

A Performance Evaluation of Circuit Minimization Algorithms for Mentorship Education of Informatics Gifted Secondary Students

Hyung-Bong Lee[†] · Ki-Hyeon Kwon^{**}

ABSTRACT

This paper devises a performance improvement and evaluation process of circuit minimization algorithms for mentorship education of distinguished informatics gifted secondary students. In the process, students learn that there are several alternative equivalent circuits for a target function and recognize the necessity for formalized circuit minimization methods. Firstly, they come at the concept of circuit minimization principle from Karnaugh Map which is a manual methodology. Secondly, they explore Quine-McCluskey algorithm which is a computational methodology. Quine-McCluskey algorithm's time complexity is high because it uses set operations. To improve the performance of Quine-McCluskey algorithm, we encourage them to adopt a bit-wise data structure instead of integer array for sets. They will eventually see that the performance achievement is about 36%. The ultimate goal of the process is to enlarge gifted students' interest and integrated knowledge about computer science encompassing electronic switches, logic gates, logic circuits, programming languages, data structures and algorithms.

Keywords : Gifted Students, Mentorship Education, Circuit Minimization, Karnaugh Map, Quine-McCluskey Algorithm

중등 정보과학 영재 사사 교육을 위한 회로 최소화 알고리즘 성능 평가

이 형 봉[†] · 권 기 현^{**}

요 약

이 연구에서는 중등 심화 과정을 마치고 사사 과정에 진입한 최우수 정보과학 영재 교육을 위한 회로 최소화 알고리즘의 성능 개선 및 평가 과정을 보인다. 이 과정에서 학생들은 원하는 목표 기능을 얻기 위한 논리 회로는 꼭 한 가지가 아니고 다양하게 구성할 수 있다는 점과 이들 중 가장 간단한 회로를 찾을 수 있는 방법의 필요성을 인식하게 된다. 수작업으로 이루어지는 카르노 맵에서 회로 최소화를 위한 기본 원리를 터득하고, 그 과정을 소프트웨어로 수행하는 Quine-McCluskey 알고리즘을 탐구한다. Quine-McCluskey 알고리즘은 기본적으로 집합 연산의 반복에 의해 중복성을 도출하고 축약하는 과정을 반복한다. 집합 연산은 두 집합을 구성하는 원소들에 대한 비교 연산으로 이루어지므로 복잡도가 높다. 이를 해결하는 방법으로 원소 나열식 집합을 비트 정보로 표현하는 방안을 모색하고, 그 결과 약 36%의 성능 향상이 이루어짐을 보게 된다. 이 과정의 궁극적 목표는 영재 학생들이 전자 스위치, 논리 게이트, 논리 회로, 프로그래밍 언어, 데이터 구조, 알고리즘 등을 포괄하는 컴퓨터과학 학문에 대한 흥미와 지식 통합적 안목을 기르는 데 있다.

키워드 : 영재, 사사 교육, 회로 최소화, 카르노 맵, Quine-McCluskey 알고리즘

1. 서 론

2002년 영재교육이 시작된 이후, 우리나라 영재교육은 양적·질적 구분 없이 크게 성장하고 있다. 특히, 양적인 측면에서는 초·중·고생 전체의 1%에 가까운 100,000명 이상이

영재교육에 참여하고 있다[1]. 그러나 이와 같은 영재교육 대상자의 양적 확대는 영재 교육 수준 향상의 걸림돌로 이어져 정보 영재의 경우 약 30% 학생들이 교육 내용에 대하여 너무 쉽다는 의견을 보이고 있다[2]. 즉, 다수의 학생 교육에는 보편적 교재 활용이 용이하므로 대부분의 정보 영재 교육이 대학 교재의 단편적 발췌 인용 수준으로 이루어지고 있다는 것이다[3]. 이런 지적에 따라 정보 영재교육의 질적 제고를 위한 특화된 교수 학습자료[4-5], 교수 학습 방법론[6], 교육과정[7]에 대한 연구들이 활발하게 이루어지고 있다. 한편, 정부에서는 2013-2017 기간의 제3차 영재교육진흥

[†] 종신회원 : 강릉원주대학교 컴퓨터공학과 교수
^{**} 정 회 원 : 강원대학교 전자정보통신공학부 교수
Manuscript Received : September 2, 2015
First Revision : November 27, 2015
Accepted : November 29, 2015
* Corresponding Author : Ki-Hyeon Kwon(kweon@kangwon.ac.kr)

종합계획[8]에서 기초, 심화, 사사 과정 등 수준별 영재교육 과정 및 학생 개인별 맞춤형 영재교육과정을 개발하는 등의 영재교육의 질적 향상을 강조하고 있다.

이 연구에서는 중등 심화 과정을 마치고 사사 과정에 진입한 최우수 정보과학 영재들의 맞춤형 교육을 위한 주제 통합적 교수 학습 소재[9]를 제안한다. 이 교수 학습 소재의 목적은 기초와 심화 과정에서 논리 회로, 프로그래밍 언어, 데이터 구조, 알고리즘 등 정보과학 전공 요소를 단편적으로 접했던 정보과학 영재 학생들이 이들 요소들의 유기적인 연관 관계 및 상호 파급 효과를 이해함으로써 계산학적 문제의 창의적 해결 능력과 기존 문제들에 대한 개선점을 발견할 수 있는 날카로운 비판적 사고를 함양할 수 있도록 하는 데 있고, 교수자들이 2-4장 내용을 골격으로 하는 스토리텔링을 구성하여 약 30시간 전후 분량의 사사 과정 세부 교안을 작성하는 데 적절할 것으로 판단된다. 이 논문의 2장은 전자 스위치, 논리 게이트, 회로 설계, 회로 최소화 등의 개념 및 필요성에 대하여 논하고, 3장과 4장에서는 기존 회로 최소화 알고리즘에 대한 개선 방안 도출 과정과 개선된 알고리즘에 대한 실험 및 평가 과정을 각각 보인다. 그리고 마지막 5장의 결론으로 논문을 맺는다.

2. 회로 설계 및 최소화

2.1 회로 설계

1) 전자 스위치

일반적인 스위치는 어떤 기계 장치에 물리적 힘을 가하여 해당 기계의 기능을 개폐하는 장치를 말한다. 이에 반하여 전자 스위치는 물리적 힘이 아닌 순수한 전류를 사용해서 다른 전류의 흐름을 개폐할 수 있는 소자를 일컫는다[4]. 이러한 전자 스위치가 중요한 이유는 이들이 모든 전자 회로의 기본이 되는 AND, OR 등의 논리 게이트 제조의 핵심 기술이 되기 때문이다. 최초의 전자 스위치는 진공 상태에서 한쪽 도체에 전류를 보내면 다른 쪽 도체에도 전류가 흐르고 그렇지 않으면 차단되는 성질을 이용하여 1906년 완성된 진공관이었다[10]. 두 번째 전자 스위치는 게르마늄 등 반도체 물질을 세 층으로 쌓은 후 가운데 부분에 전원을 인가하면 양쪽 반도체 사이에 전류가 흐르고 그렇지 않으면 차단되는 현상에 착안하여 1947년 발명된 트랜지스터[11]이다. 최근의 IT 기기는 1958년 개발된 트랜지스터의 대규모 집적 기술에 의해 생산된 IC 소자를 활용한다.

2) 전자 논리 게이트와 전자 회로

전자 논리 게이트는 전자 스위치 소자를 이용하여 두 개의 입력 전원에 대하여 AND, OR, XOR, NOT 등의 논리 연산 결과를 출력한다. 이들 논리 게이트를 조합하면 미리 결정된 입력 상황별 대응 결과를 자동으로 계산하여 출력하는 이른바 전자 회로를 만들 수 있다. 이를테면 비트 단위 가산기를 생각하면 Fig. 1과 같이 더하려는 두 비트 x, y 와 하위에서 올라오는 올림 비트 c 가 입력이고, 이에 대하여 덧

셈 결과 S 와 다음 자리로의 올림 비트 C 가 출력인데, 입력의 모든 조합에 대하여 각각의 올바른 결과를 출력하도록 논리 게이트를 조합한 것이 가산기 회로가 된다. 이를 반대로 말하면 입력 및 출력의 수에 관계없이 입력 조합별 출력 결과가 정의되면 이를 자동으로 계산하는 회로를 만들 수 있다는 것이다.

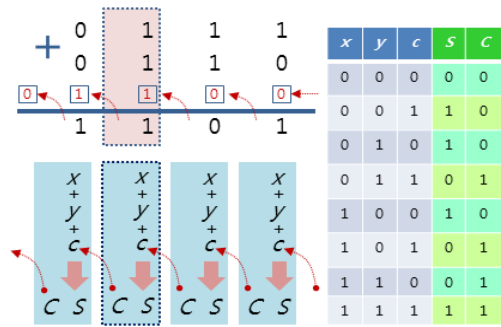


Fig. 1. Concept of the Bit-wise Adder

3) 전자 회로 설계

전자 회로를 설계하기 위해서는 위 Fig. 1과 같이 각각의 주어진 입력 조합에 대한 출력 결과를 정의하는 표를 필요로 하는데 이를 진리표라 한다. 작성된 진리표로부터 각 출력에 대하여 그들이 0 혹은 1이 되는 조건을 찾아 이를 논리 게이트의 연결로 표현하는 과정을 전자 회로 설계라 부르는데, 보통은 1이 되는 경우를 중심으로 진행한다(취합된 조건 이외의 경우는 모두 0이 됨). 회로 설계의 가장 쉬운 방법은 1이 되는 입력 조건 항목을 있는 그대로 모두 나열하는 것이다. 예를 들면 Fig. 1에서 C 가 1이 되는 경우 네 가지의 부울 논리식은 아래 Equation (1)과 같다.

$$C = x'yc + xy'c + xyc' + xyc \tag{1}$$

① ② ③ ④

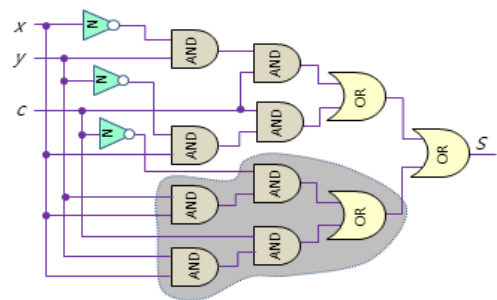


Fig. 2. A Raw Circuit Diagram for Expression (1)

위 Equation (1)에서 $x'yc$ 는 “ x 는 0이고 y 와 c 는 1일 때”를 의미하는 것으로 세 조건에 대한 AND(곱셈) 연산 결과임을 의미한다. 그리고 $x'yc + xy'c$ 는 “ $x'yc$ 이거나 $xy'c$ 중 어느 한쪽 경우라도 만족할 때”라는 뜻으로 양쪽 결과에 대한 OR(덧셈) 연산을 의미한다. 이를 논리 게이트의 연결, 즉 회로로 표현하면 위 Fig. 2와 같다.

2.2 회로 최소화

Equation (1)과 이에 대한 회로도 Fig. 2를 자세히 관찰해 보면 $xyz + xyz$ (③+④)에 해당되는 부분은 보다 짧게 단축이 가능함을 알 수 있다. 즉, 두 항의 결과는 c 와 관계없이 양쪽 공통 조건인 xy 에만 의존하고 있음을 알 수 있는데 이에 대한 부울 대수 증명은 Equation (2)와 같다.

$$xyz' + xyz = xy(c' + c) = xy1 = xy \quad (2)$$

이런 성질은 $x'yz + xyz$ (①+④)와 $xy'z + xyz$ (②+④)에도 존재한다. 이와 같이 기초 회로를 축소시키는 과정을 회로 최소화라 한다. 회로 최소화는 위의 예에서와 같이 직관적으로 이루어질 수도 있으나 회로가 복잡할 때에는 한계가 있을 뿐 아니라 오류 가능성이 높기 때문에 보다 체계적인 방법론이 필요하다. 이런 방법론 중의 하나가 카르노 맵(Karnaugh map)[12]인데, Fig. 3과 같이 부울 논리식을 테이블로 그려 중복성 발견의 용이성을 제공한다. Fig. 3의 카르노 맵에 의한 최소화된 부울 논리식은 Equation (3)과 같다.

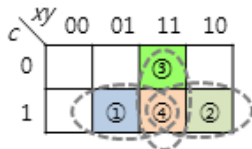


Fig. 3. Karnaugh Map for Logic Expression (1)

$$C = xc + yc + xy \quad (3)$$

Equation (1)~(3)과 같이 논리곱 항들의 합으로 표현된 부울 논리식을 SOP(Sum Of Products)라 하고, 각각의 논리곱 항들을 최소항(minterm)이라 한다. 따라서 회로 최소화는 가급적 길이가 짧고 적은 수의 최소항들로 구성된 SOP를 찾는 과정이라 말할 수 있다.

2.3 Quine-McCluskey 알고리즘

Quine-McCluskey 알고리즘(이하 QM 알고리즘이라 칭함)[13-14]은 우선 카르노 맵 방법에 스며있는 공통 묶음 추출 개념을 비트 패턴에 적용하여 두 개 이상의 최소항을 축약하여 보다 짧은 하나의 최소항으로 유도한다. 이와 같이 축약되어 유도된 최소항을 내포항(implicant)이라 하고, 이러한 내포항들은 다음 단계의 내포항으로 더 짧게 축약될 수 있다. 편의상 Equation (1) 형태의 기초 최소항들을 1차 내포항, 이들로부터 단계별로 유도된 내포항들을 2~n차 내포항이라 부른다. 이들 내포항들 중, n차 내포항 유도에 관여하지 않은 것들을 기본 내포항(prime implicant)이라 한다. 위와 같이 기본 내포항들이 취합되면 이들 중 중복된 것들을 제거하고 최종적으로 꼭 필요한 내포항들만 선택하는데 이러한 내포항들을 필수 내포항(essential prime implicant)이라 부른다.

1) 내포항의 유도(Step 1)

내포항을 유도하는 과정은 다음과 같다.

- Equation (1) 형태의 부울 식을 아래 Equation (4)와 같이 부울 함수 형태로 표현한다. 이때, x 는 22, y 는 21, c 는 20으로 대응시켜 각각의 최소항을 3비트 정수로 대응시킨다. 즉, $x'yz$ 는 3(0112), $xy'z$ 는 5(1012), xyz 는 6(1102), xyz 는 7(1112)로 각각 표현된다.

$$F_c(x, y, c) = \sum m(3, 5, 6, 7) \quad (4)$$

- 비트 패턴으로 대응된 최소항(1차 내포항)들 사이에 오직 하나의 비트만 다른 모든 쌍을 찾아 내포항을 도출한다. 이를테면 Fig. 4에서 최소항(1차 내포항) $i(3)$ 과 $i(7)$ 은 $2^2(x)$ 만 다르고 $2^1(y)$, $2^0(c)$ 는 같은데, 그 의미는 두 최소항의 합 $x'yz + xyz$ 의 결과는 x 와는 무관하게 yz 조건만 만족하면 된다는 것이다. 이와 같이 내포항 $i(3,7)$ 이 유도되면서 비관여로 제외되는 2^2 비트를 와일드 비트라 부르기로 한다.

최소항	$x y c$	최소항	$x y c$	최소항	$x y c$
$i(3)$	0 1 1	$i(5)$	1 0 1	$i(6)$	1 1 0
$i(7)$	1 1 1	$i(7)$	1 1 1	$i(7)$	1 1 1
$i(3,7)$	* 1 1	$i(5,7)$	1 * 1	$i(6,7)$	1 1 *

Fig. 4. Reduction Process for Implicants

- 추출된 내포항들 사이에 와일드 비트 부분이 동일하면서 나머지 비트들 중 오직 하나의 비트만 다른 쌍을 찾아 위와 동일한 방법으로 다음 단계의 내포항을 유도한다.
- 위 과정을 더 이상의 내포항이 유도되지 않거나 최대 n (총 비트 수 = 총 입력 변수 수) - 1개의 와일드 비트 까지 반복한다.
- 내포항 유도에 직접 관여한 이전 단계의 내포항 쌍과, 내포항 쌍은 아니었다라도 상위 내포항의 부분집합인 내포항들을 폐기하고, 최종 남아있는 내포항들만 기본 내포항으로 선택한다.
- QM 알고리즘을 위한 데이터 구조 고안에 대한 실마리를 얻기 위해 Table 1에 다단계 내포항이 포함된 부울 함수 (5)에 대한 전제 진행 과정을 Table 1에 보였고, 축약된 결과는 부울 함수 Equation (6)과 같다.

$$F(A,B,C,D) = \sum m(4,8,9,10,11,12,14,15) \quad (5)$$

$$F(A,B,C,D) = BC'D' + AB' + AD' + AC \quad (6)$$

Table 1. Prime Implicants for Boolean Function (5)

Number of bits with 1	1st implicants	2nd implicants	3rd implicants
1	i(4) 0100	i(4,12) *100	
	i(8) 1000	i(8,9) 100* i(8,10) 10*0 i(8,12) 1*00	i(8,9,10,11) 10** i(8,10,12,14) 1**0
2	i(9) 1001	i(9,11) 10*1	i(10,11,14,15) 1*1*
	i(10) 1010	i(10,11) 101*	
	i(12) 1100	i(10,14) 1*10	
3	i(11) 1011	i(12,14) 11*0	
	i(14) 1110	i(11,15) 1*11	
4	i(15) 1111	i(14,15) 111*	

2) 필수 내포항의 도출(Step 2)

도출된 기본 내포항들 모두가 꼭 필요한지, 즉 다른 내포항들로 인해 특정 내포항이 불필요한 경우가 있는지를 조사하여 제거한다. 이를 체크하는 간단한 방법으로 Table 2의 내포 차트를 활용한다. 이 테이블로부터 부울 함수의 기초 최소항들을 모두 커버하기 위한 최소한의 기본 내포항만을 선택할 수 있다. 그 구체적인 과정은 아래와 같다.

Table 2. Prime Implicants Chart

기본 내포항	기초 최소항								커버 수	선택
	4	8	9	10	11	12	14	15		
i(4,12)	o					o			2	✓
i(8,9,10,11)		o	o	o	o				4	
i(8,10,12,14)		o		o		o	o		4	
i(10,11,14,15)				o	o		o	o	4	
내포 빈도	1	2	1	3	2	2	2	1		



i(8,9,10,11)		o	o	o	o				4	✓
i(8,10,12,14)		o		o			o		3	
i(10,11,14,15)				o	o		o	o	4	
내포 빈도	x	2	1	3	2	x	2	1		



i(8,10,12,14)							o		1	
i(10,11,14,15)							o	o	2	✓
내포 빈도	x	x	x	x	x	x	2	1		



내포 빈도	x	x	x	x	x	x	x	x		
-------	---	---	---	---	---	---	---	---	--	--

- 내포 빈도가 1인 기초 최소항들이 있다면 이들 각각을 커버하는 내포항이 한 개씩 존재하므로 이런 내포항들을 필수 내포항으로 선택하고 내포 차트에서 제거한다.
- 내포 빈도가 1인 기초 최소항이 더 이상 존재하지 않으면 시행착오를 거쳐 적절한 기초 내포항을 필수 내포항으로 선택하여 내포 차트에서 제거한다.
- 필수 내포항으로 선택하여 제거할 때는 남아있는 모든 기본 내포항에서 해당 내포항이 커버했던 기초 최소항

들에 대한 커버리지를 제거하고 관련 내포 빈도를 업데이트 한다.

- 내포 차트에 남아있는 내포항이 존재하는 동안 위의 과정을 반복한다.
- Table 2의 진행 결과 부울 함수 Equation (6)은 기본 내포항 중 AD'가 생략되고 세 개의 필수 내포항으로 구성된 부울 함수 Equation (7)과 같이 최소화된다.

$$F(A,B,C,D) = BC'D' + AB' + AC \quad (7)$$

3. Quine-McCluskey(QM) 알고리즘 개선

3.1 QM 알고리즘 개선에 관한 최근 연구

여기서는 QM 알고리즘에 대한 가장 최근 연구인 [15]의 제안 내용을 조명하고 분석한다.

1) 기본 내포항의 유도(Step 1)

기존 QM 알고리즘이 기초 최소항들로부터 n차 기초 내포항까지 단계적으로 유도하는 상향식이어서 계산량이 클 것이라는 점에 주목하여, 조합 가능한 모든 기초 내포항들을 미리 도출하여 후보 테이블로 준비하고, 이들 중 주어진 최소항을 내포하는지를 검색하여 기본 내포항을 도출하는 하향식(Top Down) 방식을 제안하였다(이하 TD 알고리즘이라 칭함). 이는 기본적으로 모든 후보 내포항에 대하여 기초 최소항들로 구성된 집합의 부분집합인지를 판별하는 과정으로 이루어진다. 이를테면 Table 3에서 입력변수가 4개인 부울 함수 Equation (5)의 부분집합이면서 서로 간 포함 관계에 의한 중복이 없는 후보 내포항들이 음영 처리되어있다. 그러나 이 방안은 검색대상인 후보 내포항 테이블의 크기 대비 도출되는 기본 내포항 수의 비가 커야지만 효과적인 것이라는 점을 예측할 수 있다. Table 4와 Equation 5에 입력 변수 수에 따른 후보 테이블의 크기 변화 추이를 보였는데, 후보 테이블 크기의 증가 속도가 높다.

Table 3. Candidate Prime Implicants for 4 Inputs

단계	후보 내포항
1차 (16개)	i(0) i(1) i(2) i(3) i(4) i(5) i(6) i(7) i(8) i(9) i(10) i(11) i(12) i(13) i(14) i(15)
2차 (32개)	i(0,1) i(0,2) i(0,4) i(0,8) i(1,3) i(1,5) i(1,9) i(2,3) i(2,6) i(2,10) i(3,7) i(3,11) i(4,5) i(4,6) i(4,12) i(5,7) i(5,13) i(6,7) i(6,14) i(7,15) i(8,9) i(8,10) i(8,12) i(9,11) i(9,13) i(10,11) i(10,14) i(11,15) i(12,13) i(12,14) i(13,15) i(14,15)
3차 (24개)	i(0,1,2,3) i(0,4,8,12) i(0,1,4,5) i(0,2,4,6) i(0,1,8,9) i(0,2,8,10) i(1,3,5,7) i(1,3,9,11) i(1,5,9,13) i(2,3,10,11) i(2,6,10,14) i(2,3,6,7) i(3,7,11,15) i(4,5,6,7) i(4,5,12,13) i(4,6,12,14) i(5,7,13,15) i(6,7,14,15) i(8,9,10,11) i(8,9,12,13) i(8,10,12,14) i(9,11,13,15) i(10,11,14,15) i(12,13,14,15)
4차 (8개)	i(0,1,2,3,4,5,6,7) i(0,1,4,5,8,9,12,13) i(0,2,4,6,8,10,12,14) i(0,1,2,3,8,9,10,11) i(1,3,5,7,9,11,13,15) i(2,3,6,7,10,11,14,15) i(4,5,6,7,12,13,14,15) i(8,9,10,11,12,13,14,15)

Table 4. Trend of Candidate Prime Implicants Number

입력 변수	단계별 후보 내포항							합계
	1차	2차	3차	4차	5차	6차	7차	
3개	8	12	6	-	-	-	-	26
4개	16	32	24	8	-	-	-	80
5개	32	80	80	40	10	-	-	242
6개	64	192	240	160	60	12	-	728
7개	128	448	672	560	280	84	14	2186

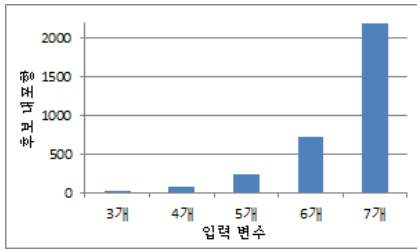


Fig. 5. Graphic Chart for Table 4

2) 필수 내포항의 도출(Step 2)

TD 알고리즘에서는 시행착오를 거쳐 내포 차트로부터 필수 내포항을 도출하는 QM 방법론을 자동화시키기 위해 아래의 필수 내포항 도출 알고리즘을 제안하였다. 여기서 F 는 원래 주어진 부울 함수의 기초 최소항 집합, P_i 는 현재 내포 차트에 있는 임의의 기본 내포항, C 는 필수 내포항 집합, f_m 은 내포 차트에 있는 기초 최소항 m 의 내포 빈도를 각각 의미한다.

SelectProcess()

- 1-1) $f_m=0$ 인 m 을 C 에 추가하고 F 에서 제거한다.
- 1-2) $f_m=1$ 인 m 에 대하여 $m \in P_i$ 인 P_i 를 C 에 추가하고, $x \in P_i$ 인 모든 x 를 F 와 $x \in P_i(\neq j)$ 인 P_j 에서 제거하며, P_i 를 내포 차트에서 제거한다.
- 1-3) $F \neq \emptyset$ 이고 $f_m=1$ 인 m 이 더 이상 존재하지 않으면 Including Process()를 수행한다.

IncludingProcess()

- (1) 모든 π 에 대해서 $|P_i|$ 가 동일하지 않은 경우
 - 2-1) $P_i \subset P_j(\neq j)$ 관계에 있는 P_i 를 내포 차트에서 제거하는 과정을 더 이상의 P_j 가 발견되지 않거나 $f_m=1$ 인 m 이 나타날 때까지 지속한다.
 - 2-2) $f_m=1$ 인 m 이 존재하면 1-2) 내용을 처리하고, 그렇지 않고 $|P_i|=1$ 인 P_j 가 다수 존재하면 이들 중 임의의 P_j 를 선택하여 C 에 추가하고 내포 차트에서 제거하는 내용을 처리한다.
 - 2-3) $F=\emptyset$ 이면 종료한다.
- (2) 모든 π 에 대해서 $|P_i|$ 가 동일한 경우
 - 3-1) 임의의 P_j 를 선택하여 C 에 추가하고 내포 차트에서 제거하는 내용을 처리한다.
 - 3-2) $f_m=1$ 인 m 이 존재하면 1-2) 내용을 처리하고, 그렇지 않으면 $|P_i|$ 가 최대인 P_j 를 C 에 추가하고 내포 차트에서 제거하는 내용을 처리한다. 이 과정을 $F \neq \emptyset$ 인 동안 반복한다.

앞의 TD 알고리즘은 전체적으로 경우의 수가 불필요하게 너무 세분화되어 단순화될 수 있는 여지가 있고, 3-1 스텝에서는 임의의 P_j 대신 좀더 나은 선택 기준이 존재할 가능성을 남겨두고 있다.

3.2 QM 알고리즘에 대한 새로운 개선

1) 집합 연산을 위한 효과적인 데이터 구조(Step 1)

QM 알고리즘 구현을 위한 주요 기초 프로시저는 두 집합에 대한 합집합(내포항) 구하기(qm_union_set()), 동일한 집합인지 체크하기(qm_is_equal()), 축약(내포항 도출) 가능한 관계인지 체크하기(qm_is_reductable()) 등이고, TD 알고리즘에서는 부분집합 관계인지 체크하기(td_is_subset()) 등이다. 이들 프로시저를 일반적인 원소 열거 기반 집합 저장 구조로 구현한 코드를 Fig. 6에 보였다. 이 그림으로부터 프로시저의 복잡도가 qm_union()과 qm_is_reductable()은 $O(n)$ 이고, qm_is_eqla()과 td_is_subset()은 $O(n^2)$ 임을 알 수 있다.

```

typedef struct set_array {
    int e[MAX_ELEMENTS]; // array of elements
    int n; // number of elements
} SET_ARR;

void qm_union(SET_ARR *d, SET_ARR *s)
{
    int i;
    for (i = 0; i < s->n; i++)
        d->e[d->n++] = s->e[s->n + i];
}

int qm_is_equal(SET_ARR *s1, SET_ARR *s2)
{
    int i, j;
    if (s1->n != s2->n) return 0; // false
    for (i = 0; i < s1->n; i++) {
        for (j = 0; j < s2->n; j++)
            if (s1->e[i] == s2->e[j]) break;
        if (j >= s2->n) return 0; // false
    }
    return 1; // true
}

int qm_is_reductable(SET_ARR *s1, SET_ARR *s2)
{
    int i, e1, e2, msk1, msk2;
    if (s1->n != s2->n) return 0; // false
    e1 = s1->e[0], e2 = s2->e[0];
    for (i = 1; msk1 = msk2 = 0; i < s1->n; i++)
        msk1 |= e1^s1->e[i], msk2 |= e2^s2->e[i];
    if (msk1 == msk2) {
        e1 &= ~msk1, e2 &= ~msk2;
        if (n_of_bit_with1(e1 ^ e2) == 1)
            return 1; // true
    }
    return 0; // false
}

int td_is_subset(SET_ARR *p, SET_ARR *c)
{
    int i, j;
    if (p->n <= c->n) return 0; // false
    for (i = 0; i < c->n; i++) {
        for (j = 0; j < p->n; j++)
            if (c->e[i] == p->e[j]) break;
        if (j >= p->n) return 0; // false
    }
    return 1; // true
}

int n_of_bit_with1(int m)
{
    int i, cnt;
    for (i = cnt = 0; i < MAX_BITS; i++, v >>= 1)
        if (m & 0x01) cnt++;
    return cnt;
}
    
```

Fig 6. Implemented C Code Using General Set Structure

그런데 집합 원소의 특성이 한정된 범위의 정수라는 점에 착안하여 집합을 비트 정보로 표시할 수 있다. 이를테면 부울 함수 입력 변수가 6개이면 원소들이 0~63(=2⁶-1) 범위의 정수이기 때문에 이들은 C 언어의 64비트 long 타입 비트로 대응될 수 있다. 입력 변수가 더 많은 경우에는 여러 개의 long 타입으로 수용 가능하다. 이 제안을 BMS(Bit Map Set)이라 칭하고 TD 알고리즘에 적용한다. 더 나아가 QM 알고리즘에서 내포항에 적용되어 있는 와일드 비트 패턴을 관리하면 알고리즘이 더욱 간단해진다. 예를 들어 내포항 i(8, 9)는 두 원소 100₂와 100₂에 와일드 비트 0001₂가 적용되어 100*₂를 의미하므로 두 원소 중 하나와 와일드 비트 등 두 가지만 유지하더라도 나중에 8과 9를 복원해낼 수 있다. 마찬가지로 i(10, 11)은 101*₂를 의미하고 와일드 비트는 0001₂이다. 이들 두 내포항이 축약 가능하기 위해서는 와일드 비트 패턴이 일치하고 와일드 부분을 제외한 나머지 부분에서 오직 한 자리의 비트만 다르다면 된다. 또한, n_of_bit_withl() 프

```

typedef struct set_bit {
    int e, w; // first element & wild bits
    int n; // number of implied elements
} SET_BIT;

void qm_union(SET_BIT *d, SET_NEW *s, int w)
{
    d->w = w, d->n *= 2;
}

int qm_is_equal(SET_BIT *s1, SET_NEW *s2)
{
    if (s1->w == s2->w &&
        s1->e & ~s1->w == s2->e & ~s2->w)
        return 1; // true
    return 0; // false
}

int qm_is_reductable(SET_BIT *s1, SET_NEW *s2)
{
    int w;
    if (s1->w == s2->w) {
        w = s1->e & ~s1->w ^ s2->e & ~s2->w;
        if (n_of_bit_withl(w) == 1)
            return s->w | w; // true, return new wild
    }
    return 0; // false
}

int td_is_subset(SET_BIT *p, SET_NEW *c)
{
    if (p->n >= c->n &&
        (p->e | c->e) == p->e) return 1; // true
    return 0; // false
}

int Nof1[] = {0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4, /*~0x0f */
              1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, /*~0x1f */
              1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, /*~0x2f */
              2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, /*~0x3f */
              1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, /*~0x4f */
              2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, /*~0x5f */
              2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, /*~0x6f */
              3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, /*~0x7f */};

int n_of_one_mask(int m)
{
    return(Nof1[m]);
}
    
```

Fig. 7. Implemented C Code Using the Proposed Bit Based Set Structure

로시저는 0~2ⁿ-1(n은 입력 변수 수)의 확정된 범위의 정수에 대해 연산하므로 비트 체크 대신 테이블 색인(Table Look-up) 방식을 도입하면 복잡도를 낮출 수 있다. 이런 제안들을 WMS(Wild Map Set)이라 칭하고 QM 알고리즘에 적용한다. 이러한 개선점을 반영한 프로시저를 Fig. 7에 보였는데 복잡도가 모두 O(1)로 완화되었음을 볼 수 있다.

2) 필수 내포항 도출 알고리즘 보완(Step 2)

현재의 내포 차트 상에서 내포 빈도가 1인 기초 최소항을 포함하는 기본 내포항을 모두 선택하여 제거하고 냈을 때, 부분집합 관계에 있는 내포항들이 나타날 수 있고 이들을 처리하면 다시 내포 빈도가 1인 기초 최소항이 나타날 수 있다. 즉, 부분집합 관계의 내포항들이 존재하지 않을 때만 비교 우위인 최선의 내포항을 선택한다. TD 알고리즘의 복잡성을 제거하고 단순화시킨 결과를 Fig. 8에 보였다. 또한, 최선의 기본 내포항을 선택함에 있어서 TD 알고리즘에서는 기초 최소항에 대한 커버 수가 동일할 때는 임의의 것을 선택한다고 했으나 가급적 내포 차수가 높은 것을 선택하는 것이 실제 회로 최소화에 도움이 된다. 예를 들어 Table 5의 내포 차트에서 TD 알고리즘에 의하면 기본 내포항 ①과 ② 중 어느 것을 선택해도 좋지만, 논리 게이트 수를 고려하면 내포 차수가 높은 ①을 선택하도록 개선되어야 한다.

```

while (F != ∅)
{
    if (n_of_minterms_with_frequency1() > 0)
        select_all_prime_implicants_of_the_minterms();
    if (n_of_subset_prime_implicants() > 0)
        remove_all_subset_prime_implicants();
    else
        select_best_prime_implicants();
}
    
```

Fig. 8. Proposed Algorithm for Essential Prime Implicants

Table 5. An Example for the Best Selection Procedure

기본 내포항	기초 최소항									커버 수	선택
	0	2	3	4	5	6	7	8	9		
i(0,2,4,6) ... ①	o									1	✓
i(0,8) ②	o									1	
내포 빈도	2	x	x	x	x	x	x	x	x		

4. Quine-McCluskey(QM) 알고리즘 실험 및 평가

4.1 실험 방법

1) 실험 데이터

회로 입력 변수가 n개이면 부울 함수에 대한 서로 다른 입력 조합의 총수는 2ⁿ 가지이다. 이 연구에서는 n이 4, 5, 6 일 때 각각에 대하여 난수로 생성된 65,536가지의 데이터를 처리하되, n이 4일 때는 모든 경우를 포함하도록 한다.

