

초등학교 소프트웨어교육에서 학습자의 알고리즘 구성 패턴 연구

김정랑

광주교육대학교 컴퓨터교육과

요약

초등학교 6학년 학생을 대상으로 소프트웨어교육이 실시되고 있다. 본 연구에서는 초등학생의 알고리즘 구성 패턴에 대해 탐색하였다. 초등학교 6학년 학생을 대상으로 알고리즘을 구상하는 문항을 투입한 후 구조적 프로그래밍 기법에 기반한 MacCabe의 사이클로매틱 복잡도를 산출하여 학습자의 알고리즘 구성 패턴을 탐색하였다. 학생들은 문제 해결을 위해 주로 1~2가지의 선택구조를 사용하며, 이는 문제의 출발점, 도착점에 편중되는 경향이 있다. 또한 선택 구조 사용에 있어 소극적인 모습을 보인다. 알고리즘 구성에 있어서는 눈에 보이는 구체물과 자신의 배경지식에 의존하는 모습을 보인다. 따라서 초등학교 소프트웨어교육에서는 알고리즘 구성 패턴에 따라 학생들의 경험과 친숙한 문제 상황에서 알고리즘 구조를 복합적으로 사고할 수 있는 과제를 제시할 필요가 있으며, 구체적 조작물을 활용하여 지도하는 것이 유효할 것으로 보인다.

키워드 : 소프트웨어교육, 알고리즘, 컴퓨팅 사고력, 구조적 프로그래밍, 복잡도

A Study on Algorithm Composition Patterns of Learners in Elementary Software Education

Jeongrang Kim

Dept. of Computer Education, Gwangju National University of Education

Abstract

Software education is provided for 6th grade students. This study explored the algorithmic composition patterns of elementary school students. After investigating the algorithm for the 6th grade students, the algorithmic pattern of the learner was explored by calculating the cyclomatic complexity of MacCabe based on the structural programming technique. Students often use one or two choice structures to solve problems, which tend to be biased towards the starting and ending points of the problem. It is also passive in the use of selection structures. Algorithm composition depends on visible objects and one's own background. Therefore, in elementary school software education, it is necessary to present the task of thinking about the algorithm structure in the context of the algorithm and the students' experiences in accordance with the algorithm composition pattern.

Keywords : Software education, algorithms, computing thinking, structured programming, complexity

1. 연구의 필요성 및 목적

소프트웨어 교육은 언플러그드 활동, 교육용 프로그래밍 언어를 활용한 학습, 피지컬 컴퓨팅을 활용한 학습 등을 통해 학생들의 컴퓨팅 사고력을 향상시켜 컴퓨터가 문제를 해결하는 것과 같이 일상생활 속의 문제를 효율적이고 정확하게 해결하는 능력을 기르는 것을 목적으로 한다. 문제를 해결하는 절차를 알고리즘이라고 하며, 학생들은 소프트웨어 교육을 통해 효율적인 알고리즘을 구성하는 방법에 대해 학습한다. 이를 위해 2015 개정 교육과정에서는 초등학교 학생들이 알고리즘적 사고를 바탕으로 문제를 해결하는 과정에서 순차, 선택, 반복의 구조를 이해하도록 하고 있다. 순차, 선택, 반복의 구조를 바탕으로 문제를 해결하는 구조적 프로그래밍 기법은 문제를 해결할 때, 필요한 과정을 빠짐없이 구성할 수 있는 장점을 가지고 있어 실제 프로그램을 제작할 때 주로 사용된다[1][2].

초등학교 소프트웨어교육은 학생들이 구조적 프로그래밍 기법에 따라 문제를 해결하며 컴퓨팅 사고력을 신장할 수 있도록 구성되어 있고, 학생들은 이 과정에서 각자의 컴퓨팅 사고력의 수준에 따라 자신만의 알고리즘을 생성한다. 소프트웨어교육이 학습자의 컴퓨팅 사고력 향상에 긍정적인 영향을 주는 것은 사실이나, 개별 학습자의 산출물인 알고리즘 구성에 대한 분석과 관련한 연구는 다소 미흡한 것으로 보인다.

본 연구에서는 초등학교 학생들이 문제를 해결하며 생성해내는 알고리즘 구성의 패턴을 탐색하고자 한다. 이를 위해 두 개 문항을 선정하고 무작위로 선정된 초등학교 6학년 학생 80명을 대상으로 투입하였다. 수집한 결과의 분석을 위해 F.T.Baker 이론, M.H.Halstead이론 등 여러 알고리즘 성능 측정 이론 중 구조적 프로그래밍에 기반한 T.J. McCabe이론에 따라 복잡도를 분석하였다. 이를 바탕으로 초등학교 소프트웨어 교육에서 학습자의 알고리즘 구성의 패턴을 탐색하고 그 의미를 살펴보고자 한다.

