SNU 046.016 컴퓨터과학이 여는 세계(Computational Civilization) Part I

Prof. Kwangkeun Yi

Department of Computer Science & Engineering

차례

다음

컴퓨터라는 도구

인류역사에 유례가 없던 도구

컴퓨터 v.s. 칼, 활, 바퀴, 고무밴드, 자동차, 냉장고, ···

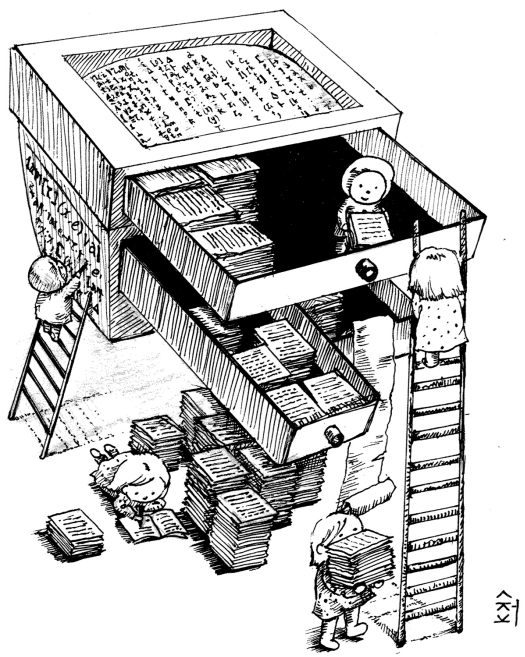
- 컴퓨터라는 도구의 놀라운 특이점은 뭘까?

컴퓨터라는 도구

인류역사에 유례가 없던 도구

컴퓨터 v.s. 칼, 활, 바퀴, 고무밴드, 자동차, 냉장고, ···

- 컴퓨터라는 도구의 놀라운 특이점은 뭘까?
- "만능"



컴퓨터 디자인의 "탄생비화"

"보편만능의 도구(universal machine)"

- 20세기 수학의 좌절을 재확인하는 데 동원된 소품
- 이것이 20세기 정보혁명의 주인공이 되는 아이러니

수학자들의 꿈

수리명제 자동판결 문제(Entscheidungsproblem, decision problem)

1928년 @ 국제 수학자대회(ICM)

Gottried Leibniz(-1716)

Gottlob Frege(-1925)

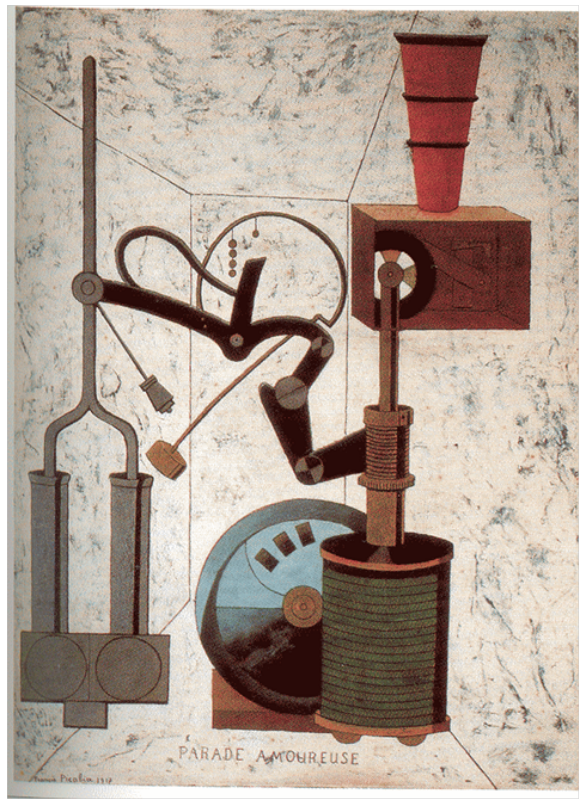
David Hilbert(-1943)

- 모든 명제들의 참/거짓을 "기계적으로" 판명할 수 없을까?
- 참인 모든 명제들을 "기계적으로" 만들어낼 수 없을까?

