

2. 컴퓨팅과 컴퓨터

2010 데이터로 표현하는 세상 요약본
고려대학교 김현철 교수
hkim64@gmail.com

문제 해결을 위한 머릿속의 해결 방법 생각을 자동화 될 수 있는 형태로 표현해 보자.

기계의 등장

근대사에 있어서 가장 큰 변화를 가져온 것은 아마도 산업혁명이며, 산업혁명의 시작을 이야기 할 때 흔히 우리가 예로 드는 것이 1800년대 말의 방직 기계일 것이다. 그 이전에는 일일이 수공업으로 직물을 만들었는데, 사실 그 직물을 만드는 과정은 매우 단순한 일련의 작업들의 반복이라는 것이다. 기계는 작은 단위의 작업들을 순서대로 진행할 수 있는, 그래서 반복적으로 자동적으로 그 작업들을 진행할 수 있게 하는 도구이다. 이 작은 기계의 성공은, 그 당시 사람들에게 세상의 육체적 노동의 과정을 다시 한번 분석적으로 보게 만들어 비슷한 방식으로 수많은 기계가 만들어 지게 되었다. 그리고 그 기계로 인한 결과는 표준화된 똑 같은 물건의 (하지만 획일화된) 대량생산이며, 그 과정에서 효율성(비용, 시간, 노력)이 강조되는 가치가 자리잡게 되었다. 그러한 효율성과 획일화된 대량생산의 가치는 그 이전 시절의 배고프고 무엇이든 부족했던 인간의 고통을 해소시키고 토지와 농업 중심의 사회에서 산업 사회로 변화하는 과정에서 여러 가지 사회적 변화 ((부와 권력의 분배, 시민 사회 등장 등)도 가져 왔지만 한편으로는 에너지 고갈과 공해 문제, 인간성 상실 문제 그리고 부의 편중과 과도한 경쟁 체제 등의 부작용도 생기게 되었다. 기계의 성공은 결국 그러한 식으로 사람들의 사고방식을 바꿔 놓게 되었다. 예를 들어 1800년대 말에 등장했던 교실에서의 칠판과 학교에서의 사지선다형 시험 방식은 전형적인 산업시대 사고방식의 결과라고 생각된다. 이것은 기계와는 관계가 없을 지 모르지만 짧은 시간에 대량으로 획일적으로 일을 처리하려는 방식으로 등장한 것이므로 산업시대의 사고방식이 그대로 반영된 것이라고 할 수 있다.

기계는 본질적으로 “자동화(automation)”이라는 특성을 가지고 있으며, 기계를 가지고 “문제해결”을 하기 위하여서는 문제 자체를 추상화(abstraction)를 해야 하며 추상화를 통하여 문제를 분해(decomposition)하고 그것을 효율적인 절차(procedure)로 재구성하는 것을 필요로 한다. 예를 들어, 쌀을 가지고 밥을 하는 과정 (즉 문제해결)을 생각해 보자. 이것을 대신 해주는 “기계”를 만들어 보자. 어떤 추상화와 어떤 분해, 그리고 어떤 절차의 재조합이 필요한가. 또한 100여 년 지속된 이러한 기계기반의 산업시대를 살아 오면서 우리 인간의 사고 방식은 이러한 “분해”와 “절차”, “재조합”을 통한 문제해결에 익숙해져 왔다.

컴퓨팅 기계의 등장

산업시대가 어느 정도 성숙되어 오면서, 우리 인간은 또 다른 목적의 기계를 생각하게 되었다. 즉, 인간의 육체적 한계가 아닌, 정신적 한계, 즉 인간의 정신적 작업을 대신 해주는 “기계”의 필요성을 느끼게 되었다. 그리고 그 해결방법은 인간의 정신적 작업을 추상화하고 분해하여 재구성함으로써 “자동화”하고자 하였다. 그리고 그렇게 만들어 진 것이 바로 computing machine, 즉 computer 인 것이다. 왜 thinking machine이라고 하지 않았는가? 인간의 thinking의 본질은 무척 광범위하고 아직도 우리에게 알려진 것이 그리 많지 않다. 그 중에서 어떠한 문제 해결 과정을 기본적인 “추상”과 “반복”과 “절차”를 가지고 “자동화”할 수 있는 형태로 만드는 것을 “computing”이라고 한다.

