

펌핑 정리 학습을 위한 웹기반 적응형 시스템 개발

정호숙[†] · 민경실^{††} · 박성빈^{†††}

요 약

본 논문은 정규 언어에 대한 펌핑 정리를 학습하는 학생들을 돕기 위한 웹 기반 상호작용적 적응형 학습 시스템을 제안하고 있다. 본 시스템은 웹 기반 교육을 통해서 학습자들이 개인차에 따라 학습을 진행할 수 있도록 하고, 학습과 관련된 예제들을 연습하고 그에 대한 피드백을 즉시 제공할 수 있는 상호작용의 기회를 제공한다. 특히, 본 시스템은 펌핑 정리의 각 증명 단계를 이해할 수 있는 적응적 비계를 제공한다. 펌핑 정리를 학습할 수 있는 기존의 시스템들과 달리 본 논문에서 제안하는 시스템은 가능한 오류들을 미리 정하여 각 오류에 대한 메시지들을 적응적으로 제공한다. 또한 학습자들이 직접 문자열을 셋으로 나누도록 함으로써 펌핑 정리를 정확히 이해하도록 도움을 준다.

주제어 : 형식 언어, 적응성, 펌핑 렘마, 비계

Development of a Web-based Adaptive System for Learning Pumping Lemma

Hyosook Jung[†] · Kyungsil Min^{††} · Seongbin Park^{†††}

ABSTRACT

This paper presents a Web-based interactive and adaptive learning system that helps students learn the pumping lemma for the family of regular languages. Our system allows the students to proceed with their learning according to their individual differences through Web-Based Instruction and gives them opportunities for the interaction so that they can practice exercises related to the learning and gain feedbacks on the results of the exercises immediately. Especially, the system provides adaptive scaffolding that helps learners understand each step for the proof of the pumping lemma. Unlike existing systems that support learning the pumping lemma, the proposed system defines possible errors in advance and provides appropriate messages for corresponding errors. In addition, the system allows the learners to decompose a string into three parts so that they can understand the pumping lemma precisely.

Keywords : Formal language, Adaptation, Pumping Lemma, Scaffolding

† 종신회원: 고려대학교 컴퓨터교육과 박사과정
 †† 준 회원: 고려대학교 컴퓨터교육과 학부과정
 ††† 종신회원: 고려대학교 컴퓨터교육과 교수(교실의식)
 논문접수: 2009년 7월 15일, 심사완료: 2009년 8월 25일

1. 서 론

디지털 컴퓨터에 대한 추상적 모델을 다루는 오토마타 이론은 알고리즘 그리고 계산 복잡도 이론과 밀접한 관련을 갖는 분야 중 하나이다. 특히, 펌핑 정리는 오토마타가 문자열을 인식하는 과정에서 어떤 형식 언어가 특정 형태의 언어들의 집합에 속하지 않음을 증명하기 위해서 사용된다[2].

정규 언어에 대한 펌핑 정리는 어떤 형식 언어가 정규 언어가 아님을 증명하기 위해서 사용되며 임의의 언어 A 가 무한 정규 언어라면, 다음 성질을 만족하는 양의 정수 m 이 존재한다는 것이다[1][13][14][15].

만일 A 에 속하는 문자열 w 의 길이가 m 이상이면 w 는 다음과 같은 두 가지 조건을 만족하면서 $w=xyz$ 가 되는 세 개의 문자열 x, y, z 로 분할될 수 있다. 첫째 조건은 $|xy| \leq m$ 이라는 것이고 (즉, 문자열 x 와 문자열 y 모두의 길이의 합은 m 이하이고) 두 번째 조건은 $|y| \geq 1$ 이라는 것이다 (즉, 문자열 y 의 길이는 1 이상이다). 이러한 두 조건을 만족시키면서 분할한 w 를 이용하여 만든 $xz, xyz, xyyz, xyxyz \dots$ 등은 모두 A 에 속한다. 이는 A 에 속한 충분히 긴 문자열을 세 부분으로 나누었을 때, 중간 부분을 임의의 회수만큼 반복하여 만들어진 문자열이 A 에 속한다는 것이다.

