튜링의 1935년:
컴퓨터 원조논문이 나오기까지의 1년여 과정에 대한 추측

이 광근
서울대학교
컴퓨터공학부

12/07/2017 @ 서울대 교육상 기념

살펴보게된 동기

- 튜링의 그 1년이 궁금
- 뭔가 구겨진 느낌
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
- 뭔가 구겨진 느낌
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
- 뭔가 구겨진 느낌
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 "천재"라는 말로 튜링을 수식
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 "천재"라는 말로 튜링을 수식
  - 생각없는 오버라는 심증. 학생들의 자신감을 좀먹는
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 “천재”라는 말로 튜링을 수식
  - 생각없는 오버라는 심증. 학생들의 자신감을 좀먹는
  - “난 천재가 아니므로 그런 원천 아이디어를 만들지는 못할게 분명해.”
- 그 일년을 복기해 보기로

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 "천재"라는 말로 튜링을 수식
  - 생각없는 오버라는 심증. 학생들의 자신감을 좀먹는
  - "난 천재가 아니므로 그런 원천 아이디어를 만들지는 못할게 분명해."
- 그 일년을 복기해 보기로
  - 막스 뉴만교수(38세)의 강의 (상상)

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 "천재"라는 말로 튜링을 수식
  - 생각없는 오버라는 심증. 학생들의 자신감을 좀먹는
  - "난 천재가 아니므로 그런 원천 아이디어를 만들지는 못할게 분명해."
- 그 일년을 복기해 보기로
  - 막스 뉴만교수(38세)의 강의 (상상)
  - 튜링의 1936년 논문이 나오기 까지의 과정 (상상)

살펴보게된 동기

- 튜링의 그 1년이 궁금
  - 튜링이 괴델의 증명을 배운 1935년(23세)
  - 괴델의 증명을 자신만의 방식으로 재구성(1936년 논문)
  - 소품: 컴퓨터의 원천 설계도(Universal Machine)
- 뭔가 구겨진 느낌
  - 매스컴은 "천재"라는 말로 튜링을 수식
  - 생각없는 오버라는 심증. 학생들의 자신감을 좀먹는
  - "난 천재가 아니므로 그런 원천 아이디어를 만들지는 못할게 분명해."
- 그 일년을 복기해 보기로
  - 막스 뉴만교수(38세)의 강의 (상상)
  - 튜링의 1936년 논문이 나오기 까지의 과정 (상상)

그 기간을 복기하면 구체적으로 그 근거가 드러나지 않을까

Turing's 1935: my guess about his intellectual journey to "On Computable Numbers"

Kwangkeun Yi

Dept. of Computer Science & Engineering
Seoul National University

Turing's 1936 Paper

"*On Computable Numbers, with an Application to the Entscheitungsproblem*" *Proceedings of the London Mathematical Society, ser.2, vol.42 (1936-37). pp.230-265; corrections, Ibid, vol 43(1937) pp.544-546*

- shows a variant of Gödel's Incompleteness proof(1931)
- contains the blueprint of computer (Universal Machine)

My Motivation

Curious: how did Turing get the ideas underlying this foundational paper of modern computer?

- a computer = Universal Machine
- a computer = a single machine that can do any mechanical computation

This talk:

- the content of the 1936's paper and
- its intellectual "pedigree" (my guess)

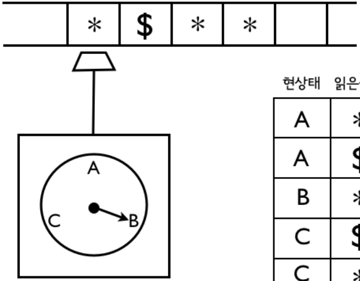
Turing's 1936 Paper

Theorem. By mechanical way we cannot generate all true formulas.

