

Knowledge Representation에 관하여

金永暹* 崔炳旭**

漢陽大學校 電子通信科 博士課程*教授**(工博)

I. 머릿말

인공지능(Artificial Intelligence : 이하 AI로 기술)의 연구는 컴퓨터 사이언스의 한 분야로서 지적 능력을 갖는 컴퓨터 시스템의 개발을 목적으로 한다. 즉, 자연언어의 이해, 학습, 추론, 문제해결 등과 같은 인간의 지적행위를 컴퓨터상에서 실현하고자 하는 것이다.

AI 연구의 일반적인 방법론은 지적 행위의 실현을 가능하게 하는 프로그램을 개발하는 것이다. 그러므로 인간이 행하는 지적 행위를 관찰하여 얻어진 지식을 컴퓨터가 처리할 수 있는 프로그램으로 기술하는 것에 의해 지적 능력을 갖는 컴퓨터의 개발을 시도한다.

한편 인간의 지적 행위의 관찰 대상으로서의 지식은 수학과 물리학등에서 법칙이나 정리로서 연구된 지식뿐만 아니라 아직 수식화, 법칙화되지 않은 경험적인 지식을 모두 포함하기 때문에, 인간이 갖는 지식을 컴퓨터 내에 기술하는 지식 표현이 커다란 문제로 대두되게 된다. 따라서 이러한 지적 컴퓨터 시스템이 갖는 데이터의 기술문제, 즉 컴퓨터 내에 지식을 기억 시키고 그 기억된 지식을 효율적으로 사용하기 위한 데이터 구조를 검토하는 지식 표현이 AI 연구의 근본 문제로 되는 것이다. 지식 표현의 문제는 인간의 지식정보 처리 모델에 관한 연구에 의해 발전되어 왔으며, 인간이 갖는 지식 정보 처리계를 모델링하여, 인간의 지식 처리 과정의 시뮬레이션을 시도하는 것이 인지 과학에서 지식 표현의 검토 방법이다. 인지 과학의 연구에 의해 인간이 갖는 지식의 구조가 기호 처리의 모델로 제안되었으며, 몇개의 모델이 철학, 언어학, 심리학 등의 분야에서 논의되고 있다.

지식의 표현 방식은 표현의 용이함과 이해의 편리를 도모하기 위한 선언적 기술과 추론 과정의 효율 또는 실용적인 시스템의 구현을 목표로 하는 절차형(procedure) 기술로 대별되며, 다수의 표현 방식이 AI 연구자들에 의해 제안되어 있다.

본고에서는 인지 과학을 임의의 이해 시스템내의 지식을 다루는 것으로하여 기억내의 지식 표현의 문제에 접근하며, 최근의 인공지능 및 인지 심리학의 주류를 집약함으로써 해서 Minsky의 지식 프레임 이론과 Quillian의 의미 네트로 대표되는 지식 구조론을 그 기본으로 하고 여기에 논리에 의한 지식 표현과 지식 표현이 중심 과제로 되는 문제해결, 게임을 부가하여 기술한다.¹⁾¹⁾

II. Quillian의 의미 네트

1960년대초 이래로 AI 시스템상에서 정보를 표현하는 한 방식으로 의미 네트가 사용되어 왔다.¹⁾¹⁾ 의미 네트는 임의의 발화 영역(domain of discourse)에서 대상을 할당하는 노드와 이 대상들 사이에 존재하는 관계를 표현하는 링크로 구성된 그래프이다.

인간의 정보처리 모델내의 데이터 표현 형식으로 의미 네트를 도입하는 최초의 시도는 지식 표현의 구조로 검토된 것이 아니고, 언어 이해에 관한 기억 모델을 계산기 상에서 실현하고자 한 것이었다. TLC(teachable language comprehender)라 불리는 이 모델은 Quillian에 의해 작성되었으며, 모델상에서 개개의 단어 또는 개념의 이해 프로세스를 시뮬레이션한 것이다. 그러므로 이 TLC 모델은 인간의 장기 기억내에 기억된 일반적인 의미의 단어군이 공통으로 갖는 의미 구조를 해석하는 것으로 되어, 결과적으로는 인간 기억의 표현 문제에 결부되게 되었다. 예를들어, "기계", "가족", "의자"와 같은 일상 단어의 의미 개념은 장기 기억내에 각각 다른 개념으로 기억되어 있는 것이 아니고, 공통적인 개념하에 기억되어 있는 것으로 가정한다.

TLC 모델을 시스템내에 기억시키기 위해 최초로 기억화 단어(Token이라 부른다.)의 의미를 설명하는 문장을 사전에서 추출한다. 단, 이 때 동일한 철자의 다

의어에 대해서는 번호를 붙여 각각을 별개의 단어로 다룬다. 그리고 추출된 설명문을 그림 1과 같은 표현 도식을 이용하여 의미 네트로 변화한다. 이 도식은 단어의 설명문에 대한 구문 처리, 의미 처리라고 생각해도 무방하지만, 의미에 대해서는 표층 레벨에서의 단어간의 결합(링크) 정도라고 생각할 수 있다.

