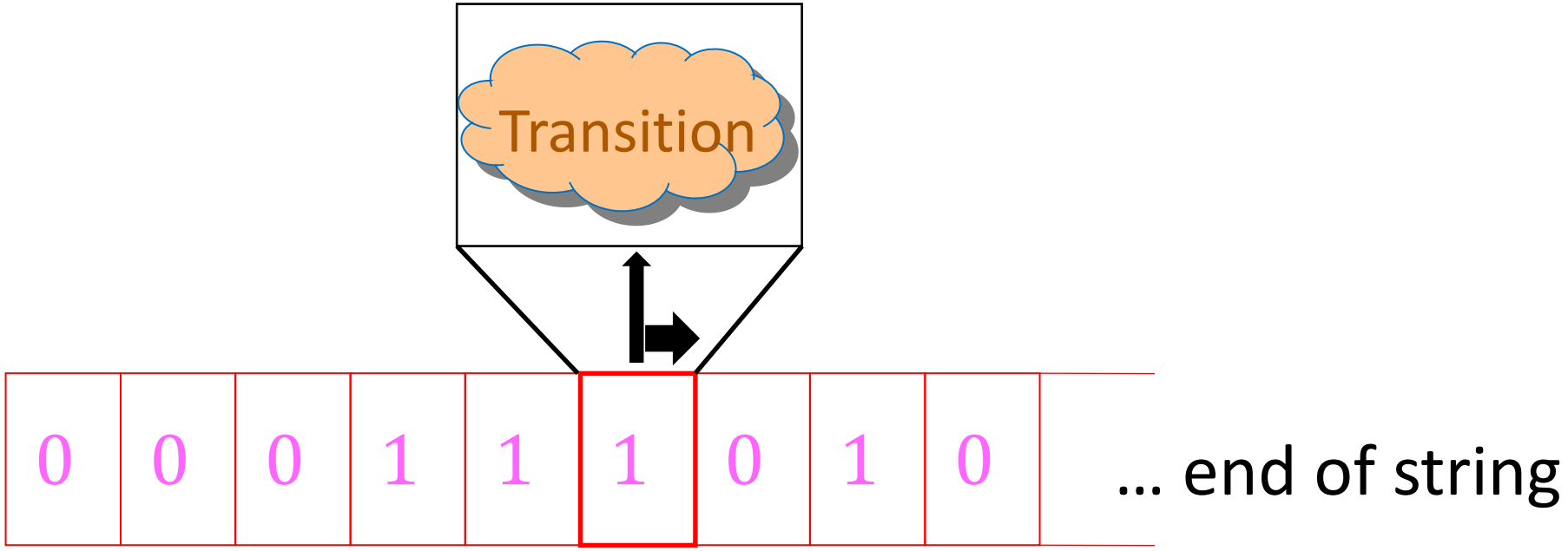
PS7 due next Tue (April 1)
Coming soon: PS8, PRR9

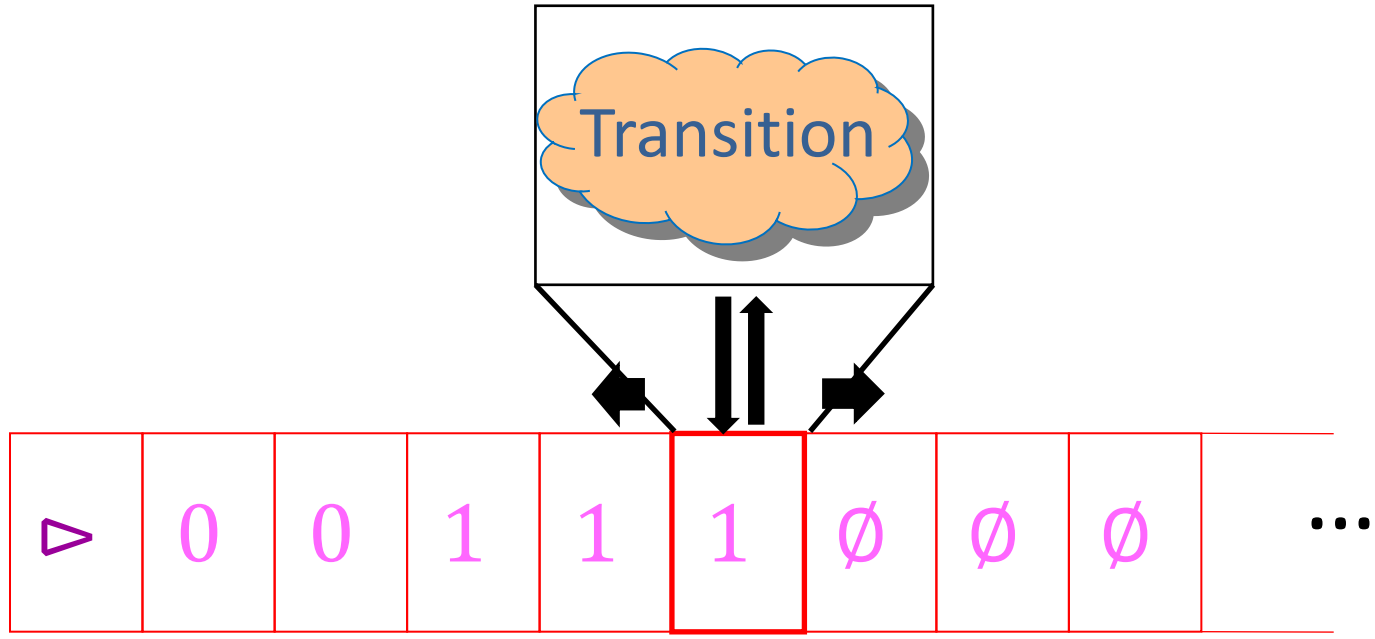
Class 18:
Computability

University of Virginia
cs3120: DMT2
Wei-Kai Lin

Recall: How DFA worked



Recall: Turing Machine



Tape initially contains the input followed by blanks

If transition "halts", → tape contains the output

Semi-infinite tape (memory)

Plan for today

Main question:
What functions can a Turing Machine compute?

Turing Machine model

- formal description and intuition

Are there functions that cannot be computed by a TM?

