

웹기반 그래프 알고리즘 탐구학습을 위한 Maple 워크시트 개발

서정현^o 이형욱

순천대학교 컴퓨터학과, 순천대학교 컴퓨터교육과
jhseo@sunchon.ac.kr^o, oklee@sunchon.ac.kr

Development of Maple Work Sheet for Web Based Graph Algorithm Exploratory Learning System

Jeonghyun Seo^o, Hyeongok Lee

Computer Science of Sunchon National Univ.^o, Computer Education, Sunchon National Univ.

요 약

Maple은 수학적 표현에 가까운 프로그래밍 언어로, 함수, 표현열(sequence), 집합, 리스트, 배열, 테이블, 등의 자료구조를 가지고 있다. 단순히 과학 계산과 관련된 수식처리뿐만 아니라 수식기호와 표현을 해석하여 그 문법과 의미를 파악할 수 있는 기능을 갖추고 있다. 본 연구에서는 Maple을 이용하여 그래프 이론 학습에 사용할 수 있는 white box형 워크시트를 개발하고, 개발된 워크시트를 웹에서 서비스 할 수 있도록 html로 변환하였다.

1. 서론

알고리즘을 대수적 관점에서 구현할 수 있는 컴퓨터 시스템을 사용하면 기존의 언어에 비해 알고리즘의 구현시간을 줄이고 수학적으로 구현할 수 있다. 이와 같은 시스템을 컴퓨터대수 시스템(Computer algebra System)이라고 한다. 이것은 수학 문제와 같이 수에 관련된 대상을 기호(symbol)로 처리하여 연산을 수행하는 컴퓨터 시스템으로서 수치계산과 대수적 연산을 수행하고 필요한 경우 함수를 그래프로 처리하는 시스템이다. 컴퓨터 대수 시스템에는 Mathematica, Derive, Reduce, MuPAD, Macsyma 그리고 Maple 등이 있다[1,2,3].

Maple은 컴퓨터 대수 시스템의 하나로서, 단순히 과학 계산과 관련된 수식처리뿐만 아니라 수식기호와 표현을 해석하여 그 문법과 의미를 파악할 수 있는 기능을 갖추고 있다. 또한 명령어가 수학의 수식과 거의 같기 때문에 쉽게 배우고 해석할 수 있으며, 수학 및 응용 분야의 이론식을 얻기 위한 시뮬레이션이 가능하다. 다른 고급 프로그래밍 언어들의 명령어와 같은 흐름 제어 등의 명령어를 이용해 프로그래밍 할 수 있으므로 확장성이 뛰어나다[4,5].

Maple은 수식의 표현을 웹에서 가능하게 하는 표준화 작업을 OpenMath 프로젝트에서 추진하고 있다. 현재 미국과 유럽을 중심으로 각각의 작업이 이루어지고 있는데, 이런 표준화 작업에 관련된 국내 연구는 매우 미미하다. 웹 연동 기술을 바탕으로 수식을 처리할 수 있는 기술은 매우 중요한 계산 엔진이다. 수학, 과학교육 그리고 컴퓨터 알고리즘 교육용 웹 엔진 개발은 매우 유용하고 강력한 학습 환경을 가능하게 한다. 특히 과학계산과 관련된 알고리즘 학습에 컴퓨터 대수 시스템의 다양한 기능을 사용하면 논리적인 알고리즘 설계와 구현이 매우 체계적이고 효율적이다[6,7].

본 연구는 탐구 사이클에 따라 학습할 수 있도록 Maple을 사용한 그래프 워크시트를 개발하였다. 본 논문의 구성은 2장에서는 Maple의 기능 및 특징, 기본 연산, 탐구 학습을 위한 방법을 살펴보고, 3장에서는 그래프 이론을 웹기반에서 Maple을 이용해 개발한 워크시트를 보였다. 마지막으로 본 연구의 효과와 발전된 학습탐구를 위해 더 연구해야 할 분야를 제시하였다.

