

# 초등학생을 위한 알고리즘 교수학습방법과 평가

김철

광주교육대학교 컴퓨터교육과

## 요 약

본 연구에서는 초등학교의 알고리즘 교육에 필요한 교육 내용과 교육 방법, 평가 유형을 제안하였다. 첫째, 교육 내용으로는 한국정보교육학회에서 제안한 정보교육과정을 보완하여 알고리즘의 표현, 알고리즘의 이해, 알고리즘과 순서도, 알고리즘의 구조, 알고리즘의 결과, 알고리즘의 수정, 알고리즘의 개선으로 구분하였다. 둘째, 교육 방법으로는 생활 속 코딩, 언플러그 활동, 블록 프로그래밍, 체감형 프로그래밍을 제안하였다. 셋째, Code.org에서 제공하는 'Hour of Code'의 모든 미션을 분석한 후 평가 유형을 알고리즘 선택하기, 알고리즘 채우기, 알고리즘 수정하기, 알고리즘 예측하기로 구성하였다.

키워드 : 알고리즘 교육, 교육 내용, 교육 방법, 교육 평가

## A Study on Algorithm Teaching and Learning Methods and Assessment for Elementary School Students

Chul Kim

Dept. of Computer Education, Kwangju National University of Education

## ABSTRACT

In this study, we suggested the contents, teaching and learning method, and assessment types of algorithm education in elementary schools. First, we suggested the algorithm education contents; the expression, understanding, flowcharts, structure, results, correction, and improvement of algorithm. Second, we showed the algorithm teaching and learning methods; algorithm in our daily life, the unplugged activity, block programming and tangible programming. Finally, we analyzed all missions of 'Hour of Code' in Code.org, and suggested the algorithm assessment 4 types, which includes selecting, filling, correcting, predicting of appropriate algorithm.

Keywords : Algorithm Education, Content, Teaching and Learning Method, Assessment

## 1. 서론

소프트웨어(이하 SW) 중심 사회가 도래하고 있다. SW가 혁신과 성장, 가치 창출의 중심이 되고, 개인, 기업, 국가의 경쟁력을 좌우하는 사회를 말한다[8]. SW 산업은 이미 자동차 산업을 압도하고 있지만, 우리의 SW에 대한 원천 기술은 매우 부족하다. 2014년 2분기 기준으로 우리나라의 S기업의 매출은 52조 원으로, 경쟁 기업인 미국의 A사 매출액 38조보다 1.4배가량 많았지만, 순이익은 7조 2천억 원으로 A사의 7조 9천억 원보다 오히려 적었다[1]. 우리 기업이 더 많은 매출액을 올렸지만, 그 수익이 적은 이유는 여러 가지 있겠지만 그중 하나는 SW에 대한 원천 기술이 부족하기 때문이다. S사는 구글이 개발한 모바일 OS 안드로이드로 구동되는 스마트폰과 태블릿을 판매하는데, 안드로이드에 MS 특허(웹 브라우저에 창을 여러 개 띄우는 기술 등)를 사용하는 대가로 연간 10억 달러 이상의 특허료를 지급하고 있다[11].

옥스퍼드 연구팀에 따르면 향후 주요 직업이 로봇과 인공지능으로 대체된다고 한다. 그중에서 제일 먼저 대체되는 직업으로는 텔레마케터이며, 파쇄기계 운전기사나 굴착기 운전기사, 약제사 등이 대체될 직업으로 선정하였다[2]. 또한, 한국직업능력개발원에 따르면 일자리가 가장 빠르게 증가하는 직업 중 하나가 컴퓨터시스템 설계 전문가가 뽑혔다[3].

소프트웨어 중심 사회에서 살아 갈 학생들이 갖추어야 할 21세기 핵심 역량에는 4C가 있다[13]. 즉, 의사소통(Communication), 협력(Collaboration), 비판적 사고(Critical Thinking), 창의성(Creativity)이 필요하다. 이러한 핵심 역량은 전통적인 학습 기능인 읽기(reading), 쓰기(writing), 셈하기(arithmetic) 이외에도 프로그래밍하기(programming) 능력이 요구된다[6]. 이러한 프로그래밍 능력은 알고리즘과 코딩 능력을 포함하며, 학생들의 창의적인 사고력을 향상시키고, 문제를 해결하는 능력을 기를 수 있다[7].

2015년에 개정된 초중등학교 교육과정 총론에 따르면, 초등학교 5~6학년에서 배우는 실과 교과에 컴퓨터 과학적 사고를 키우기 위한 기초적인 소프트웨어 교육을 포함하고 있다[8]. 컴퓨터 과학적 사고는 문제를 분석하고 분해하는 과정을 통해 문제를 이해하고, 그것을

모델화하고 추상화하면서 알고리즘을 작성하며, 작성된 알고리즘을 프로그래밍 과정을 통해 자동화하여 문제에 적용하고, 나아가 오류를 수정하거나 더 나은 생각으로 개선하면서 최선의 생각을 창출하는 것을 의미한다[7].

