

지식? 네가 뭘 알아!

기호처리로 출발한 인공지능은 지식이라는 더욱 추상적인 개념을 처리하는 시스템으로 발전한다. 지식이란 무엇이며 컴퓨터는 지식을 어떻게 다루는지에 대해 알아보고, 가장 성공한 지식처리 인공지능으로써 전문가 시스템에 대해 알아보자.

지식과 지식의 표현

컴퓨터 같은 정보처리 장치는 데이터를 가공해서 정보를 생성한다. 즉, 컴퓨터에 전달하는 재료를 데이터라고 부르고 처리된 결과를 정보라고 부른다. 이렇게 생성된 정보가 다른 처리를 위해 입력으로 전달된다면 데이터가 되고 다시 정보를 생성한다. 가공된 데이터가 정보인 셈인데, 정보가 쌓여 체계를 갖추면 지식이 되고, 더 높은 수준으로 추상화되어 이론과 원리로 발전한다.

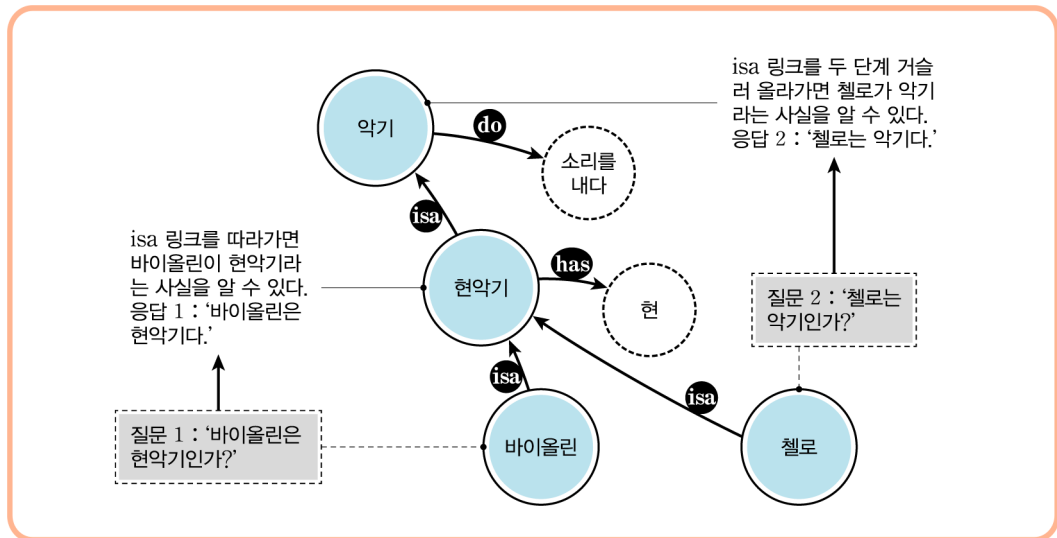
컴퓨터가 단순한 계산장치를 넘어 지능적인 장치가 되기 위해서는 지식을 다룰 수 있어야 한다. 학문적으로 지식을 엄밀하게 정의하는 건 어려운 일이다. 그 어려운 일은 학자들에게 맡기고, 우리는 간단하게 “알고 있는 것으로써 체계적으로 잘 조직된 정보의 덩어리”로 생각하면 큰 무리가 없겠다.

컴퓨터가 처리할 데이터를 전달하기 위해서는 컴퓨터가 이해하고 다룰 수 있는 형태로 표현할 방법이 있어야 한다. 단순한 데이터의 경우 정수, 실수 등의 수치 데이터는 이진수의 형태로, 문자 데이터는 ASCII, 유니코드 등의 형태로, 그림은 각 픽셀의 색상을 담은 비트들의 조합 형태로 표현한다.

마찬가지로 지식을 다루려면 지식을 컴퓨터가 처리할 수 있는 형태로 표현할 수 있어야 한다. 초기 인공지능 연구자들은 지식처리의 중요성을 인식한 후 지식표현에 관해 많은 연구를 수행했다. 기호처리에 기초한 규칙, 의미망, 스크립트, 프레임 등 다양한 지식표현 개념들이 등장했으며, 비록 그것이 인간의 지능을 가진 시스템 개발로 이어지지는 못했으나, 온톨로지, 객체지향 등의 개념으로 발전했으며, 특히 상업적으로 가장 성공한 인공지능 시스템인 전문가 시스템 개발의 든든한 기초가 되었다. 오늘날 사용되는 대부분의 지능 시스템은 넓은 범위에서 전문가 시스템이다. 알파Go는 딥러닝과 강화학습을 응용한 몬테카를로 트리 탐색 시스템이자 바둑을 두는 전문가 시스템이다.

의미망

의미망은 개념들의 관계를 네트워크의 형태로 표현하는 지식표현 방법이다. 다음은 “악기”에 대한 지식을 의미망으로 표현한 것이다. 의미망에서 개념은 노드라 부르는 원으로, 관계는 링크라 부르는 화살표를 사용해서 표현된다. 아래와 같은 의미망 통해서 “악기란 소리를 내는 것이고, 현악기는 악기의 일종이며 현을 가지고 있고, 바이올린과 첼로는 현악기이자 악기”임을 나타낸다.



isa 링크 => 대상을 정의
has 링크 => 구성 요소를 정의
do 링크 => 대상의 행위를 정의

ex 바이올린/첼로는 현악기다.
ex 현악기는 현을 가지고 있다.
ex 악기는 소리를 낸다.

악기에 대한 지식을 위와 같이 표현한 경우, “바이올린은 현악기인가?”라는 질문에 대해 isa 링크를 통해 “바이올린은 현악기다”라는 결론을 추론할 수 있다. 마찬가지로 “첼로는 악기인가?”라는 질문에 대해 먼저 첼로는 현악기라는 사실, 다시 현악기는 악기라는 사실로부터 첼로는 악기라고 추론할 수 있다.

의미망으로 표현된 지식을 컴퓨터 내부에 표현하기도 매우 쉽다. 다음은 위 의미망을 컴퓨터 내부에 표현하는 한가지 사례를 나타낸다.

```
node 0
label : 악기
isa :
do : 소리를 낸다
isalnknk:
node 1
label : 현악기
isa : 사물
has : 현
isalnknk: 0
node 2
label : 바이올린
isa : 현악기
isalnknk: 1
node 3
label : 첼로
isa : 현악기
isalnknk: 1
```

의미망은 지식을 매우 직관적으로 나타낼 수 있다는 장점이 있다. 위 그림처럼 지식의 내용이 많지 않다면 추론도 매우 쉽다. 따라서 다루려는 지식의 범위가 제한적인 경우라면 효과적으로 사용할 수 있다. 실제로 프레젠테이션 등에 의미망 형태가 흔히 사용된다. 그러나 우리 인간의

머릿속에 든 지식의 양은 상상 이상으로 다양하고 많다. 그러한 다양한 개념의 복잡한 상호 관계를 의미망으로 일관성 있게 표현하기가 매우 어렵고(어떤 노드는 do를 다른 노드를 has를 갖는 등 일관성이 없다), 그렇게 표현된 복잡한 지식을 효율적이며 효과적으로 다루기도 어려우며 결과를 검증하기도 어렵다(악기에 대한 지식을 의미망으로 완성해보면 알 수 있다).

규칙

초기 인공지능 연구자들은 주로 수학자와 논리학자 등이었음을 고려하면 지식을 규칙의 형태로 표현하려는 생각은 매우 자연스럽다. 다음은 현악기에 대한 일련의 지식을 규칙으로 표현한 사례다(대응하는 의미망을 그려보자). 지식을 규칙의 형태로 표현할 때는 보통 if x is/has/do/... A then y is/has/do/... B의 형태로 나타내는데, 여기서 x와 y는 같을 수 있다.

```
if 현이 있다 then 현악기이다
if 현악기이다 and 활로 연주한다 then 바이올린 계열이다
if 바이올린 계열이다 and 고음 then 바이올린이다
if 바이올린 계열이다 and 중음 then 바이올린이다
if 바이올린 계열이다 and 저음 then 첼로이다
if 현악기이다 and 손가락으로 연주한다 then 발현악기이다
if 발현악기이다 and 가슴에 안는다 then 기타이다
if 발현악기이다 and 바닥에 둔다 then 가야금이다
```

사람이 사용하는 다양한 지식을 $p \rightarrow q$ 같이 단순해 보이는 규칙으로 표현할 수 있을지 의문을 가질 수 있겠다. 하지만 ~이다, ~이 아니다 등을 사용해서 사람들이 사용하는 언어의 형태로 사물이나 대상의 관계, 추천, 지시, 전략, 경험 등을 표현할 수 있다(Durkin, 1994).

