CSC 438/2404

HW3 OUT! DUE NOV 11

Today: computability
Turing Machines

“On computable Numbers, with an application to the Entscheidungsproblem”

1936

- Concept of 1st generally convincing general model of computation.
- Proved there is no algorithm for deciding truth in mathematics
- Code breaking of Nazi ciphers WW2
- Also worked in mathematical biology
- Prosecuted in '52 for homosexuality

1912 - 1954
Turing Machines

\[ M = \{ Q, \Sigma, \Gamma, s, q_0, B, \varepsilon q_3 \} \]

\[ Q = \{ q_0, \ldots, q_k \} \text{ states, } k \geq 2 \]

\[ \Sigma = \text{finite input alphabet, including } 0, 1 \]

\[ \Gamma = \text{finite tape alphabet, } \Sigma \subseteq \Gamma, \text{ includes } B \] (blank symbol)

\[ q_0 : \text{start state} \]

\[ q_1 : \text{halt state} \]

\[ s : Q \times \Gamma \rightarrow Q \times \Gamma \times \Sigma \cup \{\varepsilon\} \cup \{R\} \]
\begin{itemize}
\item Initially \( M \) is in start state \( q_0 \), input in 1st cells, then B's
\item At any point in time, tape head points to some tape cell
\item Every cell contains an element of \( \Gamma \)
\end{itemize}
Turing Machines

Input $x = 011011$

- Initially $M$ is in start state $q_1$, input in 1st cells, then B's
- At any point in time, tape head points to some tape cell
  Initially head points to left most cell
Turing Machines  Input \( x = 011011 \)

\[ \begin{array}{cccccccccccccccc}
0 & 1 & 1 & 0 & 1 & 1 & B & B & B & B & B & B & B & B & B & B & B & B & \ldots \ldots
\end{array} \]

- Initially, \( M \) is in start state \( q_0 \), input in 1st cells, then B's
- At any point in time, tape head points to some tape cell
- Initially head points to left most cell
- At every time step, \( M \) makes one transition according to \( \delta \)
Turing Machines  

Input \( x = 011011 \)

\[
\begin{array}{cccccccccccc}
\text{0} & \text{1} & \text{1} & \text{0} & \text{1} & \text{1} & \text{B} & \text{B} & \text{B} & \text{B} & \text{B} & \text{B} & \ldots & \ldots \\
\hline
\end{array}
\]

\[ M = \{ Q, \Sigma, \Gamma, \delta, q_0, B, \{ q_2 \} \} \]

\[ Q = \{ q_0, q_1, q_2, q_3 \} \]

\[ \Sigma = \{ 0, 1, B \} \]

\[ \Gamma = \{ 0, 1, B \} \]

\[
S: \quad
\begin{align*}
(0, q_0) & \rightarrow (0, q_1, R) \\
(1, q_0) & \rightarrow (1, q_3, R) \\
(B, q_0) & \rightarrow (B, q_1, R) \\
(0, q_1) & \rightarrow (0, q_1, R) \\
(1, q_1) & \rightarrow (1, q_2, R) \\
(B, q_1) & \rightarrow (B, q_2, R) \\
(0, q_2) & \rightarrow (0, q_2, R) \\
(1, q_2) & \rightarrow (1, q_3, R) \\
(B, q_2) & \rightarrow (B, q_3, R) \\
\end{align*}
\]
Turing Machines

Input \( x = 011011 \)

\[
\begin{array}{cccccccc}
0 & 1 & 1 & 0 & 1 & 1 & B & B & B & B & B & B & B & \ldots \\
\end{array}
\]

\[ q_{01} \]

\[ S : \]

\[
\begin{align*}
(0, q_{01}) & \rightarrow (0, q_{01}, R) \\
(1, q_{01}) & \rightarrow (1, q_{01}, R) \\
(B, q_{01}) & \rightarrow (B, q_{01}, R) \\
(0, q_{03}) & \rightarrow (0, q_{03}, R) \\
(1, q_{03}) & \rightarrow (1, q_{02}, R) \\
(B, q_{03}) & \rightarrow (B, q_{03}, R)
\end{align*}
\]
Turing Machines

Input $x = 011011$

$$\begin{array}{cccccccccccc}
0 & 1 & 1 & 0 & 1 & 1 & B & B & B & B & B & B & B & \ldots & \ldots
\end{array}$$

$S:\n
(0, q_1) \rightarrow (0, q_1, R)\\
(1, q_1) \rightarrow (1, q_3, R)\\
(B, q_1) \rightarrow (B, q_1, R)\\
(0, q_3) \rightarrow (0, q_3, R)\\
(1, q_3) \rightarrow (1, q_2, R)\\
(B, q_3) \rightarrow (B, q_3, R)\\
$$
Turing Machines