이러한 컴퓨터 과학적 사고의 핵심은 문제를 이해하고, 그것을 해결하기 위해 프로그램으로 구현하기 전에 해결 방법과 절차를 구안하는 알고리즘을 설계하는 것이 중요하다. 따라서 본 연구에서는 초등학생들에게 가르치고자 하는 알고리즘의 내용을 간략히 제시하고, 국내외 소프트웨어 교육 플랫폼을 분석하여, 초등학생들에게 알고리즘을 가르치기 위한 교수학습 방법과 평가 방안을 제시하였다.

## 2. 알고리즘 교육의 내용과 방법

### 2.1 알고리즘 교육 내용

한국정보교육학회에서는 국내외 소프트웨어 교육 과정과 사례를 분석한 후 초등학생들에게 가르쳐야 할 알고리즘의 내용을 다음과 같이 알고리즘의 이해, 알고리즘의 표현, 알고리즘의 분석 등 3가지 영역을 제시하였다[7][4].

첫째, 알고리즘 이해 영역에서는 알고리즘의 뜻을 이해하고, 간단한 문제 상황을 일이 일어난 순서에 따라 말이나 글로 표현할 수 있으며, 그 순서에 대한 문제를 해결할 수 있다.

둘째, 알고리즘 표현 영역에서는 주어진 조건에 따라 특정 구간을 반복하는 알고리즘을 작성할 수 있고, 문제 해결 과정을 순서도로 표현하거나 일부만을 나누어 알고리즘으로 표현할 수 있다.

셋째, 알고리즘 분석 영역에서는 주어진 알고리즘의 오류를 찾아 수정할 수 있으며, 문제를 효과적으로 해결할 수 있는 알고리즘을 찾을 수 있다.

그러나 제시된 알고리즘에 대한 내용이 학생들의 능력에 따른 알고리즘의 학습 요소보다는 학년 구분에 따른 알고리즘의 성취 기준을 제시하였다. 따라서 학년에 따른 알고리즘 수업 시간이 확보되지 않아, 창의적 체험 활동 시간이나 방과후학교 시간에 무학년제로 운영해야 하는 초등학교 실정에 적합하지 않다. 따라서 본 연구에

서는 유치원생부터 초등학교 6학년까지 학생들의 능력에 따라 자유롭게 학습할 수 있도록 알고리즘 교육의 학습 요소를 7단계로 구분하고, 다음과 같이 각 단계별로 2개를 제시하여 총 14개의 학습 요소를 제시하였다.

#### 1단계 : 알고리즘의 표현

- (1)일의 순서에 따라 그림이나 문장을 나열할 수 있다.
- (2)일상생활의 활동을 순서에 따라 말이나 글로 표현할 수 있다.

#### 2단계 : 알고리즘의 이해

- (3)문제 해결 과정을 순서에 따라 표현할 수 있다.
- (4)알고리즘은 일이 일어난 순서임을 이해할 수 있다.

#### 3단계 : 알고리즘과 순서도

- (5)문제 해결 과정을 그림이나 기호로 표현할 수 있다.
- (6)문제 해결 과정을 순서도로 표현할 수 있다.

#### 4단계 : 알고리즘의 구조

- (7)반복 구조를 순서도로 표현할 수 있다.
- (8)선택 구조를 순서도로 표현할 수 있다.

#### 5단계 : 알고리즘의 결과

- (9)컴퓨터의 기능과 알고리즘의 관계를 이해할 수 있다.
- (10)알고리즘의 작동 결과를 예측할 수 있다.

#### 6단계 : 알고리즘의 수정

- (11)여러 알고리즘의 장단점을 비교할 수 있다.
- (12)알고리즘의 오류를 찾아 수정할 수 있다.

#### 7단계 : 알고리즘의 개선

- (13)가장 효과적인 알고리즘을 찾을 수 있다.
- (14)보다 효과적인 알고리즘으로 수정할 수 있다.

## 2.2 알고리즘 교육 방법

알고리즘을 어린 학생들에게 가르치기 위해서는 생활 속에서 일어나는 일에서 규칙을 찾아 알고리즘으로 표현하는 활동에서 시작해야 한다. 즉, 생활 속 알고리즘을 찾아 그것을 언플러그드(unplugged) 활동을 통해

체험하고, 컴퓨터를 활용한 블록프로그래밍(block programming)이나 체감형(tangible programming) 프로그래밍을 통해 알고리즘을 프로그램으로 구현하고, 문제를 해결하는 과정이 필요하다.