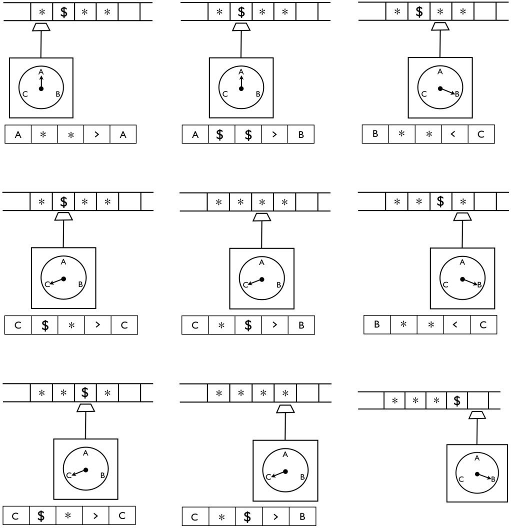
Turing's Definition

mechanical computation $\stackrel{\text{def}}{=}$ execution by a turing machien(TM)

An example TM:

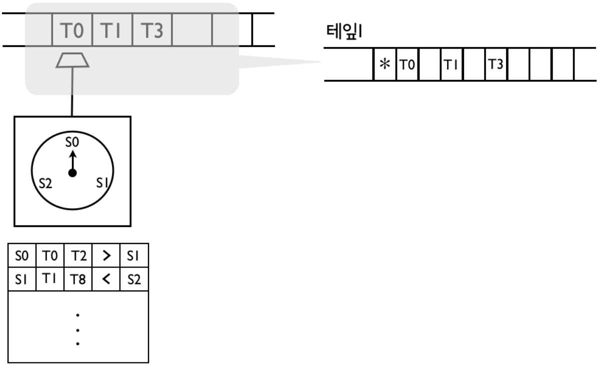


현상태	읽은심볼	쓸심볼	다음칸	다음상태
A	*	*	>	A
A	$	$	>	B
B	*	*	<	C
C	$	*	>	C
C	*	$	>	B


* $ * *
A * * > A
* $ * *
A $ $ > B
* $ * *
B * * < C
* $ * *
C $ * > C
* * * *
C * $ > B
* * $ *
B * * < C
* * $ *
C $ * > C
* * * *
C * $ > B
* * * $

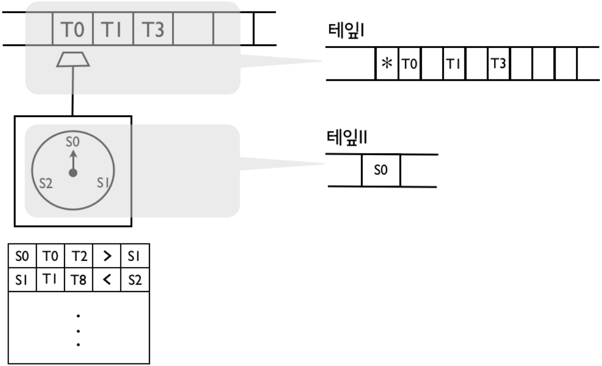
Universal TM: a Turing machine

Expressing a TM in a tape (1/3)



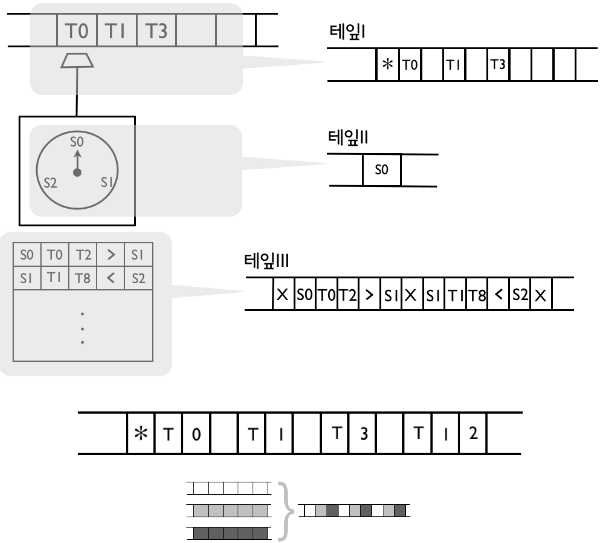
Univeraal TM: a Turing machine

Expressing a TM in a tape (2/3)

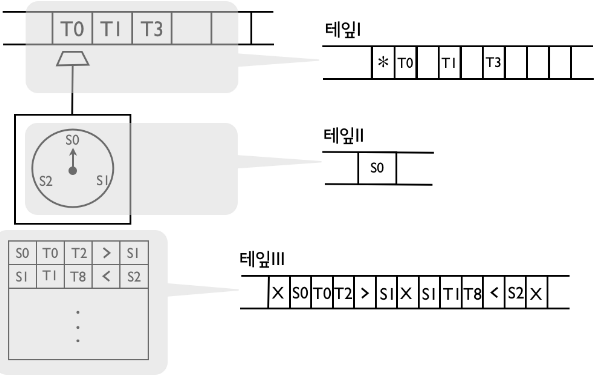


Universal TM: a Turing machine

Expressing a TM in a tape (3/3)



