韓國數學敎育學會誌 시리즈 E <數學敎育 論文集>
https://doi.org/10.7468/jksmee.2024.38.2.247
제 38집 제 2호, 2024. 6. 247-261

J. Korea Soc. Math. Ed. Ser. E:
Communications of Mathematical Education
Vol. 38, No. 2, Jun. 2024. 247-261

프로그래밍을 이용한 복잡한 도형의 한붓그리기 문제의 교육적 가능성 탐색

정 용 욱 (경상국립대, 교수)

본 연구는 스크래치 같은 드래그 앤 드롭 방식 교육용 프로그래밍 언어를 활용하여 복잡한 도형의 한붓그리기 과제를 해결하는 활동의 교육적 활용가능성을 논의하고자 한다. 주어진 도형이 한붓그리기가 가능한지를 판별하는 문제와 실제로 한붓그리기의 경로를 찾아서, 프로그래밍으로 구현하는 것은 별개의 문제가 된다. 특히 규칙성을 가지는 복잡한 도형에 대해 한붓그리기가 가능한 규칙적인 경로를 찾고, 이를 프로그래밍으로 구현하는 것은 다양한 수학지식의 융합을 바탕으로 하는 문제해결 역량을 요구한다. 이에 본 연구에서는 다각형 관련 도형들, 테셀레이션 관련 도형들, 프랙탈 도형들 중에 한붓그리기와 관련된 문제를 제시하고, 해당 도형의 한붓그리기 프로그래밍 결과를 예시하였다. 또 예시된 문제의 해결과정을 위해 필요한 수학지식과 계산적 사고 요소들을 분석하였다. 본 연구는 수학과 정보가 융합하는 수학교육에 대한 새로운 예시라는 의미를 갖는다.

Ⅰ. 서론

수학 교육을 통해 학생들은 수학의 지식을 이해하고 기능을 습득하는 것을 넘어서, 문제해결, 창의융합, 정보처리 등과 관련된 다양한 역량을 키울 수 있어야 한다(교육부 2022). 특히 최근에 인공지능의 충격에 대한 대비가 수학교육에서도 요구되고 있다. 이를 위해 교육과정에서 인공지능 관련 수학 지식이 적절히 포함되어야 한다는 요구가 거세며, 수학과 정보의 만남을 위한 계산적 사고가 주목받고 있으며, 수학과 프로그래밍이 융합된 수학교육의 필요성도 제기된다(이승우 2020; 권미선 2024).

최근에 수학교육과 계산적 사고의 접점을 찾으려는 여러 시도들이 있었다(임해미 외 2014; 장경윤 2017; 신동조, 고상숙, 2019; 강하람 외 2021; 박지연, 2023). 이러한 융합이 실질적으로 시도될 때 스크래치와 엔트리같은 블록기반 교육용 프로그래밍 언어(EPL)들이 활발하게 활용되고 있다(강순자 등 2020; 정진환, 조한혁, 2020). 이 언어들은 블록을 선택하고 이동하여 붙이는 드래그 앤 드롭(drag & drop) 방식으로 프로그래밍이 이루어지면서, 프로그래밍 언어 습득에 필요한 부하를 줄일 수 있기 때문에 교육적 활용가능성이 높다. 최근에는 정수론, 기하학을 중심으로 영재교육의 맥락에서 EPL을 활용한 여러 수학교육 콘텐츠들이 개발, 활용되고 있다. 이를테면 스크래치를 활용한 아트패턴 그리기, 프랙털 도형 그리기는 영재교육의 콘텐츠로 주목받고 있다(강순자 등 2019).

본 연구에서는 EPL과 연동되는 새로운 수학교육의 주제로 한붓그리기에 주목하고자 한다(최근배, 2005). 오일러 정리를 이용하면 주어진 그림에 대해 매우 기계적인 방식으로 한붓그리기가 가능한 지를 판별할 수 있다. 그렇지만, 한붓그리기가 가능하다는 것이 확인된 이후에도 규칙성을 띠는 복잡한 형태를 갖는 그래프에 대해 실제로 한붓그리기의 경로를 찾는 것은 학생들에게 도전적인 과제일 수 있다. 또한 손으로 그린 한붓그리기 경로를 프로그래밍을 통해 컴퓨터로 구현하는 것은 학생들에게는 또 다른 차원의 과제일 수 있다. 여기에 주목하여 본

* 접수일(2024년 2월 29일), 심사(수정)일(2024년 5월 7일), 게재확정일(2024년 6월 10일)
* MSC2020분류 : 97K30, 97P40
* 주제어 : 한붓그리기, 프로그래밍, 스크래치, 테셀레이션, 프랙탈

연구에서는 복잡한 것처럼 보이지만 규칙성을 갖는 한붓그리기 가능한 그림에 대해 스크래치를 이용하여 한붓그리기를 구현하는 문제의 교육적 가능성에 대해 논의하고자 한다. 이를 위해 본 연구에서는 규칙성과 복잡성을 함께 띠는 도형 중에 한붓그리기 문제로 적합한 유형들을 탐색하였다. 본 연구에서 주목한 문제유형은 세 가지로, 다각형과 관련된 한붓그리기, 테셀레이션과 관련된 한붓그리기, 프랙탈 도형과 관련된 한붓그리기가 그것이다. 각 유형별로 교육용 프로그래밍 언어로 한붓그리기를 예시적으로 구현하였다. 또 이러한 한붓그리기 구현을 위해 활용되는 수학지식과 수학 교과역량, 계산적 사고를 분석하여 제시하였다. 끝으로 프로그래밍을 통한 한붓그리기 문제들이 갖는 교육상의 장점에 대해 논의하였다.

Ⅱ. 연구의 배경

문제해결에서 컴퓨터가 점점 중요한 역할을 담당하게 되면서 수학교육에서 계산적 사고가 점차 주목받고 있다. 계산적 사고는 컴퓨터 과학의 기본개념을 바탕으로 한 문제해결체계와 관련된다(Wing 2006). 신동조와 고상숙(2017)은 수학교육에서 계산적 사고의 의미을 탐색하고 계산적 사고와 수학적 사고의 관계에 대해서도 분석하였다. 여러 연구자들이 다양한 방식으로 계산적 사고의 하위 실행을 범주화하여 제안하고 있다. 이러한 하위 실행에는 복잡한 문제를 다루기 쉬운 작은 단위로 쪼개는 분해, 문제에서 규칙성을 찾는 패턴인식, 중요하지 않은 요소들을 무시하고 중요한 정보에만 초점을 맞추는 추상화 등이 포함된다(장경윤 2017).

한붓그리기란 정점(vertex)과 변(edge)으로 구성된 주어진 그래프에 대해 붓을 종이에서 떼지 않고 모든 정점과 변을 지나도록 그리되, 하나의 변은 한 번만 지나도록 그리는 것이다. 주어진 그래프에 대해 모든 변을 반복 없이 지나는 경로를 오일러 경로(Eulerian path)라 부른다. 오일러 경로를 찾을 수 있는 그래프 중에 시작점과 끝점이 같으면서 그래프의 모든 변을 한 번씩만 지나는 경로를 오일러 회로(Eulerian circuit), 또는 오일러 순환(Eulerian cycle)이라 부른다. 한 정점에 연결된 변의 개수가 홀수인 정점이 없거나 2개인 그래프에 대해 오일러 경로를 찾을 수 있고, 특히 한 정점에 연결된 변의 개수가 모두 짝수인 그래프에 대해서는 오일러 회로를 찾을 수 있다. 그 외의 경우에는 한붓그리기가 불가능하다(박주미, 2021).

한붓그리기와 관련된 기존의 교육적 논의들은 주로 주어진 그래프에 대해 한붓그리기가 가능한지를 판별하는 쪽에 관심을 두었다고 할 수 있다. 오일러의 정리를 적용하면 주어진 그래프에 대해 한붓그리기가 가능한지를 판별하는 것은, 쉽고 기계적인 과정이 된다. 그런데 한붓그리기가 가능하다는 것을 판별하는 것과, 주어진 그래프에 대해 한붓그리기가 가능한 경로를 찾아내는 것은 다른 문제이다. 또한 컴퓨터 프로그래밍으로 한 붓그리기를 구현하는 것은 단순히 한붓그리기의 가능한 경로를 손으로 찾아내는 것 이상을 요구한다. 한붓그리기의 경로는 다양할 수 있지만, 그 중에서 규칙성을 갖는 경로를 찾아야 프로그래밍으로 구현하기 쉽기 때문이다. 이에 착안하여 본 연구에서는 한붓그리기가 가능하다고 판단되면서, 규칙성을 띠는 복잡한 도형에 대해 실제로 한붓그리기의 경로를 찾아서 프로그래밍으로 구현하는 문제의 교육적 활용가능성에 주목하고자 한다.

손으로 한붓그리기를 구현하는 것과 비교할 때 컴퓨터를 통한 한붓그리기 구현은 수학지식의 정확하고 명시적 활용을 촉진할 수 있다. 손으로 한붓그리기 경로를 찾는다면 주어진 도형을 손으로 따라 그리면서 한붓그리기가 가능한 경로를 찾으면 된다. 이때 관련된 순서와 규칙을 명시적으로 명제화하지 못하거나, 암묵적 차원에서만 규칙성을 찾을 수도 있다. 이렇게 찾은 한붓그리기 경로를 컴퓨터로 구현하려면 한붓그리기의 과정과 절차를 보다 명시적으로 표현해야 하고 관련된 수학지식의 활용도 동반되어야 한다. 이를테면 정삼각형을 손으로 따라 그려서 한붓그리기 경로를 찾는 문제의 경우, 손으로 도형을 따라 그리면 그 과정에서 정삼각형의 특징에 대한 지식을 명시적으로 활성화하지 않고, 한붓그리기의 경로를 쉽게 찾을 수 있다. 그런데 이 정삼각형 그리기를 프로그래밍으로 구현하려면, 정삼각형의 세 변의 길이를 동일하게 하고, 한 변에서 다른 변으로 넘어갈 때 정확하

게 120°(외각) 만큼의 방향 회전이 필요하며, 기본 선긋기가 3차례 반복되어야 한다는 보다 명시적인 규칙성을 도출해야 한다. 결과적으로 컴퓨터가 수행할 수 있는 수준으로 한붓그리기의 과정을 명세화 하는 것은 손으로 한붓그리기를 하는 것보다 구체적이고 명시적인 규칙성의 추출과 수학지식의 활용을 요구한다. 이런 점에서 손으로 한붓그리기 경로를 찾는 것보다 컴퓨터 프로그래밍을 통해 한붓그리기 경로를 구현하는 것이 수학지식의 명시적 활성화를 촉진한다는 장점이 있다.

그런데 프로그래밍 과정에서 학생들이 프로그래밍 언어를 배우는 인지적 부담이 크다면, 한붓그리기 프로그래밍에서 수학에 초점을 둔 교육이 어려울 수 있다. 다행히도 스크래치와 엔트리같이 블록을 선택, 이동하여 붙이는 드래그 앤 드롭(drag & drop) 방식의 교육용 프로그래밍 언어를 통하면, 프로그래밍을 익히는 데 필요한 인지적 부담을 낮출 수 있다. 실제로 정수론, 대수, 기하학, 프랙탈 기하, 스트링 아트와 관련하여 스크래치를 이용한 교육 콘텐츠가 개발되어 활용되고 있다(강순자 외 2019; 강순자 외 2020).

스크래치로 한붓그리기를 구현하는 기초 문법도 상당히 쉬운 편이다. 스크래치는 스프라이트의 동작과 관련하여 직진 이동하거나 방향을 바꿔주는 블록, 스프라이트의 이동 궤적을 따라서 선을 그려주는 블록을 포함하는데, 이 블록들을 조합하면 스프라이트가 이동하면서 한붓그리기를 구현할 수 있다. [그림 II-1]은 오일러 경로그리기를 위한 기본 블록들과 반복기능을 위한 블록의 조합으로 삼각형 한붓그리기를 구현한 기본 프로그래밍이다. 이 프로그래밍에서 박스 A 안에 들어있는 부분은 한붓그리기를 위한 준비이며, 박스 B에 들어있는 부분이 삼각형을 그리는 파트이다. 박스 A의 프로그래밍은 그대로 놔두고 박스 B의 프로그래밍을 바꾸어주면서 다양한 한붓그리기 그림을 그릴 수 있다. 박스 A는 스프라이트가 그림을 그리는 시작점, 처음 그림을 그리는 방향 같은 세부적인 규정을 포함한다. 그림에서 보듯이 학생들이 한붓그리기 프로그래밍을 위해 익혀야 할 최소한의 프로그래밍 문법은 매우 단순하다.

