Section: Other Models of Turing Machines

Definition: Two automata are equivalent if they accept the same language.

Turing Machines with Stay Option

Modify δ, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$

Theorem Class of standard TM's is equivalent to class of TM's with stay option.

Proof:

- **($\Rightarrow$): Given a standard TM M, then there exists a TM M' with stay option such that L(M)=L(M').**

easy

- **(⇐): Given a TM M with stay option, construct a standard TM M' such that L(M)=L(M').**

  M=(Q,$\Sigma, \Gamma, \delta, q_0$,**B,F)**

  M'=$(Q', \Sigma, \Gamma, \delta'', q_0', B, F')$

  For each transition in M with a move (L or R) put the transition in M'. So, for

$$\delta(q_i, a) = (q_j, b, \textbf{L or R})$$

  put into δ'

$$\delta'(q_i', a) = (q_j', b, L \text{ or } R)$$

  For each transition in M with S (stay-option), move right and move left. So for

$$\delta(q_i, a) = (q_j, b, \mathbf{S})$$

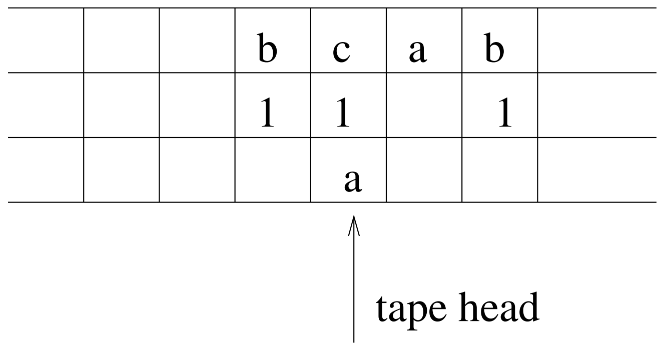
add a new state
add to δ'

$$\delta'(q_i', a) = (q_{js}', b, R)$$
$$\delta(q_{js}', c) = (q_j', c, L) \quad \forall c \in \Gamma$$

  L(M)=L(M'). QED.

Definition: A *multiple track* TM divides each cell of the tape into k cells, for some constant k.

A 3-track TM:



A multiple track TM starts with the input on the first track, all other tracks are blank.

3-track

δ: $Q \times \Gamma \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \Gamma \times \Gamma \times \{L, R\}$

Theorem **Class of standard TM's is equivalent to class of TM's with multiple tracks.**

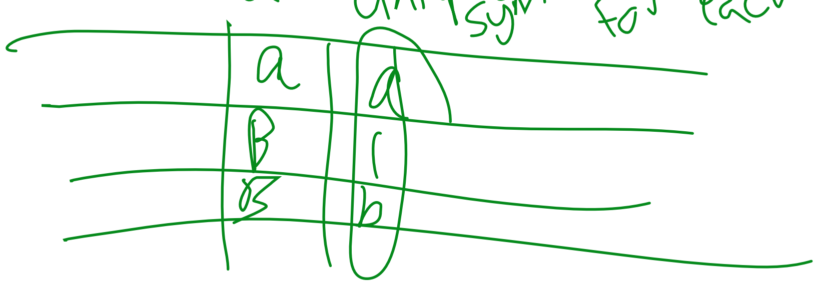
Proof: (sketch)

- **(⇒): Given standard TM M there exists a TM M' with multiple tracks such that L(M)=L(M').**

  use first track

- **(⇐): Given a TM M with multiple tracks there exists a standard TM M' such that L(M)=L(M').**

