ECE 664
Lecture 12
Avi KAK

Turing Machines for String Processing

- As in the Post-Turing model of computation, a Turing Machine uses a linear tape on which all the input strings are placed.
- However, unlike the Post-Turing system that uses a program to control the scanhead, there does <u>not</u> exist a program now.
- The Turing Machine model of computation is based on the notion of the machine being in some internal state.
- Let $q_1, q_2, \ldots, q_K$ be the different states of a machine. (We can also think of them as the states of the scan head.)
- Let $\{s_0, s_1, \ldots, s_n\}$ be the set of <u>all</u> the symbols that are allowed to be placed in the squares of the linear tape. As for Post-Turing machines, we will assume that $s_0 \equiv B$, the symbol for a blank square. We will consider the <u>rest</u> of the symbols to constitute the $\boxed{\text{alphabet}}$ of the Turing Machine
- Now we $\boxed{\text{define}}$ a Turing Machine to be a finite set of quadruples of the form

$$q_i \quad s_j \quad s_k \quad q_l$$

means if the machine is in state q_i with the scanhead looking at the symbol s_j, the scanhead prints the symbol s_k in that square, and the machine transitions into state q_l

$$q_i \quad s_j \quad R \quad q_l$$

$$q_i \quad s_j \quad L \quad q_l$$

The machine is in state q_i with the scanhead looking at symbol x_j. That causes the scanhead to move right by one square and then enter state q_l.

- For a deterministic TM, no two quadruples begin with the same $q_i s_j$. This means that in any given state and for any given symbol in a square, there can be only one action.
- By convention, a TM always begins in state q_1.
- By convention, a TM halts if in state q_i scanning symbol s_j, there does NOT exist a ~~transition~~ quadruple that begins with $q_i s_j$
- A TM $\mathcal{M}$ computes $f(x_1, \ldots, x_m)$ if when started in the tape configuration

$$\underset{\uparrow}{B}\, x_1\, B\, x_2\, B \ldots B\, x_m$$

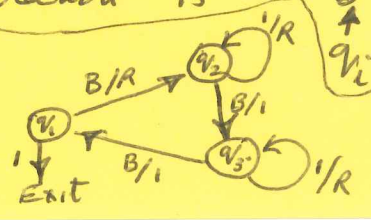
it eventually halts iff f is defined and if after halting, the output can be read off the tape by ignoring any blank squares.

- $\mathcal{M}$ computes f, a partial function on A^*, $\boxed{\text{strictly}}$ if the alphabet of $\mathcal{M}$ is a subset of A and the final tape configuration is $\underset{\uparrow\, q_i}{B}\, y$ with y containing no blanks.
- Example: $A = \{1\}$

$q_1\, B\, R\, q_2$ $\quad q_3\, 1\, R\, q_3$
$q_2\, 1\, R\, q_2$ $\quad q_3\, B\, 1\, q_1$
$q_2\, B\, 1\, q_3$

(state vs symbol table)

	q_1	q_2	q_3
B	$R q_2$	$1 q_3$	$1 q_1$
1		$R q_2$	$R q_3$



start with $\underset{\uparrow}{B}\,111$

end with $B\,1\,1\,1\,1\,1$ $\uparrow q_3$

$f(x) = x + 2$

TH: Any partial function which can be computed by a Post-Turing program can be computed by a Turing Machine using the same alphabet.

This theorem is <u>extremely important</u> because it tells us that a TM is <u>at least</u> as "powerful" as the Post-Turing language.

PROOF: This is proof by construction of a solution. More specifically, this is proof by simulation. Given any Post-Turing program $\mathcal{P}$, we will construct a TM $\mathcal{M}$ that simulates $\mathcal{P}$.

