논리학자, 기호, 컴퓨터 - 027.013 Homework 1

경제학부 [redacted] 김서영

2012-09-24

1 서론: 주판에서 스마트폰 국민게임까지

요즘 한국에서 볼 수 있는 진풍경은 지하철/버스 안에서 사람들이 일제히 핸드폰에 얼굴을 바짝 대고 똑같은 게임을 하고 있는 모습이다. 한창 잘나가고 있는 '애니팡'이라는 선데이토즈 사의 카카오톡 내 퍼즐게임 어플리케이션 때문이다. 이 게임은 동물을 움직여 가로나 세로로 세 마리 이상 붙이면 동물들이 사라지면서 점수가 오르며, 1분이라는 제한시간 안에 얼마나 더 많은 점수를 낼 수 있는지 겨루는 게임[1]이다. 다음은 애니팡에 관한 아시아경제의 21일자 기사를 인용한 것이다.

> 21일 대신증권 김회재 애널리스트 분석에 따르면 데이터 사용량을 기초로 한 애니팡 하트의 가격은 개당 5원으로 추정된다. 애니팡은 서버에 접속해서 아이템 등이 소비되는 방식으로, 한 게임에 평균 93KB 데이터, 약 5원이 소모된다. 애니팡의 고정 사용자 800만명이 매일 한 건의 하트를 전송한다고 가정했을 때, 데이터 통신에 소비되는 비용은 하루 평균 4억원인 셈이다. 이들 사용자가 하루 제한된 전송량 50건을 모두 사용한다고 가정했을 때는 무려 20억원에 달한다. 애니팡 관계자는 "애니팡은 카카오톡 네트워크를 기반으로 애니팡에서 만날 수 있는 친구 수는 수십명에 이른다"며 "동시접속자수 300만명, 게임 설치 사용자 수 1500만명인 점을 감안했을 때 하트 유통으로 이뤄지는 데이터 가치는 어마어마할 것"이라고 말했다. [2]

[1] http://www.thisisgame.com/board/view.php?id=1286144&category=102&subcategory=
[2] http://www.asiae.co.kr/news/view.htm?sec=it5&idxno=2012092109181202091

<그림 1> 애니팡 화면 예시

작고 앙증맞은 동물들이 어지러이 놓인 화면에서 민첩하고 성실하게, 동물들을 한 칸씩 한 칸씩 움직이게 하는 게임의 로직을 따라가다 보면 점수가 오르며 게임의 목표에 도달해간다. 이 게임을 보면 마찬가지로 한 번에 한 단계씩, 셀과 상태와 행동표와 동작을 오가며 꾸준히 알고리즘 구현을 위해 작동하는 튜링 기계가 생각날 뿐 아니라, 1936년 그 튜링기계의 발명에서부터 불붙어 2012년 고작 스마트폰 내 오락 프로그램들 중 하나가 20억원의 경제적 가치를 창조하며 1500만명을 쥐락펴락 하는 경지까지 컴퓨터의 발전과정을 곱씹게 된다.

컴퓨터의 역사는 논리를 자동적으로 구현하려던 역사이다. 약간 논란의 여지는 있지만, 초대의 컴퓨터는 주판(abacus)이었다.[3] 주판은 기원전 2700년 메소포타미아까지 그 유래를 찾아낼 수 있으며[4] 수천 년 동안 인류의 계산기 역할을 해왔다. 심지어 대한민국에서는 주산이 상업고등학교의 교과목 중 하나이자 검정 시험을 통한 취업 입문턱이었던 60년대 이후에도 오늘날 여전히 아이들 교육 완구로 적극 활용되고 있다. 하지만 결국 주판을 통해 실제 계산을 하는 것은 주판을 다루는 사람이지 주판 자체가 아니므로 주판을 자동화된 컴퓨터라는 도구의 시초로 보기엔 약간의 무리가 있다.