[그림 II-1]한붓그리기 프로그래밍의 기본 구조

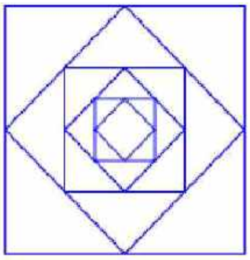
본 연구에서는 복잡한 도형에 대해 한붓그리기의 실제 경로를 찾는 문제의 몇 가지 유형을 제시하고, 각 유형별로 사례를 제시하여 교육용 프로그래밍 언어로 한붓그리기를 구현하였다. 본 연구에서 주목한 문제유형은 세 가지로, 다각형과 관련된 한붓그리기, 테셀레이션과 관련된 한붓그리기, 프랙탈 도형과 관련된 한붓그리기가 그것이다. 정다각형 기반 한붓그리기란 정다각형 속에 정다각형이 반복되거나, 정다각형과 대각선을 모두 이은 도형처럼 정다각형에 기반을 둔 경우를 말한다. 테셀레이션 기반 한붓그리기란 일정한 형태의 도형들로 평면을 빈틈없이 채우는 상황과 관련된 한붓그리기를 말한다. 프랙탈 도형과 관련된 한붓그리기는 시어핀스키 삼각형의 경계선 같은 프랙털 도형과 관련된 한붓그리기를 말한다. 본 연구는 세 영역에서 한붓그리기 경로 찾기에 대한 문제해결의 사례를 제시하고, 각 문제를 해결할 때 활용되는 수학지식, 계산적 사고를 분석하여 제시하였다. 이를 바탕으로 각 한붓그리기 문제들이 수학교육에 어떻게 활용될 수 있는지를 논의하였다.

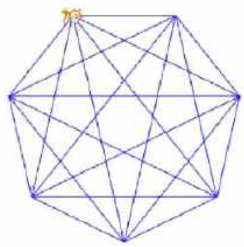
Ⅲ. 연구 결과 및 논의

프로그래밍을 통해 한붓그리기로 그릴 그림을 탐색한 결과 본 연구에서는 규칙성을 띠면서 복잡한 형태를 갖는 세 가지 유형의 한붓그리기 경로찾기 도형(그래프)을 찾아서 예시적으로 오일러 경로(회로)를 구현하였다. 아래에 각각의 유형별로 한붓그리기를 예시하고 필요한 수학적 지식과 계산적 사고를 분석하여 제시하였다.

1. 정다각형에 기반한 한붓그리기 문제

정다각형에 기반한 한붓그리기 문제는 정다각형 속에 정다각형이 반복되거나, 정다각형과 대각선을 모두 이은 도형처럼 정다각형에 기반을 둔 한붓그리기 가능한 도형에 대해 오일러 경로(회로)를 프로그래밍으로 구현하는 것을 목표로 한다. 정다각형 관련 한붓그리기가 가능한 도형의 예시는 [그림 Ⅲ-1]과 같다. 제시된 도형 이외에도 정다각형과 원이 중첩되는 장미도형 등도 정다각형에 기반한 한붓그리기로 그릴 수 있다. 정다각형 속 정다각형은 각 정점을 지나는 변의 개수가 모두 짝수이므로, 오일러 회로를 찾을 수 있다. 한편 홀수인 n에 대해 정 n각형과 그 대각선을 모두 이은 도형도 각 정점을 지나는 변의 개수가 모두 짝수이므로 오일러 회로를 찾을 수 있다.

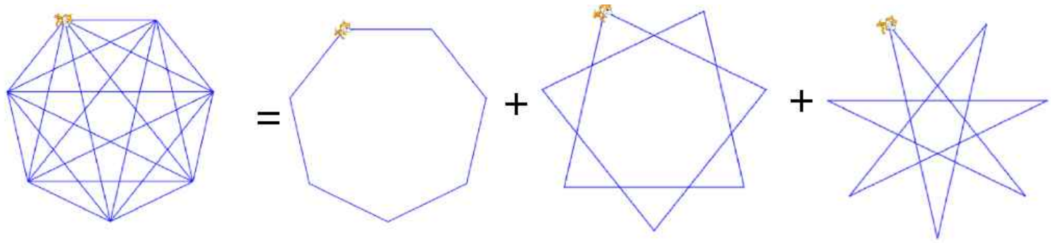




[그림 Ⅲ-1] 정다각형에 기반한 한붓그리기 도형의 예시

스크래치를 이용한 다각형기반 장미꽃 그리기 문제나, 정다각형과 그 대각선을 모두 이은 도형은 영재교육 등을 중심으로 이미 활용되고 있는 문제이다(강순자 외 2019). 그렇지만 기존의 문제는 한붓그리기의 조건을 만족하지 않고 붓을 때서 이동하여 그리는 해답을 제시하였다. 본 연구는 기존의 접근과 달리 도형을 그릴 때 한붓그리기 조건을 추가하여 구현한다는 점에서 기존의 교육콘텐츠와 차이가 있다. 또한 한붓그리기라는 조건의

추가로 기존의 과제보다 난이도가 높아지며, 한붓그리기라는 조건을 만족하는 새로운 규칙성을 갖는 그리기 절차를 찾아야 한다.



[그림 III-2] 정다각형과 대각선 모두그리기 문제를 별다각형 그리기의 합으로 분해한 도식

정다각형과 대각선 모두 그리기의 경우 원칙적으로는 오일러 정리에 의해 홀수인 n에 대해, n각형과 그 대각선을 모두 이은 도형에 대한 한붓그리기가 가능하다. 그런데 n이 소수가 아닌 홀수의 경우, 소인수분해와 관련되어 오일러 회로를 실제로 구현하는 알고리즘이 상대적으로 복잡해진다. 본 연구에서는 홀수 소수인 n에 대해서 오일러 회로를 스크래치 프로그래밍으로 예시적으로 구현하는 문제를 분석하였다. 홀수 소수 다각형과 대각선으로 이루어진 그래프에 대한 한붓그리기 경로의 개요를 n=7인 경우에 대해 도식적으로 설명한 것이 [그림 III-2]이다. 그림처럼 우선 다각형을 그린 후에 길이가 같은 대각선으로 이루어지는 정규 별다각형을 차례로 그려서 합치면 원하는 오일러 회로가 구현된다. 다각형과 별다각형을 그리는 순서가 달라도 한붓그리기가 가능하므로, 다양한 경로의 오일러 회로가 가능하다. 다각형의 모든 변의 길이가 같고, 모든 각이 같고, 변들이 규칙적으로 교차하여 만들어진 다각형을 정규 별다각형(regular star polygon)이라 한다. 정규 별다각형은 순서쌍 (n, m)으로 나타낼 수 있는데, 여기서 n은 꼭지점의 수, m은 turning number로 불리며 한 꼭지점과 연결되는 이웃 꼭지점까지 건너뛴 수를 말한다. 그림에 나타난 두 정규 별다각형은 각각 (7,2), (7,3) 에 해당한다. 결국 다각형과 대각선을 모두 그리는 한붓그리기 과제는 정규 별다각형과 관련된 응용 문제가 된다. '정다각형과 대각선 모두 그리기' 문제에 대해 별다각형 그리기를 합하는 방식으로 구현하는 오일러 회로는 구체적으로 다음과 같은 순서를 통해 구현될 수 있다.