🔍 관계

if 연료 탱크가 비었다 then 차가 멈췄다

🔍 추천

if 가을이다 and 하늘이 흐리다 and 일기예보는 보슬비다 then 차가 멈췄다

☉ 지시

if 차가 멈췄다 and 연료 탱크가 비었다 then 차에 연료를 공급한다

☉ 전략

if 차가 멈췄다 then 연료 탱크를 확인한다 and 1단계를 완료했다

if 1단계를 완료했다 and 연료 탱크가 가득 찼다 then 배터리를 확인한다 and 2단계를 완료했다

☉ 경험

if 시료는 액체다 and 시료의 pH < 6 and 시료의 냄새가 시큼하다 then 시료는 아세트산이다

또한, 다음과 같이 수치나 수식도 표현할 수 있다,

```
if 고객의 나이 < 18 and 현금인출액 > 1,000
then 부모의 동의가 필요하다
```

```
if 과세 소득 > 20,000
then 의료보험 금액 = 과세 소득*1.5/100
```

규칙과 추론은 수학과 논리학의 오랜 연구 대상으로 많은 이론이 완성되어 있다. 사실 괴델의 불확정성 원리가 증명되기 전까지, 기계에 의한 논리적 추론이 완벽하게 실현될 것으로 기대되었으며, 인공지능의 실현 수단인 컴퓨터의 수학적 모델인 튜링머신이 기계에 의한 계산 불가능성의 증명 과정에 만들어졌다는 것은 매우 아이러니한 상황이다. 규칙으로 인간의 지식을 모두 나타내기에는 한계가 있다. 특히 애매하거나 불확실한 정보나 지식을 다루기도 쉽지 않다. 그러나 그러한 한계에도 불구하고 명확한 논리적 추론 등이 가능해서 컴퓨터로 다루기에 효과적인 규칙표현 방법으로 널리 사용된다.

전문가 시스템

단순한 기호처리를 넘어 지식에 기초하여 인공지능을 실현하려는 노력이 큰 성공을 거둔 것이라는 기대는 지식의 병목현상이라는 난관을 접하게 된다. 인간의 지식이란 게 당시의 기술력으로 감당하기에는 생각보다 방대했다. 도로 폭이 좁아지는 곳에서 혼란이 야기되듯 방대한 인간의 지식을 처리하기에 역부족임을 알게 된 것이다. 이러한 상황에서 돌파구를 마련한 인공지능 기술이 전문가 시스템이다. 실제로 전문가 시스템은 상업적으로 큰 성공을 거둔 최초의 인공지능 시스템이다.

모든 지식을 다룰 수 없다면 잘 할 수 있는 것만 처리하자는 게 전문가 시스템의 핵심 개념이다. 사실 사람도 모든 걸 다 잘하지 않는다. 대신 사람은 특정 분야에서 다른 사람보다 더 잘하는 게 한두 가지쯤 있다. 일반적으로 특정 분야에 관해서 (사실과 규칙 둘 다에 관한) 해박한 지식과 풍부한 경험을 쌓은 사람을 전문가라고 부른다. 그렇다면 모든 걸 알고 있으며 모든 문제를 처리할 수 있는 신과 같은 시스템을 만드는 대신 특정 분야의 문제를 척척 해결하는 시스템을 만들면 되지 않겠는가?

제한된 좁은 문제 영역에서 (사람) 전문가 수준으로 동작하는 컴퓨터 프로그램 혹은 컴퓨터 시스템이 전문가 시스템이다. 즉, 특정 분야에 대해서 전문가들이나 할 수 있는 정도의 지식처리 능력을 갖춘 인공지능 시스템이 전문가 시스템이다. 이제 전문가 시스템을 만들어보자.

