21 April 202	25	Turing 2 The	Machines Church - Turing	Thesis
	· · · · · · · · · · · · · · · · · · ·			
$\frac{1}{P \cdot lan} = \frac{1}{P \cdot lan}$	· · · · · · · · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · ·
* Defining	"Algorithms"			
* Announce	ments	· · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·
* TM Exa	mples			
· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · ·
	· · · · · · · · · · · · · · · · · · ·			
	· · · · · · · · · · · · · · · ·			
	· · · · · · · · · · · · · · · · · · ·			

So Far Ð. E CORE  $\mathcal{D}$ NP

$\sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{i$	
$A = \frac{1}{2} $	is decidable (R) if there exists D such that $x \in L \implies D$ accepts x $x \notin L \implies D$ does not accept x

Theorem.	There are		e cidable	problems.	· · · · · · · · ·
$\downarrow$ $N_{\circ}$	Program	Solves ·	the probl	en -	· · · · · · · · ·
· · · · · · · · · · · ·				· · · · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · ·	· · · · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · ·		· · · · · · · ·	· · · · · · · ·	· · · · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · ·		 	· · · · · · · ·	· · · · · · · · ·	· · · · · · · · · ·
· · · · · · · · · · · ·		 	· · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · ·
· · · · · · · · · · ·		 	· · · · · · · ·		· · · · · · · · · ·
· · · · · · · · · · · ·		 	· · · · · · · ·		· · · · · · · · ·
· · · · · · · · · · · ·					
· · · · · · · · · · · ·					

Theorem. There are undecidable problems.
Ly No program solves the problem.
But what do we mean by "program"?
* Python program? C program?
* Physical derice (transistors, abacus, quantum system)
* Physical device (transistors, abacus, quantum system) * Human cognition?
× Human cognition?
× Human cognition?
× Human cognition?

Major Endeavor of 1920s - 1940s
L, Define "Effective Computation"
· · · · · · · · · · · · · · · · · · ·
Proposed Model (From Alan Turing)
* Turing Machine
H Obvionsly programmable
Hy Powerful enough to capture all
Ly powerful enough to capture all other proposals (e.g. 2-calculus, p-recursive) the
Ly powerful enough to capture all other proposals (e.g. 2-calculus, p-recursive) fus
Es powerful enough to capture all other proposals (e.g. X-calculus, pr-recursive) fus
other proposals (e.g. 2-calculus, pr-recursive) fus
other proposals (e.g. 2-calculus, p-recursive) fus
other proposals (e.g. 2-calculus, pr-recursive) fus

Major Endeavor of 1920s - 1940s
L, Define 'Effective Computation'
· · · · · · · · · · · · · · · · · · ·
Proposed Model (From Alan Turing)
* Tuving Machine
6 Obviously programmable
Ly powerful enorgh to capture all
other proposals (e.g. 2-calculus, p-recursive) fus
Church - Turing Thesis Every physically - realizable computational model
Every physically-realizable computational model
can be simulated by a Tuving Machine.
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

· · ·	C.														te		Ję		· · · · ·	•			i i i i i i i i i i i i i i i i i i i i				· · ·			•				· · · · · · · · · ·			· · ·	· · ·	
• •		×	N N	) 0-	ť.	۰ م	•	.+	<u>h</u>	201	/ e (	<u>M</u>	•	•		۲ ۲ Σ	>∨_€	2	U	i le	ح	•	<u>م</u>	•	F	<u>v</u>			) )	. \	ar			£	. ∨	na	tu	ve	• •
• •	•	•	•	· ·	 <del>.</del> . 	M 10	vy	el	√ ¤ ∕_	ה ה	us C		ر د	tr p	es te	1 1	۰ ۲	•	•	•	•	•	• •		•	•	· · ·	•	•	•	· · ·	•	•		•		• •	•	•
• •	•	•	•	•	· ·	•	•	•	• •	•	•	•	•	•	•	• •	•	•	•	•	•	•	• •	•	•	•	· ·	•	•	•	••••	•	•	•	•	• •	• •	•	•
• •	•	•	•	•	• •	•	•	•	• •	•	•	•	•	•	• •	• •	•	•	•	•	•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	•	•		•	•
• •	•	•	•	•	• •		•	•	• •	•	•	•	•	•	• •	• •	•	•	•	•	•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	•	•		•	•
• •	•	•	•	•	• •	•	•	•	••••	•	•	•	•	•	• •	• •	•	•	•	•	•	•	• •	•	•	•	· ·	•	•	•	• •	•	•	•	•	•	• •	•	•
																																						•	
• •	0	•	•	٠	• •	•	٠	0	• •	٠	٠	٠		٠	• •		٠	٠	٠	٠	•	•	• •	٠		•	• •	*	٠	•	• •		•	•	•	• •	• •	•	
• •	•	•	•					•		•	•	•	•	•	• •	• •	•	•	•	•	•	•	•••	•	•	•	••••	•	•	•	• •	•	•	•	•	• •		•	
																																						•	