Universal TM: execution rules



- read tape-I, tape-II
- look for match in tape-III

S1	T1	T2	>	S2

- do as specified in the matched rule

$|\text{TMs}| \leq |\mathbb{N}|$

The number of TMs cannot be more than $|\mathbb{N}|$

- how many symbols for expressing TM? S,T,<,>,||, 0,$\cdots$,9, X,$*$
- a finite sequence: 17-ary number (a natural number)

Turing's Proof, by Universal TM and $|\text{TMs}| \leq |\mathbb{N}|$

- Lemma1 [$\exists$VERI $\implies \exists H$]. If a TM can generate all true formulas, then it can solve the halting problem.
- Lemma2 [$\nexists H$]. No TM can solve the halting problem.

Thus no TM can generate all true formulas.

Lemma1 proof: ∃VERI $\implies \exists H$

If a TM can generate all true formulas, then it can solve the halting problem.
Proof. $H(M) =$

1. run VERI by Universal TM
2. because VERI generates all true formulas, it generates either "M halts" or "not(M halts)."
3. answer accordingly. QED

Lemma2 proof: $\nexists H$

Proof. All TM and its inputs can be indexed by natural numbers: $M_1, M_2, \cdots, I_1, I_2, \cdots$.

1. If H exists, we can fill the following table.

	Input			
	I_1	I_2	I_3	$\cdots$
M_1	1	1	0	$\cdots$
M_2	1	0	1	$\cdots$
M_3	1	0	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$

2. Then, following TM M is different from all TMs

$$M(n) = \textit{Table}(M_n, I_n) \times U(M_n, I_n) + 1$$

Contradiction. Hence H is impossible. QED

Turing's 1936 Paper: Wrap up

"*On Computable Numbers, with an Application to the Entscheitungsproblem*"

1. define mechanical computation: turing machine
2. persuade us that TM is enough
3. assume machine VERI that generate all true formulas
4. show that machine VERI can solve the halting problem
5. prove that the halting problem is not computable

Hence VERI is not possible. QED

How Turing come up with the 1936 paper?

"Genius" Turing?

- What talents can generate "foundational knowledge"?
- Only "genius" can do that?
- Misleading message to students
- Not encouraging
- Maybe not true either

Did the 1936 paper come from an epiphany available only to "geniuses"?

Followings are my investigation on Turing's 1935

Turing's 1935

- Turing took Max Newman's class (Foundations of Mathematics and Gödel's Theorem) in 1935
- Turing learned about Gödel's Incompleteness proof there
- Turing was puzzled; why not more down-to-earth approach?
- Turing began his own style of the same proof

Given a 1st-order finite proof system about natural numbers, the incompleteness holds if such X exists as

$$X \text{ is not provable} = X$$

- X is either true or false.
- Suppose X is false, then X is provable.
  - inconsistent system, out of our discussion
- Suppose X is true, then X is not provable
  - only this is possible

$$X \text{ is not provable} = X$$

Is such X a 1st-order assertion about natural numbers? Gödel showed yes.

- unique natural numbers $\underline{f}$ and $\underline{p}$ for every 1st-order assertion f about natural numbers and every its proof tree p
- "X is not provable" is also an assertion about natural number: "$\underline{X}$ is a factor of a proof"

For $X = \mathit{UnProvable}(\underline{X})$

Gödel proved such X is

$$\begin{aligned} X &\stackrel{\text{def}}{=} G[x \mapsto k] \\ k &\stackrel{\text{def}}{=} \underline{G} \\ G &\stackrel{\text{def}}{=} \mathit{UnProvable}(\underline{\mathit{subst}(x, n, x)}) \quad (\text{Note } \mathit{fv}(G) = \{x\}) \\ & \quad \text{where } n = \underline{x} \text{ and } \mathit{subst}(a : \mathbb{N}, b : \mathbb{N}, c : \mathbb{N}) = \overline{a}[\overline{b} \mapsto c] \end{aligned}$$