규칙기반 전문가 시스템

전문가 수준의 지식을 알고 있으며 전문가처럼 문제를 해결하는 게 전문가 시스템이다. 그렇다면 먼저 전문가가 어떻게 사고하는지, 어떻게 자신의 지식을 표현하고 문제의 해답을 추론하는지 알아야 그런 시스템을 만들 수 있겠다. 엄밀하게 따지자면 그게 쉽지 않다! 내가 알고 있는 지식이 머릿속에 어떻게 저장되어 있으며, 사고 과정이 어떤지 알고 있는 사람은 없다. 하지만 전문가가 어떻게 사고하는지 아는 방법이 전혀 없는 것은 아니다.

전문가 대부분은 자신의 지식을 문제 풀이에 관한 규칙으로 표현하는 데 능숙하다. 동아시아 특정 나라에서는 오토바이와 사람이 인도와 차도 구분 없이 몰 흐르듯 이동한다. 처음 그 나라를 방문한 사람은 건널목조차 건너기가 쉽지 않다. 여행객은 아마추어고 현지인은 적어도

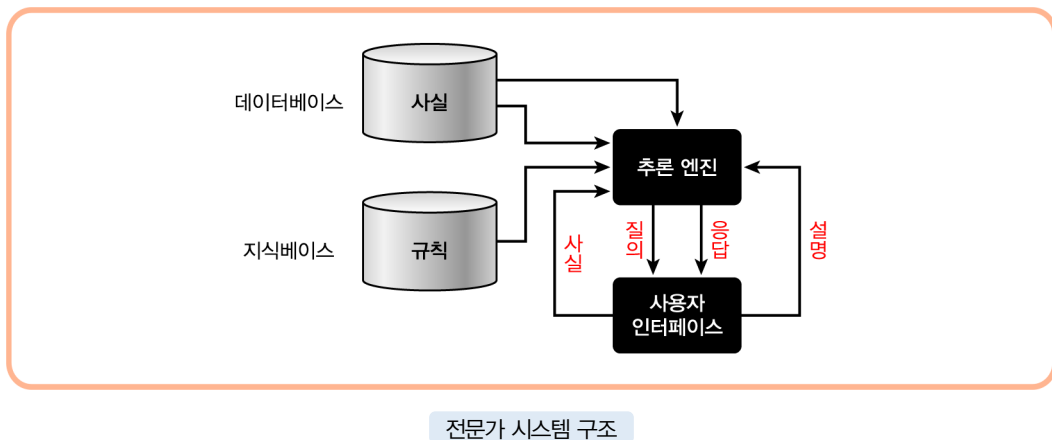
건널목 건너기 선수 즉, 전문가다. 현지인은 친절하게 길 건너는 방법을 알려줄 수 있다. 첫째 상대방과 끝까지 눈을 맞추라. 둘째 뛰지 말고 속도를 유지하라. 이게 규칙이고 이 규칙을 알고 있으면 모두 전문가다(남들이 모르는 무언가를 알고 있으면 - 옳고 그름은 별개로 하고 - 그것에 관해서 당신은 전문가다. 정보가 넘쳐나는 지식기반 사회는 전문가가 너무 많은 게 문제다).

의사: 어떻게 오셨나요?
 방문자: 숨쉬기가 어렵습니다!
 의사: 발열이 있습니까?
 방문자: 38도가 넘어요!
 의사: 마른기침을 하나요?
 방문자: 콜록 콜록!
 의사: 코로나19가 의심됩니다!

의사는 다음과 같은 규칙(편의상 57번 규칙이라고 가정하겠다)을 아주 많이 알고 있는 전문가다. 의사는 오랜 기간 학교와 환자를 다룬 경험으로부터 그러한 규칙을 배워 익힌다.

R57: if 호흡곤란 and 발열 and 마른기침 then 코로나19 의심

전문가 시스템을 만드는 방법은 지식의 표현방법과 추론 방법에 따라 여러 가지가 있을 수 있다. 그중에서 지식을 규칙으로 표현하고, 규칙에 기초한 추론으로 문제를 해결하는 전문가 시스템을 규칙기반 전문가 시스템이라고 한다. 다음은 전형적인 전문가 시스템 구조를 나타낸다.



