

전문계 고등학교 학습자의 알고리즘 시간효율성에 관한 오개념 분석

이은경[†]

요 약

본 연구에서는 전문계 고등학교 학생들의 알고리즘 시간효율성에 관한 개념 이해 수준을 확인하고 빈번하게 발생하는 오개념 유형들을 확인하고자 하였다. 선행연구에서 확인된 세 가지 유형의 오개념들을 토대로 9개의 오개념 측정 문항을 개발하였으며, 프로그래밍 관련 과목을 이수한 51명의 전문계 고등학교 학생들을 대상으로 조사를 실시하였다. 연구결과, 알고리즘 시간효율성에 관한 개념 이해도는 매우 낮은 수준으로 나타났으며, 견고하고 반복적으로 발생하는 여러 유형의 오개념들을 확인하였다. 마지막으로, 확인된 오개념들을 과학적 개념으로 변화시키기 위한 교육적 전략인 시뮬레이션 및 알고리즘 효율성 평가 틀의 활용 방안과 함께 교육과정 개선의 필요성을 제안하였다.

주제어 : 알고리즘 시간효율성, 오개념, 프로그래밍 교육

Analysis of Vocational High School Learners' Misconceptions on the Time-Efficiency of Algorithms

Eunyoung Lee[†]

ABSTRACT

The purpose of this study is to explore vocational high school students' understanding levels and misconceptions regarding the time-efficiency of algorithms. In this study, a questionnaire with nine questions was developed. The questionnaire is base on three misconceptions about the time-efficiency of algorithms that are examined through prior studies. The questionnaire was applied to 51 students who completed a programming course. The result shows that the students didn't sufficiently understand the concept of time-efficiency of algorithms and several common misconceptions appeared repeatedly. Finally, teaching strategies using simulation and assesment framework for algorithm's time-efficiency were presented and some recommendations were suggested for refining the curriculum relating computer programming and algorithms.

Keywords : Time-Efficiency of Algorithm, Misconception, Programming Education

[†] 종신회원: 연세대학교 교육대학원 특임교수

논문접수: 2011년 06월 29일, 심사완료: 2011년 08월 13일, 게재확정: 2011년 08월 19일

1. 서론

컴퓨터와 관련된 교육과정은 다른 교과에 비해 실생활과 밀접한 관련을 갖고 있으며, 학습한 지식과 기술의 실제적 활용 능력을 강조한다. 특히, 전문계 고등학교에서 이루어지는 컴퓨터 교육은 다른 계열이나 학교급에서 이루어지는 교육에 비해 기능적 훈련을 통한 실무 능력의 함양이 강조되는 편이다. 이에 따라 현장에서는 각종 컴퓨터 관련 자격증을 취득하거나, 다양한 응용 소프트웨어 사용 기술 습득을 위한 교육이 주로 이루어지고 있다. 그러나 현대와 미래 사회가 요구하는 인재가 갖추어야 할 실제적 역량은 기능적 숙련의 단계를 넘어 창의적 문제해결력과 같은 보다 고차원적인 인지적 능력을 바탕으로 강화될 수 있다. 이러한 관점에서 2007 개정 교육과정을 통해 전문계 고등학교의 컴퓨터 관련 교과들의 성격, 목표, 교육 내용이 대폭 변경되었으며, 2009 개정 교육과정으로 연계되고 있다. 특히, 프로그래밍 관련 교과의 경우, 컴퓨터 과학의 기본 개념과 원리를 통한 문제해결력의 향상을 위해 알고리즘 관련 내용이 강화되었다[1][2].

그러나 학교현장에서 이루어지고 있는 프로그래밍 교육은 여전히 특정 프로그래밍 언어의 문법 또는 개발 도구 활용법 습득에 치중하고 있으며, 이러한 현상의 원인은 학습자의 인지적 수준, 현행 교육과정 운영 방식, 교수자의 인식 등 여러 가지 요인이 복합적으로 작용한 결과로 볼 수 있다.

첫째, 전문계 고등학교 학습자들은 다른 계열 고등학생들에 비해 인지적 수준 및 동기 수준이 낮은 편이지만, 프로그래밍 관련 교육 내용은 높은 인지 발달 수준을 요구한다. 프로그래머는 프로그래밍 언어에 관한 이해 및 개발 도구 사용 능력 뿐 아니라, 알고리즘 설계와 분석을 위한 문제해결력, 비판적 사고 능력을 필수적으로 갖추어야 하기 때문이다[3].