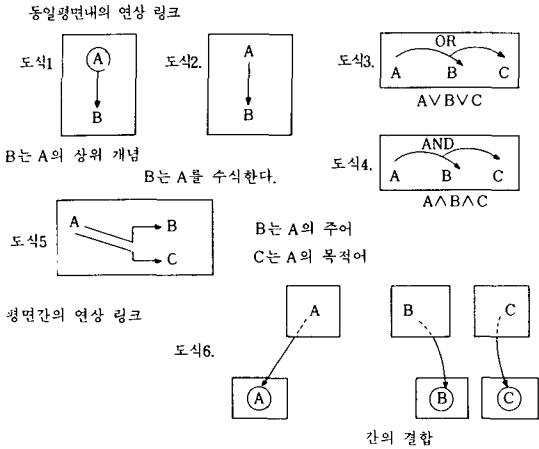


그림 1. Quillian의 표현 도식

또한 도식 1의 "B는 A의 상위 개념이다"와 같은 표현 형식은 의미 구조를 명확하게 기술하며, 효율적인 추론을 가능하게 하는 지식의 유전성을 갖는다. 도식 3, 4와 같은 논리적 관계도 의미 단계를 표현하는 방식으로 생각할 수 있다. 링크는 의미 평면내의 관계와 의미 평면 사이의 관계를 표현하는 두 개의 형식을 갖는다.

이러한 표현도식(의미 평면이라 한다.)에 의해 구성된 의미 네트를 다시 추출된 단어의 설명문 중에 보이는 각 단어에 대해서도 적용한다. 그러나 만일 어휘에 대해서 의미 항목이 다수일 때는 적당한 곳에서 의미 평면의 구성을 중지한다. 그림 2는 영어 단어 PLANT에 대한 의미 평면을 보인 것이다. PLANT의 각각의 설명문에 대해서 의미 평면을 구성하며, Plant, Plant 2, Plant 3의 논리합으로 Plant에 대한 전체의 구조를 표현한다. 결국 어휘에 대한 설명문을 그림 1과 같은 결합 형식으로 변환시킴으로서 문에 대한 의미 처리를 하는 것이다.

실험 목적에 준하는 영어의 단어를 사전에서 선택하여 의미 네트로 구성한 것을 인간의 기억에 대한 컴퓨터 모델이라 생각한다. 그리고 이 모델에 기억된 단어

에서 임의 한쌍을, 예를들면(Plant, Food)를 선택하여 이들 두 단어의 의미를 대조, 비교하여, 의미의 교차가 발견되면 그 의미의 교차에 기초하여 문을 생성한다. 모델의 평가는 실제로 인간이 이들 두 단어의 비교를 행하여 생성한 문과 컴퓨터에 의한 출력력을 비교하는 것에 의해 행한다. 이러한 방법은 심리학적 기

- PLANT. 1. Living structure which is not an animal, frequently with leaves, getting its food from air, water, earth. (植物)
2. Apparatus for any process in industry (裝置)
3. Put (seed, plant, etc.) in earth for growth. (生育)

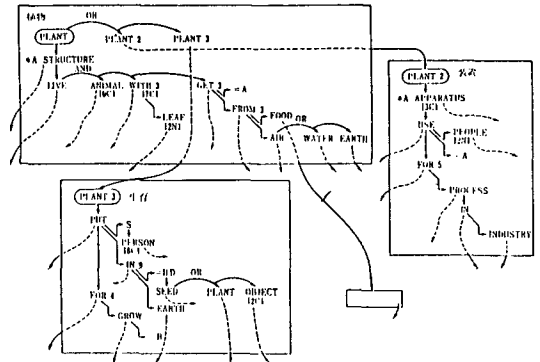


그림 2. Plant에 대한 표현 도식

초로서 장기 기억의 구조를 명확하게 규명하고자 한 Brown, McNeil 등의 연상기억에 대한 실험적연구-피 실험자에게 설명문을 부여하고, 그 설명문이 의미하는 단어를 연상하게 하는 실험-에 대응한다.

Quillian의 모델은 영어 단어의 의미적인 관련을 표현 도식에 의해 의미 네트로 구성하기 때문에, 실제 계산기상에 구현하고자 할 때 기억 용량이 커다란 문제로 대두된다. 예를들어 약 850개의 기초 영어 단어가 기억 모델에 저장되어 있을 때 약 360,000개 조합의 의미 교차가 발생하며, 컴퓨터 시뮬레이션은 기억 용량상의 제약 때문에 실현되지 않는다. 그러나 이러한 기억 용량상의 제약에도 불구하고, Quillian의 의미 네트는 의미에 대한 컴퓨터 모델에 새로운 기법을 부여하며, 인간 기억내의 데이터 표현 기초로 되었다.

이상의 Quillian의 모델로 생각한 의미 네트의 의의

를 종합하면 다음과 같다.

- 1) 종래의 언어 처리가 구문 해석을 중심으로 한 것에 비해서 인간의 언어 활동을 기억내에 표현한다는 입장에 접근했다.
- 2) 의미 표현에 의미 네트를 도입하여, 사실의 관계 구조 특히, 상위 개념간의 의미 관계(의미의 교차)에서 의미 처리를 시도한다.
- 3) 실제 계산기에 의해 인간 기억의 시뮬레이션을 시도하여, 의미 이해에 대한 개념을 명확하게 한다.