(사진출처: Google)

"기계적"

- 맛있는 라면만들기: 라면공장 기계
- 두 정수의 최대공약수 찾기: 막대재기 기계
- 참/거짓 판별하기: 추론 기계



(사진출처: Google)

추론 기계: 추론 규칙

- 쌍 uncle(a, b)들의 추론 규칙

$$\frac{\mathsf{male}(u) \quad \mathsf{father}(f, i) \quad \mathsf{brother}(f, u)}{\mathsf{uncle}(u, i)}$$

$$\frac{\mathsf{father}(c, a) \quad \mathsf{father}(c, b)}{\mathsf{brother}(a, b)}$$

$\ldots$

추론 기계: 추론 규칙

- 쌍 $\mathsf{uncle}(a, b)$들의 추론 규칙

$$\frac{\mathsf{male}(u) \quad \mathsf{father}(f, i) \quad \mathsf{brother}(f, u)}{\mathsf{uncle}(u, i)}$$

$$\frac{\mathsf{father}(c, a) \quad \mathsf{father}(c, b)}{\mathsf{brother}(a, b)}$$

$$\cdots$$

- 쌍 $(\{g_1, \cdots, g_n\}, f)$들의 추론 규칙

$$\frac{}{(\Gamma, T)} \quad \frac{}{(\Gamma, f)}\ f \in \Gamma \qquad \frac{(\Gamma, F)}{(\Gamma, f)} \qquad \frac{(\Gamma, \neg\neg f)}{(\Gamma, f)}$$

$$\frac{(\Gamma, f_1) \quad (\Gamma, f_2)}{(\Gamma, f_1 \wedge f_2)} \qquad \frac{(\Gamma, f_1 \wedge f_2)}{(\Gamma, f_1)}$$

$$\frac{(\Gamma, f_1)}{(\Gamma, f_1 \vee f_2)} \qquad \frac{\begin{array}{c}(\Gamma, f_1 \vee f_2) \\ (\Gamma \cup \{f_1\}, f_3) \quad (\Gamma \cup \{f_2\}, f_3)\end{array}}{(\Gamma, f_3)}$$

$$\frac{(\Gamma \cup \{f_1\}, f_2)}{(\Gamma, f_1 \Rightarrow f_2)} \qquad \frac{(\Gamma, f_1 \Rightarrow f_2) \quad (\Gamma, f_1)}{(\Gamma, f_2)}$$

$$\frac{(\Gamma \cup \{f\}, F)}{(\Gamma, \neg f)} \qquad \frac{(\Gamma, f) \quad (\Gamma, \neg f)}{(\Gamma, F)}$$

1931년, 수학계의 "좌절" 혹은 "희소식"

"기계적인 방식만으론 사실인지 판정할 수 없는,
그런 명제가 존재한다."

Kurt Gödel(-1978)

(사진출처: Google)

1936년, 8월 9일 vs 5월 28일

손기정

알란 튜링(Alan Turing)

"계산가능한 수에 대해서, 수리명제 자동판별 문제에 응용하면서"(On Computable Numbers, with an Application to the Entscheidungsproblem)

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*. it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions. the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel†. These results

† Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte Math. Phys.*, 38 (1931), 173–198.

have valuable applications. In particular, it is shown (§ 11) that the Hilbertian Entscheidungsproblem can have no solution.

In a recent paper Alonzo Church† has introduced an idea of "effective calculability", which is equivalent to my "computability", but is very differently defined. Church also reaches similar conclusions about the Entscheidungsproblem‡. The proof of equivalence between "computability" and "effective calculability" is outlined in an appendix to the present paper.

1. *Computing machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach § 9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ which will be called "m-configurations". The machine is supplied with a "tape" (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol". At any moment there is just one square, say the r-th, bearing the symbol $\mathfrak{S}(r)$ which is "in the machine". We may call this square the "scanned square". The symbol on the scanned square may be called the "scanned symbol". The "scanned symbol" is the only one of which the machine is, so to speak, "directly aware". However, by altering its m-configuration the machine can effectively remember some of the symbols which it has "seen" (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration q_n and the scanned symbol $\mathfrak{S}(r)$. This pair $q_n, \mathfrak{S}(r)$ will be called the "configuration": thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (*i.e.* bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down

† Alonzo Church, "An unsolvable problem of elementary number theory", *American J. of Math.*, 58 (1936), 345–363.

‡ Alonzo Church, "A note on the Entscheidungsproblem", *J. of Symbolic Logic*, 1 (1936), 40–41.

튜링의 그 때 그 시절(1/2)

- 1934년 11월, Cambridge U 학부졸업논문 제출
- 1935년 봄, Part III course on Foundations of Mathematics수강 (강사: Max Newman)
- Newman의 강의는 Gödel의 불완전성의 증명(Incompleteness Theorem)으로 마무리됨.
  - "기계적인 방식으로는 참/거짓을 판명할 수 없는 명제가 존재한다."
  - Newman: "참/거짓을 판명해주는 기계적인 방식은 있을 수 없겠지..."

