

공식 원리에 기반한 대화식 문제해결 학습을 지원하는 수학교수 모형

국 형 준[†]

요 약

학습 효과와 실용성이 모두 높은 교수 체계의 개발 필요성에 부응하여, 본 연구는 수학 영역의 문제해결 학습을 위한 지능형 교수기 개발을 목표로하였다. 본 연구에 의해 구축된 CyberTutor는 수학 영역 문제해결 학습의 각 단계에서 학습자의 추론 수준과 요구에 부응하여 상호 작용하는 학습 환경을 제공하는 것을 가장 큰 특징으로 하고 있다. 이러한 대화식 학습을 주도하는 교수 정보는 물리, 수학과 같이 공식 원리의 이해와 응용을 학습의 주 내용으로 하는 영역에 공통적으로 사용되는 교수 전략이라 할 수 있는 학습자 가설 평가, 힌트 제공, 설명 제시 등의 전략 정보로 이루어진다. 본 교수 체계의 적성성과 실행 내용을 기하 영역의 문제해결 학습 예를 통해 제시한다. 제안 교수 체계는 기존 시스템에 비해 다음과 같은 점에서 진전된 교수 체계의 모형을 제시한다. 첫째, 증명 외에 해결 유형의 문제해결 학습도 지원하므로 실용성이 높다. 둘째, 다양한 수준의 학습자에 대한 적용을 추구함으로써 학습자 위주의 개인 교수식 환경을 지향한다. 마지막으로, 미리 저장된 것이 아닌, 실시간 문제해결 추론에 기반한 교수 대화는 학습 효과를 극대화한다

A Mathematics Tutoring Model That Supports Interactive Learning of Problem Solving Based on Domain Principles

Hyung Joon Kook[†]

ABSTRACT

To achieve a computer tutor framework with high learning effects as well as practicality, the goal of this research has been set to developing an intelligent tutor for problem-solving in mathematics domain. The main feature of the CyberTutor, a computer tutor developed in this research, is the facilitation of a learning environment interacting in accordance with the learner's differing inferential capabilities and needs. The pedagogical information, the driving force of such an interactive learning, comprises of tutoring strategies used commonly in various domains such as physics and mathematics, in which the main contents of learning is the comprehension and the application of principles. These tutoring strategies are those for testing learner's hypotheses test, providing hints, and generating explanations. We illustrate the feasibility and the behavior of our proposed framework with a sample problem-solving learning in geometry. The proposed tutorial framework is an advancement from previous works in several aspects. Firstly, it is more practical since it supports handling of a wide range of problem types, including not only proof types but also finding-unknown types. Secondly, it is aimed at facilitating a personal tutor environment by adapting to learners of varying capabilities. Finally, learning effects are maximized by its tutorial dialogues which are derived from real-time problem-solving inference instead of from built-in procedures.

키워드 : 지능형 교수 시스템(intelligent tutoring system), 수학 문제해결 학습(problem-based learning in mathematics), 대화식 학습 환경(interactive learning environment), 개별 학습(individualized learning), 공식 원리(domain principles)

1. 서론 및 연구 배경

교육 정보화에 대한 관심이 증가하면서 사용 효과가 높은 교육용 소프트웨어에 대한 수요도 증대되고 있다. 이는 컴퓨터를 교육에 활용하고자 하는 노력과 맞물려 근래에 이에 관련한 많은 연구 개발이 진행되어 오고 있다. 그리

나, 기존 대부분의 제품이나 개발 결과 가운데 아직까지 실제 학습에 직접 활용할만한 수준의 학습 효과와 실용성을 가진 것은 그리 많지 않다. 출시된 제품들의 경우, 대개 멀티미디어를 활용한 단순 주입식 영상물이나 게임과 결합된 단답식 학습물 등 영상, 오락물이 주종을 이루고 있어 학습 효과는 그다지 높다고 할 수 없다. 한편으로, 인공 지능 분야에서 지능형 교수 시스템에 관한 연구가 활발히 진행되어 다양한 결과들이 얻어지고 있다. 이 가운데는 추론, 지식 표현 등 인공 지능의 여러 기법을 교육에 응용하여 좋

* 이 논문은 1997년도 한국과학재단의 특정기초연구과제(과제번호: 97-0100-0801-3) 연구비에 의해 지원되었음.

† 종신회원 : 세종대학교 컴퓨터공학과 교수
논문접수 : 2001년 5월 9일, 심사완료 : 2001년 8월 22일

은 성과를 얻은 것들이 많다. 그러나 이들 대부분은 지식 구조나 문제해결 능력의 한계로 인해 제한된 학습 영역에서 제한된 범위의 학습 효과만을 제공하는데 그침으로써, 실용 가능한 교수기로의 확장이 어렵다는 한계를 가진다.

학습 효과와 실용성이 모두 높은 교수 체계의 개발 필요성에 부응하여, 본 연구는 수학 문제해결 학습에 활용 가능한 지능형 교수기 개발을 목표로하였다. 개발의 주요 목표로서 첫째 풍부한 문제해결 전략을 보유함으로써 실용성이 높을 것, 둘째 학습자 위주의 학습 환경을 조성함으로써 개인 교수와 같은 학습 효과를 기할 것, 셋째 시스템의 교수 전략 정보를 학습 영역 정보로부터 분리시킴으로써 과학 등 유사한 학습 영역에도 응용 가능할 것 등을 추구하였다. 그리고 이러한 연구 목표들을 성취하기 위해서 학습자 인터페이스, 추론, 문제해결, 교수 전략 등 다양한 부문에서 상당 기간의 연구가 진행되어 왔다. 부문별 연구의 진전에 따른 중간 결과들은 기 발표 논문[30,31]을 통해 소개한 바 있다. 이 중 [30]은 본 연구의 문제해결에 기반한 추론 체계 부문의 설계와 구현 내용을 제시하며, [31]은 본 연구의 학습자 인터페이스 설계 내용을 중심으로 설명한다. 이들 연구에서는 전체 시스템이 아닌 시스템의 일부 부문, 즉 추론 또는 학습자 인터페이스 부문의 설계가 연구의 초점으로 다루어졌기 때문에 교수 전략 부문은 상대적으로 단순한 수준에 머물러 있었다. 최근 교수 전략 체계의 설계와 구현이 완성됨에 따라 본 논문은 이 부문을 초점으로 하여 본 연구의 결과를 전체적인 틀에서 제시하고자 한다. 본 논문의 자체 설명성을 높이기 위해 기 발표 논문에서 다루어진 부분에 대해서도 개략적 차원에서의 최소한의 언급은 있겠지만, 보다 상세한 내용에 관해서는 기 논문을 참조하도록 문헌 번호만을 제시하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 관련 연구를 소개한 후, 3장에서 수학 문제해결 학습을 위한 정보 표현 체계에 관해 본 연구의 접근 방식을 설명한다. 4장에서 본 연구가 구현한 시스템의 실행 예를 제시한다. 5장에서는 시스템의 전체적 구성 내용을, 6장에서는 주요 모듈의 구현 내용을 설명한다. 마지막으로 7장에서는 향후 과제와 결론을 제시한다.

2. 관련 연구

수학이나 과학 영역에서의 개념과 문제해결 방법은 다른 일상 개념들보다 분명하고 정확하게 정의되어 있으나 개념 인지 과정과 문제해결 과정을 교수하는 것은 어려운 과제로 인식되어 왔다. 보다 쉽게 영역 지식의 개념과 문제해결 방법을 교수하는 방법론을 얻고자 하는 시도는 전산, 교육 및 인지 과학을 비롯한 여러 분야에서 수행되어 왔다. 그러나 많은 선행 연구들이 수학이나 과학의 특정한 분야에 국

한하여 설계된 개별적인 교수 방법론을 설명하고 있으며, 유사한 영역에 포괄적으로 적용할만한 일반화된 방법론에 대한 모색은 인지 과학 방면에서 검토된 추상적인 수준에 머무르고 있다. 이 장에서는 기존의 수학이나 과학 분야 컴퓨터 교수기 연구에서 학습 내용을 교수하는데 적용한 방법론에 대해 고찰한다.

Anderson[3, 4, 5]은 ACT* 이론을 기반으로 구현된 Geometry 및 Algebra Tutor를 소개하였다. 기하 증명기의 선구적 시스템이랄 수 있는 Gelernter[6]의 Geometry Theorem-Proving Machine의 발전된 형태랄 수 있는 이들 증명 교수기는 모델 추적(model tracing) 기법을 사용하여 학생의 인지 상태를 알아낸다[24]. 모델 추적은 문제해결에 필요한 정, 부규칙들의 집합인 수행 모델(performance model)을 이용하여 문제에 따른 학생의 수행 능력을 해석한다. 학습 모델(learning model)은 학생의 지식 상태가 문제해결의 각 단계 이후에 어떻게 변화되는지에 대한 가정들로 구성되어 있고, 문제간에 발생하는 학생의 지식 변화를 추적하는데 사용된다. 추적된 정보는 모델 추적에서 해석에 대한 모호성을 제거하는데 사용되거나 학습을 최적화하는 문제를 선택하는데 사용된다. Anderson의 교수기는 수학 영역의 증명 유형의 문제에 관해 미세한 추론 경로에 의해 제어된 학습을 제공한다는 장점을 가진 하지만, 교수기-학습자간 상호 작용의 단위가 되는 추론 경로의 길이가 미리 지정되어 있어 증명 외의 학습에 활용되기에는 제약이 가진다. 즉, 비약적 추론 경로를 종종 수반하는 문제해결 유형의 학습에서 현재 방식의 모델 추적은 적용이 어렵기 때문이다.