III. FRAME

프레임 이론은 Minsky가 이해 시스템에 대한 지식 표현의 기초로 제안하여, 인지 시스템의 데이터 구조로 검토한 것이다.

인간은 무엇인가를 인식하고, 행할 때 직접 대상만이 아니라 여기에 관련된 다수의 구조화 정보를 이용하여 대응하는 상황을 이해한다고 생각된다. 한편, 이와 같은 구조화 정보를 프레임이라 부르며, Minsky는 인간의 인지과정에서 이러한 데이터 구조의 역할에 주목한다.¹⁰⁾

프레임의 구조는 노드(절감)와 링크로 이루어진 네트이며, 어떤 대상이 가질 수 있는 속성들의 리스트라고 할 수 있다. 프레임은 몇 층의 레벨을 갖는 계층형 네트이다. 그러므로 의미 네트와 같은 평면적인 구조로는 용이하게 표현할 수 없는 지식의 표현에 특히 유용하게 사용되어진다. 프레임 구조의 최상층은 하나의 상황에 대하여 일의적으로 결정된 네임 라벨-일종의 표지어-로 되어 있으며, 다음 계층에는 최상층의 노드를 구성하는 요소들로 구성된 서브프레임이 있다.

이 서브프레임의 요소에는 특정의 데이터나 사실을 할당할 수 있는 엠퍼티 노드가 존재하며, 각각의 노드는 슬롯이라 불리운다. 한편 각각의 슬롯에는 입력 조건이 부가되어 조건을 만족하는 데이터나 사실만이 이 슬롯에 입력된다. 슬롯에 데이터를 적용시킬 때의 조건을 슬롯 할당 조건이라 한다.¹¹⁾

예를들어, 자동차를 대상으로 하는 프레임을 구성할 때는 프레임의 네임 라벨로서 자동차가 할당되며, 생산자, 출고년도, 허가번호, 차체, 엔진 등이 서브프레임으로 기술된다. 프레임의 기본 구조는 그림 3과 같다. 프레임의 중단에는 구체적인 수치가 할당되며, 불변의 특징치 이외는 데이터가 경신되어지므로 중단 슬롯은 상식치라고 불리운다. 상식치(default value)는 보통 초기치로 터미널 슬롯에 할당되며, 상황에 따라 변하는 값에 변수로 간주할 수도 있다. 그림 3

에서의 상식치 할당 조건은 다음과 같이 기술될 수 있다. “만일 월 일때는 그 값은 정수로 1에서 12까지이다.” 이는 월에 대해서 생각할 때는 정수라는 조건을 만족하고 동시에 1에서 12사이의 어떤 수를 슬롯에 할당 한다는 의미로 해석되며, 슬롯에 부여된 조건은 데이터의 적절 여부를 조사하여 해당 데이터를 추출하는 기능을 갖는다.¹¹⁾

상식치 할당 조건과 같은-“만일...하면, ...하라”-프로덕션 타입의 지식 기술을 절차형(procedure) 지식이라 한다. 따라서 프레임은 사실 또는 사실과 사실사이의 관계에 의해 표현되는 선언적 지식과 절차형의 지식을 동시에 그 표현 구조로 갖는다.

프레임이 갖는 또 하나의 중요한 특징은 지식의 유전성에 관한 것이다. 지식의 유전성은 상위의 성질(개념)이 하위에 전해진다는 의미이다. 예를들어 생물이 라는 지식의 하위에는 동물과 식물이 각각 결부되어 있으며, 그 결과 상위의 생물의 성질은 하위의 동물과 식물에 전해진다고 생각한다. 지식의 유전성을 이용하면, 하나의 프레임에 관계하는 상위프레임을 호출해서 그 성질을 현재의 하위 프레임의 성질로서 사용할수 있게되며, 개개의 프레임에 여러가지 지식을 기술하더라도 상위와 하위와의 관계에서 필요한 지식을 하위프레임에 보출할 수 있다. 따라서 지식 표현이 경제적으로 된다.

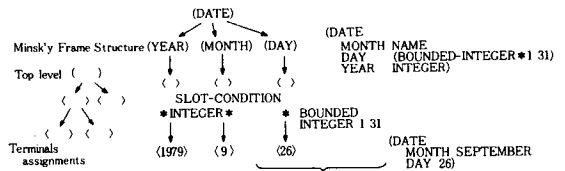


그림 3. 프레임의 기본 구조와 그 구체례

프레임의 슬롯에 부가된 절차형 지식을 실행하여 조건에 부합되는 데이터를 추출하는 것에 의해 새로운 상황에 대응하는 동적인 지식 표현 형식을 갖는다. 이러한 프로세스는 절차를 이용한 추론에도 존재하며, Demon이라 부른다. 프레임의 상식치를 변경하는 동적 지식 표현으로 Demon은 통상 데이터 추출, 소거, 부가등의 프로세스로 구성된다. 그림 4는 프레임에서의 Demon의 이용을 개략적으로 보인 것이다. a)는 상위 프레임 내의 IF-NEEDED Demon을 사용하여, 하위 프레임 내의 데이터를 추출하는 경우이며, b)는 IF-

REMOVED Demon 을 이용해서 하위 프레임 내의 데이터를 경신하는 경우를 보인다.

