MATH230: Tutorial Nine

Introduction to Turing Machines

Key ideas and learning outcomes

- Know what is required to specify a Turing machine
- Give high-level (English) descriptions of Turing machines
- Comment your code
- Write example Turing machines
- Learn some "admin" tasks to help with other Turing machines

Make sure the Python code works for you before you leave this tutorial.

Relevant lectures: Turing Machine lecture slides.
Relevant reading: The Annotated Turing, *Petzold*.

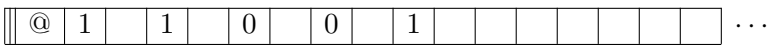
Hand in exercises: 1, 6, 8
Due following Friday @ 5pm submit .txt files online.

Discussion Questions

1. Turing describes a tape with E-squares (for working; "liable to erasure") and F-squares for holding the content of the calculation. Turing assumes F-squares are never blank and contain either a 0 or 1.

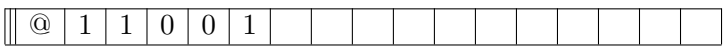
   Assume the input is a binary string with a blank E-square between each bit.

   Input:

   Write a Turing machine which has the input binary string next to the home (@) square without the blank E-squares between the bits.

   Output:

2. Write this Turing machine into a .txt file that can be tested with the script. Make sure you know how this script works by the end of the tutorial.

Tutorial Exercises

1. Print "Kia Ora, Ao!" on a blank tape.

   Input: a blank tape.

   Output: the characters on individual cells followed by blank cells.

2. Mark the end of a finite binary string.

   Input: Assume tape has a non-empty binary block at the start.

   Output: Input string with a symbol to mark the first blank cell after it.

   Test input: @,1,1,1,0,1,1,b,b,b Test output: @,1,1,1,0,1,1,x,b,b

3. Remove any blank cells at the start of the tape.

   Input: Assume tape has some finite binary block of cells.

   Output: Input tape without the blank cells at the start.

   Test input: @,b,b,b,b,b,1,0,1,0,b,b,b,b,b Test output:@,1,0,1,0,b,b,b,b,b

4. Flip all bits i.e. 1s to 0s and 0s to 1s.

   Input: Assume tape has a non-empty finite binary block at the start.

   Output: Binary string with all input bits flipped.

   Test input: @,1,1,1,1,1,0,1 Test output: @,0,0,0,0,0,1,0

5. Blank out the tape.

   Input: Assume tape has a non-empty binary block at the start.

   Output: Blank tape.

   Test input: @,1,1,1,0,1,1,b,b,b Test output: @,b,b,b,b,b,b,b,b,b

6. Reverse a binary string.

   Input: Assume tape has a non-empty binary block at the start.

   Output: Input string reversed.

   Test input: @,1,1,1,0,1,1,0,1 Test output: @,1,0,1,1,0,1,1,1

7. Separate characters with blank cells.

   Input: Assume tape has a non-empty binary block at the start.

   Ouput: Input string with all bits spaced by a blank cell.

   Test input: @,1,1,1,0, Test output: @,1,b,1,b,1,b,0,b,

8. Copy a binary string.

   Input: Assume tape has a non-empty binary block at the start.

   Ouput: Two copies of the input string together.

   Test input: @,1,1,1,0,b,b, Test output: @,1,1,1,0,1,1,1,0,b,b,b,