

COS 126 Precept

April 16, 2009

Agenda

- ▶ Turing Machines
- ▶ Universality
- ▶ Computability
- ▶ Return TSP
- ▶ Guitar Hero



Turing Machines

- ▶ **Assignment 8 part 2**
 - ▶ Design a touring machine that accepts a sequence of well balanced parenthesis and angle brackets
- ▶ **Running the touring machine**
 - ▶ `java -jar turing.jar`
 - ▶ File -> Load Machine
- ▶ **Matching parenthesis example**

Syntax of .tur File

5 sections (title, description, vertices, edges, tapes)

Title

Parenthesis Matcher

Description (Including name/login/precept)

Accepts the input if it is a well-balanced sequence of parentheses.

Vertices

0 R 0.5 0.25

1 L 0.9 0.25

2 L 0.5 0.5

3 N 0.5 0.9

4 Y 0.65 0.5

tapes

[<] >

[()] (< >))

[()] ()) ((())) ()

[()] ((())

[()] (()))

[()])) (() () [] ()))

Edges

0 1) X 0.2

1 0 (X 0.2

0 2 # #

1 3 # #

2 3 ((

2 4 # #

Universality Booksite 7.5 Harel 1-2

▶ Church-Turing Thesis

- ▶ Turing machines can do any computation that can be done by any real computer.
- ▶ Church's lambda calculus is another equivalent language example.
- ▶ *(Under the current model of computation)

Computability Booksite 7.6 Harel 2

- ▶ There exist unsolvable problems
- ▶ True or False?
 - ▶ A Universal Turing Machine can compute anything that any other Turing Machine could possibly compute
 - ▶ The undecidability of the halting problem is a statement about Turing machines: it is not applicable to real computers
 - ▶ The Turing machine is a universal model of computation: with a Turing machine we can solve any decision problem that can be solved with a DFA or with a Pentium M running Linux.

Return TSP