컴퓨터의 좀 덜 알려진 초기 역사로는 1623년 독일의 빌헬름 쉬카르트(Schickard)라는 튀빙겐 대학 교수의 시계 계산기(calculating clock)가 있다.[5] 쉬카르트는 천문학자 케플러에게 보내는 편지에서 숫자를 입력하여 계산을 돕는 이 기계에 대해 이야기하고 있다. 하지만 이 기계가 실제로 작동을 했는지는 30년 전쟁 때 도면이 불타 없어지면서 영원히 미스터리로 남고 말았다. 쉬카르트의 기계 장치가 디지털 컴퓨터의 시초였다면 숫자가 아닌 다른 양적 측량치를 이용한

[3] Paul Strathern (1999), Turing and the Computer: The Big Idea, Anchor Books, p. 13

[4] Ifrah, Georges (2001), The Universal History of Computing: From the Abacus to the Quantum Computer, New York: John Wiley & Sons

[5] Marguin, Jean (1994) (in fr). Histoire des instruments et machines à calculer, trois siècles de mécanique pensante 1642-1942

아날로그 컴퓨터의 시초는 윌리엄 오트레트 (Oughtred)가 발명한 계산자(slide rule) 이며 이 또한 초보적인 계산기의 형태이다. 네이피어(Napier)의 연구를 발전시켜 두 개의 자를 이리저리 밀어 계산하는 이 도구는 차후 와트(Watt)나 만하임(Mannheim)에 의해 발전되어 보급되었다.[6]

하지만 쉬카르트 20년 후 등장한 블레즈 파스칼(Pascal)이야말로 세금 관리였던 아버지를 위해 19살의 나이에 덧셈과 뺄셈이 가능한 가산기(加算機)를 발명함으로써 컴퓨터 발전의 첫 주춧돌을 놓았다. 파스칼린(pascaline)이라고도 불렸던 이 엄청나게 복잡했던 계산기는 1645년에 발명되었고, 당시 공학의 한계로 인해 아직 초보적인 계산밖에 할 수 없었다. 특히 덧셈 또는 뺄셈을 연쇄적으로 함으로써 겨우 곱셈을 할 수 있었으며, 후대의 학자들이 이런 한계를 극복하기 위한 니즈를 인식함으로써 컴퓨터 역사의 본 막이 올랐던 것이다.[7] 본문에서는 라이프니츠에서 튜링까지 이어지는 컴퓨터의 초대 역사를 살펴보겠다.

2 라이프니츠(Leibniz)

수학자이자 철학자인 고트프리트 빌헬름 라이프니츠(Leibniz)는 1646년에 태어났고, 자라면서 아리스토텔레스(Aristotle)의 논리학 체계에 매료되었으며 이는 그가 계속해서 기호 및 기호화 그리고 기호들을 다루는 논리체계를 개발하는데 집중하는 길로 이끌었다. 특히 그는 인간의 추론(reasoning)을 수학처럼 오류를 곧장 찾을 수 있어 논쟁을 종식시킬 수 있는 형태로 만드는 것을 원했다.

> The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right.[8]

라이프니츠는 수학 연구를 하다가 1673년 파스칼린에서 더 발전한 곱셈과 나눗셈까지도 가능한, 즉 사칙연산이 가능한 계산기를 발명하였다. 또한 그는 기계를 통해 사칙연산뿐만 아니라 대수방정식을 풀고 또 논리적 추론을 기계적 절차로 변환하여 기계적으로 추론할 수 있게 하는 장치들을 꿈꾸었다. 또한 후에 라이프니츠는 미적분 기호 체계를 세우며 언어의 알파벳과 같이 의미 없는 소리가 아닌 개념을 표현하는 실제 기호체계(real characteristic)의 중요성을 더욱 실감하였다. [9] 비록 그가 후원 가문인 하노버 공작 가문의 가계사를 쓰는데 남은 인생의 대부분을 허비하게 되긴 했지만, 라이프니츠는 수학뿐만 아니라 인간의 모든 사고를 기호 체계로 담기 위한 노력을 끊임없이 하였다. 그가