즉, 우리 인간의 정신적 행위 중에서 단순 반복하는 작업이 있을 것이고, 그러한 단순 반복 작업을 자동화 할 수 있는 형태로 만들면, 그것이 ‘컴퓨팅’ 모델이고, 그러한 컴퓨팅 모델을 실제로 자동으로 반복 수행 시켜주는 기계가 바로 ‘컴퓨터’라고 할 수가 있다. 그래서 초기의 컴퓨팅과 컴퓨터는 숫자의 반복계산 작업에 많이 이용되었다. 예를 들어, 우리가 128x9763이란 것을 계산 한다고 하자. 우리는 이 계산을 직관적으로 하는 것은 아니다. 곱하기에 대한 우리가 배운 규칙을 순서대로 적용해서 (기계처럼) 답을 구해내는 것이다. 따라서 그 진행 순서를 우리가 분석해서 일반화 시켜서 적어 놓고 (이것이 곱하기에 대한 컴퓨팅 모델) 그 순서대로 진행하는 기계를 만들면 그것은 숫자 곱하기 기계가 되는 것이다. 혹은 우리는 128을 9763번을 반복해서 더하기만 해도 답을 구할 수 있는데, 그러면 그 곱하기 기계는 더하기 기계만으로도 만들 수가 있는 것이다. 만약 이런 기계가 없다면, 어떤 직업의 사람은 하루 종일 더하기만 혹은 곱하기만 10시간씩 단순 반복 노동을 하고 있을지도 모른다. 1800년대 기계의 도입이 가져온 사고방식의 변화가 (‘단순반복 작업은 기계에게’) 우리의 지금을 만들어 오고 있는 것이라고 생각해 볼 수 있다.

여기서 한가지 강조해야 할 부분 하나는, 어떤 문제 해결에 대한 아이디어나 생각을 컴퓨팅 모델로 변환 시켜주는 과정은 절대로 자동으로 할 수 있는 일이 아니며 오직 인간만이 스스로의 사고로 해야 한다는 것이다. 일단 그렇게 해서 컴퓨팅 모델이 만들어지면 그것은 컴퓨터라는 기계에서 자동으로 처리해주게 되는 것이다.

또 하나의 이슈는, 우리 인간의 정신적 활동 중에서 어떤 것이 컴퓨팅 모델로 만들어 질 수 있으며, 또 어떤 것이 만들어 질 수 없는 것인가에 대한 근본적인 질문이다. 인간의 모든 정신적 활동을 복잡하긴 하더라도 결국 컴퓨팅 모델로 만들 수 있을 것인가?

우리 머리 속의 생각이 몇 단계를 거쳐 단순한 형태로 바뀌어 해결되는 과정은 생각해 보면 매우 놀라운 일이다. 이것은 추상화의 힘이라고 할 수도 있다. 조금 과장해서 이것을 다시 이야기 해 보도록 하자. 사실 과거의 컴퓨터는 이진수에 대하여 사칙연산만 하는 회로만 들어 있는 것이었다. 어떤 것은 이진수의 덧셈만 하는 회로만 있기도 했다. 그렇다면 우리가 수학의 파이 값을 소수점 100자리까지 구하는 작업이나 필기체 한글을 인식하는 작업을, 누군가 분석하여 작은 작업 단위들도 분해하고 수행 절차를 만들어 알고리즘으로 ‘변환(transform)’ 시켜 놓으면, 그것을 다시 누군가가 컴퓨터 프로그래밍 언어로 코드화 시켜 놓고, 그러면 그것은 다시 컴파일러에 의하여 기계어로 변환되어, 결국은 원래의 그 복잡해 보이는 문제에 대한 해결이 덧셈 회로만 있는 컴퓨터에서 이진수의 덧셈의 반복의 형태로 수행되는 것이다. (놀랍지 않은가?) 만약에, 이 세상의 모든 문제들이 덧셈의 반복의 형태로 변환될 수 있다면, 우리는 덧셈 계산기가 세상의 모든 문제를 해결할 수 있다고 말할 수 있는 것이다. 실제로 세상의 웬만한 문제들은 (그래픽이든, 게임이든) 몇 가지의 간단한 기능들을 반복하는 절차적 과정으로 변환할 수 있고 (알고리즘과 프로그램의 형태로 표현) 그것은 다시 우리가 가지고 있는 컴퓨터에서 “빠르게 계산” 할 수가 있다.

무엇이 필요한가?

문제해결과정에서 발생하는 thinking에 대하여 다시 생각해 보면서 무엇이 필요한가를 살펴보자.

먼저 문제상황에 대한 정보나 작업들을 최소단위로 분해 하는 것이 필요하다. 큰 문제를 잘게 쪼개는 것, 실마리 정보들을 정리하는 것이 필요하다. 그 다음에는 그것을 이용하여 어떤 절차적 구조로 재 구성해보는 과정을 통하여 문제 해결 방법을 명시화 할 수 있게 되는 것이다. 그 분해되어 절차적 구조로 재 구성된 문제해결 방법을 일반적으로 그 문제에 대한 컴퓨팅 모델 혹은 알고리즘이라고 하고, 그것을 프로그램 언어로 코딩하면 컴퓨터에서 돌아가는 컴퓨터 소프트웨어가 되는 것이다.

컴퓨터소프트웨어는 결국 우리 인간의 정신적 노동, 문제해결 과정을 대신 해주는 역할을 하는 것이며, 우리 인간처럼 똑 같은 정보 처리 절차를 가진다. 즉, input-process-output (IPO)의 절차를 가진다. 우리 인간은 눈, 귀, 코, 등을 통하여 정보를 입력(input)받고 그것을 기반으로 내가 가지고 있는 어떤 문제해결 방식으로 그것을 처리(process)하고, 그 결과를 출력(output)시킨다. 예를 들어, 우리가 필기체 문자를 인식한다고 생각해보자. 무엇이 input 으로 들어와야 하는가. 그 들어온 input 단위들만 가지고 문제를 해결하는 구체적 절차 혹은 규칙을 써보자. 새로운 문자가 input 으로 들어 왔을 때 그 절차대로만 하면 정확한 output 이 나오는가. 만약 당신이 성공했다고 하면, 당신은 필기체 문자 자동 인식 기계를 만든 것이다.

컴퓨팅 모델에서 그 입력(input)되는 정보를 분해하여 정보 단위로 만들고 디지털화 하는 과정을 통틀어서 정보의 ‘코딩’이라고 말한다. 사실, 우리가 사용하는 언어, 문자, 숫자 등도 모두 어떠한 정보를 상호 구별할 수 있는 단위로 코딩된 것이라고 할 수 있다. 이 책에서는 정보의 코딩 뿐만 아니라, 다양한 컴퓨팅모델을 만들 수 있는 방법에 대해서 다룰 것이다.

컴퓨팅 도구(Computing Tools)의 간단한 역사

인간의 “육체적” 반복적 노동의 한계를 극복하고자 하는 “물리적” 기계에 세상이 몰두 해 있던 20세기 이전 시대에, 인간의 “정신적” 반복적 노동의 한계를 같은 방법으로 극복하고자 시도했다는 것은 아무리 생각해봐도 대단한 것이었다고 생각된다. 기계라기 보다는 도구에 가깝기는 했지만 이러한 computing tools이 처음 만들어 진 것은 1600년대이며, 우리가 이야기 하는 기계의 모습은 1700년대에 등장할 하게 된다. 그리고 전자식 컴퓨터는 1900년대 초 중반에 등장할 하는 것이다. 즉 우리가 지금 사용하는 컴퓨터는 갑자기 등장한 것은 아니며, 오랜 인간의 역사에서 등장한 계산 도구서부터 시작하여 1700-1800년대의 산업화 시대를 거치면서 기계의 본질적 의미를 띠고 그러한 모습을 가지게 되었고 20세기 초반의 전기전자물리학의 발전에 힘입어 다시 전자 기계의 형태를 가지게 되었으며 최근에는 인터넷과 연결되면서 지금도 계속 진화해 나가고 있다. 이것이 바로 1940년대 알란 튜링이 이야기 했던 생각기계(Thinking Machine)의 현 모습이다.