Statics Tutor[29]는 학생에게 정역학(statics) 문제를 제시하고 학생으로 하여금 힘 벡터나 토크(torque)를 찾아서 정역학적 평형 상태를 해결하도록 한다. Statics Tutor는 학생의 오류에 대해서 직접, 그리고 간접적인 표현 방법을 모두 제공한다. 특히, 모의 실험을 통한 그래픽 피드백 기능은 '회고(reflection)'[11]를 상당히 지원하고 있다. 이와 유사한 물리 영역의 ITS로서 Crane Boom Tutor[28]를 들 수 있다. Crane Boom Tutor는 영역 지식의 선연적 지식에 해당하는 영역내의 개념들과 상관 관계를 객체 프레임 형태로 표현하고 각 객체에 따라 정의된 객체 아이콘으로 영역내의 현상들을 모의 실험할 수 있게 한다. Statics Tutor나 Crane Boom Tutor는 학생의 오류에 대해서 다양한 설명 방식을 제공하고 있으나, 문제해결 과정에서 학생의 추론에 대한 단계적 접근은 허용하지는 않는다. 본 연구에서 제안한 교수 모듈은 학생의 논리적 비약을 허용하지만 이와 함께 단계적 문제해결 방식도 지원한다. 이 방식은 초심자에 특히 유용한 교수 전략으로서 학생이 직접 논리적인 사고를 할 수 있도록 유도한다.

SOPHIE[7, 8, 25]는 주입식 지도에 의한 교육보다는 학생

이 자신의 생각을 시도해 봄으로써 문제를 해결하는 기법을 터득하게 하는 교육 환경을 제공한다. 학생의 질문에 대답하기 위해서 SOPHIE는 가설 평가, 가설 생성 등과 같은 교수 행위를 구사한다. 교수 행위를 위한 추론은 범용 모의 실험기를 사용하여 행한다. SOPHIE의 설계자 DeKleer는 자신의 문제해결에 사용되는 인지 모델의 인과 구조는 구성 원소간의 연결성을 분석한 위상 구조에서 추론될 수 있다고 설명한다. 이런 가정에 의하면 이 인과 관계란 단지 논리적으로 이해될 수 있는 것으로, 구조에 포함된 기능에 대한 원리는 배제된다. 이 가정이 위배되는 경우에 DeKleer는 수학적인 처리를 통해 문제를 해결하는데, 이런 방법은 인지 차원에서 모호하다고 볼 수 있다[4]. 따라서 한층 인간의 사고와 밀접한 모델로 발전될 필요가 있고 특히 문제 해결 모델이 학습에 도입된다면 그 필요성은 가중된다.

White[26, 27]는 질적 모델을 기반으로 한 문제해결 모델을 교육에 적용하였다. 교육 방식은 저항, 기전력 등 기본 회로 원리를 가르치고, 회로의 한 부분이 변하면 다른 부분에 오는 변화를 예측하도록 한다. White 전략의 특징은 이른바 '열린 학습' 형태를 취한다. 즉, 자연스러운 실험 환경 및 설명 제공, 문제 유도 학습, 그리고 학생 주도 학습(즉, 학생이 스스로 무슨 문제를 풀고, 어떤 전략을 쓸지 결정함)이다. White가 제시한 교육 환경은 기존의 모의실험 방식에서 나타난 단점을 상당히 줄이고 시스템 내부의 추론 과정을 학생에게 설명할 수 있도록 한다[4]. 그러나 앞 절에서 언급한 바와 같이 White의 질적 모델을 기하, 물리 등의 학습 영역에까지 확대 적용하기 위해서는 질적 모델의 한계를 극복하고 활용성을 높이기 위한 구체적 방안에 대한 심층 연구가 필요하리라 본다.

Lelouche[18]는 수학, 과학, 공학 등 '정밀 과학(exact science)' 영역의 문제해결 교수기를 구축하기 위한 이론적 모델을 제시하였다. 그는 교수기가 가져야 할 세 부분의 필수 지식 요소로서 학습 영역 지식, 문제해결 지식 그리고 교수 지식을 정의하고, 이들 각각이 독립적으로 모델링되고 연결되어야 전체 교수 시스템의 학습 성능을 고취시킬 수 있다고 주장하였다. 그의 이론은 본 연구에서 구축된 시스템이 영역 지식, 문제해결 규칙들 그리고 교수 모듈의 세 부분의 독립 모듈로 구현된 점에서 본 연구의 설계 원리와 상당히 일치하고 있다. 또한 Lelouche는 이들 지식을 다양한 레벨의 추상성(abstraction)과 복잡성(complexity)으로 분류하고 그 차별성을 학습자와의 질의 응답, 설명, 힌트 등에 활용함으로써 학습기의 사용자 친숙성과 학습 효과를 높일 수 있다고 보았다. 이 같은 관점은 본 시스템의 교수 모듈에서 구사하는 학습자 능력에 따른 차별적 힌트나 설명 제공 전략에서 구현된 내용과도 통하는 점이 있다고 하겠다. 그러나 Lelouche의 연구는 문제해결 교수기의 일반적 설계 원리만을 제시할 뿐, 추상성과 복잡성 등의 요소가 실제 어떻게 구현

될지에 대한 구체적 방법론은 제시하지 못한다는 한계를 가진다.

근간에 연구 개발되는 CAI의 특기할 경향 가운데 하나는 인터넷의 발달과 함께 기하 등을 포함하는 학습을 웹 상에서 구현하고자 하는 시도이다. 이들 대부분[12, 21]은 웹 상에서 자바 애플릿 등을 통해 학습 내용을 상호작용 방식의 다이어그램으로 제시하는 방식을 취함으로써 종래의 교수기에 비해 시각적 효과나 상호 작용성에서 진일보한 형태의 교수 모형을 제시하고있다. Chen[10]은 중등 평면기하 학습의 개념과 관계 원리들을 계층구조화하고 자바 애플릿을 사용하여 학습 내용을 데이터베이스화하였다. 웹 상에서 학습자에게 제시되는 특정 기하원리의 텍스트는 자바 애플릿에 의한 다이어그램과 함께 제시된다. 학습자는 해당 원리의 충분한 이해를 위해 다이어그램 상의 일부 객체를 움직여 볼 수 있으며 시스템은 여기에 적절히 대응한다. Hwang[13]의 VPL(Virtual Physics Lab)은 연속 신호에 의한 교통 흐름 조절이나 속도에 따른 자동차 안전거리 유지 등에 관한 kinematics 원리 이해 및 연습을 가능케 웹 학습 시스템이다. 이 역시 주요 화면이 자바 애플릿에 의한 다이어그램으로 제공되며 학습자는 다이어그램 상에서 신호 지속 시간, 자동차 속도 등의 변수값 변화를 시도하고 시스템은 이에 대응하는 결과를 다이어그램에 보여주는 방식의 상호 작용을 지원한다. 이들 연구는 다이어그램 보조 학습이라는 점에서 본 연구와 맥을 같이 하고는 있으나, 본 연구에서와 같은 실전 학습 문제해결에 대한 지원은 하지 않는다.

Alpert[1]의 AlgeBrain 역시 웹 상에서 구현되고 있는 초급 대수(1원의 선형방정식) 문제해결 교수기지만 앞서의 두 예와 달리, 학습자에게 선형방정식을 제시하고 학습자가 이를 해결하는 과정의 각 단계에서 다양한 교수 행위를 제공한다. 여기에는 학습자의 옳고, 그른 행위에 대한 찬성, 반대 표시와 (요구가 있을 시) 힌트 및 도움말 제공 등이 포함된다. 이러한 교수 행위를 얻기 위해 시스템이 사용하는 전문가 모듈은 기존의 많은 지능형 교수기와 마찬가지로 학습 영역의 일반 원리를 표현하는 규칙들로 구성된다. 따라서 본 연구가 설계한 교수기와 유사한 시스템 구성과 교수 행위를 제공한다고 할 수 있으나, 몇 가지 점에서 차이가 있다. 학습자의 입력이 옳고 그름을 판단하기 위해 시스템은 현재 적용 가능한 원리 집합과 비교하여, 이 가운데 있는 것이면 옳고, 그렇지 않으면 그른 것으로 간주한다. 여기에 반해 본 연구의 교수기는 학습자 입력을 증명 시도하여, 증명될 경우 옳고, 그렇지 않으면 그른 것으로 간주한다. 이러한 차이는 AlgeBrain의 문제해결이 단일 단계의 정방향 추론에 의존한다는 점에 기인하며, 결과적으로 학습자의 추론 단계 비약(logical jump)을 허용하지 않게 된다. 이에 반해 본 연구의 경우 다단계 추론에 의한 비약을 허용함으로써 상이한 수준의 학습자 문제해결을 지원한다. 또 하나의

