Section: Other Models of Turing Machines

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

Turing Machines with Stay Option

Modify $\delta$, \[ \delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R,S}\]

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 $L(M) = L(M')$. 

\text{easy}
• \((\iff)\): 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

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

put into \(\delta'\)

\[\delta'(q'_i, a) = (q'_j, b, \text{L or R})\]

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

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

\[\delta'(q'_i, a) = (q'_j, b, \text{R})\]

\[\delta'(q'_i, c) = (q'_j, c, \text{L}) \quad \forall c \in \Gamma\]

\(L(M) = L(M'). \) QED.
Definition: A *multiple track* TM divides each cell of the tape into $k$ cells, for some constant $k$.

A 3-track TM:

\[
\begin{array}{cccc}
  & b & c & a & b \\
 1 & 1 & 1 & 1 \\
a & & & \\
\end{array}
\]

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

$\delta: Q \times (\Gamma^3 \times \Gamma^3) \to Q \times (\Gamma^3 \times \Gamma^3) \times \{L, R\}$
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 $L(M)=L(M')$.
  
  \[ \text{easy } M' \text{ just use one track} \]

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

Encode each combination of symbols from $\Gamma$ that can appear in a column of a track. Code would be the new tape alphabet.
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 $L(M) = L(M')$. 

Easy, but check for falling off left end. Put a special marker there.
Definition: An Multitape Turing Machine is a standard TM with multiple (a finite number) read/write tapes.

For an n-tape TM, define $\delta$: 

$$s_0 : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \Sigma^* \times \Delta^*$$
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 $L(M) = L(M')$.

\[
\begin{array}{cccc}
# & a & b & c \\
# & 1 \\
# & a & a & a & a \\
# & 1 \\
# & b & b & b & b \\
# & 1 \\
\end{array}
\]

Construct M’ to be a 2n-track TM.

Easy, just run it.
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 $\delta$:

\[
\begin{array}{c|c|c|c|}
 & a & b & c \\
\hline
\text{Control Unit} & \\
\hline
b & b & d
\end{array}
\]

input tape
(read only)

read/write tape

really 2 tape TM
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

Example:

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

Write a 3-tape TM for this problem.

Start: Input on tape 1

Copy input to tapes 2 & 3

Repeat till all as are marked

Simultaneously mark a on tape 1

& mark b on tape 2, c on tape 3

Input strings of length \( n \) in \( O(n) \) time

Standard TM running time \( O(n^3) \)

Faster running times w/ multiple tapes
Definition: An Multidimensional-tape Turing Machine is a standard TM with a multidimensional tape

\[\delta: Q \times S \rightarrow Q \times \Gamma \times \{L, R, U, D\}\]
Theorem Class of standard TM’s is equivalent to class of 2-dimensional-tape TM’s.

Proof: (sketch)

• (⇒): Given standard TM M, construct a 2-dim-tape TM M’ such that L(M)=L(M’).
  
  Easy; just uses one row. Doesn’t know about U/D directions.

• (⇐): Given 2-dim tape TM M, construct a standard TM M’ such that L(M)=L(M’).

  number the cells with x-y coordinates
Construct $M'$ to be a 2-track TM

First track stores Symbols (\#) separator
Second track x-y cord
Positive numbers unary w/ 1's
Neg. nos. unary with 0's

\[\begin{array}{ccc}
\# & a & \# \\
\# & 1 & \# & 1 & \# & b & \# & 1 & \# & 1 & \# & 1 & \# & c & \# & 1 & \# & 1 & \# & 1 & \# & 1
\end{array}\]

2-track TM is equiv.

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

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

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 $L(M)=L(M')$. (Easy, run it)  

• ($\Leftarrow$): 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.
The one move has three choices, so 2 additional machines are started.

```
# # # # # #
# b b c #
# q1 #
# a b c #
# q2 #
# c b c #
# q1 #
# # # # # #
```

2-dim type TM is equiv. standard TM
Definition: A 2-stack NPDA is an NPDA with 2 stacks.

Define $\delta$:

$$\delta: Q \times \Sigma \times (\mathcal{P}(\Gamma))^2 \rightarrow \mathcal{P}(Q \times \Gamma^*)$$
Consider the following languages which could not be accepted by an NPDA.

1. \( L = \{ a^n b^n c^n | n > 0 \} \)  
   **yes**

2. \( L = \{ a^n b^n a^n b^n | n > 0 \} \)  
   **yes**

3. \( L = \{ w \in \Sigma^* | \text{number of } a\text{'s equals number of } b\text{'s equals number of } c\text{'s} \}, \Sigma = \{a, b, c\} \)  
   **yes**
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 $L(M) = L(M')$. 

Diagram:
- NPDA
- Stack
- Tape
- Control unit
- Corresponding 3-tape TM
- $a\,a\,a\,a\,a\,a$
- $2\,c\,b\,a\,l$
- Tape 1
- Tape 2
- Tape 3
• ($\Leftrightarrow$): Given standard TM $M$, construct a 2-stack NPDA $M'$ such that $L(M)=L(M')$. 

Construct 2 stack NPDA

Copy input into stacks
Simulate tape w/ 2 stacks
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

- $\Gamma = \{ a_1, a_2, \ldots, a_m \}$
  - $a_1$ will always represent the B.
For example, consider the simple TM:

![Diagram of a simple TM]

\[ \Gamma = \{ B, a, b \} \] which would be encoded as

\[ \exists 1, 11, 111 \ldots \]

The TM has 2 transitions,

\[ \delta(q_1, a) = (q_1, a, R), \quad \delta(q_1, b) = (q_2, a, L) \]

which can be represented as 5-tuples:

\[ (q_1, a, q_1, a, R), (q_1, b, q_2, a, L) \]

Thus, the encoding of the TM is:

010110101101101011101101101011011010
For example, the encoding of the TM above with input string “aba” would be encoded as:

0101101011011011011010011011101101101101110110110

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

stopped here
Universal TM

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

![Diagram of a 3-tape TM]

- Control Unit
- Tape contents of $M$:
  - 0 1 1 0 ...
- Encoding of $M$:
  - 0 1 0 1 ...
- Current state of $M$:
  - 1 1 1
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 $\delta(p,a) = (q,b,R)$.)

   (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:

- $S = \{\text{positive odd integers}\}$ countable
- $S = \{\text{real numbers}\}$ not countable
- $S = \{w \in \Sigma^+, \Sigma = \{a, b\}\}$ countable
- $S = \{\text{TM’s}\}$ countable
- $S = \{(i,j) | i, j > 0, \text{are integers}\}$ countable

List all the numbers over 0,13 is w code of a TM, if so list it out.
Linear Bounded Automata

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

\[
\begin{array}{c|c|c|c}
| & a & b & c \\
\hline
\uparrow
\end{array}
\]

Definition: A linear bounded automaton (LBA) is a nondeterministic TM $M=\langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$ 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. $L(M)=\{w \in (\Sigma - \{[, ]\})^* | q_0[w] \vdash [x_1q_fx_2]\}$

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