2. 이론적 배경

2.1 구조적 프로그래밍

구조적 프로그래밍은 E. W. Dijkstra(Holland)가

1969년에 최초로 사용하였다. 구조적 프로그래밍 기법은 불필요한 GOTO 구문 사용을 제한하여, 복잡함을 줄이는 프로그램 작성 방법이다[3].

프로그램의 흐름을 간결하게 함으로서 수정이 용이하고, 프로그램의 질을 높일 수 있는 프로그래밍 방법으로 3가지 기본 형태의 흐름구조와 2가지 추가된 흐름구조에 의해 프로그램을 표현하는 방식이다. 어떠한 논리의 프로그램이라도 3가지 기본 형태인 순차(Sequence), 선택(If then else), 반복(While do)을 사용하면 모두 기술할 수 있다[4].


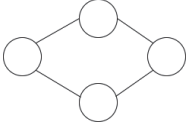

2.2 복잡도 측정 이론

복잡도는 소프트웨어 프로그램의 구조를 측정하는 체계로, 알고리즘을 개발할 때 어느 정도의 노력을 소요해야 하는지 등을 결정하는 근거를 제공한다.

복잡도를 측정하는 표준적인 방법에는 LOC(line of code), MacCabe's complexity metric, Halstead's metric, Tom DeMarco이론 등이 있다.

그중 구조적 프로그래밍 기법에 기반한 T.J.McCabe의 복잡도 측정 이론은 구조적 프로그래밍에서 규정한 3가지 기본 구조인 순차(SEQUENCE), 선택(IF THEN ELSE), 반복(WHILE DO) 구조를 노드의 수, 흐름선의 수, 연결성분을 바탕으로 하여 사이클로메틱(Cyclomatic) 복잡도로 나타내었다. 사이클로메틱 복잡도를 $V(G)$ 라고 할 때 $V(G) = e - n + 2p$ 와 같이 정의할 수 있으며, e 흐름선의 수 n 은 노드의 수, p 는 연결성분의 수이다. 기본 구조에 대한 복잡도는 <Table 1>와 같다[5].

<Table 1> Relationship between basic structure and complexity

Structured programming theory (Basic control structure)	Complexity measurement theory	
	rescue	$V(G)=e-n+2p$
SEQUENCE		$V(G)=1-2+2=1$
IF THEN ELSE		$V(G)=4-4+2=2$
WHILE DO		$V(G)=3-3+2=2$

보통 사이클로메틱 수가 10이상인 경우 알고리즘을 분할하도록 제안하며, 5~7의 경우 가장 효율적인 알고리즘이라고 한다. 또한 복잡도가 높아짐에 따라 프로그램의 오차율이 증가하며 이는 <Table 2>와 같다[6].

<Table 2> Relationship between complexity and error rate

Complexity	Error rate(%)
below 10	5
11~20	20
21~50	40
51 or more	60

이처럼 복잡도가 증가함에 따라 오차율도 함께 증가하며, 복잡도가 10이하가 될 수 있도록 알고리즘을 구성하는 것이 정확한 알고리즘이라고 할 수 있다.

MacCabe의 사이클로메틱 복잡도 측정은 구조적 프로그래밍에서 알고리즘의 효율성을 판단하는 데 중요한 의미를 갖는 것으로 보인다. 또 복잡도를 하나의 정량적 수치로 제시해준다는 점에서 이해하기 쉽다는 장점이 있다.

2.3 선행 연구

소프트웨어교육과 관련한 연구로 김정량(2017)은 학생들이 소프트웨어 교육을 통해 게임을 만들거나 소프트웨어 작품을 만들어보면서 학생 내적으로 무엇인가 학습하고 배워보고자 하는 동기, 무엇인가 창조하고 성취하고자 하는 동기 등을 추구한 결과로서, 학생들의 학습 동기 중 내재적 동기, 내사 조절, 동일시 조절에서 통계적으로 유의미한 차이가 있다고 하였으며, 소프트웨어 교육 후 자신감과 자기조절효능감에서 통계적으로 유의미한 차이가 나타났다고 하였다[7]. 노지예(2017)는 로봇 활용 소프트웨어 교육 프로그램의 효과성 검증에서 로봇 활용 소프트웨어 교육이 학생들의 컴퓨팅사고력, 창의성, 몰입을 향상시키는 것으로 나타났으며, 로봇 활용 소프트웨어 교육은 초등학생에게 필요한 수업이고, 초등학생들의 고등사고력 신장에 유용하다고 하였다[8]. 이수환(2018)은 SW교육용 보드게임을 활용한 언플러그드 활동을 적용한 뒤 컴퓨팅 사고력을 측정한 결과, SW교육용 보드게임이 학습자의 컴퓨팅 사고력 향상에 긍정적인 영향을 미친다고 하였다[9].