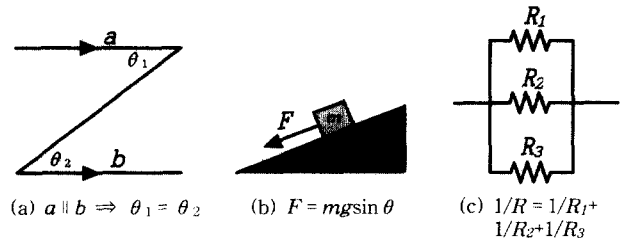
비교할 점은 힌트 선정 방식이다. 문제해결의 어떤 시점에서 과연 어떤 힌트가 최선인가 하는 것은 논란의 여지가 많다고 할수 있으며 따라서 시스템마다 접근을 달리할 수 있는 부분이기도 하다. **AlgeBrain**에서는 힌트 요구 시점에서 정방향으로 적용 가능한 원리 가운데 단순히 목표와의 거리가 가장 가까운 것을 힌트로 선정함으로써 문제해결에 직접적 도움이 되도록 추구는 하나, 힌트 선정에 있어서 학습자 수준과의 부합성은 고려하지 않는다. 이에 반해 본 연구에서는 역방향에서 즉, 최종 목표를 얻기 위해 적용 가능한 원리 가운데 학습자 수준에 가장 적절한 것을 힌트로 선정함으로써 결과적으로 문제해결에 직접적 도움이 되는 동시에 학습자의 수준이 고려된 힌트 제공이라는 두 가지 목적을 함께 달성하는 힌트가 제공될 수 있도록 추구한다.

3. 수학 문제해결 학습을 위한 정보 표현 체계

이 장에서는 본 연구에서 개발한 교수기 **CyberTutor**의 설계에 관련한 원리와 개념을 논한다. 수학 영역의 학습은 공식 원리의 이해와 습득을 주 내용으로 한다. 제시된 원리는 다양한 난이도의 문제해결을 통해 학습되어진다. 이 같은 수학 영역 학습의 기본 체계는 **CyberTutor**의 정보 표현 체계에 그대로 반영된다. 이어지는 절에서는 다양한 수학 영역의 공식 원리들과 문제 및 문제해결 절차의 표현에 관해 본 연구가 어떻게 접근하는 지에 관해 설명한다.

3.1 공식 원리

먼저, 수학 영역의 교수기가 필요로 하는 정보 가운데 하나는 공식들(principles)로서, 이들은 해당 학습 영역의 문제 해결에 사용될 원리들의 집합이다. 수학뿐 아니라 과학 영역에서도 자주 학습의 대상이 되는 이들 원리들은 상황론적 명제의 형태를 취한다. 즉, "상황 S에서 명제 P가 성립한다" 형식을 말하는데, 여기서 S는 현 상황에 등장하는 객체나 객체간의 구조적 관계를, P는 객체간 또는 객체 관련 값들에 관한 수리적 제약을 주로 나타내는 명제로서 단일 명제 또는 조건 명제의 형식을 취한다. (그림 1)은 다양한 수학 및 과학 영역의 공식 원리의 예를 보인다. (그림 1a)는 기하에서 "두 개의 선분과 이들을 가로지르는 또 하나의 선분이 주어진 경우(S), 원래 두 선분이 평행이면, 이들을 가로지르는 선분에 의해 생긴 양 엇각, θ_1 과 θ_2 의 크기는 같다(P)"는 원리를, (그림 1b)는 물리에서 "기울기 θ 의 경사면 위에 놓인 질량 m 의 물체가 있다(S). 이 경우 물체가 경사면과 나란한 방향으로 받는 힘의 크기는 $mg\sin\theta$ 이다(P)"는 원리를, 그리고 (그림 1c)는 전기 회로에서 "저항값이 R_1, R_2, R_3 인 세 개의 저항이 병렬로 연결되어 있다(S). 이 경우 전체 저항값의 역수, $1/R$ 은 각 저항값의 역수의 합, $1/R_1+1/R_2+1/R_3$ 과 같다(P)"는 원리를 보인 것이다.



(그림 1) 공식 원리

다수의 공식 원리들은 학습 저자(예 : 교사)에 의해 시스템에 초기 공급되어야 하며, 문제 유형의 확장에 따라 필요하다면 추가 공급될 수도 있어야 할 것이다. 그러나, 대개 컴퓨터 비전문가랄 수 있는 학습 저자에게 시스템의 복잡한 문법 체계에 의해 공식들을 입력하도록 요구하는 것은 비현실적인 발상으로 교수기의 사용성을 극히 저하시키는 요인이 될 수 있다. 이 점에 착안하여, **CyberTutor**는 학습 저자에 의한 공식 원리 입력이 손쉬운 방식에 의해 이루어지도록 하는 저자 인터페이스 체계를 지원함으로써 외부에서 공급되어야 할 지식 기반의 구축 및 확장을 용이하게 지원한다. 저자 인터페이스 설계와 구현 내용에 관한 상세한 설명은 [31]에서 참조할 수 있다.

3.2 문제 및 해결 절차

문제해결 학습을 위한 교수기가 다루어야 할 또 다른 정보는 문제들(problems)인데 대개의 컴퓨터 교수기가 문제를 내부적으로 미리 저장하는데 반해 본 교수기는 이를 외부 정보로 취급한다는 특징을 가진다. 학습 영역에 종속적인(domain-dependent) 정보인 문제들을 수학 문제해결 학습 공통인(domain-independent) 교수 정보로부터 분리, 외부에서 공급되도록 하는 것은 시스템을 세부 모듈화하는 단순한 의미 이상의 몇 가지 중요한 의의를 포함한다. 첫째 문제를 내부적으로 미리 저장하는 방식은 내장된 문제 집합 외의 문제에 대한 학습은 전혀 고려치 않은 방식이므로 문제해결 학습의 범위가 문제 데이터베이스의 범주에만 제약 받는다는 단점이 있다. 기존 대부분의 문제해결 학습용 교수기가 학습자가 일회 또는 수회 사용 후 흥미를 잃게 되는 것도 바로 이 때문이다. 반면, **CyberTutor**에서는 시스템이 외부 파일 형태로 제공하는 문제 집단 외에도 사용자(학습자)가 공급한 문제에 대한 학습까지 지원할 수 있어서 기존의 교수기와는 비교할 수 없이 높은 사용성을 가진다. 한편, 손쉬운 방식에 의한 공식 입력을 필요로 하는 학습 저자를 위한 인터페이스와 마찬가지로 형태의 문제 입력 인터페이스가 제공됨으로서 학습자의 문제 입력을 손쉽게 함은 물론 학습의 흥미를 유발시킨다.

문제를 외부 정보화하는 두 번째의 의의는 개별 학습(personalized learning)을 가능하게 한다는 점이다. 기존 문제해결 학습용 교수기는 문제 집단을 문제해결과 함께 미리 저

장하는 방식을 취한다. 단순한 객관식 문제의 경우는 해답을, 다소 절차적 해결을 요구하는 문제의 경우 해당 해결 절차를 저장하는 것이다. 하지만 이같은 설계 방식으로는 외부(학습자 등 교수기 사용자)로부터의 새로운 문제를 받아들이지 못하는 것은 물론이고, 학습자 능력에 따라 개별화된 교수를 제공할 수도 없다. 누구에게나 똑같이 정해진 해결 절차만이 제공될 뿐인 것이다. 본 교수기가 문제를 외부 정보로 취급한다함은 입력된 문제에 대해 실시간에, 학습자 추론 수준에 적응하며 교수하는 체계로 설계되었음을 말한다. 실제, 수학 문제해결 학습에 있어서 학습자에 따라서는 시행 착오 등 다소 부진한 경로를 통해 해결에 이른다거나, 반대로 비약적인 절차를 통해 해결을 성취하는 등 다양한 수준의 사용자가 존재한다. 이 때문에 문제해결 학습용 교수기의 성패는 개별 학습의 지원 여부와 정도에 달려있다고도 말할 수 있는 것이다.

실시간 교수 방식으로부터 얻는 부수 효과로는 해결 절차 저장을 위한 기억 장소 공간의 절약을 들 수 있다. 아주 단순하지 않은 수학 문제의 경우 학습자 수준차에 의한 다양성은 차치하고 원천적으로 단일이 아닌 복수의 해결 절차를 가지는 경우가 많다. 문제 및 해결 절차 내장식 교수기의 경우 문제마다 다양한 해결 절차를 모두 저장해야 하는 부담은 시스템의 확장에 큰 장애가 된다. 우회 전략으로서 이 가운데 저자가 추천하는 단일의 해결 절차만을 저장하여 교수하는 것은 교수기의 유연성을 크게 제약하는 결과만 초래할 것이다.

4. 대화식 학습 실행 예 : 초급 평면 기하

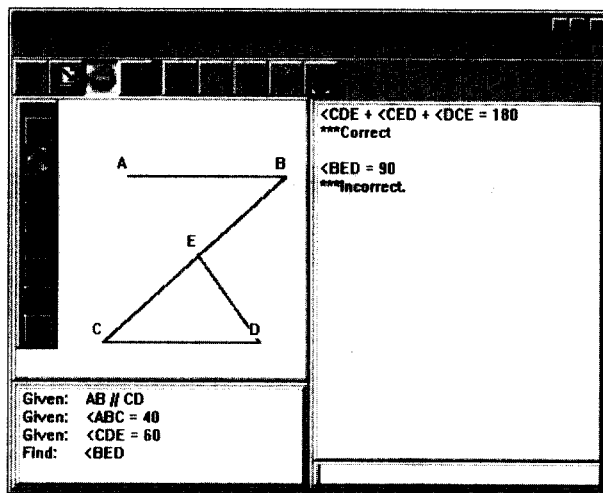
CyberTutor의 교수 행위는 내부적으로 구축된(built-in) 정보 체계가 외부에서 공급된 지식을 적절히 해석, 처리함으로써 발생한다. 내부 구축된 지식 체계는 본 교수기의 코어에 해당하며 수학 문제해결 학습을 실제 수행할 교수 정보로 구성된다. 교수 정보 모듈은 앞에서 설명한 바와 같이 외부 공급된 영역 공식 원리와 문제를 활용하여 학습자의 문제해결을 대화식으로 지원하는, 즉 영역에 상관없이 수학 문제해결 학습에 공통적으로 적용 가능하도록 설계된 모듈이다. 구체적으로는 학습자 문제해결의 세부 단계에서 학습자의 해결 절차를 감독, 추적, 조연하는 양식으로 작동한다. 본 연구에서는 제안된 교수 체계의 적정성을 평가하기 위해, 수학 영역 가운데 하나인 초급 수준 평면 기하의 공식과 문제를 실험적으로 구축하고 이를 본 교수 체계와 연계 수행하였다. 이 장에서는 이 가운데 하나의 학습 실행 예를 제시한다.

CyberTutor는 공급된 기하 공식 베이스를 사용하여, 학습자가 시스템의 문제 베이스로부터 선택한 또는 인터페이스를 통해 새로이 입력한 문제를 중심으로 대화식 학습을

진행한다. (그림 2)는 기하 문제가 주어진 초기 화면을 보인다. 그림의 왼편 아래 창에 보였듯이, 문제는 일련의 주어진 조건들과 구하고자 하는 미지수 또는 증명되어야 할 명제로서 구성된다.

학습자는 주어진 문제해결에 관해 몇 가지 선택을 할 수 있다. 학습자 가설 입력, 힌트 요구, 그리고 힌트에 대한 설명 요구가 그것이다. 먼저, 학습자 가설 입력은 학습자가 문제해결을 위한 한 단계의 명제를 입력하는 것을 말한다. 교수기는 학습자의 가설이 성립하는 것인지를 평가하여, 옳고 그름에 관해 긍정 또는 부정으로서 응답한다.

학습자의 문제해결은 각자의 수준에 따라 상이한 경로를 진행하며 학습자에 따라 다양한 수준의 논리적 비약을 수반하기도 한다[5, 15, 17]. 이에 대해 CyberTutor는 미리 저장된 해결 경로가 아닌 실시간 추론에 의하여 학습자의 가설을 평가한다. 따라서 기존의 많은 교수기처럼 시스템의 정해진 해결 경로를 벗어나는 학습자의 정당한 추론이나 논리 비약을 허용치 않고 시스템의 해결 방식만을 그대로 학습하도록 강요하는 우를 범하지 않는다. 또한 학습자의 오류에 대해 즉각적으로 개입, 구체적인 도움말을 제공함으로써 자율적인 문제해결을 저해하기보다는, 요청이 없는 한 학습자의 해결 과정에 대해 평가하고 지켜보는 방식을 취함으로써 시행 착오를 통한 학습이 이루어지도록 한다. (그림 2)의 오른쪽 창에는 문제에 대한 학습자 가설과 이에 대한 교수기의 평가를 보인다.

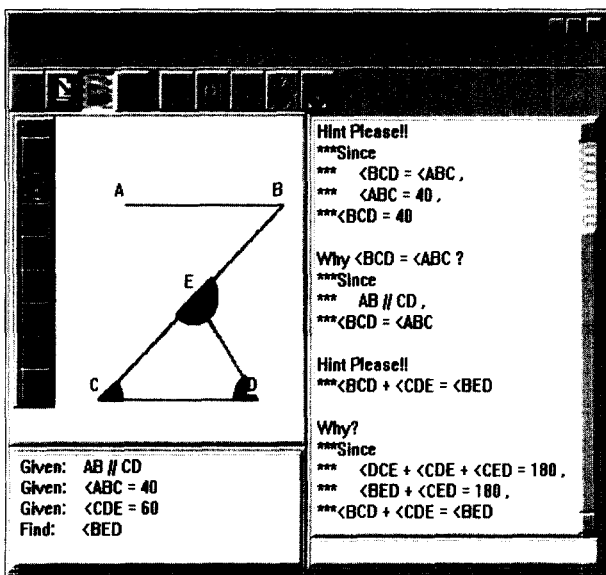


(그림 2) 학습자 가설에 대한 평가

힌트 요구는 학습자가 자신이 입력한 가설이 교수기에 의해 부정되거나, 문제해결에 관해 더 이상 진행이 어렵거나 할 때 취할 수 있는 선택이다. 문제해결 학습용 교수기에 있어서 적절한 힌트를 제공하는 것은 어려운 과제라고 할 수 있다. 힌트의 내용이 너무 쉬운 수준이거나, 학습자가 이미 인지하고 있는 원리에 의한 것이거나, 문제해결에

직접 도움이 안 되는 것일 경우 힌트로서의 가치가 적거나 없다고 할 수 있다. CyberTutor는 이런 점들을 모두 고려한 힌트를 생성, 제시한다.

마지막으로, 힌트에 대한 설명 요구는 교수기가 제시한 힌트에 대해 학습자가 설명을 요구하는 것이다. 이 같은 부가 설명 기능은 제시된 힌트가 일반적인 것으로 끝나지 않고 연이은 학습의 재료가 될 수 있도록 보장하는 효과를 가진다. 구체적으로, CyberTutor에서의 힌트는 특정 공식에 기반하여 "P임"과 같은 무조건 형식, 또는 " P_1, P_2, \dots, P_N 이므로, P임"과 같은 조건 형식 가운데 하나로 주어진다. 주어진 힌트에 대하여 학습자는 어떻게 그러한 유추가 가능한지 묻거나(Why?), 혹은 $P_i(1 \leq i \leq N)$ 가 어떻게 성립되는지(Why P_i ?) 물을 수 있다. 전자의 'Why?'에 대해 교수기는 '우회 설명(roundabout explanation)', 즉 현 힌트에 사용된 공식 원리를 사용하지 않고 동일한 결론(P)에 이르는 설명을 제공한다. 우회 설명이 존재하지 않는 경우에는 학습자에게 현재의 공식을 그대로 학습할 것을 권한다. 하나의 해결 경로만을 미리 저장하는 교수 체계에서 이 같은 우회 설명은 근본적으로 불가능하다고 하겠다. 다음, 후자의 'Why P_i ?'는 명제 P_i 의 성립 근거에 대한 설명 요구이다. 이에 대해 교수기는 '근거 설명(grounds explanation)', 즉 조건 형식의 명제 " R_1, R_2, \dots, R_N 이므로, P_i 임"를 학습자에게 제공함으로써 설명한다. 그러한 명제가 없을 경우 교수기는 학습자가 현재의 P_i 공식을 그대로 학습할 것을 권한다. 학습자는 교수기의 설명에 대해 충분히 이해가 될 때까지 재귀적으로 'Why?' 또는 'Why P_i ?' 설명 요구를 할 수 있다. (그림 3)은 현 문제에서 'Why? P' 및 'Why?'에 대해 교수기가 근거 설명 및 우회 설명을 각각 제공한 예를 보인다.

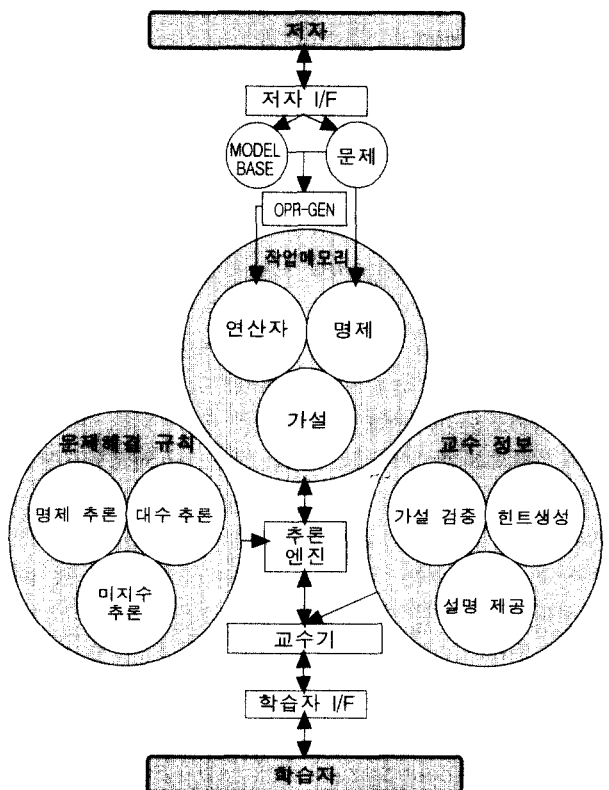


(그림 3) 힌트 및 설명 제공

CyberTutor의 학습자 인터페이스는 학습자와의 상호 작용에 있어서 여러 가지 사용자 친숙한 환경을 제공함으로써 학습의 효과를 높인다. 기하 영역에서 구현된 학습자 인터페이스의 경우, 그림에 보인 것처럼 문제 다이어그램의 객체들 가운데 현재 대화의 문맥에 해당하는 객체들을 시각적으로 강조한다거나, 마우스에 의한 선택을 가능케 하는 등이 그 예이다.