튜링의 그 때 그 시절(2/2)

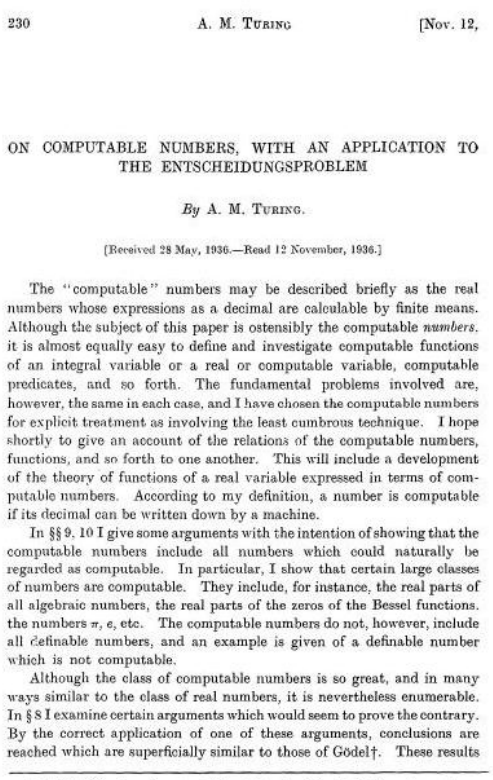
- Newman강의를 수강후인 1935 초여름,
  - 1935년 4월 말, group theory에 대한 논문 제출 및 출판(London Mathematical Society). 이 논문은 폰 노이만 논문을 작게 개선한 것.
  - 1935년 동안은 또 양자역학에도 연구를 할까 생각함. 수리물리학 Fowler교수를 찾아가 연구꺼리를 얻었으나 진전이 없었슴.
  - Newman이 강의때 던진 말이 튜링의 관심을 붙듬.
- 이 때쯤, 장거리달리기 취미를 가지기 시작. 튜링 왈, "달리기를 마치고 풀밭에 누워 있는데 힐버트의 세번째 문제를 어떻게 풀지가 생각나더라."

떠나자: 오리지날 논문 속으로

"On Computable Numbers, with an Application to the Entscheidungsproblem"

Proceedings of the London Mathematical Society, ser.2, vol.42 (1936-37). pp.230-265;

corrections, Ibid, vol 43(1937) pp.544-546



230 A. M. TURING [NOV. 12,

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

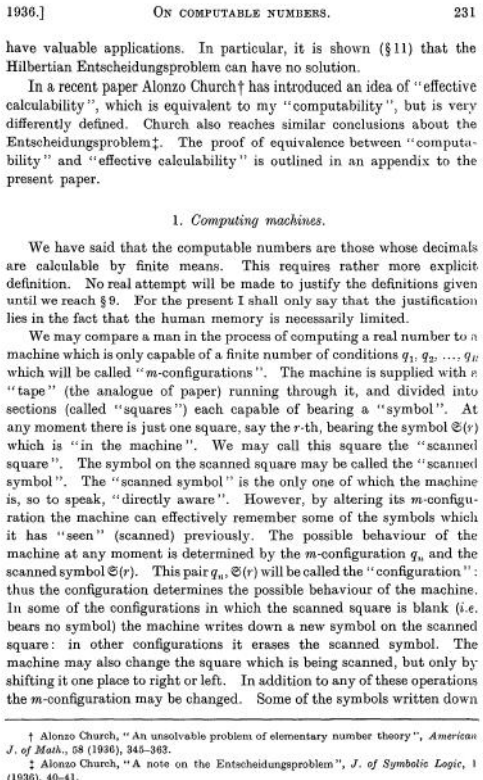
By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*. it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions. the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel†. These results



1936.] ON COMPUTABLE NUMBERS. 231

have valuable applications. In particular, it is shown (§ 11) that the Hilbertian Entscheidungsproblem can have no solution.

In a recent paper Alonzo Church† has introduced an idea of "effective calculability", which is equivalent to my "computability", but is very differently defined. Church also reaches similar conclusions about the Entscheidungsproblem‡. The proof of equivalence between "computability" and "effective calculability" is outlined in an appendix to the present paper.

1. *Computing machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach § 9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ which will be called "m-configurations". The machine is supplied with a "tape" (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol". At any moment there is just one square, say the r-th, bearing the symbol $\mathfrak{S}(r)$ which is "in the machine". We may call this square the "scanned square". The symbol on the scanned square may be called the "scanned symbol". The "scanned symbol" is the only one of which the machine is, so to speak, "directly aware". However, by altering its m-configuration the machine can effectively remember some of the symbols which it has "seen" (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration q_n and the scanned symbol $\mathfrak{S}(r)$. This pair $q_n, \mathfrak{S}(r)$ will be called the "configuration": thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (i.e. bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down

