CMSC 141 Automata and Language Theory

Turing Machines

Mark Froilan B. Tandoc

October 17, 2014

Turing Machines

The simplest machines that
can get the job done

Turing Machines

- A finite-state automaton that uses its tape as a form of storage (the tape is re-writeable)
- The tape is infinitely long giving a TM an infinite amount of memory
- The read/write head is bi-directional - the head can move left, right, or stay in place after each step
- Despite its simplicity, it can do anything a real computer can do - hence, it captures the very essence of algorithms and computability

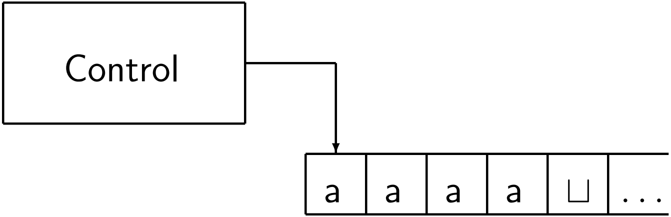
Turing Machines

There are two types of Turing machines

- **Decider TM** - for an input string x and a language L, determine if x is a member of L
- **Transducer TM** - for an input string x and a string function f, compute $f(x)$, and leave the resulting string $f(x)$ on the tape

Turing Machines

- Initially, the tape only contains the input and blanks everywhere else
- The machine continues the computation until it produces an output
- The outputs *accept* and *reject* are obtained by entering designated accepting and rejecting states. Otherwise, it will go on forever, never halting.



FA vs TM

The following list summarizes the differences between finite automata and Turing machines

- A Turing machine can both write on the tape and read from it
- The read-write head can move both to the left and to the right
- The tape is infinite
- The special states for rejecting and accepting take effect immediately

Example

- A Turing machine M for testing membership in the language $L = \{w\#w | w \in \{0,1\}^*\}$
- M will accept the input if it is a member of L and reject otherwise (Decider TM)
- The language describes two identical strings separated by a $\#$

Idea

- To *zig-zag* to the corresponding places on the two sides of the $\#$ and determine whether they match
- Place marks on the tape to keep track of the symbols checked

Example

Snapshots of *M* computing on input 011000#011000

```
⇓
011000#011000⊔...
 ⇓
x11000#011000⊔...
       ⇓
x11000#011000⊔...
 ⇓
x11000#x11000⊔...
  ⇓
xx1000#x11000⊔...
             ⇓
xxxxxx#xxxxxx⊔... accept
```

References

- Previous slides on CMSC 141
- M. Sipser. Introduction to the Theory of Computation. Thomson, 2007.
- J.E. Hopcroft, R. Motwani and J.D. Ullman. Introduction to Automata Theory, Languages and Computation. 2nd ed, Addison-Wesley, 2001.
- E.A. Albacea. Automata, Formal Languages and Computations, UPLB Foundation, Inc. 2005
- JFLAP, www.jflap.org
- Various online LaTeX and Beamer tutorials