23 April 2025	Similatie Algorithms
· · · · · · · · · · · · · · · · · · ·	w/ Tuving Machines
· · · · · · · · · · · · · · · · · · ·	
Plan - Plan	· · · · · · · · · · · · · · · · · · ·
$\frac{1}{2} + \frac{1}{2} + \frac{1}$	· · · · · · · · · · · · · · · · · · ·
X ANNOUNCEMENTS	· · · · · · · · · · · · · · · · · · ·
* Multi-lape IMIS	· · · · · · · · · · · · · · · · · · ·
$\frac{1}{2} + \frac{1}{2} + \frac{1}$	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

Turing Machines
TM = defined by $T = Input Alphabet$ $T = Tape Alphabet$ $F = Left end$ $F = Left end$
* Q = Finite set of internal states including [start] [Accept] [Reject]
* S = transition rule
$\int \left(\left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \end{array} \right) \left(\begin{array}{c} \cdot \\ \end{array} \right) \left(\begin{array}{c} \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \end{array} \right) \left(\begin{array}{c} \cdot \end{array} \right) \left(\begin{array}{c} \cdot \end{array} \right) \left(\begin{array}{c} \cdot \\ \end{array} \right) \left(\begin{array}{c} \cdot \end{array} \right) \left($
· · · · · · · · · · · · · · · · · · ·

Turing Machines 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} \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} \right) \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \right) \right) + \frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \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} \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} \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} \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} \left(\frac{1}{2} \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} \left(\frac{1}{2} \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} \left(\frac{1}{2} \left(\frac{1}{2} \right) - \frac{1}{2} \left(\frac{$ more director E JLIRG Current state Vead Symbol to be written

Example TMs	$\sum_{i=1}^{n} = \sum_{i=1}^{n} O_{1} \setminus \sum_{i=1}^{n} O_{1}$	$\int_{-\infty}^{\infty} = \int_{-\infty}^{\infty} O_{\chi}$	
· · · · · · · · · · · · · · · ·	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · ·
(start) <u>F, F, R</u>	$ \begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \\ \end{array} \end{array} \end{array} \end{array} \\ \begin{array}{c} \\ \end{array} \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} $	$\begin{array}{c} R \\ {\longrightarrow} \\ {\longrightarrow} \\ \end{array} \begin{pmatrix} R \\ \mathsf$	
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			
	$ \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0 \end{array} \right) \left(\begin{array}{c} 0 \\ 0 \end{array} \right) \left(\begin{array}{c} 0$		· · · · · · · · · · · · · ·
			. .
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$			unspecified transitions
$\mathcal{A}_{\mathcal{A}} = \mathcal{A}_{\mathcal{A}} \left(\mathcal{M}_{\mathcal{A}} \right)_{\mathcal{A}} = $	· · · · · · · · · · · · · · · · · · ·	Accept	Reject
$+(\chi_0\chi_1\chi_2 \Lambda_1$		<u> </u>	

 $= \frac{1}{2} O_{1} O_{1}$ Example TMs $\sum_{i=1}^{n} = \sum_{i=1}^{n} O_{i} \setminus \sum_{i=1}^{n} O_{i}$ 11, 14, R O, O, R $(stouvt) \xrightarrow{\vdash_{1}\vdash_{1}R} (stouvt) \xrightarrow{\vdash_{1}}R$ β_{1}, β_{2}, R unspecified 1 transitions $\mathcal{I}(\mathcal{M}) = \{0, 1^n : n \in \mathbb{N}\}$ Accept Reject

Translating TMs to high-level programs O, O, R $(starvt) \xrightarrow{\vdash, \vdash, R} (starvt) \xrightarrow{\vdash, \vdash, R} (starvt) \xrightarrow{\cup, \perp, L} (starvt) \xrightarrow{\cup, L} (starvt) \xrightarrow{\cup, L} (starvt) \xrightarrow{\cup, L} (starvt) \xrightarrow{\cup,$ Accept

Translating TMs to high-level programs "Move tape head right until a l'is read" (0,0), R $(stouvt) \xrightarrow{\vdash_{1}\vdash_{r}R} (stouvt) \xrightarrow{(l_{1}\vdash_{r}R)} (stouvt) \xrightarrow{(l_{1}\vdash$ (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 $(stowet) \xrightarrow{I_1, I_2, R} (stowet) \xrightarrow{I_1, I_2, R} (stowet)$ (1, 1), (1, 1), RAccept

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	n, and a second se
* Repeat	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	M_{ove} R
· · · · · · · · · · · ·	if - Read O: Reject
· · · · · · · · · · · ·	Write blank, More R
	While Read (" Move R J move to end of On 1 m While Read (" Move R J
	I if Read blank: Reject Move L, Write blank, Move L
	- if Read blank: [Accept] J Equal # of O and I evased
· · · · · · · · · · · ·	While Read (): Move L While Read (): Move L

Church - Turing	Tresis		•
Every thing	Compta Se 1	Compute by a INI	•
Extended Ch	wch - Turity	. Thesis	
Every thing	efficiently_	-computable is	•
efficientl	y-computable	they are the second sec	
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	 	•
· · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
· · · · · · · · · · · · · · ·	· · · · · · · · · · ·	. .	