규칙기반 전문가 시스템은 사실과 규칙으로부터 추론을 통해 문제를 해결한다. 보통 사실은 작업 메모리 또는 데이터베이스에, 지식은 지식베이스에 저장되며, 추론 엔진이 사실로부터 대응하는 규칙을 찾아 일련의 논리적 추론과정을 적용해서 적절한 해결방안을 제시한다. 이때 사용자와 상호작용하기 위한 장치가 사용자 인터페이스다.

앞선 예에서 의사를 전문가 시스템 Doc으로 바꿔보자. 초기 Doc 내부 상태 Q는 아래와 같다고 가정한다.

Q = { 사실={ }, 규칙={ ..., R57, ... } }

//사용자 인터페이스가 다음과 같은 질문을 생성한다
Doc: 어떻게 오셨나요?

방문자: 숨쉬기가 어렵습니다!
//사용자 인터페이스가 답변을 분석해서 호흡곤란을 사실에 추가한다

Q = { 사실={호흡곤란}, 규칙={ ..., R57, ... } }

//추론 엔진이 새로운 사실을 적용할 규칙을 찾아본다
//R57의 if 조건 중 하나와 일치됨을 확인하고, 다른 조건들이 사실인지 확인한다
//사실에 그런 내용이 없으므로, 사용자 인터페이스가 다음과 같은 질문을 생성한다
Doc: 발열이 있습니까?

방문자: 38도가 넘어요!
//사용자 인터페이스가 답변을 분석해서 발열을 사실에 추가한다

Q = { 사실={호흡곤란, 발열}, 규칙={ ..., R57, ... } }

//추론 엔진이 갱신된 사실을 적용할 규칙을 찾아본다
//R57의 if 조건 중 2개가 일치됨을 확인하고, 다른 조건이 사실인지 확인한다
//사실에 그런 내용이 없으므로, 사용자 인터페이스가 다음과 같은 질문을 생성한다
Doc: 마른기침을 하나요?

방문자: 콜록 콜록!
//사용자 인터페이스가 답변을 분석해서 마른기침을 사실에 추가한다

Q = { 사실={호흡곤란, 발열, 마른기침}, 규칙={ ..., R57, ... } }

//추론 엔진이 갱신된 사실을 적용할 규칙을 찾아본다
//R57의 if 조건이 모두 사실임을 확인하고, 사용자 인터페이스가 다음과 같은 진단 결과를 생성한다
Doc: 코로나19가 의심됩니다!

이상으로 규칙기반 전문가 시스템의 구조와 동작 과정을 잘 이해할 수 있겠다. 물론 실제 추론과정은 훨씬 복잡하고, 규칙의 충돌과 해소 등 관련된 이론이 많이 있다. 그러한 내용은 우리의 주제는 아니므로 더는 다루지 않는다.

앞에서 의사 전문가를 대상으로 전문가 시스템이 어떻게 동작하는지 살펴보았다. 이번에는 자동차 수리 전문가 시스템을 만든다고 가정해보자. 어떻게 해야 할까? 아주 간단하다. 지식베이스 내 규칙(과 필요한 최소한의 사실들)만 교체하면 그만이다. 즉, 해당 전문가의 지식을 수집해서 지식베이스에 넣어주면 새로운 분야의 전문가 시스템을 생성할 수 있다는 말이다. 아무런 지식을 포함하지 않은 전문가 시스템을 전문가 시스템 틀(shell 또는 tool 또는 frame)이라고 한다. 이는 전문가 시스템이 갖는 중요한 특징의 하나다. 처음부터 의사로 태어나는 사람은 없다. 배움과 경험을 통해 의사로서의 지식을 갖추면 의사가 된다. 우리 모두 배우는 능력을 타고났으니, 무엇이든 될 수 있듯이, 잘 만들어진 전문가 시스템 틀이 있다면 그것을 이용해서 특정 분야의 전문가 시스템을 개발할 수 있다는 뜻이다.

규모가 큰 전문가 시스템을 개발하려면 다양한 인력이 필요하다. 다음은 전문가 시스템 개발과 관련한 인력을 요약한 것이다.

☉ 주제 전문가

특정 주제나 분야에 관한 지식과 경험이 풍부한 사람. 자신의 지식이 완전하게 전달된 지식베이스가 구축되어야 전문가 시스템으로써의 목적을 이룰 수 있다.