수판 혹은 주판 (abacus)에 대한 기록은 기원전부터 전해 내려 오는데 아마도 가장 오래된 숫자 계산 도구일 것이다. 이것은 숫자를 코드화 하고 계산 방식을 일정한 규칙을 두어 쉽게 처리하고 표시할 수 있게 한 것이지만, 반복해서 자동으로 할 수 있

는 기계의 수준은 아니었다.

기록에 나온 최초의 기계식 계산기는 프랑스의 파스칼(1623-1662)이 1642년에 처음으로 만든 계산기일 것이다. 세금계산일을 하는 아버지를 도울 생각으로 10대 소년일 때 처음 만들었던 것인데 톱니바퀴의 동작을 이용해서 덧셈과 뺄셈을 연산 하는 것이었으며 곱셈과 나눗셈은 덧셈과 뺄셈의 반복을 통해서 수행되도록 제작되었다. 이러한 방식은 몇 세기 후 1971년에 개발된 '마이크로프로세서'로 그대로 이어진 것이며 그 마이크로프로세서는 오늘날 모든 컴퓨터의 중심부품이 되었다. 또한 파스칼의 계산기는 그 이후에 만들어진 기계식 계산기에 직접적인 영향을 주었다. 라이프니츠(1646-1716)는 1671년에 파스칼 계산기에 자동 곱셈과 나눗셈 기능을 추가한 것을 개발하였으며, 1820년에 파스칼과 라이프니츠의 계산기에 영감을 받은 Colmar는 최초의 상용 계산기인 arithmometer를 만들었고, 1885년에 Felt는 파스칼 계산기의 입력 톱니바퀴를 손가락으로 누를 수 있는 키로 바꾸어 comptometer를 발명했는데 이것이 아마도 손가락으로 누르는 키로 입력을 받는 최초의 계산기일 것이다. Felt는 예산 부족으로 마카로니 포장 박스 꼬치구이용 꼬치, 스테이플과 고무줄로 만들었다고 한다. 그 이후 모두 전자식으로 바뀐 최초의 탁상용 계산기는 1961년에 등장한다.

컴퓨터의 역사는 계산기의 역사와는 맥락을 같이 하긴 하지만 조금 다른 관점을 가지고 있다. 원래 컴퓨터(computer)라는 용어는 위의 그러한 계산기(calculator)를 다루고 사용하는 사람이라는 의미였다고 한다. 계산기와 컴퓨터를 구분 짓는 가장 큰 특징 중의 하나는 바로 사용자 프로그램의 사용과 저장 여부라고 할 수 있다. 프로그램은 기계의 동작을 컨트롤 할 수 있기 때문이다. 즉 넣어주는 프로그램에 따라서 기계가 다르게 움직이게 한다는 것이다. 이것의 예로서는 1801년에 프랑스의 Jacquard에 의하여 만들어진 직조기(천을 짜주는 기계)인데 천공카드로 천에 새겨지는 무늬를 다르게 만드는 것이었다. 직조기를 전혀 변경하지 않고 카드의 천공 순서만 바꾸어도 다른 무늬의 천이 만들어지게 한 것인데, 이것이 '프로그램'이 가능하게 하는 최초의 업적이라고 할 수 있다. 이러한 천공 방식은 정보를 기록하여 그것을 재생하는 데에도 사용되는데 예를 들면 음악을 판에다가 천공하여 그것을 돌리면서 그 음악이 나오게 하는 음악 기계 등이 그 이후에 등장하였다.

