

제목 : 컴퓨터를 이용한 수리논리학 교육¹⁾
 저자: 정주희 (경북대학교)

(대한수학교육학회 논문집, Vol 9 #1, pp111-119, 1999.7에 실린 내용입니다.)

서론

컴퓨터는 교육의 매체로서 책이 가지지 못하는 여러 장점을 가진다. 이 중에서 가장 중요한 것으로 피교육자가 교육받는 과정에서 대화식의(interactive) 실험이 가능하다는 점을 들 수 있다. 수학교육에서 필요한 실험이란 주로 계산과 그래프 등의 그림 그리기이며 이 둘을 위하여는 펜과 종이보다 컴퓨터가 나음은 두말할 나위도 없다. 컴퓨터가 인간 교사를 완전히 대체할 수는 없겠으나 잘 구성된 소프트웨어의 적절한 사용은 수학교육의 질을 한층 더 높일 수 있음이 당연하다.

수학의 모든 분야에서 컴퓨터를 이용한 교육이 유용하겠으나 본 연구에서는 수리논리학 교육용 소프트웨어 개발을 목표로 하였다. 현대의 논리학이란 기본적으로 기호논리학이라고 말할 수 있으며 컴퓨터가 기호계산(symbolic computation)에 능하다는 것은 잘 알려진 사실이다. 따라서 논리교육의 매체로서 컴퓨터를 사용하는 것은 대단히 효과적일 것으로 기대된다.

컴퓨터를 이용한 수학교육은 국내에서도 최근 수년간 몇몇 연구자들이 시작하여 점차 활발해지고 있으나 수리논리교육용 소프트웨어는 개발된 사례가 없다. 외국에서도 기하학, 대수학 등에는 비교적 많은 소프트웨어가 개발되어 쓰이고 있으나, 논리 분야에서는 단 하나, 스탠포드 대학의 언어정보연구소(The Center for the Study of Language and Information)에서 개발한 Tarski's World를 찾아볼 수 있는 정도이다. 본 연구 분야는 시작할 바탕이 되는 기존 연구가 국내외에 극히 미비하다고 하겠으며, 본 연구자는 앞서 언급한 Tarski's World 하나만을 참고로 하여 새롭게 소프트웨어 개발을 시작하였다.

기호논리학의 기본인 명제논리(propositional logic)만 가지고도 우리가 흔히 접하는 논리퍼즐은 간단히 풀 수 있으므로, 이를 통하여 학생의 논리학에 대한 흥미와 학습의욕을 고취시킬 수 있다. 본 연구에서 개발한 소프트웨어 jLogic은 명제논리의 다음 단계인 1계논리(first-order logic), 혹은 술어논리(predicate logic)에서의 모델(model)과 만족도(satisfiability)의 개념을, 컴퓨터에 실제로 구현(implement)할 수 있는 유한 모델을 예로 들어서 가르친다.

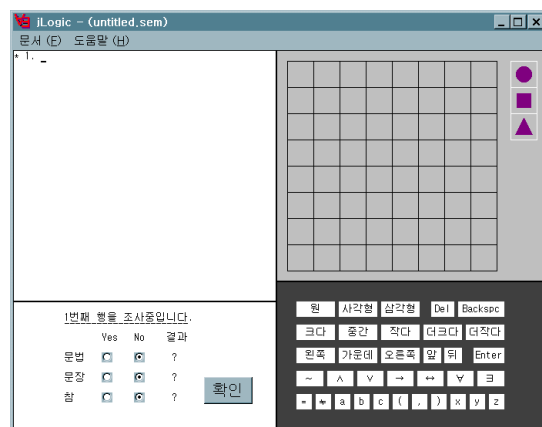
소프트웨어의 플랫폼으로는 교육효과의 극대화를 위하여 현재 가장 널리 사용되고 있는 GUI인 마이크로소프트사의 윈도우를 택하였다.

