교재 7장 논리적 에이전트

4주차: 지식 표현과 논리

학습 내용:

* 명제적 논리에서 각 개념을 학습한다.
  + - 구문(syntax)
    - 의미론(semantics)
    - 타당성(validity)
    - 만족감(satisfiability)
    - 해석과 모델(interpretation and models)
    - 함의(entailment)
* 명제적 논리에서 다른 추론 메커니즘을 학습한다.

학습 목표:

* 지능형 에이전트에서 지식 표현의 중요성을 이해한다.
* 지식 표현 언어로서의 공식 논리의 사용을 이해한다.
* 자연어 기술을 논리 문장으로 표현할 수 있다.
* 추론 규칙을 적용하여 새로운 문장을 추출할 수 있다.

4.1 지식 표현과 추론

지식의 표현과 추론 과정은 인공 지능 분야의 핵심이다.

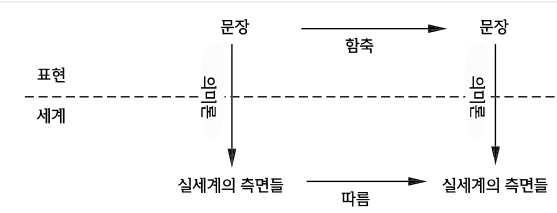
* 지능형 에이전트의 필수 요건
  + 1. 인지: 환경으로부터의 정보의 습득 능력
    2. 지식 표현: 문제의 세계를 이해하고 표현하는 능력
    3. 추론: 지식의 함의와 과 그것이 가지고 있는 선택의 함의를 추론
    4. 행동: 원하는 것을 선택하고 수행하는 능력

지식 기반 에이전트의 기본 구성 요소는 지식 기반이다.

* 지식 기반의 특징
  + - 지식 기반은 일련의 문장이다.
    - 각 문장은 지식 표현 언어라고 하는 언어로 표현된다.
    - 문장은 세계에 대하여 몇 가지 주장을 표현한다.
    - 오래된 주장으로부터 새로운 문장을 유도하는 메커니즘(추론 과정)이 있어야 한다.
    - 추론은 새로운 문장은 이전 문장과 논리적으로 타당성이 담보되어야 한다.
* 인공지능과 논리
  + - 지식을 표현하고 추론하는 주요 수단
    - 인공지능의 언어는 ‘형식화 된 논리(formal logic)’
    - 형식화 된 논리로 표현된 지능은 정확성과 확실성을 제고
    - 논리의 표현으로 선언적 프로그래밍이 가능
    - 논리는 사실만을 설명하고 문제 해결 방법은 미제공
    - 논리는 범용적 추론을 위한 자동 추론 기술 허용
* 논리 또는 형식화된 논리의 한계
  + - 인간은 불확실한 지식을 사용하여 추론하는 반면 논리는 불확실한 지식의 표현 한계
    - 자연어를 이용한 추론에서 인간은 숨겨진 상태나 화자의 의도를 유추가능하나, 논리는 불가
    - 인간은 통상적이고 유한한 지식을 사용하여 다양하고 무한한 표현에 대처 가능하나 논리는 불가
    - 논리는 모호성에 취약
* 논리의 구성

1. 언어와 추론 방법으로 구성
2. 논리 언어도 구문과 의미의 2가지 측면을 보유

* 특정 논리의 정의 또는 지정하기 위한 3가지
  + 1. **구문(Syntax)**: 구문은 허용되는 문장들을 정의하고 원자적 문장(atomic sentence)은 하나의 명제 기호(proposition symbol)로 구성된다. 간단한 문장들을 괄호와 논리 접속사로 결합하여 복합 문장을 구성한다. 세상의 사실은 논리에서 문장으로 표현된다.
    2. **의미론 (Semantics)**: 논리의 원자적 문장의 의미와 비원자적 표현의 의미들의 진리를 결정하는 규칙을 정의. 문장이 의미하는 세계의 사실이 명시되므로 세계에서의 의미에 따라 문장에 진리 값을 할당하는 방법도 지정한다. 사실은 세상에 대한 주장이므로 참이거나 거짓 일수도 있다.
    3. **구문 추론 방법(Syntactic Inference Method):** 논리의 정리(theorems)라고 불리는 논리식의 부분 집합을 결정하기 위한 규칙, 기존 문장에서 새로운 또는 참(true)인 문장을 계산 (유도)하는 기계적 방법을 의미한다.
* 사실은 참 또는 거짓 인 세계에 대한 주장이다. 반면 표현은 컴퓨터 프로그램에서 인코딩되고 세계의 대상과 관계를 나타낼 수 있는 표현(문장)이다. 따라서 그 표현이 현실과 일치하도록 다음 그림이 성립하도록 해야 한다. 여기서 문장들은 에이전트의 물리적 구성들이고 추론은 기존의 물리적 구성들로부터 새로운 물리적 구성을 구축하는 공정이다. 논리적 추론은 새 구성이 나타내는 세계의 측면들이 기존 구성들이 나타내는 측면들을 따른다는 점을 보장해야 한다.



* 논리적 시스템의 분류: 다른 구문과 의미론을 갖는 다양한 논리적 시스템이 존재하며 대표적인 것 들은 다음과 같다.
  + 1. **명제적 논리**: 설명된 모든 객체는 고정되거나 고유하다. 예) 영희는 학생이다. 여기서 영희는 유일한 사람을 가리킨다.
    2. **일차 술어 논리**: 설명 된 객체는 고유하거나 고유한 객체를 나타내는 변수 일 수 있다. 예) 학생들은 돈이 없다. 여기서 학생은 특정한 학생으로 대체가 가능하다.

장점: 프로그램을 간단하게 작성할 수 있게 해준다.

예) For All(A, B)[brother(A, B) → brother (B, A)]는 형제에 관한 가능한 진술을 반으로 줄일 수 있다.

4.1.1 명제적 논리

명제적 논리에서 명제적 기호들의 집합을 정의하고 각 심볼들의 의미가 정의된다.