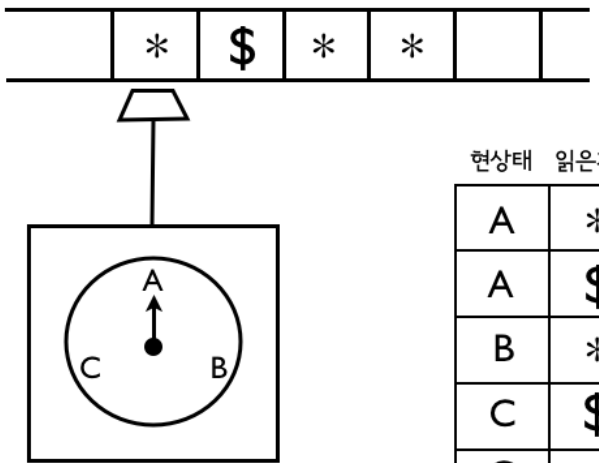
† Alonzo Church, "An unsolvable problem of elementary number theory", *American J. of Math.*, 58 (1936), 345–363.

‡ Alonzo Church, "A note on the Entscheidungsproblem", *J. of Symbolic Logic*, 1 (1936), 40–41.

“1. Computing Machines”, “2. Definitions”

기계적인 방식 = 아래의 부품들로 만드는 기계로 돌리는

- 무한: 빈칸이 무한히 많은 테잎
- 유한: 기계상태들 $\mathbb{S}$, 테잎 심볼 $\mathbb{T}$, 규칙표 $\mathbb{R}$
- 규칙표 $\mathbb{R} \subseteq \mathbb{S} \times \mathbb{T} \times \mathbb{T} \times \{>, <, ||\} \times \mathbb{S}$



현상태	읽은기호	쓸기호	다음칸	다음상태
A	*	*	>	A
A	$	$	>	B
B	*	*	<	C
C	$	*	>	C
C	*	$	>	B

"3. Examples of computing machines"

계산가능한 숫자열들

- 0 1 0 1 0 1 0 1 $\cdots$를 만드는 튜링기계
- 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 $\cdots$를 만드는 튜링기계

“4. Abbreviated tables”

“skeleton tables”: 반복해서 사용할 라이브러리/서브루틴 룰들

- rules for finding the left most symbol
- rules for writing a symbol at the end of the first symbol
- etc.

"5. Enumeration of computable sequence"

튜링기계마다 자연수 하나로 표현가능. 규칙표를 보자:

상태심볼 $\{A, B, C\}$

테이프심볼 $\{*, \$\}$

A	$*$	$*$	$>$	A
A	$\$$	$\$$	$>$	B

"5. Enumeration of computable sequence"

튜링기계마다 자연수 하나로 표현가능. 규칙표를 보자:

$$\begin{array}{rll} \text{상태심볼} & \{A, B, C\} & \text{은 } \{S_0, S_1, S_2\} \\ \text{테이프심볼} & \{*, \$\} & \text{은 } \{T_0, T_1\} \end{array}$$

A	$*$	$*$	$>$	A
A	$\$$	$\$$	$>$	B

"5. Enumeration of computable sequence"

튜링기계마다 자연수 하나로 표현가능. 규칙표를 보자:

$$\text{상태심볼} \quad \{A, B, C\} \quad \text{은} \ \{S_0, S_1, S_2\}$$
$$\text{테이프심볼} \quad \{*, \$\} \quad \text{은} \ \{T_0, T_1\}$$

A	$*$	$*$	$>$	A
A	$\$$	$\$$	$>$	B

따라서 일렬로 표현하면

$$A \quad * \quad * \quad > \quad A \quad \text{끝} \quad A \quad \$ \quad \$ \quad > \quad B$$
$$S\,0 \quad T\,0 \quad T\,0 \quad > \quad S\,0 \quad X \quad S\,0 \quad T\,1 \quad T\,1 \quad > \quad S\,1$$

즉, $S, T, 0, \cdots, 9, <, >, ||, X$로 표현한 □진수.