소프트웨어 jLogic의 개요

jLogic은 현대 수리논리학의 입문을 위하여 마련된 교육용 소프트웨어이다. 대학 1,2학년 정도면 혼자서 이 소프트웨어를 사용해 가면서 학습할 수 있겠으며, 뛰어난 자질을 가진 중고등학생의 경우 교사의 지도하에 충분히 수업이 가능하다고 본다.

논리학은 추론의 과학이며 수리논리학은 수학의 명제를 표현하고 증명하는데 사용하는 논리체계를 연구하는 학문이다. 여러 논리체계 중에서 가장 널리 사용되는 것이 1계논리이며 jLogic은 이 1계논리의 논리식과 의미론을 공부하는 입문과정에 유용하게 사용될 수 있다.

학습자가 윈도우 95에 대한 어느 정도 경험이 있어야만 바라는 학습효과를 거둘 수 있음은 당연

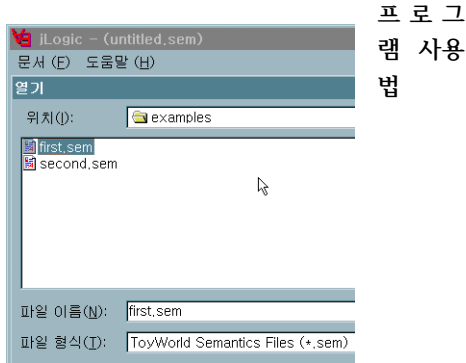


<그림 1> 전체 화면

하며, 본고는 이것을 가정하고 쓴 것이다.

1) 이 논문은 1996년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

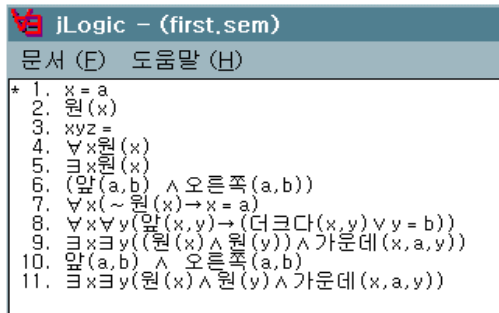
jLogic을 실행시키면 4부분으로 구성된 화면이 나타난다. 화면의 왼쪽 위 부분은 논리식 윈도우라고 부르고 오른쪽 위 부분은 토이우주 윈도우, 왼쪽 아랫 부분은 조사 윈도우, 오른쪽 아랫 부분은 키보드 윈도우라고 부른다.



프 로 그 램 사용 법

<그림 2> 예제 데이터 파일 읽어 들이기

예제 데이터 파일을 가지고 작업을 해 보자. 메뉴



<그림 3> 논리식 윈도우

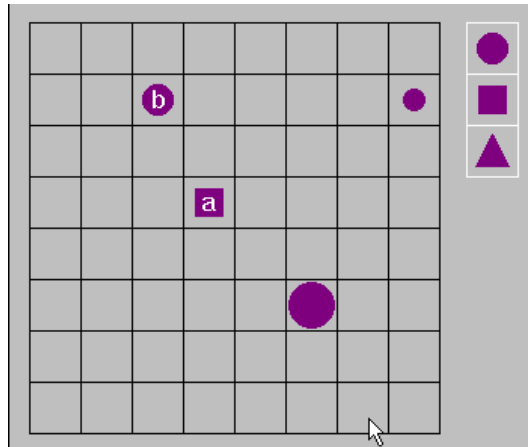
바에서 [문서][불러오기]를 실행시킨 다음 examples 디렉토리로 들어가 보면 2개의 예제 파일 first.sem과 second.sem이 보일 것이다.

이 중에서 first.sem을 읽어들인다. 이제 화면 좌상단의 논리식 윈도우를 보자.

11개의 논리식이 보일 것이다. 이 중에서 1번 논리식은 *로 표시되어 현재 선택되어 있음을 보여준다.