연구 방법과 관련하여 백수진(2009)은 카테시안 곱의 역 맥락에서 나타난 학생들의 분수 나눗셈 알고리즘 구성 활동을 교수 실험을 통해 분석하여 학생들의 문제 해결 전략 선택과 알고리즘화 과정을 구체화 하고, 이를 바탕으로 분수 나눗셈 알고리즘 지도에 주는 시사점을 도출하였다. [10] 유사한 연구로는 여러 교과에서 정현도(2009)의 서술형 평가를 통한 수학학습에서의 학습자 오류 유형 분석과 같은 형태[11]나 황선우(2016)의 초등학교학생들이 수행한 창의 산출물의 특징 분석과 같은 학습자의 산출물을 분석하는 형태의 연구[12]가 있었다.

국의 연구로 Maccabe(1979)는 구조적 프로그래밍에 기반하여 알고리즘의 복잡도를 측정하여 효율성을 분석하였다. Vassilios Dagdilelis, Maya Satratzemi, Georgios Evangelidis(2004)는 알고리즘과 프로그래밍이 중요한 지적 도구가 되며 기초교육에 포함되어야 한다고 주장하였으며[13], Donald Chinn(2005)는 학생들이 알고리즘에 대해 학습하며 동료평가를 할 때 학생들이 서로의 알고리즘에서 낮은 수준의 오류보다 높은 수준의 오류를 더 자주 식별하며 평가에 어떠한 경향성이 있다고 하였다[14]. 또 Jeffrey Bonar & Elliot Soloway(2009)는 프로그래밍 과정 초기에서 초보 학습자가 빈번하게 생성하는 버그를 설명하는 프로세스 모델을 제시하였다[15]. 이는 학습자의 알고리즘적 사고 과정에서 패턴이 있음을 시사한다.

선행연구를 통해 얻은 시사점으로는 최근의 연구는 주로 소프트웨어교육의 방법적인 측면에서 학습자에게 적용하였을 때 나타나는 효과성에 중점을 두고 있으며, 다른 교과의 연구와는 달리 개별 학습자의 산출물인 알고리즘 구성에 대한 분석과 관련한 연구는 다소 미흡한 것으로 보인다. 또한 문제 상황에서 학습자가 생성해 낸 알고리즘을 분석한 연구도 한정적이다. 따라서 본 연구에서는 학습자가 문제해결 과정에서 생성한 알고리즘을 바탕으로 알고리즘 구성의 패턴을 탐색하고자 한다.

3. 연구 방법 및 절차

3.1 연구 대상 및 방법

소프트웨어교육 관련 정규교육과정을 이수한 초등학교 6학년 학생 80명을 무작위로 선정하였다.

위 학생을 대상으로 문제 상황에 따라 알고리즘을 생성하는 유형의 문항 두 개를 투입한다. 학생들은 두 개 문항에 대한 알고리즘을 자연어로 응답한다. 응답한 알고리즘의 복잡도를 산출하고 이를 바탕으로 초등학교 학생들의 알고리즘 구성 패턴을 탐색한다.

3.2 문항 구성

2015개정 교육과정을 토대로 개발된 실과 6종 교과서의 절차적 문제해결 단원에 제시된 12개의 문항 중 중복된 것을 제거하고, 학습자의 역량에 따라 다양한 응답을 할 수 있는 발산적인 문항 2개를 선정하였다. 본 연구를 위해 컴퓨터 교육학 박사 1인과 석사 3인 현장 교사 2인의 자문을 얻어 선정한 문항을 타당도에 영향을 주지 않는 범주에서 수정하였다. 본 연구에서 사용한 문항의 구성은 <Table 3>과 같다.

<Table 3> Item composition

Number	Item Content
1	You live on the seventh floor of an apartment with the highest floor on the 15th floor. There is one elevator and the two houses are facing each other around the elevator. Your house is on the left side when you stand back from the elevator. It is 10 steps from the house's entrance to the elevator. All floors of this apartment are the same form. You have a robot that moves only as commanded. From now on, you are going to program your robot so that you can climb from the first floor to the seventh floor without human help and stand in front of your home. In what order and in what order do you enter the robot correctly in front of the house? Use the procedural troubleshooting below to create an algorithm that will solve the problem by listing the commands on the back.
2	Think of your back to school every day. I want to explain to my friends how to go to school. How do we represent the process from the front door to the front door of the classroom, using procedural problem solving?

문항 1은 M출판사의 교과서에 제시된 절차적 사고로 일상생활 문제해결하기 문항을 토대로 해당 문항의 주제를 엘리베이터를 타고 1층에서 7층까지 이동하는 것으로 특정한 것이다. 엘리베이터를 이용하는 주제는 엘리베이터가 학생들에게 친숙한 장치이며, 장치의 특성상 상하 운동을 하는 점, 조작이 명료한 점 등으로 알고리즘으로 표현하기에 수월하여 선택하였다. 또한 문항에 알고리즘을 설계하기 위한 구체적인 자료를 함께 제시

함으로써 문항을 해결하며 구성하는 알고리즘에 학생의 자료 수집 및 분석 역량도 함께 확인할 수 있도록 하였다. 문항 2는 6종 교과서에서 공통적으로 나타나는 계획 세우기 형태의 문항을 본 연구에 알맞게 수정한 것으로, 학생들이 충분히 이해하고 있는 자신의 등갓길을 알고리즘으로 표현하는 것이다. 문항 1과 비교했을 때 함께 제공되는 자료가 상대적으로 적어, 학생들이 응답할 수 있는 범위가 넓다. 이처럼 다른 특성을 가진 두 개 문항을 활용하여 알고리즘 구성 패턴을 탐색함으로써 알고리즘 구성의 특징을 분명하게 살펴볼 수 있다.