예를 들어 명제 기호 P와 Q를 정의한다면

* + - * P는 ‘덥다’라는 의미이다
      * Q는 ‘습기가 있다’라는 의미이다.
      * R은 ‘비가 내리고 있다’라는 의미이다.
* 문장(공식 또는 잘 형성된 공식(wff))은 다음과 같이 정의된다.
  + 1. 기호
    2. 만약 S가 문장이라면, ~S는 문장이다. 여기서 ‘~’는 ‘not’의 논리 연산자이다.
    3. 만약 S와 T가 문장이라면, (S∨T), (S∧T), (S⇒T)와 (S⇔T)들은 문장이다. 여기서 4개의 논리적 접속사는 순서대로 ‘AND’, ‘OR’, ‘함의(implies)’, ‘if and only if’ 이다.
    4. 위 1~3 까지를 유한한 혼합 적용
* 명제적 논리의 문장 예제
  + - * (P∧Q) ⇒ R: ‘만약 덥고 습기가 있다면 비가 오고 있다’라는 명제적 논리의 문장이다.
      * Q ⇒ P: ‘만약 습기를 있다면 덥다’ 라는 명제적 논리의 문장이다.
      * Q: ‘습기가 있다’ 라는 명제적 논리의 문장이다.
* 명제적 논리 문장의 해석: 문장에 구성된 모든 원자적 기호의 진리값이 주어지면 그 문장의 진리값을 결정하기 위한 ‘평가’하는 것을 의미한다.
* 모델은 각 문장이 참이 되도록 한 세트의 문장을 해석(예를 들면 기호들의 진리값을 할당)이다. 즉, 모델은 세상에 ‘서있는(stand in)’공식적인 수학적 구조일 뿐이다.
* 유효한 문장(유의어 반복이라고도 함)은 모든 해석에서 참인 문장이다. 따라서 세상이 실제로 어떤 것인지 또는 의미가 무엇이든 간에 문장은 참이다. 예를 들면 (R∧~R: 비가 오고 비가 오지 않는다)는 참이다
* 문장 ‘P는 Q를 함축한다.’는 P ⊨Q로 표기 된다. P가 참이면 Q도 참이다. 즉, P의 모든 모델은 Q의 모델이다.

**논리적 함축(entailment)**: 함축이란 한 문장이 다른 문장을 “논리적으로 따른다(follow)”는 개념을 나타내는 용어이다. 표기와 의미는 아래와 같다.

α⊨β: ‘문장 α 가 문장 β를 함축한다’고 하며 그 의미는 “만일 α 가 참인 모든 모형에서 β역시 참이면, 그리고 오직 그럴 때에만, α⊨β” 이다.

예) 함축의 증명

p∧(p ⇒q) ⊨ q가 성립함을 보여라.

풀이: p∧(p⇒q)가 성립하는 임의의 모델 M에 대해서 p는 M에서 성립되고 p⇒q 도 M에서 성립하므로 q는 반드시 M에서 성립되어야 한다. 그러므로 p∧(p⇒q)가 성립되는 모든 모델에서 q가 성립되므로 p∧(p⇒q)⊨q.

**등가성(equivalence)**: 만약 2개의 문장이 동일한 모델에서 정확히 성립한다면 2문장은 동일하다.

p ⇒q≡┓p∨q

예) 등가성의 증명

p ⇒q≡┓p∨q가 성립함을 보여라.

풀이: p ⇒q 가 성립하는 임의의 모델 M에 대해서 p는 M에서 성립되고 q도 M에서 성립되거나 또는 p가 성립이 안되면 ┓p는 성립된다. q는 M 또는 ¬p는 M에서 유지되고 ¬pVq는 M에서 유지되어 ┓p∨q=p⇒q.

┓p∨q 가 성립하는 임의의 모델 M에 대해서 ┓p 는 M에서 성립되고 q도 M에서 성립된다. 그렇지 않으면 q가 M에서 유지되므로 p⇒q는 M에서 성립한다. 그러므로 p⇒q가 M에서 성립한다.

4.1.2 명제 논리의 추론

대상세계에 관한 어떤 사건의 진술(statement)을 명제 (proposition) 라고 한다. 아래와 같은 진술들이 명제의 예이다.

(1) 영이는 여성이다.

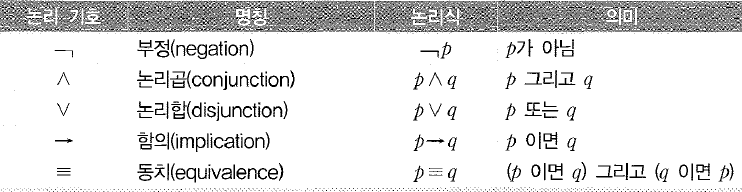
(2) 길동이는 영이를 좋아한다.

(3) 길동이와 영이는 학생이다.

(4) 기린은 신성한 동물이다.

(5) 눈은 희다.

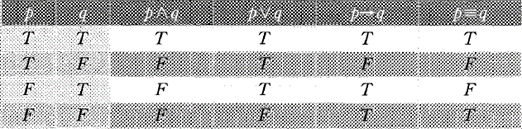
* 명제는 진술의 내용을 파고들지 않고 진술 그 자체가 맞는지 아닌지(참과 거짓)가 대상이 된다. 예를 들어 위에 기술한 예 (4) 에서 “기린”의 존재에 대해서는 문제삼지 않고 진술 그 자체의 진위를 논의의 대상으로 한다
* 하나의 진술로 이루어지는 명제의 최소단위를 기본명제 (primitive proposition)라고 하고， 복수의기본명제로 구성되는 명제를 복합명제 (compound proposition)라고 한다. 앞의 예 (3)에서는 “길동이는 학생이다 “영이는 학생이다” 라는 두 개의 기본명제가 하나의 복합명제로 표현되어 있는 예가 된다.
* 명제 간의 논리적 관계를 논의하는 것이 명제 논리이며, 명제의 진위를 기호를 사용하여 형식화가 된다.
* 명제의 기호표현형식을 논리식(logical formula)이라고 하고, 명제변수(p, q, r 등으로 표현)라고 하는 명제를 표현하는 변수 및 아래에 나타낸 논리기호에 의해 표현된다. ⇒



4.1.2.1 명제 논리식의 귀납적 정의

다른 명제로 분할되지 않는 기본 명제를 원자식 (atomic formula) 이라고 한다. 명제 논리식과 명제 논리식을 결합하는 기호 ┓, ∧, ∨, ⇒ (또는 →), ≡를 논리기호라고 한다.

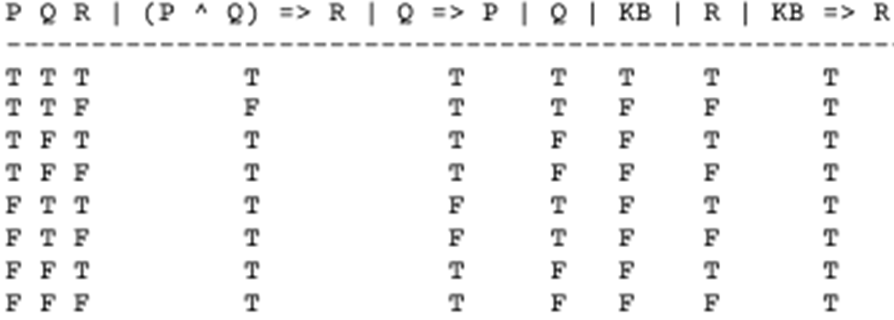
* 명제 논리식의 정의
  1. 원자식은 명제 논리식이다.
  2. p, q가 명제 논리식이라면， 논리 기호를 사용하여 구성되는 명제 논리식, ┓p, p∧q, p∨q, p⇒q, p≡q도 명제 논리식이다.
  3. ①과 ②에서 정의되는 논리식만 명제 논리식이다.
* 논리 기호의 우선순위 및 논리식의 해석
* 결합된 논리식에서 논리 기호의 연산 우선 순위는 ┓, ∧, ∨,⇒, ≡의 순이다.
* 논리식의 진위는 그 논리식을 구성하고 있는 원자식의 진위를 각각 구하고, 논리기호로 결합된 원자식들을 고려하여 원래의 논리식 전체의 진위를 결정한다. 이 과정을 ‘논리식을 해석한다 (interpretation)’ 라고 한다.
* 논리식의 진위를 결정하기 위하여 진리표(truth table)를 사용한다. 진리표란 원자식의 여러 가지 조합에 대한 논리식의 진위를 표 형식으로 정리한 것이다. 참을 T, 거짓을 F로 나타내며, 진리표의 예는 다음과 같다.