because

For $X = \mathit{UnProvable}(\underline{X})$

Gödel proved such X is

$$
\begin{array}{rcl}
X & \stackrel{\text{def}}{=} & G[x \mapsto k] \\
k & \stackrel{\text{def}}{=} & \underline{G} \\
G & \stackrel{\text{def}}{=} & \mathit{UnProvable}(\underline{\mathit{subst}(x, n, x)}) \quad (\text{Note } \mathit{fv}(G) = \{x\}) \\
 & & \text{where } n = \underline{x} \text{ and } \mathit{subst}(a : \mathbb{N}, b : \mathbb{N}, c : \mathbb{N}) = \overline{a}[\overline{b} \mapsto c]
\end{array}
$$

because

$$
\begin{array}{rcl}
X & = & G[x \mapsto k] \\
 & = & (\mathit{UnProvable}(\underline{\mathit{subst}(x, n, x)}))[x \mapsto k] \\
 & = & \mathit{UnProvable}(\underline{\mathit{subst}(k, n, k)}) \\
 & = & \mathit{UnProvable}(\underline{G[x \mapsto k]}) \\
 & = & \mathit{UnProvable}(\underline{X}) \qquad \text{QED}
\end{array}
$$

$$
\begin{aligned}
X &\stackrel{\text{def}}{=} G[x \mapsto k] \\
k &\stackrel{\text{def}}{=} \underline{G} \\
G &\stackrel{\text{def}}{=} \mathit{UnProvable}(\underline{\mathit{subst}(x, n, x)}) \\
&\quad \text{where } n = \underline{x} \text{ and } \mathit{subst}(a : \mathbb{N}, b : \mathbb{N}, c : \mathbb{N}) = \overline{a}[\overline{b} \mapsto c]
\end{aligned}
$$

- no nonsense: G has x replaced by itself(encoding $\underline{G}$ of G)
- two points from equation $X = \mathit{UnProvable}(X)$
  - an assertion about self is expressable within the given proof system
  - can interpret it as specifying an infinite object: a fixpoint of *UnProvable*: $X \stackrel{\text{def}}{=} \underbrace{\mathit{UnProvable}(\mathit{UnProvable}(\cdots))}_{\infty}$

$$
\begin{aligned}
x &= x + 1 \\
x &= x \multimap
\end{aligned}
$$

My Guess about Turing's Thought Process (1/3)

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.
- What would its correspondence in TM world?

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.
- What would its correspondence in TM world?
  - Machine about machines

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.
- What would its correspondence in TM world?
  - Machine about machines
  - Machine about machines that runs infinitely

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.
- What would its correspondence in TM world?
  - Machine about machines
  - Machine about machines that runs infinitely
- Machine that decides whehter machines would run infinitely or not

My Guess about Turing's Thought Process (1/3)

- Something infinite in TM must be impossible.
  - Look, something infinite "true yet unprovable assertion" in Gödel's proof.
- What is infinite in TM then?
  - Every TM is finite, except for its runs
- What would be impossible for TM that runs infinitely?
- See, $X = \mathit{UnProvable}(X)$: this is about self.
- What would its correspondence in TM world?
  - Machine about machines
  - Machine about machines that runs infinitely
- Machine that decides whehter machines would run infinitely or not
- BTW, how a machine can see machines? A machine that has machines as its inputs?

My Guess about Turing's Thought Process (2/3)

- Can a machine have a machine as an input? Sure, we can express everying in finite symbols (encodings). And, Gödel too encodes assertions in natural numbers.

My Guess about Turing's Thought Process (2/3)

- Can a machine have a machine as an input? Sure, we can express everying in finite symbols (encodings). And, Gödel too encodes assertions in natural numbers.
- Would it be impossible to decide whether the input machine will run infinitely or not?

My Guess about Turing's Thought Process (2/3)

- Can a machine have a machine as an input? Sure, we can express everying in finite symbols (encodings). And, Gödel too encodes assertions in natural numbers.
- Would it be impossible to decide whether the input machine will run infinitely or not?
- Would it be impossible? I don't know for now. What would be possible? What would be possible, given a machine as an input?

My Guess about Turing's Thought Process (2/3)

- Can a machine have a machine as an input? Sure, we can express everying in finite symbols (encodings). And, Gödel too encodes assertions in natural numbers.
- Would it be impossible to decide whether the input machine will run infinitely or not?
- Would it be impossible? I don't know for now. What would be possible? What would be possible, given a machine as an input?
- "Univeral Machine" is possible. A machine that mimicks the runs of the input machine.

