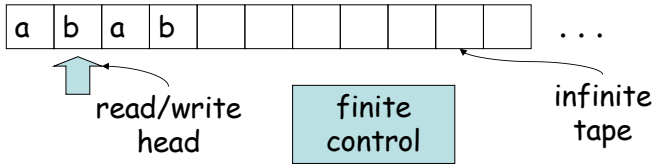
Umans Complexity Theory Lectures

Lecture 1b: Turing Machines

Turing Machines

- need a **model of computation** to define classes that capture important aspects of computation
- Our model of computation: Turing Machine



Turing Machines

- **Q** finite set of states
- ∑ alphabet including blank: "_"
- $\mathbf{q_{start}}$, $\mathbf{q_{accept}}$, $\mathbf{q_{reject}}$ in Q
- **δ** : Q x ∑ → Q x ∑ x {L, R, -} transition fn.
- input written on tape, head on 1st square, state $\mathbf{q_{start}}$
- sequence of steps specified by **δ**
- if reach $\mathbf{q_{accept}}$ or $\mathbf{q_{reject}}$ then halt

Turing Machines

- three notions of computation with Turing machines. In all, input x written on tape…
  - **function computation**: output f(x) is left on the tape when TM halts
  - **language decision**: TM halts in state q_{accept} if $x \in L$; TM halts in state q_{reject} if $x \notin L$.
  - **language recognition**: TM halts in state q_{accept} if $x \in L$; may loop forever otherwise.

Example:

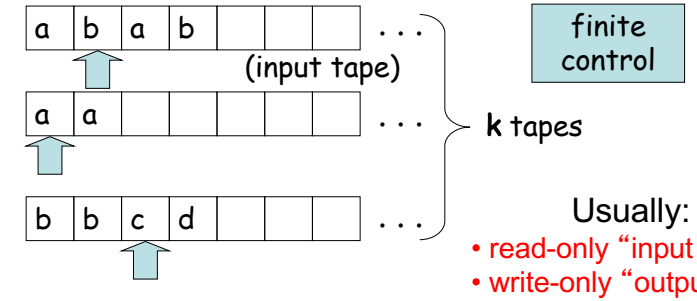
q	σ	δ(q,σ)
start	0	(start, 0, R)
start	1	(start, 1, R)
start	_	(t, _, L)
start	#	(start, #, R)

q	σ	δ(q,σ)
t	0	(accept, 1, -)
t	1	(t, 0, L)
t	#	(accept, #, R)

#	0	1								start
#	0	1								start
#	0	1								start
#	0	1								start
#	0	1								t
#	0	0								t
#	1	0								accept

Turing Machines

- **multi-tape** Turing Machine:

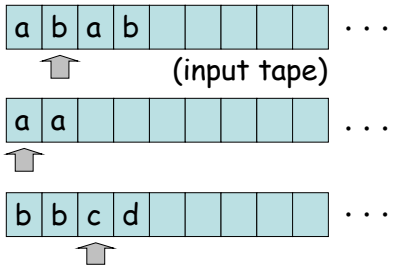


$\delta: Q \times \Sigma^k \rightarrow Q \times \Sigma^k \times \{L,R,-\}^k$

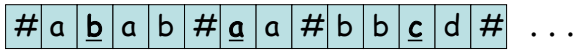
Usually:
- read-only "input tape"
- write-only "output tape"
- k-2 read/write "work tapes"

Multitape TMs

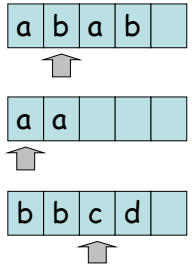
simulation of k-tape TM by single-tape TM:



- add new symbol **x** for each old x
- marks location of "virtual heads"



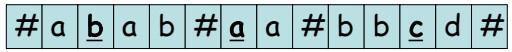
Multitape TMs



Repeat: O(t(n)) times

- scan tape, remembering the symbols under each virtual head in the state
  O(k t(n)) = O(t(n))
- make changes to reflect 1 step of M; if hit #, shift to right to make room.
  O(k t(n)) = O(t(n))

when M halts, erase all but output string
O(k t(n)) = O(t(n))



Extended Church-Turing Thesis

- the belief that TMs formalize our intuitive notion of an efficient algorithm is:

> The "extended" Church-Turing Thesis
>
> everything we can compute in time t(n) on a physical computer can be computed on a Turing Machine in time $t^{O(1)}(n)$ (polynomial slowdown)

- quantum computers challenge this belief

Extended Church-Turing Thesis

- consequence of extended Church-Turing Thesis: all reasonable physically realizable models of computation can be *efficiently* simulated by a TM

- e.g. multi-tape vs. single tape TM
- e.g. RAM model