예제) 아래와 같이 3개의 날씨관련 문장 P, Q, R과 지식 베이스 KB 가 주어졌을 경우, KB는 일반적인 날씨에 관한 서술 문장들(① 덥고 습기가 있으면 비가 오고 있다, ② 습기 있으면 덥다, ③ 습기가 있다.) 해당하는 논리 표현이다. “비가 오고 있다”의 질의문장 R은 KB에 함축되어 있는가? 진리표를 이용하여 이 질의에 답하시오.

* P는 "덥다"
* Q는 "습기가 있다"
* R은 "비가 오고 있다"
* KB = (((P ^ Q) => R) ^ (Q => P) ^ Q)

(풀이)



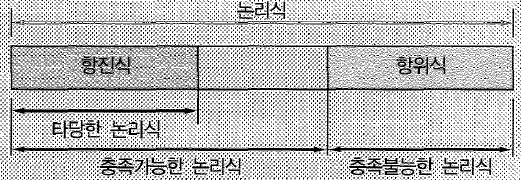
진리표의 우측열을 보면 KB가 R을 함축함을 알 수 있다.

4.1.2.2 항진식과 항위식

논리식에서는 그 논리식을 구성하는 원자 논리식이 취하는 진리값에 관계없이 항상 참이 되는 것이나 항상 거짓이 되는 것이 존재한다.

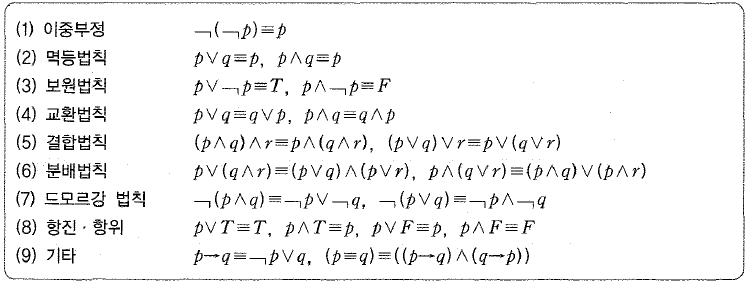
* 항진석 (tautology): 원자식에 어떠한 값이 주어지더라도 항상 참이 되는 논리식.
* 항위식 (contradiction): 원자식에 어떠한 값이 주어지더라도 항상 거짓이 되는 논리식.
* 타당(valid): 항진식인 논리식은 어떠한 해석에 대해서도 참이 되므로 이 논리식을 ‘타당하다’고 말함.
* 충족 가능한(satisfiable): 논리식이 항위식이 아닐 경우 그 논리식을 참이 되게 하는 해석이 적어도 하나는 존재하므로 이 논리식은 “충족 가능하다” 고 말함.
* 충족불능(unsatisfiable): 항위식인 논리식에 대해 항진식이라는 해석은 결코 성립되지 않으므로 이 논리식은 ‘충족불능이다’고 말함.

이상의 관계를 그림으로 표현하면 다음과 같다.



4.1.2.3 논리식의 동치관계

진리값이 같은 논리식은 동치관계(equivalence relation)에 있다고 한다. 두 개의 논리식이 동치라는 것을 증명하기 위해 아래에 나타내는 것과 같은 대표적인 동치관계를 빈번하게 증명에 이용한다.



4.1.2.4 논리식의 표준형

여러 가지 논리식 및 논리 기호를 이용하여 동치가 되는 논리식이 다수 존재하기 때문에 논리식의 표준형식을 정해두는 것이 논리식을 다룰 때 편리하다.

표준형을 구성 하는 리터럴(literal), 절(clause), 절 형식(clause form)에 대하여 아래에 설명한다.

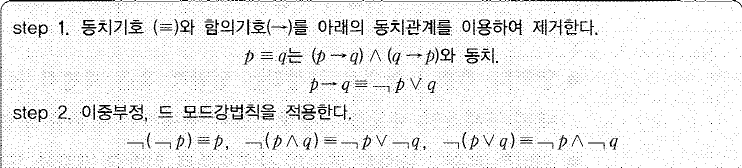
* 리터럴: 원자식(p), 혹은 원자식의 부정 (┓P)를 말함.
* 절: 리터럴의 논리합(∨)만으로(혹은 논리곱(∧)만으로) 구성되는 논리식을 말함.
* 절 형식: 리터럴의 논리합(∨)만으로 구성된 절들이 논리곱(∧)만으로 구성된 논리식, 또는

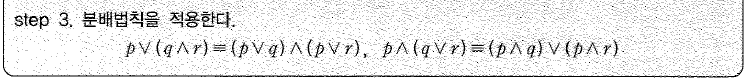
리터럴의 논리곱(∧)만으로 구성된 절들이 논리합(∨)만으로 구성되는 논리식을 말함.

* pij를 리터럴이라고 하면 리터럴의 논리합만으로 구성된 논리식 ci≡pi1∨ pi2∨…∨pin은절이다. 이 때, 절의 논리곱만으로 구성된 논리식 q≡c1∧c2 ∧… ∧cm은 절 형식이다.
* 논리식의 표준형(normal form)은 절 형식으로 표현된다.
* 논리곱표준형(합곱표준형, 연언표준형:CNF): 리터럴의 논리합(∨)만으로 구성된 절들이 논리곱(∧)만으로 구성된 논리식
* 논리합표준형(곱합표준형, 선언표준형:DNF): 리터럴의 논리곱(∧)만으로 구성된 절들이 논리합(∨)만으로 구성된 논리식

4.1.2.5 논리식의 절 형식 변환

임의의 논리식은 절 형식으로 변환할 수 있다. 그 과정은 다음과 같다.





예 1) B1⇔P1∨P2인 문장을 논리곱 표준형으로 변환하라.

단계 1: ⇔를 소거하고, α⇔ß를 (α ⇒ß)∧(ß⇒α)로 대체한다.

(B1⇒(P1∨P2)) ∧((P1∨P2) ⇒B1)

단계 2: ⇒를 제거하고, α ⇒ß를┓α ∨ß로 대체한다.

(┓B1∨ P1∨P2) ∧(┓(P1∨P2) ∨B1)

단계 3: CNF에는 ┓이 리터럴에만 있어야 한다. 이를 위해, 동치식들 중 다음 세 개를 반복 적용해서 “┓을 괄호 안으로 옮긴다”.

┓(┓α) ≡α (이중 부정 소거)

┓(α ∧ß)≡(┓α∨┓ß) (드모르간)

┓(α∨ß)≡(┓α∧┓ß) (드모르간)

여기서는 마지막 규칙을 한번만 적용한다.