3.3 분석 도구 및 방법

알고리즘을 분석하기 위한 도구로는 MacCabe의 사이클로매틱 복잡도 측정을 사용하였다. 사이클로매틱 복잡도는 $V(G) = e - n + 2p$ 로 구할 수 있으며, 간단하게 구하는 방법으로는 $V(G) = \text{선택구조의 수} + 1$ 이 있다. 본 연구에서는 후자의 방법으로 복잡도를 산출하였다.

학생들의 알고리즘은 자연어로 이루어져 있어서 기본 구조인 순차, 선택, 반복이 명확하게 구별되지 않는 경우가 있다. 따라서 순차, 선택, 반복 구조를 판별하기 위한 키워드를 선정하여 체크하였다. 이 때 판별하기 위한 키워드는 학생들에게 친숙한 엔트리 블록을 기준으로 하였다. 순차구조의 명령의 경우 판별을 위한 키워드는 <Table 4>와 같다.

<Table 4> Natural Language Algorithm Structure Discrimination Keyword

Basic structure	Key keywords
SEQUENCE	Entry instruction block unit (E.g. move 1 space in the direction of rotation, rotate 90 degrees, etc.)
IF THEN ELSE	-If -If ~ or
WHILE DO	-Keep repeating -Repeat until -Repeat ~ times

순차 구조는 학생의 응답이 모호한 경우, 엔트리 명령 블록을 기준으로 판단하였다. 엔트리 명령 블록을 기본 명령 단위로 판단하여, 블록의 수만큼 순차 구조를 가지는 것으로 판단하였다.

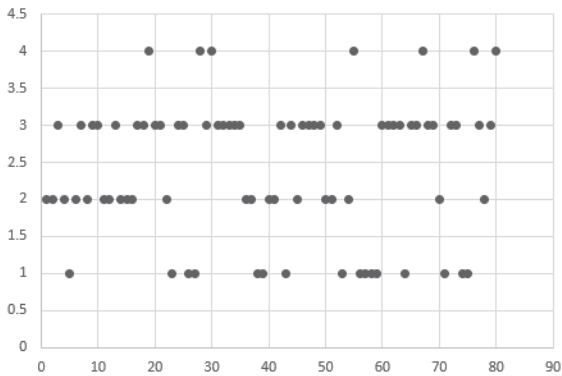
선택 구조는 만약 ~ 라면 또는 만약 ~ 라면, 아니면 블록을 기준으로 판단하였고 문맥상 이와 동일하거나 유사한 의미를 갖는 낱말 또는 문장을 사용한 경우 선택 구조를 가지는 것으로 판단하였다.

반복 구조는 계속 반복하기, ~할 때까지 반복하기, ~번 반복하기를 키워드로 판단하였고 문맥상 이와 동일하거나 유사한 의미를 갖는 낱말 또는 문장을 사용한 경우 반복 구조를 가지는 것으로 판단하였다.

4. 연구 결과

4.1 문항 1 알고리즘 분석

문항 1에 대해 초등학교 6학년 학생이 응답한 알고리즘을 사이클로매틱 복잡도 측정을 통해 분석한 결과는 (Fig. 1)과 같다.



(Fig. 1) Result of Algorithm Complexity Analysis (N=80)

문항 1의 알고리즘 복잡도 분석 결과 최소값은 1, 최대값은 4, 평균 복잡도는 2.425로 나타났으며, 최빈값은 3이다. 복잡도별 응답 수는 복잡도 1은 16, 복잡도 2는 21, 복잡도 3은 36, 복잡도 4는 7이다.

문항 1의 알고리즘 구성 패턴은 복잡도에 따라 분류할 수 있으며, 복잡도는 선택구조의 사용 정도에 따라 영향을 받으므로 선택구조의 사용 빈도와 용도에 따라 탐색할 수 있다.

복잡도 1을 대표하는 표본 알고리즘은 <Table 5>과 같다.