논리식은 기본적으로 그것의 대상에 대해서 어떤 주장을 하는 것이다. jLogic에서는 논리식이 토이우주(Toy World)를 대상으로 말하고 있다. 토이우주에는 물체들이 놓여 있다. 물체들은 모양과 크기라는 두 속성을 가진다. 현재 first.sem을

읽어 들인 결과로 토이우주에는 4개의 물체가 있

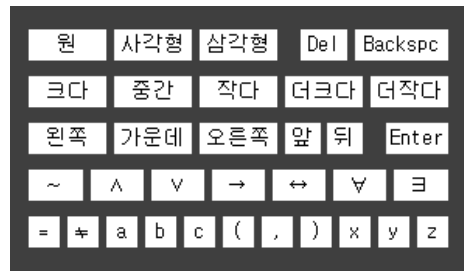


<그림 4> 토이우주 윈도우

으며 이 중 3개는 원 모양이고 하나는 사각형 모양이다. 화면 우상단의 토이우주 윈도우를 보자.

맨 앞쪽(즉 화면의 맨 아래쪽)에 있는 원은 크기가 크고, 맨 오른쪽에 있는 원은 크기가 작으며 나머지 2개의 물체는 크기가 중간이다. 물체에는 또한 이름을 줄 수도 있으며 왼쪽에 있는 원은 b, 가운데 있는 사각형은 a로 이름 붙여져 있다. 그리고 나머지 두 물체에는 이름이 붙여져 있지 않다.

논리식 윈도우에는 현재 11개의 논리표현들이 있다. 이 중에 어떤 것은 문법에 맞는 논리식이고 어떤 것은 그렇지 못하다. 문법에 맞는 논리식 중에 어떤 것은 참과 거짓을 판별할 수 있는 문장이



<그림 5> 키보드 윈도우

며 어떤 것은 그렇지 못하다.

논리식에 사용되는 기호는 (거의) 모두 화면 오른쪽 하단의 키보드에 나타나 있다. 키보드에 나타난 키 중에서 논리기호가 아닌 것이 3개 있는데 그것은 Del, Backspc와 Enter이다.

논리식 중 가장 간단한 것은 변수 = 변수, 변수 = 상수, 상수 = 변수, 상수 = 상수, 변수 ≠ 변수, 변수 ≠ 상수, 상수 ≠ 변수, 상수 ≠ 상수 형태이다. 다음으로 간단한 논리식은 원(x), 원(a), 왼쪽(x,y) 등의 형태

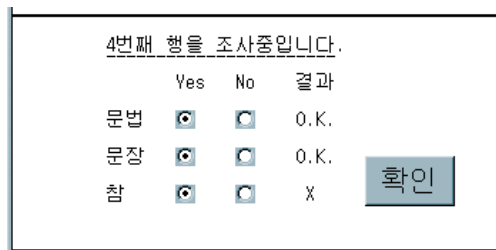
를 가진다. 이런 형태의 논리식을 아톰논리식이라고 부른다. <그림 3>에서 1번과 2번의 논리표현은 문법에 맞는 (아톰)논리식이며 3번은 문법에 맞지 않다.

1번의 논리식은 문법에 맞기는 하지만 진위를 판별할 수는 없다. 왜냐하면 변수 x 가 어떤 값을 취하느냐에 따라 진리값이 달라지기 때문이다. 2번 논리식도 마찬가지로 문법에 맞기는 하지만 진위를 판별할 수 없다.

4번 논리식에서 사용된 기호 \forall 는 전칭한정기호라고 부르며 $\forall x$ 는 "for every x ", 즉 '모든 x 에 대하여'라는 뜻을 가진다. 하지만 <그림 4>의 토이우주 윈도우를 보면 원이 아닌 물체 (a로 표시된 사각형)이 존재하므로 4번 논리식은 문장이기는 하지만 참은 아니다.