2. Maple 프로그래밍 언어

Maple은 매우 조직적으로 짜여져 있고, 고급언어에 속한다. 또한 명령어가 수학의 수식과 거의 같기 때문에 쉽게 배우고 해석할 수 있으며, 수학 및 응용 분야의 이론식을 얻기 위한 시뮬레이션을 수행할 수 있다. Maple은 함수, 순열, 집합, 목록, 배열 등의 자료구조를 갖추고 있다. 따라서 자료형태를 검사하는 연산, 하나의 자료형태에서 다른 자료형태로 변환하는 연산, 복잡한 자료형태를 생성하는 연산 등이 제공되고 있다.

2.1 Maple의 기능과 특징

Maple은 수학에서 사용하는 대부분의 함수를 정의할 수 있으므로 표현이 매우 용이하다. 예를 들어 $f(x) = x^2 + 1$ 과 같은 함수를 표현하는데 수식 그 자체를 [예 1]처럼 입출력에 사용할 수 있다.

> f(x) = x^2+1;

f(x) = x^2+1

[예 1] Maple에서 수식 입력

'>'로 시작하는 줄은 입력이며 다음 줄은 출력이다.

Maple은 정확한 값을 계산하는 계산기이다. 일반적인 계산기에서 할 수 있는 사칙연산이나 제곱근 계산뿐 아니라 $n!$ 이나 π 의 근사값 계산 등 복잡한 계산을 할 수 있다. >Digits:=100;

>evalf(pi);는 $\sqrt{\pi}$ 를 100자리까지 구한다.

일반적으로 이전의 고급언어들은 수치들의 계산만 허용되었다. 그러나 Maple은 수식의 전개, 인수분해, 통분, 부분분수로 분해하기, 약분하기 등을 수행할 수 있을 뿐만 아니라 극한 미분, 적분, 미분 방정식, 멱급수 등에서도 기호연산이 가능하다.

Maple은 고수준의 프로그래밍 언어이다. 516개의 내장함수와 3000가지의 라이브러리가 있다. 또한 고급프로그래밍 언어에서 사용되는 각종 제어문과 자료구조를 제공한다. 이 함수들과 제어문을 이용하면 고수준의 프로그래밍이 가능하다.

Maple은 표준 패키지로 제공되는 TCP/IP 소켓을 사용하면 인터넷의 자료나 다른 컴퓨터의 데이터를 실시간으로 이용할 수 있다. 또한 계산 결과를 HTML의 확장 언어인 XML 문서로 출판할 수 있다. 수식표현과 문서를 MathML을 지원하는 웹 브라우저에서 읽을 수 있도록 MathML 2.0 규격을 지원하고 있다. MathML 브라우저가 없는 경우는 변환된 문서를 WebEQ 자바 애플릿으로 구동시킬 수 있다. Maple의 커널 자체는 외부

프로그램에 의해서 일반적으로 접근이 불가능하다 그러나 EXCEL과 데이터나 식을 주고받을 수 있다. 그리고 사용자가 개별적으로 개발한 c 코드를 공유 라이브러리에 연결하여 사용할 수 있다.

2.2 기본 연산

Maple에서 기본적인 연산은 수치계산 대수적 연산 그리고 함수의 그래픽 표현이 있다. 수식표현은 일반적으로 수학기호를 사용하는 방법이나 고급프로그래밍언어에서 라이브러리를 이용하는 방법이 있다. [예 2]는 각각 수학적 표현방법과 라이브러리를 이용한 방법이다

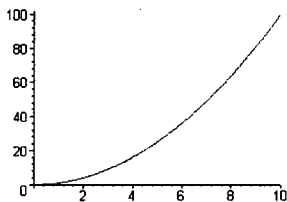
```
>4;
24
>ifactor(6);
(2)(3)
[예 2] Maple에서 수치연산
```

대수적 연산은 방정식 또는 다항식에 관련된 연산의 결과를 수행하거나 대수적인 성질을 가진 집합의 원소에 대한 연산의 결과를 보여주는 것이다. 다항식의 곱이나 인수분해, 전개식, 수식의 단순화, 변형, 함수의 정의. 연립방정식의 해 구하기, 부등식의 해 구하기 등이 이에 속한다. 대수적 연산에 해당하는 몇 가지 예를 [예 3]에 보였다.

```
>expr := (x+y)^2;
expr:=(x+y)^2
>expand(expr);
x^2+2xy+y^2
>factor(expr);
(x+y)^2
[예 3] Maple에서 대수적 연산
```