영국의 캠브리지 대학의 수학교수였던 찰스 배비지는 1820년대에 좀 더 복잡한 기계식 계산기, 다항함수를 계산하는 Differential Engine을 만들었는데 2500개의 부품과 무게가 15톤이나 되는 엄청난 크기였다. 이렇듯 조금만 계산이 복잡해져도 그 크기가 커지고, 조금만 다른 계산을 하려고 해도 또 다른 기계를 만들어야 하는 문제가 있었다. 배비지는 천공카드를 이용한 프로그램 기계에 착안하여, 같은 기계라도 프로그램에 따라 여러 작업을 할 수 있는 범용 계산 기계를 생각해 내게 된다. 1837년 그는 최초의 범용 프로그램 컴퓨터라고 할 수 있는 Analytical Engine에 대한 아이디어를 발표한다. 입력은 천공카드로 받고, 전력은 증기엔진으로, 그리고 기어의 위치를 사용하여 숫자를 표현했다. 기계 부품의 정확성과 예산 문제 때문에 그 당시에 완성되지는 못했지만 그 과정에서 '프로그래밍'이라는 개념이 영국에서 잡혀 나가기 시작한다.

1880년대 말에 미국인 Hollerith는 기계에서 읽을 용도의 데이터 저장장치를 개발하고 있었는데, 기차표 검표인이 승객의 기차표에다가 그 승객의 특징에 따라 다른 위치에 펀치(천공)하는 것에 착안하여 시작하였는데, 키 천공기(키보드 같은 것으로 카드에 천공하는 도구)와 천공카드 정보 처리기(Tabulator)를 만들어 1890년에 미국 인구센서스에 도입하여 사용하기 시작했다. 천공 카드에서의 천공의 유무와 위치를 가지고 특정 데이터를 표현하고, 그 tabulator는 천공카드의 그 데이터를 전기 신호로 읽어 들어서 테이블의 형태로 요약하고 계산 같은 것을 자동으로 해주는 전기장치이다. 이것은 컴퓨터 역사에서 또 다른 큰 발자취인데, 엄청난 양의 데이터를 자동으로 처리하게 되는 기술의 등장을 의미 한다. Hollerith는 이러한 천공기술을 이용한 대규모 데이터 처리 기술을 가지고 1911년 CTR이라는 회사를 설립하고 이것은 1924년에 왓슨에 의해 IBM으로 이름이 바뀌어 공공용 및 상업용 대규모 데이터 처리 사업과 컴퓨터와 같은 사무용 자동화 기기 사업의 새로운 장을 열게 된다.

이렇듯 계산 기계, 혹은 데이터 처리 기계의 입력부분은 1800년대 말에 등장한 키보드와 천공카드로 큰 변화를 가져왔다. 반면 계산 기계의 실제 작동은 1900년대대 초로 들어오면서 여전히 톱니 바퀴를 이용한 기계식이거나 혹은 전기의 전압과 전류의 양을 이용한 아날로그 방식이었다. 계산 기계는 1900년대 전반기 세계2차대전 동안에 급속한 발전을 하게 된다. 그 시기는 계전기, 진공관, 회로 등과 같은 많은 전기 전자적 발견이 있었으며 컴퓨팅에 대한 많은 새로운 이론(파인만, 튜링, 새논, 폰노이만 등)이 등장한 시기였다. 그러한 노력의 결과로 1940년대에 들어서 현대적 컴퓨터의 모습을 갖춘 전자계산기가 등장하는데, 어느 것이 최초의 전자식 디지털 범용 컴퓨터인가에 대해서는 아직도 많은 논란이 있다. 그 논란은 여러 기준에 대한 관점 때문에 생기는데, 그것이 완전 전자식 혹은 전기기계식인지, 이진수 혹은 십진수를 쓰는지, 프로그램의 여부는 어떻게 되는지, 튜링머신과 같은 방식인지(이것은 Turing-complete이라고 함), 개발연도인지 첫 운영연도인지 등에 대한 것이다.