[6]Paul Strathern, Op. cit, p. 16-19

[7]Ginsburg, Jekuthiel (2003). Scripta Mathematica (Septembre 1932-Juin 1933). Kessinger Publishing

[8]G.W. Leibniz (1685) The Art of Discovery, Wiener 51

[9]마틴 데이비스 (2005), 수학자, 컴퓨터를 만들다: 라이프니츠에서 튜링까지, 지식의 풍경, p.17-23

만들어낸 추론 계산법(calculus ratiocinator) 즉 항들의 연산인 A⊕A 등을 골자로 하는 논리 대수는 인간의 모든 논리를 대수학적으로 환원할 수 있게 해 준 매개적 연산체계였다.

3 불(Boole)과 프레게(Frege)

라이프니츠의 아이디어를 역사적으로 발전시킨 조지 불(Boole) 또는 부울은 수리 논리학의 아버지가 되었다. 학교 선생님을 하며 겨우겨우 생계를 잇던 불은 틈틈이 수학을 연구하며 대수(algebra)가 갖고 있는 무궁무진한 가능성에 대해 탐구하게 되었다.

미리엄-웹스터(Merriam-Webster) 사전에서 제시하는 대수의 정의는 다음과 같다:

1. a generalization of arithmetic in which letters representing numbers are combined according to the rules of arithmetic
2. any of various systems or branches of mathematics or logic concerned with the properties and relationships of abstract entities (as complex numbers, matrices, sets, vectors, groups, rings, or fields) manipulated in symbolic form under operations often analogous to those of arithmetic.

특히 컴퓨터의 역사를 탐구하기 위해서는 2번째 정의에 주목할 필요가 있는데, 요컨대 대수라는 것은 추상적인 개념을 기호로 잡아내어 마치 산수처럼 그 관계와 속성을 공리 등으로 시스템화하는 수학이나 논리학인 것이다. 대수의 어원인 al-jabr는, 아랍어의 정관사 'al'에 '흐트러진 것을 묶음'의 의미를 갖는 'jabr'라는 말의 결합어[10]라는 것을 고려한다면, 체계가 없던 것에 산술적인 체계를 부여하는 것이 대수의 본질이라는 것을 쉽게 알 수 있다. 특히 부울은 수량과 연산을 나타내는 기호가 얼마 안 되는 기본 규칙이나 법칙을 따른다는 사실에서 대수의 위력이 비롯된다는 것을 깨달았다고 한다.[11]

라이프니츠보다 부울이 한 발 앞서나갔다고 판단할 수 있는 근거는 그의 논문과 그 논문이 발전된 책인《사고의 법칙 The Laws of Thought》또는 원제인《논리와 확률의 수학적 이론의 기초가 되는 사고의 법칙 연구 An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities》때문이다. 불은 "모든 말은 포유동물이다"와 같은 낱말의 논리들은 낱말에 등장하는 개체들의 집합으로 표현할 수 있으며, 집합의 논리는 다시 숫자 연산으로 환원할 수 있음을 밝혀내었다. 이를테면 x가 "흰 것들"을 대표하고 y가 "양"을 대표하면 xy는 "흰 양"에 속하는 집합을 나타내게 되는 것이다. [12] 부울 이 당시 제기한 규칙들에는 (1) 곱셈은 교집합, (2)

[10] http://en.wikipedia.org/wiki/Algebra
[11] 마틴 데이비스, Op. cit, p. 41
[12] 마틴 데이비스, lbid, p. 44

