

Draw the machine schema for a TM which when started on a blank tape (input ε) halts with abbb on the tape in our standard input format.

Announcements

Final exam tutorial: Sunday Dec. 4, 12:00pm, Room TBA.

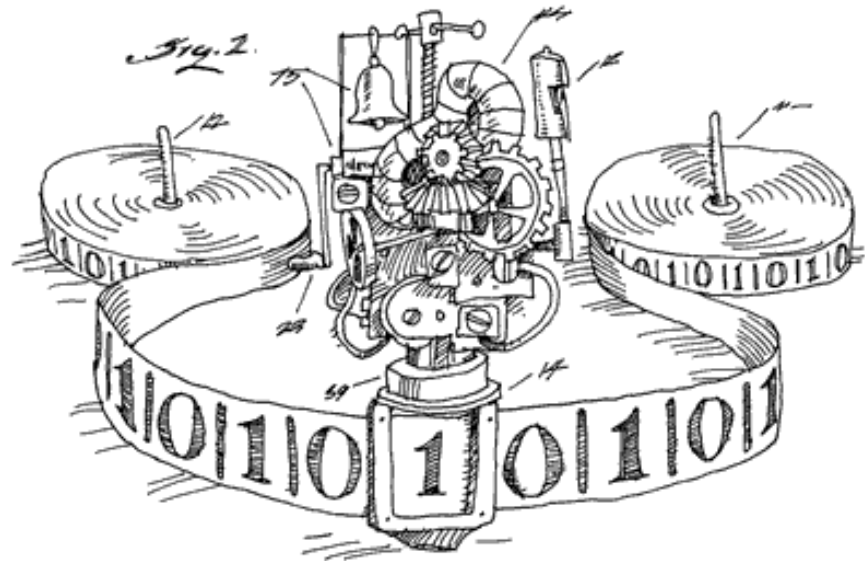
Our final exam is Tues. Dec. 6, 2pm.

Assignment #4 has been posted: due Friday Nov. 18.

Tutorials this week: bring any questions you have about the assignment, class material (the pumping theorem?) or old final exams.

Extensions of TM's/UTM's

It can be proven that adding extra power to a TM by adding multiple tracks, tapes, or tape heads does not change what it is able to compute. These more powerful models can be simulated on our single tape/one head machine.

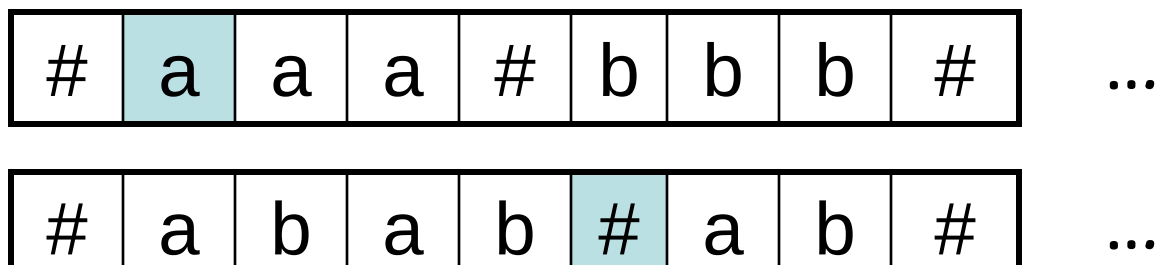


Turing Machine by Tom Dunne
American Scientist, March-April
2002

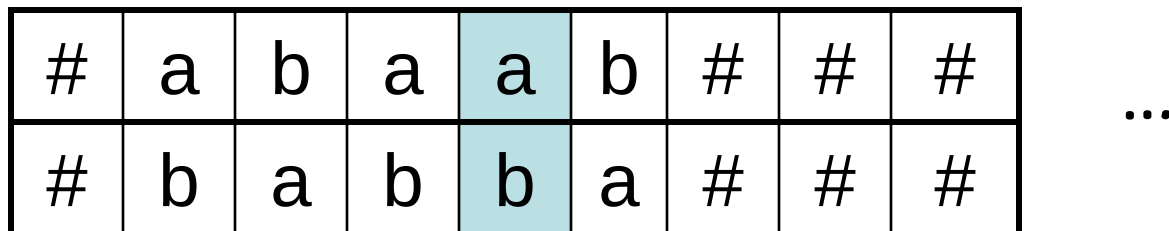
Two-way infinite tape:



Multiple tapes:



Multiple tracks:



Two tracks- Each tape square has one symbol on upper track and one symbol on lower track:

#	a	b	b	#	#	#	#	#
#	#	#	#	#	#	#	#	#

...

To simulate this on a standard TM:

On standard TM initially:

#	a	b	b	#	#	#	#	#
---	---	---	---	---	---	---	---	---

...

Initial alphabet is $s_1, s_2, \dots, s_k$.