- Instructions of $\mathcal{P}$: $I_1, I_2, \ldots, I_K$
- List of symbols mentioned in $\mathcal{P}$: $s_0, s_1, s_2, \ldots, s_n$ with $s_0 \equiv B$
- Now, construct a TM $\mathcal{M}$ with states

$$q_1 \quad q_2 \quad q_3 \quad \ldots, q_K$$

$\mathcal{M}$ will be in state q_i just as $\mathcal{P}$ is about to execute I_i

- Translate the instructions of $\mathcal{P}$ into the quadruples of $\mathcal{M}$ in the following manner:

If I_i is $\boxed{\text{PRINT } s_k}$ $\Longrightarrow$ $q_i \; s_j \; s_k \; q_{i+1}$ for $j = 0, 1, 2, \ldots, n$

IF I_i is $\boxed{\text{RIGHT}}$ $\Longrightarrow$ $q_i \; s_j \; R \; q_{i+1}$ "

IF I_i is $\boxed{\text{LEFT}}$ $\Longrightarrow$ $q_i \; s_j \; L \; q_{i+1}$ ... "

IF I_i is $\boxed{\text{IF } s_k \text{ GO TO } L}$ $\Longrightarrow$ $\begin{cases} q_i \; s_k \; s_k \; q_m \\ q_i \; s_j \; s_j \; q_{i+1} & \text{for } j \neq k \end{cases}$

m is the least number such that I_m carries label L. If no such m, $m = K+1$

It is clear that the actions of $\mathcal{M}$ correspond precisely to the instructions of $\mathcal{P}$.

Given the previously established equivalence of Post-Turing languages with $\mathcal{S}$, we can now say that TMs are <u>at least</u> as "powerful" as $\mathcal{S}$. In other words:

TH: Let f be a partially computable function in $\mathcal{S}$, then there exists a TM that <u>computes</u> f strictly.

Now we would like to show a proof in the <u>opposite direction</u>. That is, we would like to demonstrate that if a function f is computed by a TM, then it is also computable by all the languages we have learned so far: $\mathcal{S}$, $\mathcal{S}_n$, PT. To establish this opposite direction proof, we need the notion of a quintuple TM.

Quintuple TM A quintuple TM consists of a finite set of quintuples of the kind:

$$q_i \; s_j \; s_k \; R \; q_l$$
$$q_i \; s_j \; k_k \; L \; q_l$$

in state q_i scanning s_j, print s_k in the same square, and <u>then</u> move one square to the right, and then enter state q_l.

TH: Any partial function that can be computed by a TM can be computed by a quintuple TM using the same alphabet.

Proof by Simulation: Let $\mathcal{M}$ be a TM with states $q_1, \ldots, q_K$ and alphabet $\{s_1, \ldots, s_n\}$. We construct a quintuple TM $\overline{\mathcal{M}}$ with states $q_1, \ldots, q_K, q_{K+1}, \ldots, q_{2K}$

$\mathcal{M}$	$\overline{\mathcal{M}}$
$q_i \; s_j \; R \; q_l$	$q_i \; s_j \; s_j \; R \; q_l$
$q_i \; s_j \; L \; q_l$	$q_i \; s_j \; s_j \; L \; q_l$
$q_i \; s_j \; s_k \; q_l$	$q_i \; s_j \; s_k \; R \; q_{K+l}$
	Additional quintuples:
	$q_{K+i} \; s_j \; s_j \; L \; q_i$ for $i = 1, 2, \ldots K$, $j = 0, 1, \ldots n$

TH: Any partial function which can be computed by a quintuple TM can be computed by a Post-Turing program using the same alphabet.

PROOF: We associate with each ~~label~~ state q_i a label A_i and with each pair $q_i s_j$ a label B_{ij}. Each label A_i is placed next to the first instruction of the following block:

$[A_i]$ IF s_0 GOTO B_{i0}
IF s_1 GOTO B_{i1}
⋮
IF s_n GOTO B_{in}
⋮
$[B_{ij}]$ PRINT s_k
RIGHT
GOTO A_l

If the quintuple TM contains $q_i s_j s_k R q_l$

Show the great Circle of <u>Equivalences</u> between $\mathcal{S}$, $\mathcal{S}_n$, PT, $\mathcal{M}$, $\overline{\mathcal{M}}$