*, *A*> ("다음과 같은…"의 의미)는 *T*>7을 만족하는 <*I*, *N*, *T*, *A*>들의 모임이 결과가 됨을 의미함.
- 다음이 되도록 고쳐볼 것 :
  - 'Joe'라는 이름이면서, 18세 이상이거나 등급이 9 이하인 뱃사람들을 구하라.

배번호 103을 예약한, 등급>7인 뱃사람들을 구하라

{<*I*, *N*, *T*, *A*> | <*I*, *N*, *T*, *A*> ∈ 뱃사람 ∧ *T* > 7 ∧
∃*Ir*, *Br*, *D* [<*Ir*, *Br*, *D*> ∈ 예약 ∧ *Ir* = *I* ∧ *Br* = 103]}

- ∃*Ir*, *Br*, *D* (…)는 ∃*Ir*(∃*Br*(∃*D*(…)))를 줄인 표기임.
- ∃는 주어진 조건 하의 뱃사람 투플과 예약 투플을 찾기 위해 사용한 것임.

적색 배를 예약한, 등급>7인 뱃사람들을 구하라

{<*I*, *N*, *T*, *A*> | <*I*, *N*, *T*, *A*> ∈ 뱃사람 ∧ *T* > 7 ∧
∃*Ir*, *Br*, *D* [<*Ir*, *Br*, *D*> ∈ 예약 ∧ *Ir* = *I* ∧
∃B, BN, C[<B, BN, C> ∈ 배 ∧ B = Br ∧ C = 'red']]}

- 괄호로 각 정량자가 미치는 범위를 제한함을 볼 것.
- 복잡해 보이지만, 사용자 인터페이스만 좋다면, 직관적으로 처리 가능(QBE!)

모든 배를 예약한 뱃사람들을 구하라

{<*I*, *N*, *T*, *A*> | <*I*, *N*, *T*, *A*> ∈ 뱃사람 ∧
∀*B*, *BN*, *C* [¬[<*B*,*BN*,*C*> ∈ 배] ∨
[∃*Ir*,*Br*,*D* [<*Ir*, *Br*, *D*> ∈ 예약 ∧ *I* = *Ir* ∧ *Br* = *B*]]]}

- 3쌍 <*B*, *BN*, *C*>가 배 테이블에 존재하지 않거나, 뱃사람 *I*가 이를 예약했다는 사실이 예약 테이블에 있는, 그러한 *I*를 모두 구하는 것이다.

모든 배를 예약한 뱃사람들을 구하라(다시 한번!)

{<*I,N,T,A*> | <*I,N,T,A*> ∈ 뱃사람 ∧
∀<*B,BN,C*> ∈ 배
[∃<*Ir, Br, D*> ∈ 예약 [*I=Ir* ∧ *Br=B*]]}

- 동일한 질의에 대한, 더 단순한 표현(더 명쾌해졌음!)
- 모든 적색 배를 예약한 사람들을 구하려면 :
  …[*C* ≠ '*red*' ∨∃<*Ir,Br,D*> ∈ 예약 [*I=Ir* ∧ *Br* = *B*]]}

불안전한 질의와, 질의의 표현력

- 구문상으로는 옳지만 결과가 무한집합이 되는 질의를 불안전한(*unsafe*) 질의라고 한다.
  - e.g., { S | ¬ [S ∈ 뱃사람] }
- 관계 대수로 표현 가능한 질의는 안전한 DRC/TRC로도 표현 가능하고, 그 역도 성립한다.
- 관계적 완전성(*Relational Completeness*) : 관계 대수/해석으로 표현 가능한 모든 질의를 질의어(SQL 등)가 표현 가능할 때.

요약

- 관계 해석은 비절차적으로서, 사용자가 원하는 것만 명세하며 그 방법은 명세하지 않는다(명세형)
- 대수와 안전한 해석은 표현력이 동일하며, 이로부터 "관계적 완전성"이 정의된다.