두 개의 미지수를 갖는 함수가 정의 되었을 때 그 함수를 시각적으로 보기 위해 2차원 평면위에 2개의 축으로 정의역에 해당하는 치역의 대응관계를 점으로 찍어 보는 것이다. 다음의 예는 $f(x)=x^2, 0 \leq x \leq 10$ 함수를 그래프로 표현한 것이다.

```
>plot({x^2}, x=0..10)
```



[예 4] Maple에서 함수의 그래프 표현

2.3 알고리즘 탐구학습과 Maple

Maple은 사용자에게 수학의 원리, 개념, 이해와 관련된 계산 환경을 제공한다. Maple에 내장된 명령어나 라이브러리 그리고 패키지는 사용자가 지시한 입력에 대해 즉각 출력해주는 black box 형태이다. 그러므로 사용자는 계산을 수행하는 명령어를 단순히 실행하여 그 결과를 얻는다고 해서 수학의 원리, 개념, 이해를 증진시킨다고 보기 어렵다. 그러나 사용자가 어떤 수학문제나 알고리즘을 해결하기 위하여 필요한 명령어를 선택 조합하는 일련의 과정은 수학적인 추론을 하는 것과 같다. 문제해결에 관련된 함수 또는 모듈을 찾아내고 그 함수 또는 모듈에 대응되는 Maple 명령어를 선택하여 조합하는 일련의 절차

는 수학적 사고에 대한 스크립트를 작성하는 것과 같다. 다시 말하면, Maple의 내장된 기본 명령어를 합성함으로써 수학 문제를 해결하는 것은 궁극적으로 수학 지식을 구축하는 과정으로 볼 수 있다. 이와 같이 조합된 명령어 실행과정을 단계마다 학습자가 볼 수 있다면 이것이 white box이다. 본 논문에서 개발된 워크시트는 그래프이론을 쉽게 알 수 있도록 전체 black box를 단계별 white box로 나누어 나타냈다.

3. 그래프를 표현한 Maple 워크시트

2장에서 Maple의 기능, 특성, 식의 표현방법 그리고 학습 탐구에 필요한 알고리즘을 white box형태로 표현하는 방법을 보였다. 여기에서는 그래프 이론을 학습자가 보다 쉽게 이해할 수 있도록 white box형태로 Maple 워크시트를 개발 하였다. 이 장은 그래프 이론 학습을 위해 작성된 워크시트 중 몇 가지의 워크시트 내용을 소개하고 개발된 웹 기반 워크시트의 목록과 구현된 웹 사이트 화면을 보였다.

3.1 워크시트

워크시트의 내용 중 중요한 몇 가지를 소개한다. 다음은 그래프의 정의 및 기본 용어 소개, 오일러 경로를 이해하기 위한 워크시트이다.

정의 : 단순 그래프 $G=(V, E)$ 에서 V 는 공집합이 아닌 정점들의 집합이고, E 는 간선으로서 방향성이 없는 정점들의 쌍의 집합이다.

```
>with(networks);
G:=new();
addvertex(s, b, g, d, c, u, i, G);
adddedge({(s, c), (s, i), (i, b), (b, g), (g, d), (g, i), (g, u), (d, u), (u, c)}, G);
ends(G);
draw(G);
```

설명 : 6개의 노드 (s, b, g, d, c, u, i)와 9개의 에지 {(s, c), (s, i), (i, b), (b, g), (g, d), (g, i), (g, u), (d, u), (u, c)}를 가진 단순 그래프 $G=(V, E)$ 를 그래픽으로 나타낸다.