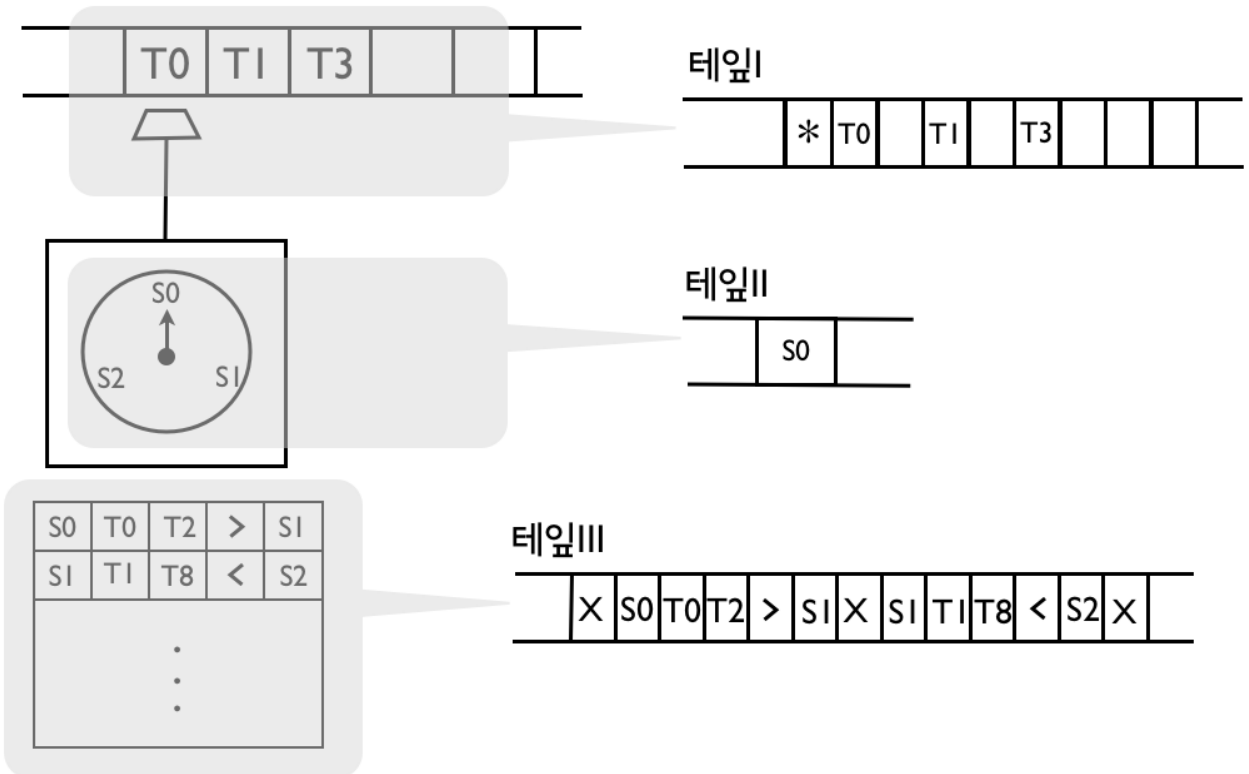
“6. The universal computing machine”, “7. Detailed description of the universal machine”

하나의 튜링 기계를 만들 수 있다,
임의의 튜링기계를 입력으로 받아 그 기계의 작동을 하는.
보편만능의 기계(“universal machine”)