(┓B1∨ P1∨P2) ∧( (┓P1∧┓P2) ∨B1)

단계 4: 이제 ∧, ∨ 연산자와 리터럴들로 이루어진 부분들이 중첩된 형태의 문장이 나왔다. 분배법칙을 적용해서 ∨을 ∧에 대해 적절히 분배한다.

(┓B1∨ P1∨P2) ∧ (┓P1∨B1) ∧(┓P2) ∨B1)

이와 같이 원래의 문장이 세 절의 논리곱 형태로 된 CNF로 변환되었다. 이전보다 훨씬 읽기 어렵긴 하지만, 완결적인 분해 절차의 입력으로 사용할 수 있다.

예 2) 다음 논리식을 논리곱 표준형의 절 형식으로 변환해보자. ⇒

p∧(q⇒r) ⇒ p∧q

단계 1: 함의(⇒)를 제거한다.

≡┓(p∧(┓q∨r)) ∨(p∧q)

단계 2: 이중부정， 드모르강 법칙을α 적용한다.

≡(┓p∨┓(┓q∨r))∨(p∧q)

≡(┓p∨ (q∧┓r))∨(p∧q)

단계 3: 분배 법칙을 적용한다.

≡(┓p∨q) ∧ (┓p∨┓r))∨(p∧q)

≡((┓p∨q)∨(p∧q))∧((┓p∨┓r)∨(p∧q))

≡(((┓p∨q)∨p)∧((┓p∨q) ∨q))∧(((┓p∨┓r)∨p)∧((┓p∨┓r)∨q))

≡(┓p∨q)∧(┓p∨q∨┓r)

4.1.2.6 추론과 논리적 귀결

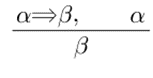
* 추론(reasoning):

“함의” 의 논리적 관계를 이용하여 전제(premise)가 되는 명제로부터 결론(conclusion)을 이끌어내는 행위이다. 추론은 크게 나눠서 연역적 추론(deductive reasoning)과 귀납적 추론(inductive reasoning)의 두 종류로 분류된다.

* **연역적 추론**: 함의로 구성되는 일반적 혹은 보편적 논리 명제인 전제에 대해 구체적인 사실을 적용함으로써 구체적인 결론을 도출하는 추론 방법.
* **귀납적 추론:** 관측된 복수의 사실로부터 일반적 명제를 도출하는 추론 방법.
* 추론 규칙:

증명(proof)을 이끌어 내는 데 사용할 수 있는 추리 규칙(inference rule)들을 살펴본다. 증명이란 어떤 원하는 목표로 이어지는 결론들의 사슬이다.

* 1. 긍정식: 증명을 구축하는 데 쓰이는 가장 잘 알려진 규칙은 긍정식(Modus Ponens; 전건 긍정)이다. 이 식의 의미는 α⇒ß와 α형태의 임의의 문장들이 주어졌을 때 문장 ß를 추리할 수 있음을 뜻한다.

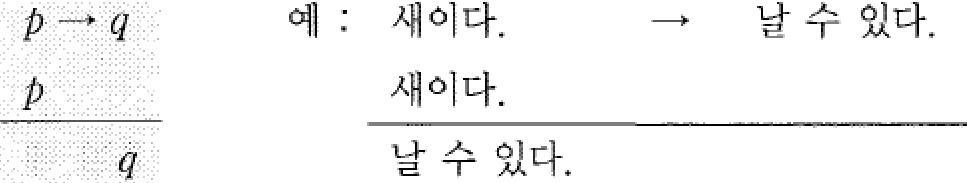


* 1. 논리곱 소거: 논리곱 소거(AND–elimination)는 논리곱 문장에서 임의의 논리곱 성분을 추리할 수 있음을 의미한다.

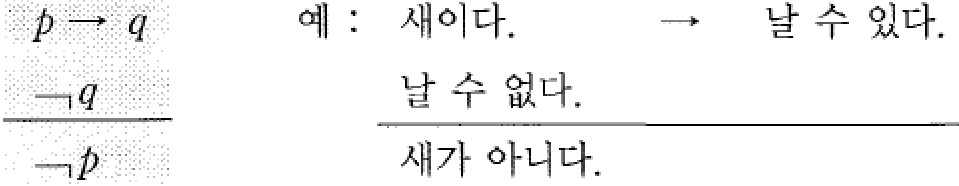


* **연역적 추론의 규칙**: 명제 논리에서는 주로 연역적 추론을 다루며, 필요한 기본 규칙은 다음과 같다. 이 추론규칙들은 모두 “타당한” 추론으로 인정받는 것으로써, 일반적으로 추론은 이들을 기초로 하여 추론 형식을 조합하여 구성된다.

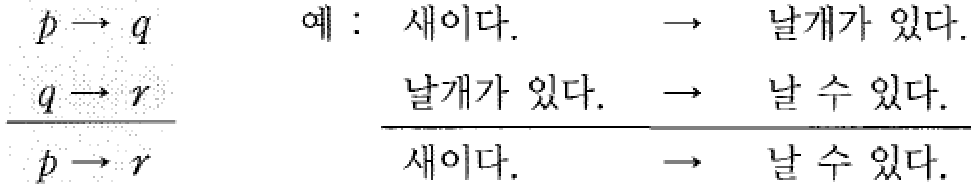
* + - 긍정식



* 부정식



* 삼단논법



* **논리적 귀결:** (p1∧ p2 ∧…∧ pn) ⇒ q와 같이 나타나는 논리식에서 전제가 참이 될 때 결론도 반드시 참이 되는 경우, q는 p1, p2,…, pn으로부터의 논리적 귀결 (logical consequence) 이라고 한다.
* **건전성:**

논리적 귀결이 구해졌을 때 논리식의 집합 {p1, p2, … pn}으로부터 논리식 q를 도출하는 추론은 건전하다고 한다. 이것은 전제가 참이라면 결론도 반드시 참이라는 진리보존성 이 성립되는 것으로부터 이렇게 불린다. 이처럼 진리보존성이 성립하고 있는 추론을 **타당한 추론**이 라고 한다.

* 공리와 정리:

논리식이 참이라는 것이 추론에 의해 도출될 때 추론의 최초가 되는 항진의 논리식을 공리(axiom)라고 한다. 공리에 건전한 추론규칙을 적용하여 얻어지는 논리식을 정리 (theorem) 이라고 한다. 공리의 집합을 공리계(axiom system) 라고 한다. 공리계는 주어진 논리식이 항진식인지 증명하기 위하여 이용되어, 주어진 논리식이 참인지를 판정하는 추론의 근거를 제공한다.

4.1.2.7 형식적 증명

공리계로부터 출발하여 추론규칙을 적용함으로써 정리를 도출하는 것을 형식적 증명 (formal proof) 이라고 한다.

* 논리식의 형식적 증명

논리식의 유한계열 p1, p2, … pn 이 있고， k = 1, 2, …， n의 각각에 대해서

(1) p1은 공리이다.

(2) pk (1‹k≤ n)는

(a) 공리이다. 혹은，

(b) i, j ‹ k인 pi와 pj로부터 추론규칙 r1 에 의해 직접 도출된 것이다.