5. CyberTutor 전체 구성

이 장에서는 수학 문제해결 학습용 교수기 CyberTutor 시스템의 구현 내용을 제시한다. (그림 4)에 보인 CyberTutor의 전체 시스템 구성도를 설명하면 다음과 같다. CyberTutor는 학습 저자로부터 학습 영역의 공식들과 학습 문제를 공급받아 공식 베이스(Model Base)와 문제 베이스(Problem Base)를 각각 구축한다. 각 공식과 문제는 일련의 명제(propositions)로 표현된다. CyberTutor에 의한 문제해결 학습은 문제 베이스에서 선택되거나, 혹은 학습자에 의해 제공된 새로운 문제를 대상으로 진행된다. 학습 문제가 명제의 형태로 시스템에 로드되면, 해당 문제의 해결에 관련하는 연산자(operators)들이 OPR-GEN(Operator-Generator)에 의해 자동 생성된다. 이 연산자들은 <좌변>=><우변>의 rewriting rules 형태를 취하며, 공식 베이스 내의 일반적인 공식 원리들 중 현



(그림 4) CyberTutor 시스템 구성도

문제에 관련된 것들만이 현 문제의 문맥으로 개별화(instantiated) 형태를 취한다. 현재 기하 영역에서 저자나 학습자의 공식 또는 문제 입력이 손쉽게도 효율적으로 수행될 수 있도록 하는 저자 인터페이스가 지원되고 있다. 인터페이스에 관한 상세한 설명과 이로부터 생성되는 공식 및 문제 명제, 그리고 공식 베이스를 기반으로 작업 메모리에 생성되는 연산자들에 대한 생성 과정과 세부 내용에 관해서는 [31]을 참조할 수 있다.

CyberTutor의 추론 엔진은 규칙 메모리라 할 수 있는 문제해결 규칙 베이스에 기반하여 앞서 설명한 작업 메모리의 내용을 해석한다. 여기에는 기존의 명제를 읽거나, 새로운 명제를 쓰거나, 명제의 일부 내용을 수정하는 행위가 포함된다. 문제해결 규칙들은 수학 공식 원리 문제해결 학습이 공통적으로 수반하는 문제해결의 절차 전략들로 구성되는데, 크게 명제(propositions)에 관한 추론 규칙들과, 대수(algebra)에 관한 추론 규칙들, 그리고 미지수(unknowns) 해결에 관한 추론 규칙들의 세 부분으로 이루어진다. 이들 세 부분의 규칙들은 교수기의 엄밀한 제어 하에 문제해결 학습 진행의 세부 단계에서 문제의 단계적 해결을 시도하고 그 결과를 교수기로 반환한다. [30]는 본 추론 체계에 관한 동적 특성에 관해 상세히 설명한다.

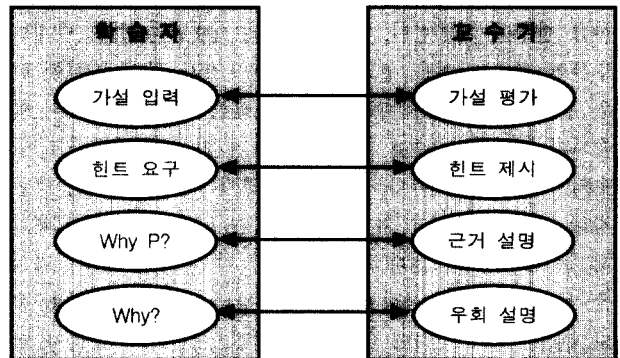
CyberTutor의 핵심이랄 수 있는 교수기는 문제해결 학습의 세부적 단계에서 교수 행위를 제공한다. 교수를 위해서 필요한 문제해결에 직접 관련된 정보는 추론기를 구동시킴으로써 얻는다. 구체적으로, 교수기는 추론 엔진을 동적으로 제어, 즉 교수기의 필요에 따라 추론기를 부분 구동하는 방식으로, 문제해결의 각 단계에서 필요한 만큼의 정보를 얻는다. 따라서 추론기는 하나의 문제에 대한 학습의 시작과 함께 작업 메모리가 초기화되면서 시동(start)되어 그 문제에 대한 학습이 계속되는 동안 재개(resume) → 추론(inference) → 결과반환(return) → 정지(pause) 사이클을 되풀이하면서 작업하게 된다. 교수기의 제어 하에 이루어지는 추론 사이클의 일례를 들면, 학습자의 입력 명제를 가설(hypothesis)로 설정하여 이를 추론기에 전달, 추론을 재개시키고, 추론기가 반환한 결과에 따라 그 학습자 명제를 평가하는 것이다. 본 교수기가 채택한 재개(resume) 방식의 추론기 제어가 일상적인 추론기 제어 방식인 재시동(restart) 방식과 크게 다른 점은 추론 사이클간의 갱신된 작업 메모리 내용을 그대로 유지한다는 점이다. 작업 메모리는 항상 현 시점에서의 학습자 문제해결 학습 경로를 반영하게 된다. 따라서 재시동 방식에 의해 처음부터 다시 추론을 시도할 경우에 야기될 복잡성과 중복 계산에 따른 시간 낭비를 배제함으로써 대화식 교수기의 시간 성능을 크게 향상시킨다.

CyberTutor의 교수 행위는 모두 내부 구축된 교수 정보로부터 발생한다. 교수 정보는 공식 원리에 기반한 문제해결 학습 공통의 교수 전략들로 구성된다. 이는 크게 세 부

문으로 나누어지는데 가설 평가 전략, 힌트 생성 전략, 그리고 설명 제공 전략이다. 본 교수기의 학습 성능을 결정한다고 말할 수 있는 각 교수 전략의 구현 및 작업 내용은 이어지는 장에 상세히 설명한다. 참고로 CyberTutor는 CLIPS [32]에 의해 작성된 추론 부분이 Visual C++에 의해 작성된 인터페이스 및 교수기와 연동하여 Win9X상에서 동작하도록 구현되었다.

6. 개별 학습을 위한 교수 전략

CyberTutor는 학습자에게 제시된 문제의 해결에 관하여 대화식 학습 환경을 제공한다. 교수기는 학습자가 선택한 문제를 로드함과 동시에 추론기를 시동한다. 이 시점에서부터 학습자와 교수기간의 상호 작용은 구체적으로 (그림 5)의 흐름도에서 보는 바와 같이, 학습자로부터의 가설 입력, 힌트 요구, 근거 설명 요구, 그리고 우회 설명 요구에 대해 교수기가 각각 학습자 가설 평가, 힌트 제시, 근거 설명 제공, 그리고 우회 설명 제공으로서 반응하는 양식으로 진행된다. 이어지는 절에서는 각각의 교수 행위에 관한 CyberTutor의 설계 내용을 설명한다.



(그림 5) 학습자-교수기 상호 작용

6.1 학습자 가설 평가

문제해결의 어느 시점에서나 학습자는 가설, 즉 문제해결을 위한 단위 명제를 입력할 수 있다. 학습자 가설에 대해 교수기는 추론을 재개시킨다. 이때 추론기는 문제해결 규칙들을 적용하여 이를 증명 시도한다. 추론의 결과는 가설에 대해 진(true), 위(false), 항진(always true), 항위(always false), 확인불가(unknown) 등으로 반환된다. 이 가운데 진(true)인 경우, 교수기는 학습자에게 'correct' 메시지를 제시하고, 해당 명제를 학습자가 '알고' 있는 것으로 기록한 후, 학습자와의 대화를 계속한다. 이 같은 학습자 정보는 추후 힌트 제공시 동일 원리에 의한 명제가 가지는 힌트의 우선 순위를 낮추도록 하는 역할을 한다. 위(false)는 학습자 가설이 작업 메모리의 어떤 명제와 모순을 일으킨 경우에 발생한다. 해당 명제는 현 추론 사이클 재개 이전부터

이미 존재했거나 또는 재개 이후 가설 평가 과정에서 생성된 것일 수도 있다. 어떠한 경우에는 교수기는 'incorrect' 메시지를 제시하고 학습자의 반응을 기다린다. 학습자는 이에 대하여 정정된 가설을 입력하거나, 다른 각도에서 문제해결을 시도하거나, 힌트를 요구하는 등 다양한 반응을 할 수 있게 된다.