둘째, 전문계 고등학교의 경우, 교육과정 편성·운영 권한에 있어 학교 재량권이 강화되어 있다. 따라서 알고리즘 및 프로그래밍에 관한 교사의 역량에 따라 학습자들에게 제공되는 교육내용에 질적 차이가 발생할 수 있다.

이러한 측면에서 학습자의 개념적 상태와 인지

적 수준을 확인하는 과정은 보다 실용적인 교육과정 운영 및 교수 학습 방법 설계를 위해 반드시 필요하다. 보다 효율적인 알고리즘에 기반한 프로그램이 무엇인지 판단하기 위한 알고리즘 효율성에 관한 개념은 컴퓨터 과학 분야에서 가장 기초적이고 핵심적인 개념이지만, 학습자들이 이해하기 어렵다는 특성을 지닌다. 특히, 인지적 수준이 낮은 학습자들의 경우 알고리즘 효율성에 관한 다양하고 견고한 오개념들을 형성하고 있다. 이러한 오개념들을 과학적 개념으로 변화시키는 과정을 거쳐 학습이 이루어질 수 있지만, 개념 변화의 과정은 단지 반복적 교수나 강조를 통해 쉽게 이루어지지 않기 때문이다[4]. 따라서 과학적 개념 형성 및 개념 변화 유도를 위한 면밀히 교수 설계가 요구되며, 학습자의 개념적 상태를 파악하기 위한 연구가 선행되어야 한다.

이에 따라 본 연구에서는 전문계 고등학교 학생들을 대상으로 알고리즘 시간효율성과 관련된 개념 이해 수준을 확인하고, 빈번하게 나타나는 오개념 유형들을 파악함으로써 체계적인 프로그래밍 교육 개선을 위한 기초자료를 제공하고자 한다.

2. 관련 연구

학생들은 일상생활에서의 경험을 통해 현상에 관한 나름대로의 개념을 형성하며, 이미 형성된 개념들을 바탕으로 또 다른 개념들을 획득하게 된다. 수업 이전 일상생활을 통해 직관적으로 형성된 개념은 주로 과학적 개념과 일치하지 않는 경우가 많지만, 상당히 오랫동안 지속되며 교육에 의해 쉽게 과학적 개념으로 변화되지 않는다[5][6][7]. 이처럼 과학적 개념에 모순된 개념이나 원칙을 오개념이라 하며, 수학이나 과학 분야의 교육자들은 각 영역에서의 오개념을 확인하고 어떻게 과학적 개념으로 변화시킬 것인가에 관해 지속적으로 연구해왔다.

이에 반해, 컴퓨터 과학 분야의 오개념 연구는 미흡한 실정이며, 과거의 관련 연구들은 특정 주제에 치중하여 지엽적으로 이루어져 왔다. 초창기 연구들은 주로 프로그래밍과 관련된 연구들로 교육적 처치를 위한 오개념 확인이라기보다 초보

프로그래밍 학습자의 정신 모델이나 인지적 표현을 확인하는데 중점을 두고 이루어졌다[8][9]. 또한 자료 할당[10], 매개 변수[11], 객체[12]와 같이 프로그래밍과 관련하여 매우 세부적인 주제에 관한 오개념을 확인하는 연구가 이루어졌으나, 교육과정 개선이나 교육적 처치를 위한 근거로 활용하기에 충분하지 못하다. 그러나 가장 최근의 연구들은 컴퓨터 과학 관련 교육과정 개선 및 교육적 처치를 위한 자료 수집에 초점을 두고 진행되고 있으며, 크게 두 유형으로 분류된다.

첫째, 컴퓨터 과학 분야의 개념 목록(CI: Concept Inventory) 개발과 관련된 연구들이다.

개념 목록은 과학 분야 교육자들에 의해 처음 시도된 것으로, 중요하지만 학습시키기 어려운 과학적 개념들과 관련하여 학습자들이 지니고 있는 오개념들을 목록화한 것이다. 교사들은 교수 학습 설계 전 해당 목록을 살펴봄으로써 학습자의 개념적 상태를 미리 파악하고, 오개념 예방 및 교정을 위한 교육적 처치 설계가 가능하기 때문에 이러한 개념 목록의 확보 및 개발은 매우 중요한 과정이라고 할 수 있다. 이에 따라 컴퓨터 과학 분야에서는 이산 구조, 디지털 논리, 프로그래밍 기초와 관련된 개념 목록들의 개발이 시도되었다[13][14][15].

