# ESO207A: Data Structures and Algorithms
# Theoretical Assignment 1 Solution

Deadline: 17th Jan, 2018

January 31, 2018

**Instruction**

- All submission must be in pdf format.
- Please write precise answer, don't write unnecessary details.

**Problem 1.** (15 marks) Design ADT for the universe of $n$ (constant) digit natural number with operations (constant, succ, addition multiplication, etc.) to support arithmetics involving $n$ digit natural numbers using arrays and linked list ADTs. Define exceptions if any. Assume that you can not perform n digit operations on computer.

**Solution 1.**

- **Constant(string a) or representation**(2 Marks) store digits of given numbers in an array(or Linked-list)
- **Addition/subtraction (a,b)**(5 Marks) Do a digit wise addition(subs) with keeping track of carry.
  Exceptions : in addition overflow can happen while in subtraction answer can be a negative number.
- **Successor/predecessor(a)**(3 Marks) successor(a) = addition(a,constant(1)).
  predecessor(a) = subtraction(a,constant(1)).
  Exceptions : Same as previous one.
- **Multiplication(a,b)**(5 Marks) For every digit in b do multiplication with a and then add all of them after proper shifting of the individual results.
  Exceptions : overflow of the result similar to addition.

(-2) each for not writing addition and multiplication procedure. Since we can't add/multiply any n digit number directly.
(-1) for not writing exceptions.

**Problem 2.** (10 marks) Design ADT for following-

- **Queues** with all standard operations.
- **Sequence**, this can be sequence of any abstract data type. For example sequence of Stacks.

Think of what operations are needed for ADT and then provide short explanation about what those operation do. Writing pseudo code is not required but explanations needed to be precise.

**Solution 2.**

- **Queue**
  //Universe set of all queues of type t.
  $Create_t : \rightarrow Queue_t$ // create a new Queue
  $Empty_t : Queue_t \rightarrow Boolean$ //Returns 1 if Queue is empty
  $Dequeue_t : Queue_t \rightarrow t \times Queue_t$ // returns element from the front of the Queue and removes it from Queue
  $Enqueue_t : t \times Queue_t \rightarrow Queue_t$ // Appends the element at the back of the Queue.
  Along with above we may have functions for Front(which does not deletes first element), back etc. to enhance uses of the Queue.

- **Sequence**
  **Note :** Many of you seems to be confused sequences by sequences of queues but sequence is an ADT of any ADT as its element. Example: sequence of stacks, sequence of numbers(which you may have seen in mathematics) etc. However we are giving marks for your effort in this assignment since their was little confusion created due to PA1.
  Also few of the functions of Sequences will be dependent on the data type stored in Sequence. Example: for Queues we have En-queue and De-queue operations at a particular element of Sequence.

  //Universe set of all Sequence of type t.
  $Create_t : \rightarrow Seq_t$ // create a new Seq
  Apply some function on $i^{th}$ element of Seq, similar to En-Queue(i,j) where function in En-queue and operated on $i^{th}$ element of Seq.
  $Func_t : InputOfFunc \times i \times Seq_t \rightarrow Seq_t$ //apply function on $i^{th}$ element of Seq.
  Along with that we may have insert, delete, find function.
  $Insert_t : t \times i \times Seq_t \rightarrow Seq_t$ // inserts t after $i^{th}$ element
  $Delete_t : i \times Seq_t \rightarrow Seq_t$ // deletes $i^{th}$ element
  $Find_t : i \times Seq_t \rightarrow t$ // returns $i^{th}$ element of Seq.

**Problem 3.** (15 marks) Give a RAM Program for computing $n^k$, using squaring each time.
**A.**

```
READ 1
READ 2
LOAD=1
STORE 4
LOAD 2
JZERO END1
JGTZ A

END1 : WRITE=1
```

```
          HALT

A : DIV=2
    MULT=2
    STORE 3
    LOAD 2
    SUB 3
    JZERO E
    JUMP O

E : LOAD 2
    DIV=2
    STORE 2
    LOAD 1
    MULT 1
    STORE 1
    LOAD 2
    JUMP A

O : LOAD 2
    SUB=1
    JZERO END2
    LOAD 1
    MULTIPLY 4
    STORE 4
    LOAD 1
    MULTIPLY 1
    STORE 1
    LOAD 2
    SUB=1
    DIV=2
    STORE 2
    JUMP A

END2 : LOAD 1
       MULT 4
       WRITE 0
       HALT
```

**Problem 4.** (10 marks) Write a TM program for doubling of an input consisting of $k$ consecutive 1s. Replace the input with $2k$ consecutive 1s.

**A brief description of the program :**

There will be 3 Symbols - 0, 1, B (blank symbol)
Assume the current position is at the start of input (leftmost 1)
If start position has B, then HALT (no input), else do the following

1. Replace 1 with 0, move left till a $B$ is encountered
2. Replace it with a 1, move right until you are at a (the only) 0
3. Replace this zero with a 1, move right
4. If current symbol is $B$, then $HALT$
5. Else go to step 1

For Example - Input = 1111

$$(1)111 \rightarrow (B)0111 \rightarrow 1(0)111 \rightarrow 11(1)11 \rightarrow 1(1)011 \rightarrow (1)1011 \rightarrow (B)11011 \rightarrow$$

$$1(1)1011 \rightarrow 11(1)011 \rightarrow 111(0)11 \rightarrow 1111(1)1 \rightarrow 111(1)01 \rightarrow \cdots \rightarrow$$

$$(B)111101 \rightarrow 1(1)11101 \rightarrow \cdots \rightarrow 111111(1) \rightarrow 11111(1)0 \rightarrow \cdots \rightarrow 11111111$$

The state diagram is give in Figure 1 and the program is given in Table 1.

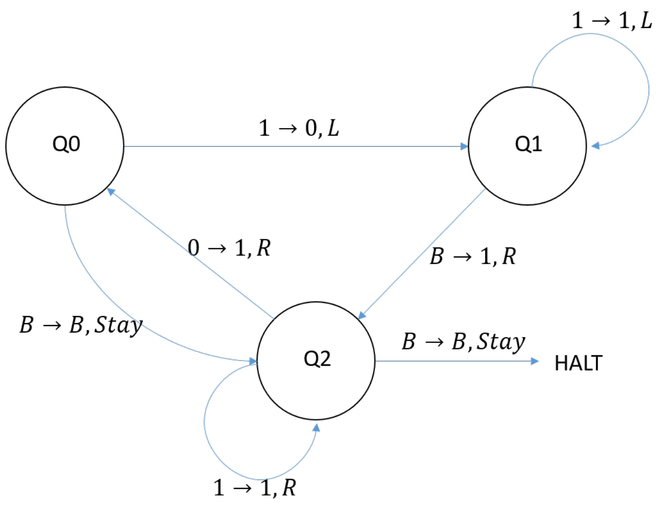

Figure 1: **Q4: State Diagram**

| **State** | **Symbol Read** | **Symbol to Write** | **Move** | **Next State** |
|---|---|---|---|---|
| $Q_0$ | 1 | 0 | L | $Q_1$ |
| $Q_0$ | B | - | - | HALT |
| $Q_1$ | 1 | 1 | L | $Q_1$ |
| $Q_1$ | B | 1 | R | $Q_2$ |
| $Q_2$ | 1 | 1 | R | $Q_2$ |
| $Q_2$ | 0 | 1 | R | $Q_0$ |
| $Q_2$ | B | - | - | HALT |

Table 1: **Q4: Turing Machine Program to double the number of input ones**