타로 간주할 수 있으며, 지식 추론을 수행하기 위해서는 논리식을 처리하는 프로그램을 시스템상에 실현해야 한다. 그림 5는 논리식을 이용하는 지식 처리 시스템의 개략적인 구성도 이다.¹⁾

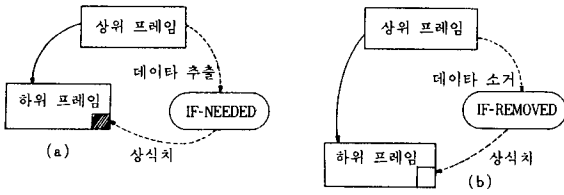


그림 4. 상위-하위 프레임간의 관계

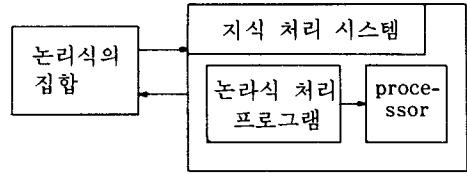


그림 5. 논리식을 이용한 지식 처리 시스템

프레임에서는 지식의 유전성, 상식치 할당 그리고 Demon의 이용에 의해 지식 데이터가 전달되며, 또한 슬롯 내에 데이터가 부가된다. 이는 최소 데이터의 간섭에 의해 각종의 상황에 대응할 수 있는 기초가 프레임에 존재 한다는 의미이며, 프레임이 지식 정보 처리의 데이터 표현 구조로 이용되게된 연유라고 생각할 수 있다.

술어 논리식은 상호간에 독립되어 있으며, 표현형식이 일정하다. 그러므로 술어 논리식은 modality란 잇점을 갖는다. 임의의 술어 논리식의 부가나 삭제, 그리고 시스템내의 다른 지식 데이터에 영향을 미치지 않고 지식 베이스를 단순히 수정할 수도 있게 되는 것이다.

지식 표현 요소로서 프레임이 갖는 특징을 요약하면 다음과 같다.

일개 술어 논리를 이용한 추론 방식으로 Robinson의¹⁶⁾ 도출 원리(resolution principle)가 유명하다. 도출 원리는 일반적으로 논리식을 처리하는 프로그램으로 기술되며, 도출에 의하여 얻어진 새로운 논리식은 기존 논리식의 집합에 부가된다. 다음에 도출 원리를 이용하는 일개 술어 논리의 추론례를 보인다.

- 1) 계층별 네트
- 2) 프레임 내부는 링크와 슬롯에 의해 구성된다.
- 3) 상식치의 설정
- 4) 절차 부가에 의한 조건기술
- 5) 상새 기술을 위한 절차의 부가

Minsky의 프레임 이론은 지적 정보 처리 시스템이 갖는 데이터의 성질을 일반적으로 기술한 것으로 구체적 표현 언어의 개발은 아니었다. 구체적인 데이터 구조는 Bobrow와 Winograd¹⁷⁾에 의해 기술되었으며, 이후 프레임을 이용한 구체적인 표현 언어로 KRL, FRL, UNITS¹⁸⁾ 등이 개발되었다.

예 1) “윤택이는 회사원이고, 병준이와 혜경이는 학생이다. 그리고 병준이와 윤택이는 남자이고 혜경이는 여자이다. 한편 학생은 할인권을 갖는 것으로 한다.

IV. 술어논리

술어논리(predicate logic)는 인간의 사고를 형식적으로 다루기 위하여 연구되어 왔다. 인간이 갖는 지식과 정보를 표현하기 위하여, 고개논리(higher order logic)와 양상 논리(modal logic), 내포 논리(intentional logic), 애매논리(fuzzy logic) 등 각종의 논리가 고려되고 있지만, 컴퓨터에서의 처리를 대상으로 한 경우에는 일개 술어 논리가 일반적으로 사용된다.¹⁷⁾

그렇다면, 할인권을 가진 여자는 누구인가?”

예 1)은 일개 술어 논리를 이용하여 다음과 같이 표현할 수 있다!¹⁴⁾ 단, 여기에서 학생, 회사원, 여자, 남자를 표현하는 술어로 각각 Student, Employee, Female, Male를 할당하며, “할인권을 갖다”를 표현하는 술어는 Discount로 한다.

술어 논리식은 전형적인 선언적 표현이며, 개개의 사실(fact)과 그들 사이의 관계는 모두 논리식으로 기술된다. 따라서 지식 처리 시스템 상에서 논리식은 데이

- (1) 윤택이는 회사원이다. Employee (윤택)
- (2) 병준이는 학생이다. Student (병준)
- (3) 윤택이는 남자이다. Male (윤택)
- (4) 병준이는 남자이다. Male (병준)
- (5) 혜경이는 여자이다. Female (혜경)
- (6) 혜경이는 학생이다. Student (혜경)
- (7) 학생은 할인권을 갖는다.

$$\forall x (\text{Student}(x) \rightarrow \text{Discount}(x))$$

→ ~Student (x) ∨ Discount (x)

(8) 할인권을 가지고 있는 여자는 누구인가?

