Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Utility ENDMARK)

- Design a Turing machine ENDMARK that will shift the input string one space to the right, leaving an end-marker $\triangleright$ in the leftmost cell.
- That is, a Turing machine that computes $f(w) = \triangleright w$.

- We will write $w$ to mean that the read head is positioned at the leftmost symbol in the string $w$.
- So, more specifically, ENDMARK will compute $f(w) = \triangleright w$. 
Example (The Utility ENDMARK)

- **Strategy:**
  - Read the first symbol and write the end-marker.
  - Move right.
  - Read the second symbol and write the first symbol.
  - Move right.
  - Read the third symbol and write the second symbol.
  - Move right.
  - Continue in this way until a blank is read.
  - Write the last symbol.
  - Move left until the end-marker is read.
  - Move right one cell and halt.
Example (The Utility ENDMARK)

Strategy, continued:
- Continue in this way until a blank is read.
- Write the last symbol.
- Move left to the end-marker space.
- Move right one cell and halt.
Example (The Utility ENDMARK)

- Transition: \( 0 \rightarrow 0, R \)
- Transition: \( 0 \rightarrow @, R \)
- Transition: \( 1 \rightarrow @, R \)
- Transition: \( 1 \rightarrow 1, R \)
- Transition: \( + \rightarrow 0, L \)
- Transition: \( + \rightarrow 1, L \)
- Transition: \( @ \rightarrow R \)
- Transition: \( @ \rightarrow L \)
Examples of Turing Machines

- The Utility ENDMARK
- **The Utility LEFTEND**
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Utility LEFTEND)

- Design a Turing machine LEFTEND that will move the head to the first cell to the right of the end-marker.
- That is, ERASE computes $f(w) = \_w$. 
Example (The Utility LEFTEND)

- Standard assumption: The end-marker is present and the head is positioned in the first cell to the right of the head.
- Strategy:
  - If the current cell is not the end-marker, then move left.
  - If the current cell is the end-marker, then move right and halt.
Example (The Utility LEFTEND)

[@ → L]

[@ → R]
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Utility ERASE)

- Design a Turing machine ERASE that will erase the input, writing blanks in its place, but it will keep the end-marker $\rhd$.
- That is, ERASE computes $f(w) = \varepsilon$.
- This machine will erase all symbols from the end-marker to the first blank cell to the right.

- Can we build a Turing machine that erases all symbols from the tape?
Example (The Utility ERASE)

- **Strategy:**
  - Move to the left end of the tape.
  - If the current cell is not blank, write ⊓ and move right.
  - Repeat the previous step until the current cell is blank.
  - Then return to the cell to the left end and halt.
Example (The Utility ERASE)

\[ \text{LEFTEND} \xrightarrow{\neg + \rightarrow +, R} \text{LEFTEND} \xrightarrow{+ \rightarrow L} \text{LEFTEND} \]
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Decider EVEN)

- Design a Turing machine EVEN that will decide whether the binary input represents an even integer.
- That is, EVEN computes \( f(n) = \begin{cases} 1, & n \text{ is even;} \\ 0, & n \text{ is odd.} \end{cases} \).
Example (The Decider EVEN)

- **Strategy:**
  - Move right to the first blank (just past the end of the input).
  - Move left one cell.
  - If the current cell is 0, the number is even.
  - If the current cell is 1, the number is odd.
  - Erase the tape and write 0 or 1, accordingly.
Example (The Decider EVEN)

- \(+ \rightarrow R\)
- \(+ \rightarrow L\)
- \(0 \rightarrow L\)
- \(1 \rightarrow L\)
- \(1 \rightarrow R\)
- \(0 \rightarrow R\)
- \(\text{ERASE}\)
- \(\rightarrow 1, R\)
- \(\rightarrow 0, R\)
- \(\rightarrow L\)
An alternative to writing 0 or 1 is to use the accept and reject states.
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Accepter EQUAL)

- Design a Turing machine EQUAL that will accept the input if it contains an equal number of 0's and 1's.
- That is, EQUAL computes

\[ f(w) = \begin{cases} 
  1, & \text{if } w \text{ has an equal number of 0's and 1's;} \\
  0, & \text{if } w \text{ has an unequal number of 0's and 1's.} 
\end{cases} \]
Strategy

Example (The Accepter EQUAL)

- Strategy:
  - Move right to the first 0.
  - Write $ and move right.
  - Move to the left end.
  - Move right to the first 1.
  - Write $ and move right.
  - Move to the left end.
Example (The Accepter EQUAL)

**Strategy:**
- Continue the previous six steps until one of two things happens.
  - We find a 0, but we fail to find a 1. Reject the input.
  - We fail to find a 0.
- If we fail to find a 0, then we search for a 1. One of two things happens.
  - We find a 1. Reject the input.
  - We fail to find a 1. Accept the input.
The Accepter EQUAL

Example (The Accepter EQUAL)

Diagram showing transitions for the accepter EQUAL.
1 Examples of Turing Machines
- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

2 Assignment

Robb T. Koether  (Hampden-Sydney College)  Turing Machines - Examples  Fri, Oct 19, 2012  27 / 45
Example (The Function evaluator INCR)

- Design a Turing machine INCR that will add 1 to the input.
- That is, INCR computes \( f(n) = n + 1 \).
- For example, if the input is 100111, the output will be 101000.
Example (The Function evaluator INCR)