New symbols = $\left\{ \begin{pmatrix} s_i \\ s_j \end{pmatrix} : i=1, 2, \dots, k, j=1, 2, \dots, k \right\}$

#	a	b	b	#	#	#	#	#	...
---	---	---	---	---	---	---	---	---	-----

Reformat initial tape:

$\begin{pmatrix} \# \\ \# \end{pmatrix}$	$\begin{pmatrix} a \\ \# \end{pmatrix}$	$\begin{pmatrix} b \\ \# \end{pmatrix}$	$\begin{pmatrix} b \\ \# \end{pmatrix}$	$\begin{pmatrix} \# \\ \# \end{pmatrix}$	#	#	#	#
--	---	---	---	--	---	---	---	---

Each time TM moves onto a # square,
reformat it to be: $\begin{pmatrix} \# \\ \# \end{pmatrix}$

For each state q add a transition:

$$\delta(q, \#) \rightarrow (q, \begin{pmatrix} \# \\ \# \end{pmatrix})$$

Multiple tape heads:

#	a	a	b	b	a	a	#
---	---	---	---	---	---	---	---

Keep track of tape head positions
on extra tracks:

#	a	b	b	a	a	#			
0	0	0	0	0	1	0	#	#	#
0	1	0	0	0	0	0			

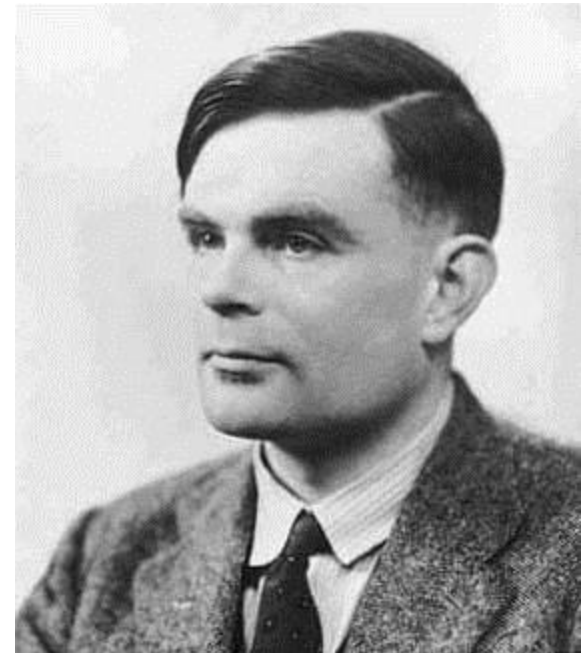
Multiple tapes:

\$	#	b	b	a	b	b	#	#	#	#
\$	0	1	0	0	0	0	0			
\$	#	b	b	a	b	b	#			
\$	0	0	0	1	0	0	0			

Universal Turing Machines

A Universal TM (UTM) is like my java TM simulator but written in TM. We can argue that a 3-tape TM can be used to create a UTM which can execute instructions from an arbitrary TM program.

The existence of a UTM is used to acquire some problems which can be proven to not be Turing-decidable.



Alan Turing (1912-1954)



Universal TM's:

the birth of the idea of having programmable computers.

Software can be used instead of designing new hardware.

"Universal Turing Machine" Jin Wicked.

Problem: A UTM is a Turing machine and hence it must have a fixed finite alphabet.

But it must be able to simulate any TM with an arbitrary alphabet.

How can we do this?

Hint: Our computers which we use to run Java and C programs have an underlying alphabet of 0/1. But they still can represent lots of symbols!

First number the states and symbols:

State	Sym	Next state	Head
s	#	s	L
s	a	t	a
s	b	t	L
t	#	h	#
t	a	t	R
t	b	t	b

Num	State	Head
0	h	L
1	s	R
2	t	#
3		a
4		b

Num	State	“State”	Head	“Head”
0	h	q00	L	a000
1	s	q01	R	a001
2	t	q10	#	a010
3			a	a011
4			b	a100

Assumptions: h is always state 0.

The start state is state 1.

The symbols always have $L = 0$, $R = 1$, $\# = 2$.¹³

State	Sym	Next state	Head
s	#	s	L
s	a	t	a
s	b	t	L
t	#	h	#
t	a	t	R
t	b	t	b

0	h	q00	L	a000
1	s	q01	R	a001
2	t	q10	#	a010
3			a	a011
4			b	a100

"M" is a string representing a TM M which uses the alphabet { (,), q, a, 0, 1, , }

"w" is a string representing w which uses the alphabet { a, 0, 1 }.

State	Sym	Next state	Head
s	#	s	L
s	a	t	a
s	b	t	L
t	#	h	#
t	a	t	R
t	b	t	b

0	h	q00	L	a000
1	s	q01	R	a001
2	t	q10	#	a010
3			a	a011
4			b	a100

"M"=

(q01, a010, q01, a000), (q01, a011, q10, a011), (q01, a100, q10, a000),
(q10, a010, q00, a010), (q10, a011, q10, a001), (q10, a100, q10, a100)

"abaa"= a011a100a011a011

"#ab#"= a010a011a100a010

Initially:

Tape 1: # "M" [#]

Tape 2: #q00... 01[#]

Tape 3: # "Tape contents for M" #

#ab[#] $\rightarrow$ #a010a011a100[a]010#

Original TM M:

Head moves left/right: move head on third tape left/right until reaching "a" (or #).

Blanks to right of input: reformat to "#" = a0...10.

Hit blank to left of input: original TM hangs.

To do one move:

Search for the current symbol (head on tape 3 is on "a" of its encoding) and the current state (from tape 2) in "M" on tape 1.

When the applicable transition is found:

1. update the current state name on tape 2,
2. move the head on tape 3 (head instruction is $a0\dots0 = L$ or $a00\dots01 = R$) or replace the current symbol encoding on tape 3.

Sample final exam question:

For each of the following languages, indicate the most restrictive of the classes below into which it falls

- (a) finite
- (b) regular
- (c) context-free
- (d) Turing-decidable
- (e) Turing-acceptable
- (f) None of the above.

1. Φ
2. $(a \cup b)^*$
3. $\{a^n b^n : n \geq 0\}$
4. $\{w \in \{\#,(,),0,1,a,q,,\}^* : w = "M" \text{ for some TM } M\}$
5. $H = \{ "M" "w" : M \text{ halts when run on input } w \}$
6. $\{ "M" "w" : M \text{ does not halt on input } w \}$