### 2.2.1 생활 속 알고리즘

생활 속 알고리즘 교육을 위해서는 학생들의 인지적 수준에 적합한 내용이어야 하며, 규칙을 찾고, 그것을 명료화하며, 실제 문제에 적용할 수 있는 것이어야 한다. 초등학생 교육에 적합한 알고리즘을 선정하는 기준은 다음과 같다.

첫째, 실생활과 관련성이 있는 알고리즘이어야 한다. 학생들의 삶과 직접적으로 관련된 문제를 다룰수록 알고리즘에 대한 개념을 쉽게 이해한다. 예를 들면, 아침에 일어나서 등교하기까지의 과정, 횡단보도를 건널 때의 과정, 전화 걸 때의 과정, 엘리베이터를 탈 때의 과정, 라면을 끓이거나 밥을 지을 때의 과정 등 일상생활에서 일어나는 일들을 알고리즘 교육을 위한 소재로 활용할 수 있다.

둘째, 학생들의 발달 단계와 인지적 수준에 적합해야 한다. 특히 타 교과와 연계한 알고리즘을 지도할 때에는 해당 학년의 교육과정에서 제시한 학습 요소와 일치하는지를 확인하고, 지나치게 높은 학년이나 낮은 학년의 학습 요소가 포함되지 않도록 한다. 예를 들면, 초등학교 3학년 학생들에게 최대공약수를 구하는 알고리즘을 설명하는 것은 적절하지 않다.

셋째, 알고리즘의 기본적인 구조를 포함해야 한다. 알고리즘을 구안하는 행위는 프로그램을 작성하기 전에 추상적으로 접근하는 사고의 과정이다. 프로그램 작성에 필요한 순차성(sequence), 선택성(condition), 반복성(iteration) 등 기본적인 알고리즘의 구조를 습득할 수 있어야 하며, 이를 바탕으로 보다 복잡한 알고리즘을 접할 수 있도록 해야 한다.

넷째, 추상적인 활동보다는 조작이나 체험을 통한 구체적인 활동이 가능해야 한다. 구체적 조작기에 있는 초등학교 저학년의 경우에는 추상적인 개념보다는 구체적인 활동을 통해서 알고리즘을 발견하도록 해야 한다. 시각적으로 보이는 물체를 만지거나 조작하면서 문제 해결 절차를 체득할 수 있도록 해야 한다.

### 2.2.2 언플러그드 활동

언플러그드 활동은 뉴질랜드의 팀벨(Tim Bell) 교수가 구안한 교수학습 방법이다. 팀벨 교수는 ‘놀이로 배우는 컴퓨터 과학’이라는 교재에서 학습자들이 컴퓨터의 동작 원리와 창의적인 아이디어를 게임이나 놀이, 구체적인 조작 활동을 통해 학습함으로써 소프트웨어 교육의 원리를 배울 수 있는 교육 방법으로 제시하였다[15].

언플러그드 활동을 알고리즘 교육에서 적용할 때 교육 내용 측면, 활동 주제 측면, 교육 방법 측면, 수업 시간 측면, 교육 자료 측면에서 고려해야 할 사항을 정리하면 다음과 같다[14][12].

첫째, 교육 내용 측면에서 학생을 고려한 추상화 수준을 설정해야 하며, 다양한 수준의 교육 내용을 준비해야 한다. 학생들의 발달 단계와 추상화 및 구체화 능력, 알고리즘 교육에 대한 사전 지식에 따라 학생들이 느끼는 난이도가 다르다. 따라서 그에 맞는 다양한 수준의 내용을 제시하고, 학생들이 개발한 알고리즘이나 해결 방법을 개선하여 범용적인 해결책으로 일반화할 수 있도록 지도해야 한다.

둘째, 활동 주제 측면에서 학습 효과를 고려하여 알고리즘 교육과 언플러그드 활동 주제를 연결해야 한다. 흥미 중심의 놀이로만 끝나는 언플러그드 활동이 아니라, 놀이를 통해 알고리즘 사고력을 키울 수 있어야 한다. 초기에는 복잡한 알고리즘을 이해하지 못하더라도 놀이나 게임을 통해 학습자들의 동기를 유발하고, 언플러그드 활동 자체가 재미있어야 한다.

셋째, 교육 방법 측면에서 알고리즘 교육을 할 때 항상 컴퓨터가 필요한 것은 아니다. 알고리즘의 교육 목적이 컴퓨터 활용 능력을 키우는 것보다 컴퓨터 과학적 사고력을 키우는 것이 더 중요하기 때문에 언플러그드 활동이 더 효과적이라고 생각된다면 컴퓨터를 활용하지 않을 수 있다. 특히 저학년의 경우 컴퓨터의 모니터 앞에서 하는 수업보다 실습 중심의 수업, 놀이 중심의 수업, 게임 중심의 수업 등 언플러그드 활동이 더 효과적일 수 있다.