펌핑 정리는 학생들이 다양한 형식 언어들 사이의 관계를 이해하고 형식 언어들의 구조 특성을 구별할 수 있는 논리적인 방법을 제시해 주기 때문에, 계산 이론 학습에서 반드시 이해해야 하는 중요한 정리이다. 그러나 펌핑 정리가 복잡하고, 그것을 적용하는 데 있어 실수하기 쉽기 때문에 대부분의 학생들은 펌핑 정리의 개념을 이해하는데 어려움을 느끼고 있다[3]. 따라서 학생들은 다양한 예제를 통해 펌핑 정리를 이용한 증명 과정을 충분히 연습하여 이를 이용하는 방법에 숙달될 필요가 있지만 일반적인 학습 상황을 고려해 볼 때, 교수자가 많은 인원의 학습자를 대상으로 제한된 시간에 이러한 학습 활동을 적용하는 것은 쉽지 않다[4]. 최근 웹 기반 수업의 발달로, 학습자들은 시간과 공간의 제약 없이 학습 자원에

접근할 수 있을 뿐만 아니라, 다양하고 풍부한 관련 예제들을 연습할 수 있고 이에 대한 즉각적인 피드백도 받을 수 있으며, 자신의 학습 속도에 맞게 학습을 진행할 수 있게 되었다.

본 논문에서는 학생들의 개인차에 따라 펌핑 정리를 학습할 수 있도록 도움을 주는 적응형 비계 (adaptive scaffolding) 를 제공하는 웹 기반 학습 시스템을 제안하고자 한다. 본 시스템에서 제공하는 적응형 비계란 학습자가 입력한 데이터의 유효성, 학습자의 이해 정도, 학습자가 예제를 풀 때 보이는 오류의 형태 등을 고려한 적응적인 내용 제시와 링크를 말한다. 적응적인 내용 제시란 학습 내용에 대한 학습자의 이해 정도에 따라 제시되는 내용의 어려운 정도를 조정하여 정보를 제공하는 것을 의미한다. 적응적인 링크란 사용자가 예제를 푸는 과정에서 생성한 오류에 따라 알맞은 단계로 이동할 수 있는 링크를 제공하는 것을 의미한다.

본 시스템은 펌핑 정리를 이용한 증명과정을 여러 단계로 나눈 후, 각 단계에서 학생들이 입력한 데이터의 정확성과 적합성을 확인한다. 만일 특정 단계에서 오류가 발생한다면, 그 오류를 어떻게 수정해야 하는지에 대한 설명을 제시하고, 이를 수정하기 위해 이동해야 하는 곳으로 이동할 수 있는 링크 앵커도 제시한다. 특히, 본 시스템은 학생들이 스스로 학습해 나갈 수 있도록, 학생들이 이해 정도가 증가함에 따라 제시되는 설명을 점차 줄여나가며 컴퓨터과학 또는 컴퓨터공학을 전공하는 대학생들이 형식 언어에 대한 내용을 학습하면서 활용할 수 있게 설계되었다.

2. 관련 연구

웹 기반 수업은 특정 학습 목표를 달성시키기 위해서 원격 학습자들에게 다양한 웹 자원과 상호작용의 기회를 제공하는 하이퍼미디어 기반 수업 프로그램이다. 학습자들은 편리한 시간에 원하는 장소에서 필요한 정보를 다양하게 탐색할 수 있을 뿐만 아니라[5], 학습자의 요구사항을 반영하고, 조직화된 학습 과정을 제공하며, 부가적인 학습 활동 및 고도의 상호작용을 제공할 경우, 전통적 수업보다 효과적인 교수-학습 활동을 가능하

게 한다[6].

적응형 하이퍼미디어는 하이퍼 공간을 탐색하는 동안 발생하게 되는 인지적 과부하와 방향 상실의 문제를 해결하기 위해서 등장하였다. 사용자들의 특성을 저장하고 있는 사용자 모델에 기반하여 각 사용자에게 알맞은 내용과 링크 구조를 적응적으로 제공한다[7][8].

