그들만의 것이 아니었던 - 우리의 인생이라는 회로

자유전공학부 양태훈

1. 그들만의 것이 아니었다.

2000년. New millenium 시대가 시작된 해. 그 시절 나는 초등학교 3학년이었고 아파트 뒤편에 있는 나라 컴퓨터 학원에 다녔다. 그리고 그 당시 유행처럼 퍼져있었던 대한상공회의소에서 주관하는 워드프로세서 1급 자격증을 취득했다. 그것을 위해 간단히 배웠던 초기 컴퓨터의 모습은 아직도 내 뇌리 속에 떠나지 않고 깊숙이 박혀 있다. 디지털 원주민(Digital Native)[1)]이든 디지털 이주민(Digital Immigrant)[2)]이든 관계없이, 디지털 시대를 경험하기는 한 사람이라면 최초의 컴퓨터에 대해 대부분 '진공관' 혹은 '에니악(ENIAC)'이라고 말할 것이다. 그리고 컴퓨터는 공학자(Engineer)들의 연구의 산물이라고 말할 것이다.

『The Universal Computer』를 읽기 전까지는, 나도 별반 다르지 않았다. Active X에 불만을 가지지만 결국 그것을 접고 설치를 누르기만 하는 정도의 관심을 가진 사용자로서, 애플의 스티브 잡스는 알아도 C언어의 선조인 데니스 리치에는 문외한인 것처럼, 에니악은 마르고 닳도록 들었지만 튜링 기계는 '?'를 짓게 하는, 그런 것이었다. 하지만 튜링 기계는 현대적 컴퓨터의 명백한 선조이고, 튜링은 그것을 제시한 인물이다. 그리고 이 책에 따르면, 튜링이 1937년 ≪On Computable Numbers, with an Application to the Entscheidungsproblem≫이라는 논문을 발표하기까지 그 기반을 닦고 소스를 제공했던 수많은 저명한 학자들이 있었다. 특기할만한 것은 그 사람들은 공학자가 아닌 논리학자 혹은 수학자들이었다는 것이다. 그렇게 보면 이 책의 이야기는 진정한 컴퓨터 개발사(開發史)라고도 할 수 있다. 튜링은 어떤 바탕들에서 상기한 논문을 썼고, 그 과정에서 현대적 컴퓨터의 모체를 만들 수 있었을까. 이는 먼 과거, 17c 라이프니츠(G. Leibniz)로 거슬러 올라간다.

■ 기호, 논리, 대수 - Leibniz, Boole, Frege

컴퓨터, 그리고 그것의 모체인 튜링 기계는 논리적 연산을 기계적으로 수행하는 물체이다. '컴퓨터 과학이 여는 세계' 수업을 듣고 사후적으로 생각하게 된 것이지만, 컴퓨터에 있어 핵심은 클릭감 좋은 Logitech 마우스도, 최신 게임이 원활히 돌아가는 VGA카드도 아니다. 컴퓨터는 기계적으로 논리 연산을 수행하는 물체에 다름 아니다. 따라서 튜링에게 기호논리학적 기반을 제공한 인물들을 탐구해 보는 것은 충분한 당위성을 지닌다.

최초의 시발(始發)은 라이프니츠(Gottfried Leibniz)다. 흔히 미적분 기호의 창시자로 알려져 있는 그는 아리토텔레스(Aristotele) 논리학에 강렬히 매료되었다. 그는 인간의 모든 추론들을 아주 단순한 기호들 몇 가지와 그것들의 논리적 연산 체계로 표현하기를 원했다. 그가 미적분 기호를 제시한 것에서 알 수 있듯이, 그는 기호 자체가 단순히 표음적인 성격을 띠는 것이 아니라, 표의적이기를, 즉 개념을 표시하기를 바랐다. 그것을 그는 실제 기호체계(real characteristic)라고 불렀다. 그러한 실제 기호체계를 통해 인간의 모든 사고를 몇 종류 안 되는 단순한 기호들 간의 관계로 표현하고자 하였으며, 추론계산법(calculus ratiocinator)과 같은 대수학적 환원은 그의 이런 생각을 잘 드러내 준다.