$A_{\rm MM}$	on venue		· · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · ·
· · · · · · · ·	Kieval	Lecture	Tonight 4:	30° e° e° e° e°	M_{\sim} lott 228
· · · · · ·	· · · · · ·	Mike	Sipser on	Pvs. NP	· · · · · · · · · · · · · · · · · ·
· · · · · ·	HW8	Released	· · · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · ·				· · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · ·	
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · ·
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · ·
· · · · · ·	· · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · ·

Similating	Algorithmic Primities on TMs
* Copy &	Paşte
<u>Start</u>	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
· · · · · · · · · · · · · · · · · · ·	= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +
$\sum_{n=1}^{n} \sum_{n=1}^{n} \sum_{n$	$\frac{\left \frac{1}{2} \right ^{2}}{\left \frac{1}{2} \right ^{2}} = \frac{\left \frac{1}{2} \right ^{2}}{\left \frac{1}{2} \right ^{2}} = \frac{1}{2} \left \frac{1}{2} \right ^{2}$
· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · ·	. .
· · · · · · · · · · · · ·	
· · · · · · · · · · · · · ·	
· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

Similating	Algorithmic Primitives on TMS
* Copy 2	Paste
Stort	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
End	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
· · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	$\sigma_{r}, \sigma_{r}, R$
· · · · · · · · · · · · · · · · · · ·	O, L, R, Mave 2 End L, O, L Write O Back Back
· · · · · · · · · · · · ·	Read
· · · · · · · · · · · · ·	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
	· · · · · · · · · · · · · · · · · · ·

Similating Algorithmic Primitives on TMs * Copy & Paste Start Xo | H Xo | X1 | X2 ····· Xn # $X_1 \mid X_2 \mid \dots \mid X_n \mid \ddagger$ $X_1 X_2$ Xo \vdash **太**。 • () • , • () • , • R. r, r, R. σ, σ σ., Move Back write 1 (Move 2) End I, L, R O, L End End Write O (Move 2) End Write O (Move 2) End Write O (Move 2) Write O (Move 2) Write O (Move 2) - (Read)

Similating Algorithmic Primitives on TMs * Copy & Paste Start Xo K X1 X2 Xn # .- .- . . . Xol X₁ X₂ X1 X2 Xn # Xo $\times_{\!\!\!\!n}$ \vdash • () • , • () • , • R. σ, σ, τ σ., Move Back write 1 (1, L End 1, L, R End 1, L, R End 1, L, R Nove 2 End Write 0 Nove 2 End Vill, R Nove 2 End Vill, R Read Ð.

Similating	Algorithmic	Primitires a	on TMs	· · · · · · · · · ·
* Copy 2	Paste	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·
Stort-	$ \begin{array}{c c} & & \\ \hline \\ \\ \hline \\$	X _n #		
End	$ \left \begin{array}{c} \downarrow \\ \downarrow $	$ X_n$ \mp X_o	$X_1 X_2 $	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
· · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·
· · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · ·		
if xi=1 move to right	t wit: 1		It Xi = 0 Move to ri	ght until
and return	$\tau \sim \chi_{1+1}$	Read	and return	\sim to χ_{i+1}
· · · · · · · · · · · · · · ·	+, =	#, L (E	nd)	

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 / subrontines
- Functions must be computable - Finite depth call stack.
(otherwise, requires argument that state can be maintained by finite state machine using type.)