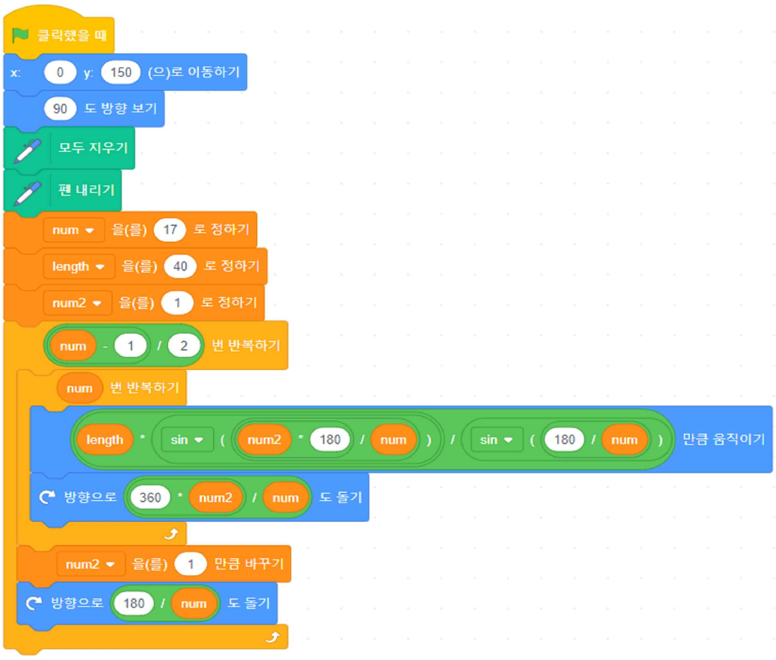
1) 한변 (길이 l)을 그리고 방향을 외각(180º/n)만큼 시계방향으로 회전하는 유닛을 n번 반복하여 다각형을 그린 후 시계방향으로 외각만큼 회전한다.

2) 가장 짧은 길이를 갖는 대각선(길이 $l\sin(2\pi/n)/\sin(\pi/n)$)을 그리고 외각의 2배만큼 시계방향으로 회전하는 유닛을 n번 반복하여 가장 짧은 대각선으로 이루어진 정규 별다각형(n,2)을 그린 후 시계방향으로 외각만큼 회전한다.

3) (m이 3에서 (n-1)/2가 될 때 까지 다음을 반복) m-1번째로 짧은 길이를 갖는 대각선 ($l\sin(m\pi/n)/\sin(\pi/n)$)을 그리고 외각의 m배만큼 시계방향으로 회전하는 유닛을 n번 반복하여 m-1번째로 짧은 대각선으로 이루어진 정규 별다각형 (n,m)을 그린 후 시계방향으로 외각만큼 회전하고 m을 1만큼 키운다.

n이 소수인 홀수이면, m<(n-1)/2 인 모든 m에 대해 n과 m이 서로 소가 되어, 위와 같이 그림을 그릴 때 각 단계마다 정규 별다각형 (n,m)이 그려진다. 결과적으로 모든 변과 모든 대각선을 빠짐없이 한붓그리기로 나타낼 수 있다. 결과적으로 한붓그리기 문제의 해결을 위해 삼각함수와 대각선의 길이와 관련된 수학지식을 활용해야 할 뿐 아니라 한붓그리기와 무관해 보이는 '서로 소'라는 정수론의 개념도 활용해야 한다. 이과 같이 프로그래밍을 이용한 한붓그리기 구현은 문제에 따라 다양한 수준의 수학지식의 활용까지 요구할 수 있다. 계산적 사고의

측면에서는 한붓그리기 문제를 여러 개의 별다각형 그리기의 합으로 바꾸는 분해, 별다각형에 대한 패턴인식, 반복문의 적절한 활용에 대한 알고리즘 등의 요소가 필요하다. 결과적으로 한붓그리기 문제를 해결하기 위해서 학생 입장에서는 상당히 복합적인 지식과 사고를 동원해야 한다. 이 문제에 대해 스크래치로 구현한 프로그래밍 결과가 [그림 Ⅲ-3]이다. [그림 Ⅲ-3]에서 n각형을 나타내는 변수로 num을, 한 변의 길이 l을 나타내는 변수로 length을, turning number를 나타내는 변수로 num2을 각각 지정하였다. 규칙성을 가지는 방식으로 한붓그리 경로를 찾았기 때문에, 그림과 같이 프로그래밍 결과를 단순화할 수 있다.



[그림 Ⅲ-3] 정다각형과 대각선 모두 그리기 문제에 대한 스크래치 프로그래밍

본 연구에서 제시한 정다각형과 대각선 모두그리기 문제에 대한 한붓그리기 알고리즘은 한붓그리기 차수가 홀수인 정점의 개수가 0이거나 2인 경우에 한붓그리기가 가능하다는 증명과 다르다. 이와 관련된 한 증명은 차수가 홀수인 정점의 개수가 0이거나 2인 그래프 G에 대해 한붓그리기를 시도하여 생성된 부분 그래프(이를테면 G1)이 G의 모든 변을 지나지 않았다면 G에서 G1을 제거한 부분그래프 G2도 차수가 홀수인 정점의 개수가 0이거나 2인 그래프라는 것을 이용하여 G의 모든 변을 한번 씩 지나가는 경로를 찾을 수 있다는 방식의 알고리즘을 포함한다(박주미, 2021). 그런데 이러한 증명에서 사용되는 알고리즘은 한붓그리기가 가능한 그래프의 조건에 대한 증명을 제공하지만, 이 알고리즘이 구체적인 도형에 대해 컴퓨터를 통해 한붓그리기 경로를 구현하는 코딩을 위한 지침을 제공해주지는 않는다. 코딩을 통해 한붓그리기 경로를 구현하려면 [그림 Ⅲ-2]과 같이 규칙성을 띠는 방식으로 한붓그리기 경로를 찾은 후에 [그림 Ⅲ-3]과 같이 그러한 규칙성을 코딩으로 구현하는 과정을 거쳐야 한다. 결과적으로 홀수인 정점의 개수가 0이거나 2인 경우에 한붓그리기가 가능하다는 증명과 한붓그리기가 가능한 도형에 대해 규칙성을 갖는 오일러경로를 찾고 이를 코딩으로 구현하는 것은 별개의 문제에 가깝다.