정의 : 차수(degree)는 단순 그래프에서 정점에 결합된 간선의 수이다. 입력차수(indegree)는 방향 그래프에서 종료 정점으로서 v 가 갖는 간선의 수이다. 출력차수(outdegree)는 방향 그래프에서 시작 정점으로서 v 가 갖는 간선의 수이다.

```
>vdegree(d, G);
>indegree(d, G);
>outdegree(d, G);
```

설명 : vdegree()는 그래프 G의 정점 d의 차수를 출력한다. indegree(), outdegree() 라이브러리는 방향 그래프에서 정점 d의 입력차수와 출력차수를 출력한다

정의 : 인접행렬은 n개의 정점을 가진 그래프 G에서 정점 i와 j가 인접하고 있다면 i행 j열을 1로 나타내는 $n \times n$ 행렬이다. 결합행렬은 n개의 정점과 m개의 간선을 가진 그래프 G에서 정점 i와 간선 j가 결합되어 있다면 i행 j열에 1을 나타내는 $n \times m$ 행렬이다.

```
> AdjList:=proc(G::graph)
local i;
for i from 1 to nops(vertices(G)) do
print(vertices(G)[i], 'is adjacent to', neighbors(vertices(G)[i],G));
od;
end;
```

> adjacency(G);
> incidence(G);

설명 : 그래프 G의 인접 리스트를 출력하는 프로시저와 인접행렬, 결합행렬을 출력하는 adjacency(), incidence() 라이브러리로 이루어져 있다.

정의 : 그래프 G가 연결그래프라면 그래프 G가 오일러 그래프가 되기 위한 필요충분조건은 그래프 G의 모든 정점이 짝수개의 차수를 가져야 한다.

```
> Eulerian:=proc(G::graph)
local v,V,is_eulerian;
if hastype(degreeseq(G),odd) then false else true fi;
end;
```

설명 : 위의 코드는 그래프 G의 정점들이 홀수 차수를 가지고 있는지를 검사한다. degreeseq() 라이브러리를 사용하여 정점 차수리스트를 만들고, hastype() 라이브러리를 사용하여 차수가 홀수 인지 를 검사한다.

Maple에서 제공하는 하나의 black box에서 출발하여 그 알고리즘을 현재의 데이터에 적용해보고 일종의 실험적 수학 활동을 거쳐 나름대로 수학적 현상을 관찰하거나 특정 부분에 대하여 수학적 추측을 할 수 있도록 유도해야 한다. 그리고 이런 추측은 수학적 추론에 따라 증명을 시도하는 단계로 나아가게 되고, 증명을 하든지 아니면 반례를 제시함으로써 추측한 부분을 채택 또는 기각하는 검증 절차를 마치게 된다. 만약, 추측한 부분이 증명되었다면, 그 부분에 한하여 그 증명이 제시하는 알고리즘을 프로그래밍 언어로 구현하여 프로시저를 작성하게 되는데 이런 컴퓨터 대수 시스템을 이용하는 것을 black box/white box 원리라 한다. 이런 black box/white box 원리를 이용할 수 있도록 Maple로 작성된 워크시트를 온라인 콘텐츠로 변환하기 전에 데이터 프로세싱 → 추측 → 알고리즘 탐색 → 프로그래밍 단계의 탐구 사이클에 부합되는 탐구 학습용 모듈을 개발하는 단계가 필요하다. 위와 같은 학습을 위한 수업에서의 단계를 다음과 같은 5단계로 나누어 형성할 수 있다.

- ① 준비학습단계, ② 학습목표제시, ③ 본시학습 단계, ④ 학습 정리 단계, ⑤ 평가단계

이 장에서는 학습을 위한 추론적이고 구체적인 탐구학습용 white box를 워크시트로 만들었다.

3.2 워크시트의 구성

워크시트는 트리의 형태로 구성하였으며 학습자가 이해하기 쉽도록 그래프의 기초적인 이론부터 좀 더 복잡한 이론으로 순차적으로 구성했다. 다음은 워크시트의 구성도이다.