추론규칙 r1 는 전형적인 것으로 p와 p⇒q가 정리식이라면, q는 정리식이다(이 추론규칙은 긍정식이다). 가 있다

이 때， 논리열 p1, p2, … pn 을 논리식 pn 의 형식적 증명이라고 한다.

논리식 p에 대해 형식적 증명이 있을 때 이것을  P라고 표기한다.

예) 명제의 공리계로 다음을 형식적 증명을 한다.

A1: p⇒(q⇒p)

A2:(p⇒(q⇒r)) ⇒ ((p⇒q) ⇒(p⇒r))

A3: (┓p ⇒ ┓q ) ⇒((┓p⇒q) ⇒p)

여기서 추론 규칙은 B1: p와q가 정리식이라면, q도 정리식이다.

풀이) 위에 기술한 공리 A1, A2, A3 그리고 추론규칙 B1을 이용하여  P ⇒P를 증명한다.

단계 1: 아래의 논리식이 도출하기 위하여 공리 A1에서 p 三P, q 三P ⇒ P로 놓자.

P ⇒ ((P ⇒ P) ⇒ P)

단계 2: 공리 A2에서 p 三 P， q 三P ⇒ P, r 三P로 놓으면 아래와 같다.

(P ⇒ ((P ⇒ P) ⇒ P)) ⇒ ((P ⇒ (P ⇒ P)) ⇒ (P ⇒ P))

단계 3: 단계1, 2의 결과에 추론규칙 B1을 적용한다.

(P ⇒ (P ⇒ P)) ⇒ (P ⇒ P)

단계 4: 공리 A1에서 p 三P， q 三P로 두면 아래와 같이 된다.

P ⇒ (P ⇒ P)

단계 5: 단계 3, 4의 결과에 추론규칙을 적용함으로써 아래의 논리식이 도출된다.

P ⇒ P

이로써 P ⇒P가 증명되었다.

4.1.2.8 가설로부터 연역

논리식이 참인 것을 증명하기 위해 항진인 공리에 건전한 추론을 적용하는 것을 반복함으로써 그 논리식을 도출하는 작업을 수행한다. 그러나 어떤 공리를 사용하고 어떤 추론을 적용해야 하는지 등이 문제가 되어 증명하고 싶은 논리식이 참인 것을 도출하는 것은 쉽지 않다. 그래서 어떤 가설을 정의하고 그 아래에서 주어진 논리식이 참이라는 것을 증명한다. 이 방법을 가설로부터의 연역(deduction from hypothesis)라고 한다.

* 가설로부터 연역

a1, a2, … am 을 논리식의 유한계열이라고 할 때, 다음과 같은 조건을 만족하는 논리식의 유한계열 p1, p2, … pn 을 가설 a1, a2, … am 으로부터의 논리식 pn 의 연역이라고 한다

(1) pi (1≤ i ≤n)는 공리이거나 가설 a1, a2, … am의 논리식이다.

또는

(2) pi (1≤ i ≤n)는 pi 와 pk (1 ≤j , k ≤i) 로부터 추론규칙에 의해 직접 도출된 논리식이다. 여기서 pi 와 pk (1 ≤j , k ≤i) 는 공리이거나 가설에 추론규칙을 적용하여 도출한 정리식이다.

논리식 p에 대하여 가설 a1, a2, … am 으로부터의 연역이 존재할 때 p는 가설 a1, a2, … am 으로부터 연역 가능하다고 하여,

a1, a2, … am P로 표기한다.

가설로부터 가설이 없는 연역을 도출하기 위해서는 연역정리 (deductive theorem)을 이용한다. 이것은 가설 상에서 성립하는 논리식은 가설을 논리식의 전제조건으로 가정하여 가설을 줄여나가는 방법이다.

* 연역 정리

만약

a1, a2, … am p라면,

a1, a2, … am-1  am ⇒p이며, 그 역도 성립한다.

이 연역정리를 반복하여 이용함으로써

a1, a2, … am p로부터

a1⇒(a2⇒ … (am⇒p) …) 가 구해진다. 즉, 가설에 대하여 하나씩 가설로부터의 연역을 수행해 나가서 최종적으로 가설이 없는 연역을 구하는 것이 가능하다.

4.2. 술어논리

술어 논리 (predicate logic)란 명제 논리와 달리 각 명제의 내용에 관하여 논하기위하여 명제 안에 변수를 도입하고 그 변수값에 의해 참·거짓을 취하는 기호논리를 가리킨다. 명제 논리는 진술로써 참인지 거짓인지를 묻는 것이고, 진술 안에 나타나는 내용에 관한 추론은 불가능하다.

예를 들어,

(Pl) 모든 사람은 평화를 사랑한다.

(P2) 길동은 사람이다.

따라서，

(P3) 길동은 평화를 사랑한다.

위에 기술한 (Pl) 에서 (P3)까지 명제 논리의 틀에서는 (Pl), (P2)가 참이라고 해도 (P3) 가 항상 참이라고 주장할 수 없다. 왜냐하면 명제 논리 하에서는 명제 안에 포함된 개체 (individual, 즉 사람, 길동, 평화 등))에 관한 내용에는 주목하지 않고 명제 그 자체의 진위성에만 주목하기 때문이다.

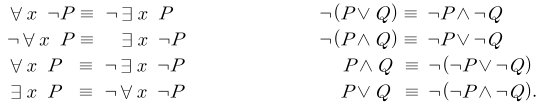
이와 달리 술어 논리는 명제에 포함된 개체변수에 주목하여 그 명제에서의 개체변수의 성질이나 상태를 술어 (predicate, 즉 ~사랑한다, ~이다 등))를 이용하여 추론을 수행한다.

4.2.1 술어 논리식의 정의

술어 논리에서는 명제의 내용을 개체변수를 인수로 가지는 술어의 형태로 표현하고, 이 술어를 이용하여 추론을 수행한다. 술어 논리는 명제 논리와 달리 개체변수를 표현하는 변수를 도입함으로써 명제 논리를 확장한다. 술어 논리로 표현되는 명제는 컴퓨터 상에서 인간의 지식을 표현하는 수단으로도 널리 이용되고 있으며， 특히 추론으로 도출되는 지식을 표현하는 데에 적합하다. 술어 논리를 구성하는 요소들은 다음과 같다.

* 개체정수(individual constant): 특정한 개체를 나타내는 기호(a, b, … 등으로 정수로 표현된다).
* 개체변수(individual variable): 임의의 개제를 나타내는 기호(x, y, … 등의 변수로 표현된다).
* 술어기호(predicate symbol): 개체에 관한 성질이나 상태를 나타내는 기호(p, q, … 등의 기호로 표현된다).
* 함수 기호(function symbol) 개체간의 관계를 나타내는 기호 (f(x) 의 f 등으로 표현된다).
* 논리기호(logical symbol): 명제 논리의 논리 기호(┓, ∧, ∨ , ⇒)와 같다.
* 정량기호(quantifier): 대상영역에서 대상이 되는 개체의 범위를 나타내는 기호이며 아래의 두 가지가 있다.
  + 1. ∀: 전칭기호(universal quantifier)라고 하며 술어가 「임의의」 항(후술)에 대하여 성립하는 것을 나타낼 때 사용된다.
    2. ∃: 존재기호(existential quantifier)라고 하며， 술어가 「적어도 하나의」 항에 대하여 성립하는 것을 나타낼 때 사용된다.
    3. ∀와∃의 관계

∀는 사실 모든 객체의 논리곱이고 ∃는 논리합이므로, 이들이 드모르간의 법칙을 만족한다고 해서 놀랄 일은 아니다. 다음은 문장들에 대한 드모르간 법칙이다.