둘째, 컴퓨터 과학 분야의 교육과정 개선 및 교수 학습 전략 제시를 위한 기초 자료로 활용하기 위한 오개념 수집 및 분석과 관련된 연구들이다.

Gal-Ezer and Zur[16]는 이스라엘 고등학교 10-11학년을 대상으로 알고리즘 시간효율성에 관한 오개념 연구를 진행하고, 이들을 예방하기 위한 교육적 방법들을 제시하였다. 특히, 연구 결과 도출된 각 오개념들의 형성 원인을 공통적인 사고 양식인 직관적 규칙에 기인한 것으로 해석하였다[17].

이은경과 이영준[4]은 알고리즘 설계와 분석에 관한 국내 고등학교 학습자들의 개념 이해도와 오개념 유형을 분석하였다. 연구 결과, 보다 효율적인 알고리즘 설계를 위해 요구되는 개념 이해도가 낮은 수준으로 나타난 것을 확인하였으며, 오개념 형성에 기인하는 직관적 사고 규칙들을 발견하였다.

Özdener[18]는 터키의 전문계 고등학교 및 대학

생들을 대상으로 알고리즘 시간효율성에 관한 오개념 연구를 진행하였다. 연구 결과, 고등학생 및 대학생 모두 동일한 유형의 오개념을 지속적으로 가지고 있다는 것을 확인하고 이를 해결하기 위한 방안으로 프로그래밍 수업 이전에 알고리즘에 관한 수업이 선행되어야 함을 제시하였다. 또한 프로그래밍 수업을 위한 교육과정이 단지 프로그래밍 언어의 명령어를 익히는 것이 아니라, 프로그래밍의 논리적 절차를 이해하고 알고리즘을 학습할 수 있도록 변경되어야 함을 지적하였다.

오개념 연구는 학생의 개념적 상태를 미리 확인하기 위한 단계로 현행 교육과정 및 방법의 문제점을 확인하고 개선사항을 도출하기 위한 기초 자료로 활용될 수 있다. 가능한 많은 내용 영역에 관한 오개념 정보를 수집하는 것도 중요하지만, 모든 내용 영역의 학습에 기본이 되는 핵심 영역에 관한 오개념을 확인하는 것이 선행되어야 한다. 또한 확인된 오개념 분석을 통해 오개념을 유발하는 일반화된 사고 패턴을 발견하고 이를 수정하기 위한 교육적 처치가 이루어진다면 보다 근본적인 개념 변화 학습이 가능할 것이다.

3. 연구방법

3.1 연구가설

선행 연구[16][18]에서 도출된 알고리즘 시간효율성과 관련한 오개념들을 토대로 다음 세 가지 가설을 설정하였다.

- 가설 1. 학습자들은 더 짧은 프로그램의 시간효율성이 더 좋다고 판단할 것이다.
- 가설 2. 학습자들은 문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다고 판단할 것이다.
- 가설 3. 학습자들은 변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다고 판단할 것이다.

3.2 연구대상

본 연구에서는 경기도에 위치한 전문계 고등학교 3학년 2개반 51명의 학생들을 대상으로 알고

리즘의 시간효율성에 관한 오개념을 측정하였다. 해당 학생들은 2학년 과정에서 ‘프로그래밍 실무’ 과목을 이미 이수하였으며, 해당 과목의 수업은 주당 5시간 총 170시간에 걸쳐 이루어졌다.

3.3 연구도구

알고리즘의 시간효율성에 관한 오개념 측정 도구는 선행연구[16][18]에서 사용한 3개의 문항을 포함한 총 9개의 문항들로 구성된다. 각 문항들은 알고리즘 시간효율성에 관해 학생들이 지니고 있을 것으로 예상되는 3가지 유형의 오개념을 토대로 개발되었으며, 문항의 타당성 확보를 위해 알고리즘 및 프로그래밍 교육 전문가 5인의 검토를 통해 수정·보완하였다. 문항의 형태는 같은 문제를 해결하기 위한 두 프로그램 코드를 함께 제시하고, 더 효율적이라 판단되는 프로그램을 선택하도록 제작하였다. 또한 학습자의 개념적 상태를 구체적으로 확인하기 위해 해당 답안을 선택한 이유를 함께 서술하도록 구성하였다. 모든 문항에서 분석 대상으로 제시된 프로그램들은 C 언어로 작성되었으며, 표준입출력함수, 변수, 기본적인 제어문들을 포함한다. 구체적인 문항 구성은 <표 1>과 같다.

<표 1> 문항 구성

오개념 유형	문항번호	형태
더 짧은 프로그램의 시간효율성이 더 좋다.	1, 4*, 7	선택형 + 설명
문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	2, 5*, 8	
변수의 수가 적은 프로그램의 시간효율성이 더 좋다.	3, 6*, 9	

* Gal-Ezer & Zur[16], Özdenler[18]에서 제시한 문항과 동일

3.4 측정 및 분석방법

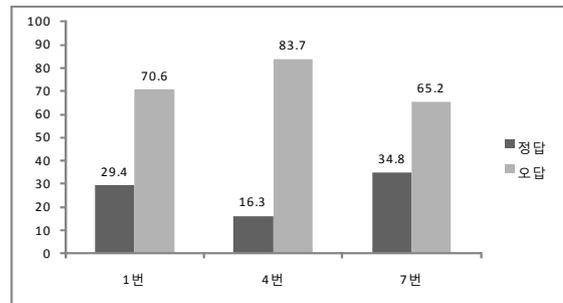
측정 전 1차시에 걸쳐 알고리즘 시간효율성이 무엇이며 어떻게 판단하는지에 관한 사전교육을 실시하였으며, 모든 측정 문항에 시간효율성의 개념과 판단 기준을 함께 제시하였다. 조사된 자료는 문항별 빈도분석을 실시하여 과학적 개념 형성 및 오개념 형성의 정도를 비교하였다. 정답과 오답의 처리는 선택형에서 옳은 답을 선택하고, 해당 답안을 선택한 이유에 대한 설명이 과학적

으로 타당할 경우 정답으로 처리하였으며, 선택형에서 옳은 답을 선택하였으나, 정확한 설명이 이루어지지 못하거나 무응답의 경우 오답으로 처리하였다. 또한 추가적으로 기술한 설명 분석을 통해 학습자가 지니고 있는 오개념의 유형을 구체적으로 확인하고 최빈 오개념을 조사하였다.

4. 연구결과

4.1 가설 1

‘더 짧은 프로그램의 시간효율성이 더 좋다.’ 라는 오개념을 확인하기 위한 3문항의 정답 및 오답 비율은 <그림 1>과 같다.



<그림 1> 가설 1의 정답률 비교

또한 응답자의 답안 선택에 관한 설명 분석을 통해 발견한 오개념 유형들은 <표 2>와 같다.

<표 2> 가설 1의 오개념 유형들

문항번호	오개념 유형	빈도 (명)	비율 (%)
1	더 짧은 프로그램의 시간효율성이 더 좋다.	28	54.9
	while 문을 사용하는 것이 for 문을 사용하는 것보다 시간효율성이 더 좋다.	8	15.7
	전체	36	70.6
4	더 짧은 프로그램의 시간효율성이 더 좋다.	35	71.5
	변수의 수가 적은 프로그램의 시간효율성이 더 좋다.	5	10.2
	같은 결과를 출력하면 시간효율성은 같다.	1	2.0
	전체	41	83.7
7	더 짧은 프로그램의 시간효율성이 더 좋다.	23	50.0
	같은 결과를 출력하면 시간효율성은 같다.	4	8.7
	코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.	3	6.5
	전체	30	65.2

1번 문항의 경우, 반복문의 종류가 다르지만 동일한 시간효율성을 가진 두 프로그램을 제시하였으나, 대부분의 학생들은 이를 인지하지 못하고

코드 길이가 더 짧고 간결하게 표현된 프로그램의 시간효율성이 더 좋다고 판단하였으며(54.9%), ‘while 문을 사용하는 것이 for 문을 사용하는 것보다 시간효율성이 더 좋다.’라는 추가적인 오개념(15.7%)을 가지고 있는 것으로 드러났다.

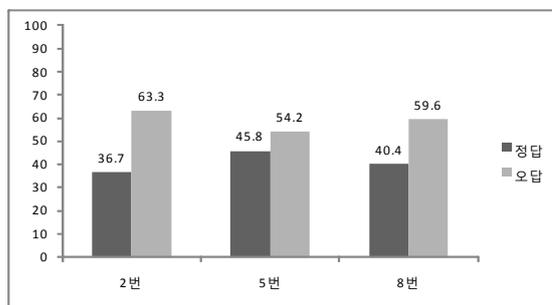
4번 문항은 두 수의 곱을 덧셈 연산만을 사용하여 계산하기 위한 프로그램으로 첫 번째 프로그램이 더 많은 변수를 사용하고, 코드가 길지만, 두 번째 프로그램에 비해 덧셈 연산의 반복 횟수가 적기 때문에 실행시간 측면에서 더 효율적이다. 그러나 83.7%의 학생들이 오답을 제시하였으며, 이 중 71.5%의 학생들은 ‘더 짧은 프로그램의 시간효율성이 더 좋다.’라고 판단하였으며, ‘변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.’(10.2%), ‘같은 결과를 출력하면 두 프로그램의 시간효율성은 같다.’(2.0%) 라는 추가적인 오개념들을 가진 것으로 확인되었다.

7번 문항은 세 개의 양의 정수들 중 가장 큰 값을 찾기 위한 프로그램으로 두 번째 프로그램의 코드가 더 길지만, 실제적인 비교 횟수가 적기 때문에 실행시간 측면에서 더 효율적이다. 1번, 4번 문항에 비해 상대적으로 높은 정답률(34.8%)을 보이고 있지만, 여전히 ‘더 짧은 프로그램의 시간효율성이 더 좋다.’(50.0%) 라는 오개념을 가장 많이 지니고 있는 것으로 확인되었다. 이에 더하여 ‘같은 결과를 출력하면 두 프로그램의 시간효율성은 같다.’(8.7%), ‘코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.’(6.5%) 라는 추가적인 오개념들을 확인하였다. 이 중 ‘코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.’ 라는 판단은 컴파일러가 아닌 프로그래머의 입장에서 변수 및 변수명의 적절한 사용이 해당 코드에 관한 이해를 돕고 이것이 더 효율적인 프로그램이라 판단하는 것으로 보인다.

4.2 가설 2

두 번째 가정인 ‘문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.’라는 오개념을 확인하기 위한 3문항의 정답 및 오답 비율은 <그림 2>와 같다. 또한 응답자의 답안 선택에 관한 설명 분석을 통해 발견한 오개념

유형들은 <표 3>과 같다.



<그림 2> 가설 2의 정답률 비교

<표 3> 가설 2의 오개념 유형들

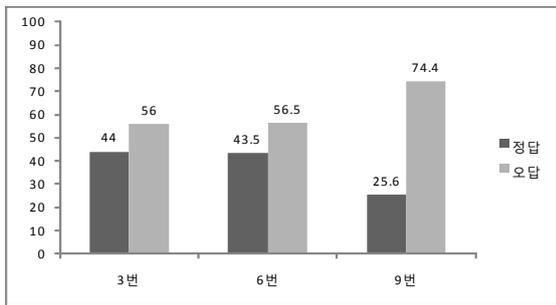
문항 번호	오개념 유형	빈도 (명)	비율 (%)
2	문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	9	18.4
	연산문이 순서적으로 앞에 배치된 프로그램의 시간효율성이 더 좋다.	22	44.9
	전체	31	63.3
5	문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	12	25.0
	연산문이 순서적으로 앞에 배치된 프로그램의 시간효율성이 더 좋다.	14	29.2
	전체	26	54.2
8	문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	15	31.9
	연산문이 순서적으로 앞에 배치된 프로그램의 시간효율성이 더 좋다.	12	25.5
	코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.	1	2.1
	전체	28	59.6

2번 문항에서 제시한 프로그램들은 동일한 문으로 구성되어 있으며, 같은 결과를 출력하지만, 첫 번째 프로그램의 경우, 연산문이 반복문 외부에 배치되어 연산의 횟수가 줄기 때문에 시간적으로 더 효율적이다. 5번과 8번 문항 또한 2번 문항과 동일한 형태로 구성되었다. 세 문항 모두 정답보다 오답 비율이 더 높게 나타났으며, 두 종류의 오개념 유형이 확인되었다. 첫 번째 유형은 기본 가정과 일치하는 것으로 학생들은 동일한 문을 포함한 두 프로그램의 시간효율성은 문의 순서와 상관없이 동일하다고 판단하였다. 추가적으로 발견한 두 번째 유형은 동일한 문을 포함하더라도 연산을 수행하기 위한 문이 순서적으로 앞부분에 배치되는 것이 훨씬 빠르다고 판단하는 것으로 드러났다. 특히, 이 유형의 오개념은 8번

문항을 제외하고, 기본 가정인 첫 번째 유형의 오개념보다 오히려 높은 비율로 나타났다.