넷째, 수업 시간 측면에서 학생들의 사고를 촉진할 수 있는 최소한의 시간을 확보해야 한다. 학생들이 언플러그드 활동을 통해 새로운 알고리즘을 구안하는 데

에는 많은 시간이 필요하다. 언플러그드 활동은 학습 내용을 전달하는 것보다 새로운 해결책을 알고리즘으로 표현하는 것이 중요하므로 깊이 있는 사고를 할 수 있도록 수업 시간이 충분히 확보되어야 한다. 또한, 자신만의 편견에 빠지지 않으려면 모둠 활동과 구성원 간의 토론 및 학급 발표가 필수적이다.

다섯째, 교육 자료 측면에서 학생들이 쉽게 접할 수 있는 자료를 준비하고, 학생의 활동 규모를 고려하여 자료를 준비해야 한다. 언플러그드 활동 규모에 따라 동일한 내용의 자료라도 그 크기를 달리해야 한다. 즉, 협동적으로 활동하는 자료는 대형 자료를 준비하는 것이 좋다. 언플러그드 활동에 참여하는 학생뿐만 아니라 그것을 지켜보는 동료들도 자신의 활동과 비교하고 검토해봄으로써 자신의 해결 방법을 수정하고 보완할 수 있다. 그러기 위해서는 교수 자료의 크기가 적절한 규모를 갖는 것이 필요하다.

### 2.2.3 블록 프로그래밍

언플러그드 활동은 학생들이 재미있게 체험하며, 깊이 있는 사고를 할 수 있다. 그러나 체험하는 경우의 수가 실제보다는 적고, 실행하는 데 걸리는 시간과 준비하는 데 걸리는 시간이 너무 많이 소요된다. 따라서 스크래치나 Code.org, 엔트리와 활용한 블록 프로그래밍 교육이 필요하다. 스크래치와 Code.org, 엔트리와 같은 블록형 프로그래밍 도구는 웹상에서 쉽게 사용할 수 있고, 다양한 미션이 제공되어 학생들이 흥미를 갖고 도전할 수 있다.

특히 스크래치(Scratch)는 아이들에게 그래픽 환경을 통해 컴퓨터 프로그래밍에 관한 경험을 쌓기 위한 목적으로 설계된 대표적인 블록 프로그래밍 언어이다. 스퀸(Squeak)을 기반으로 스몰토크(Smalltalk)라는 언어로 작성되었으며, 블록 조각을 조립하는 형태로 쉽게 프로그래밍할 수 있다. 스크래치는 2007년 MIT Media Lab의 연구팀이 어린 아이들에게 프로그래밍의 기본 개념과 알고리즘을 가르치기 위해 개발하였으며, 다양한 멀티미디어 지원을 통해 쉽게 게임이나 애니메이션 등을 만들 수 있다. 프로그래밍 구조를 익히는 것과 논리적인 문제에 초점을 맞추어 초보자의 프로그래밍 입문과 정으로 적합하다[10].

또한, Code.org나 Entry와 같은 미션 중심의 블록 프로그래밍 도구를 활용하면 오랜 시간을 준비해야 하는 언플러그드 활동의 불편함을 최소화시킬 수 있다. 언플러그드 활동을 위해서는 사전에 자료를 제작해야 하고, 학생들이 생각한 알고리즘을 직접 실행하는 데 많은 시간이 소요될 수 있다. 그러나 블록 프로그래밍을 활용하면 이미 개발된 미션을 한 단계씩 밟아가면서 알고리즘의 핵심 개념을 이해할 수 있다.

### 2.2.4 체감형 프로그래밍

블록형 프로그램을 통해 얻어진 결과를 실제로 학생들이 만들어보고 체감하는(Tangible) 활동이 필요하다. 컴퓨터 화면 속에서만 이루어지는 활동이 아니라 로봇이나 아두이노(Arduino) 등을 활용하여 구안한 알고리즘을 직접 프로그램으로 작성해보고, 그 결과에 따라 움직이는 모습을 체감함으로써 알고리즘이 더 이상 머릿속에 머물러 있는 것이 아니라, 실세계에 영향을 미칠 수 있음을 알게 해야 한다. 특히 구체적 조작기에 있는 어린 학생들에게 물체를 직접 만져보면서 알고리즘을 배우는 것은 매우 효과적이다.

체감형 프로그래밍은 로봇이나 아두이노와 같이 직접 학생들이 교구를 만져보면서 알고리즘 교육을 할 수 있다. 로봇을 활용한 알고리즘 교육의 장점은 다음과 같다.

첫째, 로봇을 이용한 교육은 인간이 가진 움직임이나 형태에 관한 인지 매커니즘이 강하게 작용함으로써 학습자에게 흥미와 동기를 유발할 수 있다.