My Guess about Turing's Thought Process (2/3)

- Can a machine have a machine as an input? Sure, we can express everying in finite symbols (encodings). And, Gödel too encodes assertions in natural numbers.
- Would it be impossible to decide whether the input machine will run infinitely or not?
- Would it be impossible? I don't know for now. What would be possible? What would be possible, given a machine as an input?
- "Univeral Machine" is possible. A machine that mimicks the runs of the input machine.
- Possibility expanded, now back to impossibility. I hope the halting problem is impossible. How can I prove it?

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

And he proves the halting problem is not computable by TM.

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

And he proves the halting problem is not computable by TM.

Then he proves the goal:

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

And he proves the halting problem is not computable by TM.

Then he proves the goal:

- $\exists$VERI $\implies \exists H$
  - run VERI and wait & see
  - VERI will eventually print either "M halts" or "not(M halts)"

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

And he proves the halting problem is not computable by TM.

Then he proves the goal:

- $\exists$VERI $\implies \exists H$
  - run VERI and wait & see
  - VERI will eventually print either "M halts" or "not(M halts)"
- However, $\nexists H$.

My Guess about Turing's Thought Process (3/3)

- One thing I didn't borrow from Gödel's proof is the diagonalization technique.
- Yet, Gödel's diagonalization does not fit with TM because it was about assertion formulas. How about Cantor's diagonalization?

And he proves the halting problem is not computable by TM.

Then he proves the goal:

- $\exists$VERI $\implies \exists H$
  - run VERI and wait & see
  - VERI will eventually print either "M halts" or "not(M halts)"
- However, $\nexists H$.
- Hence $\nexists$VERI. QED

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
  - a teacher who taught the details of Gödel's proof

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
  - a teacher who taught the details of Gödel's proof
  - a student with self-esteem ("chutzpah" spirit, maybe)

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
  - a teacher who taught the details of Gödel's proof
  - a student with self-esteem ("chutzpah" spirit, maybe)
  - helps at every its crisis from Gödel's proof

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
    - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
    - a teacher who taught the details of Gödel's proof
    - a student with self-esteem ("chutzpah" spirit, maybe)
    - helps at every its crisis from Gödel's proof
    - inventing "universal machine" as a tool during this process

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
  - a teacher who taught the details of Gödel's proof
  - a student with self-esteem ("chutzpah" spirit, maybe)
  - helps at every its crisis from Gödel's proof
  - inventing "universal machine" as a tool during this process
  - a teacher who archived Turing's down-to-earth-style proof

Conclusion

My guess about Turing's 1935, on how he came up with his 1936 paper:

- Each step of the paper matches with underlying ideas of Gödel's proof
- Turing replayed them in his machine world.
  - Did the 1936 paper come from an epiphany available only to "geniuses"? No fear, maybe not.
- Turing's 1936 paper, by a collective work than by a single "genius"
  - a teacher who taught the details of Gödel's proof
  - a student with self-esteem ("chutzpah" spirit, maybe)
  - helps at every its crisis from Gödel's proof
  - inventing "universal machine" as a tool during this process
  - a teacher who archived Turing's down-to-earth-style proof

Thank you.

정리

제 추측이긴 하지만 이게 튜링의 1935년이 아니었을까.

- 거울같은 짝
  - 튜링 논문의 고비와 실마리들 = 괴델 논문의 기술들
- 있었던 일
  - 괴델증명을 자세히 강의해 준 선생님
  - 자의식 넘친 우등생
  - 그 학생의 줏대있는 행보
  - 고비마다 방향을 잡아준 괴델증명 강의노트
  - 자연스런 소품으로 등장한 컴퓨터의 원천 설계도
  - 색다른 증명을 기록으로 남기도록 도운 선생님
- 이제 접어도 되지않을까
  - 튜링을 천재라고, 불필요하게 주변을 겁주지 말자
  - 튜링이 천재이기 때문에? 꼭 그렇지만은 않은듯

결론

비슷한 성과는 우리 주변에서도 싹틀 수 있다고 본다.

결론

비슷한 성과는 우리 주변에서도 싹틀 수 있다고 본다.

"원천지식", "탈추격", "선진국형" 연구 성과
vivacity

결론

비슷한 성과는 우리 주변에서도 싹틀 수 있다고 본다.

“원천지식”, “탈추격”, “선진국형” 연구 성과
vivacity

감사합니다. QnA