☉ 지식공학자

전문가 시스템을 설계, 구현, 평가하는 사람으로, 특히 주제 전문가로부터 전문가의 지식을 규칙의 형태로 추출하는 일을 수행한다. 지식처리뿐 아니라 소프트웨어 개발에도 상당한 지식을 갖춘 사람으로 때에 따라 직접 프로그래밍에 참여할 수도 있다.

☉ 프로젝트 관리자

프로젝트 목표, 진행 일정, 개발 인력의 관리 등을 담당하는 전문가 시스템 개발의 총 책임자.

☀ 프로그래머

지식공학자 또는 프로젝트 관리자의 지시에 따라 프로그램을 만드는 프로그래밍 담당자. 전문가 시스템 틀을 사용하는 경우 프로그래머는 없을 수 있다.

전문가 시스템 사례

1장에서 언급한 허먼의 두뇌우성 모델에 따르면 사람의 뇌는 크게 4가지 영역으로 나뉘어 있고, 각 영역의 역할이 다르다. 사람마다 상대적으로 발달한 영역이 다를 수 있으며 어느 영역이 더 발달했는지에 따라 머리를 쓰는 방식이 다르다. 여기서는 특정인의 행동유형과 학습유형에 따라 그 사람의 두뇌우성 유형을 구분하는 두뇌우성 전문가 시스템에 대해 알아보자.

허먼의 두뇌우성 모델을 분석해보면 사람의 행동유형 및 학습유형 그리고 부분과 전체에 대한 태도와 두뇌우성의 관계를 다음과 같이 요약할 수 있다(이런 요약을 생성하는 게 주제 전문가와 지식공학자가 하는 일이다).

☀ 행동유형에 따른 성격유형 대응 규칙(4개)

행동유형	성격유형
<ul style="list-style-type: none">• 데이터와 정보를 검색한다.• 가설을 세우고 검증한다.• 사실과 기준, 논리로 판단한다.• 사례를 분석한다.	성과지향
<ul style="list-style-type: none">• 지침에 따라 단계적으로 처리한다.• 구체적인 예산을 세운다.• 반복해서 기술을 익힌다.• 배운 지식의 응용에 대해 생각한다.	작업중심
<ul style="list-style-type: none">• 사람들과 대화를 좋아한다.• 관찰하고, 검증하기를 좋아한다.• 다른 사람의 권리와 의견을 존중한다.	가치지향
<ul style="list-style-type: none">• 직감에 의지한다.• 미래를 예견한다.• 말보다 그림을 좋아한다.	위험주도

☉ 학습유형에 따른 사고유형 대응 규칙(4개)

학습유형	사고유형
<ul style="list-style-type: none"> 강의를 듣는다. 교재를 읽는다. 예제를 풀어본다. 	논리적
<ul style="list-style-type: none"> 꼼꼼하게 필기한다. 정돈된 환경에서 공부한다. 이론이나 절차를 검토한다. 	조직적
<ul style="list-style-type: none"> 음악을 틀어놓고 공부한다. 다른 사람들의 아이디어를 듣고 배운다. 오감으로 배운다. 	감성적
<ul style="list-style-type: none"> 학습을 위해 활발하게 활동한다. 답이 없는 문제를 골라 가능한 해결책을 모색한다. 다른 방법을 찾아본다. 	직관적

☉ 성격유형, 사고유형, 부분과 전체에 대한 태도와 두뇌우성 대응 규칙(6개: 더 많은 조합이 가능하나 여기서는 6개 규칙만 다루기로 한다)

성격유형, 사고유형, 부분과 전체에 대한 태도	두뇌우성
가치지향 & 조직적 & 부분 , 전체	A, C(b)
위험주도 & 직관적 & 부분 , 전체	D(a), D
작업중심 & 감성적 & 전체 , 부분	C, B(c)
작업중심 & 논리적 & 부분 , 전체	A(c), AB(cd)
성과지향 & 직관적 & 부분 , 전체	A(d), D(a)
성과지향 & 논리적 & 부분 , 전체	A, A(cd)