<Table 5> Complexity 1 algorithm sample

Complexity 1 (No. 39)	Analysis content	
	structure	amount
<Start>		
1. Get in the elevator.	SEQUENCE	5
2. Press the 7th floor button.		
3. Get off the elevator.	IF THEN ELSE	0
4. Turn left.		
5. Go forward 10 steps.	WHILE DO	0
<Arrival>		

위와 같은 형태는 순차구조로만 이루어진 알고리즘으로, 프로그램의 크기는 복잡도에 영향을 주지 않는 사이클로매틱 분석의 특성이 반영된 것으로 보인다. 이와 같은 응답은 선택구조가 없는 알고리즘으로 알고리즘의 완성도가 다소 떨어지는 것으로 판단된다. 복잡도 1의 알고리즘은 대부분 사람의 움직임만을 추상적으로 표현하고 있으며, <Table 6>의 예와 같이, 엘리베이터가 동작하는 절차를 ‘엘리베이터에 탄다.’라는 명령과 같이 생략하여 표현하는 경향이 있다.

복잡도 2를 대표하는 표본 알고리즘은 <Table 6>과 같다.

<Table 6> Complexity 2 algorithm sample

Complexity 2 (No. 70)	Analysis content	
	structure	amount
<Start>		
1. Walk forward	SEQUENCE	5
2. Press the elevator button.		
3. Wait for the elevator.	IF THEN ELSE	1
4. Take the elevator.		
5. When you arrive on the 7 th floor, get off.	WHILE DO	1
<Arrival>		

위의 알고리즘은 순차구조 1개로 이루어진 알고리즘으로 학생들은 1층에서 7층까지 이동하는 과정에서 1가지 선택구조를 사용하였으며 이는 두 가지 유형으로 분류할 수 있다. 첫 번째 유형은 1층에서 또는 목적지에 도착했을 때 문이 열려있다고 가정하는 경우이다. 이 같은 경우 탑승할 때, 또는 내릴 때만 ‘문이 열리면 ~’ 이라는 조건

문을 사용하였다. 두 번째 유형은 <Table 6>과 같이, 복잡도 1인 경우와 유사한 유형으로 복잡도 2의 알고리즘 대부분은 ‘엘리베이터가 도착하면~’이라는 조건문을 사용하였다. 이러한 응답을 한 학습자는 문제를 해결하는 과정에서 판단이 필요하다는 사실을 인지하고 있으나, 그 범위가 다소 넓다는 한계점을 가지고 있다.

복잡도 3을 대표하는 표본 알고리즘은 <Table 7>과 같다.

<Table 7> Complexity 3 algorithm sample

Complexity3 (No. 72)	Analysis content structure	amount
<Start>		
1. Look where the elevator is.		
2. Move to the elevator.		
3. Check where the elevator buttons are located.	SEQUENCE	16
4. Press the elevator button.	NCE	
5. Wait for the elevator to come down (until you hear the elevator arrival).		
6. When the elevator door is open, enter the elevator.		
7. See where the elevator floor number button is located.	IF	2
8. Find the 7th floor button in the Floors button.	THEN	
9. Press the 7th floor button.	ELSE	
10. Wait for the elevator to reach the seventh floor (until you hear the elevator arrival).		
12. When the elevator door opens, go out of the elevator.		
13. Turn 90° counterclockwise around the elevator	WHILE	2
14. Look where the door is.	DO	
15. Move to the door.		
16. Stand at the door.		
<Arrival>		

위의 알고리즘은 2가지 선택구조를 사용한 알고리즘으로 문항 1의 알고리즘 최빈값에 해당한다. 복잡도 3에 해당하는 알고리즘은 출발할 때 엘리베이터의 문이 열리는 것과 도착하였을 때 엘리베이터의 문이 열리는 것의 2가지 상황에 대한 선택구조를 사용한 것으로 학생들은 엘리베이터를 이용할 때의 핵심 상황으로 문의 작동 여부를 생각한다는 사실을 알 수 있다. 이는 학생들이 문이 열리고 닫히는 것을 통해 엘리베이터의 작동 여부를 눈으로 쉽게 확인할 수 있으며, 이는 구체적 조작기에 해당하는 초등학교의 인지발달단계에 따라 구체적으로 작동하는 것이 보이는 표면적인 부분인 움직임, 생김새 등에 집중하는 경향이 있는 것으로 보인다.

복잡도 4를 대표하는 표본 알고리즘은 <Table 8>과 같다.

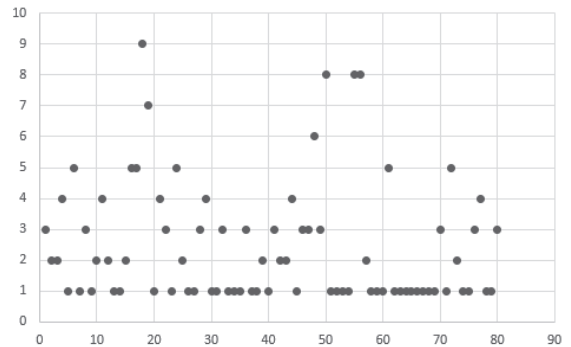
<Table 8> Complexity 4 algorithm sample

Complexity 4 (No. 28)	Analysis content structure	amount
<Start>		
1. Walk to the elevator door.	SEQUENCE	7
2. Press the elevator up button.	NCE	
3. When the elevator door opens, go inside.	IF	3
4. If anything is missing, give it to me.	THEN	
5. Press the 7th floor button	ELSE	
6. The elevator goes up to the 7th floor and the door opens.	WHILE	0
7. Walk towards the left door.	DO	
<Arrival>		