4.2.1.1 술어 논리식의 정의

아래에 술어 논리식의 정의를 나타낸다. 술어의 인수가 되는 항(term)은 술어논리가 대상으로 삼는 세계인 대상영역 (domain) 에서의 요소를 표현하는 것이고, 술어의 인수에 해당한다. 항에는 개체정수, 개체변수 외에 함수 자체도 항이 될 수 있기 때문에 아래와 같이 귀납적으로 정의할 수 있다.

* 항:

(1) 개체정수, 개체변수는 항이다.

(2) t1, t2,…, tn이 항이 되고, f가 n변수의 함수기호일 때, f(t1, t2, …, tn)도 항이다.

(3) 이상의 (1), (2)로부터 항으로 판별될 수 있는 것만 항이 되고 항은 대상 영역의 원소이다.

* 술어 논리식

(1) t1, t2,…, tn항이고, p가 n변수의 술어기호일 때, p(t1, t2,…, tn)을 원자식이라고 한다. 원자식은 논리식이다.

(2) p, q가 논리식일 때 ┓p, p∧q, p∨q, p⇒ q, p 三q는 논리식이다.

(3) p가 논리식이고, x가 개체변수일 때∀xp, ∃ xp는 논리식이다.

(4) 이상의 (1), (2), (3)으로부터 논리식으로 판별될 수 있는 것만 논리식이다.

예) 위에서 열거한 술어의 명제 (Pl) ~ (P3) 에서 개체정수 “길동”, “평화” 등은 항이 된다. 또한 “사람이다”를 술어라고 하면 (P2)는 “사람이다(길동)"과 같이 술어 논리식으로 나타낼 수 있다.

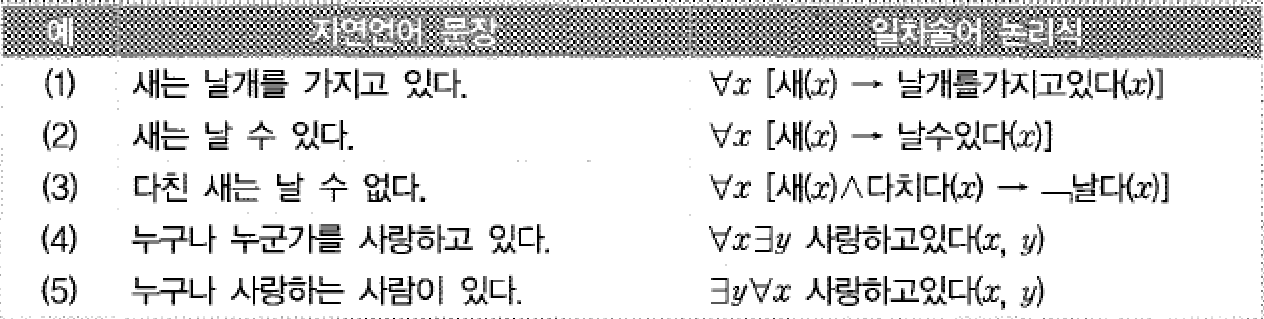
4.2.2 일차 술어 논리(first order predicate logic)

일차 술어 논리는 명제에서의 술어가 자신의 항으로 술어를 포함하지 않는 것을 말한다. 술어를 항으로 포함하는 것을 이차 술어 논리라고 하고, 나아가 반복해서 포개진 상태로 술어를 포함하는 술어 논리를 일반

화하여 고차 술어 논리 (higher-order predicate logic)이라고 한다. 이것은 자연언어에서 문장을 생각하면 이해하기 쉽다. 예를 들어 “눈이 하얗다”에 대하여 항으로 개체정수 “눈”을 취하고 있다. 이것과 비교하여 “나는 눈이 희다고 생각한다” 라는 명제는 술어 “생각하다” 에 대해 “눈이 희다” 라는 술어를 포함하는 명제를 항으로 가지고 있으므로 이차 술어 논리로 취급된다 이 장에서는 일차 술어 논리에 대해서만 설명한다.

* 일차 술어 논리에 의한 지식 표현

자연언어 문장의 의미를 일차 술어 논리로 나타낸 예를 아래에 나타낸다.



예(1) “새는 날개를 가지고 있다” 는 일차술어 논리 표현에서 ∀x [새(x) ⇒ 날개를 가지고 있다(x)]로 표현된다. 여기서 x는 개체변수이고， 새 (x)는 “x는 새이다” 라는 술어 논리식을 나타낸다. 정량기호에 전칭기호 ∀를 이용함으로써 임의의 개체변수 x를 대상으로 하여 [새 (x) ⇒날개를 가지고 있다(x)]가 성립하는 것을 나타내고 있다.

또한, 예 (4), 예 (5) 에서는 정량기호의 순서에 주의한다. 예 (4) 에서의 술어 논리식 ∀x∃y, 사랑하고 있다(x. y)는 ∀x(임의의 개체변수 x) 에 대하여 ∃y 사랑하고 있다(x， y) 가 성립함을 기술하고 있다. 즉 모든 x에 대하여 “사랑하고 있다(x, y)를 만족하는 y가 존재한다  라는 것을 의미한다.

마지막으로, 예 (5)에서의 술어 논리식 ∃x∀y 사랑하고 있다(x， y)에서는 ∃ y(어떤 개체변수 y) 에 대하여 ∀x 사랑하고 있다(x， y)가 성립함을 기술하고 있다. 즉 모든 x에 대하여 사랑하고 있다(x， y)" 가 되도록 하는 y가 존재한다」라는 것을 의미한다.

이처럼 일차 술어 논리를 이용하여 자연언어 문장의 의미를 표현하기도 하며 지식표현이나 형식적인 의미표현의 한 방법으로써 큰 역할을 담당하고 있다.

4.2.2.11차 논리의 체계적 표현: 가족관계

정의역(domain)을 체계적으로 표현해 보자. 지식 표현에서 정의역은 그냥 우리가 지식을 표현하고자 하는 세계의 일부이다. 가족 관계 또는 친척 관계 정의역에는 “육영수는 박근혜의 어머니이다”나 “박정희는 빅근혜의 아버지이다” 같은 사실들과 “한 사람의 할머니는 그 사람의 부모의 어머니이다” 같은 규칙들이 포함된다.