독인인 Zuse가 1941년에 처음 운영시킨 Z3는 최초의 프로그램 기반의 범용 디지털 컴퓨터라고 불리운다. 이진수를 사용하였고 프로그램은 35mm 영화 필름을 천공하여 사용했다. Z3가 전자기계식인 것에 비하여 1942년에 처음 운영된 아이오와주립대학의 ABC는 최초의 완전 전자식 디지털 컴퓨터라고 불리운다. 하지만 프로그래밍이 가능한 범용 컴퓨터가 아니었고 당연히 튜링머신 방식이 아니었다. 영국의 암호해독팀에서 독일군 암호해독용으로 만들어서 1944년에 처음 사용한 Colossus Mark 1은 케이블과 전기스위치로 프로그램을 하도록 되어 있었으며 튜링머신 방식은 아니었다. 같은 해 1944년에 운영된 하버드 대학과 IBM

의 Harvard Mark-1은 10진수를 쓰는 전자기계식이었으며 완전한 튜링머신 방식은 아니었다. 일반적으로 최초의 완전 전자식 범용 컴퓨터라고 불리는 ENIAC은 1945년에 운영되는데 10진수 체계를 사용하였으며 케이블과 스위치로 프로그램 하는 것이었다. 그 무게는 30톤이나 되었으며 18,000개 이상의 진공관튜브를 사용하였다. 애니악은 사람이 7시간 걸리던 탄도계산을 단 3초만에 계산함으로써 컴퓨터의 잠재성을 보여주었다.

여기까지의 컴퓨터에서는 프로그램이 천공카드나 테이프를 사용하여 읽혀 들이거나 ENIAC처럼 스위치로 케이블 연결을 변경함으로써 프로그램을 하는 프로그램 외장 방식(프로그램이 컴퓨터 밖에 있음)이었는데, 현대 컴퓨터의 가장 큰 특징 중의 하나인 '프로그램 내장방식 (stored program)'이 처음 사용된 것은 1948년에 영국 맨체스터의 빅토리아 대학에서 처음 사용된 실험용 맨체스터 베이비이다. 저장식 프로그램은 프로그램 명령어들이 전자식메모리에 들어가 있어서 컴퓨터 안에 저장되어 있다라는 의미이다. 실험용이 아닌 정식 컴퓨터로는 영국 캠브리지 대학에서 만든 EDSAC이 처음으로 프로그램 내장 방식을 사용했다. 이후 1950년에 미국 펜실베이니아와 프린스턴 대학에서 폰노이만 방식의 프로그램 내장 방식을 사용한 EDVAC을 개발하였다. 최초의 상업용 컴퓨터는 1951년에 영국 맨체스터 대학에 납품된 Ferranti Mark 1 과 미국의 인구통계국에 납품된 UNIVAC-1을 이야기 하는데 UNIVAC은 최초의 대량 생산된 컴퓨터이며 입력, 연산, 출력을 동시에 할 수 있고 저장은 자기테이프 시스템을 사용하였다. 1953년에는 IBM에서 자기드럼장치로 기억장치로 사용하는 IBM650을 개발한다. 1958년에는 미국에서 텍사스인스트루먼트가 반도체 회로를 처음 개발했으며, IBM에서는 진공관이 아닌 트랜지스터를 사용한 IBM7090/7070을 발표한다.

그 이후에는 프로그램 언어와 운영체제 개발, 반도체와 메모리 분야의 혁신등이 빠른 속도로 이루어져 왔고, 1980년대와 1990년대에는 애플과 마이크로소프트, 그리고 IBM이 개인용 컴퓨터를 대량 생산하고, 인터넷이 대중화 되면서 정보화 혁명이 시작되었고, 2000년대에 들어와서는 웹2.0의 문화가 확대되면서 관련된 기술과 무선인터넷, 그리고 유비쿼터스와 스마트 기기와 환경이 인류의 문화와 역사, 사회를 변화 시켜나가고 있다.

이상과 같이 계산 기계의 역사를 보면 우리가 지금 사용하고 있는 컴퓨터는 1800년대부터 축적된 기술과 사고방식의 발전에 의한 결과물이며 상호간에 다 연결되어 영향을 받으면서 진행되어 온 것을 알 수가 있다. 분명한 것은, 사람의 생각과 사고 자체를 컴퓨팅 모델로 표현하여 컴퓨팅 기계에서 자동으로 수행하고자 한 그 노력은 분명히 우리 인류 역사에서 큰 혁명적인 일이라고 할 수 있다. 그리고 그 노력은 앞으로 진행될 인류 역사에 비추어 보면 이제 막 시작을 한 단계에 불과하며 앞으로 어떤 발전이 더 진행될지 궁금하다.

.끝.