둘째, 로봇은 만든 대로 동작하기에 학습 성과의 평가를 즉각적이고 생생하게 실현시킬 수 있다.

셋째, 로봇은 융합 기술을 배울 수 있다. 로봇 기술은 컴퓨터에서 모터 제어, 센싱 기술, 기계 요소라고 일컬어지는 융합과학기술의 결정이다. 즉, STEAM 교육의 창의적인 사고력을 가진 융합 인재 양성의 측면에서 로봇은 더할 나위 없이 좋은 형태의 교구이다.

넷째, 로봇 프로젝트 학습은 모둠 단위로 이루어지는 문제 해결함으로써 협업 능력과 리더십을 기를 수 있다.

그러나 이러한 로봇을 이용한 알고리즘 교육을 실시할 때에는 로봇에 대한 기본적인 이해가 필요하다.

첫째, 로봇의 움직임은 시간이 걸린다. 전기 모터와

기어박스 등으로 이루어진 로봇의 구동부는 일정한 속도 이상 더 빨리 움직일 수 없다. 아무리 빨리 명령을 내려도 로봇이 따라갈 수 없다면 잘못된 프로그램을 작성한 것이다. 따라서 명령과 명령 사이에는 잠시 대기하는 시간이 반드시 필요하다.

둘째, 로봇의 센서와 구동에는 시간 지연이 발생할 수 있다. 프로그램의 실행은 명령 요청에 따라 즉시 이루어지지만 로봇의 센서 변화가 프로그램에 입력되기까지 시간 지연이 발생한다. 따라서 시간 지연을 고려한 프로그램을 작성해야 한다. 또한, 계산된 값을 로봇에 전달하는 데 시간 지연이 발생하므로 이를 고려한 프로그램을 작성해야 한다.

셋째, 로봇의 입력 값은 정확하지 않다. 외부 환경을 검출하는 센서는 물론 로봇 내부 상태를 검출하는 센서 값에는 항상 부정확한 노이즈가 포함되어 있다. 특히 초음파 센서가 측정하는 거리는 측정할 때마다 조금씩 차이가 있다. 이런 경우에는 노이즈를 줄이기 위한 방법을 반드시 생각하여 부정확한 판단이 일어나지 않도록 주의해야 한다.

아두이노는 오픈소스를 기반으로 한 단일 보드 마이크로컨트롤러이다. 아두이노는 하드웨어와 통합 개발 환경으로 구성된다. 아두이노를 사용하면 빛, 소리, 접촉, 움직임 등에 반응하거나 이러한 것들을 제어할 수 있는 도구를 만들 수 있다. 임베디드 개발 경험이 전혀 없는 사람을 위해 개발된 교육용 플랫폼이기 때문에 프로그램을 작성하고 보드에 프로그램을 올리는 과정을 단순화하여 다루기 쉽게 되어 있다. 아두이노를 이용하여 만들 수 있는 것들의 예로는 악기, 로봇, 빛 조명, 게임, 인터랙티브 가구, 의상 등이 있다.

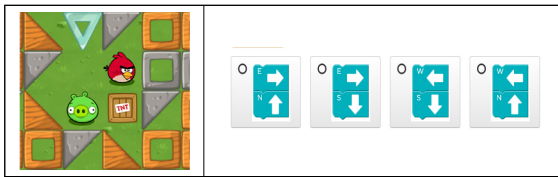
### 3. 알고리즘 교육의 평가 유형

알고리즘에 대한 평가 유형을 살펴보기 위해 Code.org에서 제공하는 ‘Hour of Code’의 800여 개 가량의 미션을 분석하였다. 알고리즘에 대한 평가 내용과 방법은 알고리즘 선택하기, 알고리즘 채우기, 알고리즘 수정하기, 알고리즘 예측하기 등 4가지 유형으로 구분하였다.

### 3.1 알고리즘 선택하기

알고리즘 선택하기는 주어진 화면이나 조건에 맞는 알고리즘을 고르거나, 주어진 규칙에 맞는 카드 세트를 선택하게 하는 활동으로 구성되어 있다.

① 두 방향의 블록을 사용하여 카드 선택하기. (Fig. 1)과 같이 왼쪽, 아래쪽이라고 쓰인 카드 대신에 W와 S 문자로 표기하거나 방향키로 기호화하여 사용하였다. 해당 미션을 완성하는 데 필요한 알고리즘을 제대로 이해하고 있는지를 확인한다.