Church - Turing Thesis
Everything compitable, computable by a TM.
* Not a theorem. More like a physical law of nature.
-very robust thesis - widely accepted
Extended Church - Turing Thesis
Everything efficiently - computable is
efficiently-computable by a TM.
- Move controversial e.g. Quantum Computers
L> But still very vobust.
(e.g. efficient simulation of ) python by TMS )

·       ·	Announcements	•
·       ·	* Nove	•
·       ·		•
·       ·		•
	· · · · · · · · · · · · · · · · · · ·	•
·       ·	· · · · · · · · · · · · · · · · · · ·	•
·       ·		•
·       ·		•
·       ·		•
·       ·	· · · · · · · · · · · · · · · · · · ·	•
		٠
		•
		•

* Simple (finite) logical controller * Infinite Read/Write memory tape HELLO-WORLD Finite Automaton 7 * Finite controller Reads cell under tape "head" * Then, - Writes new symbol to cell - Moves tape head Left or Right	Turing Machines
HELLO-WORLD Finite Automaton 7 × Finite controller Reads cell under tope "head" * Then, - Writes new symbol to cell - Moves tope head Left or Right	* Simple (finite) logical controller
HELLO-WORLD Finite Automator 7 × Finite controller Reads cell under tape "head" * Then, - Writes new symbol to cell - Moves tape tread Left or Right	* Infinite Read ( Write memory tape
HELLO-WORLD Finite Automator 7 × Finite controller Reads cell under tape "head" * Then, - Writes new symbol to cell - Moves tape tread Left or Right	
Automaton 7 × Finite controller Reads cell under tape "head" × Then, - Writes new symbol to cell - Moves tape tread Left or Right	
Automaton 7 × Finite controller Reads cell under tape "head" × Then, - Writes new symbol to cell - Moves tape tread Left or Right	
- Writes new symbol to cell - Moves tape tread Left or Right	
	- Writes new symbol to cell
- <u>Updates</u> internal (finite) state	- Moves tape tread Lett or Right - Updates internal (finite) state

Warmup: Deterministic Finite Antometa D (Stavt) Accept . . . . . Reject

Warmup: Deterministic Finite Antomata Reject . .  $\mathcal{I}(D) = \{ O': n \in \mathbb{N} \}$ 

Warmup: Deterministic Finite Automata
$\left[ \begin{array}{c} D \end{array} \right] = \left( \begin{array}{c} \\ Stavt \end{array} \right)  \left[ \begin{array}{c} \\ Accept \end{array} \right]$
Reject
DFA defined by $Z(D) = \{O': n \in \mathbb{N}\}$
X = Input Alphabet X = Finite set of internal states
$x q \in Q \equiv start state$
× Q E Q = [Accept] State
XVEQ = Reject State
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

Warmup: Deterministic Finite Antomata
$\left[ \begin{array}{c} D \end{array} \right] = \left[ \begin{array}{c} (stavt) \\ (stavt) \\ \end{array} \right] = \left[ \begin{array}{c} (stavt) \\ (stavt) \\ \end{array} \right] = \left[ \begin{array}{c} (stavt) \\ (stavt) \\ \end{array} \right]$
$(\underline{A}, \underline{A}, A$
$\mathbb{R}eject$
DFA defined by Z(D) = ZO": neN ]
* <u>Ser Enput Alphabet</u>
* Q = Finite set of internal states
$x q \in Q = start state$
X Q E Q = [Accept] State X V E Q = [Reject] State
* I = transition rule - new state
$\int \left( \left( \begin{array}{c} q \\ q \end{array} \right)^{2} \right)^{2}  \left( \begin{array}{c} q \\ q \end{array} \right)^{2}  \left( \begin{array}{c} q \end{array} \right)^{2}  \left( \begin{array}{c} q \\$
current state / Symbol read

Turing Machines ()-. (• Start Accept Reject

Turing Machines	· · · · · · · ·
TM defined by $\star \Sigma \equiv Input Alphabet 7 \Sigma \subseteq \Gamma$ $\Gamma \equiv Tape Alphabet 1 G \Gamma \equiv blant \Gamma \in \Gamma \equiv Left$	K
	of tape

Turing Machines  $\bigcirc$ · · · <del>·</del> · · · · · · · · · · · TM defined by  $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j$ \* II = Input Alphabet E F Elank I E Tape Alphebet the ET = Left end of tape Q = Finite set of internal states including [start] [Accept] [Reject]

Turing Machines
TM defined by 5 C P
and X and En a En a Lenputa Alphabet and ( and a contract and a second and a second and a second and a second a
$\Gamma \equiv Tape Alphe bet$ $F \in \Gamma \equiv Left end$ of tape
* Q = Finite set of internal states
including [start] [Accept] [Reject]
* S = transition rule
· · · · · · · · · · · · · · · · · · ·
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
Current state read
Current state read