이러한 정의역의 객체들이 사람임은 명백하다. 두 개의 객체에 관계를 짓는 이항 술어는 두 가지로, 남자와 여자이다. 친족 관계(부모자식, 형제, 결혼 등등)는 단항(속성) 술어 부모, 동기, 형제, 자매, 어린이, 딸, 아들, 배우자, 아내, 남편, 조부모, 손자, 조카, 숙모, 숙부로 표현된다. 함수로는 어머니와 아버지가 있다. 모든 사람은 어머니와 아버지가 각각 정확히 한 명이기 때문이다(적어도 생물학적으로는).

이 함수들과 술어들의 이해를 돕기 위해, 우리가 알고 있는 지식을 여러 함수와 술어, 그리고 다른 기호들로 표현한 예를 제시하겠다. 예를 들어,

한 사람의 어머니는 그 사람의 여자 부모이다:

∀ m, c 어머니(c) = m ⇔ 여자(m) ∧부모 (m, c)

한 여자의 남편은 그 사람의 배우자이다:

∀ w, h 남편(h, w) ⇔ 남자(h) ∧ 배우자(h, w)

부모와 자식은 역관계이다:

∀ p, c 부모(p, c) ⇔ 자식(c, p)

조부모는 부모의 부모이다:

∀g, c 조부모(g, c) ⇔∃p 부모(g, p)

동기(sibling)는 같은 부모의 다른 자식이다:

∀ x, y 동기(x, y) ⇔ x ≠ y∧∃p부모(p, x)) ∧ 부모(p, y)

4.2.3 술어 논리식의 동치관계

아래에 술어 논리식을 변환할 때 잘 사용되는 정량기호를 가진 술어 논리식의 동치관계를 나타낸다.

1. 개체변수의 교체

∀xp(x) 三∀yp(y)

∃xp(x) 三∃yp(y)

1. 부정의 이동

┓∀xp(x) 三∃x[┓p(x)]

┓∃xp(x) 三∀x[┓p(x)]

1. 논리식 q가 개체변수를 포함하지 않는다면

∀x[p(x) Cq] 三∀xp(x) ∧q

∃x[p(x) ∨q] 三∃xp(x) ∨q

∃x[p(x) ∧q] 三∃x[p(x)∧q

∀x[p(x) ∨q] 三∀xp(x) ∨q

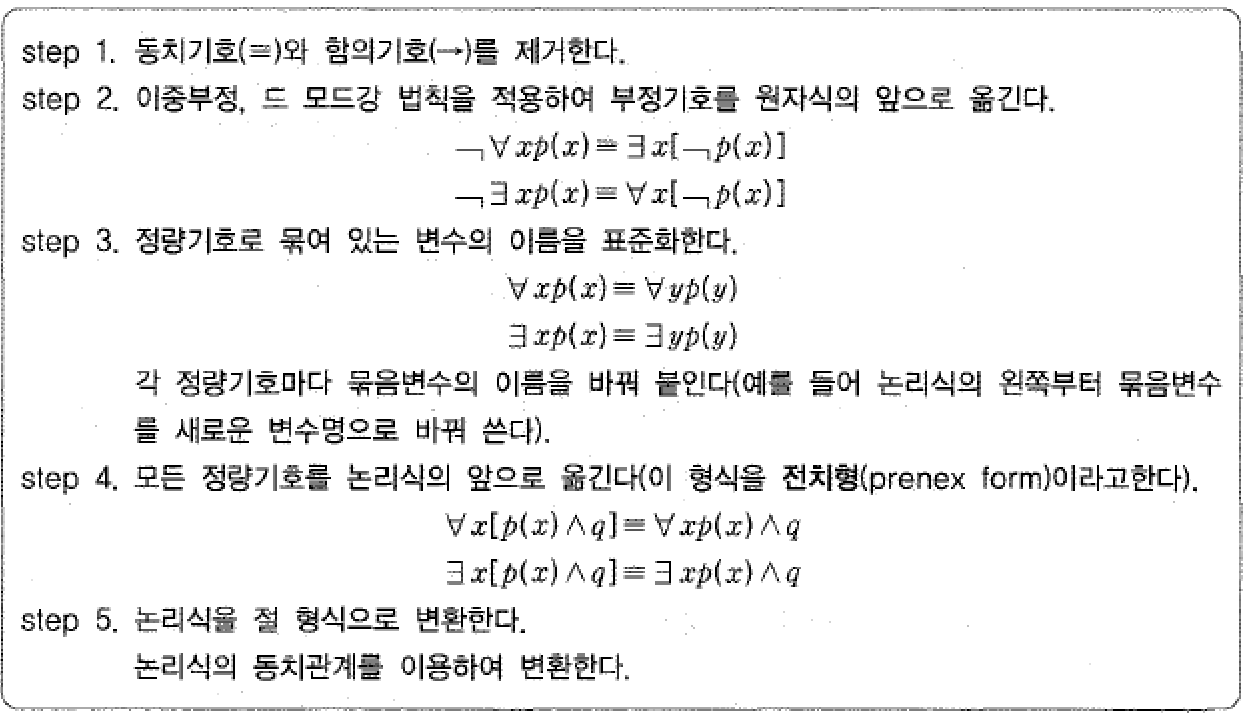
1. 정량기호의 자리바꿈

∀x[p(x) ∧q(x)] 三∀xp(x) ∧∀x q(x)

∃x[p(x) ∨q(x)] 三∃xp(x) ∨∃x q(x)

4.2.4 술어 논리식의 절 형식으로 변환

술어 논리식은 아래에 나타내는 순서에 의해 절 형식으로 변환할 수 있다.



예) 다음 술어 논리식을 절 형식으로 변환하자.

∀x∃y[p(x,y) ∨q(y)] ⇒ ∀x∀z[r(x,z)]

풀이)

단계 1: 함의(⇒)의 제거

三 ┓(∀x∃y[p(x, y) ∨q(y)])∨∀x∀z[r(x, z)]

단계 2: 원자식의 앞으로 부정기호를 이동

三 (∃x∀y[┓p(x, y) ∨q(y)])∨∀x∀z[r(x, z)]

단계 3: 묶음변수의 표준화

三 (∃x1∀x2[┓p(x1, x2) ∨q(x2)])∨∀x3∀x4[r(x3, x4)]

단계 4: 전치형으로 변환

三 ∃x1∀x2∀x3∀x4 [ ┓(p(x1, x2) ∨q(x2))] ∨[r(x3, x4)]

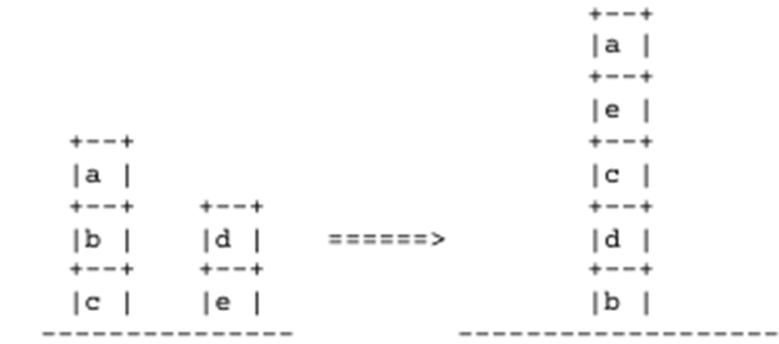
단계 5: 절 형식으로 변환

三 ∃x1∀x2∀x3∀x4[┓(p(x1, x2) ∧┓ q(x2)] ∨[r(x3, x4)]

三 ∃x1∀x2∀x3∀x4[(┓p(x1, x2) ∨ r(x3, x4)) ∧(┓ q(x2) ∨r(x3, x4))]

4.2.5 문제

1. 언어로 표현된 정보를 인공지능에서 요구되는 지식의 표현으로 변환하는 예를 블록 쌓기 놀이의 경우 예로 설명하자. 아래 그림의 좌측은 초기 상태이고 우측은 목표 상태이다.