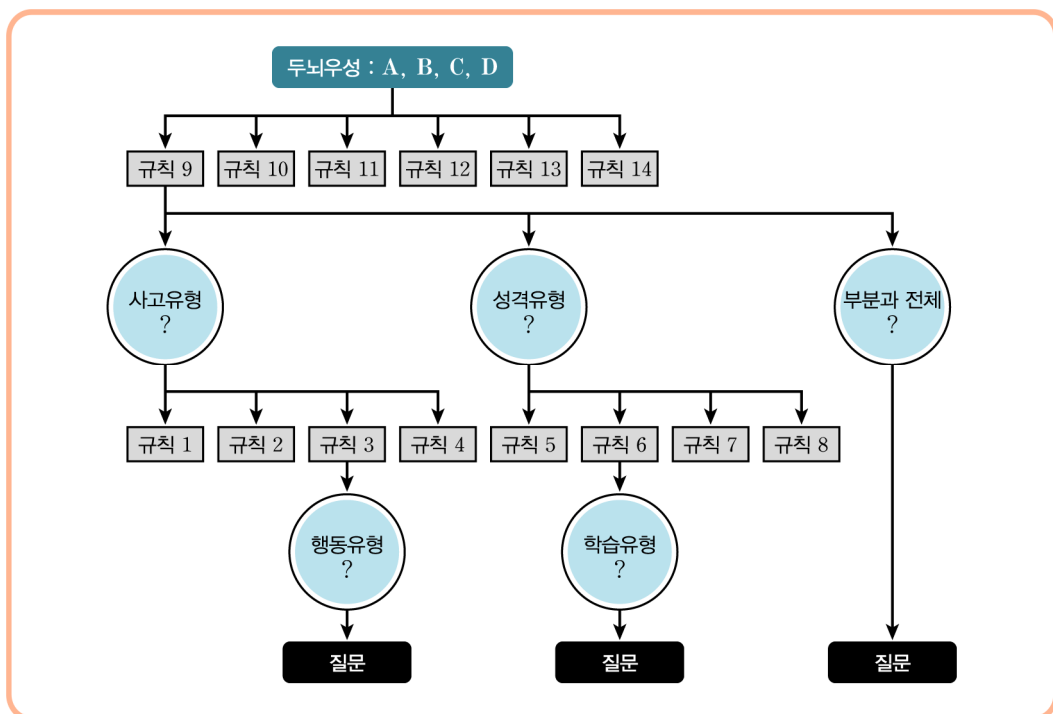
※ 푸른색은 부분과 전체에 대한 태도를 다르게 설정한 경우

표에서 두뇌우성의 괄호 안에 표시한 알파벳은 다중지능 값을 나타낸다. 예를 들어 행동유형이 작업중심적이고 학습유형은 논리적이며 부분과 전체에 대한 태도가 전체인 사람은 뇌의 전체 영역이 고르게 발달한 경우로, 여러 분야에 재능을 보이는 천재의 두뇌우성에 해당한다.

☉ 부분과 전체에 대한 태도가 취할 수 있는 값

부분과 전체에 대한 태도?	전체 부분
----------------	----------

이러한 분석에 따르면 전체 규칙의 구조는 다음과 같이, 행동유형에 따라 성격유형을 판단하는 규칙 4개(규칙1~4), 학습유형에 따라 사고유형을 판단하는 규칙 4개(규칙5~8), 마지막으로 성격유형, 사고유형, 부분/전체로부터 두뇌우성유형을 판단하는 규칙 6개(9~14)로 구성되었음을 알 수 있다.



규칙의 상호 관계

마지막으로 전문가 시스템이 사용할 규칙 14개를 if ~ then의 형태로 나타내면 다음과 같다.

[규칙1]

if 행동유형이 데이터와 정보를 검색한다.
or 가설을 세우고 검증한다.
or 사실과 기준, 논리로 판단한다.
or 사례를 분석한다.
then 성격유형은 성과지향적이다.

[규칙2]

if 행동유형이 지침에 따라 단계적으로 처리한다.
or 구체적인 예산을 세운다.
or 반복해서 기술을 익힌다.
or 배운 지식의 응용에 대해 생각한다.
then 성격유형은 작업중심적이다.

[규칙3]

if 행동유형이 사람들과 대화를 좋아한다.
or 관찰하고, 검증하기를 좋아한다.
or 다른 사람의 권리와 의견을 존중한다.
then 성격유형은 가치지향적이다.

