# CSL253 - Theory of Computation

Tutorial 6

**Team Members**


1. Krishna Jhanwar - 12341250
2. Siddharth Rai - 12342050
3. Slok Tulsyan - 12342080
4. Sneha Nagmoti - 12342090


**Question 1**

Draw & define Turing Machine's (TMs) at state level describing its L/R movements for:

1. Left shift of a binary number
2. Right shift of a binary number
3. Addition of two 8-bit binary numbers
4. Subtraction of two 8-bit binary numbers

# Solution

## 1. Left Shift of a binary number

**Example**

**Input:** | $\sqcup$ | 0 | 1 | 0 | 1 | $\sqcup$ |

**Output:** | $\sqcup$ | 1 | 0 | 1 | 0 | $\sqcup$ |

**Implementation Level Description**

**Steps:**

1. Move to the end of the binary string on the tape to find the first blank symbol ($\sqcup$), then shift one cell to the left.
   - If the symbol is '0' or '1', move right (R).
   - If the symbol is '$\sqcup$', move left (L).

2. Copy each bit to the left by overwriting the current cell with the bit to its right. The rightmost bit should be replaced with '0'.
   (a) Remember the current bit, then write either:
   - '0' if it's the rightmost bit, or
   - the last remembered bit.

   (b) Move one cell to the left (L).

   (c) If the current cell contains '$\sqcup$', halt; otherwise, repeat from step (a).

**Defining the Turing Machine**

We define the TM as a seven-tuple $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where:

- **States:** $Q = \{q_0, q_1, q_2, q_3, q_4\}$, where:
  - $q_0$: Initial state.
  - $q_1$: Indicates encountering a '$\sqcup$' while moving right; transition to the previous cell to position on the last binary digit.
  - $q_2$: Represents that the last read digit was '0'.
  - $q_3$: Represents that the last read digit was '1'.
  - $q_4$: Accept state, indicating that the binary number has been successfully left-shifted.
- **Input Alphabets:** $\Sigma = \{0, 1\}$
- **Tape Alphabets:** $\Gamma = \{0, 1, \sqcup\}$

- **Transition Function:** $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$$\boxed{\delta(\mathit{state}, \mathit{symbol}) = (\mathit{new\ state}, \mathit{write\ symbol}, \mathit{move})}$$

The transitions are defined as follows:

1. Traverse to the end of the binary string
   $\delta(q_0, 0) = (q_0, 0, R)$
   $\delta(q_0, 1) = (q_0, 1, R)$

2. Step back after encountering the first blank symbol '$\sqcup$'
   $\delta(q_0, \sqcup) = (q_1, \sqcup, L)$

3. Overwrite the last digit with '0'
   $\delta(q_1, 0) = (q_2, 0, L)$
   $\delta(q_1, 1) = (q_3, 0, L)$

4. Shift left while carrying '0'
   $\delta(q_2, 0) = (q_2, 0, L)$
   $\delta(q_2, 1) = (q_3, 0, L)$

5. Shift left while carrying '1'
   $\delta(q_3, 0) = (q_2, 1, L)$
   $\delta(q_3, 1) = (q_3, 1, L)$

6. Halt upon encountering a blank on the left
   $\delta(q_2, \sqcup) = (q_4, \sqcup, H)$
   $\delta(q_3, \sqcup) = (q_4, \sqcup, H)$