덧셈은 합집합, (3) 뺄셈은 차집합, (4) X=1은 명제 X가 참, X=0는 명제 X가 거짓이라는 것 등이 있으며 원문은 통계학에서 스피노자까지 방대한 저술을 포함하고 있다. 이 법칙이 활용되는 예시를 들어보자면, 이를테면 X=1이고 Y=1이면 XY=1인데, 명제 X가 참이고 명제 Y가 참이라면 명제 X와 Y는 동시에 참이라는 것을 수학 연산을 아는 사람이라면 누구나 풀어낼 수 있게 된 것이다. 차후에 좀 더 발전한 불 대수 또는 부울 대수, 즉 추론적 논리문제에 적용할 수 있는 수학적 기본 법칙과 규칙의 핵심은 AND, OR, NOT이 되었다. 부울 대수로 우리는 아리스토텔레스의 삼단 논법뿐 아니라 전제와 결론을 갖고 있는 기본적인 논리 체계를 증명할 수 있다. 하지만 아직 부울 대수는 모든 수학적 추론을 포괄하기엔 부족했으므로, 논리학의 추가적인 발전은 차후 프레게에 의해서 이루어졌다.

그 후 1938년 미국의 섀넌은 전기회로의 스위치가 ON, OFF의 두 상태를 갖는 점에 착안하여 전기적 스위치회로가 부울 대수에 의해 표시될 수 있음을 보여주었다.[13] 또한 미리엄-웹스터 사전에서 부울 대수(Boolean Algebra)의 정의는 다음과 같다.

> Symbolic system used for designing logic circuits and networks for digital computers. Its chief utility is in representing the truth value of statements, rather than the numeric quantities handled by ordinary algebra. It lends itself to use in the binary system employed by digital computers, since the only possible truth values, true and false, can be represented by the binary digits 1 and 0. A circuit in computer memory can be open or closed, depending on the value assigned to it, and it is the integrated work of such circuits that give computers their computing ability. The fundamental operations of Boolean logic, often called Boolean operators, are “and,” “or,” and “not”; combinations of these make up 13 other Boolean operators.

우리는 여기서 불이 컴퓨터 발전에 지대한 공헌을 했음을 알 수 있다.

하지만 아직 불의 논리체계는 세밀한 추론을 포함할 수 없어, 고틀로프 프레게(Frege)라는 독일의 수학자의 도움을 필요로 했다. 그의 《개념 표기법 Begriffsschrift》은 불의 저작에서는 없던, ’모든’을 뜻하는 보편 양화사(universal quantifier)인 ∀ 및 ’존재함’을 의미하는 존재 양화사(existential quantifier)인 ∃를 포괄하고, “$\cdots$ 이면 $\cdots$ 이다”는 ⊃로, “$\cdots$ 이고 $\cdots$ 이다”라면 ∧로, “$\cdots$ 또는 $\cdots$” 는 ∨로, 그리고 “아니다”는 ¬로 표기한다. 따라서 부울 대수로는 표현할 수 없는 “모든 실패한 학생들은 어리석거나 게으르다”라는 명제는 (1) x는 실패한 학생이다를 F(x)로, (2) x는 어리석다를 S(x)로, (3) x는 게으르다를 L(x)로 놓았을 때

$$(\forall x)(F(x) \supset S(x) \vee L(x))$$

[13]Claude Shannon, "A Symbolic Analysis of Relay and Switching Circuits," unpublished MS Thesis, Massachusetts Institute of Technology, Aug. 10, 1937.

와 같이 표현할 수 있는 것이다. [14] 이렇게 프레게는 모든 명제를 다룰 수 있는 새로운 논리 언어를 창작했지만, 너무 복잡한 나머지 안타깝게도 라이프니츠가 원하는 ’계산’까지 가능하게 하는 언어는 아니었다.