An interesting function that cannot be computed by a TM

Recap: Turing Machine model

A *Turing Machine,* is defined by (Σ, k, δ):

$\boldsymbol{k} \in \mathbb{N}$: a finite number of states

Σ : alphabet $-$ finite set of symbols

$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$

$\boldsymbol{\delta}$: transition function

$\boldsymbol{\delta}: [k] \times \Sigma \to [k] \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{H}\}$

Write dir to move

Definition. The execution of a TM, $M = (\Sigma, k, \delta)$ on input $x \in \{0,1\}^*$ is this process:

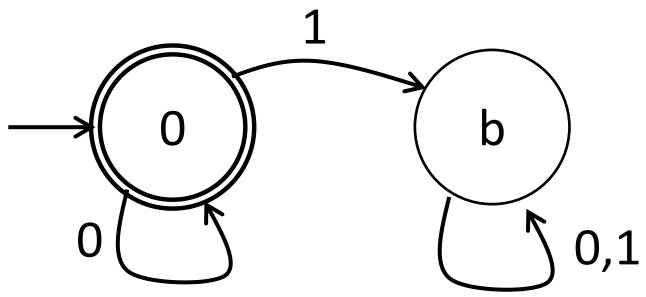
1. Initialize T as $\triangleright, x_0, x_1, \ldots, x_{n-1}, \emptyset, \emptyset, \emptyset, \ldots$ where $n = |x|$.
2. Initialize two natural number variables, $i = 0, s = 0$.
3. **repeat**
    1. $(s', \sigma', D) = \boldsymbol{\delta}(s, T[i])$
    2. $s := s', T[i] := \sigma'$
    3. if D = $\mathbf{R}$: $i := i + 1$
       if D = $\mathbf{L}$: $i := \max\{i - 1, 0\}$
       if D = $\mathbf{H}$: **break**
4. If the process finishes, the output is:

   $M(x) = T[1], \ldots, T[m]$ where $m > 0$ is the smallest integer, $T[m+1] \notin \{0,1\}$.

   Otherwise (i.e., the machine does NOT halt), $M(x) = \perp$.

Example: From DFA to Turing Machines

- DFA




```
0 0 0 r 0
0 1 1 r b
0 _ _ * halt-accept
b 0 0 r b
b 1 1 r b
b _ _ * halt-reject
```

Current state | Current symbol | Write symbol | Move direction | Next state

Was this really Turing's model?

A *Turing Machine,* is defined by (Σ, k, δ):

$\boldsymbol{k} \in \mathbb{N}$: a finite number of states

Σ : alphabet $-$ finite set of symbols

$$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$$

$\boldsymbol{\delta}$: transition function

$$\boldsymbol{\delta}: [k] \times \Sigma \to [k] \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{H}\}$$

Definition. The execution of a TM, $M = (\Sigma, k, \delta)$ on input $x \in \{0,1\}^*$ is this process:

1. Initialize T as $\triangleright, x_0, x_1, \ldots, x_{n-1}, \emptyset, \emptyset, \emptyset, \ldots$ where $n = |x|$.
2. Initialize two natural number variables, $i = 0, s = 0$.
3. **repeat**
    1. $(s', \sigma', D) = \boldsymbol{\delta}(s, T[i])$
    2. $s := s', T[i] := \sigma'$
    3. if D = $\mathbf{R}$: $i := i + 1$
       if D = $\mathbf{L}$: $i := \max\{i - 1, 0\}$
       if D = $\mathbf{H}$: **break**
4. If the process finishes, the output is:

   $M(x) = T[1], \ldots, T[m]$ where $m > 0$ is the smallest integer, $T[m+1] \notin \{0,1\}$.

   Otherwise (i.e., the machine does NOT halt), $M(x) = \perp$.


Bank of England
AD47 858569
Final
m-config. Symbol Operations m-config.
q_i S_j PS_k, L q_m (N_1)
q_i S_j PS_k, R q_m (N_2)
q_i S_j PS_k q_m (N_3)
$q_1 S_0 S_1 Rq_2$; $q_2 S_0 S_0 Rq_3$; $q_3 S_0 S_2 Rq_4$; $q_4 S_0 S_0 Rq_1$;.
1001000111100001111101111
Fifty Pounds
This is only a foretaste of what is to come
and only the shadow of what is going to be
Alan Turing 1912-1954
© THE GOVERNOR AND COMPANY OF THE BANK OF ENGLAND 2020
£50
BANK OF ENGLAND £50 2023
AD47 858569

https://blog.cloudflare.com/details-of-the-cloudflare-outage-on-july-2-2019/

From Class 16

CLOUDFLARE The Cloudflare Blog

Subscribe to receive notifications of new posts:

Email Address Subscribe

Product News | Speed & Reliability | Security | Serverless | Zero Trust | Developers | Deep Dive | Life @Cloudflare

Details of the Cloudflare outage on July 2, 2019

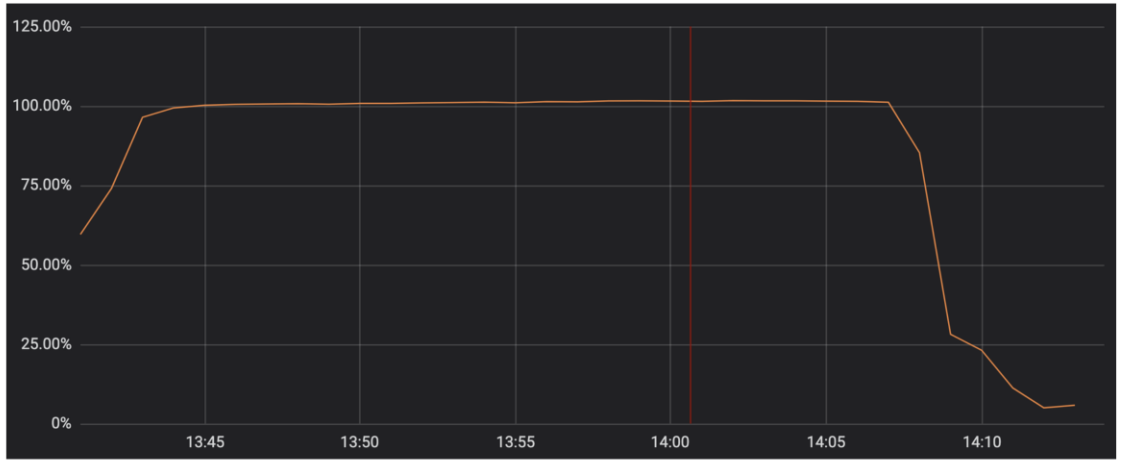
07/12/2019

John Graham-Cumming

1 alert for Global Traffic Drop

02/07/2019, 13:48 1 alert

Service: Escalate Edge SRE #53967



CPU utilization in one of our PoPs during the incident

This resulted in our customers (and their customers) seeing a 502 error page when visiting any Cloudflare domain. The 502 errors were generated by the front line Cloudflare web servers that still had CPU cores available but were unable to reach the processes that serve HTTP/HTTPS traffic.

502 Bad Gateway

cloudflare

80% of Cloudflare's customers traffic dropped due to excessive cost of matching a regular expression!

John Graham-Cumming's blog

https://blog.jgc.org/2009/06/alan-turing-deserves-apology-from.html

2009-06-23

Alan Turing deserves an apology from the British Government

When I started writing The Geek Atlas there was one name that was getting in the book no matter what: Alan Turing.