Input $x = 011011$

\[
\begin{array}{cccccccccccc}
0 & 1 & 1 & 0 & 1 & 1 & B & B & B & B & B & B & B & B & B & B & B & \cdots & \cdots \\
\end{array}
\]

$S:
(0, q_0) \rightarrow (0, q_0, R)$

(1, q_0) \rightarrow (1, q_3, R)$

(B, q_0) \rightarrow (B, q_0, R)$

(0, q_1) \rightarrow (0, q_3, R)$

(1, q_1) \rightarrow (1, q_2, R)$

(B, q_1) \rightarrow (B, q_3, R)$
Turing Machines

Turing Machines compute n-ary partial (or total) functions from \( \mathbb{N}^n \rightarrow \mathbb{N} \) by encoding input/output as strings over \( \Sigma \).

Encoding of \( (a_1, \ldots, a_n) \in \mathbb{N}^n \) example

\[
(3, 10, 8) : \quad 112 \quad 10102100
\]

\( a_1 \) in binary \( \searrow \) \( a_2 \) in binary \( \searrow \) \( a_3 \) in binary

separated by “2”

Let \( \langle a_1, \ldots, a_n \rangle \) be the encoding of \( (a_1, \ldots, a_n) \).
Turing Machines

Turing Machines compute n-ary partial (or total) functions from $\mathbb{N}^n \to \mathbb{N}$ by encoding input/output as strings over $\Sigma$

$TM \ M$ on input $x$ halts when it enters halt state $(q_2)$

If $M$ halts on $x$, the output $y$ is the shortest string on tape with no B symbol
Turing Machines

Let \( f : \mathbb{N}^n \rightarrow \mathbb{N} \) be a total function.

\( M \) computes \( f \) if for every \( n \)-tuple \( (a_1, \ldots, a_n) \in \mathbb{N}^n \),
\( M \) on input \( \langle a_1, \ldots, a_n \rangle \) outputs \( f(a_1, \ldots, a_n) \)
(in binary).

If there is a TM \( M \) that computes \( f \),
then \( f \) is a total computable function.
Turing Machines

Let $f : (\mathbb{N} \cup \{\omega \})^{n} \to \mathbb{N} \cup \{\omega \}$ be a partial function (so $f(c_{1}, \ldots, c_{n}) = \omega$ if any $c_{i} = \omega$).

$M$ computes $f$ if for all $(a_{1}, \ldots, a_{n})$ in domain of $f$ $M$ on input $<a_{1}, \ldots, a_{n}>$ outputs $f(a_{1}, \ldots, a_{n})$.

$M$ may not halt on inputs not in domain of $f$.

If $f$ (a partial function) is computed by some $M$ then $f$ is a computable partial function.
Turing Machine Configurations

- A configuration describes the entire state of a TM at some point in time.

Configuration: $0, 1, 2, B, (q_{85}, 5), 1, 0, 1$
Turing Machine Configurations

- A tableaux is a sequence of configurations describing running $M$ on some input $x$. 
**Turing Machine Configurations**

- A tableaux is a sequence of configurations describing running $M$ on some input $x$

<table>
<thead>
<tr>
<th>$t=0$</th>
<th>$(q_0, 0)$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>2</th>
<th>B</th>
<th>. .</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t=1$</td>
<td>2</td>
<td>$(q_1, 0)$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>B</td>
<td>. .</td>
</tr>
<tr>
<td>$t=2$</td>
<td>2</td>
<td>2</td>
<td>$(q_1, 1)$</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>B</td>
<td>. .</td>
</tr>
<tr>
<td>$t=3$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>$(q_1, 1)$</td>
<td>0</td>
<td>2</td>
<td>B</td>
<td>. .</td>
</tr>
<tr>
<td>$t=4$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>$(q_1, 0)$</td>
<td>2</td>
<td>B</td>
<td>. .</td>
</tr>
<tr>
<td>$t=5$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>$(q_1, 2)$</td>
<td>B</td>
<td>. .</td>
</tr>
<tr>
<td>$t=6$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>$(q_2, B)$</td>
<td>. .</td>
</tr>
</tbody>
</table>
**Turing Machine Configurations**

- A tableaux is a sequence of configurations describing running $M$ on some input $x$.

