CPS 140 - Mathematical Foundations of CS
Dr. S. Rodger
Section: Other Models of Turing Machines (handout)

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

We will demonstrate equivalence between two classes of automata by showing that for every machine in one class, there is a machine in the second class that can simulate it, and vice versa.

Turing Machines with Stay Option

Allow the tape head to stay in place after writing.

Modify δ,

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)
- ($\Leftarrow$): Given a TM M with stay option, construct a standard TM M' such that L(M)=L(M').
  M=(K,Σ,, ,δ, q_0,B,F)
  M'=
  For each transition in M with a move (L or R) put the transition in M'. So, for

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

  put into δ'

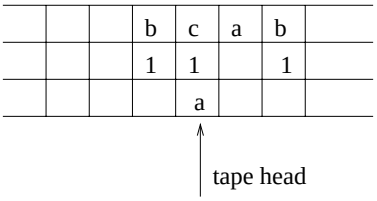
  For each transition in M with S (stay-option), move right and move left. So for

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

  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.

δ:

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

Proof: (sketch)

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

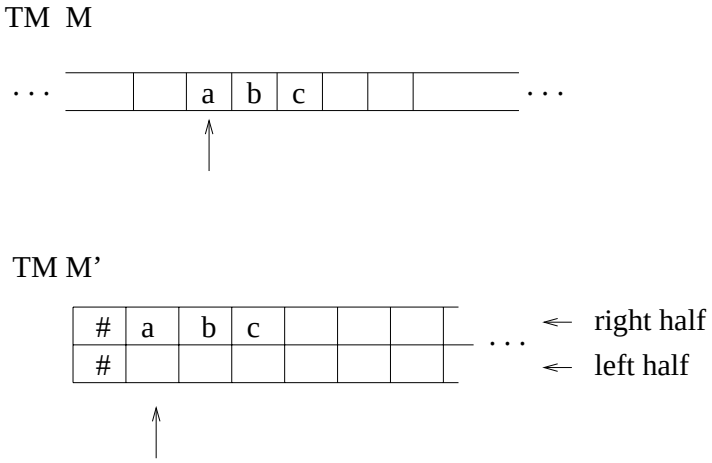
- ($\Leftarrow$): Given a TM M with multiple tracks there exists a standard TM M' such that L(M)=L(M').

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.

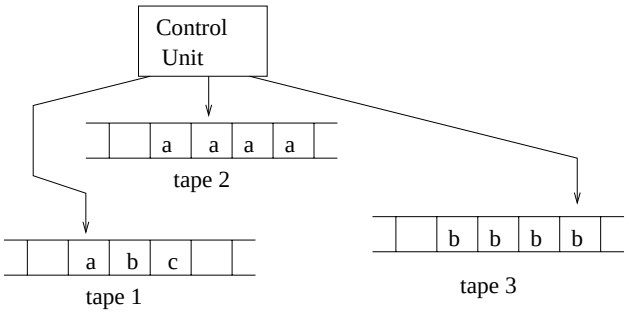
Proof: (sketch)

- ($\Rightarrow$): 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'



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

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



For an n-tape TM, define δ:

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

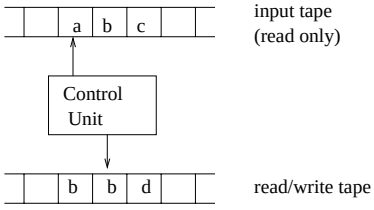
Proof: (sketch)

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

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

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:



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

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').

Running Time of Turing Machines

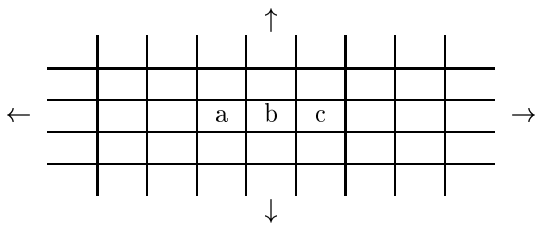
The running time of TM's is defined to be the count of the number of moves, writes and reads. Since each transition does all three of these, you can just count the number of moves, and multiple your answer by 3.

Example:

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

Write a 3-tape TM for this problem.

Definition: An Multidimensional-tape Turing Machine is a standard TM with a multidimensional tape (expands infinitely in multidimensions). (We will just examine the 2-dimensional-tape TM).



The tape head (not shown) would be pointing to one of the tape cells.

Define δ:

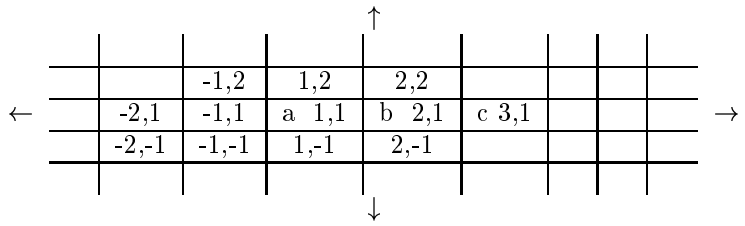
Theorem Class of standard TM's is equivalent to class of 2-dimensional-tape TM's.