학습자 가설 평가 결과 현 공식 원리와 무관한 항진 또는 항위임이 판명된 경우에는 'always true' 또는 'always false'에 해당하는 메시지를 각각 제시함으로써 학습자의 주의를 환기시킨다. 마지막으로 확인불가(unknown)는 추론의 가능한 모든 수단을 동원하였음에도 불구하고 진위를 판명할 수 없는 경우에 발생한다. 이는 수학 문제해결 학습에서 문제해결의 적절한 경로를 이탈한 학습자 가설인 경우에 주로 나타난다. 예를 들어 특정 변의 길이와는 상관없이 해결되는 기하 문제인데 변의 길이에 대한 학습자 가설이 주어진 경우가 이에 속하며 추론기는 이 가설의 진위에 대해 확인할 길이 없는 것이다. 따라서 교수기는 '확인 불가하며 현 문제해결과 관계없음' 메시지를 제시하고 학습자의 반응을 기다린다.

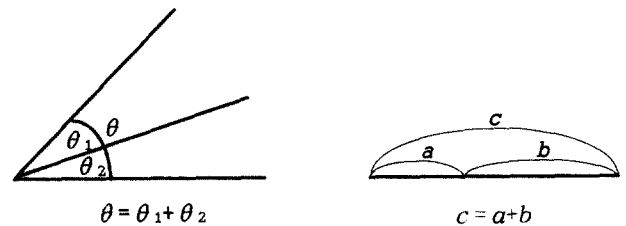
CyberTutor 교수기의 학습자 가설 평가는 학습자의 추론 수준에 동적으로 적용하는 메커니즘을 지원함으로써 개별 학습 환경을 제공한다. 그 요인은 문제해결을 위한 단계별 추론의 경로 길이를 놓고 볼 때, 교수기가 정한 일정한 단위 길이를 사용자가 따르도록 강요하는 형식이 아니라, 학습자가 자신의 수준에 따라 자유롭게 가설을 입력하고 교수기가 여기에 따라가도록 설계되었기 때문이다. 즉, 추론 경로 길이는 학습자 가설이 내포하는 크기에 의해 그때그때 동적으로 결정된다. 내부적으로, 저수준의 학습자 가설은 시스템의 추론 사이클로 볼 때 하나 또는 적은 경로 수의 추론으로써 추적이 완료된다. 한편, 다단계의 논리나 암산에 의한 비약이 반영된 고수준의 학습자 가설에 대해서는 여러 경로 수의 추론 사이클을 소모하여 추적한다. 이에 반해, 기존의 증명 교수기는 비약을 전혀 허용치 않고 시스템에 의해 미리 정해진 해결 단계만을 지원하도록 설계되어 있다[2]. 이러한 설계 원리는 기존의 증명 시스템을 보다 자유로운 추론을 허용해야 하는 문제해결 학습 시스템으로 확장하기 어렵게 한다.

6.2 힌트 제시

CyberTutor 교수 체계의 중요한 특성은 미리 정해진 해결 경로가 존재하는 것이 아니고 학습자마다 상이한 해결 경로에 동적으로 적용한다는 점이다. 따라서 학습자의 다양한 해결 과정이 성공적인 것으로 귀결될 가능성을 고려하여 시행 착오가 있더라도 이에 즉각 개입하지 않는 원칙을 가진다. 다만 학습자 가설이 주어질 때마다 이에 대한 평가를 실시하여 최소한의 진위 정보를 제공함으로써 불개입에

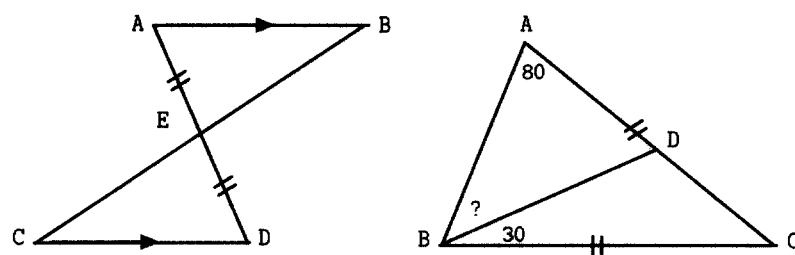
다른 학습 효율의 지나친 저하를 방지한다. 그럼에도 불구하고 학습자가 문제해결에 관해 더 이상 진행이 어려운 시점에는 교수기에게 힌트를 요구할 수 있다. 이때 교수기는 다음과 같은 원리로 힌트를 생성하여 학습자에게 제시한다.

문제해결의 어떤 시점에서 어떤 힌트가 가장 적당한지 결정하는 것은 난해한 과제라고 할 수 있다. 각 문제마다 힌트 집합을 미리 정의, 저장하는 방식은 유연성이 결여되어, 학습자의 수준이나 힌트 요구시의 문맥에 잘 적용하는 힌트를 생성하기 어렵다. 이런 점을 고려하여 교수기는 힌트 생성을 동적으로 수행한다. 힌트 생성을 위한 조건으로서, 최종 목표 해결에 직결될 것과, 이미 '알려진(known)' 사실 또는 '하찮은(trivial)' 원리를 배제할 것 등 두 가지가 고려된다. 최종 목표 해결에 직결되어야 한다는 의미는 문제를 최종적으로 해결하는 명제의 지지 명제들(supporting propositions) 가운데 하나일 것을 말한다. '알려진' 사실이란 초기에 주어진 명제나, 현 시점까지의 문제해결 과정에서 학습자 가설로서 평가되었던 명제이거나, 힌트로서 이미 주어졌던 명제들을 말한다. '하찮은' 원리란 (그림 6)에 보인 것처럼 수학 공식이라기 보다는 상식에 가까운 원리들을 말한다. 만약 이러한 조건을 만족하는 명제가 존재하지 않는 경우라면, 이것은 문제의 최종적 해결을 목전에 두고 마지막 한 단계의 유추만을 남긴 상태이므로, 해당 유추를 힌트로 제시한다.



(그림 6) 하찮은 원리의 예

이제 힌트 생성의 과정을 구체적으로 설명한다. 힌트가 요구되면 교수기는 우선 문제의 최종적 해결을 시도한다. 이를 위해 문제에서 최종적으로 구하고자 하는 미지수를 목표 미지수로 하여, 또는, 증명 유형의 문제인 경우, 문제의 증명 목표를 목표 가설로 하여, 추론을 재개한다. 추론기는 앞서 목표 미지수 또는 목표 가설에 관한 추론을 통해 이를 만족하는 명제를 생성, 작업 메모리에 추가한다. 이전의 학습자 가설 평가 과정에서 추론기는 종종 문제해결을 완료하기도 하는데, 이런 경우 최종 명제가 이미 작업 메모리에 존재하므로 이러한 추론 과정이 대부분 생략된다. 같은 이치로, 일단 최종 명제가 작업 메모리에 생성되면 이후의 힌트 생성 시에는 최종 명제 생성을 위한 추론이 생략되므로 신속한 처리가 이루어진다. 최종 명제 P_G 가 생성되면, 교수기는 명제 P_G 의 지지 명제들 가운데 앞서의 두 가지 힌트 조건을 만족하는 명제 P_H 를 찾아 이를 힌트로서

<p>예제</p>  <p>Given : $AB \parallel CD, AE=DE$ Find : $\triangle ABE \cong \triangle CDE$</p> <p>Given : $AC = BC, \angle BAC = 80, \angle CBD = 30$ Find : $\angle ABD$</p>	
<p>힌트 생성</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>$\langle P_1 \rangle$ $\triangle ABE \cong \triangle CDE$</p> <p>ASA</p> <p>$\times \langle P_2 \rangle$ $\times \langle P_3 \rangle$ $\circ \langle P_4 \rangle$</p> <p>$\angle AEB = \angle CED$ $AE = DE$ $\angle BAD = \angle CDA$</p> <p>Vertical Given Parallel</p> <p>$\langle P_5 \rangle$ $AB \parallel CD$</p> <p>Given</p> </div> <div style="text-align: center;"> <p>$\langle P_6 \rangle$ $\angle ABD = 50$</p> <p>Algebra</p> <p>$\times \langle P_7 \rangle$ $\times \langle P_8 \rangle$ $\circ \langle P_9 \rangle$</p> <p>$\angle ABC = \angle ABD + \angle CBD$ $\angle CBD = 30$ $\angle ABC = 80$</p> <p>Trivial Given Algebra</p> <p>Given</p> <p>$\langle P_{10} \rangle$ $\langle P_{11} \rangle$</p> <p>$\angle ABC = \angle BAC$ $\angle BAC = 80$</p> <p>$\langle P_{12} \rangle$ \rightarrow Isosceles Given</p> <p>$AC = BC$</p> </div> </div>	
<p>힌트 제시</p> <div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> <p>Since $AB \parallel CD,$ $\angle BAD = \angle CDA.$</p> </div> <div style="width: 45%;"> <p>Since $\angle ABC = \angle BAC$ and $\angle BAC = 80,$ $\angle ABC = 80.$</p> </div> </div>	

(그림 7) 힌트 생성의 예

결정한다. 그러한 명제가 없으면 P_G 를 힌트로서 결정한다. 힌트로 결정된 명제는 학습자에게 제시되며, 이제부터 학습자에게 ‘알려진’ 것으로 속성이 갱신된다.