[규칙4]

if 행동유형이 직감에 의지한다.
or 미래를 예견한다.
or 말보다 그림을 좋아한다.
then 성격유형은 작업중심적이다.

[규칙5]

if 학습유형이 강의를 듣는다.
or 교재를 읽는다.
or 예제를 풀어본다.
then 사고유형은 논리적이다.

[규칙6]

if 학습유형이 꼼꼼하게 필기한다.
or 정돈된 환경에서 공부한다.
or 이론이나 절차를 검토한다.
then 사고유형은 조직적이다.

[규칙7]

if 학습유형이 음악을 틀어놓고 공부한다.
or 다른 사람들의 아이디어를 듣고 배운다.
or 오감으로 배운다.
then 사고유형은 감성적이다.

[규칙8]

if 학습유형이 학습을 위해 활발하게 활동한다.
or 답이 없는 문제를 골라 가능한 해결책을 모색한다.
or 다른 방법을 찾아본다.
then 사고유형은 직관적이다.

[규칙9]

if 성격유형이 가치지향이다.
and 사고유형 조직적이다.
and 부분과 전체에 대한 태도가 전체이다.
then 두뇌우성은 C(b)이다.

[규칙10]

if 성격유형이 위험주도이다.
and 사고유형 직관적이다.
and 부분과 전체에 대한 태도가 전체이다.
then 두뇌우성은 D이다.

[규칙11]

if 성격유형이 작업중심이다.
and 사고유형 감성적이다.
and 부분과 전체에 대한 태도가 부분이다.
then 두뇌우성은 B(c)이다.

[규칙12]

if 성격유형이 작업중심이다.
and 사고유형 논리적이다.
and 부분과 전체에 대한 태도가 전체이다.
then 두뇌우성은 AB(cd)이다.

[규칙13]

if 성격유형이 성과지향이다.
and 사고유형 직관적이다.
and 부분과 전체에 대한 태도가 전체이다.
then 두뇌우성은 D(a)이다.

[규칙14]

if 성격유형이 성과지향이다.
and 사고유형 논리적이다.
and 부분과 전체에 대한 태도가 전체이다.
then 두뇌우성은 A(cd)이다.

두뇌우성 유형을 구분하는 규칙이 구축되었으니, 전문가 시스템 틀의 지식베이스에 넣으면 전문가 시스템이 완성된다. 실제로 전문가 시스템과 대화를 통해서 특정인의 두뇌우성을 판별하는 과정을 예를 통해 알아보자.

BrainExpert: 자신의 행동유형은 무엇인가?

사용자: 사람들과 대화를 좋아한다.

(규칙3의 조건과 일치하므로, 성격유형 “가치지향적”이 사실에 추가된다.)

[규칙3]

if 행동유형이 사람들과 대화를 좋아한다.
or 관찰하고, 검증하기를 좋아한다.
or 다른 사람의 권리와 의견을 존중한다.
then 성격유형은 가치지향적이다.

BrainExpert: 자신의 학습유형은 무엇인가?

사용자: 정돈된 환경에서 공부한다.

(규칙6의 조건과 일치하므로, 사고유형 “조직적”이 사실에 추가된다.)

[규칙6]

if 학습유형이 꼼꼼하게 필기한다.
or 정돈된 환경에서 공부한다.
or 이론이나 절차를 검토한다.
then 사고유형은 조직적이다.

BrainExpert: 자신의 부분과 전체에 대한 태도는 무엇인가?

사용자: 전체를 먼저 파악한다.

(“전체”가 사실에 추가되고, 규칙9의 모든 조건과 일치하므로, 두뇌우성 C(b)가 사실에 추가된다.)

[규칙9]

```
if 성격유형이 가치지향이다
and 사고유형 조직적이다
and 부분과 전체에 대한 태도가 전체이다
then 두뇌우성은 C(b)이다
```

두뇌우성 값을 조건으로 하는 규칙이 없으므로 추론을 종료하고, 마지막으로 추가한 사실을 출력한다. 즉, 해당 사용자의 두뇌우성은 뇌의 오른쪽 아랫부분이 발달한 감성형이면서 동시에 조직적인 다중지능을 가진 것으로 판별되며 따뜻한 마음을 가진 리더형에 해당한다.