a unique symbol for each triple



finite tape alphabet ⇒ finite unique symbols

Encode each comb. of 3 symbols w/ unique symbol

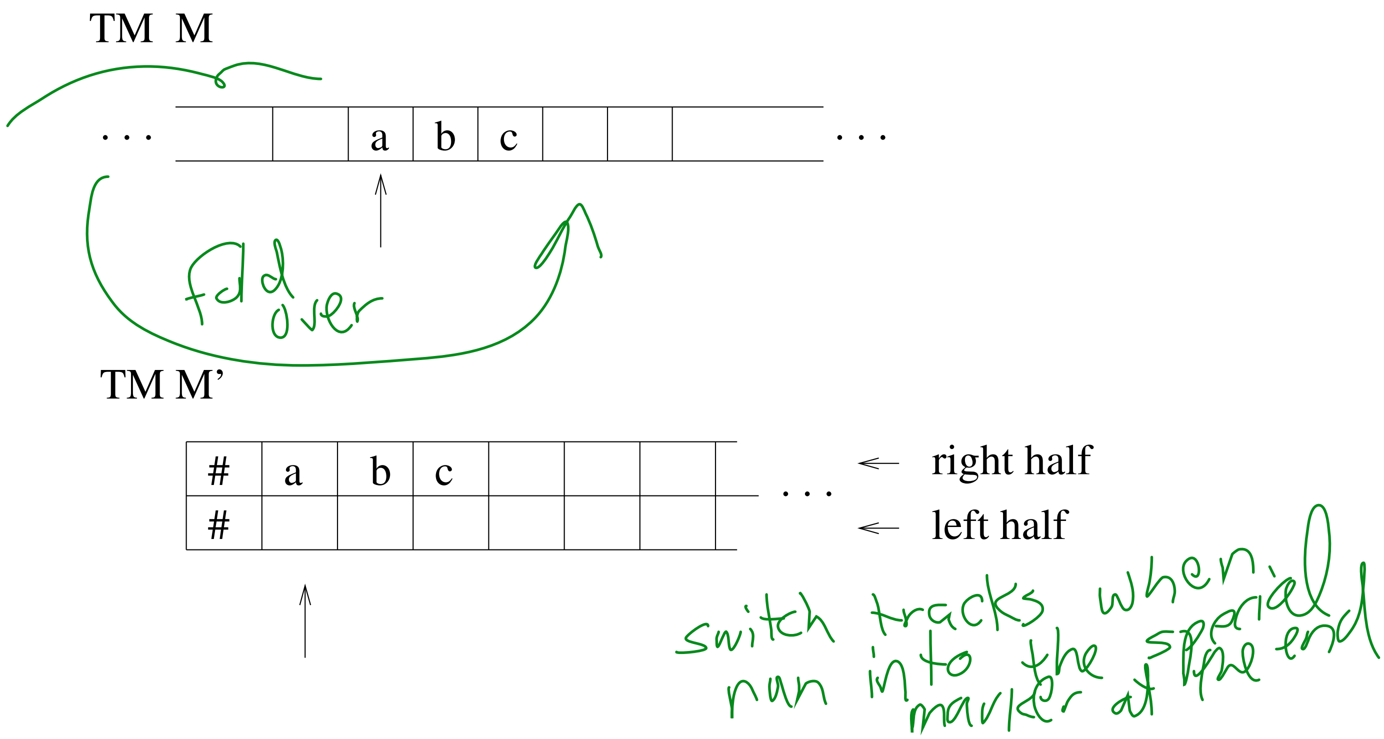
Definition: A TM with a semi-infinite tape is a standard TM with a left boundary.

Theorem Class of standard TM's is equivalent to class of TM's with semi-infinite tapes.

Proof: (sketch)

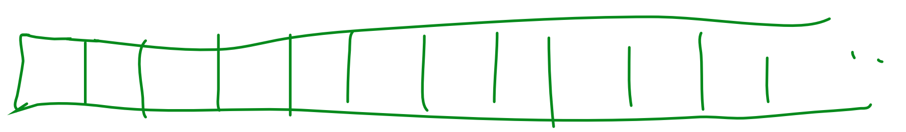
- **($\Rightarrow$): Given standard TM M there exists a TM M' with semi-infinite tape such that L(M)=L(M').**

  Given M, construct a 2-track semi-infinite TM M'