4 칸토어(Cantor)

컴퓨터의 발명을 위해서는 논리학의 발전뿐만 아니라 기존에 존재하던 수학의 영역 확장도 필요했다. 게오르그 칸토어(Cantor)는 실(actual) 무한에 관해 체계적인 수학 이론을 정립한 것에서부터 나아가 집합론의 창시자로 가장 유명하다. 그가 일생에 걸쳐 증명한 산술에서 ’무한’의 역할은 차후 튜링의 만능 컴퓨터 발명에 큰 공헌을 했다. 삼각급수 및 극한에 대한 연구 중에 무한 집합을 완결된 전체로 다루고 관련된 복잡한 계산을 수행해야[15] 했던 칸토어는 (1) 무한 집합들이 여러 크기를 갖고 있다는 직관적으로 받아들이기 어려운 사실과 (2) 실수 집합과 자연수 집합은 일대일 방식으로 대응될 수 없다는 사실을 밝혀내고, (3) 알레프-널 등의 초한이라는 무한 집합의 기수 개념 (4) 대각선 방법 등의 업적을 세웠다.

특히 대각선 방법(diagonal process)은 (2)를 증명하는 데 핵심적인 역할을 했을뿐만 아니라 후술하겠지만 괴델을 거쳐 튜링이 결정 문제에 관한 직접적인 증명을 하는 초석이 되었다. 대각선 방법이란 집합 내 원소들 종류로 그 집합에 이름표를 붙였을 때, 이름표를 붙였던 모든 집합들과 다른 어떤 새로운 집합을 얻는 데 쓰일 수 있는 방법”이다. 이를테면 각각 자연수의 집합인 M_i ($i \in N$)가 i라는 이름표를 갖고 있다고 가정했을 때 M_1부터 M_2, M_3, ... 는 자연수와 가능한 모든 일대일 대응을 나타냄에도, i가 M_i에 속하지 않을 경우만 i가 속할 수 있다는 전혀 새로운 집합인 M을 창조할 수 있어, 이와 같은 어떤 대응도 자연수들의 모든 집합들을 포함할 수 없다는 것이 증명되었다.[16]

프레게가 한 말 처럼 “무한이 이 인식론적 경향과 공존할 수 있는 길은 없” 음에도 “산술에서 무한의 역할은 부정되지 않을 것”[17]이기에 칸토어의 이론은 무한을 신의 영역으로 치부하던 기존 수학계에 엄청난 비판과 반향을 불러일으켰다.

5 힐베르트(Hilbert)와 괴델(Gődel)

독일 수학자인 다피트 힐베르트의 주요 업적 중 하나는 수학 기초론(foundations of mathematics)로써 그는 기하학의 소재인 도형을 보면서 직관적으로 알 수

[14] 마틴 데이비스, Op. cit, p. 73-77
[15] 마틴 데이비스, lbid, p. 94
[16] 마틴 데이비스, lbid, p. 110-111
[17] 마틴 데이비스, lbid, p. 118

있는 것에 의존하지 않고 공리들의 집합에서 정리들을 유추할 수 있어야 한다고 결론지었고 또 실제로 모순이 없고 일관성이 있는 공리들의 집합을 도출했다. 힐베르트는 따라서 유클리드 기하학의 무모순성을 산술의 무모순성으로 환원하였고 그 다음 단계로는 산술의 무모순성을 보이고자 했다.[18] 특히 그는 수학과 논리학을 완전히 형식적인 기호 언어로 환원하여, 기호 체계 외부에서는 의미를 고려하지 않은 채 조작할 수 있는 라이프니츠의 꿈을 실현하기 위해 노력하였다.

그 과정에서 그는 수학과 메타수학(metamathematics)를 구분하였는데, 메타수학은 형식화된 수학체계의 기호나 표현에 관한 명제들의 표현법이라고 할 수 있다. 가령 1+1=2 은 수학의 형식체계에 속하지만, 이 표현에 관한 주장인 "1+1=2은 수학의 한 명제다" 는 메타수학적 표현이다. 동시에 그를 비판하는 푸앵카레, 브로우웨르 및 바일의 논리를 넘기 위해 힐베르트의 계획(program)은 힐베르트의 결정 문제라고 알려진, 1차 논리학의 전제들과 제시된 결론이 연역적인 한정된 수의 명확하고 효과적인 단계들을 거쳐 타당한지 아닌지를 결정할 수 있는가를 증명하는 문제로 발전하였다. [19] 이는 획기적인 시도이긴 했으나 곧 괴델에 의해 뒤엎어질 것이었다.