5번 논리식에서 사용된 기호 \exists 는 존재한정기호라고 부르며 $\exists x$ 는 "there exists an x such that", 즉 "어떤 x 에 대하여"라는 뜻을 가진다. 5번 논리식은 참인 문장임을 알 수 있다.

논리식들은 $\sim, \wedge, \vee, \rightarrow, \leftrightarrow$ 등의 결합자를 사용하여 결합시켜 새로운 논리식을 형성할 수 있다. 이들 결합자들의 의미는 각각 not (아니다), and(그리고), or (혹은), implies (함의) 및 equivalent



<그림 6> 조사 윈도우

(동등)이다.

왼쪽 하단의 조사 윈도우에서, 사용자는 선택된 논리식 (논리식 윈도우에서 식 왼쪽에 *표시가 되어있는 것)에 대한 판단을 하여 Yes(예)와 No(아니오) 중에서 선택하게 되어있다. 그리고 확인 버튼을 누르면 jLogic은 사용자가 선택한 답이 맞는지 틀리는지를 대답해 준다.

<그림 6>에 4번 논리식에 대한 조사 결과의 화면을 보였다. 앞에서 말했듯이 4번 논리식은 문법에 맞고, 진위를 판별할 수 있으며, 진리값은 거짓이다. 그런데 사용자가 4번 논리식은 문법에 맞고, 진위를 판별할 수 있는 문장이며, 진리값이 참이라고 답했기 때문에, 확인 버튼을 클릭하면 그림에서 보듯이 첫 번째와 두 번째의 문법과 문장 행은 O.K.가, 그리고 세 번째의 참 행은 X가 결과로 나온다.

보충 설명을 하자면, 6번 논리식은 변수기호나 한정기호를 사용하지 않은 문장의 예를 보인 것이며, 10번은 6번과 거의 비슷하지만 가장 바깥쪽 괄호 쌍이 빠져서 문법에 맞지 않다. 11번은 9번과 비슷하지만 괄호 쌍이 하나 부족함을 알 수 있다.

지금까지 예제 데이터 파일 first.sem을 가지고 프로그램의 사용법을 간략하게 설명하였다. examples 디렉토리에 있는 또 하나의 예제 데이터 파일 second.sem에 대해서도 각자 연습을 해 보기를 바란다.

화면별 세부 사항

이제 독자들이 jLogic에 대한 어느 정도의 지식을 가진 것으로 가정하고 jLogic을 구성하는 4개의 화면별 세부 사항을 설명하기로 한다.

키보드 윈도우와 논리식 윈도우

화면의 좌상단에 위치한 것이 논리식 윈도우이다. 논리식은 한 행에 하나씩 입력되어 일련번호가 붙는다. 대부분의 기호는 화면의 우하단에 위치한 키보드 윈도우에 보이는 키보드 단추를 마우스로 클릭하여 입력할 수 있다. 직접 키보드에서 타자하여 입력할 수 있는 기호도 있으며 이는 다음과 같다. a b c d e x y z u v w 0 1 2 3 4 5 6 7 8 9 () , = <space>. 여기서 <space>란 키보드에서 스페이스바를 치는 것을 말한다. 물론 Del, Backspc, Enter도 마우스 클릭 대신에 키보드에서 직접 타자해도 된다. 여기서 d, e, u, v, w와 <space>는 마우스로는 입력이 불가능하고, 타자에 의해서만 입력할 수 있음을 알 수 있다. 그리고 Home, End와 위, 아래, 좌, 우 4개의 화살표 키는 커서를 움직이는데 사용된다

빈 행을 삽입할 때는 그냥 Enter하면 되는데 이때 커서가 행의 맨 앞에 있으면 현재 행의 위에, 그렇지 않으면 현재 행의 아래에 새로운 빈 행이 삽입된다.

토이우주 윈도우

토이우주 윈도우는 화면의 우상단에 위치하고 있다. 토이우주는 기본적으로 8 x 8 크기의 격자판에 원, 사각형, 삼각형 등의 물체를 놓음으로써 구성된다.