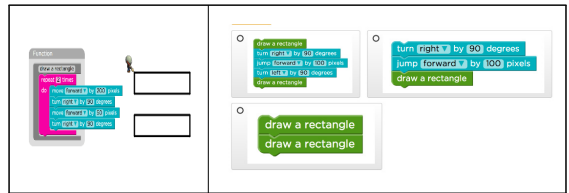
(Fig. 1) select the appropriate card

② 주어진 화면에 맞는 블록 선택하기. (Fig. 2)와 같이 2~3개 방향의 카드와 미로 퍼즐을 이용하여 알고리즘을 생각하게 한다. 각 미로 퍼즐로부터 각각의 블록 알고리즘을 작성해본 후 선택지와 맞춰보거나, 각 알고리즘이 의미하는 위치 변화를 머릿속으로 그려 해결한다.



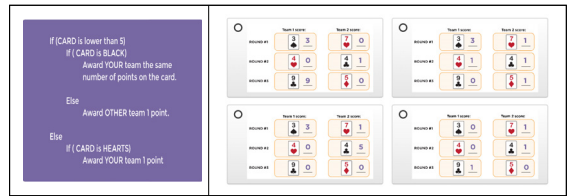
(Fig. 2) select the appropriate blocks

③ 주어진 화면에 맞는 프로그램 선택하기 주어진 함수의 기능을 파악하여 문제를 해결하도록 한다. 예를 들면, (Fig. 3)과 같이 주어진 함수를 이용하여 두 개의 직사각형을 그리는 알고리즘을 선택하는 문항이다. 함수의 기능을 적절히 사용할 수 있는지 평가한다. 각각의 선택지의 블록 알고리즘을 이용한 그림을 그려본 후 적절한 선택지를 선택하게 한다.



(Fig. 3) select the appropriate program

④ 규칙에 맞는 카드 선택하기. (Fig. 4)와 같이 보기 화면의 If문과 Else문을 읽어보며 선택지의 점수를 계산한 후 해당된 카드를 선택한다. 조건이 중첩되어 있을 때의 조건과 Else문의 의미를 파악하여 문제를 푼다.



(Fig. 4) select the appropriate cards

⑤ 결과에 맞는 함수 선택하기. (Fig. 5)와 같이 주어진 그림과 블록 알고리즘을 보고 적절한 함수를 찾을 수 있는지 평가한다.



(Fig. 5) select the appropriate function

### 3.2 알고리즘 채우기

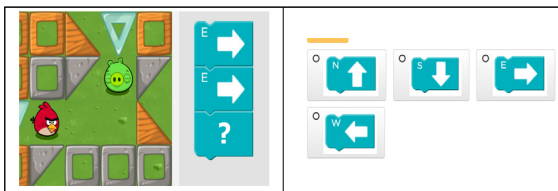
알고리즘 채우기는 알고리즘의 일부나 전체, 시작이나 끝 부분을 채우는 활동이다.

① 처음에 시작하는 알고리즘 채우기. (Fig. 6)과 같이 이를 닦는 하나의 큰 활동을 여러 개의 작은 활동으로 나누어 논리적인 순서대로 완성하도록 한다. 이를 닦는 활동의 알고리즘을 완성하면서 현실 상황과 알고리즘을 더욱 잘 연관시키게 된다.



(Fig. 6) Fill the first blank of the algorithm

② 마지막에 끝나는 알고리즘 채우기. (Fig. 7)과 같이 적절한 알고리즘을 구성할 수 있는지를 평가하기 위해 새의 위치 바꾸면서 마지막 알고리즘을 완성하게 한다.

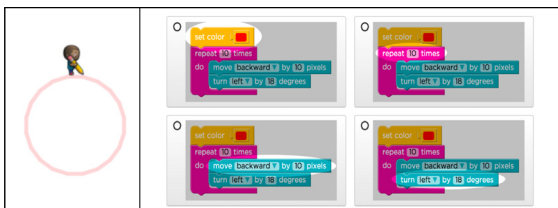


(Fig. 7) Fill the last blank of the algorithm

### 3.3 알고리즘 수정하기

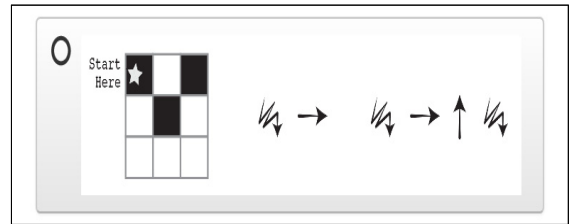
알고리즘 수정하기는 주어진 알고리즘에서 어떤 부분이 잘못되었는지를 찾거나, 그것을 수정하게 하는 활동이다.

① 주어진 알고리즘에서 오류 찾기. (Fig. 8)와 같이 원을 그리기 위한 알고리즘에서 잘못된 부분을 찾는다. 예를 들면, 원의 내부 각도가 360°임에 유의하여, 왼쪽으로 18°씩 회전하며 10픽셀씩 뒤로 이동하는데, 이것을 10번 반복했을 때 어디까지 그려지는지 예상해본다.