⇒ z (Female (z) ∧ Discount (z))

(8') 부정형 ~Female (z) ∨ ~Discount (z)

영문의 소문자 x, y는 변수형을 의미하며, (1)-(6)의 논리식은 구체적 사실의 표현이며, (7)의 논리식은 사실사이의 관계를 표현한다. 한편 (8)식은 의문문에 대응하는 논리식이며, 도출 원리는 그 자체가 부정 증명(refutation)이므로 의문문의 부정형인(8')을 추론에 이용한다. 논리식은 예문과 1대1로 대응하지 않으며, 개개의 사실과 관계가 최소 단위로 표현된다. 그리고 예문의 표현이 바뀐다 해도 의미상의 변화가 없다면 논리식은 변화하지 않는다. 또한, 새로운 정보가 추가되면 기존 논리식의 집합에 새로운 논리식을 부가하며, 정보의 삭제가 있을 때는 논리식을 제거한다. 예를들어, 예 1의 문에서 "혜경이는 여자이다"라는 문을 "혜경이는 아름다운 여자이다"라는 문으로 바꾼다면, (5)식에 "아름답다"를 표현하는 술어 Beautiful을 이용한 논리식 Beautiful이 부가되며, "병준이는 학생이다"라는 사실이 삭제되면 (2)의 논리식이 제거된다.

예 1의 의문에 대하여 도출 원리를 이용한 추론 과정을 그래프로 보인것이 그림 6이다. 그림에서 각 Branch에 할당된 번호는 논리식의 번호를 의미하며, Branch에 할당된 논리식과 상위 노드의 식에서 하위 노드의 논리식이 도출에 의해서 얻어진다.

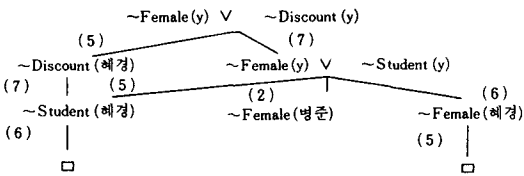


그림 6. 도출 원리에 의한 추론의 예

예 1은 매우 간단한 경우이지만 논리식의 집합이 복잡한 경우에도 동일한 방식으로 도출 원리를 사용하면 증명이 보장된다. 또한 도출 원리를 효율적으로 적용하기 위하여, 논리식의 적용 순서와 도출 과정에서의 제한 조건을 기술하는 다수의 전략(strategy)이 제안되어 있다.

술어 논리에는 공리와 추론 규칙이 존재한다. 그러나 이러한 공리나 추론 규칙은 지식 표현 자체 보다 지식 처리에 더 밀접하게 관련되어 있다. 예 1의 경우에도

추론 규칙은 몇개의 부여된 논리식에 적용되어 새로운 논리식을 생성하고 있다. PROLOG는 논리를 베이스로한 언어로 지식 표현과 문제 해결등에 효과적인 tool이라 생각되며, 주로 유럽과 일본을 중심으로 연구되고 있다. 또한 LISP의 (s-expression)도 논리식을 표현하는데 유용한 데이터 구조를 가지고 있다.

V. 문제해결

인간은 어떤 문제를 해결하고자할 때, 먼저 문제를 인식하고 그 해를 구하기 위한 어떤 수단, 방법을 생각하며, 가능한 좀더 효과적인 해결 방법을 모색할 것이다. 그러므로, 컴퓨터의 빠른 정보 처리 능력을 이용하여 어떤 문제를 해결할 경우에도 인간이 갖는 문제 해결 프로세스와 마찬가지로, 먼저 컴퓨터가 처리할 수 있는 형태로 문제를 기술하고, 이를 처리하여 문제를 해결할 수 있는 어떤 수단과 방법을 컴퓨터에 부여해 주어야 한다.

일반적인 문제 해결 시스템은 개략적으로, 문제를 기술(표현)한 데이터 베이스와 이 데이터 베이스를 조작하는 오퍼레이터, 그리고 효율적이고 신속한 문제 해결을 위한 제어 전략(strategy)의 세 부분으로 구성된다. 한편 문제 해결 능력은 제어 전략뿐만 아니라 데이터 베이스와 오퍼레이터를 선택하는데 따라 크게 좌우되므로 이들을 적절하게 선택하는 것이 중요한 문제로 되며, 이러한 선택의 문제를 AI에서는 표현 문제(representation)한다. 그러나 아직 일반적인 최적 표현 방법은 인간의 경험에 의존해서 선택되고 있다.