비계(scaffolding)는 사회-문화적 구성주의에서 보다 유능한 협력자가 상대방의 과제 수행능력에 따라 도움의 수준을 조절해 가는 과정을 설명하기 위해 사용되는 용어이다. 수업에서의 비계 설정은 학생의 학습을 도와주는 모든 장비나 전략을 의미한다. 학습자의 인지적 향상을 위해서 처음에에서의 학습자가 학습 목표를 달성모든 장비나 전략을 의제공하지만, 점차 략을 의결어나가면서 학습자가 스스로 과제를 해결할 수 있도록 한다[9].

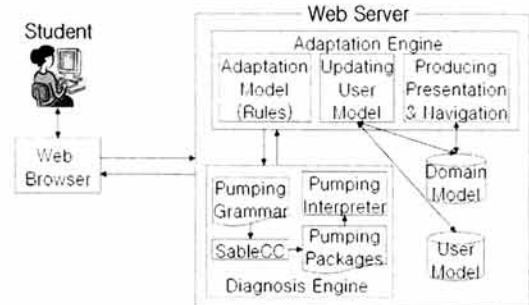
Bilksa 등은 형식언어와 오토마타 이론에 대한 개념을 실험하기 위한 다양한 도구를 제안하였다. 그 중에서 PumpLemma는 특정 언어가 정규 언어 인지를 증명하기 위한 도구로써, 학습자가 임의의 언어를 정의하면, 이 언어가 정규 언어인지 증명하기 위해 사용자로부터 입력된 다양한 문자열을 검증한다[10]. Bilksa 등이 제안한 시스템은 사용자가 다양한 문자열을 입력하고 결과를 확인할 수 있지만, 오류가 발생하였을 때, 그 원인이 무엇이고 어떻게 수정하는 것이 좋은지에 대한 설명이 부족하기 때문에, 학습자들이 복잡한 증명 과정을 명확히 이해하는데 한계가 있다. 본 논문에서 제안한 시스템은 증명 과정에서 발생할 수 있는 오류들을 미리 파악하여, 학습자의 지식수준 및 오류 유형에 따라 이를 수정하기 위해 필요한 정보를 적응적으로 제시한다.

Cogliati가 제안한 시스템은 정규 언어에 대한 펍핑 정리를 시각화하여 학습자의 이해를 돕고 있다. 웹 브라우저에서 자바 애플릿을 실행시켜 유한 상태 오토마타를 대상으로 펍핑 정리를 적용한다[11]. 시각화는 학습자의 이해를 돕는 보조수단이 될 수 있으나, 펍핑 정리를 적용할 수 있는 능력은 다양한 예제를 통해 이를 논리적으로 증명해나가는 과정에서 습득되기 때문에, 논증에 대한 연습이 무엇보다 중요하다. Cogliati의 시스템이 사용자가 입력한 문자열이 유한 상태 오토

마타로 인식이 되는지를 확인하는데 초점을 두고 있는 것과 달리, 본 논문에서 제안한 시스템은 학습자들이 펍핑 정리를 이용한 증명 과정을 실습하는데 초점을 두고 있다. 예를 들어, 본 논문에서 제안한 시스템은 펍핑 정리 적용시 중요한 과정인 입력 문자열을 세 개의 문자열로 나누는 단계를 학습자가 직접 입력 하도록 하고 있으나, Cogliati의 시스템은 문자열을 자동으로 분할해준다.

3. 시스템

본 논문에서 제안한 시스템은 도메인 모델 (Domain Model), 사용자 모델 (User Model), 적응형 모델 (Adaptation Model), 적응형 엔진 (Adaptation Engine), 진단 엔진(Diagnosis Engine) 으로 구성되어 있다. <그림 1> 은 본 논문에서 제안한 시스템의 구조를 제시하고 있다.

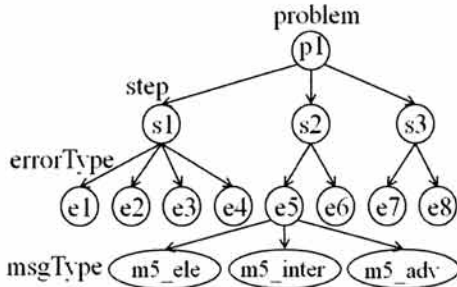


<그림 1> 시스템 구조

도메인 모델은 사용자가 시스템을 이용해 학습할 내용들과 관련된 정보를 표현하는데 구체적으로 펍핑 정리 예제, 예제의 증명 단계, 및 각 증명 단계의 오류 유형들을 정의하고 있다.