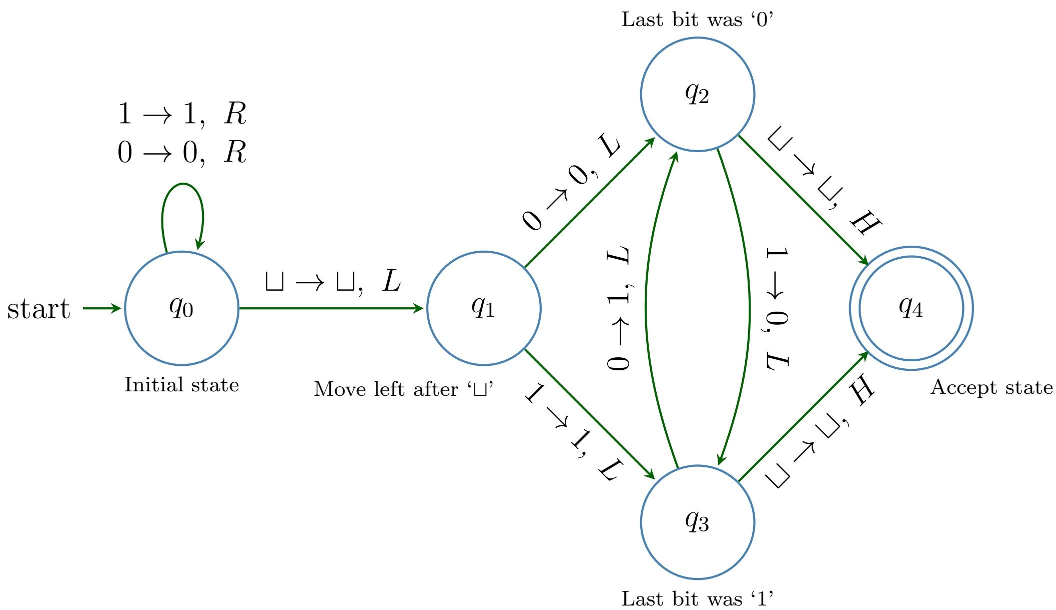
**Note:** '$H$' indicates the machine halts

Explanation:

The Turing Machine begins in state $q_0$, where it moves right across the tape until it encounters the first blank symbol '$\sqcup$'. Upon finding this blank, it transitions to state $q_1$ and moves one step left to position the head on the last binary digit. Regardless of whether this digit is a '0' or '1', it is overwritten with '0', effectively discarding it in the left shift operation. The machine then transitions to state $q_2$ or $q_3$, based on whether the overwritten digit was '0' or '1', and remembers this value. In state $q_2$ or $q_3$, the machine continues moving left, overwriting each cell with the remembered digit from the previous step. This process continues until it encounters another blank on the left, at which point it transitions to the accept state $q_4$, indicating that the left shift of the binary number has been completed successfully.

- **Initial State:** $q_0$
- **Acceptance States:** $q_{accept} = \{q_4\}$
- **Reject States:** $q_{reject} = \{\}$

**State Diagram for the Turing Machine**



## 2. Right Shift of a binary number

### Example

**Input:**

| | | | | | |
|---|---|---|---|---|---|
| $\sqcup$ | 0 | 1 | 0 | 1 | $\sqcup$ |

**Output:**

| | | | | | |
|---|---|---|---|---|---|
| $\sqcup$ | 0 | 0 | 1 | 0 | $\sqcup$ |

### Implementation Level Description

**Steps:**

1. Copy each bit to the right by overwriting the current cell with the bit to its left. The leftmost bit should be replaced with '0'.

   (a) Remember the current bit, then write either:
   - '0' if it's the leftmost bit, or
   - the last remembered bit.

   (b) Move one cell to the right (R).

   (c) If the current cell contains '$\sqcup$', halt; otherwise, repeat from step (a).

### Defining the Turing Machine

We define the TM as a seven-tuple $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where:

- **States:** $Q = \{q_0, q_1, q_2, q_3\}$, where:
  - $q_0$: Initial state.
  - $q_1$: Represents that the last read digit was '0'.
  - $q_2$: Represents that the last read digit was '1'.
  - $q_3$: Accept state, indicating that the binary number has been successfully right-shifted.
- **Input Alphabets:** $\Sigma = \{0, 1\}$
- **Tape Alphabets:** $\Gamma = \{0, 1, \sqcup\}$

- **Transition Function:** $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

$$\boxed{\delta(state, symbol) = (new\ state, write\ symbol, move)}$$

The transitions are defined as follows:

1. Overwrite the first digit with '0'

   $\delta(q_0, 0) = (q_1, 0, R)$
   $\delta(q_0, 1) = (q_2, 0, R)$

2. Shift right while carrying '0'

   $\delta(q_1, 0) = (q_1, 0, R)$
   $\delta(q_1, 1) = (q_2, 0, R)$

3. Shift right while carrying '1'

   $\delta(q_2, 0) = (q_1, 1, R)$
   $\delta(q_2, 1) = (q_2, 1, R)$

4. Halt upon encountering a blank on the right

   $\delta(q_1, \sqcup) = (q_3, \sqcup, H)$
   $\delta(q_2, \sqcup) = (q_3, \sqcup, H)$

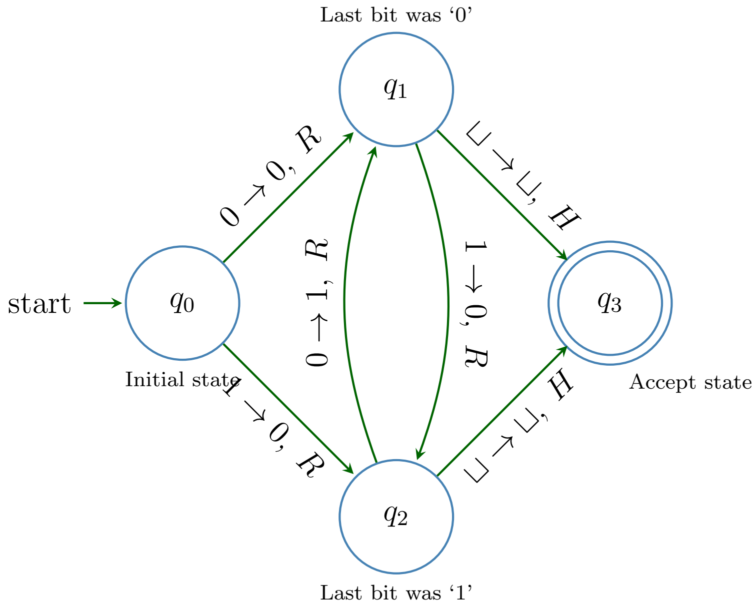
**Note:** '$H$' indicates the machine halts

Explanation:

The Turing Machine begins to position the head on the first binary digit with state $q_0$. Regardless of whether this digit is a '0' or '1', it is overwritten with '0', effectively discarding it in the right shift operation. The machine then transitions to state $q_1$ or $q_2$, based on whether the overwritten digit was '0' or '1', and remembers this value. In state $q_1$ or $q_2$, the machine continues moving right, overwriting each cell with the remembered digit from the previous step. This process continues until it encounters another blank on the right, at which point it transitions to the accept state $q_3$, indicating that the right shift of the binary number has been completed successfully.

- **Initial State:** $q_0$
- **Acceptance States:** $q_{accept} = \{q_3\}$
- **Reject States:** $q_{reject} = \{\}$

## State Diagram for the Turing Machine



## 3. Addition of two 8 bit binary numbers

### Idea for Constructing the Turing Machine

- Given two binary numbers $A$ and $B$, we compute $A + B$ using a simple iterative approach.
- We treat the second number $B$ as a **counter**. For each decrement of $B$ by one, we increment $A$ by one.
- This process is repeated until $B$ becomes zero. At that point, $A$ holds the result of $A + B$.

### STEP BY STEP PROCESS

1. Move the head on the tape until you encounter the beginning of the second number.
2. Now for addition, we treat the second number as a **counter**.
3. Keep moving the head to the end of the second number.
4. Reverse iterate the second number and decrement it by one:
   - If you encounter a `0`, change it to `1` and move left.
   - If you encounter the first `1`, change it to `0`, move left — this completes the decrement.
   - If no such `1` is found (i.e., the second number is already zero), clean up the second number and halt the entire process.
5. After decrementing the second number, reverse iterate the first number and increment it by one:
   (a) If you encounter a `1`, change it to `0` and move left.
   (b) If you encounter the first `0`, change it to `1`, move left — this completes the increment.
   (c) If no such `0` is found (i.e., the number was all `1`s), this is a case of overflow.
   (d) If overflow is allowed, write `1` in the blank cell to the left; otherwise, reject.
6. After completing the increment of the first number, move the head forward and repeat from step 3.