힌트는 학습자에게 충분한 도움말이 되도록 하기 위해 단순히 ‘...임’이 아닌, 지지 명제와 함께 ‘...이므로, ...임’, 즉 ‘<좌변> \Rightarrow <우변>’ 형식으로 제시된다. 이때 좌변의 지지 명제들 가운데 앞서 말한 ‘하찮은’ 원리에 기반한 것들을 제외시키고 제시함으로써 교수기의 설명에 그와 같은 내용이 포함되지 않도록 한다. 학습자는 문제해결의 어느 시점에서나 여러 번에 걸쳐 힌트를 요구할 수 있다. 각각의 힌트 요구에 대해 교수기는 언제나 그 시점에서 가장 적절한 힌트를 위와 동일한 방식에 의해 생성, 제시한다.

(그림 7)은 두 개의 간단한 문제를 사용하여 힌트 생성 원리를 예시한다. 그림 설명을 위해 각 문제해결의 현 시점에서 명제 P_2 는 ‘알려진’ 것이고, 명제 P_4, P_9 은 아직 ‘알려진’ 것이 아니라고 가정하자. 그러면, 왼편의 문제에서는, 명제 P_2 와 P_3 가 각각 학습자가 이미 알거나 주어진 것이고 명제 P_4 는 아직 알려진 것이 아니기 때문에 명제 P_4 가 힌트로 선택됨을 보인다. 오른편의 문제에서는, 명제 P_7 과 P_8 은 각

각 하찮은 것, 주어진 것이고 명제 P_9 은 아직 알려진 것이 아니기 때문에 이것을 힌트로 선택한다. 그림에서 최종 명제의 지지 명제들 가운데 ‘ \times ’ 표시된 것들은 힌트에서 제외됨을, ‘ \circ ’ 표시된 것들은 힌트로 선택됨을 의미한다.

6.3 근거 및 우회 설명

많은 고려를 거쳐 생성된 힌트이지만 학습자 입장에서는 간혹 이해 곤란한 것일 수가 있다. 여기에는 두 가지 경우가 있는데, 힌트 명제 P 의 지지 명제들, 즉 좌변에 사용된 명제들 가운데 한 명제 P_L 의 생성 근거가 이해되지 않는 경우와, 명제 P 의 지지 명제들로부터 어떻게 명제 P 가 유추될 수 있는지 이해되지 않는 경우이다. 이런 경우, 학습자는 각각 ‘Why P_L ?’ 또는 ‘Why?’ 질문을 할 수 있다. 그러면 교수기는 전자의 질문에 대해서는 ‘근거 설명’을, 후자의 질문에 대해서는 ‘우회 설명’을 제공한다. 다음은 이들 각각에 대한 설명이다.

근거 설명(grounds explanation)은 특정 명제 P_L 의 성립 근거를 제시하는 것이다. 이에 대해 교수기는 명제 P_L 을 P_L 의 지지 명제들과 함께 제시함으로써 그 성립 근거를 설명

한다. 만약 명제 P_L 이 지지 명제를 하나도 가지지 않는다면, P_L 에 대한 근거 설명을 P_L 에 대한 우회 설명으로 대체하여 제공한다.

우회 설명(roundabout explanation)은 특정 명제 P 를 우회적으로 설명하는 것을 말한다. 이는 학습자가 P 에 사용된 공식 M 이 너무 어렵거나 비약적이라서 이해하지 못한 경우에 요구되는 설명이기 때문에, 교수기는 '공식 M 을 사용하지 않고' 다른 방향에서 동일 명제 P 를 유도, 제시할 필요가 있다. 이것이 명제 P 에 대한 우회 설명인데, 교수기가 이를 얻는 과정은 다음과 같다. 먼저, 작업 메모리 명제간의 의존(dependency) 관계에서 공식 M 에 관련된 일련의 명제들을 임시 제거한다. 즉, 작업 메모리의 모든 명제가운데 M 에 근거한 명제들, 그리고 이 명제에 의해 지지되는 명제들을 재귀적으로 임시 제거한다. 따라서 명제 P 도 이 과정에서 격리된다. 그리고는 공식 M 에 의한 설명이 또다시 생성되지 않도록 하기 위해 공식 M 에 근거한 연산자들도 임시 제거한다. 여기까지, 우회 설명을 위한 작업 메모리 초기화가 완료되면, 교수기는 명제 P 를 가설 H 로 변환하고 이를 목표 가설로 하여 추론을 재개시킨다. 추론은 학습자 가설 평가를 위한 추론에서와 동일한 방식으로 진행한다. 추론이 정지하면 교수기는 방금의 추론 사이클에서 생성된 명제들을 제거하고 임시 제거되었던 명제와 연산자들을 모두 복구함으로써 의존 관계의 논거를 원래대로 유지시킨다. 한편 추론의 결과는 우회 설명에 사용된다. 만약 목표 가설이 진(true)으로 검증된 상태로 추론이 정지하면서 명제 P 와 동치인 명제 P' 가 생성되면, 교수기는 이를 우회 설명에 직접 사용한다. 즉, P' 를 학습자에게 P 의 지지 명제와 함께 제시하는 것으로써 원래 명제 P 에 대한 우회 설명을 제시할 수 있다. 반대로 만약 목표 가설이 진(true)임을 검증하는데 실패한다면 이는 교수기가 현재 동원 가능한 정보로는 우회 설명이 불가능한 경우로써, 이때는 해당 공식 M 을 있는 그대로 학습할 것을 학습자에게 권하는 것으로써 설명을 대신한다.

마지막으로, '...이므로, ...임'의 '<좌변> ⇒ <우변>' 형식으로 제시되는 근거 및 우회 설명은 힌트 제시 때와 마찬가지로, 지지 명제들 가운데 하찮은 원리에 기반한 것은 좌변에서 제외시킨다.

앞장 (그림 3)의 실행 예는 힌트 요구에 대한 힌트 생성, 'Why ...?'에 대한 근거 설명, 그리고 'Why?'에 대한 우회 설명이 주어지는 실행례 모두 보여준다.

7. 향후 과제 및 결론

이 장에서는 먼저 현 교수 체계의 보완 및 확장을 위해 고려중인 몇 가지 연구 과제들을 소개하고, 마지막으로 결론을 제시한다.

7.1 향후 과제

첫째, 공식 원리의 난이도 개념을 도입하여 이를 교수에 활용할 필요가 있다. 즉, 어떤 명제에 대해 우회 설명할 때, 이보다 더 어려운 공식에 의한 설명보다는 더 쉽거나 같은 수준의 난이도를 가지는 공식들에 의해 설명함을 말한다. 그러나 어떤 학습 영역의 공식을 난이도에 의해 완전 정렬한다는 것은 비현실적이라는 문제가 있다. 따라서 전체가 아닌 부분 순서에 의한 난이도를 도입하는 것이 효과적인 접근 방법이 될 수 있다. 구현상으로는, 연산자 속성의 하나로 난이도를 추가하고 교수기의 필요에 따라서는 특정 난이도 이상의 연산자들을 '격리'한 상태에서 추론을 수행하는 방법이 있다. 다른 방편으로는, 문제마다 난이도 속성을 도입, 연산자 자동 생성 시에 현 문제가 요구하는 수준을 초과하는 난이도의 연산자는 처음부터 생성하지 않는 방법도 있을 수 있다. 하지만 이 방법은 문제마다 난이도를 정의해야 한다는 학습 저자의 인지적 부담을 완화할 방안이 함께 연구되어야 할 것이다.

둘째, 교안(lesson)을 도입하는 것이다. 현재 본 교수기는 문제간의 연관성을 고려치 않고 개개의 문제에 관해서만 교수한다. 이를 수정하여, 현 문제에 대한 학습 성취도를 측정하고 이를 바탕으로 다음 문제를 추천하도록 확장할 것을 고려해 볼 수 있다. 성취도 측정은 상호 작용의 내용을 근거로 행해질 수 있으며, 성취도가 높은 경우 상위 단계의 문제를 추천하며, 그렇지 못한 경우 현 문제와 유사한 문제를 추천하도록 하는 것이다. 상위 단계 또는 유사 문제의 선택을 위한 데이터는 학습 저자에 의해 공급되도록 하는 방식이 가장 간편할 수도 있으나, 저자가 이 같은 정보를 자동화된, 손쉬운 방식에 의해 공급할 수 있도록 하는 체계가 함께 연구되어야 할 것이다.