쿠르트 괴델의 동시대인인 버트런드 러셀과 화이트헤드는 모든 수학을 논리학 속에 포괄할 수 있다는 것을 보여주었지만, 괴델은 여기서 메타수학을 연구 중에 형식 논리 체계 외부에서 바라보면 참이라고 보일 수 있지만 체계 내부에서는 증명될 수 없는 명제들이 있기 때문에 수학적 진리의 범위는 얼마나 강력한지와는 상관 없이 어떤 주어진 형식 체계에서 증명될 수 있는 것보다 더 넓다는 결론을 지을 수 있었다. 특히 그는 수학 원리(PM)를 주어진 형식 체계로 놓았을 때, “U는 PM 안에서 증명될 수 없다”라고 주장하는 명제 U가 (1) (외부에서 봤을 때에는) 참이지만 (2) PM 안에서는 본 명제도 부정 명제도 증명 가능하지 않은 특이한 명제라는 것을 보였다. 따라서 U는 본 명제도 부정 명제도 PM 안에서 증명할 수 없어 '결정 불가능한 명제'라는 이름을 갖게 되었으며 1930년 괴델의 발표는 무모순성을 증명하려던 힐베르트의 계획을 완전히 무너뜨렸다. [20] 수학에 관한 형식 체계는 본질적으로 불완전하다는 이 이야기는 괴델의 불완전성 정리(incomplete theorem)이라는 이름을 갖게 되었다.

후에 괴델은 라이프니츠를 읽으며 라이프니츠가 꿈꾸었던 보편 문자(characteristica universalis) 개념에 찬사를 보냈고, 라이프니츠가 꿈꾸었던 인간 이성의 계산으로의 환원은 두 세기가 지나서도 괴델을 통해 살아 숨쉬었다.

6 컴퓨터의 아버지, 튜링(Turing) 그리고 그 이후

1936년 튜링이 발표한 논문《계산가능수와 결정문제에 대한 응용에 관하여 On Computable Numbers, with an Application to the Entscheidungsproblem》는 그

[18]마틴 데이비스, lbid, p. 127-128
[19]마틴 데이비스, lbid, p. 141-146
[20]마틴 데이비스, lbid, p. 165-168

직전 해 그가 케임브리지에서 수학 기초론에 관한 뉴먼의 강의를 들었기에 탄생할 수 있었다. 괴델의 불완전성 정리를 접한 튜링은 그것을 증명하는 방법을 '계산 행위'로 증명하는 방법에 몰두하였다. 그는 사람이 기호를 종이 테이프 위에 작업하는 상황을 상정하고 모든 계산들을 다음과 같은 절차들을 통해 할 수 있도록 변환하였다.

- 계산은 네모칸이 그려진 종이 테이프 위의 네모칸 안에 기호를 써서 실행된다.
- 각 단계에서 계산을 수행하는 사람/기계는 정확히 이 네모칸들 중 하나에 적힌 기호에 주의를 기울인다.
- 사람/기계의 다음 행동은 이 읽어들이는 기호와 사람의 마음상태/기계의 설정값에만 달려 있다.
- 사람/기계는 위에 따라 테이프 위해 기호를 적을 것이고 그 다음에는 똑같은 네모칸을 계속 읽어 들이거나 테이프의 왼쪽이나 오른쪽으로 위치를 옮길 것이다.[21]