- 임의의 튜링기계를 테이프에, 심볼의 일차원 실로 받기
- 그것대로 실행하는 튜링기계 규칙표 만들기

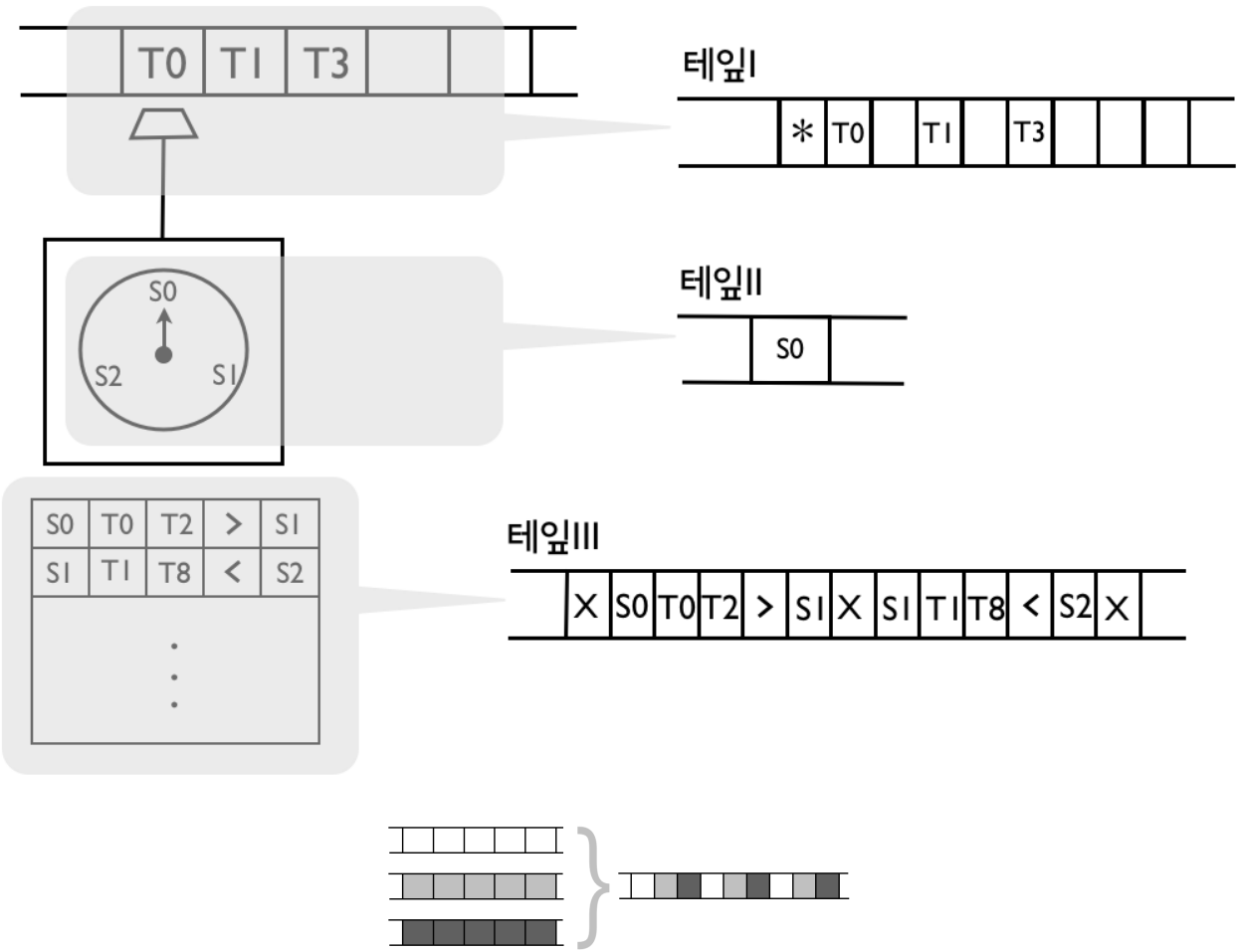
보편만능의 기계(1/2): 임의의 튜링기계를 테이프에 표현하기

17개 테이프 심볼이면 충분: S, T, <, >, |, 0, · · · , 9, X, ∗



보편만능의 기계(1/2): 임의의 튜링기계를 테이프에 표현하기

17개 테이프 심볼이면 충분: S, T, <, >, |, 0, · · · , 9, X, ∗



보편만능의 기계(2/2): 규칙표

아래과정을 반복하는 규칙표

1. 읽기: 테잎II에서 현재상태 심볼 $\mathtt{S}i$
2. 읽기: 테잎I에서 위치심볼$(*)$의 테잎심볼 $\mathtt{T}j$
3. 규칙찾기: $\mathtt{S}i$와 $\mathtt{T}j$와 매치되는 규칙을 테잎III에서 찾기.
4. 찾은 규칙 $\boxed{\mathtt{S}i \mid \mathtt{T}j \mid \mathtt{T}j' \mid m \mid \mathtt{S}i'}$ 이 시키는 일하기: 심볼복사 + 위치심볼이동

예술로 표현한 “Univeral Turing Machine”

Roman Verostko, 1998, www.verostko.com/manchester/manchester.html

(그림출처: Google)

증명 준비: 튜링기계의 급소

- 튜링기계의 개수는 무한히 많지만
- 자연수의 개수를 넘을 수 없다
  - 튜링기계마다 자연수 하나(17진수의 자연수)에 대응
- 무한의 세계에도 크기 차이가 있다

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

- 모순: 그런데 모든 튜링기계와 다른 튜링기계 D가 있다!

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

- 모순: 그런데 모든 튜링기계와 다른 튜링기계 D가 있다!

- 따라서 멈춤문제를 풀 수 있는 튜링기계 H는 없어야.

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

- 모순: 그런데 모든 튜링기계와 다른 튜링기계 D가 있다!

- 따라서 멈춤문제를 풀 수 있는 튜링기계 H는 없어야.

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

- 모순: 그런데 모든 튜링기계와 다른 튜링기계 D가 있다!

- 따라서 멈춤문제를 풀 수 있는 튜링기계 H는 없어야.

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"8. Application of the diagonal process"

멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다,
는 것을 보임.

- 만약 있다고 하자 H. 그러면,
- 모든 튜링기계를 줄 세워 아래 테이블 가능:

		입력			
		I_1	I_2	I_3	$\cdots$
튜	M_1	1	1	0	$\cdots$
링	M_2	1	0	1	$\cdots$
기	M_3	1	0	1	$\cdots$
계	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

- 모순: 그런데 모든 튜링기계와 다른 튜링기계 D가 있다!

$$D(n) = UM(M_n, I_n) \times H(M_n, I_n) + 1$$

- 따라서 멈춤문제를 풀 수 있는 튜링기계 H는 없어야.