셋째, 학습자 모델의 적극적인 도입이다. 학습자 개개인의 강점과 약점을 파악하고 이에 근거한 차별화된 교수 전략을 구사하도록 더욱 확장할 필요가 있다는 것이다[24]. 현재의 구현에서도 학습자 추론 수준에 동적으로 적응하는 가설 평가라든가, 힌트나 설명 제공시 학습자가 이미 '알고 있는' 사실에 대해서는 부연하지 않는 등 개별 학습의 환경을 일부 성취하고는 있다. 이에 더하여, 학습자의 오류 입력에 대한 정밀한 추적을 통해 오류 카타로그 방식으로 관리한다든가, 현 문제해결 학습에서 관찰된 학습자 데이터를 차후의 문제해결 학습에서 활용하도록 개선할 경우 본격적인 의미에서의 개별 학습이 성취될 수 있을 것이다.

마지막으로, 시스템에 대한 정량적 평가와 검증이 필요하다. 현재로서는 [31]에서 사용자 친숙성, 인지적 타당성 그리고 학습 성능에 관해 본 교수 모형이 가지는 비교우위성에 관해 정성적인 차원에서 분석 제시함으로써 기존 연구에 비해 진일보된 연구라는 점을 강조한 바 있다. 더욱 엄밀한 의미에서의 정량적 평가와 검증은 차후 본 교수 모형

이 통신 환경 등을 활용, 실제 학습에 적용되는 단계에 이르면 이루어질 수 있을 것이다.

7.2 결 론

본 논문의 연구 영역인 문제기반 학습의 경우, 문제해결에 관한 모든 지식을 규칙화할 경우 규칙베이스의 규모가 방대해질 수 밖에 없으며 다른 영역에의 적용도 유연하지 않게 된다. 규칙베이스에 의한 시도되었던 기존의 문제기반 학습용 ITS가 실효성과 확장성 면에서 제한되었던 이유도 여기에 기인한 것이라 할 수 있다.

기존 연구의 한계를 극복하기 위한 시도로서, 본 연구는 문제해결에 관한 지식을 영역 고유한 것과 그렇지 않은 것으로 분리하여 접근하였다. 첫째, 영역 고유한 것은 문제기반 학습의 초기에 (규칙이 아니라) 탐색 연산자 형태로 자동 생성되는데 이들은 해당 문제해결에 직결된 공식원리 인스턴스들로만 구성되므로 수적으로 방대하지 않으며 상호 작용의 실시간에 문제 로드와 동시에 시간 지연없이 생성될 수 있다. 둘째, 학습 영역으로부터 독립된 문제해결 지식은 규칙베이스로 구현되었으며, 이 규칙들은 앞서의 영역 연산자들을 언제 어떻게 적용하느냐에 관한 추론 전략을 내용으로 하고 있다. 이 규칙베이스는 영역 지식을 직접 다루지 않고 문제해결에 관한 메타 지식(예: “수식에 포함된 특정 quantity의 값(goal)을 구하려 한다면 그 수식에 포함된 다른 quantity의 값(subgoal)을 먼저 찾아내도록 시도하라”)을 취급한다. 따라서 영역 지식이 혼재할 경우 방대해질 수밖에 없는 규칙베이스에 비해 규모가 현저히 작은 이점과, 다른 공식 기반의 문제해결 학습 영역에도 유연하게 적용될 수 있는 장점을 가진다.

본 연구가 제안한 교수 체계는 실험적 규모로 구축된 평면 기하 영역의 공식과 문제들을 통해 상호 작용 내용이 제시되었다. 기하외에도 수학, 과학 등 다양한 문제해결 학습 영역에 적용 가능한 본 연구 결과는 기존 시스템에서 비해 크게 세 가지 점에서 진전된 교수 체계의 모형을 제시한다.

첫째, 기존 시스템이 비교적 단순한 증명 문제를 위주로 한 학습만을 제공한데 비해, 제안된 시스템은 증명 외에 해결 유형의 문제해결 학습도 지원하므로 실용성 면에서 진전된 교수기의 모형으로서 활용될 수 있다.

둘째, 학습자의 문제해결의 세부적 단계에 정밀 조율된 교수를 가능케 함은 물론 상이한 학습자 추론 수준과 공식 인지 수준에 유연히 적용하는 체계를 가진다. 이런 점에서 기존의 시스템 위주가 아닌, 학습자 위주의 개인 교수식 환경을 제공한다.

마지막으로, 미리 정해진(built-in) 도움말과 해결 경로만을 주입하는 방식을 지양하고 실시간 문제해결 추론에 기반한 교수 대화를 구사함으로써 학습 효과를 극대화한다.

참 고 문 헌

- [1] Alpert, S. R., Singley, M. K., Fairweather, P. G., “Deploying Intelligent Tutors on the Web : An Architecture and an Example,” *Int. J. of Artificial Intelligence in Education*, Vol.10, pp.183-197, 1999.
- [2] Anderson, J., “Acquisition of Proof Skills in Geometry” in [20].
- [3] Anderson, J., *The Architecture of Cognition*, Harvard University Press, 1983.
- [4] Anderson, J., “The Expert Module,” in [22].
- [5] Anderson, J., Boyle, F., Corbett, A., Lewis, M., “Cognitive Modeling and Intelligent Tutoring,” *Artificial Intelligence*, Vol.42, pp.7-49, 1990.
- [6] Barr, A., Feigenbaum, E., *The Handbook of Artificial Intelligence, Vol.1*, William Kaufmann, 1981.
- [7] Barr, A., Feigenbaum, E., *The Handbook of Artificial Intelligence, Vol.2*. William Kaufmann, 1982.
- [8] Brown, J., Burton, R., DeKleer, J., “Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III,” in [23].
- [9] Burns, H., Parlett, J., Redfield, C. (eds), *Intelligent Tutoring Systems : Evolution in Design*, Lawrence Erlbaum, 1991.
- [10] Chen Y. R., Hornig, T. L. A., Yang, D. L., “A Java-Based Interactive Learning System of Junior High School Level Geometry,” in [12].
- [11] Collins, A., Brown, J., “The Computer As a Tool for Learning Through Reflection,” in [19].
- [12] Greer, J. et. al., *Learning Societies in the New Millennium : Creativity, Caring and Commitments (Proc. of ICCE 2000 - International Conf. on Computers in Education)*, 2000.
- [13] Hwang, F. K., “A Constructivist Virtual Physics Laboratory,” in [12].
- [14] Kearsley, G. (ed), *Artificial Intelligence and Instruction : Applications and Methods*, Addison-Wesley, 1987.
- [15] Larkin, J., McDermott, J., Simon, D., Simon, H., “Models of Competence in Solving Physics Problems,” *Cognitive Science*, Vol.4, pp.317-348, 1980.
- [16] Lawler, R., Yazdani, M. (eds), *Artificial Intelligence and Education, Vol.1*, Ablex, 1987.
- [17] Lawson, M., Chinnappan, M., “Generative Activity During Geometry Problem Solving : Comparison of the Performance of High-Achieving and Low-Achieving High School Students,” *Cognition and Instruction*, Vol.12, pp.61-93, 1994.
- [18] Leloche, R., “Use of Abstraction Levels in the Design of Intelligent Tutoring Systems,” in [12].
- [19] Mandl, H., Lesgold, A. (eds), *Learning Issues for In-*

telligent Tutoring Systems, Springer-Verlag, 1988.

[20] Michalski, R., Carbonell, J., Mitchell, T. (eds), *Machine Learning*, Morgan Kaufman, 1983.

[21] Ogata, H., Yano, Y., "Combining Knowledge Awareness and Information Filtering in an Open-Ended Collaborative Learning Environment," *Int. J. of Artificial Intelligence in Education*, Vol.11, pp.33-46, 2000.

[22] Polson, M., Richardson, J. (eds), *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum, 1988.

[23] Sleeman, D., Brown, J., *Intelligent Tutoring Systems*, Academic Press, 1982.

[24] VanLehn K., "Student Modeling," in [22].

[25] Wenger, E. (ed), *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, 1987.

[26] White, B., Frederiksen, J., "Qualitative Models and Intelligent Learning Environment," in [16].

[27] White, B., Frederiksen, J., "Causal Model Progressions As a Foundation for Intelligent Environments," *Artificial Intelligence*, Vol.42, pp.97-157, 1990.

[28] Woolf, B., "Theoretical Frontiers in Building a Machine

Tutor," in [14].

[29] Woolf, B., "Representing, Acquiring, and Reasoning about Tutoring Knowledge," in [9].

[30] 국형준, "기하 문제 학습을 위한 동적 추론체계", 한국정보과학회논문지-소프트웨어 및 응용, Vol.27, pp.412-421, 2000.

[31] 국형준, "효과적 문제해결 학습을 위한 지능형 기하 교수 시스템", 한국정보과학회논문지(B), Vol.25, pp.1090-1100, 1998.

[32] URL1, <http://www.ghg.net/clips/CLIPS.html>, "Clips : A Tool for Building Expert Systems."



국형준

e-mail : kook@sejong.ac.kr

1979년 서울대학교 공과대학 졸업(학사)

1983년 Univ. of South Carolina(Computer Science 석사)

1989년 Univ. of Texas at Austin(Computer Science 박사, Post-Doc)

1989년~현재 세종대학교 컴퓨터공학과 부교수.

관심분야 : 인공 지능, 전문가 시스템, 지능형 교수시스템, 지능형 에이전트