2) 알고리즘 명명

“QM”은 Quine-McCluskey 기본 알고리즘을, “TD”는 [15]

에서 QM을 개선한 Top-Down 알고리즘을, “ARR”는 기존의 일반적인 원소 열거형 집합 자료구조를, “BMP”와 “WMS”는 비트 및 와일드 기반 집합 자료구조 등 이 논문에서 제안된 내용을 각각 의미하는 것으로 한다.

4.2 실험 결과 및 분석

1) Step 1(기본 내포항 유도) 성능

Table 6에 알고리즘별 Step 1의 기본 내포항 유도 성능을, Table 7에는 65,536가지 각 경우에 대한 성능 순위별 빈도를 비교해 보였는데, 이로부터 아래의 두 가지 사실을 추론할 수 있다.

- [15]의 TD 개선은 기존 QM보다 우수하지 못하다.
- 본 연구의 비트 기반 와일드 방법론(WMS)이 다른 어떤 방법론보다 우수하고, 입력 변수의 수가 늘어날수록 그 격차는 더 커진다(실험 범위에서 최대 36% 성능 향상).

2) Step 2(필수 내포항 도출) 성능

Step 2의 필수 내포항 유도를 위한 알고리즘별 시간적 성능과 정확도를 Table 8에 비교해 보았다. 이 표에서 오류율은 전체 실험 데이터 대비 상대 쪽 알고리즘보다 미흡한 최소화 결과를 얻은 빈도의 비율을 의미한다. 이를테면 아래의 부울 함수 Equation (8)에 대한 TD_ARR과 TD_BMS의 축약 결과는 각각 부울 함수 Equation (9)와 (10)인데 TD_ARR의 결과가 논리 게이트 하나를 더 필요로 하므로 미흡하다.

$$F(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 9) \quad (8)$$

$$= A'C + A'B + B'C'D' + AB'C \quad (9)$$

$$= A'D + A'C + A'B + AB'C \quad (10)$$

Table 8로부터 시간과 정확도 측면에서 충분히 유의할 만한 개선이 이루어졌음을 확인할 수 있다.

5. 결론

이 연구에서는 전자 스위치, 논리 게이트, 논리 회로 설계 및 회로 최소화, 프로그래밍 언어, 데이터 구조, 알고리즘 등 컴퓨터의 하드웨어와 소프트웨어를 아우르는 정보과학 영재 사사 교육을 위한 주제 통합적 교수 학습 소재를 제안하였다. 특히, 알고리즘 못지않게 알고리즘을 뒷받침하는 창조적 데이터 구조 착안의 중요성 인식을 위해 논리 회로와 관계 있는 Quine-McCluskey 알고리즘을 발굴하여 성능 개선점을 발견하고, 최대 36% 성능 향상을 이루는 과정을 보임으로써 영재들이 비판적 사고에 기반한 창의적 정보과학 문제 해결 능력을 배양함은 물론 개선 결과의 성취감에 따라 학문에 대한 흥미를 북돋울 수 있도록 하였다.

제안된 교수 학습 소재는 교수자의 개성이나 철학에 따라 Table 9의 예와 같이 3시간 단위의 10차시 전후의 스토리텔링으로 구성하여 프로젝트형 수업을 진행한다면 정보과학 영재의 한 학기 사사 교육 과정 운영에 실질적인 기여가 클 것으로 기대하는 바이다.

Table 6. Step1 Performance of the 4 Algorithms(unit:s)

입력변수 크기(n)	QM		TD	
	ARR	WMS	ARR	BMS
4	2.35	2.22	2.98	2.29
5	5.48	4.87	8.51	5.47
6	18.01	11.61	29.04	15.73

Table 7. Step1 Rank Frequency of the 4 Algorithms

변수(n)	순위	QM		TD	
		ARR	WMS	ARR	BMS
4	1	28,712	57,925	73	22,670
	2	22,537	6,885	462	24,999
	3	14,282	725	2,084	17,759
	4	5	1	62,917	108
5	1	2,002	64,907	0	1,583
	2	46,980	618	0	30,518
	3	16,554	11	0	33,435
	4	0	0	65,536	0
6	1	0	65,535	0	2
	2	58	1	0	65,515
	3	65,478	0	0	19
	4	0	0	65,536	0

Table 8. Step2 Performance of the 2 Algorithms

입력변수 크기(n)	Time(S)		Error rate(%)	
	TD_ARR	TD_BMS	TD_ARR	TD_BMS
4	3.05	2.93	1.5	0
5	9.06	8.55	4.7	0
6	35.50	30.58	14.1	0