칸토르(Georg Cantor)의 대각선 논법($|\mathbb{N}| < |2^{\mathbb{N}}|$)

"9. The extent of the computable numbers"

기계적으로 계산가능한 수들의 범위

튜링이 정의한 기계적인 계산과정(튜링머신으로 돌릴수 있는 과정)이 과연 "기계적인 계산"의 모두인가에 대한 방어. 세가지로 설득

- 직관적이지 아니한가
- 두 개의 정의가 결국 같지 아니한가
  - Hilbert's restricted functional calculus = Turing computable
- 이렇게 계산가능한 수의 범위가 꽤 크다, 보라

“10. Examples of large classes of numbers which are computable”

- 계산가능한 함수로 꾸민 계산가능한 함수는 계산가능
- 계산가능한 함수로 꾸민 재귀함수는 계산가능
- 계산가능한 수열의 극한값은 계산가능
- 등등

“11. Application to the Entscheidungsproblem”(1/3)

기계적인 과정으로 명제의 참거짓을 판단할 수 있을까?
혹은
기계적인 과정으로 모든 참인 명제를 만들어 낼 수 있을까?

- 불가능하다. 왜냐하면,
- 가능하다고 가정하고 살펴보면 모순이기때문이다.

"11. Application to the Entscheidungsproblem" (2/3)

모든 참인 명제를 만들어내는 튜링기계 A

증명목표: A는 존재하지 않는다.

사실: A가 존재하면 H가 존재한다.

사실: H는 존재하지 않는다.

따라서 A는 존재하지 않는다.

"11. Application to the Entscheidungsproblem" (3/3)

사실: A가 존재하면 H가 존재한다.

A를 부품으로 써서 H를 쉽게 만들 수 있기때문:

- 입력: 멈출지 알고싶은 튜링기계 M
- UM으로 A를 흉내낸다.
  - 참인 명제를 빠뜨림없이 만들것이므로
  - "(튜링기계 M은 멈춘다)" 또는 "¬(튜링기계 M은 멈춘다)"를 반드시 만들것이다.

UM이 만드는 참인 명제는, 명제에 대한 명제가 아닌 1차명제(first-order predicates)뿐으로 한정된다. 예를들어,

"(M이 멈춘다는 참거짓을 알 수 없다)"는 1차명제가 아니다.

여담: H가 존재한다면?

많은 오리무중이 자동 증명됨.

여담: H가 존재한다면?

많은 오리무중이 자동 증명됨.

예를들어,

- Fermat: 자연수 $n > 2$에 대해서 $a^n + b^n = c^n$인 자연수a, b, c는 없다.
- Goldbach: 모든 짝수는 두 솟수의 합이다.

여담: H가 존재한다면?

많은 오리무중이 자동 증명됨.

예를들어,

- Fermat: 자연수 $n > 2$에 대해서 $a^n + b^n = c^n$인 자연수a, b, c는 없다.
- Goldbach: 모든 짝수는 두 솟수의 합이다.

참임을 확인해가는 튜링 기계를 만들고 멈춤문제 풀이로 해결:

- (a, b, c, n)를 모두 "훑으면서", $a^n + b^n = c^n$이면 멈추는 기계
- 모든 짝수 k를 "훑으면서", 두 솟수 (p, q)를 "훑으면서", $k \neq p + q$이면 멈추는 기계

400년의 축적: 컴퓨터의 탄생

- 꿈: 인간 논리의 자동화(기계화)
- $\cdots$, Leibniz, Frege, Cantor, Hilbert, Gödel, Turing
- 튜링학생의 불가능 재증명 논술에서 나온 소품

보편만능의 기계(universal machine)

다른 트랙

자동계산장치들

점점 보편만능의 기계에 가까워가고···

다른 트랙

자동계산장치들

- 주판

점점 보편만능의 기계에 가까워가고···

다른 트랙

자동계산장치들

- 주판
- 1600년대 사칙연산: 파스칼(Pascal), 라이프니츠(Leibniz)

점점 보편만능의 기계에 가까워가고···

다른 트랙

자동계산장치들

- 주판
- 1600년대 사칙연산: 파스칼(Pascal), 라이프니츠(Leibniz)
- 1830년대 사칙연산이상: 배비지(Babbage)

점점 보편만능의 기계에 가까워가고···

다른 트랙

자동계산장치들

- 주판
- 1600년대 사칙연산: 파스칼(Pascal), 라이프니츠(Leibniz)
- 1830년대 사칙연산이상: 배비지(Babbage)
- 1920년대이후 회계장부(IBM), 통계계산(ABC)

점점 보편만능의 기계에 가까워가고···

다른 트랙

자동계산장치들

- 주판
- 1600년대 사칙연산: 파스칼(Pascal), 라이프니츠(Leibniz)
- 1830년대 사칙연산이상: 배비지(Babbage)
- 1920년대이후 회계장부(IBM), 통계계산(ABC)
- 계산내용을 외부입력으로: Mark(Harvard U), ENIAC(U Penn)