그가 튜링 기계라고 부르는 이 작업 플랫폼의 핵심은 무한히 긴 테이프를 갖고 있는 유한 상태 기계였다. 이런 수학적으로 추상적인 개념을 통해 튜링은 모든 수학적인 계산들을 튜링 기계에서 할 수 있다는 것을 증명했으며, 이것은 다시 말하면 튜링 기계로 할 수 없는 작업이라면 그 계산을 결정할 수 있는 알고리듬 절차가 없는 것이나 마찬가지인 것이다. 설정값이 없는 경우, 즉 작업이 완수되어 최종 출력물을 얻은 경우 기계는 작동을 멈추게 되는데, 기계가 작동을 중지하게 하는 모든 작업들의 집합을 튜링 기계의 멈춤 집합(halting set)라고 부른다. 이 멈춤 집합에 칸토어의 대각선 방법을 응용하면 우리는 튜링 기계의 그 어떤 멈춤 집합과도 다른 집합을 만들어 낼 수 있다. [22] 그런데 계산 작업이 멈추지 않는다는 것은 결국 모든 문제가 계산가능한 것은 아니라는 것을 뜻하며 힐베르트의 계획을 무너뜨리고 괴델의 논지를 다시 한 번 증명하는 방법이 된다는 뜻이다.

지금까지 이어져 온 수학/논리학의 맥의 꽃이 이렇게 튜링과 튜링 기계로 피어났다면, 컴퓨터 자체는 이 논리학의 발전이 아닌 그 발전의 부산물로 탄생하게 되었다. 튜링은 자신의 작업의 타당성을 보이기 위해 모든 튜링 기계의 규칙표를 받아 시행하는 하나의 단일한 보편 기계(universal machine)을 고민하게 되었고, 1938년부터 독일 군대의 암호를 푸는 일을 하면서 블레츨리 파크에서 그 기계의 실제 설계를 위한 전자 공학적 기초들을 익혀나가기 시작했다. 이 때부터 컴퓨터의 역사는 본격적으로 순수 논리학이라는 하나의 기둥에 공학이라는 또 하나의 중요한 기둥까지 두 개의 결합으로 발전하기 시작했다. 만약 튜링이 테이프 위에 기호를 적는 '기계'라는 발상에 착안하지 않았다면, 또 논문에서 실제로 그 기계부품들을 공들여 정의하려 하지 않았다면, 우리는 발전된 수학은 얻을 수 있었으나 컴퓨터라는 것은 영영 알 수 없게 되었을 지도 모르는

[21] 마틴 데이비스, lbid, p. 209-210
[22] 마틴 데이비스, lbid, p. 221

일이다. 다행히도 실제로 이 기계를 공학적으로 구현하는 것에 튜링 및 다른 사람들이 착안하게 됨으로써 우리는 계산 능력(computation)의 자동화의 시대를 열게 된 것이다.

아마 튜링의 논의를 접한 것으로 추정되는 존 폰 노이만(Neumann)은 논리학을 토대로 존 프리스퍼 에커트 주니어라는 공학자와 함께 18,000개의 진공관으로 이루어진 당시 최고의 계산기인 에니악(ENIAC: Electronic Numerical Integrator and Computer)을 탄생시킬 수 있었다. 튜링의 보편 기계의 핵심은 말할 것도 없이 '보편성'이다. 이는 바꾸어 말하면 컴퓨터의 발전은 이제 속도와 메모리 크기 외에는 없다[23]는 뜻이 된다. 수많은 특허 논쟁과 튜링의 고용 및 자금난으로 인한 우여곡절 등이 있었지만 에니악은 시간의 흐름에 따라 수많은 공학자들 및 논리학자들에 의해 에드삭(EDSAC: Electronic Delay Storage Automatic Calculator), 에드박(EDVAC: Electronic Discrete Variable Automatic Computer), 유니박(UNIVAC: Universal Automatic Computer) 등으로 발전해가며 실제 오늘날 우리가 볼 수 있는 컴퓨터의 모태가 되었다.