4.3 가설 3

세 번째 가정인 ‘변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.’ 라는 오개념을 확인하기 위한 3문항의 정답 및 오답 비율은 <그림 3>과 같다.



<그림 3> 가설 3의 정답률 비교

또한 응답자의 답안 선택에 관한 설명 분석을 통해 발견한 오개념 유형들은 <표 4>와 같다.

<표 4> 가설 3의 오개념 유형들

문항 번호	오개념 유형	빈도 (명)	비율 (%)
3	변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.	20	40.0
	문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	8	16.0
	전체	28	56.0
6	변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.	16	34.8
	더 짧은 프로그램의 시간효율성이 더 좋다.	1	2.2
	문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.	9	19.6
	전체	26	56.5
9	변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.	11	25.6
	같은 결과를 출력하면 시간효율성은 같다.	3	7.0
	코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.	14	32.6
	재귀함수를 사용한 프로그램이 for 문을 사용한 프로그램보다 시간효율성이 더 좋다.	4	9.3
	전체	32	74.4

3번 문항에서 제시한 두 프로그램의 경우, 모두 동일한 문을 활용하여 결과를 출력하지만, 첫 번째 프로그램은 추가적인 변수를 사용하여 비교에

필요한 연산을 반복문 바깥에서 실행함으로써, 반복문 안에서 매번 비교할 때마다 연산이 이루어지는 두 번째 프로그램에 비해 연산 횟수가 줄어들기 때문에 시간적으로 더 효율적이다. 그러나 학생들은 이를 인지하지 못하고, ‘변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.’(40.0%) 라는 오개념을 가장 많이 가지고 있는 것으로 드러났다. 추가적으로 드러난 오개념은 두 번째 가정과 일치하는 것으로 변수의 사용 여부와 상관없이 ‘문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.’(16.0%) 라고 판단하는 것으로 드러났다. 6번 문항 또한 3번 문항과 유사한 결과를 드러냈으며, 첫 번째 기본 가정인 ‘더 짧은 프로그램의 시간효율성이 더 좋다.’(2.2%) 라는 오개념을 추가적으로 확인하였다. 9번 문항의 경우, 세 번째 기본 가정으로 제시한 오개념 유형보다 ‘코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.’(32.6%) 라는 오개념이 가장 높은 비율로 드러났다. 이는 ‘재귀함수를 사용한 프로그램이 for 문을 사용한 프로그램보다 시간효율성이 더 좋다.’(9.3%)라는 오개념과 더불어 재귀함수를 통해 표현한 코드가 사용자 입장에서 보다 명확하게 이해하기 쉽다는 측면에서 더 효율적이라 판단한 것으로 보인다.

5. 결론

본 연구에서는 프로그래밍 교육을 이수한 전문계 고등학교 학생들을 대상으로 알고리즘 시간효율성에 관한 개념 이해 수준과 오개념을 조사하였다. 연구 결과 확인된 학습자들의 개념 이해 수준은 모든 측정 문항에서 낮게 나타났다. 또한, Gal-Ezer and Zur[16] 및 Özdener[18]의 연구에서 확인된 오개념 유형들이 본 연구에서도 마찬가지로 반복적으로 확인되었으며, 추가적으로 다른 유형의 오개념들이 확인되었다.

