Contents

1 The Church-Turing Thesis

1.1 Turing machines

1.1.1 Introduction

TM is a **much more powerful model**, it can do everything that a general computer can do.

Nonetheless, even a Turing machine cannot solve certain problems. These problems are beyond the theoretical limit of computation.

- Uses an **infinite tape** as its unlimited memory. It has a tape head that can read and write symbols and move around on the tape.

Initially the tape contains only the **input string**, and is blank everywhere else.

If the machine needs to store some information, it may **write information** on the tape.

To read the information *it has written*, the machine can move its head back over it.

The machine continues computing until it decides to produce an output. The outputs *accept* and *reject* are obtained by entering designated accepting and rejecting states.

1.1.2 Differences between FA & TM

1. A tm can both read and write the tape.
2. The read-write head can move both to the left and to the right.
3. The tape is **infinite**

4. The special states for rejecting and accepting take dffect immediately.

1.1.3 A example: TM to test membership of {w#w | w ∈ {0,1}*}

The langauge B means a string comprises two identical strings separated by a # symbol.

Imagine you are on a mile long road with inputs on the ground. How to determine the input is in B? **Zig-zag** to the correspinding palces on the two sides of the # and determine whether they match, is a obvious strategy.

And we design a machine to work in that way. M_1 = "On input string w

1. Zig-zag across the tape to correspinding positions on either side of the # symbol to check whether these positions contain the same symbol. If they do not, or if no # is found, `reject`. Cross off symbols as they are checked to keep track of which symbols correspond.

2. When all symbols to the left of the `#` have been crossed off, check for any remaining symbols to the right of the `#`, if any symbols remain, *reject*; otherwise, *accept*.

"

1.1.4 Formal Definition of a Turing Machine

The heart: transition function σ For a Turing machine, σ takes the form: Q X γ -> Q X γ X {L,R}. That is, when the machine is in a certain state q and the head is over a tape square containing a symbol a, and if σ(q,a) = (r,b,L) -> The machine writes the symbol b replacing the a, and goes to state r. The third componnent is either L or R and indicates whether the head moves to the ledt or right after writing. In this case, the L indicates a move to the left.

A Turing machine is a 7-tuple, (Q, , , , q 0 , q accept , q reject), where Q, , are all nite sets and

1. Q is the set of states,

2. is the input alphabet not containing the blank symbol

3. is the tape alphabet, where ␣ and ,

4. : Q × →Q × × {L, R} is the transition function,

5. q 0 Q is the start state,

6. q accept Q is the accept state, and
7. q reject Q is the reject state, where q reject = q accept .

1.1.5 More on the definition: