

A Brief Review on Mathematicians' Influence on the Initial Developments of Computers

컴퓨터 발전 초기과정에 영향을 준 수학자들에 관한 고찰

LEE Seung-Woo 이승우

Computer is a modern day invention integrated with mathematics, engineering, and logics. The purpose of this study is to examine mathematicians' roles and influences on the invention, establishment, and developments of computers, particularly in the areas of hardware and software, and to emphasize the importance of mathematics on the computer sciences. To implement these purposes, this study firstly examines the mathematicians based on the period. Secondly from the mathematicians' roles in the development of programming, the correlation between mathematics and computers has been investigated. Finally, mathematicians who gave influence on establishing the current development of computer science are highlighted.

Keywords: Mathematician, Computer, Computer science, Program; 수학자, 컴퓨터, 컴퓨터과학, 프로그램.

MSC: 97B10 ZDM: M55

1 서론

인간 생활에 있어서 수와 양을 표시하고 계산하는 방법은 인류의 역사와 더불어 꾸준히 발전되어 왔다. 인류는 생활수준의 향상으로 보다 복잡한 수를 취급할 필요성에 의해서 계산기를 개발하게 되었고, 오늘날에는 컴퓨터의 등장으로 고도로 복잡한 계산을 손쉽게 할 수 있게 되었다.

컴퓨터는 기술 문명이 발달한 현대사회에서 생활 속의 편리함과 작업 효율성을 극대화시켜주는 역할을 하고 있다. 우리는 컴퓨터 없이는 단 하루도 살 수 없는 세상에 살고 있다. 그렇다면 컴퓨터는 어떻게 누구에 의해서 발명되었는가? 하는 의문이 든다. 일반인들의 생각으로는 공학자들에 의해서 컴퓨터가 창조되고, 개발 및 발전시키는데 큰 역할을 담당

했다고 인식하고 있다. 그러나 컴퓨터를 창조할 뿐만 아니라 작동하는 근본적인 원리는 수학자들에 의해서 제공되었다는 중요한 사실을 간과하고 있다.

컴퓨터는 기본적으로 하드웨어 요소와 시스템 소프트웨어 요소가 결합되어 구성된다. 하드웨어는 컴퓨터의 내부적인 동작을 담당하는 물리적인 기계장치이고, 시스템 소프트웨어는 하드웨어 구동과 관련된 프로그램들의 집합체로서, 소프트웨어는 외부적인 운영을 담당하는 논리적인 명령어들의 집합이다.

초기의 컴퓨터 발전 단계에서 사용된 하드웨어와 소프트웨어(프로그램)는 수학자들에 의해서 창조되었고 개발된 후, 수많은 과학자와 공학자들이 반세기 동안 연구한 결과가 바탕이 되어 오늘날의 컴퓨터로 급진적으로 진화되었다.

본 논문에서는 컴퓨터 창조와 개발에 영향을 미친 수학자들의 역할에 대하여 하드웨어와 소프트웨어(프로그램) 관점에서 각각 파악해보고 컴퓨터과학 분야에서의 수학의 중요성을 밝히는 것이 목적이다.

본 논문에서는 연구내용을 크게 세 관점에서 수행하고자 한다. 첫째, 초기 컴퓨터 발전 과정에서 영향을 준 수학자의 역할을 파악하고자, 고대에서 사용한 계산도구, 수동기계식 컴퓨터, 자동기계식 컴퓨터, 전기기계식 컴퓨터, 전자식 컴퓨터의 시대로 각각 구분하여 수학자들의 업적을 분석하고자 한다. 둘째, 컴퓨터프로그램의 생성과정에서 수학자들의 역할을 밝힘으로서 컴퓨터과학의 프로그래밍과 수학자들 간의 관련성을 알아보하고자 한다. 셋째, 현재 컴퓨터 발전과정에서 컴퓨터과학 분야에 영향을 준 수학자들의 역할에 관하여 파악하고자 한다. 이를 통하여 오늘날의 컴퓨터는 공학자만의 발명품이 아닌 수학자들의 공동 연구결과를 토대로 발전했음을 밝히고자 한다.

2 초기 컴퓨터 발전과정에서 영향을 준 수학자의 역할

2.1 하드웨어 발전 초기과정에서 영향을 준 수학자의 역할

2.1.1 고대의 계산도구

숫자의 기원은 메소포타미아에서 사용된 췌기 문자가 역사상 최초로 사용된 숫자로 추측되며, 이로부터 수 천년동안 인간은 숫자의 개념을 생활에 적용하면서 정확하고 빠른 계산의 필요성을 인식하게 되었다. 이집트, 그리스, 인도, 중국에서 10진법과 같은 수의 체계를 정립하고 효율적이고 정확한 계산방법을 연구하였다.

컴퓨터의 기원은 계산도구에서 유래되었다. 계산을 할 때 편리하게 사용하는 도구를 발명하였는데 주판과 같은 계산기가 이에 속한다. 주판은 인류 역사상 가장 오래된 계산도구이며 동양과 서양에서 각각 독자적으로 발전된 것으로 추측된다.

동양에서는 중국에서 주판이 기원전 26세기경에 발명되었으며 15세기 중반부터 널리

보급되었고 그 이전까지는 산목(算木)으로 썼하였다.

서양에서는 메소포타미아에서 3,000~4,000년 전에 널빤지에 모래나 분말을 놓아 셈하는 토사주판을 사용했으며, 로마에서는 3~4세기에 흙을 판 널빤지 위에 여러 개의 줄을 그고 이에 바둑알을 놓아 셈하는 흙주판을 사용했다. 로마인들은 주판알을 'Calculi' 라고 불렀으며, 오늘날 'Calculate'의 어원이 되었다. 이후 서양식 주판은 6세기경 로마의 철학자 보이티우스(Boethus), 10세기경 교황 실베스터 2세(Sylvester II) 등에 의해 개량되었다. 17세기에는 전투 기술을 향상시킬 목적으로 총의 탄도를 계산해내는 도구에 대한 필요성에 따라 '섹터'라고 불리는 메커니즘을 갈릴레오(Galileo Galilei: 1564~1642)를 포함한 여러 발명가들이 고안하였다[2]. 그러나 17세기 아라비아숫자의 보급으로 종이위의 필산 위주의 셈법이 활성화되면서 쇠퇴하였다.

네이피어(John Napier: 1550~1617)는 영국의 수학자로서 곱셈을 쉽게 계산할 수 있는 봉을 고안했다. 이것을 네이피어의 봉(Napier's bones)이라고 한다. 네이피어의 봉은 아홉 개의 긴 막대로 구성이 되어 있는데, 각 막대에는 1부터 9까지 숫자들의 곱셈표가 그려져 있다. 당시 네이피어의 봉에 의한 곱셈의 계산법은 놀라울 정도로 획기적인 발명이었고, 이것은 20세기 초까지 공업계의 기술자들이 사용했던 계산기의 모태가 되었다. 그는 산술·대수(代數)·삼각법 등의 단순화·계열화를 꾀했으며, 특히 계산의 간편화를 목적으로 한 수학사상 커다란 업적인 로그(logarithm)를 발명하였다[9].

2.1.2 계산기; 수동기계식 컴퓨터

20세기 중반까지 수동 또는 모터로 구동되는 상업적이고 실용적인 기계식 계산기들이 세계각지에서 제작되었으며, 이중에 수학자가 제작한 계산기는 다음과 같다.

파스칼(Blaise Pascal: 1623~1662)은 프랑스의 수학자로서 톱니바퀴를 이용한 최초의 계산기의 원형을 1642년에 고안하였다. 이 기계식계산기는 치차식계산기라고 하며 0부터 9까지의 숫자를 톱니바퀴의 회전 원리를 이용하여 덧셈, 뺄셈, 자리 올림이 가능하도록 제작되었다[2].

라이프니츠(Gottfried Wilhelm Leibniz: 1646~1716)는 독일의 수학자로서 사칙연산이 가능한 탁상용 계산기인 단계적계산기를 1671년에 최초로 고안하여 발전시켰다. 그는 파스칼 계산기에 기어를 추가하여 덧셈을 반복하는 곱셈 동작과 뺄셈을 반복하는 나눗셈 동작을 제작했고, 당시 수학을 비롯한 기초과학에 적용하였다. 그러나 이 단계적계산기는 파스칼의 기계식계산기에 자동 곱셈과 나눗셈 기능을 추가했으나, 받아 올림과 받아 내림을 완벽하게 자동화시키지는 못한 단점이 있었다. 라이프니츠는 1685년에 핀 톱니바퀴 계산기를 최초로 고안했으며, 최초로 대량생산된 기계적계산기인 라이프니츠 휠을 발명했다[11].

라이프니츠는 그 어떤 수도 0과 1만을 이용해서 표현할 수 있는 2진법 수 체계를 정

립시켰고 이 체계의 간결성으로 오늘날 모든 디지털 컴퓨터의 기반이 되었다. 그는 1679년에 펀치 카드의 초기 형태인 공깃돌로 이진수를 표현하는 방식의 기계를 고안했다. 현대 컴퓨터는 레지스터¹⁾와 전위차에 의해 생성된 전자의 흐름으로 작동되는데, 이 방식은 그가 1679년에 상상한 것과 비슷하다.

컴퓨터 초기발전과정부터 현재까지 라이프니츠가 발견한 2진법이 컴퓨터에서 사용되는 용도로서는 높은 전압(H)을 1로, 낮은 전압(L)을 0으로 하는 논리 형식을 기초로 전기 신호의 유(有)와 무(無), 스위치의 ON과 OFF 등으로 표현 가능하며 전화 교환기나 컴퓨터의 논리 회로에 응용된다. 컴퓨터에서 처리하는 숫자는 이진법을 사용하기 때문에 논리의 조립으로 간단하게 활용되므로 컴퓨터 내부의 소자²⁾에서 사용된다. 특히 디지털 신호는 기본적으로 이진법 수들의 나열이다[6].

2.1.3 계산기; 자동기계식 컴퓨터

19세기는 산업화와 함께 식민지 개척이 한창이던 때이므로, 항해나 기계 제작 또는 포탄의 탄착점 계산 등에서 많은 수학계산이 필요했다. 더욱이 산업혁명이 시작되면서 경제는 급속도로 발전해서 복잡한 계산이 요구되는 시기였다. 당시는 미리 계산된 곱셈표, 로그함수표, 삼각함수표 등이 쓰였으나, 모든 종류의 계산에 대한 답을 표로 만들어 놓을 수 없었다. 미리 정해진 몇 가지 범위를 벗어난 나머지 계산들은 필산으로 직접 계산했다. 이런 불편을 해소하기 위해 수학문제를 즉시 계산해낼 수 있는 기계가 필요한 시기였다[6].

배비지(Charles Babbage: 1792~1871)는 영국의 수학자로서 컴퓨터의 기초 원리에 대한 기본 착상을 갖고 임의의 프로그래밍으로 하나의 장치에서 모든 계산이 가능하도록 한 최초의 기계를 고안했다. 이 기계는 이전에 개발된 수동기계식과는 차원이 다른 자동기계식 계산기로서 숫자와 계산명령을 같이 입력하여 다양한 계산을 할 수 있는 전자식 컴퓨터의 모태가 되었다. 1823년 로그함수와 삼각함수의 값을 계산해 대수표를 계산하는 데 사용된 차분 기관(differential engine)을 제작하였고 1833년 모든 일반계산을 하나의 기계에서 할 수 있는 해석 기관(analytical engine)을 제작했다. 이 기계는 기억·연산·입출력·제어 장치 등을 갖춘 현대적인 컴퓨터의 원형으로 설계되었다.

배비지는 계산기에 관한 연구의 선구자로서 수의 계산과 오차에서, 기계에 의한 계산과 수치표의 작성을 실현하는 등 계산기의 이론을 연구하였으나, 그 당시의 기술 수준으로는 정교한 부품을 제작하지 못해, 계산기를 완성하는 단계까지는 이르지 못하였다. 그러나 연산의 순서를 자동으로 제어하는 방법은 자동기계식 컴퓨터의 시초가 되었으며, 오늘날 디지털 컴퓨터의 기본적인 바탕을 이루었다는 점에서 높이 평가되고 있다[2].

1) 레지스터(register)는 PC의 CPU에 들어있는 데이터 기억장치이다.

2) 소자란 기계 회로나 전자 회로 등의 구성 요소가 되는 낱낱의 부품으로, 독립된 고유의 기능을 갖고 있는 것을 말한다.

2.1.4 전기기계식 컴퓨터

전기기계식 컴퓨터는 전기적인 요소와 기계적인 요소를 결합한 방식으로 릴레이(relay)의 작동원리를 이용한다. 릴레이는 전기로 구동되는 기계식 접속장치로서, 2진법을 이용한다. 즉, 릴레이는 전기 회로의 개폐를 통해 접점이 붙거나 떨어지는 기계적인 스위칭으로 2진법을 이용하여 표현한다. 디지털 논리회로에서 1과 0을 전달하기 위해 조건에 따라 전류가 흐르거나 흐르지 않도록 만든다. 전기기계식 컴퓨터는 디지털 논리의 전달을 위해 릴레이를 사용해 그 역할을 대신한다. 여기서 사용된 작동원리는 불(George Boole: 1815~1864)의 수학적 이론이 토대가 되었으며 컴퓨터가 디지털 연산장치로 진화할 수 있었다.

불은 영국의 수학자로서 논리대수인 불 대수(Boolean algebra)를 창안했고, 논리학자로서 기호논리학(記號論理學) 분야에 큰 업적을 남겼다. 불 대수는 현대 컴퓨터에서 매우 중요한 착상인데, 이것은 컴퓨터에 입력되는 값이 0, 1과 같은 전기적 신호이기 때문이다. 이를 통하여 전기 회로 장착 방법에 응용하여 전자식 컴퓨터를 만드는 토대가 되었고 전자 공학, 컴퓨터 하드웨어 및 소프트웨어 등에서 넓게 응용되고 있다.

컴퓨터 초기발전과정부터 현재까지 불은 논리 대수를 창안하여 현재 디지털 시스템의 논리설계를 위한 기반을 마련했다. 논리회로(logic circuit)란 불 대수의 기본 연산인 논리합, 논리곱, 논리부정 등의 연산을 실행하기 위한 회로이다. 2진 정보를 취급하며 보통 2개 이상의 입력 단자와 하나의 출력 단자로 구성되는데 디지털 시스템의 기본 요소가 된다. 디지털 논리 회로의 기본 연산은 OR, AND, NOT이지만, 실제 컴퓨터에서는 NOR, NAND 게이트가 OR과 AND 게이트보다 트랜지스터로 쉽게 만들 수 있기 때문에 많이 사용되고 있다. 즉, 컴퓨터 회로에 논리 게이트를 설계한 후, 논리게이트는 비트 1과 비트 0으로 불 논리에 일치하는 출력을 생성하도록 전기의 흐름을 제어할 수 있다. 즉, 컴퓨터 사용자가 입력한 정보를 논리적으로 작동시키기 위해서 불 대수를 이용하여 컴퓨터 내부에서 입력된 데이터를 처리하는 방식이다[2].

2.1.5 전자식 컴퓨터

20세기, 전자공학의 발달과 2차 대전을 통한 군사적인 필요성 때문에 기계식계산기는 전자식계산기로 발달하는 계기가 되었다. 숫자와 계산명령을 같이 입력하여 다양한 계산을 할 수 있는 전자식 컴퓨터가 도입되자 전기기계식 컴퓨터와는 비교할 수 없을 정도로 계산 속도가 빨라졌다. 오늘날 컴퓨터는 전자회로로 만들어진 전자식 컴퓨터를 말한다.

오늘날까지 현대 컴퓨터의 발전은 컴퓨터의 성능을 결정하는 중앙연산처리장치와 전자회로의 발달에 따라 1세대~5세대 컴퓨터로 구분되어지며, 특히 5세대 컴퓨터(1984~)부터는 고도의 시스템 분야뿐만 아니라 수학/통계를 도구로 하는 인공지능, 전문가 시스템, 패턴인식 시스템, 의사결정 시스템, 퍼지이론 등에 활용되며 보다 복잡한 계산을 수행하고 있다.

앨런 튜링(Alan Mathison Turing: 1912~1954)은 영국의 수학자로서 컴퓨터과학에 지대한 공헌을 했기 때문에 컴퓨터과학의 아버지로 불린다. 1936년 대학원 석사논문인 '계산가능한 수에 관하여, 수리명제 자동생성 문제에 응용하면서 (On Computable Numbers, with an Application to the Entscheidungsproblem)'을 런던 수리학회(London Mathematical Society)에 발표하였다. 이 논문은 계산기계가 어디까지 논리적으로 작동할 수 있는가에 대해 처음으로 연구를 시도한 것이다. 이 논문에서 튜링은 컴퓨터의 근본적인 디자인을 최초로 선보였고 컴퓨터의 수학적 모델을 제시하였다[6].

앨런 튜링이 창안한 튜링 기계(Turing machine)는 무한대의 저장용량과 고장이 없는 수학 원리로 구성된 일종의 가상기계인 초보적 형태의 컴퓨터로서, 복잡한 계산과 논리 문제를 처리할 수 있었다. 그가 제안한 튜링 테스트(Turing test)는 컴퓨터가 얼마나 사람과 비슷하게 대화할 수 있는지 판단해 인공지능을 갖는 정도를 판정하는 테스트로 유명하다[6].

또한 앨런 튜링은 인공지능, 암호해독, 프로그래밍시스템 등을 연구했으며, 특히 컴퓨터과학의 발전에 매우 영향력 있는 범용 컴퓨터의 모델인 튜링 기계와 알고리즘 개념을 공식화하여 오늘날 컴퓨터의 원형을 최초로 제시하였다.

폰 노이만(John von Neumann: 1903~1957)은 미국의 수학자로서 컴퓨터 중앙처리장치의 내장형 프로그램을 처음 고안했다. 1949년 에드박(EDVAC: Electronic Discrete Variable Automatic Computer)이라는 새로운 개념의 컴퓨터를 만들었으며 이때 고안한 방식은 오늘날에도 거의 모든 컴퓨터 설계의 기본이 되고 있다.

이 기계는 연산처리·제어명령·기억기능이 저장되는 프로그램 내장 방식의 컴퓨터로서, 설계 방법과 동작 원리는 현재까지 컴퓨터에 도입되어 폰 노이만식 컴퓨터라고 할 정도로 컴퓨터의 역사상 중요한 위치를 차지하게 되었다. 그의 프로그램 내장 방식은 한 프로그램에서 다른 프로그램으로의 전환이 가능하고, 컴퓨터가 다양한 업무를 수행할 수 있도록 해주는 획기적인 계기를 마련하였으며, 순차적으로 작업이 수행되는 컴퓨터라고 할 수 있다[2].

폰 노이만은 기하학, 양자이론, 게임이론, 수리경제학, 선형계획법, 컴퓨터 등에 많은 업적을 남겼다. 특히 그는 CPU·메모리·프로그램 구조를 갖는 프로그램 내장 이론을 적용하여 계산기를 탈피한 현재 사용하고 있는 컴퓨터의 모델을 제시하였고, 이 프로그램 내장 이론은 오늘날까지 모든 컴퓨터에 활용되고 있다.

2.2 소프트웨어(프로그램) 발전 초기과정에서 영향을 준 수학자의 역할

근래에 와서는 컴퓨터 프로그래밍의 기초이론과 기호논리학 사이에 밀접한 교섭이 있음이 밝혀졌다[8]. 그러므로 본 장에서는 컴퓨터프로그램에 영향을 준 수학자들에 관하여 알아보고자 한다.

17세기부터 18세기에 이르기까지 활약한 라이프니츠는 최초로 개념, 명제, 추론 등을 기호

화하여 연구하는 형식적인 논리학인 기호논리학(記號論理學)을 구상하였으며, 기호논리학을 기초로 훗날 프로그래밍 탄생의 기반을 조성했다[1]. 위 연구들을 토대로 컴퓨터과학의 하드웨어분야와 소프트웨어(프로그래밍)분야에서 앞으로 나올 개념들을 예견하는데 공헌하였다. 이를 통하여 그는 최초로 컴퓨터과학의 기초 이론을 수립했으므로, 최초의 컴퓨터과학 이론가였다.

19세기에 접어들어 수학의 발전 영향 아래에서 논리학을 새로운 견지에서 만들려는 시도가 불에 의해서 제안되었다. 그 후 칸토어(Georg Ferdinand Ludwig Philipp Cantor: 1845~1918)가 창시한 집합론을 전후해서 이 방면의 연구는 큰 발전을 거듭했고, 독일의 수학자 프레게(Friedrich Ludwig Gottlob Frege: 1848~1925)에 의해서 개념표기법(Begriffsschrift)이 1879년에 출판됨으로서 기호논리학의 체계는 일단 완성을 보았다고 할 수 있다. 즉, 전통적 논리학이 기호를 사용해서 엄밀하게 표현하는 방법을 거의 확립했다[8].

프레게는 논리적인 언어에 있어서 일상 언어를 사용함으로써 일어나는 애매성, 모호성, 다의성을 제거하고 명료성과 엄밀성을 갖춘 기호언어를 발명하여 언어를 기호화한 언어철학자이다. 그는 개념 표기법을 구문론이라는 엄격하게 정밀한 규칙을 가진 인공 언어로 발전시켰다. 이를 통해 논리적 추론을 기호들이 배열된 방식에만 관련 있는 순수하게 기계적인 조작, 이른바 추론 규칙으로 나타내는 것이 가능하게 되었으며 이것은 정밀한 구문론으로 짜여진 형식 인공 언어의 첫 번째 사례가 되었다. 이런 관점에서 볼 때, 개념 표기법은 오늘날 일반적으로 사용되는 모든 컴퓨터 프로그래밍 언어의 시초가 되었다[1].

또한 프레게는 수학이 논리학으로 귀결될 수 있다는 논리주의(logicism)의 초기 주창자 중에 한명이었다. 그는 근대 수리철학과 분석철학(analytic philosophy)의 기초를 마련했으며 현대 수리논리학의 창시자로 인정받고 있다[16].

페아노(Giuseppe Peano: 1858~1932)는 이탈리아의 수학자로서, 1890년에는 평면 위의 모든 점을 통하는 페아노 곡선을 구성했고 1891년에는 수학 잡지를 창간했다. 그는 이 잡지에 ‘수학적 논리학의 공식’, ‘수의 개념에 관해’ 등의 논문을 페아노 기호를 사용하여 발표함으로써 기호논리학의 개척자로 불린다. 페아노의 공리(公理)로 알려진 자연수론(自然數論) 공리를 확립하고 기호논리학의 도움을 받아 자연수 체계의 여러 가지 성질을 명확하게 한 것은 페아노의 가장 큰 공적이다[15].

화이트헤드(Alfred North Whitehead: 1861~1947)와 러셀(Bertrand Russell: 1872~1970)은 영국의 수학자들로서, 기호논리학에 공헌하였다. 그들의 공저인 수학원리(Principia Mathematica)에서 집합론을 포함한 수학의 범위를 완전히 기호논리로서 형성하는 결실을 보았고, 현대의 기호논리학과 분석철학의 기초를 이루었다[10].

특히 러셀은 수학자들이 사용하는 일상 언어가 수학과 같은 엄밀한 학문에는 어울리지 않을 만큼 애매모호하다고 생각했다. 그는 이런 애매성을 제거하기 위해서는 기호의 의미에

의존하기보다 엄밀한 기호 조작이 훨씬 더 도움이 된다고 믿었다. 그는 이러한 수학적 증명 체계의 출발점을 논리학의 영역에서 제공할 수 있다고 생각했다. 즉, 수학은 논리학에 전적으로 환원될 수 있다는 논리주의(logicism)를 주장했다[12].

또한 러셀은 논리학의 인공 언어로 실제의 수학을 발전시키는 데 성공했으며 이를 통하여 컴퓨터 프로그래밍 언어의 기초를 이룰 수 있는 토대를 만들었다[1].

19세기 말부터 20세기 초까지 가장 위대한 수학자 중 한명인 힐베르트(David Hilbert: 1862~1943)는 독일의 수학자로서, 1989년 기하학 기초(Grundlagen der Geometrie)에서 유클리드 기하학에 대한 공리들을 모아 중요성을 분석하였고 그 뒤 수학의 형식주의 기초를 세우는 데 공헌했다. 형식주의는 수학의 기초 확립에 관하여 힐베르트가 제창한 입장으로 수학의 이론을 공리화하고 나서 이것을 논리기호로서 철저하게 형식화하여 이 형식화된 체계를, 기호배열(記號配列)의 변형과정으로 보고, 이 과정에서 어떠한 특정의 기호 배열이 나타나는지 어떤지에 주목하는 방법, 이른바 메타 수학적 방법(metamathematical method)을 제시하였다[17].

힐베르트에 의한 힐베르트공간론의 창설과 1909년 적분방정식의 성과는 20세기 함수해석학의 기초가 되었다. 또한 일반 상대성 이론을 수학적으로 정의하는 데 핵심적인 역할을 하였다. 힐베르트는 당대 수학계의 지도자였으며, 힐베르트의 23가지 문제의 연구과제로 명성을 얻었다. 그는 수학의 문제들(The Problems of Mathematics)이라는 발표에서 그 당시의 거의 모든 수학문제를 조사해서 20세기 수학자들에게 중요하다고 생각되는 문제들을 설명하려고 노력했고, 심지어 오늘날까지도 수학계의 흐름에 큰 영향을 끼치고 있다. 그의 업적은 수학의 거의 모든 부문에 미치고 있다[18].

힐베르트의 형식주의는 인간의 언어가 가지는 결함을 제거하기 위해 모든 수학의 명제를 형식적 기호의 나열로 바꾸고, 그런 명제들 중에서 모순이 생성되는 요소를 피해 수학의 공리를 세우고, 거기서 새로운 명제를 유도하는 법칙을 세운다. 즉 일종의 기계적 프로그램으로 수학을 이해하자는 것으로서, 이러한 형식적이고 기계적 체계 하에서 수학의 기본 공리를 세심히 설계하면, 모순이 절대로 유도되지 않는 것을 보일 수 있으리라 예상했다. 이것을 힐베르트 프로그램이라고 부른다[7]. 이를 토대로, 그에 의해 고안된 형식주의 논리 체계들이 컴퓨터 프로그램을 연구하는 데 기초가 되었다[1].

20세기 가장 영향력 있는 수학자 가운데 한 사람으로 추앙받는 괴델(Kurt GÖdel: 1906~1978)은 불완전성의 정리로 유명한 오스트리아의 수학자이다. 괴델은 1931년에 20세기 수학기초론, 논리학에서 가장 중요한 발견으로 여겨지는 불완전성 정리를 발표했다. 이는 힐베르트가 수학의 무모순성을 증명하기 위해 추진했던 힐베르트 프로그램의 일부로 연구되었던 것이지만, '수학은 자신의 무모순성을 증명할 수 없다'는 것을 보인 불완전성 정리는 역으로 힐베르트 프로그램을 좌절시켰다. 이 사실은 힐베르트의 프로그램이 성취될 수 없다는 것을

의미한다[7].

괴델의 제1, 제2정리를 요약하면, ‘진리이나 증명되지 않는 수학적 명제가 존재한다’이다. 괴델은 참인 수학적 명제들의 범위가 인간이 궁극적으로 증명의 방법을 통해 참으로 확인해 인식할 수 있는 명제들의 범위를 넘어서는 것을 보인 것이다. 이를 위해 우선 ‘인간이 참으로 인식 또는 증명할 수 있는 명제’들의 범위를 규정하기 위해 ‘계산 가능하다’는 개념을 최초로 제안하게 된다. 나중에 앨런 튜링과 후대 학자들에 의해 정의가 확장 보장되었다. 그렇다면 2.1.5절에서 소개한 앨런 튜링은 기계적 프로그램으로 ‘계산 가능하다’는 개념을 실행하는 장치를 만들 수 있지 않을까 생각했고, 실제 컴퓨터를 설계하기에 이른 것이다[7].

불완전성 정리에 사용된 코딩이론, 계산이론 등은 컴퓨터에서 사용될 이론적 배경이 내포되어 있다. 괴델의 업적과 생각을 발전시켜 세계 최초로 현대 컴퓨터 이론의 기초를 세우고, 실제 컴퓨터 설계에 확장 구현한 사람은 2.1.5절에서 소개한 앨런 튜링과 폰 노이만이다[7].

3 현재 컴퓨터 발전과정에서 영향을 준 수학자의 역할

컴퓨터의 하드웨어와 소프트웨어 분야의 성장 및 개발에 영향을 준 수학자들은 수없이 많지만, 본 장에서는 2장에서 언급한 수학자들만 한정하여 그 업적을 살펴보기로 한다.

파스칼이 창시한 근대 확률이론은 오늘날 여러 가지 형태의 데이터를 분석하고 모델링하는 데 이용되는 가장 강력한 도구로 사용되고 있다[14]. 즉, 컴퓨터 공학을 비롯한 전기·전자·정보통신공학 분야에서 생성된 데이터나 신호들이 랜덤한 특성을 갖고 있기 때문에 수학만을 이용해 파악하는 데는 한계가 있으므로 확률을 도구로서 분석해야 한다. 특히 위의 공학 분야에서 개발되는 기술 해석 및 그 기술과 관련된 여러 가지 형태의 데이터 분석과 모델링을 시행하는 데 있어서 확률을 통해 해석하고 확률을 활용하여 관련 시스템을 설계해야 한다. 근래에 들어 공학 분야 전반에 걸쳐 확률을 기본적인 도구로 요구하고 있는 실정이다.

위의 공학 분야를 소프트웨어와 하드웨어 분야로 구분하여 확률을 기본도구로 활용하고 있는 분야들을 파악해 보면 다음과 같다. 소프트웨어 분야에서는 디지털 영상처리, 디지털 음성처리, 멀티미디어, 네트워크 및 데이터통신, 패턴인식, 생체인식, 인공지능 등이며 하드웨어 분야에서는 신호 및 시스템, 통신이론, 통신공학, 이동통신, 컴퓨터네트워크, 디지털 신호처리 및 분석, 멀티미디어 신호처리, 지능시스템, 컴퓨터비전, 제어공학, 로봇공학, 지능로봇시스템 등에서 확률을 필수 도구로 사용한다[3].

칸토어가 창시한 집합론은 데이터베이스³⁾의 기본 개념을 정립시켜주었다. 데이터베이스는 데이터의 가장 작은 논리적 단위인 개체(entity), 인스턴스와 관계로 구성되는데, 수학적 관점에서 보면 개체는 원소에, 인스턴스는 집합에, 관계는 함수 또는 사상에 해당된다. 데이터베이스

3) 데이터베이스(database)란 여러 가지 업무에 공동으로 필요한 데이터를 유기적으로 결합하여 저장한 집합체를 말하며, 데이터를 효율적으로 처리하기 위하여 개발된 것으로 데이터를 보관하고 관리한다.

시스템 및 관계데이터모델에서의 릴레이션(relation)은 불 대수를 기본으로 하는 관계 대수(relational algebra)⁴⁾와 관계 해석(relational calculus)을 기반으로 한다[5].

현재 컴퓨터 프로그래밍의 기초이론과 기호논리학 사이에 밀접한 교섭이 있음이 밝혀졌다. 프로그래밍 언어 분석에 기호논리학을 활용하여 프로그래밍 언어의 설계 및 구현에 대한 원리, 구문구조의 본질 등을 수학적 논리학을 토대로 전개함으로써 프로그램 논리의 흐름을 효율적으로 이해할 수 있다[1]. 이를 통하여, 첫째, 프로그래밍 언어 자체의 구조를 파악하고 최신의 프로그래밍 언어들을 서로 비교 분석함으로써 문제 해결을 위한 시스템 설계 및 프로그래밍 기술을 향상시키고, 둘째, 개발코자하는 시스템에 적합한 프로그래밍 언어 선정 및 프로그래밍 언어 개발 능력을 배양시킬 수 있다[4]. 컴퓨터 프로그래밍 분야는 역사는 짧으나 응용분야가 넓고 다양해서 큰 전망을 가진 학문으로 지금도 끊임없는 발전을 거듭하고 있다

앨런 튜링이 발명한 튜링 테스트(Turing test)는 기계가 인간과 얼마나 비슷하게 대화할 수 있는지를 기준으로 기계에 지능이 있는지를 판별하고자 하는 테스트로서 1950년에 제안했다. 앨런 튜링은 프로그래밍이 가능한 가설적 기계 장치인 튜링 머신을 구상하여 제시함으로써 컴퓨터과학의 토대를 마련했으며, 그가 제시한 튜링 테스트는 아직도 인공지능의 기본 개념이 되고 있다[13].

앨런 튜링이 제시한 인공지능은 컴퓨터에게 지적인 능력을 부여하기 위한 제반 분야로서, 우리 일상생활의 여러 영역에서 다양하게 활용되고 있으며, 앞으로 지능로봇·음성인식·영상인식·문자인식·자연언어처리·전문가시스템 관련 기술은 진보를 거듭하여 인간을 보다 편리하게 하고, 풍요로운 세상을 만드는데 기여할 것이다.

앨런 튜링이 연구한 암호해독 분야는 현재 대칭키 암호, 프로토콜 암호, 공개키 암호, 시스템 보안, 무선모바일 보안, 멀티미디어 보안, 금융 보안, 해킹대응기술, 신호정보해독 등에서 활발히 연구를 진행하고 있다.

4 결론

오늘날의 컴퓨터는 수학과 공학, 그리고 논리적 사고가 혼합된 발명품으로서, 공학자들만의 발명품이 아닌 수많은 수학자와 논리학자들의 결정체임을 밝히는 것이 본 논문의 목적이다. 이를 위하여 첫째, 계산기에서 컴퓨터로 진화하는 과정에서 컴퓨터 발전과정을 수학자들을 기준으로 구분해 보았고 둘째, 컴퓨터 발전과정에서 그 시대의 수학자의 역할을 하드웨어와 소프트웨어(프로그램)로 각각 구분하여 분석해 보았으며 셋째, 컴퓨터과학 분야의 성장 및 개발에서 수학의 중요성을 2장에서 언급한 수학자들만을 한정하여 파악해보았다.

특히, 본 논문에서는 수학자인 네이피어, 파스칼, 라이프니츠, 배비지, 불, 프레게, 페아노,

4) 관계 대수는 합집합, 교집합, 차집합, 카티션 곱(cartesian product)인 집합 연산으로 이루어져 있고, 릴레이션은 카티션 곱의 부분집합이다.

화이트헤드, 러셀, 힐베르트, 괴델, 앨런 튜링, 폰 노이만에 이르기까지 순차적으로 그들의 업적을 살펴보았다.

최초의 전자식 컴퓨터가 개발된 이후 오늘날까지 컴퓨터와의 관련된 기술은 눈부실 정도로 발전해왔다. 컴퓨터의 발전은 앞으로도 인류의 역사와 함께 계속 진보할 것이지만, 수세기에 걸친 컴퓨터의 발전 과정을 살펴보는 것은 현재와 미래의 수학 및 컴퓨터 기술 발전의 이해에 큰 도움이 될 것이라고 사료된다.

오늘날 수학은 모든 학문분야에서 중요한 도구로서 사용된다. 수학을 과학, 공학, 경제, 의학 분야들에 적용한 응용수학은 그 결과로서 수학자체의 발전을 이끌고 새로운 분야를 발생시켰다. 특히, 수학적 발전과 과학적 방법들의 상호작용에 의하여 컴퓨터 분야의 연구들이 진행되었고 인류문명에 큰 영향을 미치게 되었으며 이는 현재까지 계속되고 있다.

References

1. Martin DAVIS, *The Universal Computer The Road from Leibniz to Turing*, (Korean translation by PARK J. I. and JANG Y. T.), Knowledge of Landscape, 2006. 박정일, 장영태 역, 수학자 컴퓨터를 만든다, 지식의 풍경, 2006.
2. KIM D. S., *Introduction to Computers*, Sengneung Publisher, 2013. 김대수, 컴퓨터 개론, 생능출판사, 2013.
3. LEE S. W., An Analysis on the Relationship of Mathematics/Statistics in Software and Hardware Fields, *The Mathematical Education* 47(4) (2008), 505–517. 이승우, 소프트웨어 및 하드웨어분야의 트랙과 수학/통계와의 연관성 정도 파악, 한국수학교육학회지 시리즈 A <수학교육> 47(4) (2008), 505–517.
4. LEE S. W., A Curriculum Analysis with respect to Computer Programming, *The Korean Journal for History of Mathematics* 22(2) (2009), 69–80. 이승우, 컴퓨터프로그래밍과 관련된 교과목 내용분석, 한국수학사학회지 22(2) (2009), 69–80.
5. LEE S. W., Exploration of Teaching Method through Curriculum and Teaching Case connecting Math/Stat: Focused on Software Field, *The Mathematical Education* 50(3) (2011), 385–396. 이승우, 수학/통계와 연계된 교과과정과 교육사례를 통한 교육방안의 연구—소프트웨어 교과과정을 중심으로—, 한국수학교육학회지 시리즈 A <수학교육> 50(3) (2011), 385–396.
6. YUN N. I., LEE K. W., *Computer Architecture*, Sengneung Publisher, 2013. 윤남일, 이강우, 컴퓨터구조론, 생능출판사, 2013.
7. 괴델의 불완전성 정리, <http://navercast.naver.com>
8. 기호 논리학, <http://terms.naver.com>
9. 네이피어, <http://terms.naver.com>
10. 논리주의, <http://terms.naver.com>
11. 라이프니츠, <http://ko.wikipedia.org>
12. 러셀, <http://navercast.naver.com>
13. 앨런 튜링, <http://terms.naver.com>
14. 파스칼, <http://100.daum.net>

15. 페아노, <http://navercast.naver.com>
16. 프레게, <http://navercast.naver.com>
17. 형식주의, <http://terms.naver.com>
18. 힐베르트, <http://100.daum.net>