반복적으로 확인된 오개념들 중 ‘더 짧은 프로그램의 시간효율성이 더 좋다.’ 와 ‘변수의 수가 더 적은 프로그램의 시간효율성이 더 좋다.’ 라는 두 가지 유형의 오개념은 ‘A가 증가하면, B도 증가한다.’ 라는 직관적 사고 패턴에 따른 것으로 [17], 해당 오개념들은 이들을 확인하기 위해 개발

된 문항들 뿐 아니라, 다른 문항들에서도 반복적으로 드러나는 경향을 보였으며, 오개념 유형들 중 가장 높은 빈도로 나타났다. 이는 해당 오개념 유형이 학습자들에게 가장 보편화되어 있으며, 견고하여 쉽게 변화되지 않는다는 것을 의미한다. ‘문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.’와 ‘같은 결과를 출력하면 시간효율성은 같다.’ 라는 오개념들은 ‘A가 같으면, B도 같다.’ 라는 직관적 사고 패턴에 따른 것으로[17], 같은 내용을 포함하고 있다거나, 같은 결과를 도출한다거나, 같은 작업을 하는 두 프로그램의 실행시간은 같다고 판단하는 것이다. 이 오개념들은 위에서 제시한 유형의 오개념들보다 낮은 빈도로 나타났으며, 해당 오개념을 측정하기 위해 개발된 문항들의 경우, 다른 문항들에 비해 정답률이 높게 나타났다. 이는 해당 오개념 유형이 다른 유형에 비해 견고하지 않으며, 교육적 처치를 통해 비교적 개념 변화를 쉽게 일으킬 수 있다는 가능성을 의미한다.

추가적으로 발견한 오개념 유형 중 ‘코드 해석이 용이한 프로그램의 시간효율성이 더 좋다.(코드 길이가 길거나, 변수가 추가되더라도)’ 는 위에서 제시한 첫 번째 오개념 유형과 상반된 경향을 보인다. 이는 1번, 7번, 8번, 9번 문항에서 공통적으로 확인된 오개념으로 1번 문항의 경우, 동일한 연산을 수행하기 위한 두 반복문이지만, while 문이 for 문에 비해 코드가 더 길어짐에도 불구하고, 코드 해석과 이해가 용이하므로, 시간적으로 더 효율적이라고 판단하였으며(15.7%), 7번 문항에서는 max라는 변수가 추가적으로 사용된 프로그램이 해당 변수의 사용에 따라 코드 해석과 이해가 용이하므로 시간적으로 더 효율적이라고 판단하였다(6.5%). 8번 문항 또한 7번과 마찬가지로 적절한 변수명을 지닌 변수를 사용한 프로그램이 실행시간 측면에서 더 효율적이라고 판단한 경우가 확인되었다(2.1%).

추가적으로 발견한 또다른 오개념은 ‘연산문이 순서적으로 앞에 배치된 프로그램의 시간효율성이 더 좋다.’ 라는 것으로, 동일한 문을 포함한 두 프로그램이더라도, 학습자가 인지하기에 중요하다고 판단되는 연산문이 순서적으로 코드의 앞부분에 배치되어 먼저 실행되는 것이 시간적으로 더

효율적이라고 판단하는 것이다. 이 오개념은 위에서 제시한 두 번째 오개념 유형인 ‘문의 순서가 다르더라도 같은 문을 포함한 두 프로그램의 시간효율성은 같다.’와 ‘같은 결과를 출력하면 시간효율성은 같다.’에 상반되지만, 완전하게 과학적 개념으로 변환되지 못하고, 또 다른 오개념으로 확장된 것으로 보여진다. 즉, 같은 문을 포함하더라도 문의 배치 순서에 따라 시간효율성에 차이가 있을 것으로 인지하였지만, 코드 실행의 시간적 순서가 빠른 것과 실행시간의 양이 짧은 것을 구분하지 못한 것으로 판단된다. 이 오개념은 위에서 제시한 두 번째 오개념 유형을 확인하기 위해 개발된 문항들 모두에서 공통적으로 확인되었으며, 8번 문항을 제외하고 가장 높은 비율(2번 44.9%, 5번 29.2%)로 나타났다.

마지막으로, 본 연구를 통해 확인된 오개념 극복을 위한 제언을 제시하면 다음과 같다.

첫째, 알고리즘 효율성과 관련된 개념을 프로그래밍 교육 도입 단계에서 제공하거나, 선수 과목에 포함시킬 필요가 있다. 현행 전문계 고등학교 교육과정은 이러한 내용이 포함되어 있지 않으며, 이에 따라 학생들은 프로그래밍 교육과정을 이수하더라도 더 효율적인 프로그램이 무엇인지 판단하지 못하며, 본인이 지니고 있는 오개념들을 과학적 개념으로 변화시키지 못하는 결과를 초래하였다.

둘째, 발견된 오개념 유형에 따른 오개념 형성 예방 및 교정을 위한 교육 방법에 관한 연구가 필요하다.