토이우주에 물체를 집어넣으려면 토이우주 우상단에 보이는 원, 사각형, 삼각형 중 하나를 마우스

로 집어서 원하는 위치에 놓으면 된다. 토이우주에 이미 놓여있는 물체를 이동시켜 위치를 변경시키려면 마우스로 집어서 원하는 위치에 옮겨 놓으면 된다. 토이우주에 놓여있는 물체를 없애려면 마우스로 집어서 격자판 밖에서 놓으면 된다.

마우스로 집을 때는 왼쪽 마우스 버튼을 써도 되고 오른쪽 마우스 버튼을 써도 된다. 단 오른쪽 마우스 버튼을 쓰는 경우에는 버튼을 놓은 후에 메뉴화면이 뜨면서 크기를 조정하거나 이름을 줄 수 있게 된다.

조사 윈도우

화면의 좌하단에 위치하고 있는 조사 윈도우에서는 선택된 논리식에 대한 다음의 세 가지 사항을 조사.확인한다. 첫째로 논리식이 문법에 맞는지, 둘째로 문법에 맞다면 진위를 판별할 수 있는 문장인지, 마지막으로 만일 문장이라면 진리값이 참인지의 여부를 사용자가 판단하여 답을 하고 확인 버튼을 누르면 jLogic이 사용자의 답을 확인해주는 것이다. 이상의 세 가지 사항은 조사 윈도우에서 각각 문법, 문장 및 참으로 표시된 행에 대응된다.

선택된 논리식은 논리식 윈도우에서 가장 왼쪽 열에 *로 표시되어 있다. 가령 n번 논리식이 선택되어 있다면 조사 윈도우의 맨 윗줄에 "n번째 행을 조사중입니다"라고 나타난다. 1번째(Yes) 열과 2번째(No) 열은 각 행에 대한 라디오버튼으로서 항상 둘 중의 어느 하나만 선택되도록 되어있다. 확인 버튼을 누르기 전에는 3번째(결과) 열은 ?로 표시된다. 확인을 누르면 jLogic이 사용자가 선택한 답이 맞는지의 여부를 계산.판단하여 맞으면 O.K., 틀리면 X로 나타내 준다.

맺음말

저자는 본고에서 설명한 논리교육용 프로그램 jLogic을 가지고 경북대 과학영재교육센터에서 중학교 2학년에 재학중인 학생들을 대상으로 교육해본 경험이 있다. 먼저 소프트웨어 없이 칠판에 기본 이론을 설명하고 이어서 컴퓨터 화면을 강의실의 스크린에 프로젝트해서, 앞에서 설명한 데이터 파일 first.sem을 예로 들어서 실제 사용과정을 보여주며 설명을 하였다. 학생들은 컴퓨터가 논리계산을 순간적으로 해내는 것에 비사한 관심을 보였다. 소프트웨어를 사용하지 않은 클래스와 사용한 클래스의 학업성취도를 비교하는 본격적인 연구는 하지 않았으나, 학생들의 수업에의 집중도가 소프

트웨어를 사용하지 않을 때와는 비교도 되지 않을 정도로 높음을 확인할 수 있었다.

본고에서는 자세히 논하지 않았으나 jLogic은 다양한 방법으로 수업에 활용될 수 있다. 예를 들어 현재의 토이우주에서 진리값이 거짓인 문장을 주고, 진리값이 참이 되도록 하기 위하여 주어진 문장을 어떻게 바꾸어야 할까, 혹은 토이우주를 어떻게 바꾸어야 할 것인가를 묻은 것이 하나의 방법이다. 물론 이 물음에 대한 답은 유일하지가 않고 여러 가지가 있을 수 있으며, 가능한 답을 찾아내는 과정에서 학생들은 논리식의 정확한 구문론과 의미론을 익히게 된다.