- Strategy:
  - Move right to the first blank.
  - Move left one cell.
  - If the symbol is $0$, write $1$ and move left.
  - If the symbol is $1$, write $0$ and move left.
  - Repeat the previous two steps until one of two things happens.
    - A $0$ is read (and changed to $1$).
    - The end-marker is read.
Example (The Function evaluator INCR)

- **Strategy:**
  - If 0 is read, then move to the left end and halt.
  - If the end-marker is read, then the original input was $111\ldots1$, in which case there is a final carry-out.
  - In this case, write 1, move to the right end, and write 0 in the first blank.
Example (The Function evaluator INCR)

\[\rightarrow + \rightarrow R\]
\[\rightarrow + \rightarrow L\]
\[\rightarrow 0 \rightarrow 1, L\]
\[\rightarrow @ \rightarrow R\]
\[\rightarrow + \rightarrow 0, L\]
\[\rightarrow 1 \rightarrow 1, R\]
\[\rightarrow 1 \rightarrow R\]
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Accepter EQUAL
- The Function Evaluator INCR

The Language Processor NEXTLEX

- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Language Processor NEXTLEX)

- Design a Turing machine NEXTLEX that will output the lexicographical successor of the input.
- That is, NEXTLEX computes \( f(w) = \text{succ}(w) \).
- For example,

\[
\begin{align*}
  f(\varepsilon) &= 0 \\
  f(0) &= 1 \\
  f(1) &= 00 \\
  f(00) &= 01.
\end{align*}
\]
Example (The Language Processor NEXTLEX)

- NEXTLEX is very similar to INCR.
- The difference is when the input is $\varepsilon$ and when it is all 1's.
- Strategy:
  - If the cell is blank, write 0.
  - Otherwise, do the same as INCR, except that in the special case of all 1's, do not write the initial 1.
Example (The Language Processor NEXTLEX)

\[
\begin{align*}
\neg + & \rightarrow R \\
+ & \rightarrow L \\
1 & \rightarrow 0, L \\
0 & \rightarrow 1, L \\
@ & \rightarrow R \\
-+ & \rightarrow R \\
+ & \rightarrow 0, L \\
\text{LEFTEND} & \\
\end{align*}
\]
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Acceptor EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Example (The Language Processor COPY)

- Design a Turing machine that will output the original input, followed by #, followed by a copy of the original input.
- That is, computes \( f(w) = w \# w \).
- For example,

\[
\begin{align*}
  f(\varepsilon) &= \# \\
  f(0) &= 0 \# 0 \\
  f(0010) &= 0010 \# 0010.
\end{align*}
\]
Example (The Language Processor COPY)

- COPY is somewhat similar to EQUAL.
- **Strategy:**
  - Move right and write #.
  - Move left to beginning of input.
  - If the cell is 0, write $, move right to blank, and write 0.
  - Move left to $ and write 0.
  - Similarly if cell is 1.
  - Continue until # is encountered.
Example (The Language Processor COPY)

The diagram illustrates a Turing machine with states and transitions. The states are labeled $q_0, q_1, q_2, q_3, q_4, q_5$. The arrows indicate transitions based on the symbol read and the action taken (move left or right, read symbol, write symbol). For example, from state $q_0$, on reading $\_\_ \to R$, the machine moves right. The transitions include:

- $q_0 \rightarrow R$ to $q_1$ on reading $\_\_ \to R$
- $q_1 \rightarrow L$ on reading $\# \rightarrow L$
- $q_1 \rightarrow R$ on reading $0 \rightarrow \$, $R$
- $q_2 \rightarrow L$ on reading $\$ \rightarrow 0, L$
- $q_2 \rightarrow R$ on reading $\$ \rightarrow L$
- $q_3 \rightarrow L$ on reading $1 \rightarrow \$, $L$
- $q_3 \rightarrow R$ on reading $\_ \rightarrow R$
- $q_4 \rightarrow L$ on reading $\$ \rightarrow 1, R$
- $q_4 \rightarrow R$ on reading $\$ \rightarrow L$
- $q_5 \rightarrow L$ on reading $\$ \rightarrow 0, R$
- $q_5 \rightarrow R$ on reading $\$ \rightarrow L$

The machine transitions through these states and actions, simulating the computation process of the language processor.
Examples of Turing Machines

- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Acceptor EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM
Example (The Language Enumerator ENUM)

- Design a Turing machine that will print all binary strings in $\{0, 1\}^*$, separated by #.
Example (The Language Enumerator ENUM)

- **Strategy:**
  - Begin with a blank tape.
  - Run COPY.
  - Run NEXTLEX.
  - Repeat the previous two steps in an endless loop.

- The machines NEXTLEX and COPY must be modified to use # as the end-marker.
Example (The Language Enumerator ENUM)
Examples of Turing Machines
- The Utility ENDMARK
- The Utility LEFTEND
- The Utility ERASE
- The Decider EVEN
- The Acceptor EQUAL
- The Function Evaluator INCR
- The Language Processor NEXTLEX
- The Language Processor COPY
- The Language Enumerator ENUM

Assignment
Assignment

- Design a Turing machine that will subtract 1 from the input.
  - Special case: if the input is 0, the output is 0.
- Design a Turing machine that will accept the language \( \{0^n1^n \mid n \geq 0\} \).
- Describe how to design a Turing machine that will print all binary strings that contain an equal number of 0’s and 1’s.