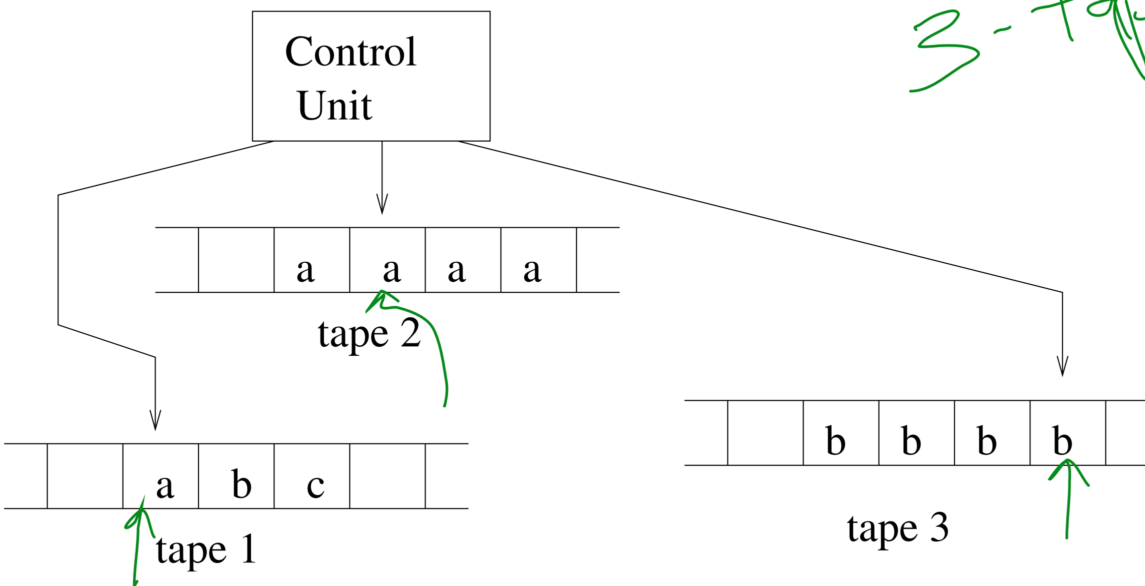
- **(⇐): Given a TM M with semi-infinite tape there exists a standard TM M' such that L(M)=L(M').**

should just work



Definition: An Multitape Turing Machine is a standard TM with multiple (a finite number) read/write tapes.

3-tape



3-tape

For an n-tape TM, define δ:

$\delta : Q \times \Gamma \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \Gamma \times \Gamma \times \{L,R\} \times \{L,R\} \times \{L,R\}$

Theorem **Class of Multitape TM's is equivalent to class of standard TM's.**

Proof: (sketch)

- **($\Leftarrow$): Given standard TM M, construct a multitape TM M' such that L(M)=L(M').**

easy just run,
just ignore other tapes

- **($\Rightarrow$): Given n-tape TM M construct a standard TM M' such that L(M)=L(M').**

Create 2n track TM

3-tapes TM
$\Rightarrow$ 6-track TM

marker

			#	a	b	c				
			#	1						
			#	a	a	a	a			
			#		1					
			#	b	b	b	b			
			#				1			

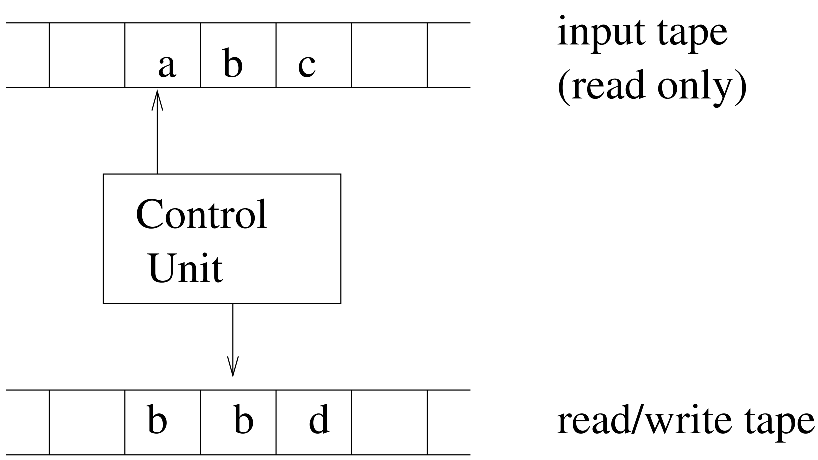
$\uparrow$

one tape head

$\Rightarrow$ multtrack TM can be converted to standard TM.

Definition: An Off-Line Turing Machine is a standard TM with 2 tapes: a read-only input tape and a read/write output tape.

Define δ: $Q \times \Sigma \times \Gamma \to Q \times \Gamma \times \{L, R\} \times \{L, R\}$



Theorem Class of standard TM's is equivalent to class of Off-line TM's.

similar to 2-track

Proof: (sketch)

- ($\Rightarrow$): Given standard TM M there exists an off-line TM M' such that L(M)=L(M').

- ($\Leftarrow$): Given an off-line TM M there exists a standard TM M' such that L(M)=L(M').

			#	a	b	c				
			#	1						
			#	b	b	d				
			#		1					

$\uparrow$

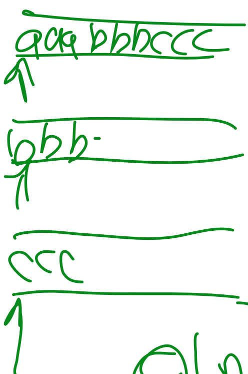
Running Time of Turing Machines

Example:

Given L=$\{a^n b^n c^n | n > 0\}$. Given w$\in \Sigma^*$, is w in L?

One tape was $\Theta(n^2)$

Write a 3-tape TM for this problem.



Copy B's to tape 2

Copy C's to tape 3 $\Theta(n)$

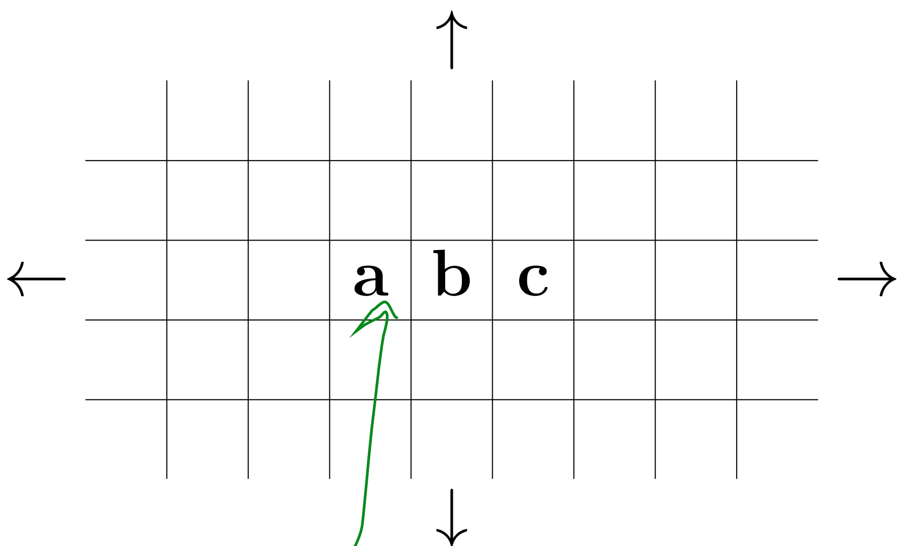
Start at beginning

Scan once & match a b, and a c for each a $\Theta(n)$

$\Theta(n)$

This model is more efficient! but not more powerful

Definition: An Multidimensional-tape Turing Machine is a standard TM with a multidimensional tape



Up
Down

Define δ**:** $Q \times \Gamma \to Q \times \Gamma \times \{R, L, U, D\}$

Theorem Class of standard TM's is equivalent to class of 2-dimensional-tape TM's.

Proof: (sketch)

- **($\Rightarrow$): Given standard TM M, construct a 2-dim-tape TM M' such that L(M)=L(M').**
  easy, just run it

- **($\Leftarrow$): Given 2-dim tape TM M, construct a standard TM M' such that L(M)=L(M').**

−2,1 is 00#1

		-1,2	1,2	2,2	
←	-2,1	-1,1	a 1,1	b 2,1	c 3,1
	-2,-1	-1,-1	1,-1	2,-1	

abc on tape

Construct M'

and then convert 2-track TM to the standard TM.

add -1,1 # #0#1

	#	**a**			**#**	**b**				**#**	**c**					
	#	**1**	**#**	**1**	**#**	**1**	**1**	**#**	**1**	**#**	**1**	**1**	**1**	**#**	**1**	
↑																

Unary 1's pos, 0's neg never use 0 to mean zero

for tape head {helpful to have one more track

Definition: A *nondeterministic* Turing machine is a standard TM in which the range of the transition function is a set of possible transitions.