본 연구에서는 정다각형과 대각선 모두그리기 문제 외에도 정다각형에 기반하는 여러 한붓그리기 문제를 프

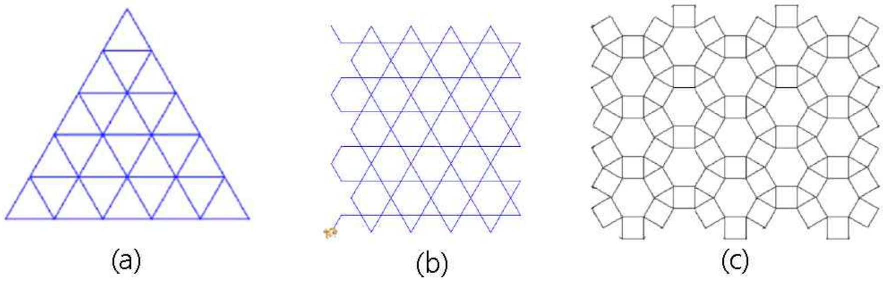
로그래밍으로 구현할 때 필요한 수학지식과 계산적 사고를 분석하였다. 예시적인 분석결과는 <표 Ⅲ-1>과 같다. 이를테면 정삼각형장미의 경우 정삼각형을 그리기 위해 정의와 외각 크기에 대한 지식이 필요하다. 또한 닮음비가 1:2인 도형을 반복하여 그려야 하므로 닮음비에 대한 이해가 필요하다. 한편 정사각형장미의 경우 닮음비가 $1:\sqrt{2}$인 정사각형 도형을 그려야 하는데 이를 위해 제곱근 개념 및 피타고라스 정리에 대한 이해가 필요하다. 한편 표에서 제시된 모든 도형을 코딩으로 간단하게 구현하려면 각 도형을 구성하는 유닛을 따로 쪼개어 생각하는 '분해' 과정이 필요하고, 반복되는 패턴속에서 규칙성을 도출하는 패턴인식이 필요하다. 또 인식된 패턴을 코딩으로 구현하는 알고리즘도 필요하다. 이와 같이 코딩으로 그 도형을 구현할 때 반드시 필요한 수학지식과 계산적 사고를 추출하였다.

<표 Ⅲ-1> 정다각형기반 한붓그리기 문제해결을 위한 수학지식, 계산적 사고에 대한 사례분석

한붓그리기 문제	필요한 주요 수학지식	필요한 계산적 사고
정삼각형장미 (정삼각형 속 정삼각형의 반복)	-정삼각형: 정의와 외각 크기 -닮음비	-분해 -패턴인식 -알고리즘(루프 반복)
정사각형장미 (정사각형 속 정사각형의 반복, [그림 Ⅲ-1] 참고)	-정사각형: 정의와 외각 크기 -닮음비 -피타고라스 정리 -제곱근	-분해 -패턴인식 -알고리즘(루프 반복)
정n각형 장미 (정다각형 속 정다각형 반복)	-정다각형: 정의와 외각 크기 -닮음비 -제곱근 -삼각비(코사인 법칙)	-분해 -패턴인식 -알고리즘(루프 반복)
정n각형과 대각선 모두 그리기 (n이 소수인 경우)	-정다각형: 정의와 내각, 외각 크기 -삼각함수(정다각형의 한 변과 대각선 길이의 비) -원주각 정리(원주각은 중심각의 두 배) -서로 소(정수론)	-분해 -패턴인식 -알고리즘(루프 반복)

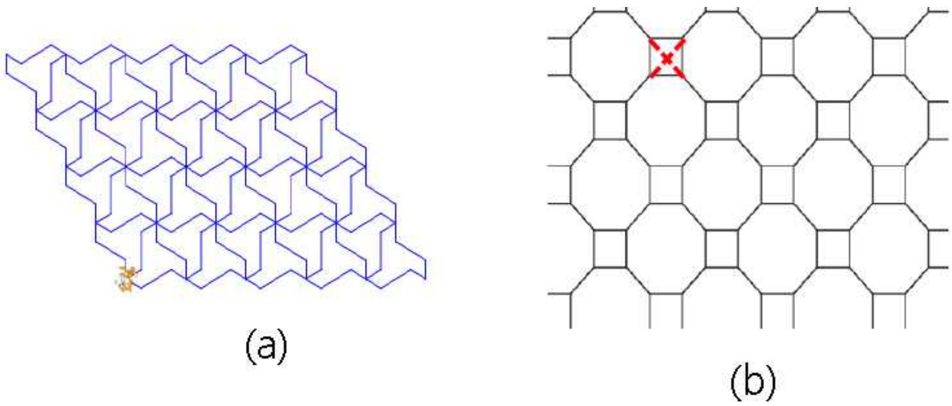
2. 테셀레이션 도형에 기반한 한붓그리기 문제

테셀레이션은 일정한 형태의 도형들로 평면을 빈틈없이 채우는 것을 말한다(임해경, 박은영 2002). 한 종류의 정다각형을 사용하여 평면을 매꾸는 것을 정규 테셀레이션(regular tessellation), 두가지 이상의 정다각형 도형들로 채우면서 임의의 점에서 정다각형의 구성이 동일한 것을 준정규 테셀레이션(semiregular tiling)이라 한다. 이 두 가지 조합으로 이루어지는 경우에는 demiregular 테셀레이션이라 한다. [그림 Ⅲ-4]는 각각 정규 테셀레이션, 준정규 테셀레이션, demiregular 테셀레이션과 관련된 한붓그리기 가능 도형을 예시하였다. 테셀레이션 기반 한붓그리기의 경우 무한히 반복하여 그릴 수 없으므로, 가장자리의 정점에 대해서는 변을 선별하여 일부만 남기는 식으로 오일러 정리를 만족하도록 경계를 조정한 후에 한붓그리기 경로를 찾아야 한다. 이를 반영하여 그림에서 가장자리 정점을 잇는 선분들의 일부를 삭제하여 각 정점을 지나는 변의 개수가 짝수가 되도록 정리하여 나타냈다.



[그림 III-4] 한붓그리기가 가능한 테셀레이션 관련 도형들. (a) 정규 테셀레이션 기반 (b) 준정규 테셀레이션 기반 (c) demiregular 테셀레이션 기반.

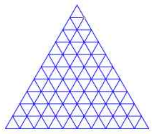
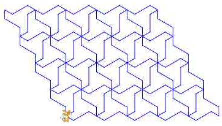
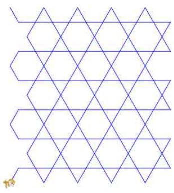
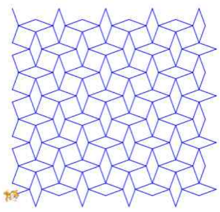
기본도형에서 변형된 형태의 테셀레이션 관련 한붓그리기도 가능하다. 이를테면 [그림 III-5]는 정삼각형의 한 변을 두 번 꺽인 선으로 치환하여 변형한 테셀레이션이다. 또한 한붓그리기가 불가능한 테셀레이션에 대해서 변을 추가하는 방식으로 수정하면, 한붓그리기가 가능한 그림으로 바꾸어줄 수 있다. 이를테면 [그림 III-5]의 (b)의 경우 각 정점을 지나는 변이 모두 홀수이므로 한붓그리기가 불가능하다. 그런데 그림 (b)의 왼편상단처럼 정사각형마다 두 대각선을 추가하면 각 정점을 지나는 변이 모두 짝수가 되어 한붓그리기가 가능한 도형으로 바뀌게 된다.