문제를 표현하는 방법은 크게 상태 공간 표현¹⁶⁾(state space)과 문제 축소 표현^{16, 22)}(problem reduction)의 두 가지로 나눌 수 있다. 상태 공간 표현에서는 문제를 상태(state)와 오퍼레이터의 기본 요소로 기술한다. 상태는 문제를 해결하는 과정의 제 단계를 의미하는 데이터 구조라 할 수 있으며, 오퍼레이터는 한 상태에서 다른 상태로의 변환에 사용된다. 오퍼레이터를 적용하여 얻을 수 있는 모든 상태들의 집합을 상태 공간이라 한다. 이러한 상태 공간은 상태를 의미하는 노드와 오퍼레이터를 의미하는 아-크를 사용하여 directed 그래프로 표현할 수 있으며, 이를 상태 공간 그래프라 한다. 그림 7은 상태 공간 그래프의 일례이다. 그래프상에서 문제를 해결한다고 하는 것은 초기상태를 나타내는 스타트 노드에서 목적 상태(goal)을 나타내는 목적 노드로의 경로(path)를 그래프상에서 탐색하는 것이 되며, 제어 전략은 경로를 신속하게 발견하기 위한 방법이라 할 수 있다.

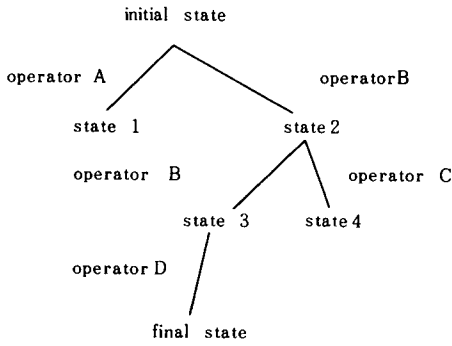


그림 7. 상태 공간 그래프

문제축소 표현은 초기 문제가 부여되면 이를 보다 해결하기 쉬운 하나 이상의 subproblem으로 분할하고, 다시 이를 반복하여 하위의 subproblem으로 분할하여 즉각적으로 해를 구할 수 있는 원시 문제(primitive)로 분할하는 방법이다.

문제축소 표현은 초기 문제, 원시 문제, 분할 오퍼레이터로 구성되며, 상태 공간 표현과 유사하게 그래프로 표현할 수 있다. 문제축소 표현에 대한 그래프를 and/or 그래프라 하며, 부여된 문제를 노드로 나타내고 분할 오퍼레이터를 적용하여 얻어진 subproblem을 하위 노드로 나타내어 방향성 아-크(directed)로 연결하여 구성한다. 또한 하나의 오퍼레이터에 의하여 생성된 모든 subproblem의 아-크는 horizontal 라인으로 연결시키는 것으로 한다. 그림 8은 and/or 그래프의 간단한 예이며, n=3인 경우의 하노이 퍼즐에 대해 표현한 것이다.

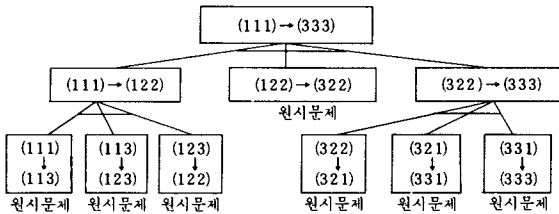


그림 8. Hanoi Tower puzzle을 원시문제로 치환한 and/or 그래프

VI. 탐색(search)

문제 해결 시스템의 제어 전략은 문제를 표현한 그래프에서의 탐색 방식이라 할 수 있다. 탐색 방식은 문제 형식과 특정 정보의 이용 여부에 따라 크게 4가지

로 나눌 수 있다. 특정 정보를 이용하지 않는 탐색 방식(-blind 탐색¹¹⁰⁾이라고도 한다.)으로는 breath first search, depth first search 등이 있다. 많은 경우에 있어서 문제에 대한 어떤 특정 정보를 이용하면 탐색해야 할 경우의 수를 대폭 감소시킬 수 있으며, 이러한 탐색 방식은 heuristic 탐색이라 한다. heuristic 정보를 이용하는 탐색 방식에는 다음과 같은 즉, 1) expand 할 때 각 노드의 expand 순서를 결정하는 방식 2) expand 할 때 부분적으로만 expand 하는 방식 3) expand 할 필요가 없는 노드를 제거(pruning)하여 탐색의 경우를 줄이는 방식 등이 있으며, 일반적으로 이러한 정보는 노드에 대한 평가(evaluation) 함수로 주어진다.

평가 함수 f*는 문제의 유형에 따라 각각 다르게 정의되어진다. 문제의 해가 다수이고 또 그 코스트(cost)가 각각 다를 경우에 가장 최소 코스트인 것을 문제의 해로 하고자 할 때 평가 함수는 $f^* = g^* + h^*$ 로 정의된다.^{111,112)} 이 때 g^* 는 초기 노드로부터 지정 노드까지의 코스트를 의미하며, h^* 는 지정 노드로부터 목적 노드까지의 코스트를 의미한다. 한편 위의 평가 함수를 사용하여 문제를 해결하고자 할 경우 엄청난 시간과 메모리가 필요할 때는, 탐색 노력과 구하고자 하는 해의 최적 정도에 주안점을 두는 동적(dynamic) weighting 방식과 bandwidth 탐색 방식을 적용한다. 동적 weighting 방식은 평가 함수를 $f^*(n) = g^*(n) + w(n)h^*(n)$ 로 정의하여 weight 함수 w가 가변적으로 각 노드에 따라 상이한 값을 갖도록 하는 것이다. 한편, 해답이 하나이거나 여러 개의 해를 갖더라도 각각이 거의 유사한 경우 탐색 노력만을 최소화하는 방식도 있다. 이는 평가 함수를 $f^* = (1 - \omega)g^* + \omega h^*$ 로 정의하여 g^* 와 h^* 에 대한 상대적인 비중을 고려한 방식으로 동적 weighting 방식과는 달리 ω 는 0와 1사이의 값을 가지는 상수가 할당된다.