본 연구에서는 전문계 고등학교 학습자들이 지닌 알고리즘 시간효율성에 관한 개념 이해 수준과 일반적인 오개념 유형들을 확인하였다. 이는 과학적 개념의 이해 증진을 위한 교육적 처치 및 교육과정 구성에 앞서 학습자들의 개념적 상태와 사고 패턴을 이해하기 위해 중요한 과정이다. 향후 연구를 통해 알고리즘 시간효율성 뿐 아니라 컴퓨터 과학의 기본적인 개념 이해 수준과 오개념 확인을 위한 연구가 지속적으로 이루어져야 할 것이다.

참 고 문 헌

- [1] 교육인적자원부 (2007). **교육인적자원부 고시 제2007-79호[별책21] 상업 정보 계열 전문 교과 교육과정**. 서울: 교육인적자원부.
- [2] 교육과학기술부 (2009). **교육과학기술부 고시 제2009-41호에 따른 고등학교 교육과정 해설 총론**. 서울: 교육과학기술부.
- [3] Hudak, M. A., & Anderson, D. E. (1990). Formal operations and learning style predict success in statistics and computer science courses. *Teaching of Psychology*, 17(4), 231-234.
- [4] 이은경 · 이영준 (2007). 알고리즘 설계와 분석에 관한 고등학교 학습자 오개념 분석. **교육과정평가연구**, 10(2), 329-348.
- [5] Gilbert, J. K., Osborne R., & Fensham P. (1982). Children's science and it's consequences for teaching. *Science Education*, 66(4), 623-633.
- [6] Gilbert, J. K., & Swift, D. J. (1985). Towards a lakatosian analysis of piagetian and alternative conceptions research programs. *Science Education*, 69(5), 681-696.
- [7] Kang, S., Scharmann, L. C., & Noh, T. (2004). Reexamining the role of cognitive conflict in science concept learning. *Research in Science Education*, 34(1), 71-96.
- [8] Bayman, P. & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Commun. ACM*, 26(9), 677-679.
- [9] Bonar, J. & Soloway, E. (1985). Pre-programming knowledge: a major source of misconceptions in novice programmers. *Human-Computer Interaction*, 1(2), 133-161.
- [10] Ma, L., Ferguson, J., Roper, M., & Wood, M. (2007). Investigating the viability of mental models held by novice programmers. In *Proc. of the 38th SIGCSE*.
- [11] Fleury, A. E. (1991). Parameter passing: the rules the students construct. In *Proceedings of the 22th SIGCSE*.
- [12] Holland, S., Griffiths, R., & Woodman, M. (1997). Avoiding Object misconceptions. In *Proceedings of the 28th SIGCSE*.
- [13] Almstrum, V. L., Henderson, P. B., Harvey, V., Heeren, C., Marion, W., Riedesel, C., Soh, L., & Tew, A. E. (2006). Concept inventories in computer science for the topic discrete mathematics. In *ACM SIGCSE Bulletin*, 38(4), 132-145.
- [14] Herman, G. L., Loui, M. C., & Zilles, C. (2010). Creating the Digital Logic Concept Inventory. In *Proceedings of the 41th ACM SIGCSE*.
- [15] Kaczmarczyk, L. C., Petrick, E. R., East, J. P., & Herman, G. L. (2010). Identifying Student Misconceptions of Programming. In *Proceedings of the 41th ACM SIGCSE*, 107-111.
- [16] Gal-Ezer, J., & Zur, E. (2004). The efficiency of algorithms: misconceptions. *Computers & Education*, 42(3), 215-226.
- [17] Stavy, R., & Tirosh, D. (2000). *How students (mis-)understand science and mathematics: intuitive rules*. New York: Teachers College Press.
- [18] Özdener, N. (2008). A comparison of the misconceptions about the time-efficiency of algorithms by various profiles of computer-programming students. *Computer & Education*, 51(3), 1094-1102.



이 은 경

1994 한국교원대학교
컴퓨터교육과(교육학학사)
2005 한국교원대학교
컴퓨터교육과(교육학석사)
2009 한국교원대학교 컴퓨터교육과(교육학박사)
2011~현재 연세대학교 교육대학원 특임교수
관심분야: 컴퓨터교육, 학습과학
E-Mail: angelsoph@yonsei.ac.kr