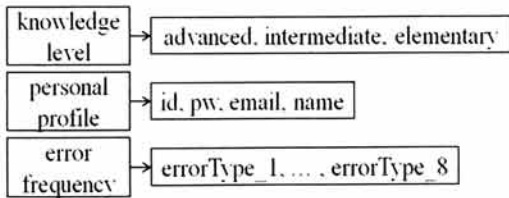
본 논문에서 제안한 시스템에 사용된 도메인 모델의 구조는 <그림 2> 와 같다. 각 예제 (p1)는 증명을 위한 세 가지 단계 (s1, s2, s3) 로 구성되며, 각 단계는 또한 그 단계에서 발생할 수 있는 오류 유형들로 구성된다. 즉, 단계 1 (s1)에서는 네 가지 가능한 오류들 (e1, e2, e3, e4) 이 있을 수 있고 단계 2 (s2)에서는 두 가지 가능한 오류들 (e5, e6) 이 있을 수 있으며 단계 3 (s3)에서는 두 가지 가능한 오류들 (e7, e8) 이 있을

수 있다. 또한 단계 2의 첫 번째 오류(e5)에 대해 시스템은 세 가지 종류의 메시지(m5_ele, m5_inter, m5_adv)를 제시할 수 있다.



<그림 2> 도메인 모델

사용자 모델은 사용자에 대한 정보를 표현하며 학습자의 속성 및 속성 값으로 구성되어 있다. 예를 들어, 학습 내용에 대한 학습자의 이해 수준, 학습 과정에서 나타나는 학습자의 오류 형태 등의 정보를 저장하고 있다. 본 논문에서 제안한 시스템에 사용된 사용자 모델의 구조는 <그림 3>과 같다. 본 시스템은 학습자의 문제 해결 능력에 따라 의 학습 내용 이해 수준 (knowledge level)을 세 단계 (advanced, intermediate, elementary)로 나누고 각 사용자의 아이디, 패스워드, 이메일 주소, 및 이름을 저장하며 오류는 종류에 따라 여덟 가지 (errorType_1, ... errorType_8)를 정의하고 있다.



<그림 3> 사용자 모델

적응형 모델은 사용자가 시스템을 사용할 때 어떤 상황에서 어떤 내용을 제시하는 지를 표현하는 다양한 적응형 규칙들로 구성되어 있다. 즉, 어떻게 사용자 모델의 데이터를 갱신하는지, 학습자의 특성에 따라 어떻게 적응적인 내용과 링크를 제공할 것인지에 대한 규칙을 정의하고 있다. 예를 들어, 학습자의 현재 학습 내용에 대한 이해 수준이 낮을 경우, 오류를 해결하는데 도움을 줄

수 있는 쉽고 자세한 설명을 제시한다. 반대로, 학습자의 현재 학습 내용에 대한 이해 수준이 높을 경우, 학습자 스스로 오류를 해결할 수 있도록 간단한 설명만 제시한다. 펌핑 정리를 이용한 증명 단계에서 학습자의 입력 데이터가 정확할 경우, 다음 단계로 넘어가지만, 그렇지 않을 경우, 오류가 발생한 단계로 이동할 수 있는 링크 앵커를 제시한다. <그림 4>는 본 논문에서 제안한 시스템에 사용된 적응형 모델의 예이다. 만일 사용자가 1번 문제(p1)를 풀고 있고 오류의 유형이 여섯 번째(e6)이며 사용자의 지식 정도가 고급(advanced)이면, 사용자에게 특정 메시지(m5_adv)를 보여주고 특정 링크 앵커(s2)를 생성해서 사용자가 그 링크를 선택할 수 있도록 한다.

```

if
  problem= p1.
  errorType = e6.
  knowledge_level=advanced
then
  show msgType = m5_adv.
  create link = s2
  
```

<그림 4> 적응형 모델

진단 엔진은 학습자가 입력한 문자열이 펌핑 정리의 조건을 만족시키는지 확인한다. 진단 엔진은 사용자가 문자열을 어떻게 입력할 것인지에 대한 문법(pumping grammar)과 SableCC [12]를 이용하여 생성된 자바 패키지(pumping packages) 및 이를 이용하여 펌핑 정리의 조건을 확인하는 인터프리터(pumping interpreter)로 구성된다.