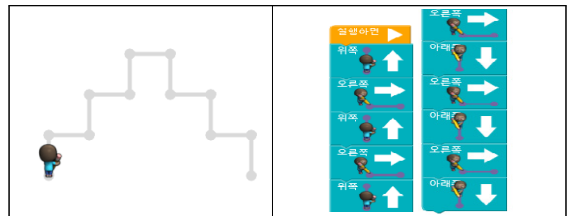
(Fig. 8) Find a error in the algorithm

② 오류 카드를 찾아 수정하기. (Fig. 9)와 같이 모눈 칸에 색을 칠하고, 그 동작을 수행하는 데 필요한 기호들을 제시한다. 그리고 그곳에서 오류가 있는지 찾고, 있다면 그것을 수정하게 한다.



(Fig. 9) Correct the error in the algorithm

③ 실패의 경험을 통해 오류 수정하기. (Fig. 10)과 같이 유사하게 반복되는 과정을 알고리즘으로 작성하여 실행해보고, 실패했을 때 힌트를 얻어 조금씩 수정해 가면서 문제를 해결하도록 한다.

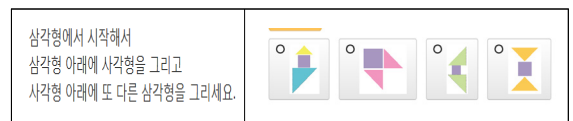


(Fig. 10) Correct the error through mistakes

### 3.4 알고리즘 예측하기

알고리즘 예측하기는 주어진 명령어나 기호, 블록 프로그램의 결과를 예측하여, 문제를 해결하는 활동이다.

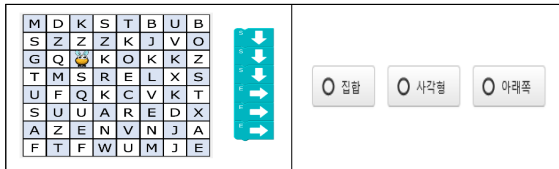
① 주어진 명령어를 보고 결과 맞추기. (Fig. 11)과 같이 도형을 그리는 명령어를 주고, 어떤 그림으로 나타날지를 선택하게 한다. 예를 들면, 각 선택지의 삼각형, 사각형의 크기와 모양에 관계없이 나타내는 도형의 이름만 추상화시키고, 그것의 실행 결과와 맞지 않는 것으로 고르게 한다.



(Fig. 11) Predict the results of the command

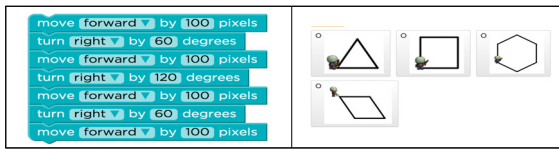
② 방향 기호를 활용하여 단어 찾기. (Fig. 12)와 같이 단어 퍼즐에서 주어진 방향 기호에 따라 이동한 후 완성될 단어를 찾는 문항이다. 블록 알고리즘을 통하여

이동했을 때 이동할 방향을 예상할 수 있는지를 평가한다. 학생들은 알고리즘대로 순차적으로 풀벌을 이동시켜 문제를 해결한다.



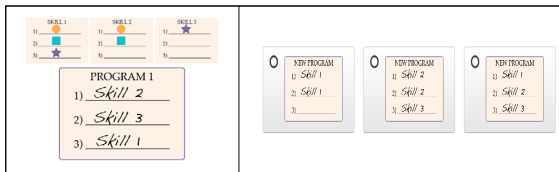
(Fig. 12) Predict the results of the sign

③ 블록 프로그램의 소스를 보고, 그 결과를 예측한다. 예를 들면, (Fig. 13)과 같이 제시된 알고리즘에 따라 그림을 그려가며 선택지와 비교해보고, 동일한 결과를 선택한다.



(Fig. 13) Predict the results of the program

④ 동일한 결과를 나타내는 알고리즘 선택하기. 함수를 이해하고 동일한 프로그램을 파악하게 한다. 예를 들면, (Fig. 14)와 같이 프로그램 1과 동일한 결과를 갖는 프로그램을 모두 선택하게 한다. 함수로 이루어진 하나의 프로그램을 파악하여 같은 프로그램을 찾아낼 수 있는지 평가한다.