[그림 III-5] 기본 테셀레이션 도형의 변형과 한붓그리기

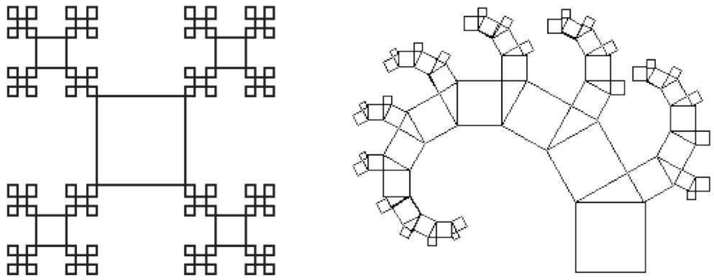
테셀레이션 한붓그리기의 경우, 필요로 하는 수학지식을 크게 요구하지 않으면서, 과제의 난이도를 높일 수 있다. 이를테면 정다각형 등 단순한 도형에 기반한 테셀레이션의 경우에 한붓그리기 문제를 해결하기 위해서 필요한 수학지식은 <표 III-2>처럼 정다각형에 대한 정의와 도형의 내각, 외각에 대한 기초적인 지식 뿐이다. 그런데 한붓그리기를 정확히 구현하려면, 반복되는 루프를 명확하게 규정하고, 루프가 갖는 규칙성을 정확하게 구현해야 한다는 점에서 과제의 난이도는 필요로 하는 수학지식에 비해 높다고 할 수 있다.

<표 III- 2> 테셀레이션에 기반한 한붓그리기 문제 해결에 필요한 수학지식 및 기능, 계산적 사고에 대한 사례분석

한붓그리기 문제	필요한 주요 수학지식 및 기능	필요한 계산적 사고
정규 테셀레이션의 변형 (피라미드식 쌓기) 	-정삼각형의 정의와 내각과 외각 크기 -규칙성 찾기(등차수열)	-분해 -패턴인식 -알고리즘(루프 반복)
정규 테셀레이션의 변형 (삼각형의 한 변을 두 번 꺽은 선으로 대체) 	-정삼각형 정의와 내각과 외각 크기 -규칙성찾기 -피타고라스 정리 -제곱근	-분해 -패턴인식 -알고리즘(루프 반복, 함수 활용)
준정규 테셀레이션 (정육각형-정삼각형 조합) 	-정삼각형, 정육각형 정의와 내각, 외각 크기 -규칙성 찾기	-분해 -패턴인식 -알고리즘(루프 반복)
정사각형-마름모 조합 	-정사각형, 마름모 정의와 내각, 외각 크기 -규칙성 찾기	-분해 -패턴인식 -알고리즘(루프 반복)

3. 프렉탈 도형에 기반한 한붓그리기 문제

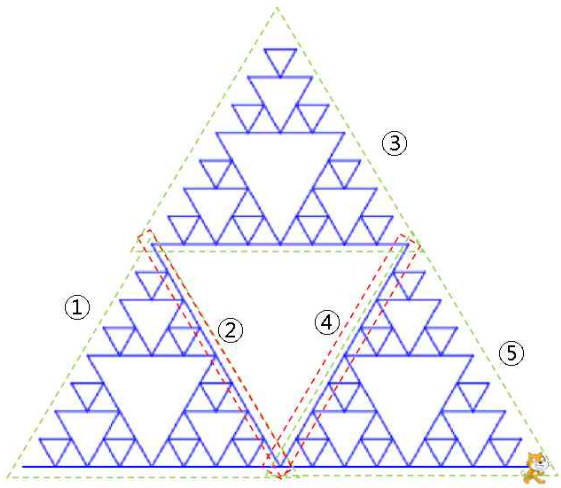
스크래치 등을 이용한 프렉탈 도형 그리기는 이미 영재교육을 중심으로 널리 활용되는 교육콘텐츠가 되고 있다(김상미 2009). 그런데 프로그래밍을 통해 그려지는 프렉탈 도형 중에는 [그림 III-6]처럼 한붓그리기가 가능한 도형들이 있다. 시에르핀스키 삼각형(Sierpinski Gasket graph)과 이것을 합성한 Sierpinski Gasket Rhombus graphs(Xavier et al., 2014), 피타고라스 나무, 코흐 눈송이 등이 그렇다. 이 도형들은 모든 정점에 짝수개의 변이 모이므로 오일러 회로를 그리는 것이 가능하다.



[그림 III-6] 프랙탈 도형과 관련된 한붓그리기 가능한 도형들.

본 연구에서는 시에르핀스키 삼각형에 대해 한붓그리기 프로그래밍을 구현하였다. 이때 [그림 III-7]처럼 시에르핀스키 삼각형에서 삼각형의 양변을 뺀 도형을 다음 과정을 통해 먼저 그린 후에 한붓그리기를 완성하는 방식을 활용하였다.

1) [그림 III-7]의 ①번 유닛(초록 점선으로 표현된 삼각형 영역 안의 파란색 도형)을 한붓그리기로 그린다.
2) ②번 유닛(빨간 점선 안의 파란색 선분)을 빗변을 올라가는 방향으로 그린다.
3) ③번 유닛(초록 점선으로 표현된 삼각형 영역 안의 파란색 도형으로 ①번 유닛과 같은 모양)에 해당하는 도형을 한붓그리기로 그린다.
4) ④번 유닛(빨간 점선 안의 파란색 선분)을 빗변을 내려가는 방향으로 그린다.
5) ⑤번 유닛(초록 점선으로 표현된 삼각형 영역 안의 파란색 도형으로 ①번 유닛과 같은 모양)에 해당하는 도형을 한붓그리기로 그린다.
6) 삼각형의 우변과 좌변을 차례로 그려서 시에르핀스키 삼각형을 완성한다.



[그림 III-7] 시에르핀스키 삼각형의 한붓그리기의 핵심 과정을 위한 분해

이때 ①, ③, ⑤번 유닛이 동형이고 이들 각각은 1)-5)의 과정을 통해 그릴 수 있는 전체 그림의 축소판이다. 즉 ①번 유닛의 그림은 다시 다섯 가지 하위 유닛으로 쪼개어 1)-5)의 과정과 유사하게 한붓그리기로 그릴 수

있다. 이러한 유사성에 착안하면 [그림 Ⅲ-8]과 같이 재귀함수를 정의하여 시에르핀스키 삼각형의 오일러 회로를 프로그래밍으로 구현할 수 있다. 코딩 속 함수정의에서 1-5)의 과정이 반영되도록 함수를 정의하였고 ①번 유닛의 가장 단순화된 기본구조도 정의하였다. 시에르핀스키 삼각형의 한붓그리기의 경우 오일러 회로를 그리기 위해 필요한 수학 지식은 정삼각형의 내각, 외각의 크기, 닮음비가 전부이다. 또 관련된 계산적 사고는 분해, 패턴인식, 알고리즘(재귀함수) 뿐이다. 이런 점에서 시에르핀스키 삼각형의 한붓그리기는 요구하는 수학지식이 상대적으로 적지만 난이도는 높은 수학문제라고 할 수 있다

[그림 Ⅲ-8] 시에르핀스키 삼각형의 한붓그리기 프로그래밍

Ⅳ. 결론 및 제언

본 연구에서는 복잡하지만 규칙성을 갖는 도형에 대해 프로그래밍을 통해 한붓그리기를 구현하는 것의 교육적 가능성을 탐색하였다. 이를 위해 정다각형에 기반한 도형들, 테셀레이션에 기반한 도형들, 프랙탈에 기반한 도형 들 중에 한붓그리기가 가능한 도형을 예시하고 이를 프로그래밍으로 구현한 후에, 문제 해결 과정에 필요한 수학 지식, 기능과 계산적 사고를 분석하였다.