수업시간에도 비중 있게 다루었던 부울(George Boole)은 그러한 라이프니츠의 위대한 생각들을 조금은 실현시킨 존재라고 할 수 있다. 대한민국 의무교육과정을 이수한 사람이라면 누구나 사칙연산에 대해 알 것이다. 덧셈, 뺄셈, 곱셈, 나눗셈. 정말 간단한 산술 관계를 나타내

1) 태어났을 때부터 발달된 디지털 문명을 겪은 세대.
2) 태어나 어느 정도 성장한 후 디지털 문명으로의 전환을 겪은 세대.

는 것들이다. 이는 간단한 대수 연산인데, 부울은 이러한 간단한 대수 연산 관계로 많은 논리 명제들을 표현했다. 결국 그가 핵심적으로 사용한 것은 and, or, not(그리고, 또는, ~이 아니다)이며, 조금의 차이는 있지만 우리가 흔히 산술 연산에서 사용하는 교환법칙, 결합법칙, 분배법칙 등을 논리 관계 표현 단순화에 활용했다.

백문이 불여일견이라는 말도 있으므로, 내 나름대로 스스로 경탄해 마지않았던 예시를 보자. 1이 True, 0이 False라고 가정하고 XY=1이라고 하자. Y=1-X라고 한다면 대입법에 의해 X(1-X)=1이 되며, 이 식을 참으로 확정하기 위해서는 X=(1-X)=1이 되어야 한다. 왜냐하면 X와 1-X는 곱셈, 즉 and로 이어져 있기 때문이다. 그런데 X가 1이 되면 1-X가 0이 되고, 1-X가 1이 되면 X는 0이 되기에 그러한 관계는 불가능하다. 따라서 임의의 집합 X와 전체집합 차집합 X, 즉 X의 여집합(negate X)는 동시에 참이 될 수 없다는 결론에 이르게 된다. 이는 개체들의 집합화와 그것의 대수 연산 환원을 통해 단순한 기호들로 표현된 대수식 자체가 중요한 논리적 관계를 표현할 수 있게 되었음을 나타내 준다. 단순하지만 그 힘에 놀라지 않을 수 없다.

부울은 이렇게 놀라운 성과를 이루었고, 그의 생각들은 부울 대수(Boolean Algebra)라는 이름으로 전해지고 있다. 하지만 아쉽게도 이는 복잡한 명제의 논리 관계는 나타내지 못했다. 이러한 한계점을 보완한, 논리학의 콜럼버스로 불리고 있는 인물이 바로 프레게(G. Frege)이다. 그는 산수의 진리들을 논리의 진리들로 환원했다. 부울 대수에서 없었던 보편양화사 '∀', 존재 양화사 '∃', ~이면 …이다의 포함관계를 나타내는 '⊂' 등이 추가되었고 따라서 보다 복잡한 명제들까지 표현할 수 있는 논리 체계를 마련했다. 프레게를 종착역으로 튜링의 논문을 위한 기호논리학적 기반은 어느 정도 마련된 것이다.

■ 무한, 결정 문제, 불완전성 정리 - Cantor, Hilbert, Gödel

전술한 것들이 모두 튜링의 1937년 논문에 그리고 튜링 기계에 기초이자 근간이라고 할 수 있는 기호논리학적 기반을 제공했다고 할 수 있다면, 지금부터는 수학적인 기반을 제공한 경우라고 할 수 있다. 그리고 그것은 칸토르의 무한에 대한 논의로부터 시작된다.