SableCC는 컴파일러, 인터프리터, 텍스트 파서 등을 생성하기 위한 객체 지향 프레임워크를 생성하는 파서 생성자로서 펌핑 정리를 만족시키는 조건을 확인하기 위한 문법 구조를 정의하면, 그 문법으로 쓰인 사용자의 입력 데이터를 해석하기 위한 자바 패키지가 자동으로 생성된다. 이러한 자바 패키지로부터 상속된 클래스를 이용하여 사용자의 입력 문자열이 펌핑 정리를 만족시키는지 확인하는 프로그램을 개발하였다. 본 논문에서 제안한 학습 시스템에서 펌핑 정리를 만족시키는 조건을 확인하기 위한 문법 구조는 <표 1>과 같다. 이 문법에서 ->는 왼편의 내용이 오른편의

내용으로 치환될 수 있다는 의미이고 오른쪽에 나타난 | 는 “또는”의 의미이다. 또한 * 는 0 이상 반복될 수 있다는 의미이고 + 는 1 번 이상 반복될 수 있다는 의미이다.

예를 들어 첫 번째 규칙은 문자열들 (strings) 이 문자열 (string) 과 문자열들 (strings) 의 연결로 이루어진다는 것을 의미하고 두 번째 규칙은 문자열 (string) 이 base_letter(exponent) 또는 base_letter(exponent)[value_i] 가 될 수 있다는 의미이다. 세 번째 규칙, 여덟 번째 규칙, 그리고 열 번째 규칙에 의하면 base_letter는 a, b, ..., z, A, B, ..., Z들을 자유자재로 연결해서 만들 수 있는 길이가 1 이상인 문자열이 되고 exponent는 0 에서 9 까지의 숫자 하나 뒤에 영어의 소문자, 대문자들이 0 개 이상 반복하여 만든 문자열이다.

<표 1> SableCC 문법 구조

```
strings -> string strings
string -> base_letter{exponent} |
        base_letter{exponent}[value_i]
base_letter -> alphabet
exponent -> poly
value_i -> number
poly -> digit (letter)*
number -> digit+
alphabet -> letter+
digit -> ['0'..'9']
letter -> ['a'..'z'] | ['A'..'Z']
```

적응형 엔진은 적응형 모델에 정의된 규칙에 따라서 학습이 진행되는 동안 변화되는 각 사용자의 특성들을 갱신하고, 갱신된 사용자의 특성에 따라 알맞은 설명과 링크 구조를 생성한다. 예를 들어, 학습자가 입력한 문자열이 문제에서 주어진 언어의 범위에 들지 않는 경우, 문제에서 제시된 언어의 조건에 맞지 않는 문자열이 입력되었음에 대한 설명을 학습자에게 제공한다. 이때, 학습자의 이해 수준에 따라, 내용을 달리하여 제시한다. 또한 해당 오류를 즉시 수정할 수 있도록 오류를 수정하기 위해서 이동해야하는 단계로 직접 이동할 수 있는 링크 앵커도 제시한다.

4. 시스템 사용

학습자가 시스템을 사용하여 펄핑 정리 예제를

해결하기 위해 학습자는 웹 브라우저를 이용하여 웹 서버에 접속한 후, 펄핑 정리 예제들 중 하나를 선택하고 예제에서 요구하는 조건을 읽은 후, 각 단계마다 문제에서 요구하는 조건에 맞는 문자열을 입력한다. 진단 엔진은 특정 단계에서 입력된 문자열이 펄핑 정리의 조건을 만족시키는지 확인한 후, 그 결과를 적응형 엔진에게 전송하고 적응형 엔진은 진단 엔진의 결과에 따라 학습자의 이해 수준, 에러 발생 회수 등과 같은 학습자의 사용자 모델을 갱신시킨다. 또한 적응형 엔진은 사용자 모델, 도메인 모델, 적응형 모델에 따라 현재 단계에서 학습자에게 제공할 적응적인 내용과 링크를 생성하고 학습자에게 필요한 설명과 링크 앵커가 담긴 페이지를 학습자에게 제시한다.