7 결론: 거인들의 어깨에 서기

"내가 더 멀리 보아왔다면, 그것은 거인들의 어깨 위에 서 있었기 때문이오." 1676년 아이작 뉴턴은 로버트 훅에게 보낸 편지에 이렇게 쓴 바 있는데, 이것은 과학을 비롯한 문명 전체가 그 이전에 이루어진 성과 위에 새롭게 구축되는 일련의 누적적인 진보라는 점을 지적해주는 말[24]이다. 얼른 보기에 공학의 결정체라고만 보이는 컴퓨터는 사실은 논리학자들/수학자들의 400년 간의 사유의 결정체라고도 할 수 있다. 즉 인간 두뇌 및 정신적인 능력의 정수(essence)만 논리 또는 기호체계로 압축된 과학적 논리로 잡아내는 것이 가능한지, 가능하다면 어떻게 가능한지, 불가능하다면 왜 불가능한지에 대한 400년 간의 대화의 소산인 것이다. 즉 컴퓨터를 가능하게 한 것은 일상적인 편의를 도모하기 위한 아주 실용적인 이유에서라기보다는 인간의 정신이란 무엇인가에 대한 훨씬 철학적이고 형이상학적인 논의, 모순이 없는 진리란 무엇인가에 대한 논의인 것이다. 다시 말하자면 컴퓨터에서 기술은 부차적인 것이다. 또는 기술보다는 '아이디어'가 컴퓨터의 핵심[25]이라고도 이야기할 수 있을 것이다.

필자의 손에 쥐어진 스마트폰, 이 작은 '컴퓨터'에는 한 입 베어문 사과 로고가 그려져 있다. 공교롭게도 컴퓨터 초대 역사의 거장인 튜링이 청산가리를 주입한 사과를 베어물고 자살하였다고 전해져 이 유명 컴퓨터 회사의 로고는 튜링을 기린 것이 아니냐는 소문도 있었으나, 창립자의 전기에 따르면 이 사과는 성경에 등장하는 진리의 과일으로서의 사과이다. 즉 한 입 베어문 사과는 지식의 습득(acquisition of knowledge)를 상징하는 것이다.

컴퓨터의 역사를 이해하는 것은 단순히 우리의 삶의 많은 부분을 의존하는 기계에 대한 이해를 넘어 어떤 지식을 얻게 해 준다. 지금까지 라이프니츠로부터 살펴본 400년의 역사를 통해 거장들이 일부는 동시대에서, 일부는 시대를

[23] 대니얼 힐리스 (1998), 생각하는 기계, 사이언스북스, p. 117
[24] 스티븐 호킹(2006), 거인들의 어깨 위에 서서 - 물리학과 천문학의 위대한 업적들, 까치글방
[25] 대니얼 힐리스 (1998), Op. cit, p. 12

뛰어넘어 대화를 해 왔고, 인간 이성의 가장 순수한 부분을 기호와 타당한 규칙이라는 가장 단순한 형태로 환원하였다. 역사의 오랜 도움닫기를 통해 20세기 후반부터 컴퓨터의 발전은 급격한 도약을 이루었으며, 반세기 정도의 시간을 거쳐 2012년 현재의 발전 상태까지 다다르게 되었다. 400년 간 학자들이 주장하고 반박하고 때로는 격렬하게 싸우면서까지 치열하게 그 환원 과정을 검증하고 또 사유하지 않았더라면 그렇게 급속도의 발전은 없었으리라고 생각된다. 이제 우리는 우리에게 너무나 친숙한 도구가 본질적으로 인간에게 무엇인지에 대한 질문 없이 단순히 하드웨어와 소프트웨어를 활용할뿐이지만, 인간이 여기까지 오기 위해 얼마나 많은 거인들의 어깨에 올라섰는지 알아보는 것은 필수적인 공부가 될 것이다.