# Formal Definition of TM

We define the TM as a seven-tuple $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where:

# States and Their Descriptions

Let $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ where each state indicates:

- $q_0$ – Reading the 1st number in forward direction.
- $q_1$ – The first number is over and we are forward iterating on the 2nd number.
- $q_2$ – Forward iteration on 2nd number complete. Now reverse iterating in 2nd number and subtracting 1 from it.
- $q_3$ – Successfully subtracted 1 from the 2nd number. Now reversed to $q_0$ and simply resumes.
- $q_4$ – 2nd number is over, and we are now reverse iterating the 1st number and adding 1 to it.
- $q_5$ – 2nd number has become 0, and we are at the clean-up stage.
- $q_6$ – Addition process after clean-up is complete.
- $q_7$ – Overflow in the 1st number.

# Turing Machine Components

- $\Sigma = \{0, 1\}$ (input symbols)
- $\Gamma = \{0, 1, \sqcup\}$ (tape alphabet, where $\sqcup$ is the blank symbol)

# Transition Function $\delta$

## Move right to end of first block

$$\begin{gathered}
\delta(q_0, 0) = (q_0, 0, R) \\
\delta(q_0, 1) = (q_0, 1, R) \\
\delta(q_0, \sqcup) = (q_1, \sqcup, R)
\end{gathered}$$

## Move right to end of second block

$$\begin{gathered}
\delta(q_1, 0) = (q_1, 0, R) \\
\delta(q_1, 1) = (q_1, 1, R) \\
\delta(q_1, \sqcup) = (q_2, \sqcup, R)
\end{gathered}$$

### Subtract one in binary

$$\delta(q_2, 0) = (q_2, 1, L)$$
$$\delta(q_2, 1) = (q_3, 0, L)$$
$$\delta(q_2, \sqcup) = (q_5, \sqcup, R)$$

### Move left to end of first block

$$\delta(q_3, 0) = (q_3, 0, L)$$
$$\delta(q_3, 1) = (q_3, 1, L)$$
$$\delta(q_3, \sqcup) = (q_4, \sqcup, L)$$

### Add one in binary

$$\delta(q_4, 0) = (q_0, 1, R)$$
$$\delta(q_4, 1) = (q_4, 0, L)$$
$$\delta(q_4, \sqcup) = (q_7, \sqcup, R)$$