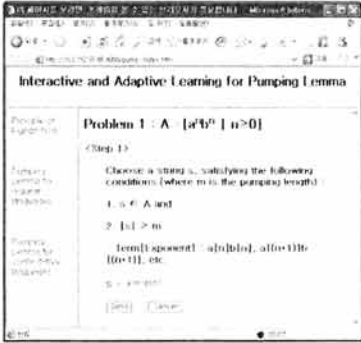
본 논문에서 제안하는 시스템을 이용하여 언어 $A = \{a^n b^n \mid n \geq 0\}$ 가 정규 언어가 아님을 보이기 위해서 학습자는 <표1>의 문법 구조에 따라 문자열을 일정 규칙에 맞게 작성해야 한다. 예를 들어, $a((m+1))b((m+1))$ 은 $a^{m+1}b^{m+1}$ 을 의미하고, $a((m-p))a(p)[2]b(m)$ 은 $a^m p (a^p)^2 b^m$ 을 의미한다.

첫 번째 단계에서, 학습자는 임의의 언어 A가 정규 언어 혹은 비정규 언어로 가정한 후, 문제에 제시된 언어의 조건을 만족시키는 문자열을 입력해야 한다. 이때, 문자열의 길이는 최소한 주어진 펄핑 길이 m과 같거나 길어야 한다. 만일 사용자가 이러한 조건에 맞지 않는 문자열을 입력한다면, 시스템은 오류 메시지를 제시한다. 이 단계에서 발생할 수 있는 오류 유형은 다음과 같다.

1. 학습자가 입력한 문자열이 주어진 언어의 조건에 맞지 않는다(예를 들어, $a^m b^n$).
2. 입력된 문자열이 주어진 언어의 요소들로 구성되어 있지 않다(예를 들어, $c^m d^n$).
3. 입력된 문자열이 펄핑 길이 m과 아무 관련 없이 정의되어 있다(예를 들어, $a^n b^n$).
4. 입력된 문자열의 길이가 펄핑 길이 m보다 짧다(예를 들어, $a^{n-1} b^n$).

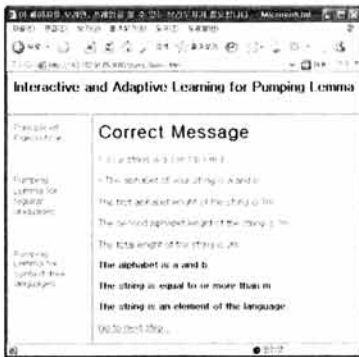
만일 학습자가 $a^{n-1} b^{n-1} (n \geq m)$ 이라는 문자열을 입력했다면, 펄핑 길이 m과 관련되는 부분이 없는 문자열을 입력했기 때문에 위의 오류 유형 3에 해당한다. 이때, 시스템은 해당 오류가 발생한 원인을 설명하는 내용을 학습자의 이해 수준에

맞게 제시하고 새로운 문자열을 입력하기 위해서 해당 페이지로 바로 이동할 수 있는 링크 앵커를 생성한다. <그림 5> 는 첫 단계에서 학습자가 오류 없이 바른 문자열을 입력하는 화면이다.



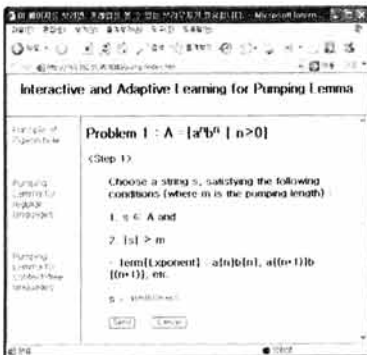
<그림 5> 바른 입력 문자열

<그림 6> 은 첫 번째 단계에서 오류 없이 바른 문자열을 입력했을 때 나타나는 화면이다.