VII. 게임(Game)

컴퓨터에 의한 대부분의 게임들 즉 체스나 장기등은 여러 면에서 서로 공통된 특징을 가지고 있다. 예컨대 각각의 게임에는 규칙이 있으며 또한 자기 자기뿐만 아니라 상대방의 상황과 반응까지도 고려해야 되는 것이다.

한편 게임에 대한 모든 가능한 경우를 문제해결 시스템에서와 유사하게 게임 트리¹¹³⁾ 표현할 수 있다. 즉 root 노드로 게임의 초기 상태-시작 상태-를 나타내고, 한 노드에 대한 successor들로 가능한 모든 경우를 표현하며 종단 노드에 승, 패, 비김 등의 게임 결

과를 표현한다. 실제 이러한 게임 트리는 and/or그래프와 유사하게 표현되며, 자신의 움직임은 or 노드로 나타내고 상대방의 움직임은 모두 다 고려해야 하기 때문에 and노드로 나타낸다. 따라서 게임 트리는 and와 or노드가 연이어 나타나는 and/or그래프라 볼 수 있으며, 컴퓨터에 의한 게임은 이러한 게임 트리상에서 이기는 경우만을 탐색하는 것으로 생각할 수 있다.

게임트리를 탐색하는 가장 기본적인 방법은 minimax^[20] 방법이다. minimax방법은 승, 패, 비김 등을 나타내는 종단 노드로부터, 역방향으로 or successor를 갖는 비종단 노드는 successor의 최대치를, and successor를 갖는 비종단 노드는 최소치를 갖는 것으로 하여, or 노드로 나타낸 자신의 successor들 중에서 max치를 갖는 successor를 택하는 방법이다. 게임의 순서에 의해 상대방도 마찬가지로 자신의 게임 트리에서 minimax방식을 적용한다. 또한 자신과 상대방의 게임 트리는 and와 or, 그리고 승과 패만이 바뀌어 있을 뿐이므로 이들을 결합하여 한 쪽은 양수치로 다른 쪽은 음수치를 부여한 다음 음수를 취한 값이 최대인 successor노드를 선택하는 negmax방식도 있다. 그러나 이는 실질적으로는 minimax방법과 등가한 방법이다.

Minimax, negmax 방법에서는 완전한 게임트리가 이미 생성된 경우를 가정하고 있으나 실제의 경우 이는 거의 불가능하다. 따라서 현재의 위치로부터 적절한 부분까지의 트리를 구성하고 이 부분 트리에서 얻어진 정보를 바탕으로 탐색을 계속하여 새로운 위치를 선정한 다음 다시 새로운 부분 트리를 생성하는 방식을 사용한다. 한편 일단 부분 트리가 생성되면 이 트리의 종단 노드인 tip노드의 값을 평가하는 수단이 요구되는데 이를 정적 평가 함수라고 한다. tip노드의 평가치로부터 minimax나 negmax방식의 적용에 의해 새로운 위치를 선정하게 된다. 그러나 이러한 방식은 생성된 트리의 모든 노드에 대한 평가를 수행해야 하므로 비효율적이다. Alpha-Beta pruning은 이러한 비효율적인 평가과정을 개선한 것이며, 임의의 or나 and노드의 successor일부에 대한 평가 과정을 생략한다.

게임 트리의 경우에도 문제 해결 시스템에서와 같이 heuristic정보를 이용 하면 보다 빠른 탐색을 수행할 수 있다.^{[21][22]} 하지만 게임 트리의 탐색에서는 heuristic정보를 탐색 범위와 위치 선정에 중점을 두어 이용하고 있는 점이 문제 해결의 경우와는 약간 다른 점이다.

Ⅷ. 맺음 말

이제까지의 기술 입장은 지식의 표현론이라는 인공지능의 새로운 테마를 개략적으로 기술하는 것이었다.

지식의 표현 방식에는 본고에서 기술된 의미 네트, 프레임, 술어 논리 이외에도, Newell이 제안한 production시스템, 그리고 일개 술어 논리의 결점을 보완해서 효과적인 처리 방식과 실용 시스템을 목표로 개발된 인공지능 언어, 의미 네트의 변형인 Schank의 개념 의존 이론^[23] 등 다수의 표현 방식이 제안되어 있다.

또한 부가적으로 기술한 문제 해결이나 탐색에 있어서도 Newell, Simon등의 일반 문제 해결 시스템^[24] 등 여러 연구 성과가 있지만 지면 관계상 모두 게재하지 못하였다. 흥미있는 독자는 참고문헌을 참고하기 바란다.