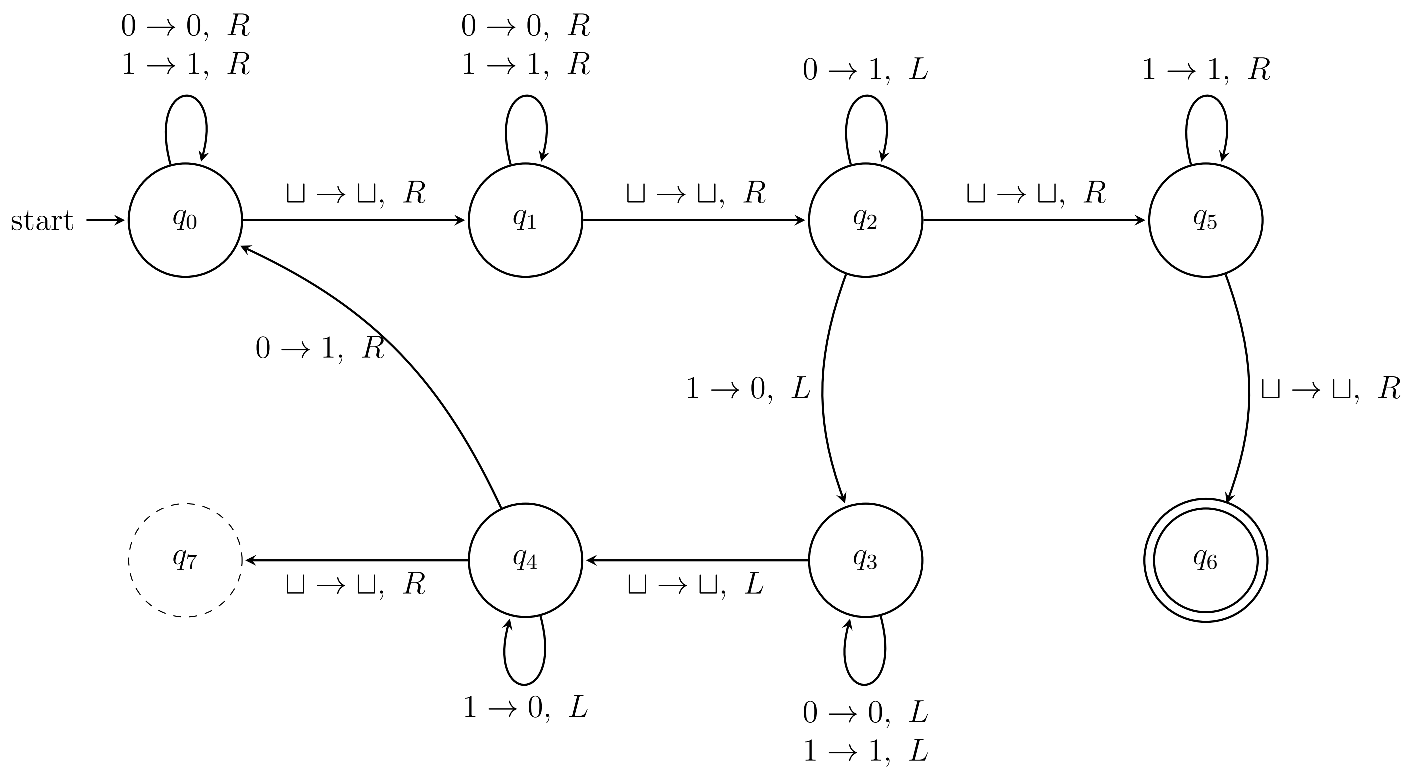
### Clean-up

$$\delta(q_5, 1) = (q_5, 1, R)$$
$$\delta(q_5, \sqcup) = (q_6, \sqcup, R) \quad \texttt{(Halt)}$$

## Start, Accept, and Reject States

- Start state: $q_0$
- Accept state: $q_6$
- Reject states: $q_7$

## State Diagram for the Turing Machine



## 4. Subtraction of two 8 bit binary numbers

### Idea for Constructing the Turing Machine

- Given two binary numbers $A$ and $B$, to compute $A - B$, we can instead perform $A +$ (two's complement of $B$).
- So, we first compute the two's complement of $B$ (the second number), and then proceed with the addition.
- For the ADDITION, we treat the second number as a counter: for each decrement of $B$, we increment $A$ by one. This process is repeated until $B$ becomes zero.

### STEP BY STEP PROCESS

1. keep moving the head on the tape until you encounter the begining of the second number.
2. Then keep moving the head forward on the second number and keep flipping the bits i.e change 0 to 1 and 1 to 0 until u reach the end of the second number.This will give you 1's complement of second number.
3. When you reach the rightmost bit of the second number ,add 1 to it and move left i.e if the bit is 0 add 1 and make its value 1 and move left and halt otherwise if it is 1 make it 0 and move left and carry the 1 and repeat the process until you encounter the first 0 or a separator (i.e beginning of the second number).Now we have successfully converted our second number into it's 2's complement.
4. move the head back to the leftmost bit of the first number and start addition.
5. Now for addition: keep moving the head until you reach the end of the second number
6. Now Reverse iterate the second number and subtract:
   - if you encounter 0 make it 1 and move left.
   - If you encounter 1st 1 make it 0 , move left and decrement is done.
   - If no such one was found i.e the second number is 0 , we clean up the 2nd number and halt the entire process.
7. Continue to reverse iterate the 1st number and add 1 to it:
   (a) if you encounter 1 make it 0 and move left.
   (b) if you encounter 1st 0 make it 1 , move left and increment is done.
   (c) If no such 0 was found then that means the number was 11111111 i.e 28-1 (limit was reached) then this is the case of overflow.
   (d) if overflow allowed we make blank -¿1 otherwise we reject.
8. in step 7 we make the 1st 0 as 1 n after that start iterating in forward direction and go to step 5.