Turing Machines . . . <del>.</del> . <del>. . .</del> . . . . TM defined by  $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i$ \* Z = E Input Alphabet E F Elank E Tape Alphebet  $E \in T = Left end$ of tape = Finite set of internal states including [start] [Accept] [Reject] transition rule - new state  $= \int_{-\infty}^{\infty} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right) + \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right) + \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right) + \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right) + \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \left( \frac{1}{2} \right) - \frac{1}{2} \left( \frac{1}{2} \right) \right) \right) \right) \right)$ more director E JLIRG Current state Vead Symbol to be written

 $= 70, 1, \dots, 100$ Example TMs  $\sum_{i=1}^{n} = \sum_{i=1}^{n} O_{i} \setminus \sum_{i=1}^{n} O_{i}$ 11, 1, R O, O, R $(start) \xrightarrow{\vdash, \vdash, R} (1, \downarrow, R) (1, \downarrow$  $\pm,\pm,$  R $\left(\begin{array}{c} 0\\ 0\end{array}\right)^{n} \left(\begin{array}{c} 0\\ 0\\ 0\end{array}\right)^{n} \left(\begin{array}{c} 0$ 0,0,0, unspecified transitions  $\mathcal{J}_{\mathrm{res}}^{\mathrm{res}}\left( \left[ \mathcal{M}_{\mathrm{res}}^{\mathrm{res}} \right] \right)_{\mathrm{res}}^{\mathrm{res}} = \left[ \left[ \mathcal{M}_{\mathrm{res}}^{\mathrm{res}} \right] \right]_{\mathrm{res}}^{\mathrm{res}}$ Accept Reject

 $= \frac{1}{2} O_{1} O_{1}$ Example TMs = 201111, 14, R O, O, R $(stouvt) \xrightarrow{\vdash_{1}\vdash_{1}R} (stouvt) \xrightarrow{\vdash_{1}}R$  $\perp$ ,  $\stackrel{i}{\leftarrow}$ ,  $\stackrel{i}{R}$ unspecified 1 transitions  $\mathcal{L}(\mathcal{M}) = \{0, 1\}$ Accept Reject

Translating TMs to high-level programs O, O, R $(starvt) \xrightarrow{\vdash, \vdash, R} (starvt) \xrightarrow{\vdash, \vdash, R} (starvt) \xrightarrow{\bigcirc, \vdash, R} (starvt) \xrightarrow{\frown, \vdash, R} (starvt) \xrightarrow{\bullet, \vdash, L} (sta$ Accept

Translating TMs to high-level programs "Move tape head right until a l'is read" 0,0,**R**  $(stouvt) \xrightarrow{\vdash_{1}\vdash_{1}R} (stouvt) \xrightarrow{(l_{1}\vdash_{1})R} (stouvt) \xrightarrow{(l_{1}\vdash$ (1, 1), (1, 1), RAccept

Translating TMs to high-level programs "Move tape head right until a l'is read " O, O, R I, I, R  $(stowet) \xrightarrow{I_1, I_2, R} (stowet) \xrightarrow{I_1, I_2, R} (stowet)$ Accept

Translating TMs to high-level programs "Move tape head right until a l'is read"  $(start) \xrightarrow{F_1}, \xrightarrow{R_1}, \underbrace{O_1, \dots, R_r}_{r_1}, \underbrace{O_1, \dots, R_r}_{r_2}, \underbrace{I_1, I_1, R_r}_{r_2}, \underbrace{I_1, I$ (1, 1), (1, 1), (R, 1)O, O, L Keplace rightmost 1 with blank Accept

TM "Progr	$\sim \mathcal{M}$
* Repeat	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	$M_{ove}$ $R$
· · · · · · · · · · · ·	if - Read O: Reject
· · · · · · · · · · · ·	if - Read O : Reject Write blank, More R
	While Read ( : Move R ] move to end of On 1 m While Read ( : Move R ]
· · · · · · · · · · · ·	I if Read blank: Reject Move L, Write blank, Move L
· · · · · · · · · · · · · · · · · · ·	- if Read blank: (Accept) J Equal # of O and I evased
.     .     .     .     .     .     .     .       .     .     .     .     .     .     .     .     .       .     .     .     .     .     .     .     .     .       .     .     .     .     .     .     .     .     .       .     .     .     .     .     .     .     .     .       .     .     .     .     .     .     .     .     .	While Read O: Move L

Rules of Thumb For TM Code
X Variables
- Finite number of variables - Taking finite number of values
× Conditionals - Finite number of nested conditionals
× Function calls ( subvontines
- Functions must be computable - Finite depth call stack.
(otherwise, requires argument that state can be maintained by finite state machine using type.)
• • • • • • • • • • • • • • • • • • •