위의 알고리즘은 복잡도 3의 엘리베이터 문이 여닫히는 조건에 엘리베이터의 작동과는 무관한 엘리베이터 내, 외부에서 일어나는 상황 1가지가 추가된 형태로 장애물이 있는 상황이나, 소지한 물건이 떨어졌을 때 대응하는 상황 등을 고려한 것으로 보인다. 이 같은 내용은 엘리베이터를 이용하는 상황에서 일반화하기 다소 어려운 것으로 학생의 개인적인 배경지식에서 비롯된 조건으로 보인다.

4.2 문항 2 알고리즘 분석

문항 2에 대해 초등학교 6학년 학생이 응답한 알고리즘을 사이클로매틱 복잡도 측정을 통해 분석한 결과는 (Fig. 2)와 같다.



(Fig 2) Result of Algorithm Complexity Analysis of Question 2 (N=80)

문항 2의 알고리즘 복잡도 분석 결과 최소값은 1, 최대값은 9, 평균 복잡도는 2.48로 나타났으며, 최빈값은 1이다. 복잡도별 응답 수는 복잡도 1은 38, 복잡도 2는 11, 복잡도 3은 13, 복잡도 4는 6, 복잡도 5는 6, 복잡도 6은 1, 복잡도 7은 1, 복잡도 8은 3, 복잡도 9는 1이다.

문항 2의 알고리즘 구성 패턴을 복잡도에 따라 탐색하면, 최빈값이 1로 38명의 학생이 선택구조 없이 알고리즘을 구성한 것을 볼 수 있는데 이는 문항 2의 자신의 등굣길을 알고리즘으로 구성하는 것을 길 찾기 문제와 유사한 문제로 접근하여 길을 찾아가는 과정을 <Table 9>와 같이 움직임을 중심으로 나열한 것으로 보인다.

<Table 9> Algorithm of Complexity 1 of Item 2

Complexity 1 (No. 28)	Analysis content structure amount
<Start>	
1.Go forward 3 steps	
2. Turn 90 ° to the right	
3. Go forward 10 steps	
4.90 ° turning left	SEQUE 24
5.Forward to the entrance of the apartment	NCE
6. Rotate 90 ° to the right	
7. 20 steps forward	
8. Turn 90 ° to the right	
9. crosswalk 3. Go to	
10. cross the pedestrian crossing	
11. turn 90 ° to the left	
12. Go forward two steps	
13. Turn 90 ° to the right	IF 0
14. Forward crosswalks 5. Go to	THEN
15. Turning 90 ° to the left	ELSE
16. Cross pedestrian crossing	
17. Go forward 9 steps	
18. Turn 90 ° to the right	
19. Go forward 5 steps	
20. Turning 90 ° to the left	
21. Go to the end of the playground	WHILE 3
22. Go forward 5 steps	DO
23. Turn 90 ° to the right	
24. Go up the stairs	
<End>	

또한 문항 2의 경우 복잡도의 스펙트럼이 1부터 9까지로 다양하게 나타나며, 이는 학교마다 학생마다 등교하는 길이 다양한 것에서 비롯되는 것으로 보인다. 또 복잡도가 3을 넘어가면 급격히 빈도가 감소하는 현상을 볼 수 있다. 이는 복잡도 3의 경우 순차구조를 2개 사용한 알고리즘으로 대부분의 학생들이 출발, 도착의 조건

을 판별할 때 선택구조를 사용하였으며, 이는 문항 1과 같이 출발점, 도착점을 판별할 때 주로 선택구조를 사용하는 것으로 보이며, 그 외의 경우는 학생들이 이동 경로나 동작을 변경하는 부분에서 부분적으로 사용하거나 문항 1과 같이 자신의 배경지식에서 비롯한 상황에 필요한 선택 구조를 사용하는 경향이 보인다.

4.3 알고리즘 구성 패턴 탐색

문항 1과 문항 2에 대한 분석을 토대로 초등학교생의 알고리즘 구성 패턴을 탐색한 결과는 다음과 같다.

알고리즘 구조 측면에서 학생들이 문제해결을 위해 대체로 1~2개의 선택구조를 활용하여 알고리즘을 구성하는 것으로 보인다. 사용된 1~2개의 선택구조는 주로 시작 조건과 도착 조건을 판별할 때 사용되었다. 이처럼 학생들은 알고리즘을 구성할 때 시작점, 도착점을 중심으로 선택구조를 사용하는 경향을 보인다.