한붓그리기와 관련된 문제는 다음과 같이 세분화되면서 확장될 수 있다; 1) 주어진 그림이 한붓그리기가 가능한지 판별하기, 2) 한붓그리기가 가능하도록 주어진 그림을 변형하기, 3) 한붓그리기가 가능한 경로를 손 그림으로 찾기, 4) 한붓그리기가 가능한 경로를 프로그래밍을 통해 구현하기. 이러한 다양한 가능성 중에서 기존에 가장 주목받았던 문제는 한붓그리기 가능여부의 판별 문제였다. 이러한 제한을 넘어서서 한붓그리기와 관련된 보다 여러 유형의 문제들이 수학교육에서 다루어질 필요가 있다.

학생의 문제해결 역량 증진은 수학교육의 중요한 목표 중 하나이다. 이를 위해 학교 수학에서 전형적으로 다루는 문제를 넘어서서, 학생의 역량 증진을 가능하게 하는 효과적인 수학문제의 개발이 필요하다. 좋은 수학 문제가 갖추어야 할 조건으로 이동환 등은 다양한 표현과 풀이가 가능한 열린 과제, 다양한 학생 수준을 고려한 낮은 문턱과 높은 천장의 과제, 학생의 자기주도성을 허용하는 과제, 탐구 기회가 제공되는 과제, 수학적 연결성을 형성하는 과제 등을 제시하였다(한국과학창의재단, 2019). 본 연구에서 제시한 복잡한 도형의 한붓그리기 실행은 이동환 등이 논한 좋은 문제의 요건을 두루 만족한다. 한붓그리기의 경우 여러 경로의 한붓그리기가 가능하다는 점에서 복수의 정답이 가능한 열린 과제이며, 한붓그리기 문제의 난이도는 다양하다. 또한 학생이 스스로 찾은 규칙성에 기반하여 한붓그리기를 구현해야 하며, 규칙 찾기와 반성적 사고 등 탐구의 기회를 제공한다. 또한 문제에 따라 다각형, 삼각함수 등의 기하학적 지식이외에도 정수론 지식 등 다양한 수학지식의 연결을 통해서 문제를 해결하게 된다.

한붓그리기 경로 찾기 문제는 수학 교과 역량의 증진에 적합한 주제이기도 하다. 한붓그리기 경로를 프로그래밍으로 구현하는 것은 수학의 지식과 기능을 활용하여 해결방법을 새롭게 찾기 위해 전략을 탐색하고 최적의 해결방안을 탐색할 기회를 제공한다는 점에서 문제해결 역량을 기울 수 있다. 또한 규칙성을 찾고 이를 프로그래밍을 통해 구현하는 동안 수학적 사실을 추측, 분석하고 올바른 해결책을 찾거나, 디버깅 과정에서 반성적 사고를 요한다는 점에서 추론역량을 키울 수 있다. 또한 위상기하학과 유클리드 기하학, 삼각비, 정수론 등 다양한 수학지식을 연결하여 통합적으로 문제를 해결하고 계산적 사고도 반영해야 한다는 점에서 창의융합 역량을 키울 수 있다.

한편 프로그래밍을 통한 한붓그리기 문제가 요구하는 프로그래밍 기능이 높지 않다는 장점도 있다. 본 연구에서 예시한 해결책에서 사용한 알고리즘은 순차문, 반복문, 함수, 재귀함수정도였고, 다수의 문제는 알고리즘에서 반복문을 적절히 사용하는 것만으로 해결되었다. 또한 본 연구에서 프로그래밍에 활용한 스크래치같은 교육용프로그래밍 언어들은 중학교에서 정보교과를 통해 교육과정 안에서 이미 학생들에게 소개되고 있다. 이런 점에서 프로그래밍을 통한 한붓그리기 문제 해결에 관련된 프로그래밍 기술은 학생들에게 새로운 것이 아니며, 문제해결의 결정적인 측면은 복합적인 프로그래밍기술이 아닌 수학적 사고라고 할 수 있다. 즉 프로그래밍을 통한 한붓그리기 문제는 정보와의 융합 맥락에서 등장한 문제이지만, 문제해결에 필요한 핵심이 수학적 사고라는 점에서 프로그래밍을 통한 한붓그리기 문제는 여전히 수학교육의 본질에 충실하다고 할 수 있다.

프로그래밍을 통한 한붓그리기 문제는 수학과 정보가 융합된 학생 탐구의 주제로 수학과 관련된 학생 탐구 맥락에서 활용할 수도 있다. 복잡성을 띠면서 한붓그리기가 가능한 여러 도형들이 있으므로 학생탐구에서 가장 어려운 단계일 수 있는 문제발견 과정을 가볍게 하고 대신에 문제해결에 집중할 수 있다. 본 연구에서 예시한 여러 문제들은 학생 탐구를 위한 도전적인 문제가 될 수 있을 것이다.

아쉽게도 현재 한붓그리기와 오일러 정리, 오일러 경로 등은 2022 개정 교육과정에서 포함되지 않았다(교육부, 2022). 이런 상황에서 한붓그리기 문제를 영재교육이나 교사교육의 맥락에서의 활용하는 것도 한 방안일 것이다. 그렇지만 한붓그리기 구현에 필요한 규칙성 찾기, 수열, 다각형, 삼각비에 대한 여러 지식들이 교육과정에서 소개되고 있고 규칙성을 갖는 경로를 찾는 것 자체는 많은 수학지식을 요구하지 않는다. 이런 점에서 프로그래밍을 통한 복잡한 도형의 한붓그리기 경로 구현은 여러 종류의 수학 지식과 기능의 융합을 요구하는 문제로 일선 학교 현장의 수학교육을 직접적으로 풍성하게 할 수도 있을 것이다.

참 고 문 헌

강순자・임해경・주재은 (2019). 스크래치와 함께하는 중등 수학: 기하편. 이모션북스.

Kang, S. J, Rim, H, & Joo, J. (2019). Secondary mathematics with Scratch: Geometry. Emotion Books.

