# Theory of Computation
# Tutorial V

Speaker: Yu-Han Lyu

December 1, 2006

# Journal

- Journal of the ACM (JACM)
- Theoretical Computer Science (TCS)
- Electronic Colloquium on Computational Complexity (ECCC)
- Theory of Computing (ToC)

# Conference

- ACM Symposium on Theory of Computing (STOC)
- ACM/SIAM Symposium on Discrete Algorithms (SODA)
- IEEE Symposium on Foundations of Computer Science (FOCS)
- IEEE Conference on Computational Complexity (CCC)

# Theory of Computing

- Publishers of scientific journals today actually impede the flow of information rather than enable it. *-- Jeff Ullman*
- Knuth sent letter to Journal of Algorithms On October 25, 2003
  - ACM Transactions of Algorithms
- http://theoryofcomputing.org/

# Optimization is Impossible!

- Given a program and input, compiler want to do extreme optimization

  ➔ If program has redundant computation, then compiler must eliminate it
- If program has infinite loop, then the result must be
  - L: goto L;
- It decides $HALT_{TM}$

# Oracle Turing Machine

- An oracle Turing machine is a modified Turing machine that has the additional capability of querying an oracle
- $T^A$ to describe an oracle Turing machine that has an oracle for language A.
- If $T^A$ can decide B, then A is Turing reducible to B
- There are still some problem are not decidable by $T^{ATM}$

# Reducibility

- Mapping Reducibility
  - Many-one reducibility
  - $A \leqq_m B$
- Turing Reducibility
  - Oracle Turing Machine
  - $A \leqq_T B$
- Mapping reducibility is special case of Turing reducibility.
- What is the difference?

# Turing vs Many-one Reducibility

| $A \leqq_X B$ | Turing | Many-one |
|---|---|---|
| B is decidable | A is decidable | A is decidable |
| B is recognizable | X | A is recognizable |
| A is undecidable | B is undecidable | B is undecidable |
| A is not recognizable | X | B is not recognizable |

# Self-Reproduce Program

- main(a){printf(a="main(a){printf(a=%c%s%c,34,a,34);}",34,a,34);}
- Quine
- We can get <M> in M
- Assume H is decider for $A_{TM}$
- B="On input w:
  - Obtain <B>, via the recursion theorem
  - Run H on input <B,w>
  - H accept, rejects. Otherwise, accepts."

# HW3 – Problem 1

- Show that single-tape TMs that cannot write on the portion of the tape containing the input string recognize only regular languages.
- Myhill-Nerode Theorem
  - L is regular **if and only** if it has finite index
- For any string x, we can find a function
  - $f_x(\{start\} \cup Q) \rightarrow \{accept, reject\} \cup Q$
- Any string with the same function are indistinguishable

# HW3 – Problem 2

- Decider is not recognizable
- Suppose a enumerator E that enumerate all decider $<M_1>$, $<M_2>$….
- D=”On input w:

  1. Let i be the index of $\Sigma^*$, that is $s_i = w$.
  2. Run $<M_i>$ on input w.
  3. If $M_i$ accepts, reject. Otherwise, accept.”

# HW3 – Problem 3

- Let $PAL_{DFA}$={<M>| M is a DFA that accepts some string with more 1s than 0s}. Show that $PAL_{DFA}$ is decidable
- Let CFL A={x | x has more 1s than 0s}
- T="On input <M> where M is a DFA:

    1. Let B = A $\cap$ L(M), so B is CFL.
    2. Test whether B is empty.
    3. If B is empty, reject. Otherwise, accept."

# HW3 – Problem 4

- Let C be a language. Prove that C is Turing-recognizable if and only if a decidable language D exists such that $C=\{x \mid \exists_y (\langle x,y\rangle \in D)\}$
- If D exists
  - search each possible string y, and testing whether $\langle x,y\rangle \in D$
- If C is recognizable
  - $D=\{\langle x,y\rangle \mid M \text{ accepts } x \text{ within } |y| \text{ steps}\}$

# Homework 4

- Due
  - 3:20 pm, December 15, 2006 (before class)

# HW4 - Problem 1

- Define the busy beaver function BB: N→N as follows.
  - For each value of k, consider all k-state TMs that halt when started with a blank tape.
  - Let BB(k) be the maximum number of 1s that remain on the tape among all of these machines.
  - Show that busy beaver function is not a computable function.
- Proof by contradiction

# HW4 - Problem 2

- Show that $AMBIG_{\mathrm{CFG}}$ is undecidable
- PCP Problem
- PCP $\leqq_m$ $\mathrm{AMBIG}_{\mathrm{CFG}}$

# HW4 - Problem 3

- ***Two-headed finite automaton*** (2DFA)
  - two read-only, bidirectional heads
- Show that $A_{\text{2DFA}}$ is decidable
- Show that $E_{\text{2DFA}}$ is not decidable
- Computation History

# HW4 - Problem 4

- Let $J = \{w \mid \text{either } w = 0x \text{ for some } x \in A_{TM}, \text{ or } w = 1y \text{ for some } y \notin A_{TM}\}$
- Show that neither *J* nor the complement of *J* is Turing-recognizable
- Mapping Reducibility

# HW4 - Problem 5

- Rice's theorem
- Prove that the problem of determining whether a given Turing machine's language has property *P* is undecidable.
- Let *P* be a language consisting of Turing machine descriptions where *P* fulfills two conditions.
  - *P* is nontrivial – it contains some, but not all, TM descriptions.
  - Second, *P* is a property of the TM's language – whenever $L(M_1)=L(M_2)$, we have $\langle M_1 \rangle \in P$ iff $\langle M_2 \rangle \in P$. Here, $M_1$ and $M_2$ are any TMs.

# HW3 - Problem 5

- Show that the problem of determining whether a CFG generates all string in 1* is decidable. In other words, show that {<G>| G is a CFG over {0,1} and $1^* \subseteq L(G)$} is a decidable language.
- Closure Property?
- Grammar?
- PDA?
- Parse tree?