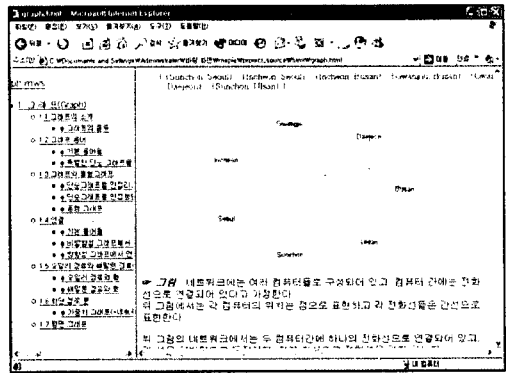
- 1. 그래프(Graph)
 - 1.1 그래프의 소개
 - 기본 그래프의 종류
 - 1.2 그래프 용어
 - 기본 용어
 - 특별한 단순 그래프들
 - 1.3 그래프와 동형 그래프
 - 단순 그래프를 인접리스트로 표현
 - 단순 그래프를 인접행렬과 결합행렬로 표현
 - 동형 그래프
 - 1.4 연결성
 - 기본 용어
 - 비방향성 그래프에서 연결성
 - 방향성 그래프에서 연결성
 - 1.5 오일러 경로와 해밀턴 경로(Euler and Hamilton Paths)
 - 오일러 경로와 회로
 - 해밀턴 경로와 회로
 - 1.6 최단 경로 문제
 - 가중치 그래프(네트워크)
 - 1.7 평면 그래프

3.3 워크시트의 변환

Maple의 구조적 수식처리기능을 웹 기반 문서로 변환하였다 이것은 단지 Maple에서 제공하는 기능을 이용한 것이다 좀 더 향상된 인터페이스를 제공하고 추측과 반론을 위한 학습자와의 상호 데이터 통신을 위해 Maple-자바 API와 웹서버 API를 연동하는 기술을 개발해야 한다. 이 연구는 OpenMath의 표준화 동향과 기술을 도입해야 한다. 다음은 변환된 워크시트의 실행 화면이다.

4. 결론

본 연구는 국내에서 거의 연구되지 않는 컴퓨터 대수 시스템의 콘텐츠를 개발함으로써 인접분야인 이산수학과 알고리즘 분야의 연구를 촉진하도록 하였다. 또한 개발된 콘텐츠를 웹기반 탐구학습용으로 변환함으로써 웹 관련 기술의 개발에 기여하고, 컴퓨터, 수학 관련학과 학생들의 알고리즘 탐구 학습능력을 고양하는 온라인 교육을 지원한다. 그러나 이 방법은 학습자와의 상호 작용이 없기 때문에 각 단계별로 사용자가 다음단계의 진행을 결정하거나 컴퓨터가 다음 단계의 진행을 결정해야 하는데 “어떻게 어느 단계까지 진행 할 것인가?”는 아주 중요한 문제이며, 연구가 진행되어야 한다. 추가로 그래프 이론뿐만 아니라 공학 분야나 컴퓨터 알고리즘 분야에 더 많은 콘텐츠를 개발해야 한다. 또한 학습을 위한 수업에서의 준비학습단계 학습목표제시, 본시학습 단계, 학습정리 단계, 평가단계를 완성된 워크시트에 맞추어 작성해야 한다.



참고문헌

- [1] M. J. Wester (Eds.), Computer Algebra Systems. Jogn Wiley, 1990.
- [2] David Tall (Ed.), Advanced mathematical thinking. Kluwer Academic Pub, 1991.
- [3] H. Lewis and C. Papadimitriou, Elements of the theory of computation. Prentice Hall, 1981.
- [4] B. Mishra, Algorithmic Algebra. Springer-Verlag, 1999.
- [5] Karian, Z. (Ed.) symbolic computation in undergraduate mathematics education. MAA Notes
- [6] Bruce Char, Keith Geddes, Gaston Gonnet, Benton Leong, Michael Monagan and Stephan Watt, First Leaves:A Tutorial Introduction to maple V, Springer-Verlag, 1992.
- [7] Bruce Char, Keith Geddes, Gaston Gonnet, Benton Leong, Michael Monagan and Stephan Watt, First Leaves: The Maple V Language Reference Manual, Springer-Verlag, 1991.