CS5371
Theory of Computation

Last (but not least) Lecture:
The Revision

What have we learnt?

- Various mathematical models (such as DFA/NFA/PDA) that can perform string decision problems
- Various mathematical expressions (such as RE/CFG) that can generate strings
- DFA = NFA = RE (regular languages)
- PDA = CFG (context-free languages)

What have we learnt?

- How to show a language is regular
  - Give a DFA/NFA, Write a RE
  - E.g., A = { even-length string ending with 00 }

- How to show a language not regular
  - Pumping Lemma (pumping length p, xy^kz)
  - E.g., B = { palindrome }
  - E.g., C = { 0^x1^y | x < y }

What have we learnt?

- How to show a language context-free
  - Give a PDA, Write a CFG
  - E.g., D = { palindrome }
- How to show a language not context-free
  - Pumping Lemma (pumping length p, uv^kxy^kz)
  - E.g., E = { $0^x1^x2^x$ }
  - E.g., F = { ww }

What have we learnt?

- Stronger mathematical models (such as DTM/NTM/Enumerator) that can solve more string decision problems
- DTM = NTM (in deciding/recognizing power)
- DTM = Enumerator (in recognizing power)
- Decidable Language
  - A_{DFA}, E_{DFA}, EQ_{DFA}, A_{CFG}, E_{CFG}, ...
- Recognizable Language
  - A_{TM}, $HALT_{TM}$, ...

What have we learnt?

- How to show a language recognizable
  - Give a TM recognizer
  - Finite steps to Accept
  - May Loop if Not Accept
- How to show a language decidable
  - Give a TM decider
  - Finite steps to Accept and to Reject

What have we learnt?

- How to show a language undecidable
  - Diagonalization Proof (E.g., A_{TM})
  - Reduction Proof
  - E.g., $HALT_{TM}$, E_{TM}, Rice, EQ_{TM}, Post, E_{LBA}, ...
- Some language and its complement are both non-recognizable
  - Mapping Reduction Proof
  - E.g., EQ_{TM} (by Reduction from A_{TM}')

What have we learnt?

- Decidable = Deciding in Finite Steps
- Finite is TOO LARGE
  - Measuring Time Complexities

- Relationship among models
- 1-tape DTM vs k-tape DTM
- DTM vs NTM

What have we learnt?

- P = Deciding in Polynomial Time by DTM
  - E.g, PATH, RELPRIME
- NP = Verifying in Polynomial Time by DTM
  = Deciding in Polynomial Time by NTM
  - E.g., SAT, COMPOSITES, HAMPATH, …
- Some problems in NP are the hardest (NP-complete)
  - E.g., SAT (Cook-Levin), 3SAT, CLIQUE, …

What have we learnt?

- How to show a language is in NP
  - Give a DTM verifier, or
  - Give an NTM decider
  - Show that running time is polynomial (in terms of input length)
- How to show a language NP-complete
  - Show that it is in NP
  - Show that every NP problem can be reduced to it in polynomial time
  - Polynomial Time (Mapping) Reduction Proof

About the Exam

- Jan 11, 2008 (next Friday) Don't Forget!!!
- Venue: This Room
- Time: 3:20pm – 6:20pm
- Format:
  - Around 7 Questions
  - Easy to Moderately Difficult
  - Most from Notes/HW, Some Unseen

- Tentative marks will be sent by email within one week
- Finalized after that

Acknowledgement

- Thanks all of you for choosing this course
- Special thanks to those who have sent me comments and suggestions, and those who have observed the bugs in the Notes/HW
- Thanks Shao-Chia (紹甲) for being a very responsible tutor
- The textbook is also wonderful
- Advertisement: Algo, Randomized Algo

GOOD LUCK in the EXAM