Alan Turing matters on many levels because he was, in the words of the memorial in Manchester:

> Father of computer science, mathematician, logician, wartime codebreaker, victim of prejudice

Turing's work has affected us all. He's best know for his involvement in Second World War code breaking (especially for helping to break Engima) and if all he had done was that we would be grateful.

But Turing was also a critical pioneer of computer science. He defined a theoretical model of computers (at a time when 'computer' meant a person, often a woman, who computed numbers) that holds true today. He suggested how we might determine whether a computer was sentient (with the Turing Test).

Turing's death should remind us how prejudice ruins and degrades.

Alan Turing was gay. And he was prosecuted for 'indecent acts' and eventually took his own life aged 41. This man, younger than me, killed himself because at the time homosexuality was illegal and having been prosecuted he was chemically castrated in an attempt to 'cure' him. He had been stripped of his security clearance.

For years, his legacy was largely ignored outside the computer community. To quote Wikipedia:

> In 1994 a stretch of the A6010 road (the Manchester city intermediate ring road) was named Alan Turing Way. A bridge carrying this road was widened, and carries the name 'Alan Turing Bridge'.

A frikkin' Ring Road!

It wasn't until 2001 that a statue was erected.

John Graham-Cumming's blog

https://blog.jgc.org/2009/06/alan-turing-deserves-apology-from.html

2009-06-23

Alan Turing deserves an apology from the British Government

When I started writing The Geek Atlas there was one name that was getting in the book no matter what: Alan Turing.

Alan Turing matters on many levels because he was, in the words of the memorial in Manchester:

> Father of computer science, mathematician, logician, wartime codebreaker, victim of prejudice

Turing's work has affected us all. He's best know for his involvement in Second World War code breaking (especially for helping to break Engima) and if all he had done was that we would be grateful.

But Turing was also a critical pioneer of computer science. He defined a theoretical model of computers (at a time when 'computer' meant a person
determine whether a comp

Turing's death should rem

Alan Turing was gay. And
than me, killed himself be
in an attempt to 'cure' him

For years, his legacy was

> In 1994 a stretch of t
> bridge carrying this road was widened, and carries the name 'Alan Turing Bridge'.

A frikkin' Ring Road!

It wasn't until 2001 that a statue was erected.

Today is Alan Turing's 97th birthday. Or at least it could have been if it were not for his prosecution and untimely death.

Isn't it time the British Government apologized for the way he was treated? We shouldn't let this anniversary of his death go by without recognizing the great works this man did and the ignominious way in which he was treated.

at June 23, 2009

Labels: alan turing, rants and raves

https://web.archive.org/web/20100301233733/http://petitions.number10.gov.uk/turing/

Number10.gov.uk BETA

The official site of the Prime Minister's Office

10

We the undersigned petition the Prime Minister to apologize for the prosecution of Alan Turing that led to his untimely death. More details

Submitted by John Graham-Cumming – **Deadline to sign up by:** 20 January 2010 – **Signatures:** 32,280

More details from petition creator

Alan Turing was the greatest computer scientist ever born in Britain. He laid the foundations of computing, helped break the Nazi Enigma code and told us how to tell whether a machine could think.

He was also gay. He was prosecuted for being gay, chemically castrated as a 'cure', and took his own life, aged 41.

The British Government should apologize to Alan Turing for his treatment and recognize that his work created much of the world we live in and saved us from Nazi Germany. And an apology would recognize the tragic consequences of prejudice that ended this man's life and career.

https://web.archive.org/web/20100301233733/http://petitions.number10.gov.uk/turing/

Number10.gov.uk BETA

The official site of the Prime Minister's Office

10

Petition update, 11 September 2009, while petition was still open

Thank you for signing this petition. The Prime Minister has written a response. Please read below.

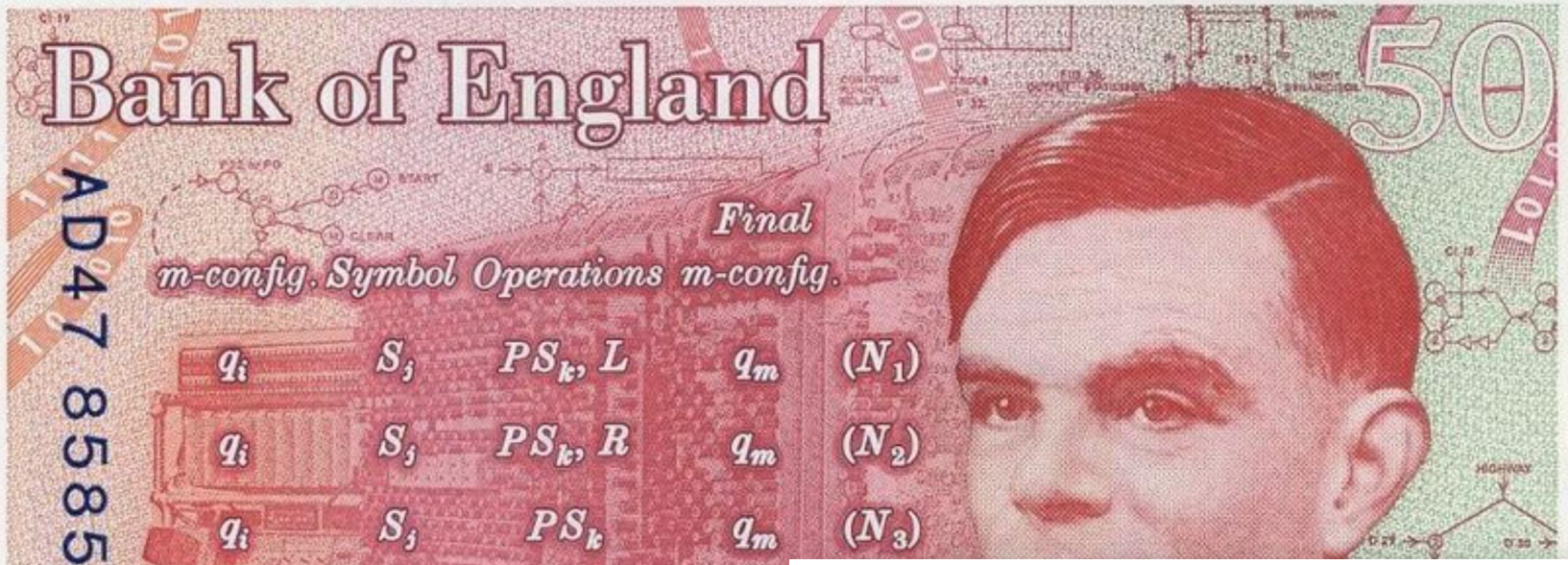
Prime Minister: 2009 has been a year of deep reflection – a chance for Britain, as a nation, to commemorate the profound debts we owe to those who came before. A unique combination of anniversaries and events have stirred in us that sense of pride and gratitude which characterise the British experience. Earlier this year I stood with Presidents Sarkozy and Obama to honour the service and the sacrifice of the heroes who stormed the beaches of Normandy 65 years ago. And just last week, we marked the 70 years which have passed since the British government declared its willingness to take up arms against Fascism and declared the outbreak of World War Two. So I am both pleased and proud that, thanks to a coalition of computer scientists, historians and LGBT activists, we have this year a chance to mark and celebrate another contribution to Britain's fight against the darkness of dictatorship; that of code-breaker Alan Turing.