강순자・임해경・김지원・주재은・장미라 (2020). 엔트리와 함께하는 중등 수학: 대수편. 이모션북스.

Kang, S. J, Rim, H, Kim J, Joo J., & Jang M., (2020). Secondary mathematics with Entry: Algebra. Emotion Books.

강하람・임채령・조한혁 (2021). 수학교과와 정보교과를 융합하는 코딩수학 교육과정 및 교육방법 연구. 수학교육, 60(4), 467-491.

Kang, H. R. , Lim, C. L., & Cho., H. H. (2021). A study on coding mathematics curriculum and teaching methods that converges school mathematics and school informatics. *The Mathematical Education*, 60(4), 457-491.

교육부 (2022). 수학과 교육과정. 교육부 고시 제2022-33호 [별책8].

Ministry of Education (2022). *Mathematics curriculum*. Ministry of Education Notice No. 2022-33 [Appendix 8].

권미선 (2024). 프로그래밍 교육 관련 일본 초등학교 수학 교과서 및 디지털 콘텐츠 분석. 초등수학교육, 27(1) 57-74.

Kwon, M. (2024) Analysis of Japanese elementary school mathematics textbooks and digital contents on programming education. *Education of Primary School Mathematics*, 27(1), 57-74.

김상미 (2009). 초등수학 영재교육원 학생들의 프랙탈 구성 방법 분석. 수학교육학연구, 19(2), 341-354.

Kim, S. M. (2009). A case study of constructions on fractals of the mathematically gifted. *Journal of Educational Research in Mathematics*, 19(2), 341-354..

박주미 (2021). 컴퓨팅 사고력을 키우는 이산수학. 한빛 아카데미(주).

Park, J. (2021). Discrete mathematics for developing computational thinking. Hanbit Academy.

박지연 (2023). 탐구형 소프트웨어를 활용한 합동과 대칭 수업에서 나타나는 초등학생의 귀납적 추론과 시각화에 관한 연구. 수학교육논문집, 37(2), 299-327.

Park, J. Y. (2023). A study on inductive reasoning and visualization of elementary school students in congruence and symmetry lessons with exploratory software. *Communications of Mathematical Education*, 37(2), 299-327.

신동조・고상숙 (2019). 수학교육에서 계산적 사고(Computational Thinking)의 의미 및 연구 동향 탐색. 수학교육, 58(4), 483-505.

Shin, D. & Koh, S. (2019). A study on investigation about the meaning and the research trend of computational thinking(CT) in mathematics education. *The Mathematical Education*, 58(4), 483-505.

이승우 (2020). 프랑스 중학교 수학 교육과정 분석: '알고리즘과 프로그래밍' 영역을 중심으로. 학교수학, 22(1), 125-159.

Lee, S. W. (2020). An analysis of the middle school mathematics curriculum in France: Focusing on 'Algorithms and Programming'. *School Mathematics*, 22(1), 125-159.

임해경・박은영 (2002). 컴퓨터 소프트웨어를 활용한 테셀레이션 교수 학습 자료 개발 및 활용 방안. 수학교육논문집, 13(2), 563-589.

Rim, H, & Park, E. Y. (2002). Development and utilization of tessellation teaching and learning materials using computer software. *Communications of Mathematical Education*, 13(2), 563-589.

임해미・최인선・노선숙 (2014). 논리·비판적 사고 신장을 위한 로봇 프로그래밍의 수학교육 적용 방안. 수학교육, 53(3), 413-434.

Rim, H., Choi, I., & Noh, S. (2014). A study on the application of robotic programming to promote logical and critical thinking in mathematics education. *The Mathematical Education*, 53(3), 413-434.

장경윤 (2017). 수학 교과에서 계산적 사고(Computational Thinking)교육. 학교수학, 19(3), 553-570.

Chang, K. Y. (2017). A feasibility study on integrating computational thinking into school mathematics. *School Mathematics*, 19(3), 553~570.

정진환・조한혁 (2020). 코딩교육 명령문의 수학화: 대수교육을 중심으로. 수학교육학연구, 30(1), 131-151.

Jeong, J. & Cho, h. (2020). Mathematising of coding education command: Focusing on algebra education, *Journal of Educational Research in Mathematics,* 30(1), 131-151.

최근배 (2005). 초등 영재교육에 적용 가능한 이산수학 주제의 내용 구성에 관한 소고 -네트워크 문제를 중심으로-. 학교수학, 7(4), 353-373.

Choi, K. B. (2005) A study on discrete mathematics subjects focused on the network problem for the mathematically gifted students in the elementary school. *School Mathematics*, 7(4), 353-373.

한국과학창의재단 (2019). 좋은 수학과제 지도 가이드 개발 연구 최종보고서. Retrieved from https://www.kofac.re.kr/brd/board/457/L/menu/244?brdType=R&thisPage=1&bbIdx=24164&searchField=title&searchText=%EC%A2%8B%EC%9D%80%20%EC%88%98%ED%95%99%EA%B3%BC%EC%A0%9C%20%EC%A7%80%EB%8F%84

Korea Foundation for the Advancement of Science & Creativity (2019). Final report on The development of a guide to teaching mathematical tasks. Retrieved from https://www.kofac.re.kr/brd/board/457/L/menu/244?brdType=R&thisPage=1&bbIdx=24164&searchField=title&searchText=%EC%A2%8B%EC%9D%80%20%EC%88%98%ED%95%99%EA%B3%BC%EC%A0%9C%20%EC%A7%80%EB%8F%84

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Xavier, D. A., Rosary, M., & Arokiaraj, A. (2014). Topological properties of Sierpinski gasket rhombus graphs. *International Journal of Mathematics and Soft Computing, 4*(2), 95-104.

Exploration of the educational possibilities of one-stroke drawing problems of complex figure using programming

Cheong, Yong Wook
Gyeongsang National University
E-mail : ywcheong@gnu.ac.kr

This study propose the educational potential of an activity that solves the task of one-stroke drawing of complex figures using a drag-and-drop type educational programming language such as Scratch. The problem of determining whether a given shape is capable of one-stroke drawing is a separate problem from actually finding the path of one-stroke drawing and implementing it through programming. In particular, finding a path that allows one-stroke drawing of complex shapes with regularity and implementing it through programming requires problem-solving capabilities based on the convergence of various mathematical knowledge. Accordingly, in this study, problems related to one-stroke drawing concerning polygon-related shapes, tessellation-related shapes, and fractal shapes were presented, and the results of one-stroke drawing programming of the shapes were exemplified. In addition, the mathematical knowledge and computational thinking elements necessary for the solution of the illustrated problem were analyzed. This study is significant as a new example of the mathematics education that combines mathematics and information.

* 2020 Mathematics Subject Classification : 97K30, 97P40
* Key words : one-stroke drawing, programming, Scratch, tessellation, fractal