블록 쌓기 세계에서 두 가지 상태를 본다. 좌측은 현재 상태이고 우측은 원하는 상태이다. 로봇 에이전트가 이 작업을 수행한다면 형식화를 위하여 객체들과 행동들을 정의되어야 한다. 이 문제는 정의역 U={a b c d e}와 아래와 같은 의미들을 갖는 술어{ON, ABOVE, CLEAR, TABLE}로 표현할 수 있다.

* ON: (ON x y) iff x 가 y의 바로 위에 있다. 좌측 세계의 ON은 {(a b)(b c)(d e)}이며 우측 세계는 {(a e)(e c)(c d)(d b)}이다.
* ABOVE: (ABOVE x y) iff x 는 y의 위에 있다. 초기 상태에서 ABOVE는 {(a, b)(b, c)(a c)(d e)}이며 목표 상태의 세계는 {(a, e)(a, c)(a, d)(a, b)(e, c)(e, d)(e, b)(c, d)(c, b)(d, b)}이다.
* CLEAR: (CLEAR x) iff x위에는 아무 것도 없다(x가 제일 위에 있다면). 초기 상태에서는 {{a, d}이며 우측 세계는 {a}이다.
* TABLE: (TABLE x) iff x가 책상 면에 닿아있다. 좌측 세계는 {c, e}이고 우측 세계는 {b}이다.

좌•우측 상태의 블록 세계에서 참인 형식화의 예들이다.

* (ON x y) 암시하다 (ABOVE x y)
* ((ON x y) AND (ABOVE (y z) 암시하다 (ABOVE x z)
* (ABOVE x y) 암시하다 (NOT(존재하다 y (ON y x)))
* (CLEAR x) iff (NOT (EXISTS y (ON x y)))
* (TABLE x) iff(NOT(존재하다 y (ON x y)))

만약 등가성을 사용하지 않는다면, 현재의 기능적 술어들을 기반으로 하는 블록 세계를 표현할 수가 없다. 예를 들면 다른 블록들 맨 위에 한 개의 블록을 놓인(ON) 상황을 표현할 수 없다. 그 이유는

if x is ON y and x is ON z then y is Z의 등가성을 이용할 경우에 표현이 가능하기 때문이다.

또한, 죄측 세계에서 참인 식들이 우측 세계에서도 참값을 가지지는 않으며 그 역도 성립한다. 예를 들어 (ON a b)는 죄측 세계에서는 참이지만, 우측 세계에서는 성립하지 않는다. 더 나아가 좌측과 우측 세계에서의 주장은 모순이 되는 경우도 있다. 예를 들면, 좌측 세계에서 (ABOVE b c)가 참이지만, 우측 세계에서는 (ABOVE c b)가 참이므로 모순이다. 이는 좌측 세계와 우측 세계가 시간에 대하여 동시에 기술되어야 함을 의미한다.

2. 아래의 명제 논리식을 포함하는 지식 기반 KB를 고려하자.

Q⇒ P

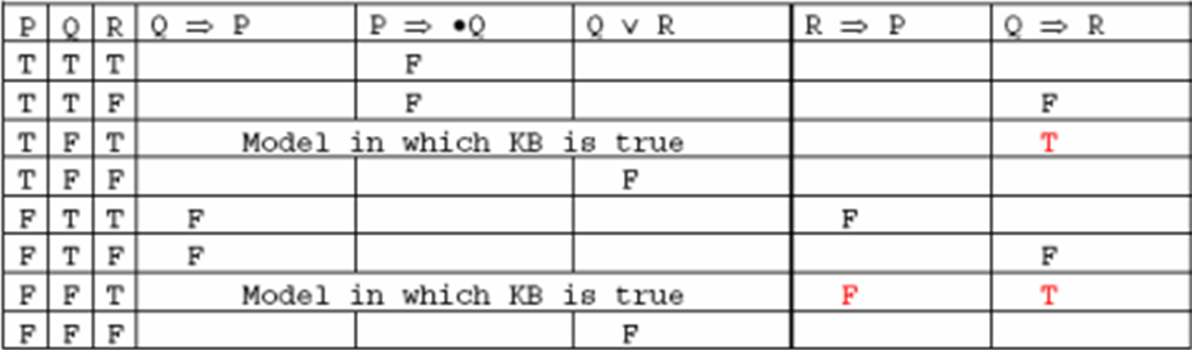
P⇒ ┓Q

Q∨R

연언표준형으로 변환하라.

1. KB의 각 문장이 참인 것을 보이는 진리표를 만들고 KB가 참인 모델을 표시하라.

풀이) 진리표



1. KB는 R을 함축하는가? 답을 선택한 이유를 밝히기 위하여 함축의 정의를 사용하라.

풀이) KB ⊨R. KB가 참(진리표에 표시된)인 모든 모델에서R도 역시 참이다. 따라서 KB는 R을 함축한다.

1. KB는 R⇒P을 함축하는가? 진리표를 만들고 답을 선택한 이유를 밝히기 위하여 함축의 정의를 사용하라.

풀이) KB는 P를 함축하지 않는다. (P, Q, R)이 (F, F, T)일 경우, R ⇒P는 거짓이다.

1. KB는 Q⇒R을 함축하는가? 진리표를 만들고 답을 선택한 이유를 밝히기 위하여 함축의 정의를 사용하라.

풀이) KB⊨는 Q⇒R. KB가 참인 모든 모델에서Q⇒R도 역시 참이다.

2. 아래의 명제 논리 중 명제 A, B, C를 사용하여 D를 증명하라.

1. P⇒(Q⊨R)
2. ┓(Q⊨R)
3. (S∧Q) ⇒P
4. ┓P∧(S⇒┓Q)

풀이) A, B, C 와 ┓D를 논리곱 표준형으로 변환하면 다음과 같다.

A.1 ┓P∨┓Q∨R

A.2 ┓P∨Q∨┓R

B.1 Q∨R

B.2 ┓Q∨┓R

C. ┓S∨┓Q∨P

D.1 P∨S

D.2 P∨Q

증명 과정은 다음과 같다.

1. Q∨┓R(D.2+A.2, factored)
2. Q(E+B.1)
3. ┓R∨(F+B.2)
4. ┓Q∨P(C+ D.1, factored)
5. P(H+D.2,factored)
6. ┓Q∨R(I+A.1)
7. R(I+A.1)
8. 0(K+G)