<table>
<thead>
<tr>
<th>$t$</th>
<th>$q_1$</th>
<th>$0$</th>
<th>$1$</th>
<th>$1$</th>
<th>$0$</th>
<th>$2$</th>
<th>$B$</th>
<th>$\cdots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t=0$</td>
<td>$(q_1, 0)$</td>
<td>$0$</td>
<td>$1$</td>
<td>$1$</td>
<td>$0$</td>
<td>$2$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=1$</td>
<td>$2$</td>
<td>$(q_1, 0)$</td>
<td>$1$</td>
<td>$1$</td>
<td>$0$</td>
<td>$2$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$(q_1, 1)$</td>
<td>$1$</td>
<td>$0$</td>
<td>$2$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=3$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$(q_1, 1)$</td>
<td>$0$</td>
<td>$2$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=4$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$(q_1, 0)$</td>
<td>$2$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=5$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$(q_1, 2)$</td>
<td>$B$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$t=6$</td>
<td>$2$</td>
<td>$2$</td>
<td>$1$</td>
<td>$2$</td>
<td>$2$</td>
<td>$2$</td>
<td>$(q_2, B)$</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>
Let $\Sigma = \{0, 1, 2, 3\}$, 
$Q = \{q_1, q_2, \ldots, q_n\}$, 
$\Gamma = \{X_1, X_2, \ldots, X_k\}$ where $X_1 = 0$, $X_2 = 1$, $X_3 = 2$, $X_4 = B$.

$D_1 = \text{left}$, $D_2 = \text{right}$

We represent transition $\delta(q_i, x_j) \rightarrow (q_k, x_2, D_m)$ by $0^i 10^j 10^k 10^l 10^m$.

Code for $M$: $111 \text{ code}_1 11 \text{ code}_2 11 \ldots 11 \text{ code}_r 111$.

where $\text{code}_1, \ldots, \text{code}_r$ are the codes for transition function.
Encoding Turing Machines

Example. \( A = \{ q_1, q_2, q_3 \}, \Sigma = \{0, 1, \} \), \( M = \{001, 001\} \)

\[
\begin{align*}
\delta(q_1, 1) &= (q_3, 0, R) \\
\delta(q_3, 0) &= (q_1, 1, R) \\
\delta(q_3, 1) &= (q_2, 0, R) \\
\delta(q_3, \lambda) &= (q_3, 1, L) \\
\end{align*}
\]

\[
M = 111c_1 111c_2 111c_3 111c_4 111
\]

\((M, 110110)\) encoded as 111c_1 111c_2 111c_3 111c_4 111 110110

\(\#(M, x)\)

* uniquely decodable
Universal Turing Machines

U: Tokes as input $(M, x)$ and outputs $y$ if
$M$ on $x$ halts and outputs $y$
If $M$ does not halt on $x$, $U$ does not halt on $(M, x)$
Universal Turing Machines

U: Takes as input \#(M,x) and outputs y if M on x halts and outputs y.
If M does not halt on x, U does not halt on \#(M,x).

We describe a 3-tape TM (at a high level) for U.
(3-tapes can be simulated by one tape)

| tape 1 | \#(M,x) |
| tape 2 |          |
| tape 3 |          |
Universal Turing Machines

1. Initial state

- Tape 1: $\#(M, x)$
- Tape 2: 
- Tape 3: 

Check that contents of tape 1 is legal encoding of $M, x$.
Universal Turing Machines

2

tape 1

\[ 11 \# \text{code}_1 11 \text{code}_2 11 \ldots 11 \text{code}_r 11 \]

\[ \begin{array}{c}
\# \ 0 \ 1 \ 1 \ 0 \\
\hline
\# \ 0
\end{array} \]

tape 3

Initialize tapes 1 and 2 as above and tape 3 to contain \$0_1\text{ in binary}

evaluation of M

contents of M's tape at start

initial state of M
Universal Turing Machines

Loop

If tape 3 contains $00$ (halt state) halt and output contents of tape 2 (to 1st "B"

Now simulate next state:
Store contents of tape 2 head and current state of $M$ in $U$'s state. Scan tape 1 to find corresponding code,
Modify tapes 2, 3 accordingly
Universal Turing Machines

2

tape 1

\[
11 \# \text{code}_2 11 \text{code}_2 11 \ldots 11 \text{code}_2 11
\]

tape 2

\[
\# 001201 BB BB \ldots
\]

tape 3

\[
\# 00 BB BB BB \ldots
\]

Say \( \delta(q_2, 1) \rightarrow (q_3, 0, R) \)
Universal Turing Machines

Say \( \delta(q_2, 1) \rightarrow (q_3, 0, R) \)
Notation

\{ \times \} = \text{Turing machine } M \text{ such that } \#M = x

\times_{\#_1} = \text{the unary function computed by } x

\times_{\#_n} = \text{the } n\text{-ary function computed by } x

\text{(can generalize earlier so } M \text{ takes } n \text{ inputs instead of } 1)\)

A \text{ set is a subset of } \mathbb{N}^n \text{ (usually } n=1)\)

\text{a set/relation/0-1 valued total function }:

A \subseteq \mathbb{N} \text{ then } A(x) = 1 \text{ if } x \in A

0 \text{ if } x \notin A