알고리즘의 완성도 측면은 MacCabe의 사이클로메틱 복잡도의 경우 5~7에 해당하는 복잡도를 가진 알고리즘이 완성도가 높은 것으로 판정할 수 있다. 이에 따라 평균 2.48의 복잡도를 가진 알고리즘을 생성하는 초등학교생의 경우 다소 완성도가 낮은 알고리즘을 생성하는 것으로 판단된다. 이는 학생들이 알고리즘 구성 시 선택구조 사용에 있어 다소 소극적이며, 문제 상황에서 선택구조를 사용하기 위한 문제 분해가 미숙한 것으로 보인다.

문항 2의 복잡도 5~7에 해당하는 알고리즘의 경우 선택구조 사용에 있어 문제 해결과 무관한 문제 상황과 유사한 자신의 배경지식에서 비롯한 자기 자신의 행동 또는 습관 평소 관심사와 관련된 것들로 이루어져있어 그 내용과 구성으로 판단하여 볼 때, 선택구조를 문제 해결의 과정과 다소 동떨어진 형태로 사용하는 것으로 판단된다. 따라서 복잡도 5~7에 해당되는 알고리즘이더라도 완성도가 높다고 판단하기는 미흡하다. 이와 같은 현상은 인지발달이론에 근거하여 장명덕(2014)은 구체적 조작기에 해당하는 학생의 주요 특징 중 하나로 사교가 여전히 개인적인 경험과 밀접하게 관련되어 있어서 대상과 상황이 자신에게 친숙한 경우에만 가능하다고 하였으며[16], 은은숙(2011)은 학습자는 우선 동화하고 다음에 조절하며, 타자로부터 정보를 그대로 수용하는 것이 아니라, 자신이 이미 가지고 있는 구조에 동

화될 수 있는 한에서 문제의 정보를 추상화 한다고 하였다.[17] 또 정미자(2011)의 연구와 같은 여러 교과 학생의 오류 유형에 대한 연구들에서도 상당수의 오류는 학생들이 자신의 배경지식에 의존하여 일으키는 것으로 나타난다.[18][19][20]

이로 미루어 볼 때, 복잡도 5~7에 해당하는 응답을 한 학생들은 주어진 문제 상황을 해결하기 위해 자신의 배경지식을 바탕으로 문제를 이해하고 분석하는 과정에서 문제와 다소 동떨어진 선택구조를 사용하는 오류를 일으켰을 가능성이 있다. 5 미만의 복잡도를 가지는 알고리즘은 모두 복잡도 5~7에 해당하는 알고리즘에 포함되는 내용을 가지고 있으므로, 학습자는 자신의 배경지식에 의존하여 알고리즘을 구성한다고 해석할 수 있다.

알고리즘의 내용 측면에서는 학생들은 눈에 보이는 구체물의 움직임이나 모양 등을 중심으로 선택구조를 사용하거나 순차구조를 생성하여 알고리즘을 구성하며, 보이지 않는 부분에 대한 고려가 미흡한 경향을 보인다. 이는 초등학생 인지발달수준에 의한 것으로, 구체적 조작기에 해당하는 초등학생의 특성으로 보인다.

5. 결론

2019학년도부터 초등학교 6학년 학생을 대상으로 소프트웨어교육을 실시하고 있다. 이와 더불어 소프트웨어교육과 관련한 연구가 활발하게 이뤄지고 있다. 그러나 학습자들의 산출물인 알고리즘에 대한 연구는 다소 미흡한 것으로 보여 본 연구에서는 초등학교 소프트웨어교육에서 학습자의 알고리즘 구성 패턴을 탐색하였다.

초등학교 학습자의 경우 대체로 2~3개의 선택구조를 활용하여 알고리즘을 구성하는 것으로 보이며, 이는 출발 동작과 도착 동작을 판별할 때 주로 사용하는 경향이 있으며, 평균 복잡도는 2.4 내외로 다소 완성도가 낮은 알고리즘을 생성하는 것으로 판단된다. 또한 알고리즘을 구성할 때 자기 자신의 행동 또는 습관 평소 관심사와 관련된 것이 드러나며, 이는 알고리즘을 생성하는 과정에서 자신의 배경지식에 의존하는 것으로 판단된다. 학생들은 눈에 보이는 물체의 움직임이나 모양 등을 중심으로 순차구조를 사용하거나, 행동을 명령하며 알고리즘을 구성하는 경향이 있으며, 눈으로 볼 수 없는 부분

에 대한 고려가 미흡한 경향을 보인다. 이는 초등학생 인지발달단계에서 기인한 것으로, 구체적 조작기에 해당하는 6학년 학생들의 특성으로 보인다.

연구결과를 바탕으로 한 초등학교 소프트웨어교육에 대한 시사점은 다음과 같다.