But even more than that, Alan deserves recognition for his contribution to humankind. For those of us born after 1945, into a Europe which is united, democratic and at peace, it is hard to imagine that our continent was once the theatre of mankind's darkest hour. It is difficult to believe that in living memory, people could become so consumed by hate – by anti-Semitism, by homophobia, by xenophobia and other murderous prejudices – that the gas chambers and crematoria became a piece of the European landscape as surely as the galleries and universities and concert halls which had marked out the European civilisation for hundreds of years. It is thanks to men and women who were totally committed to fighting fascism, people like Alan Turing, that the horrors of the Holocaust and of total war are part of Europe's history and not Europe's present.

So on behalf of the British government, and all those who live freely thanks to Alan's work I am very proud to say: we're sorry, you deserved so much better.

Gordon Brown


Bank of England
AD47 858569
Final
m-config. Symbol Operations m-config.
q_i S_j PS_k, L q_m (N_1)
q_i S_j PS_k, R q_m (N_2)
q_i S_j PS_k q_m (N_3)
$q_1 S_0 S_1 Rq_2$; $q_2 S_0 S_0 Rq_3$; $q_3 S_0 S_2 Rq_4$; $q_4 S_0 S_0 Rq_1$;.
1001000111100001111101111
Fifty Pounds
This is only a foretaste of what is to come
and only the shadow of what is going to be
Alan Turing 1912-1954
© THE GOVERNOR AND COMPANY OF THE BANK OF ENGLAND 2020
£50
BANK OF ENGLAND £50 2023
AD47 858569

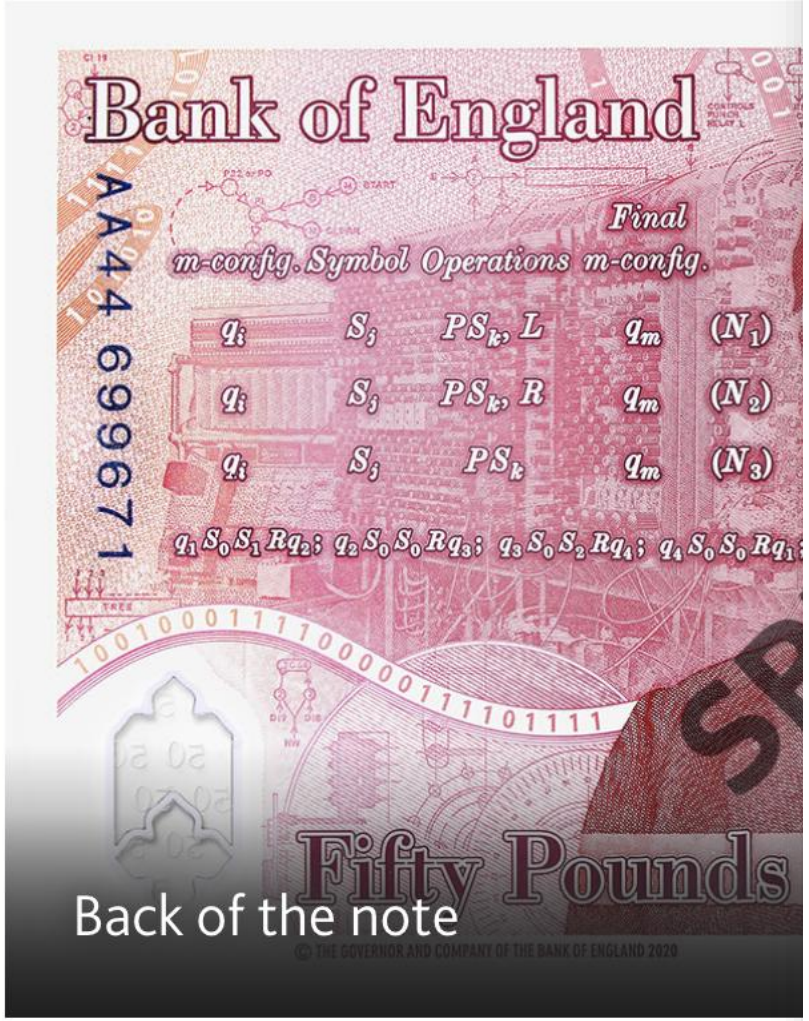


and, in particular, blank $= S_0$, $0 = S_1$, $1 = S_2$. The lines of the table are now of form

m-config.	*Symbol*	*Operations*	*Final m-config.*	
q_i	S_j	PS_k, L	q_m	(N_1)
q_i	S_j	PS_k, R	q_m	(N_2)
q_i	S_j	PS_k	q_m	(N_3)

Lines such as

https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf



Let us find a description number for the machine I of § 3. When we rename the m-configurations its table becomes:

q_1	S_0	PS_1, R	q_2
q_2	S_0	PS_0, R	q_3
q_3	S_0	PS_2, R	q_4
q_4	S_0	PS_0, R	q_1

Other tables could be obtained by adding irrelevant lines such as

q_1	S_1	PS_1, R	q_2

Our first standard form would be

$$q_1 S_0 S_1 R q_2;\ q_2 S_0 S_0 R q_3;\ q_3 S_0 S_2 R q_4;\ q_4 S_0 S_0 R q_1;.$$