칸토르(George Cantor)는 가히 혁명가라고 할 수 있다. 그 시절 다른 수학자들은 신의 영역이라고 부정하던 실무한(actual infinite)에 관해 수학적으로 정리하고 집합론을 창시한다. 그의 초한수나 실무한의 개념이 다소 복잡하여 유리된 것처럼 보인다면, 향후 튜링에게 직접적으로 영향을 준 대각선 논법(Diagonal Argument)을 통해 그의 생각을 조금은 엿볼 수 있다. 자연수 집합의 부분집합들을 M1, M2, M3,… 등으로 명명하고 이를 자연수 1, 2, 3 등과 일대일 대응 시킨다. 자연수 집합도 무한하고 자연수 집합의 부분집합의 개수도 무한이기에, 무한들 간에 대소의 편차가 없다면 둘을 일대일 대응 시키면 잔여 원소가 없어야 한다. 하지만 자연수와 자연수 집합의 부분집합 M1, M2, M3,… 등을 행과 열로 배열해 놓은 표에서 대각선 원소들만 '속한다'에서 '속하지 않는다', '속하지 않는다'에서 '속한다'로 바꿔 주면 그 어떤 자연수 집합의 부분집합과도 다른 부분집합이 나오게 된다. 따라서 이는 자연수와 자연수 집합의 부분집합들이라는 무한집합의 원소의 개수가 다르다는 것을 의미하며, 무한에도 대소가 존재한다는 것을 나타낸다. 이는 앞으로 튜링이 튜링 기계의 정지 문제(Halting Problem)를 해결하는 방안으로 활용하게 된다.

이어서 힐베르트(David Hilbert)는 수학기초론에 기여했는데, 그의 핵심 아이디어는 공리들의 집합에서 각종 정리들을 유추해낼 수 있어야 한다는 것이었고, 유클리드 기하학의 무모순성을 산술의 무모순성으로 환원한 후 산술의 무모순성을 증명해 보이려 하였다. 그러한 과정에서 수학 그 자체를 연구의 대상으로 하는, 형식화된 수학체계의 기호나 표현에 관한 메타수학(metamathematics)을 창안하였다. 튜링에게는 직간접적 영향을 미치는데, 직접적으로는 그가 제시한 23가지 문제 중 하나인 결정 문제(Entscheidungsproblem)가 튜링의 호기심을 끌었고 튜링이 결국 그 문제는 해결할 수 없다는 것을 제시했다는 점이며, 간접적인 영향으로는

그가 주창한 힐베르트 프로그램(Hilbert Program)이 괴델의 '불완전성 정리'를 통해 격파되었고, 그러한 괴델의 정리가 또 튜링의 정지 문제와 연관을 맺게 된다는 점이다.

괴델(Court Gödel)에 있어 알아야 할 것은 바로 '불완전성 정리(Incomplete Theorem)' 그것이 전부가 아닐까 한다. 그의 불완전성 정리는 제 1원리와 제 2원리로 요약된다. 1원리는 수학에는 증명도 부정도 되지 않는 명제가 반드시 존재한다는 것이고, 2원리는 수학의 근본 공리체계에 모순이 없다면 그 무모순성은 그 체계의 논리전개로 증명할 수 없다는 것이다. 간단히 우리의 일상 언어로 요약하자면, 이는 수학의 모든 것을 증명할 수 있다고 믿었던 수학자들의 생각은 틀렸다는 것이다. 증명이 불가능한 애매모호한 대상들이 존재한다는 것이며, 이는 전술했듯 수학의 공리 체계의 완전성과 무모순성을 증명할 수 있다고 주장하던 힐베르트 프로그램이 좌절되는 계기가 되었다.

■ 모두의 공헌 아래 - 튜링(Turing) 그리고 만능 튜링 기계(Universal Turing Machine)

이렇게 기호논리학적, 수학적 기반을 제공한 - 라이프니츠부터 괴델까지 - 학자들의 직간접적 영향을 받아, 튜링(Allan Turing)은 1937년 ≪On Computable Numbers, with an Application to the Entscheidungsproblem≫이라는 제목의 논문을 쓰게 되었고, 그것은 튜링 기계의 등장을 의미했다. 우습게도, 튜링은 튜링 기계를 목적으로 두고 논문을 쓴 것이 아니었다. 힐베르트의 결정 문제를 푸는 과정에서 '기계적인 계산(Computation)'의 그 자체로 튜링 기계를 제시했던 것이었다. 메모리 테이프, 상태 표시기, 규칙표 등으로 구성된 현대 컴퓨터의 모체, 튜링 기계는 그 구조와 모습은 단순하고 조금은 원시적이었으나 튜링 기계가 또 다른 튜링 기계의 입력 값으로 작용할 수 있는, 혁신적인 개념(현대 컴퓨터의 소프트웨어를 떠올리게 한다)을 담은 것이었다.