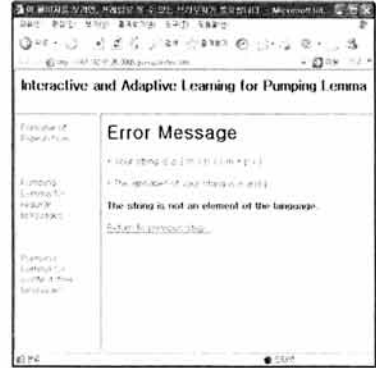
<그림 6> 바른 입력에 대한 설명

<그림 7> 은 첫 단계에서 학습자가 잘못된 문자열을 입력하는 화면이다.



<그림 7> 잘못된 입력 문자열

<그림 8> 은 첫 단계에서 학습자가 잘못된 문자열을 입력했을 때 나타나는 화면이다.

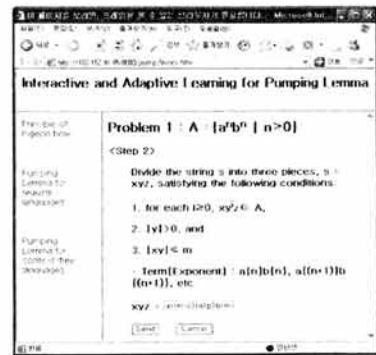


<그림 8> 첫 번째 단계의 잘못된 입력 문자열에 대한 오류 메시지

두 번째 단계에서, 학습자는 펌핑 정리의 조건에 맞게 자신이 입력한 문자열을 x, y, z 세 부분으로 분할한다. 이때, 다음과 같은 오류가 발생할 수 있다.

1. 입력된 문자열이 첫 번째 단계에서 입력된 문자열의 형태와 다르다.
2. xy 의 길이가 펌핑 길이 m 보다 길다.

<그림 9> 는 두 번째 단계에서 학습자가 문자열을 맞게 입력하는 화면이다.



<그림 9> 바른 입력 문자열

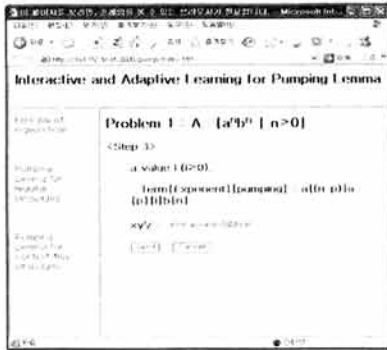
세 번째 단계에서, 학습자는 y 부분이 반복되는 회수 i 를 문자열에 적용시킨다. 즉, 자신의 문자열이 언어 A 에 속하지 않도록 i 를 선택해야 한다. 이때, 다음과 같은 오류가 발생할 수 있다.

1. 학습자가 입력한 문자열이 두 번째 단계에서 분할시킨 문자열의 형태와 다르다.
2. i 를 잘못 선택하여, 언어 A 에 속하는 문자열

을 얻게 된다.

이때, 학습자가 증명 초기에 언어 A 를 어떻게 가정하였느냐에 따라 시스템은 다른 피드백을 제시한다. 즉, 학습자가 언어 A 를 정규 언어라고 잘못 가정하고 시작했다면, 시스템은 학습자에게 학습자의 초기 가정이 잘못 되었음을 알리는 설명을 제시하고, 초기 단계로 이동할 수 있는 링크 앵커를 생성한다. 반면 학습자가 언어 A 를 비정규 언어라고 가정하였으나, i 를 잘못 선택하여 언어 A 에 속하는 문자열을 얻게 되었다면 i 가 잘못 선택되었음을 알리는 설명을 제시하고, i 를 다시 적용하는 단계로 이동하는 앵커를 생성한다.

<그림 10> 은 세 번째 단계에서 학습자가 문자열을 맞게 입력하는 화면이다.



<그림 10> 바른 입력 문자열

5. 결 론

본 논문에서는 학습자들이 웹상에서 펌핑 정리에 대한 다양한 예제를 학습하면서 이에 대한 이해를 강화할 수 있는 웹 기반 시스템을 제안하였다. 학습자들은 펌핑 정리의 증명 과정을 단계별로 학습하면서 자신들의 입력 결과를 확인하고 오류가 있을 경우, 이를 즉시 수정할 수 있으며, 학습자의 이해 수준에 따라 제시되는 설명 및 오류 수정 단계로 이동하는 링크를 통해 점차 오류를 줄여나가게 된다.