참고 문헌

- [1] A. Barr & E. Feigenbaum, *The hand-book of Artificial Intelligence*, vol. 1 William Kaufmann, Inc., 1981.
- [2] Nilsson, N.J., *Principles of Artificial Intelligence*, Palo Alto, Calif., Tioga, 1980.
- [3] Winston, P.H., *Artificial Intelligence*, Addison-Wesley, 1977.
- [4] Allwood, J., Anderson, L. & Dahl, J., *Logic in Linguistic*, Cambridge University Press, 1977.
- [5] A. Bundy, *Artificial Intelligence: An introductory course*, Edinburgh University Press, 1978.
- [6] Robinson, J.A., "A machine-oriented logic based on the resolution principle," *JACM*, 12, pp. 23-41, 1965.
- [7] R. Davis & D.B. Lenat, *Knowledge-Based systems in Artificial Intelligence*, McGraw-Hill, 1982.
- [8] 田中幸吉, "知識工學" 朝倉書店, 1984.
- [9] 齋藤正男 溝口文雄, "知的報報處理の設計", Corona Pub. Co., 1982.
- [10] Anderson, J.R., "Language, Memory and Thought," Hillsdale, N.J., Erlbaum, 1976.
- [11] Bobrow, D.G. & Collins, A.M., *Representation and Understanding*, Academic Press, 1975.
- [12] Finder, N.V., *Associative Networks: Representation and Use of Knowledge by Computer*, Academic Press, 1979.

- [13] Schank, R.C & Abelson, R.P., "Scripts, Plans, Goals and Understanding" *LEA*, 1977.
- [14] Just, M.A & Carpenter, P.A., *Cognitive Process in Comprehension*, Hillsdale, N.J., Erlbaum, 1977.
- [15] Waterman, D.A & Hayes-Roth, F., *Pattern-Directed Inference System*, Academic Press, 1978.
- [16] Nilsson, N.J., *Problem-solving methods in artificial intelligence*, McGraw-Hill, 1971.
- [17] Manna, Z., *Introduction to the mathematical theory of computation*, McGraw-Hill, 1973.
- [18] Anderson, J & Bower, G., *Human associative memory*, Washington D.C., 1973.
- [19] Norman, D.A., Rumelhart, D.E. & the LNR Research Group, "Explorations in cognition," 1975.
- [20] Winston, P.H. (Ed.), "The psychology of computer vision-Minsky: A framework for representing knowledge" pp. 211-277 McGraw-Hill, 1975.
- [21] Brachman, R.J., "A structure paradigm for representing knowledge" Rep, no. 3605, Bolt Beranek & Newman Inc. Cambridge, 1978.
- [22] Jackson, P.C., "Introduction to artificial intelligence" Petrocelli, 1974.
- [23] Feigenbaum E.A., and Feldman, J., *Computers and thought*, McGraw-Hill, 1963.
- [24] B. Meltzer and D. Michie (Eds.), *Machine intelligence 6*, American Elsevier, 1971.
- [25] Collins, N.L. & Michie, D., *Machine intelligence 1*, American Elsevier, 1967.
- [26] Gelperin, D., "On the optimality of A*" *Artificial Intelligence 8*, pp. 69-76, 1977.
- [27] Hall, P.A.V., "Branch -and-bound and beyond", *IJCAI*, 2 pp. 641-650, 1971.
- [28] Aho, A.V. & Hopcroft, J.E. and Ullman, J. D., *The design and analysis of computer algorithms*, Addison-Wesley, 1974.
- [29] Knuth, D.E. and Moore, R.W., "An analysis of alpha-beta pruning" *Artificial Intelligence 6*, pp. 293-326, 1975.
- [30] Berliner, H.J., "A chronology of computer chess and its literature" *Artificial Intelligence 10*, pp. 201-214, 1978.
- [31] Berliner, H.J., "Search and knowledge," *IJCAI*, pp. 975-979, 1977.
- [32] Bobrow, D.G and Winograd, T.G., "An overview of KRL, a knowledge representation language," *Cognitive Science*, 1977.
- [33] Rosch, E. and Lloyd, B.B. (Eds.), "Cognition and Categorization," Hillsdale, N.J., Erlbaum, 1978.
- [34] Newell, A. & Simon, H.A., *Human problem solving*, Englewood Cliffs, N.J., Prentice-Hall *

♣ 用語解説 ♣

레귤러 체인 (regular chain)

普通체인을 말한다. 체인을 구성하는 모든 小賣店이 하나의 製造會社를 本社로 하고 支店으로 결합하는 형태를 말한다. 小賣店의 經營에는 大資本大規模의 經營에 의한 利點이 가해지는 동시에 製造會社로서도 中間商人의 손에 떨어지는 이윤이 없으므로 利益이 많아진다.

基本파일 (master file)

마스터 파일이라고도 부른다.

頭腦機關 (Think Tank)

일반적으로 싱크 탱크라고 부른다.

頭腦集團 (Think Tank)

일반적으로 싱크 탱크라고 부른다.