이로서 만능 튜링 기계(Universal Turing Machine)가 탄생했고, 이런 만능의 기계조차도 정지 문제(Halting Problem)를 해결할 수 없다는 것을 보임으로써, 기계적인 계산으로는 - 즉 수학적 알고리즘으로는 - 참/거짓을 판별할 수 없는, 증명 불가능한 문제가 존재하며, 따라서 힐베르트의 결정 문제는 해결할 수 없다는 결론을 내리게 된다. 이러한 과정에 활용된 튜링 기계의 정지 문제에 사용된 것이 바로 칸토르의 무한 논의, '대각선 논법'이다. 그 어떤 정지 집합(Halting Set)과도 다른 정지 집합을 만들 수 있다는, 즉 정지 집합의 부분집합들의 집합의 원소의 개수가 정지 집합 그 자체보다 많은 상황 속에서 발생하는 모순으로 인해 정지 문제는 증명이 불가능한(참/거짓 판별이 불가능한) 문제가 된 것이다.

지금까지를 정리해서 본다면 튜링의 논문, 그리고 튜링 기계는 힐베르트의 결정 문제를 해결하는 데에서 왔고 칸토르의 무한과 대각선 논법을 십분 활용했으며, 수학적 알고리즘의 방법으로 참/거짓을 판별할 수 없는 명제가 존재한다는 괴델의 불완전성 정리와 그 맥을 같이한다. 또한 그 근간에는 단순하지만 표의적인 특성 - 개념을 지니고 있는 - 기호들 간의 논리 연산 체계를 통해 인간의 모든 추론, 명제들을 표현하고자 했던 라이프니츠와 부울, 프레게의 노력이 담겨 있다. 이 모든 것이 종합된 것이 튜링의 1937년 논문이며, 그 과정에서 탄생한 것이 현대 컴퓨터의 모체인 만능 튜링 기계(Universal Turing Machine)인 것이다.

『The Universal Computer - From Leibniz to Turing』은 서두에서 제시하던 (어쩌면 나만의 편견이었을지도 모르지만) 흔한 일반 사람들의 편견을 바꿔 놓는다. 컴퓨터는 그것을 위해 몰두하던 공학자들만의 소산이 아닌, 수학적 문제를 해결하는 과정에서 하나의 '기계적 계산(Computation)' 그 자체로 등장했던 어찌 보면 우연한 물체였고, 그 기반에는 2차 대전 이후 발전한 눈부신 공학뿐만 아니라 기호논리학, 수리논리학적 기반이 깔려있었으니 말이다. 라이프니츠에서 튜링까지 이어진 위대한 학자들의 노력 덕택에, 우연이든 필연이든 우리는 그들의 공로를 기억하지도, 느끼지도 못한 채 튜링 기계의 최신 발전물인 미니컴퓨터 스마트폰을 마음껏 즐기고 있다.

2014년 봄 학기 컴퓨터 과학이 여는 세계 감상문 - 『The Universal Computer』

2. 인생(人生)이라는 회로

동양철학의 유가 사상에서는 인간을 소우주(小宇宙)라고 부른다. 여타 동물과는 다른 존재로 여겨졌던 인간은 하늘의 혼(魂)과 땅의 백(魄)을 지니고 있기에, 규모는 다르지만 우주(宇宙)와 운행 원리가 같다는 것이다.

'부자부터 가난한 사람까지, 사람 사는 게 다 비슷하지!' 흔히 어른들께서 쓰시는 거의 격언화된 말에서 엿볼 수 있듯이, 사실 굳이 고상하게 유가 사상을 들고 오지 않더라도, 우리는 흔히 일상 속에서 그것을 통찰하고 있다. 아주 엄밀한 경우를 따지게 되면 모르겠지만, 곁가지를 쳐낸 단순한 운행의 관점에서 관찰한다면 이 우주에서 인간 사회, 인간 사회와 동물 사회, 그리고 개별 인간의 삶들도 크기와 규모, 범위의 차이만 있을 뿐, 별반 다를바 없음을 알 수 있다.