Proof: (sketch)

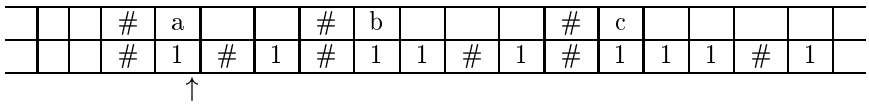
- ($\Rightarrow$): Given standard TM M, construct a 2-dim-tape TM M' such that L(M)=L(M').

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

  Imagine numbering each cell on the 2-dim tape TM using x-y-coordinates. Note that there are only a finite number of symbols (ignoring blanks) written on the 2-dim tape.



  Construct M' to be a 2 track TM. The first track stores all symbols (ignoring blanks) written on the 2-dim tape, and the second track stores the position of each symbol. Markers (#) are used to separate positions and symbols.



  The tape head in M' will point to the same symbol and position that M points to. M' will simulate a move in M by searching its tape for the next position. If the position is not listed, then the position is added with a B as the value written in that position.

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:

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

  Every time there is a choice in the transition function a new machine should be created.

  Construct M' to be a 2-dim tape TM. Two rows will represent the simulation of one machine. The first row stores the contents of the tape, and the second row stores the position of the tape head and the current state of the machine. Whenever there is a choice in M, two new rows will be added to M' to simulate the new machine. Markers (#) are used to indicate the current work area.

A step consists of making one move for each of the current machines. If any of these machines have a choice, then additional machines are added (i.e. two rows added to the 2-dim tape TM.

For example: Consider the following transition:

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

Suppose the 2-dim tape begins in state q_0 with input abc. Note that q_0 also represents the position of the tape head on M.

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

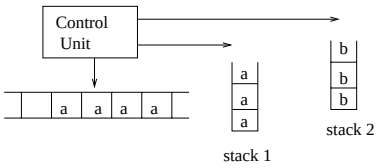
The one move has three choices, so 2 additional machines are started.

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

The first two rows represent taking the first choice above. The next two rows represent taking the second choice above. The last two rows represent taking the third choice above.

Note the tape head of M' has not be shown. It would point to a square within the marked region.

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



Define δ:

Consider the following languages which could not be accepted by an NPDA.

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

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

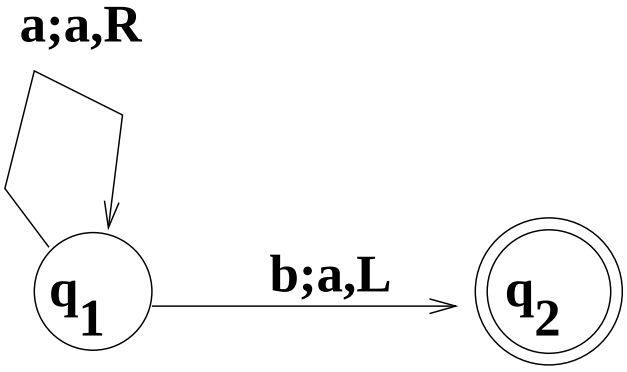
An encoding of a TM

Let TM M={K,Σ,, ,δ, q_1,B,F}

- K=$\{q_1, q_2, \ldots, q_n\}$
  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
- , $= \{a_1, a_2, \ldots, a_m\}$
  where a_1 will always represent the B.

If we use 0's as separators between symbols defining δ, then we can define a TM as a string of 0's and 1's.

For example, consider the simple TM:



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

The TM has 2 transitions,

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

which can be represented as 5-tuples:

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

Thus, the encoding of the TM is:

$$0101101011011010111011011010$$

where the first group of 1's (1) represents q_1, the second group of 1's (11) represents 'a', etc.

The first five groups of 1's represent the first transition above, and the second five groups of 1's represent the second transition above.

The input to the universal TM would be the code of a TM followed by the code of the input string. Use 2 0's to separate the TM code and the input string.

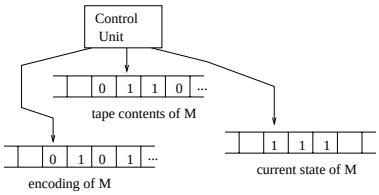
For example, the encoding of the TM above with input string "aba" would be encoded as:

$$01011010110110101110110110100110111011 0$$

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:



Tape 1 contains the encoding of a TM M. Tape 2 contains the contents of the tape of M. Tape 3 contains the current state.

Suppose TM M was in state q_3 and it's tape head was pointing to an 'a'. The universal TM would have the encoding of q_3 (represented by 111) on tape 3, and tape 2's head would be pointing to the first symbol in the encoding of an 'a'.

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:

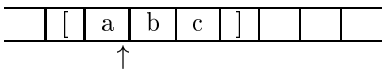
- S = { positive odd integers }
- S = { real numbers }
- S = $\{w \in \Sigma^+\}$, $\Sigma = \{a, b\}$

- 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, restricting our usage to the size of the input string. An input string is listed with '[' signifying the left end of the input and ']' signifying the right end of the input. The tape head cannot move to the left of '[' or to the right or ']'.



Definition: A linear bounded automaton (LBA) is a nondeterministic TM M=(K,Σ,, ,δ, 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