The standard description is

$$DADDCRDAA;DAADDRDAAA;$$

$$DAAADDCCRDAAAA;DAAAADDRDA;$$

A description number is

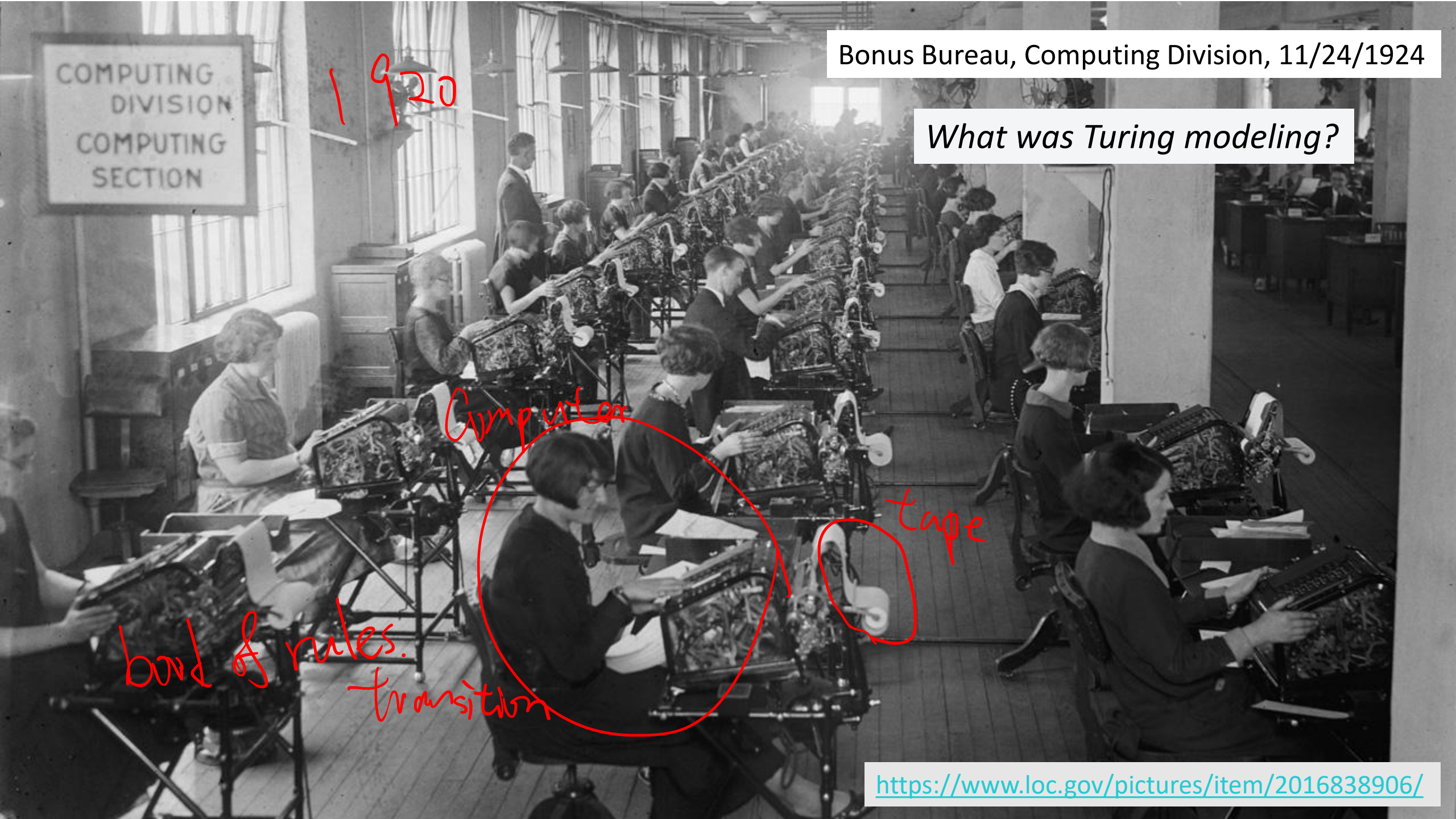
313325311731133531117311133225311117311113353173132325311 7

and so is

313325311731133531117311133225311117311113353173132325311 7

Configuration		*Behaviour*		
m-config.	*symbol*	*operations*	*final*	*m-config.*
𝔟	None	$P0, R$		𝔠
𝔠	None	R		𝔠
𝔢	None	$P1, R$		𝔣
𝔣	None	R		𝔟

Warning: Turning uses “Configuration” to mean something different from modern usage (including last week’s cohort problems)!

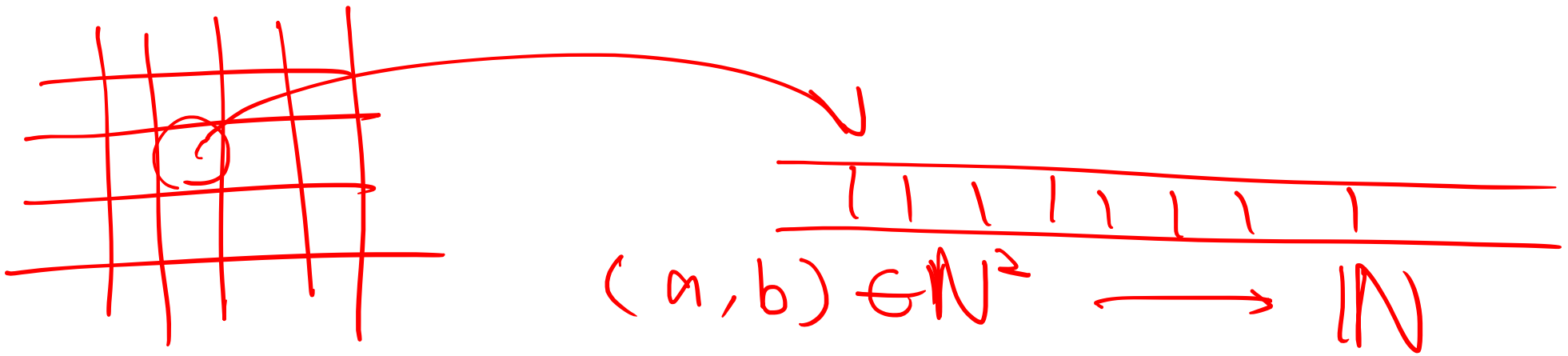


Bonus Bureau, Computing Division, 11/24/1924

What was Turing modeling?

https://www.loc.gov/pictures/item/2016838906/

Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also



Why model (human) computers?

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

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means.

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

Why is the *alphabet* finite?

A *Turing Machine,* is defined by (Σ, k, δ):

$\boldsymbol{k} \in \mathbb{N}$: a finite number of states

Σ : alphabet $-$ finite set of symbols

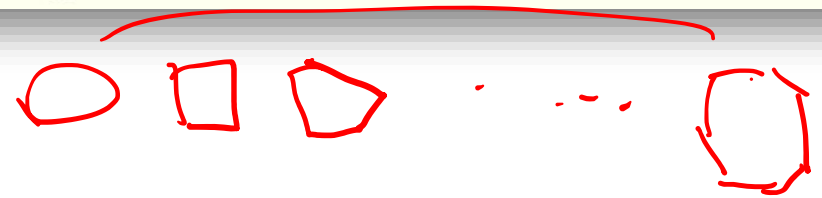
$$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$$

$\boldsymbol{\delta}$: transition function

$$\boldsymbol{\delta}: [k] \times \Sigma \rightarrow [k] \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{H}\}$$

will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent†. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as

17 or 99999999999999 is normally treated as a single symbol. Similarly in any European language words are treated as single symbols (Chinese, however, attempts to have an enumerable infinity of symbols). The differences from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, cannot be observed at one glance. This is in accordance with experience. We cannot tell at a glance whether 9999999999999999 and 99999999999999 are the same.



Why is the *number of states* finite?