(Fig. 14) Find the same results

#### 4. 결론 및 제언

초등학교의 알고리즘 교육의 목표는 컴퓨터 과학적 사고를 키우는 것이다. 문제를 분석하고 분해하는 과정을 통해 문제를 이해하고, 그것을 모델화하고 추상화하면서 알고리즘을 작성하는 능력이 요구된다[7]. 2015년

개정 교육과정에서 초등학교부터 소프트웨어 교육이 강화됨에 따라 알고리즘 교육이 중요시되고 있다. 따라서 본 연구에서는 알고리즘을 중심으로 교육 내용과 교육 방법, 평가 방법을 제시하였다. 특히 세계적으로 널리 사용되고 있는 소프트웨어 교육 플랫폼인 Code.org의 미션들을 분석하여 알고리즘의 평가 유형을 알고리즘 선택하기, 알고리즘 채우기, 알고리즘 수정하기, 알고리즘 예측하기 등 4가지로 제시하였다.

본 연구 결과를 토대로 한국정보교육학회에서 제시한 소프트웨어 교육 과정을 분석하여[7], 알고리즘의 세부 내용별로 평가 유형을 구체화한다면 학교에서 알고리즘 교육하는 데 도움이 될 것이다. 또한, 학생 평가뿐만 아니라 교수 방법으로 제안한 생활 속 코딩, 언플러그드 활동, 블록 프로그래밍, 체험형 프로그래밍 활동에 필요한 구체적인 사례집과 교재를 지속적으로 개발하여 학교 현장에 보급한다면 소프트웨어 교육을 처음 시작하는 교사들에게 알고리즘 교육에 대한 자신감을 갖게 하여 소프트웨어 교육 발전에 기여할 것이다.

#### 참고문헌

[1] Chosun Biz (2014). Apple, Up to two successive quarterly sales... Samsung bigger crisis. updated 2014. 7. 24. [http://biz.chosun.com/site/data/html\\_dir/2014/07/24/2014072400004.html](http://biz.chosun.com/site/data/html_dir/2014/07/24/2014072400004.html).

[2] Frey, C. B., & Osborne, M. A. (2013). The future of employment: how susceptible are jobs to computerisation. Retrieved September, 7, 2013.

[3] Han Sanggun, Lee Sangdon, Jang Juhee, Kim Minkyung, Yun Soorin, Park Bongnam, Lee Minyoung & Shim Jaepil (2014). 2014 Future work world : Vocational side. Ministry of Education & Korea Research Institute for Vocational Education & Training.

[4] Jeong Youngsik (2014). A Study on the Content Framework of Algorithm Education in Primary and Middle School. *Journal of The Korean association of information education*, 18(2), 275-284. Korean Association of Information Education.



- [5] Jeong Youngsik (2015). Domestic and international software education trends, and suggestion of software education curriculum in elementary schools, 1-40. Korea Information Science Education Forum.
- [6] Jeong Youngsik, Yu Jungsu, Lim Jinsuk, Son Yukyoung (2015). Theory of Software Education. Simass.
- [7] Kim Gapsu, Kim Chul, Kim Hyunbae, Jeong Ingi, Joeng Youngsik, Ahn Sunghun, Kim Jongwoo (2014). A Study on Contents of Information Science Curriculum. *Journal of The Korean association of information education*, 18(1), 161-171. Korean Association of Information Education.
- [8] Ministry of Education (2015). Software Education Operating Instructions.
- [9] Ministry of Science, ICT and Future Planning & National IT Industry Promotion Agency (2015). What is the software oriented society. <http://www.software.kr/um/um08/um0801/um080101.do>.
- [10] MIT Media Lab (2015). About Scratch. <https://scratch.mit.edu/about/>
- [11] Money Today (2015). '1 trillion won', Samsung-M S, 'Patent dispute shut down. updated 2015. 2. 10. <http://news.mt.co.kr/mtview.php?no=2015021010274891270&type=1>.
- [12] Oh Kyungsun, Ahn Seongjin (2012). "Tim Bell" using the unplugged computing on the improvement of Computational Thinking. The Proceeding of the Korean Association of Computer Education 2012, 16(2), 37-40. The Korean Association of Computer Education
- [13] P21 (2010). We're taking teaching and learning above & beyond. Partnership for 21st Century Skills.
- [14] Seo Insuk, Kim Jonghan, Kim Taeyoung (2011). Research of Instructional Design on Unplugged Cooperative Learning for Elementary Information unit area. The Proceeding of the Korean Association of Computer Education 2011, 15(1), 79-84. The Korean Association of Computer

Education.

- [15] Tim Bell, Ian H. Witten, & Mike Fellow (2015). CS Unplugged: An enrichment and extension programme for primary-aged students. <http://csunplugged.org>.

## 저자소개



### 김철

1997 전남대학교 대학원 전산통계학과(이학박사)

1998 University of Washington (객원교수)

1992~현재 광주교육대학교 컴퓨터교육과 교수

관심분야: 인터넷자원관리, 교육용콘텐츠, 로봇활용교육, e-Learning

e-mail: [chkim@gnue.ac.kr](mailto:chkim@gnue.ac.kr)