『The Universal Computer』를 덮은 순간 이와 같은, 미시에서 거시를 총괄하는 전우주적 통찰을 하게 된 이유는 무엇일까. 그리고 며칠 전 배웠던 COQ, ML, Java, C, Arm 등의 프로그래밍 언어가 표현력은 같으나 표현의 복잡도만 다를 뿐이라는 것이 생각난 것은 왜였을까. 만능 기계 장치인 컴퓨터의 숨은 역사를 다룬 이 책에서 그것을 느낀 것은 바로 라이프니츠와 부울, 프레게가 정립했던 '근본 원리로서의 기호 논리학'에 기인한다.

그 중에서도 부울 대수(Boolean Algebra)에서 등장했던 'and, or, not'. 수업시간에도 디지털 논리 회로를 구성하는 데에 활용되어 연습해 보았던, 기초적인 논리적 관계를 나타내는 이들은 무언가 우리 인생의 원리를 나타내고 있었다. 디지털 논리 회로는 우리네 인생판, 크게는 역사판의 단순화된 버전이라고 생각되었으며 – 회로에 전류가 잘 흐르려면 and, or, not의 성격을 잘 파악해서 회로를 짜야 하듯이 – 어떻게 인생을 살아갈 것인가에 대한 지침을 제공해 줄 수 있었다. 물론 초기 컴퓨터 모델인 튜링 기계가 의도치 않게 발명되는 데에 수많은 논리학자, 수학자들의 공헌이 있었고, 이것은 의외였으며 그들의 공적을 인식한 것 또한 나의 감상 중 하나였음이 분명하나, 다소 거창한 인생에의 통찰은 이것에 우위를 점하는 것이었다.

and(그리고)는 협력과 필수불가결성을 상징한다. 디지털 논리 회로에서 and를 통해 전기가 흐르기 위해서는 두 전선 모두에 1의 값이 흘러야 한다. T*T=T, T*F=F, F*T=F, F*F=F의 경우에서도 알 수 있듯이 양측 모두가 T의 값이어야 T라는 결과 값이 산출된다.

or(혹은)은 다양성에의 포용이다. or에서는 양쪽 모두에 1의 값이 입력되지 않아도, 어느 한쪽에만 1의 값이 입력되면 전기가 흐르게 된다. T+T 뿐만 아니라 T+F, F+T의 경우에도, 즉 어느 한 쪽만 T의 값이어도 결국 산출 값은 T가 된다.

not(아니다)라는 관계는 부정이다. 전기가 흘렀어도 not이 있다면 이제 흐르지 않는다. 차단이고 포기이다. T의 값이 입력되더라도 not T이면 결국 F가 된다. 그리고 그 외의 나머지 전선들에 전기 흐름의 역할을 일임한다.

and에 전류가 흐르게 위해서는 모든 전선에 1의 값이 입력되어야, 즉 전류가 흘러야 하듯이 우리는 다른 사람들의 협력을 필요로 한다. 협력이 너무나 딱딱하게 들린다면, 사람들 간에 함께 살아감이 필요하다고 하면 좀 더 편안할 것이다. 복잡다단한 것을 차치하고 가장 단순하게, 그 자체로 매우 중요한 가치이고, 인간 사회에 있어 가장 작지만 핵심이 되는 공동체 단위인 가정(family)을 구성하는 감정인 사랑의 경우에 어느 한 사람만 다른 쪽을 사랑한다고 해서 그 관계가 이루어지고, 깊어질 수 있을까? 그러한 중요한 개념에 있어서 '같이'는 필수적이다. 인간이 하는 어떤 활동도 마찬가지이다. 스스로가 아무리 능력이 뛰어날지라도, 결국 인간이 단순한 일이 아닌 어느 정도의 복잡성을 가진 일을 도모할 때는 결국 타인의 도움이 필수적이다. 선택적(Optional)인 것이 아니다. and는 이런 것이다.

or은 and와 결합하여 공통의 가치를 추구하는 협력적 관계 속에서 거시적으로 나아갈 때 다양함에 대한 포용성을 지녀야 한다는 것을 상징한다. or을 통해 전류가 흐르는 방법은 한 가