Define δ: $Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$

Theorem Class of deterministic TM's is equivalent to class of nondeterministic TM's.

Proof: (sketch)

- **($\Rightarrow$): Given deterministic TM M, construct a nondeterministic TM M' such that L(M)=L(M').**

  run it, no nondeterminism!

- **($\Leftarrow$): Given nondeterministic TM M, construct a deterministic TM M' such that L(M)=L(M').**

  Construct M' to be a 2-dim tape TM.

A step consists of making one move for each of the current machines.

For example: Consider the following transition:

$$\delta(q_0, a) = \{(q_1, b, R), (q_2, a, L), (q_1, c, R)\}$$

Being in state q_0 with input abc.

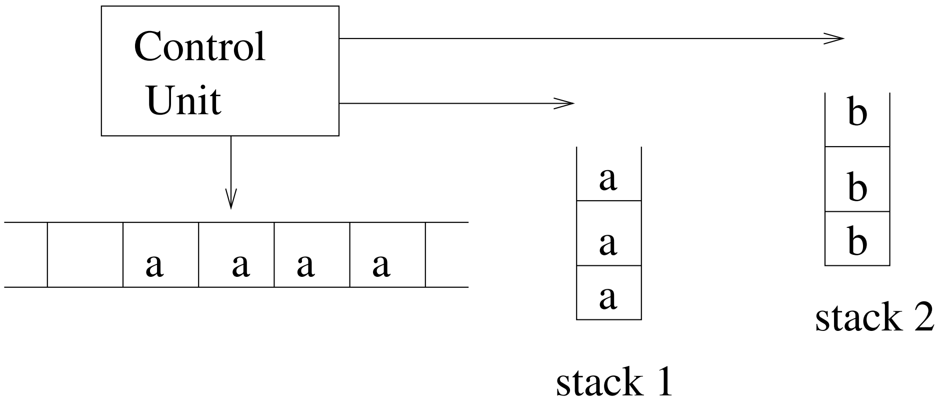
#	#	#	#	#
#	**a**	**b**	**c**	#
#	q_0			#
#	#	#	#	#

The one move has three choices, so 2 additional machines are started.

#	#	#	#	#	#
#		**b**	**b**	**c**	#
#			q_1		#
#		**a**	**b**	**c**	#
#	q_2				#
#		**c**	**b**	**c**	#
#			q_1		#
#	#	#	#	#	#

2-dim TM →
to a standard TM,

Definition: A 2-stack NPDA is an NPDA with 2 stacks.