A *Turing Machine,* is defined by (Σ, k, δ):

$\boldsymbol{k} \in \mathbb{N}$: a finite number of states

Σ : alphabet $-$ finite set of symbols

$$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$$

$\boldsymbol{\delta}$: transition function

$$\boldsymbol{\delta}: [k] \times \Sigma \rightarrow [k] \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{H}\}$$

Number of States of Mind

Follow simple rules
Remember what you are doing

"For the present I shall only say that the justification lies in the fact that the ***human memory is necessarily limited***."

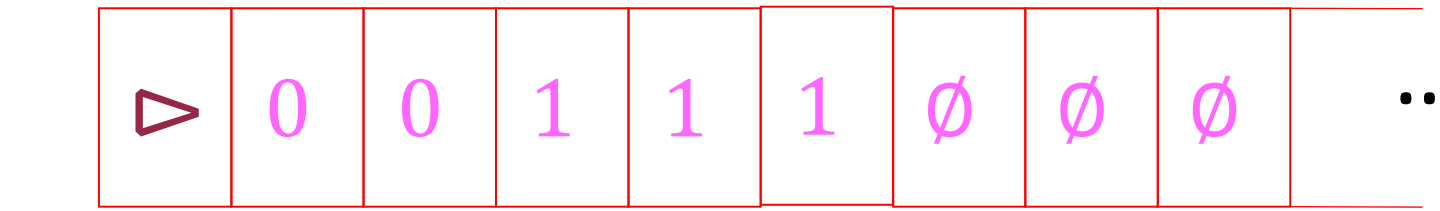
Why is the *number of states* finite?

The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

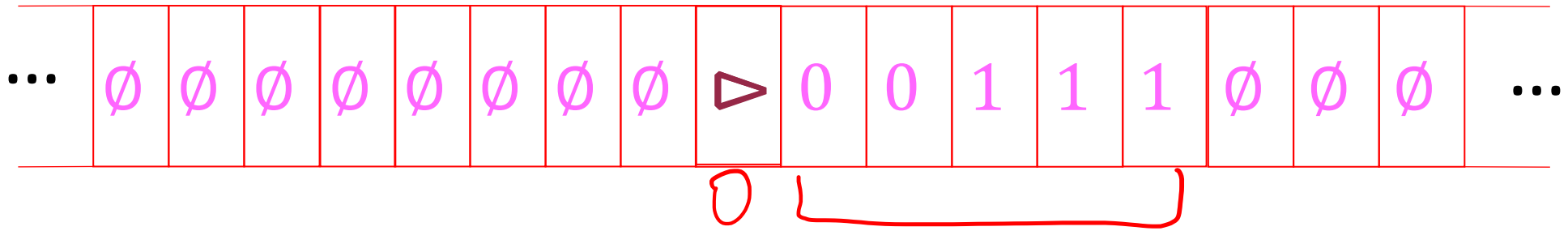
More Turing Machines

- $\{0^n 1^n : n \in \mathbb{N}\}$
  https://morphett.info/turing/turing.html?4fcc9a5cd3f4ef9d83fd6ebed105c73b
- Infinite loop
  https://morphett.info/turing/turing.html?459fca2a14f95f7d3acb789abc241261
- 4-state busy beaver
  http://morphett.info/turing/turing.html?4fee0005bc73130d8138803406a74990

Alternative TN Model: Two-Way Infinite Tape



Semi-infinite tape



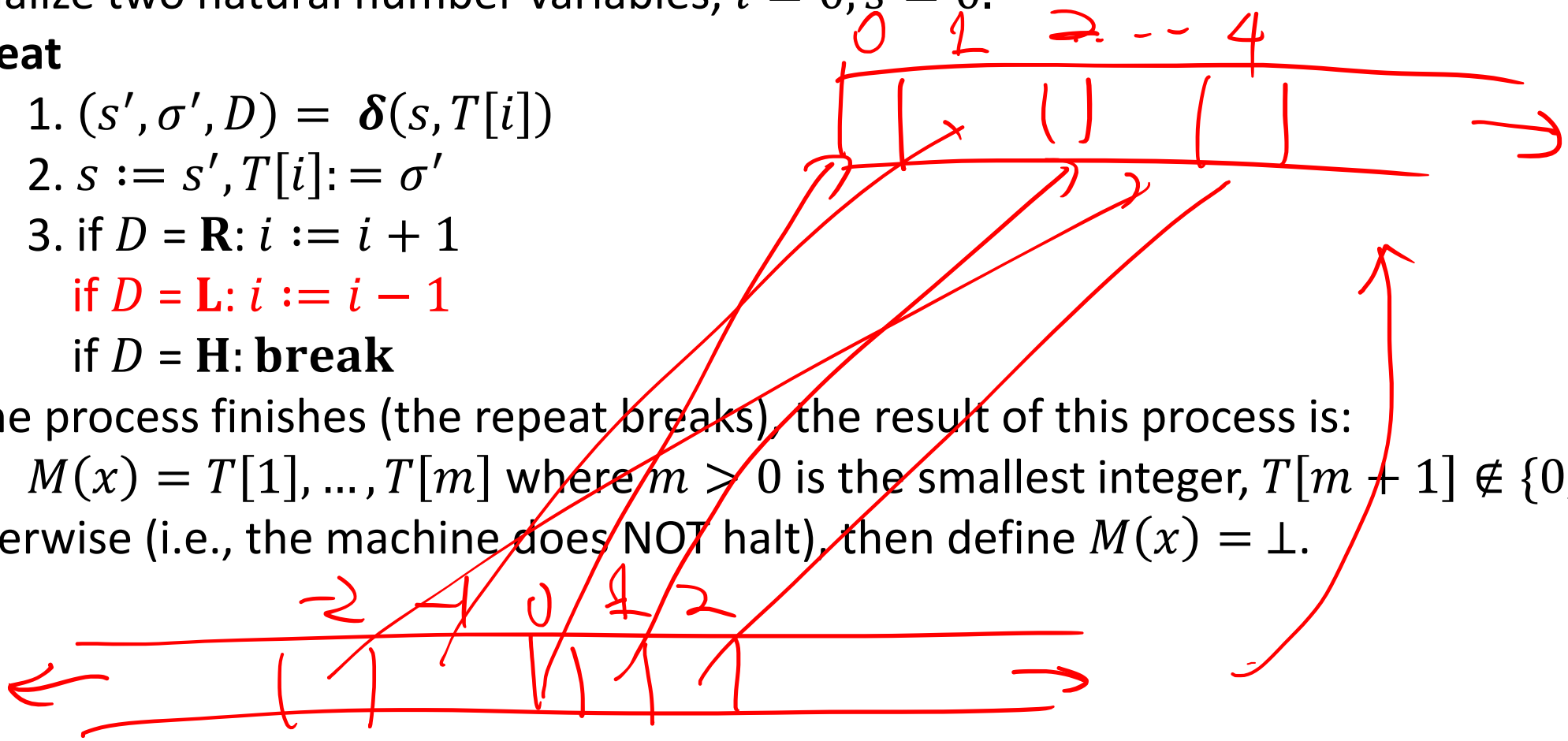
Two-way infinite tape

Definition. The **output** of the execution of a TM, $M = (\Sigma, k, \delta)$ is the result of this process:

1. Initialize $T[i] = \emptyset$ for all $i \in \mathbb{Z}$.
2. Initialize two natural number variables, $i = 0, s = 0$.
3. **repeat**
    1. $(s', \sigma', D) = \boldsymbol{\delta}(s, T[i])$
    2. $s := s', T[i] := \sigma'$
    3. if D = **R**: $i := i + 1$
       if D = **L**: $i := i - 1$
       if D = **H**: **break**
4. If the process finishes (the repeat breaks), the result of this process is:

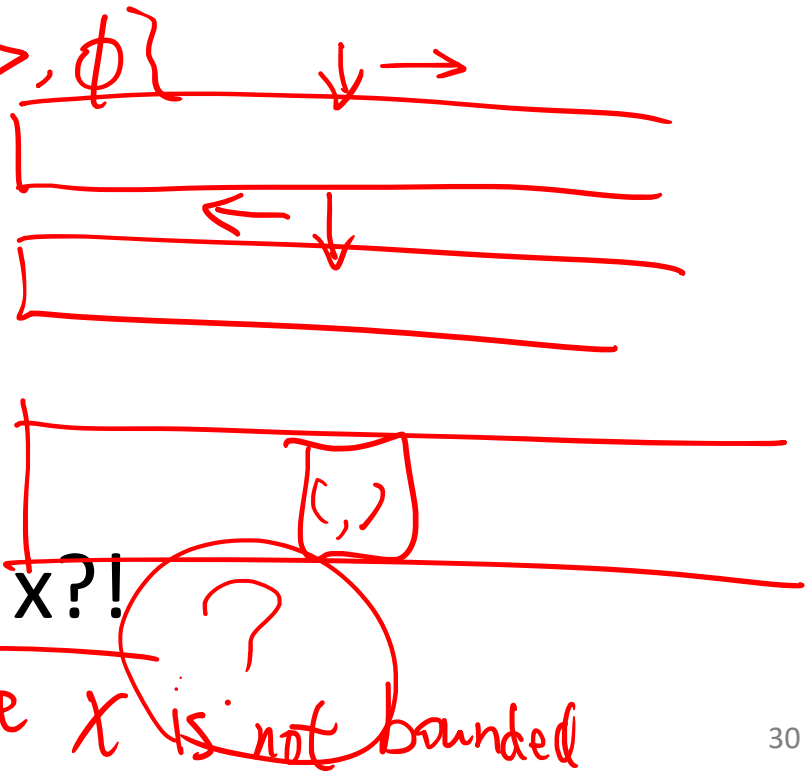
    $M(x) = T[1], \dots, T[m]$ where $m > 0$ is the smallest integer, $T[m + 1] \notin \{0,1\}$.

    Otherwise (i.e., the machine does NOT halt), then define $M(x) = \perp$.



More Turing Machine models

- 2, 3, 4, or a constant number of tapes
- Large (constant) alphabet
- 2-dimensional tape?!
- Move tape by arbitrary distance x?!



Computability

Definition 7.1 (Turing Machine)

A (one tape) *Turing machine* with k states and alphabet $\Sigma \supseteq \{0, 1, \triangleright, \varnothing\}$ is represented by a *transition function* $\delta_M : [k] \times \Sigma \to [k] \times \Sigma \times \{\mathsf{L}, \mathsf{R}, \mathsf{S}, \mathsf{H}\}$.

For every $x \in \{0,1\}^*$, the *output* of M on input x, denoted by $M(x)$, is the result of the following process:

- We initialize T to be the sequence $\triangleright, x_0, x_1, \ldots, x_{n-1}, \varnothing, \varnothing, \ldots$, where $n = |x|$. (That is, $T[0] = \triangleright$, $T[i+1] = x_i$ for $i \in [n]$, and $T[i] = \varnothing$ for $i > n$.)
- We also initialize $i = 0$ and $s = 0$.
- We then repeat the following process:
  1. Let $(s', \sigma', D) = \delta_M(s, T[i])$.
  2. Set $s \to s'$, $T[i] \to \sigma'$.
  3. If $D = \mathsf{R}$ then set $i \to i + 1$, if $D = \mathsf{L}$ then set $i \to \max\{i - 1, 0\}$. (If $D = \mathsf{S}$ then we keep i the same.)
  4. If $D = \mathsf{H}$, then halt.
- If the process above halts, then M's output, denoted by $M(x)$, is the string $y \in \{0,1\}^*$ obtained by concatenating all the symbols in $\{0,1\}$ in positions $T[0], \ldots, T[i]$ where $i + 1$ is the first location in the tape containing $\varnothing$.
- If The Turing machine does not halt then we denote $M(x) = \perp$.

Definition. The execution of a TM, $M = (\Sigma, k, \delta)$ on input $x \in \{0,1\}^*$ is this process:

1. Initialize T as $\triangleright, x_0, x_1, \ldots, x_{n-1}, \emptyset, \emptyset, \emptyset, \ldots$ where $n = |x|$.
2. Initialize two natural number variables, $i = 0, s = 0$.
3. **repeat**
   1. $(s', \sigma', D) = \boldsymbol{\delta}(s, T[i])$
   2. $s := s', T[i] := \sigma'$
   3. if D = **R**: $i := i + 1$
      if D = **L**: $i := \max\{i - 1, 0\}$
      if D = **H**: **break**
4. If the process finishes, the output is: $M(x) = T[1], \ldots, T[m]$ where $m > 0$ is the smallest integer, $T[m+1] \notin \{0,1\}$.

   Otherwise (i.e., the machine does NOT halt), $M(x) = \perp$.

Turing's Output:

arbitrary choice has been made by an external operator. This would be the case if we were using machines to deal with axiomatic systems. In this paper I deal only with automatic machines, and will therefore often omit the prefix *a*-.

Computing machines.

If an *a*-machine prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine. If the machine is supplied with a blank tape and set in motion, starting from the correct initial *m*-configuration, the subsequence of the symbols printed by it which are of the first kind will be called the *sequence computed by the machine*. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the *number computed by the machine*.

Simpler Execution Model: No Input

Definition. The **output** of the execution of a TM, $M = (\Sigma, k, \delta)$ is the result of this process:

1. Initialize T as $\emptyset, \emptyset, \emptyset, \ldots$
2. Initialize two natural number variables, $i = 0, s = 0$.
3. **repeat**
    1. $(s', \sigma', D) = \boldsymbol{\delta}(s, T[i])$
    2. $s := s', T[i] := \sigma'$
    3. if D = **R**: $i := i + 1$
       if D = **L**: $i := \max\{i - 1, 0\}$
       if D = **H**: **break**
4. If the process finishes (the repeat breaks), the result of this process is:

$M(\cdot) = T[1], \ldots, T[m]$ where $m > 0$ is the smallest integer such that $\forall l > m. T[l] = \emptyset$.