## Defining the Turing Machine

We define the TM as a seven-tuple $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where:

- **States:** $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}\}$, where:
  - $q_0$: Initial state; iterating over the first number in the forward direction.
  - $q_1$: Indicates encountering a '$\sqcup$' while moving right; continue forward iteration over the second number with bit-flipping (i.e., converting 1 to 0 and 0 to 1).
  - $q_2$: Add 1 to complete the two's complement operation.
  - $q_3$: Reached after forming the two's complement; the head moves back to the beginning of the second number.
  - $q_4$: Iterate left so that the head moves back to the beginning of the first number.
  - $q_5$: Iterate forward over the first number.
  - $q_6$: Iterate forward over the second number.
  - $q_7$: Subtract '1' from the second number: convert '0' to '1', and iterate backward. On encountering the first '1', convert it to '0' and move to $q_8$.
  - $q_8$: The second number is exhausted; this is the clean-up stage.
  - $q_9$: Subtraction and clean-up complete; accept state.
  - $q_{10}, q_{11}$: reverse iterating in 1st number and adding '1' to it i.e converting '1' to '0' until u find first '0' n then make it '1'.
  - $q_{12}$: indicates overflow in 1st number.
- **Input Alphabets:** $\Sigma = \{0, 1\}$
- **Tape Alphabets:** $\Gamma = \{0, 1, \sqcup\}$
- **Transition Function:** $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$$\boxed{\delta(state, symbol) = (new\ state, write\ symbol, move)}$$

The transitions are defined as follows:

1. move right to the end of 1st block

   $\delta(q_0, 0) = (q_0, 0, R)$
   $\delta(q_0, 1) = (q_0, 1, R)$
   $\delta(q_0, \sqcup) = (q_1, \sqcup, R)$

2. move right to the end of 2nd block n flip bits

   $\delta(q_1, 0) = (q_1, 1, R)$
   $\delta(q_1, 1) = (q_1, 0, R)$
   $\delta(q_1, \sqcup) = (q_2, \sqcup, L)$

3. add one to get 2is complement.

   $\delta(q_2, 0) = (q_1, 1, L)$
   $\delta(q_2, 1) = (q_1, 0, L)$
   $\delta(q_2, \sqcup) = (q_2, \sqcup, L)$

4. move left to the start of 2nd block.
5. move left to the start of 1st block.
6. move right to the end of 1st block.
7. move right to the end of 2nd block.
8. subtract one from 2nd number in binary.

   $\delta(q_7, 0) = (q_7, 1, L)$
   $\delta(q_7, 1) = (q_{10}, 0, L)$
   $\delta(q_7, \sqcup) = (q_8, \sqcup, R)$

9. move left to the end of first block.

   $\delta(q_{10}, 0) = (q_{11}, 0, L)$
   $\delta(q_{10}, 1) = (q_{11}, 1, L)$
   $\delta(q_{10}, \sqcup) = (q_{11}, \sqcup, L)$

10. add one to the first no in binary.

    $\delta(q_{11}, 0) = (q_5, 1, R)$
    $\delta(q_{11}, 1) = (q_{11}, 0, L)$
    $\delta(q_{11}, \sqcup) = (q_{12}, \sqcup, R)$

11. clean up.

    $\delta(q_8, 1) = (q_8, R)$
    $\delta(q_8, \sqcup) = (q_9, \sqcup, H)$ (halt)

12. **Accept State:** $q_{accept} = \{q_9\}$

- **Initial State:** $q_0$
- **Acceptance States:** $q_{accept} = \{q_9\}$
- **Reject States:** $q_{reject} = \{\}$

**State Diagram for the Turing Machine**