본 논문에서 제안한 시스템은 컴퓨터과학 혹은 컴퓨터공학을 전공하는 대학생들이 정규 언어의 구조 특성을 이해하는 데 필요한 펌핑 정리 학습을 도와주는 데 활용될 수 있을 뿐 아니라 시스

템의 기능 확장 및 변경을 통해 문맥 자유 언어의 구조 특성을 이해하는 데 도움을 줄 수 있으며 또한 학습자와 시스템간의 상호 작용이 필요한 증명 문제들을 이해하는 경우에도 활용될 수 있다.

참 고 문 헌

- [1] Sipser, M. (1997). *Introduction to the Theory of Computation*. Boston: PWS Publishing Company.
- [2] IEEE Computer Society and Association for Computing Machinery(2001). *Computing Curricula 2001 Computer Science*. ACM.
- [3] Cogliati, J. J. (2004). *Visualizing the pumping lemma for regular languages*. Master Thesis. Montana State University. Computer Science.
- [4] Truong, N., Bancroft, P., & Roe, P. (2003). A Web Based Environment for Learning to Program. *Proceedings of 25th Australian Computer Science Conference*.
- [5] Khan, B. H. (1997). *Web-based Instruction*. New Jersey: Educational Technology Publications.
- [6] Astleitner, H. (2001). *Web-Based Instruction and Learning : What Do We Know form Experimental Research?*, *Proceedings of 5th Romanian Internet Learning Workshop*.
- [7] Brusilovsky, P. (1996). Adaptive Hypermedia: an Attempt to Analyze and Generalize. *Multimedia, Hypermedia, and Virtual Reality*. 1077, 288-304.
- [8] De Bra, P., Brusilovsky, P., & Houben, G. J. (1999). Adaptive hypermedia: from systems to framework. *ACM Computing Surveys*. 31(1).
- [9] van Merriënboer, J. J., Kirschner, P. A., & Kester, L. (2003). Taking the load Off a Learner's Mind : Instructional Design for Complex Learning, *Educational Psychologist*. 38(1), 5-13.
- [10] Bilska, A. O., Leider, K. H., Procopiuc, M.,

Procopiuc, O., Rodger, S. H., Salemme, J. R., & Tsang, E. (1997). A collection of tools for making automata theory and formal languages come alive. *SIGCSE Bull.* 29(1), 15-19.

- [11] Grinder, M. T. (2002). *Active Learning Animations for the Theory of Computation*, Ph D. dissertation. Montana State University. Computer Science Department.
- [12] Gagnon, E. M., Etienne, M., & Hendren, L. J. (1998). SableCC: an object-oriented compiler framework. *Technology of Object-Oriented Languages*. 140-154.
- [13] Greenlaw, R. & Hoover, H. J. (1998). *Fundamentals of the Theory of Computation: Principles and Practice*. San Francisco: Morgan Kaufmann Publishers Inc.
- [14] Linz, P. (2006). *An Introduction to Formal Languages and Automata* (4th ed). Massachusetts: Jones and Bartlett Publishers Inc.
- [15] Moret, B. M. (1997). *The Theory of Computation*. Reading. MA: Addison Wesley.

민 경 실



2005~현재 고려대학교 컴퓨터
교육과 학부과정
관심분야: 컴퓨터교육, 시맨틱 웹
하이퍼텍스트

E-Mail: silsily2@korea.ac.kr

박 성 빈



1990 고려대학교 전산과학과
(이학사)
1993 University of Southern
California (전산학 석사)

1999 University of Southern California
(전산학 박사)

2006~현재 고려대학교 컴퓨터교육과 부교수
관심분야: 하이퍼텍스트, 컴퓨터 과학 교육,
알고리즘, 계산이론

E-Mail: psb@comedu.korea.ac.kr

정 호 속



1998 서울교육대학교
초등교육과(교육학학사)
2001 서울교육대학교 교육대학원
컴퓨터교육과(교육학석사)

2003~현재 고려대학교 컴퓨터교육과 박사과정
관심분야: 컴퓨터교육, 시맨틱 웹, 소셜 네트워킹
E-Mail: est0718@comedu.korea.ac.kr