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™
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
  – $A \leq_m B$

• Turing Reducibility
  – Oracle Turing Machine
  – $A \leq_T B$

• Mapping reducibility is special case of Turing reducibility.
• What is the difference?
# Turing vs Many-one Reducibility

<table>
<thead>
<tr>
<th>$A \leq_X B$</th>
<th>Turing</th>
<th>Many-one</th>
</tr>
</thead>
<tbody>
<tr>
<td>B is decidable</td>
<td>A is decidable</td>
<td>A is decidable</td>
</tr>
<tr>
<td>B is recognizable</td>
<td>X</td>
<td>A is recognizable</td>
</tr>
<tr>
<td>A is undecidable</td>
<td>B is undecidable</td>
<td>B is undecidable</td>
</tr>
<tr>
<td>A is not recognizable</td>
<td>X</td>
<td>B is not recognizable</td>
</tr>
</tbody>
</table>
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 $A_T^M$
• 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} ∪ Q)→{accept, reject} ∪ 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 \(\sum^*\), that is \(s_i = w\).
  2. Run \(<M_i>\) on input w.
  3. If \(M_i\) accepts, reject. Otherwise, accept."
HW3 – Problem 3

• Let $\text{PAL}_{\text{DFA}} = \{ <M> | \text{M is a DFA that accepts some string with more 1s than 0s}\}$. Show that $\text{PAL}_{\text{DFA}}$ is decidable

• Let CFL $A = \{ x \mid x \text{ has more 1s than 0s}\}$

• $T =$ ”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 $C = \{x | \exists y (<x,y> \in D)\}$

• If $D$ exists
  – search each possible string $y$, and testing whether $<x,y> \in D$

• If $C$ is recognizable
  – $D = \{<x,y>|M \text{ accepts } x \text{ within } |y| \text{ steps}\}$
Homework 4

• Due
  – 3:20 pm, December 15, 2006 (before class)
HW4 - Problem 1

• Define the busy beaver function $BB: \mathbb{N} \rightarrow \mathbb{N}$ 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 $\text{AMBIG}_{\text{CFG}}$ is undecidable
• PCP Problem
• PCP $\leq_m \text{AMBIG}_{\text{CFG}}$
HW4 - Problem 3

- **Two-headed finite automaton** (2DFA)
  - two read-only, bidirectional heads
- Show that $A_{2DFA}$ is decidable
- Show that $E_{2DFA}$ is not decidable
- Computation History
HW4 - Problem 4

• Let $J = \{w \mid \text{either } w = 0x \text{ for some } x \in A_{TM},$
  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 $<M_1> \in P$ iff $<M_2> \in P$. 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 $\{<G>| G \text{ is a CFG over } \{0,1\} \text{ and } 1^* \subseteq L(G)\}$ is a decidable language.

• Closure Property?
• Grammar?
• PDA?
• Parse tree?