점점 보편만능의 기계에 가까워가고···

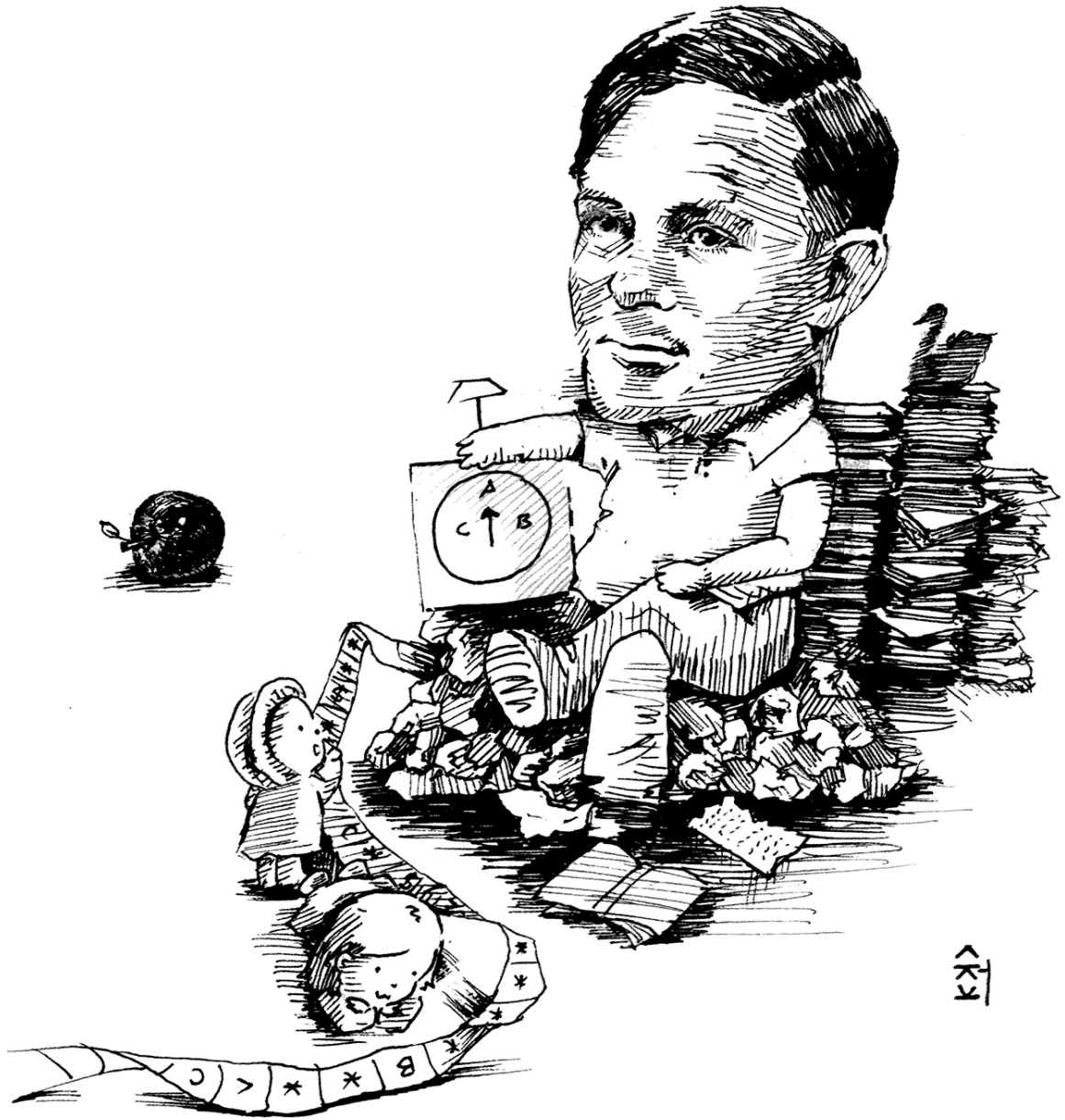
다른 트랙

자동계산장치들

- 주판
- 1600년대 사칙연산: 파스칼(Pascal), 라이프니츠(Leibniz)
- 1830년대 사칙연산이상: 배비지(Babbage)
- 1920년대이후 회계장부(IBM), 통계계산(ABC)
- 계산내용을 외부입력으로: Mark(Harvard U), ENIAC(U Penn)

점점 보편만능의 기계에 가까워가고···

- 튜링: 튜링이 던진 디자인이 올킬.
- 구현기술: 그동안 쌓인 구현 기술이 신속히 동원됨


A
C
B

다음

다음

다음