첫째, 알고리즘 구성과 관련하여 선택구조의 사용에 대한 지도가 보완될 필요가 있다는 점이다. 선택구조는 알고리즘의 완성도를 높여주며 다양한 상황에 대응할 수 있는 형태를 가지고 있어 알고리즘 구성에 필수적이거나, 학생들이 선택구조 사용에 다소 미흡한 모습을 보이고 있다. 따라서 알고리즘 지도 시 대표적으로 사용하는 길 찾기 문제를 현재의 단순 순차구조학습 형태와 함께 선택, 반복구조를 사용하여 해결할 수 있는 복합적인 문제 상황도 함께 제시하여 학생들의 사고를 심화시킬 수 있는 기회를 제공하는 것이 필요하다.

둘째, 학생들은 알고리즘을 구성할 때 자신의 배경지식에 의존하는 경향이 있으므로, 학생들의 경험에 친숙한 주제를 사용함과 동시에, 문제 해결을 위해 반드시 필요한 동작과 불필요한 동작을 구별하는 추상화에 대한 지도가 강화될 필요가 있다.

셋째, 눈에 보이는 구체물이나 움직임을 중심으로 알고리즘을 구성하는 경향이 있으므로, 구체적조작물 등을 활용한 교육방법이 좀 더 유효할 것으로 보인다.

앞으로의 소프트웨어교육은 학생들의 알고리즘 구성 패턴에 기반 하여 내실 있는 알고리즘 교육이 강화될 필요가 있다.

참고문헌

- [1] Ryu, Cheong-san. et al.(2019). *Practical Arts Textbook for elementary School grades*. Seoul: Kumsung.
- [2] Ahn, Bo hee(1986). *A Study on the program complexity measurement based on the structured programming technology*. master's thesis, dd University.
- [3] Oh, Se chul(1989). *A Study on Program Improvement by Measuring Complexity of Control Structure*. master's thesis, Chosun University.
- [4] Ahn, Bo hee.(1986). *A Study on the program complex-*

- exity measurement based on the structured programming technology. master's thesis, dd University.
- [5] T, J. McCabe.(1976). A Complexity Measure. *IEEE Transactions on Software Engineering, Vol. 1*, SE-2 pp. 308-320.
- [6] RogueWave. McCabe Cyclomatic Complexity. <https://docs.roguewave.com/en/klocwork/current/mccabecyclomaticcomplexity>.
- [7] Kim, J. R.(2019) Effects of Software Education Program for the Education Welfare Priority Support Students on Learning Motivation, Self-efficacy and Goal Orientation, *Journal of The Korea Society of Computer and Information, Vol. 24* No. 1.
- [8] Noh, Ji yae(2017) *The Effects of SW Education Using Robot*, Doctor's Dissertation, Ewha Womans University.
- [9] Lee, Soo hwan(2018) *To improve computing thinking development of board games(SHAPES) for software education*, Master's thesis, Gwang-Ju National University of Education.
- [10] Beak, Soo jin(2009). *Analysis on Students' Construction of the Algorithm for Division by Fractions in the Context of the Inverse of a Cartesian Product*, Master's thesis, Gyeong-in National University of Education.
- [11] Jung, Hyung do(2009). *In mathematics learning through narrative evaluation Error type analysis*, Master's thesis, Bu-san National University of Education.
- [12] Hwang, Sun Woo(2016). *Performed by elementary science gifted students Characterization of Creative Output*, Master's thesis, Gyeong-in National University of Education.
- [13] Vassilios DagdilelisMaya SatratzemiGeorgios Evangelidis(2004). Introducing Secondary Education Students to Algorithms and Programming. *Education and Information Technologies. June 2004, Volume 9*, Issue 2, pp 159 - 173.
- [14] Donald Chinn(2005). Peer assessment in the algorithms course. *ACM SIGCSE Bulletin Volume 37*, Issue 3. September 2005.
- [15] Jeffrey Bonar & Elliot Soloway(2009). Preprogramming Knowledge: A Major Source of Misconceptions in Novice Programmers. *Human - Computer Interaction. Volume 1*, 1985 - Issue 2: Novice Programming, pp 133-161.
- [16] Jang Myund duck(2014). *Theory and Practice of Elementary Science Education*. Academy press.
- [17] Eun, Eun suk(2011). *The Logical Justification on Piaget's Developmental Theory and Its Educational Implication*, Doctor's thesis, Gyeong-buk National University.
- [18] Jung Mi ja(2011). *On domain type errors in elementary mathematics Comprehensive Consideration*, Master's thesis, Bu-san National University of Education.
- [19] Jung, Min Jum(2018). *Math Problems for Elementary 4th Grade Students Error analysis during troubleshooting*. Master's thesis, Bu-gyung University.
- [20] Lee, Myung Ju(2014). *Study of be verb errors in elementary school students: Focusing on free writing analysis*. Master's thesis, Yon-sei University.

저자소개



김 경 량

1997 전남대학교 (이학박사)

1999 San Jose State University
 객원교수

1985~현재 광주교육대학교 컴퓨터교육과 교수

관심분야 : 컴퓨터교육, 디지털교과서, 이러닝, 교육정보화, 소프트웨어교육, 인공지능교육

e-mail : jrkim@gnue.ac.kr