지가 아니다. A를 통해서, B를 통해서 혹은 C를 통해서 흐를 수도 있다. 이는 우리네 인생이 흘러가면서 진리에, 행복에, 흔히 말하는 '잘 먹고 잘 사는 것'에 다다르는 방법이 한 가지가 아님을 나타낸다. 너의 방법도, 나의 방법도, 너의 노선도, 나의 노선도 모두 그 곳으로 향할 수 있음을 의미한다. 어떻게 사는 것이 옳은 것인가? 그것은 예수도, 부처도 확실히 제시해 주지 못했지 싶다. 어떤 사상이 옳은 것인가? 나의 사상만이 옳은 것인가? 독립적으로, 그 자체만으로 옳다고 주장할 수 있는 인생 경로가 있을까? 그렇지 않다. 사람들은 다양한 삶의 양태를 지녔고 그들만의 방식으로 진리와, 행복과 잘 사는 인생으로 나아간다. 그리고 그것이 타인에게 위해를 가하지 않는 한, 틀렸다고 할 수는 없는 것이다.

이에 더해, 한 쪽이 고장 났어도 다른 쪽을 활용하면 전류는 흐른다. 고로 or로 들어가는 전선이 많을수록 고장이라는 우발 상황에 대처할 수 있는 방안은 늘어난다. 역사적으로 폐쇄적이고 독단적인 집단은 결국 쇠락하고, 다양성을 추구하는 집단이 흥하는 것은 우연만은 아닌가 싶다. 뜬금없지만 다양성에 근간해 번영을 누리고 있는 초강대국 미국이 새삼 떠오르는 순간이다.

not의 관계는 and와 or의 관계에서 놓치고 있는 부분을 잡아 준다. 디지털 논리 회로에서 궁극적으로 전류가 흐르기 위해서 모든 전선이 사용되는 것은 아니다. 어떤 전선은 부정(negate)되어야 하는 법이다. 단순히 보면 이상하다. 하지만 회로의 구성에는 분명히 그것이 작용하고 있다. 일상 언어로 표시하자면 포기해야 하는 부분이, 나머지에 맡겨야 하는 부분들이 존재한다는 것이다. 성공적인 인생을 사는 데에 있어, 진리를 추구하는 데에 있어, 행복을 추구하는 데 있어 모든 부분이 전부 필요할 것 같지만 실상은 그렇지 않다는 것이다. 대다수의 사람들은 많은 것들을 쫓다가 결국 한 가지도 건지지 못한다. 두 마리 토끼는 잡기 힘들다. 신의 계시에 의해 모든 동물 한 쌍씩을 태우라고 명을 받았던 노아조차 전부를 태우진 못했으리라! 첨언하자면, not은 전류의 흐름을 - 인생으로 따지면 삶의 흘러감을 - 다른 편에 맡김에 따라, 집중하던 대상 이외의 나머지 것들에 대한 관심을 환기시켜주는 소중한 계기가 된다.

마지막으로, not은 괴델의 불완전성의 정리 - 일상어로 번역했을 때 증명되지 않는 것들이 존재한다는 것 - 와 결부되어 집착에의 해소라는 불가(佛家)적 진리를 생각하게 한다. 증명되지 않는 것은 존재할 수 있으며, 한계를 기대하지 않았던 부분에서도 한계는 존재할 수 있다. 적절히 인정하고 놓아줌을 실천할 때에 전류는, 우리의 인생은 잘 흐를 수 있다.

진리를 확실히 알지 못하는 우리 인간이 만들었지만, 부지불식(不知不識)간에 인생에의 진리를 보여주고 있는 컴퓨터의 디지털 논리 회로. 함께 살고 있고, 함께 살아야만 하는 and로 점철된 우리의 삶. 여기에 다원화된 세상 속에서 가져야할 덕목인 다양성에 대한 포용을 상징하는 or. 마지막으로 괴델의 불완전성의 정리와 결부하여 생각하게 되는 잔여(remainder)의, 놓아줌의 not. 인간이 생각하는 원리에 따라 고의적으로 만들어낸 인공물에서 다시금 우리가 살아가는 법을 배운다. 우리의 지식으로 만든 것에서 다시 우리가 배운다. 무어라 표현하기 힘들지만, 새삼 신기하다.