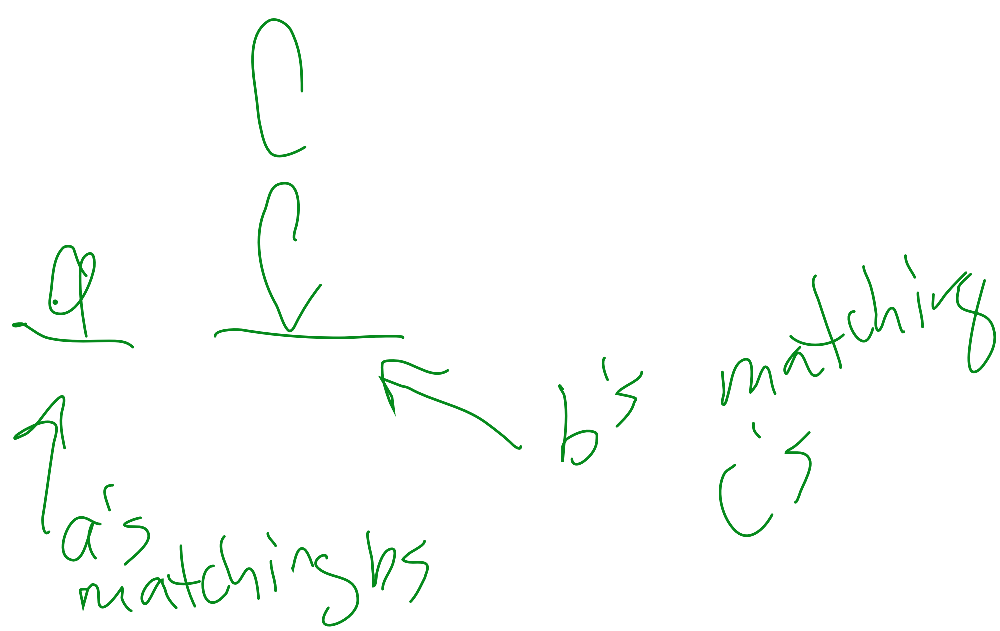


Define δ: $Q \times \Sigma \times \Gamma \cup \{\lambda\} \times \Gamma \cup \{\lambda\} \to$ subsets of $Q \times \Gamma^* \times \Gamma^*$

Consider the following languages which could not be accepted by an NPDA. Can you solve w/ 2-stack NPDA

1. $\mathbf{L}=\{a^n b^n c^n | n > 0\}$ yes
2. $\mathbf{L}=\{a^n b^n a^n b^n | n > 0\}$ yes
3. $\mathbf{L}=\{w \in \Sigma^* |$ **number of** a**'s equals number of** b**'s equals number of** c**'s**$\}$, $\Sigma = \{a, b, c\}$ yes

accbaacbb



STOPPED HERE

Theorem Class of 2-stack NPDA's is equivalent to class of standard TM's.

Proof: (sketch)

- **($\Rightarrow$): Given 2-stack NPDA, construct a 3-tape TM M' such that L(M)=L(M').**

- **($\Leftarrow$): Given standard TM M, construct a 2-stack NPDA M' such that L(M)=L(M').**

Universal TM - a programmable TM

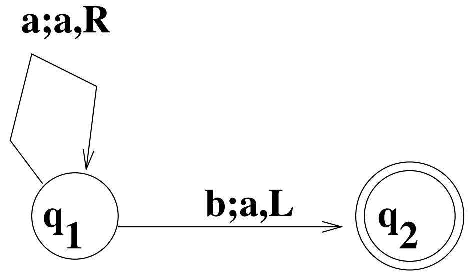
- **Input:**
  - **an encoded TM M**
  - **input string w**
- **Output:**
  - **Simulate M on w**

An encoding of a TM

Let TM M=$\{$Q,$\Sigma, \Gamma, \delta, q_1$,B,F$\}$

- **Q=**$\{q_1, q_2, \ldots, q_n\}$
  Designate q_1 as the start state.
  Designate q_2 as the only final state.
  q_n **will be encoded as n 1's**
- **Moves**
  L will be encoded by 1
  R will be encoded by 11
- $\Gamma = \{a_1, a_2, \ldots, a_m\}$
  where a_1 will always represent the B.

For example, consider the simple TM:



Γ={**B,a,b**} **which would be encoded as**

The TM has 2 transitions,

$$\delta(q_1,\mathbf{a})=(q_1,\mathbf{a},\mathbf{R}), \quad \delta(q_1,\mathbf{b})=(q_2,\mathbf{a},\mathbf{L})$$

which can be represented as 5-tuples:

$$(q_1,\mathbf{a},q_1,\mathbf{a},\mathbf{R}),(q_1,\mathbf{b},q_2,\mathbf{a},\mathbf{L})$$

Thus, the encoding of the TM is:

0101101011011010111011011010

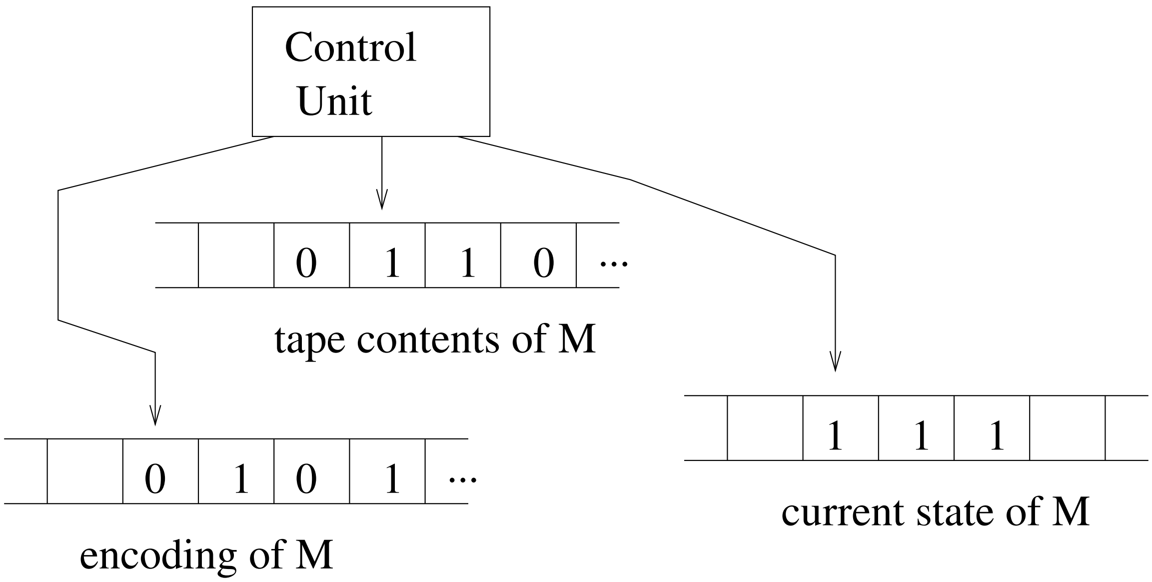
For example, the encoding of the TM above with input string "aba" would be encoded as:

0101101011011010111011011010011011101101

Question: Given $w \in \{0,1\}^+$, is w the encoding of a TM?

Universal TM

The Universal TM (denoted M_U) is a 3-tape TM:



Program for M_U

1. **Start with all input (encoding of TM and string w) on tape 1. Verify that it contains the encoding of a TM.**
2. **Move input w to tape 2**
3. **Initialize tape 3 to 1 (the initial state)**
4. **Repeat (simulate TM M)**
   (a) **consult tape 2 and 3, (suppose current symbol on tape 2 is a and state on tape 3 is p)**
   (b) **lookup the move (transition) on tape 1, (suppose δ(p,a)=(q,b,R).)**
   (c) **apply the move**
      - **write on tape 2 (write b)**
      - **move on tape 2 (move right)**
      - **write new state on tape 3 (write q)**

Observation: Every TM can be encoded as string of 0's and 1's.

Enumeration procedure - process to list all elements of a set in ordered fashion.

Definition: An infinite set is *countable* **if its elements have 1-1 correspondence with the positive integers.**

Examples:

- **S** $=$ { **positive odd integers** }
- **S** $=$ { **real numbers** }
- **S** $= \{w \in \Sigma^+\}$, $\Sigma = \{a, b\}$
- **S** $=$ { **TM's** }
- **S** $=$ {**(i,j)** $|$ **i,j>0, are integers**}

Linear Bounded Automata

We place restrictions on the amount of tape we can use,

	[	**a**	**b**	**c**	]			
	$\uparrow$							

Definition: A linear bounded automaton (LBA) is a nondeterministic TM M=(Q,$\Sigma, \Gamma, \delta, q_0$**,B,F) such that** $[,] \in \Sigma$ **and the tape head cannot move out of the confines of []'s. Thus,** $\delta(q_i, [) = (q_j, [, R)$**, and** $\delta(q_i,]) = (q_j,], L)$

Definition: Let M be a LBA. L(M)=$\{w \in (\Sigma - \{[,]\})^* | q_0[w] \overset{*}{\vdash} [x_1 q_f x_2]\}$

Example: L=$\{a^n b^n c^n | n > 0\}$ **is accepted by some LBA**