jLogic은 한국학술진흥재단의 연구비 지원을 받아 2년여에 걸쳐 Windows 95 용으로 개발된 것으로 현재 버전1.00이 처음으로 공개되었다. 누구든지 이 소프트웨어를 무료로 사용할 수 있으며 인터넷에 연결된 컴퓨터에서는

https://jhjeong.mindconnect.cc/Lecture_notes/logic/proofmood_lecture.php에서 쉽게 다운 받아서 PC에 설치할 수 있다.

jLogic은 앞으로 계속 업그레이드 될 예정이며 최신 버전은 위의 홈페이지에서 구할 수 있다.

논리교육용 소프트웨어가 전무한 국내 실정에서 본 연구의 결과인 jLogic이 널리 사용되고, 또한 사용자로부터의 많은 피드백을 받아서 버전을 거듭하며 개선되어 논리학 교육에 조금이라도 기여를 할 수 있게 되기를 바란다.

참고문헌

Barwise, Jon & Etchemendy, John (1992). The Language of First-Order Logic, CSLI Lecture Notes

[부록] 기호논리학

논리학에서 의미하고자 하는 바를 모호성 없이 명확하게 기술하기 위하여는 일반 언어가 아닌 특수 논리기호로 된 언어를 사용해야 한다. 모호성(ambiguity)이 없다는 것은 여러 가지 의미로 해석할 수 있겠으나 여기서는 주어진 논리표현을 컴퓨터에 입력하여 처리가 가능하다는 것을 의미하는 것으로 해 두자.

기호논리에는 여러 종류가 있는데 여기서는 가장 기본적인 또한 널리 사용되고 있는 명제논리(Propositional logic)와 1계논리(First-order logic)만을 설명하기로 한다

명제논리

명제논리는 모든 논리에 공통으로 사용되는 가장 핵심적인 논리체계이다. Logic이 실제로 사용하는 논리는 1계논리이나, 1계논리를 이해하기 위하여는 먼저 명제논리를 알아야 한다.

명제논리에서는 명제(statement)를 하나의 기호, 예를 들어 A, B, C 등으로 나타낸다. 이렇게 표시된 명제를 단순명제라고 부르자. 단순명제들은 결합자를 이용하여 복합명제를 이룬다.

우리가 사용할 결합자에는 \sim , \wedge , \vee , \rightarrow 및 \leftrightarrow 의 다섯 가지가 있다. 이 중에서 \sim 는 하나의 단순 혹은 복합명제 앞에 붙어서 다른 복합명제를 만들며 나머지 4개의 결합자 \wedge , \vee , \rightarrow 및 \leftrightarrow 는 두 개의 단순 혹은 복합명제 사이에 붙어서 다른 복합명제를 만든다. 예를 들어 A와 B가 단순명제일 때 $\sim A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ 및 $(A \leftrightarrow B)$ 는 새로운 복합명제이다. 조금 더 복잡한 명제로는 $\sim \sim A$, $(\sim A \rightarrow (A \vee B))$, $((A \rightarrow (B \rightarrow C)) \wedge \sim((A \rightarrow \sim B) \rightarrow C))$ 등 얼마든지 예를 들 수 있다

결합자의 의미(semantics)는 다음의 진리표에 의하여 명확하게 정의된다.

<표 1 > 결합자의 성질

A	B	$\sim A$	$(A \wedge B)$	$(A \vee B)$	$(A \rightarrow B)$	$(A \leftrightarrow B)$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

명제는 참 혹은 거짓의 진리값을 가진다. 통상 참은 1, 그리고 거짓은 0으로 간단하게 표기한다. (참을 T, 거짓을 F로 표기하는 경우도 있다.)

명제 $(A \vee \sim A)$ 를 생각해 보자. 만일 A의 진리값이 1이라면 준 명제의 진리값은 $1 \vee \sim 1 = 1 \vee 0 = 1$ 이 되며 반대로 A의 진리값이 0이라면 준 명제의 진리값은 $0 \vee \sim 0 = 0 \vee 1 = 1$ 이 된다. 즉 A가 어떠한 진리값을 취하든지 상관없이 준 명제의 진리값은 항상 1이 된다. 이러한 명제, 즉 구성요소인 단순명제의 진리값이 어떠한 값을 취하든 간에 항상 진리값이 1인 (복합)명제를 항진명제라고 부른다. 항진명제의 예를 몇 개 들어 보자면 $((A \rightarrow B) \leftrightarrow (\sim A \vee B))$, $((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$ 등이 있다. 이 두 개의 예가 실제로 항진임은 진리표를 그려보면 간단하게 확인할 수 있다.

"항진"의 반대 개념으로 "모순"이 있다. 이것은 구성요소인 단순명제의 진리값이 어떠한 값을 취하든 간에 항상 진리값이 0인 (복합)명제를 뜻하며 예를 들어 $(A \wedge \sim A)$ 는 모순이다.

1계논리

명제논리는, 추론에 유용하다는 것은 틀림없는 사실이긴 하나, 표현력이 충분하지 못하다는 단점이 있다.

아래의 유명한 3단논법은 명제논리로 쉽게 설명된다.

가설: 소크라테스는 인간이다. 모든 인간은 죽는다.

결론: 그러므로 소크라테스는 죽는다.

인간임을 H, 소크라테스임을 S, 죽는다는 M으로 놓으면 가설은 $(S \rightarrow H)$, $(H \rightarrow M)$ 으로, 그리고 결론은 $(S \rightarrow M)$ 으로 놓을 수 있다. 그러면 $((S \rightarrow H) \wedge (H \rightarrow M)) \rightarrow (S \rightarrow M)$ 이 항진이므로 위의 추론이 타당함을 확인할 수 있다.

이제 다음의 추론에도 명제논리가 적용되는지 보자.

가설: 인간은 이성적이다. 어떤 동물은 인간이다.

결론: 어떤 동물은 이성적이다.

이 추론은 타당하다. 하나 앞의 예와 같은 방법을 적용하는 데는 무리가 따른다. 즉 "어떤 동물임"을 하나의 단순명제로 놓아야 하는데 과연 이래도 되는 것인가? "어떤 동물은 인간이다"와 "어떤

동물은 인간이 아니다"는 명제논리적으로는 서로 모순이 되지만 실제로는 상호 모순이 아닌 것, 즉 동시에 성립할 수 있는 것이다.

이러한 상황을 자연스럽게 기술하기 위하여는 존재한정기호의 도입이 필요하다. 대상영역을 모든 동물(개체)의 집합으로 놓고 $H(x)$ 를 "x는 인간이다"를 뜻하는 술어로 놓았을 때 "어떤 동물은 인간이다"라는 명제는 $\exists xH(x)$ 로 나타내면 된다. 여기서 \exists 는 존재한정기호라고 부르며 $\exists x$ 는 "어떤 x에 대해서"라는 뜻이다.

1계논리식

1계논리식에서는 변수와 한정기호가 사용됨은 앞에서 보았다. 1계논리식의 구성요소로 또 하나 중요한 것은 술어이다. 예를 들어 x가 인간임을 $H(x)$ 로 나타내기로 한다고 했었는데, 여기서 H를 술어(predicate)라고 부른다. 술어의 또 다른 예로 x가 y의 형제임을 $B(x,y)$ 로 표시한다면 $B(x,y) \rightarrow B(y,x)$ 는 보통의 의미에서 참이 될 것이며, $(B(x,y) \wedge B(y,z)) \rightarrow B(x,z)$ 도 역시 참이 될 것이다.