Table 9. A Sample Syllabus including Circuit Minimization

차시	단원 주제	학습 및 평가 내용
1	스위치 개념 및 종류	• 생활 속의 스위치 사례 • 물에 대한 개폐 스위치 만들기
2	논리 게이트 개념	• 물 개폐 스위치를 이용한 AND/ XOR 논리 게이트 만들기
3	전자 스위치 및 전자 논리 게이트	• 진공관(에디슨 열전자 발명, 진공관 스위치 원리, 진공관을 이용한 AND/XOR 논리 게이트 구현 원리) • 트랜지스터(러셀 반도체 발명, 트랜지스터 스위치 원리, 트랜지스터를 이용한 AND/XOR 논리 게이트 구현 원리) • IC(킬비 제조 기술, IC 논리 게이트 기능 실험)
4	자동 계산 회로 고안하기	• 2진 수 표현법 • 2진 덧셈을 관찰하고 이를 위한 진리표 만들기 • 진리표에 대한 직관적 논리 게이트 조합 고안 • 비트 단위 자동 덧셈 회로 실험
5	회로 축약 개념 및 방법	• 등가 논리 회로 개념 • 직관적 논리 회로 축약 한계 인식 • 까르노 맵 방법론 활용
6	회로 축약 과정의 자동화	• 까르노 맵 방법론의 한계 • 부울 함수 및 비트 연산 • Quine-McCluskey 방법론 개념
7	Quine-McCluskey 알고리즘 구현	• C 언어 및 자료구조 • 나열식 집합 표현 및 연산
8	Quine-McCluskey 알고리즘 연구 동향	• 개선 여지 존재 여부 탐색 • 하향식 접근과 상향식 접근의 비교
9	Quine-McCluskey 알고리즘 개선	• 비트 배열식 집합 표현 및 연산 적용 • 시간 성능 및 정확도 비교 실험
10	종합 정리	• 논리 게이트는 컴퓨터의 세포 • 알고리즘에서 자료구조의 중요성

References

[1] Yae-Won Seo, Jae-Boon Lee, et al., "A study for establishment of the 3rd Master Plan for the Promotion of Gifted and Talented Education," *Korea Foundation for the Advancement of Science & Creativity(Research Report)*, 2011(http://www.kofac.re.kr/download.ddo?fid=bbs&bbs_cd_n=106&bbs_seq_n=37&order_no_n=1).

[2] JeongWon Choi and YoungJun Lee, "An Analysis of the Effectiveness of Informatics Gifted Education," *Korean Journal of Teacher Education*, Vol.29, No.4, pp.115-129, 2013.

[3] Wochun Jun, "A Study on the Current Status and Improvement Plan of Gifted Information Education Curriculum for Creative Human Resource Development," *Communications of the Korean Institute of Information Scientists and Engineer*, Vol.30, No.3, pp.17-23, 2012.

[4] Hyung-Bong Lee and Ki-Hyeon Kwon, "Development of a Convergent Teaching-Learning Materials based on Logic Gates using Water-flow for the Secondary Informatics Gifted Students," *Journal of The Korea Society of Computer and Information*, Vol.19, No.12, pp.369-384, 2014.

[5] Mi-Sook Han and YoungJun Lee, "Design of Algorithm Education Contents for the Informatics Gifted Secondary Students," *Proceeding of Korean Association of Computer Education Conference*, Vol.16, No.1, pp.192-195, 2012.

[6] Hosook Kim and Hyoungseok Kim, "A Study on the Teaching-Learning Methods for Improvement of Creativity on Information Science Applying Project-Based Learning," *Journal of The Korean Association of Information Education*, Vol.18, No.4, pp.529-540, 2014.

[7] KyungKyu Kim and JongYun Lee, "Computer Education Curriculum and Instruction: Design of a Stepwise Enrichment Curriculum for the Gifted and Talented in Informatics," *The Journal of Korean Association of Computer Education*, Vol.17 No.1, pp.35-50, 2014.

[8] Ministry of Education, "The 3rd Master Plan for the Promotion of Gifted and Talented Education," *Government Document*, 2013(<http://gifted.kedi.re.kr/images/gifted/policy/report04.pdf>).

[9] Ji Eun Yi and Hyeon-Suk Kang, "Application of Backward Design in Gifted Education," *The Journal of the Korean Society for the Gifted and Talented*, Vol.13, No.1, pp.129-154, 2014.

[10] Audion(vacuum tube) [Internet], http://en.wikipedia.org/wiki/Lee_de_Forest.

[11] NPN Sandwich Transistor(TR Triode) [Internet], http://en.wikipedia.org/wiki/William_Shockley.

[12] M. Karnaugh, "The Map Method for Synthesis of Combinational Logic Circuits," *Transactions of the American Institute of Electrical Engineers part I*, Vol.72 No.9, pp.593-599, 1953(<https://users.fit.cvut.cz/~staryja2/BIMLO/karnaugh-the-map-method.pdf>).

[13] W. V. Quine, "A Way to Simplify Truth Functions," *The American Mathematical Monthly*, Vol.62, No.9, pp.627-631, 1955(http://www.jstor.org/stable/2307285?seq=1#page_scan_tab_contents).

[14] E. J. McCluskey, "Minimization of Boolean Functions," *Bell System Technical Journal*, Vol.35, No.6, pp.1417-1444, 1956(<https://archive.org/stream/bstj35-6-1417#page/n13/mode/2up>).

[15] Sang-Un Lee, "An Improved Quine-McCluskey Algorithm for Circuit Minimization," *Journal of The Korea Society of Computer and Information*, Vol.19, No.3, pp.109-117, 2014.



이 형 봉

e-mail : hblee@gwnu.ac.kr

1984년 서울대학교 계산통계학과(학사)

1986년 서울대학교 계산통계학과(석사)

2002년 강원대학교 컴퓨터과학과(박사)

1986년~1993년 LG전자 컴퓨터연구소

선임연구원

1994년~1998년 한국디지털(DEC Korea) 책임연구원

1999년~2003년 호남대학교 정보통신공학부 조교수

2004년~현 재 강릉원주대학교 컴퓨터공학과 교수

관심분야: 임베디드 시스템, 센서 네트워크, 데이터마이닝 알고리즘



권 기 현

e-mail : kweon@kangwon.ac.kr

1993년 강원대학교 전자계산학과(학사)

1995년 강원대학교 전자계산학과(석사)

2000년 강원대학교 컴퓨터과학과(박사)

1998년~2002년 동원대학 인터넷정보과 교수

2002년~현 재 강원대학교 전자정보통신

공학부 교수

관심분야: 패턴인식, 미들웨어, 임베디드 소프트웨어