Otherwise, $M(\cdot) = \perp$.

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

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.

R

$\frac{1}{2} = 0.5$

$\frac{1}{3} = 0.333\ldots$

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 t
it is almost equally easy
of an integral variable
predicates, and so fort
however, the same in eac
for explicit treatment as
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.

Roughly: numbers that can be output *arbitrarily precisely* by a Turing Machine (No Input model).

Computable Numbers

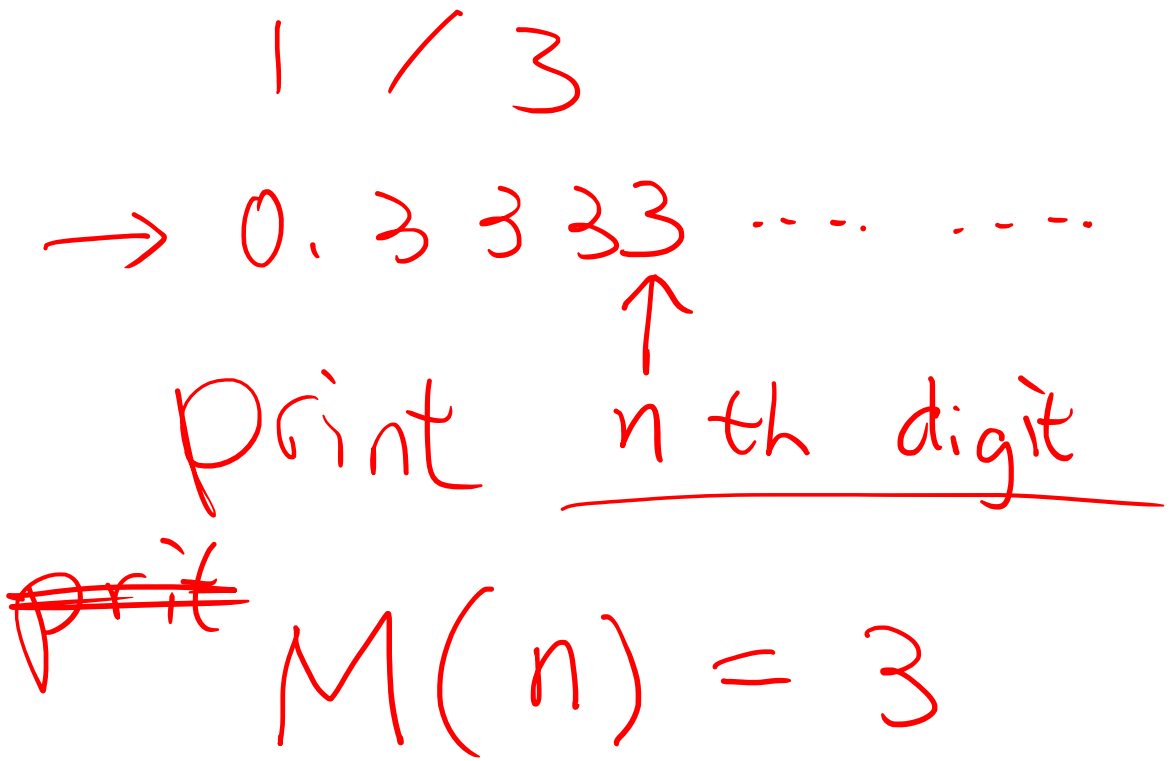
Is $\frac{1}{2}$ *computable?*

1 / 2 (to binary)

0.5 (to bin)

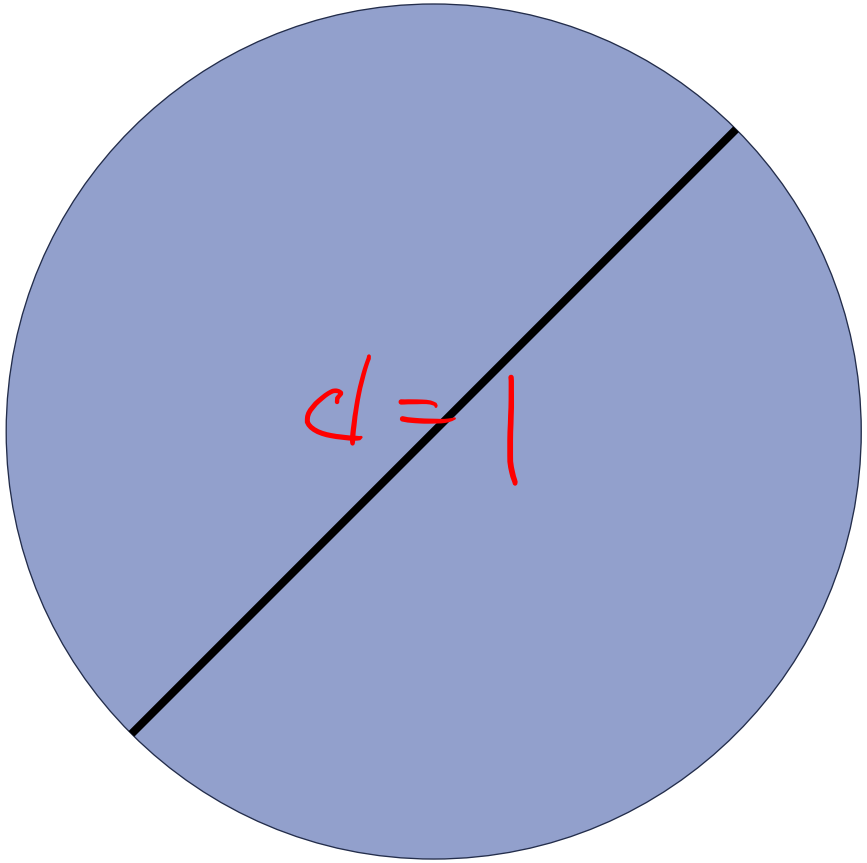
Computable Numbers

Is $\frac{1}{3}$ *computable?*



Is τ computable?

$$\pi = \frac{C}{d}$$



Gottfried Wilhelm Leibniz

$$\pi = \frac{C}{d}$$

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \cdots$$

0.01

$$- \frac{4}{401} + \frac{4}{403} \Big|$$

Gottfried Wilhelm Leibniz

1673

Digital Mechanical Calculator: +, -, *, /

Gottfried Wilhelm Leibniz

…a general method in which all truths of reason would be reduced to a kind of calculation. At the same time, this would be a sort of **universal language or script, …**

Are there any *uncomputable* numbers?