Multi-Tape TM * Finite State Controller * Constant Number of Infinite Read/Write Tapes * Transition Rule $S\left(\begin{array}{ccc} q & \chi \\ \end{array}\right) \longrightarrow q', (\chi' - - \chi'_{h}), (M_{h} - - M_{u})$ New symbols moves for each written tape head symbols read $i_{\mathcal{L}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{i=1}^{n} \sum_{i=$

Simulatives a	Multi - Tap	e TM e ~	Sizgle-Tape TM
A Idea Enco	ste nur ltip	ole tape entries	from as
· · · · · · · · · · · · · · · ·	sing	le tape entry	from T
· · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · · <u>· · · · · · · · · · · · · </u>	0 0		······································
	100		
	100		· · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	001 100 000 11		· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · ·			ben ?
· · · · · · · · · · · · · · ·			· · · · · · · · · · · · · · · · · · ·

Simulativo a	Multi-Tape TM on a Size-	Tape TM
* Idea Encod	Le multiple tape entries from	$ \begin{bmatrix} \uparrow & \downarrow & $
· · · · · · · · · · · · · · · · · · ·	single tape entry from	$\begin{pmatrix} & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & & \\ & & $
· · · · · · · · · · · · · · · · ·	$ \begin{array}{c} \begin{array}{c} \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $	include
· · · · · · · · · · · · · · · · · · ·		"marked"
· · · · · · · · · · · · · · · · · · ·		
· · · · · · · · · · · · · · ·		
· · · · · · · · · · · · · · · · · ·		
· · · · · · · · · · · · · · · · ·		· · · · · · · · · · ·
	100 1 100 11 100 100 100 100	· · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · ·		· · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·

· · · · · · · · · · ·	30 Coj 100		· · · · · · · · · · · · · · · · · · ·
Simulation - scan	from left to ri	of until every top	pe head seen via X's
- Calen - scan	late next state, from right to	vrites, and move left to make upd	s for each tape
· · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · ·	· · · · · · · · · · · · ·	. .	· · · · · · · · · · · · · · · · · · ·

· · · · · · · · · ·
· · · · · · · · · ·	· ·

	• •
Simulation of single step of mult: -tape TM - scan from left to right until every tape head seen via X's - Calculate next state, writes, and moves for each tape - scan from right to left to make updates	· · · · · · · · · · · · · · · · · · ·
Note quadratic overhead in time complexity? For each step of multi-tape machine	· · ·
La scan Le to R. (ending when tape heads are discovered). How long can this take?	· · ·
t t t t t t t t t t t t t t t t t t t	· · · · · · · · · · · · · · · · · · ·
$\tau(n) \leq feps$	· ·

Simulation of single step of mult:-tape TM - scan from left to right until every tape head seen via X's - Calculate next state, writes, and moves for each tape - scan from right to left to make updates
Note quadratic overhead in time complexity? For each step of multi-tape machine La scan L to R (ending when tape heads are discovered)
Fact All existing (classical) programming languages can be simulated on a TM w/ polynomial f Extended Church-Turing Thesis

Self-Refu	jence	· · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · ·
Key (o computability	reductions	· · · · · · · ·	· · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	input (P)	· · · · · · · · · · ·	· · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	simulate running	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	· · · · · · · ·	· · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · ·
$\begin{array}{cccccccccccccccccccccccccccccccccccc$				· · · · · · · · · ·
$\cdot \cdot $	Simu la jas al	inner Cova p		
· · · · · · · · · · · · · · · · · · ·				. .
				. .
				. .
$\begin{array}{cccccccccccccccccccccccccccccccccccc$				$\begin{array}{cccccccccccccccccccccccccccccccccccc$

Universal	Turing Machine	•
Theorem	There exists a Turine Machine U that	•
· · · · · · · · · · · ·	on input (Q, x) simulates the behavi	⊃√
· · · · · · · · · · ·	of vunning. Q on x.	•
· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•
		٠
· · · · · · · · · · · ·		•
		•
· · · · · · · · · · ·		•
· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	•

Universal Turing Machine
Theorem: There exists a Turing Machine U that
on input (Q, x) simulates the behavior
of running Q on x
Lo Step-by-step simulation, not just 1/0. Simulation
· · · · · · · · · · · · · · · · · · ·
e.g. allows for pseudocode of the form
"Run Q on x for t steps"
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

Intuition,	for construction of U.
× 3-terpe	$\mathcal{T}\mathcal{M}$
· · · · · · · · · · · · · · · · · · ·	Holds description of Q
· · · · · · · · · · · · · · · · · · ·	$\left \begin{array}{cccccccccccccccccccccccccccccccccccc$
· · · · · · · · · · · · · · · · · · ·	initialized to x
	Simulates Tape of K.
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	Records state/tape head of Q at time to
	Records State / tape head of Q at time to
	Records state / tape head of Q at time to
	Records State / tape head of Q at time t

Intuition for construction of U.
× 3-tape TM
T, : Holds description of Q
12° 1 initialized to X
Simulates tape of Q
T3 Records State/tape head of Q at time to
78 simulate one step of Q
- Look up state and tape head position on T3
- Access appropriate cell of T2 and character
- Update T2 and T3