가장 흔하게 쓰이는 술어는 아마도 =와 \neq 일 것이다. 술어와 변수기호를 사용하면 아톰논리식이 만들어진다. 예를 들어 $H(x)$, $B(x,y)$, $x=y$, $x \neq y$ 등은 아톰논리식이다. 변수기호 대신에 사용할 수 있는 것으로 상수기호가 있다. 예를 들어 수학에서 사용하는 0, 1 등은 상수기호이며, $x=0$ 와 $0 \neq 1$ 은 아톰논리식이다.

아톰논리식은 가장 단순한 형태의 논리식이다. 논리식(들)에 결합자와 한정기호를 사용하여 더 복잡한 논리식을 만들 수 있다. 결합자를 사용하는 방법은 명제논리 때와 같으므로 여기서는 설명을 생략한다. 한정기호에 대하여 설명하자면 다음과 같다.

한정기호에는 앞에서 본 존재한정기호 \exists 와 여기서 새로 도입하는 전칭한정기호 \forall 가 있다. 이들 한정기호의 바로 뒤에는 반드시 변수기호가 와야 한다. 그러면 A를 임의의 (1계)논리식이라고 하고 x를 변수기호라 했을 때 $\exists xA$ 와 $\forall xA$ 는 새로운 논리식이 된다. 예를 들어 A가 논리식 $(B(x,y) \wedge x \neq y)$ 라 했을 때 $\exists xA$, 즉 $\exists x(B(x,y) \wedge x \neq y)$ 는 새로운 논리식이다.

이상으로써 1계논리식을 구성하는 방법을 공부하였다. 1계논리식의 진리값, 즉 의미론에 대해서는 아래에 설명하였다.

(실은 1계논리에서 함수기호라는 것도 존재하나 jLogic은 기초적인 입문과정이므로 이것에 대한 설

명은 생략한다.)

1계논리의 의미론

1계논리식의 의미론에 대해서는 명제논리에서 다루었던 부분, 즉 결합자에 대해서는 설명을 생략한다.

1계논리에서 가장 특징적인 한정기호를 먼저 보기로 하자. A가 변수 하나를 택하는 술어일 때 $\exists xA(x)$ 는 A(x)가 성립하는 x가 존재한다는 것, 즉 어떤 x에 대하여는 A(x)가 참이라는 뜻이며, $\forall xA(x)$ 는 모든 x에 대해서 A(x)가 참이라는 뜻이다. 따라서 다음의 두 논리식은 서로 동등하다.

$$\sim \forall xA(x) \text{와 } \exists x \sim A(x)$$

같은 논리로 다음의 두 논리식은 서로 동등하다.

$$\sim \exists xA(x) \text{와 } \forall x \sim A(x)$$

1계논리식이 의미를 갖기 위해서는 먼저 대상영역과 술어기호의 의미가 정해져야 한다. 예를 들어 술어기호 <의 의미를 $x < y$ 는 x가 y보다 작음을 의미하는 것으로 했을 때 $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$ 는 대상영역이 실수 전체의 집합이거나 유리수 전체의 집합일 때는 참이지만 대상영역이 정수 전체의 집합일 때는 거짓이 된다.

jLogic에서는 대상영역을 토이우주로 고정시켰다. 그리고 술어기호의 의미도 (예를 들면) 원(x)는 x가 원임을 의미하는 것으로, 가운데(x,y,z)는 y가 x와 z의 사이에 있는 것을 의미하는 것으로 고정되어 있다. 이러한 제약은 jLogic의 본래 목적이 사용자가 한정기호와 결합자에 익숙해지고 그 의미의 이해를 확고히 하여 논리에 입문하는데 도움을 주기 위한 것이므로 하등의 문제가 되지 않는다고 본다.

이상으로 1계논리에 대한 설명을 대략 마쳤는데, 이는 대단히 간략한 설명이었으며 엄밀하고 상세한 부분은 많이 생략되어 있다. 하지만 논리에 대한 이 정도 수준의 엄밀성만 갖추면 전산학 혹은 수학의 학부 및 대학원 과정까지를 공부하는데 충분하다고 본다.