Fine-Grained Complexity and Algorithms
Lecture 01, Sept. 1, 2015
Russell Impagliazzo, russell@cs.ucsd.edu

Logistics: Office Hours
Location: 344 Calvin Lab.
Time: Stop by anytime, or email to make an appt.
      to ensure he is available.

Note to the reader: If the notes are unclear, ambiguous,
or contain errors, check with Professor Impagliazzo.
They may simply be board-copying errors.

Course Overview
1. Main Ideas
   1. Algorithm Design
      (called upper bounds by Complexity Theorists
      == Algorithm lower bounds)
   2. Use Multidimensional Analysis (to get more information
      about true difficulty of types of instances)
   3. Use Reductions that preserve complexity at a
      finer grain, not just qualitative gaps i.e.
      “easy” vs. “very hard,” but quantitative statements
      — hopefully a result of this course

Goal
• Introduce enough background such that even a first year
  PhD student would be able to participate in the
  Simons Institute Workshop series
• assuming as little background as possible
(i.e. > 1 strong undergraduate complexity & algorithms course)
• to take the course for credit, students must serve as a lecture scribe at least once

(Resuming discussion of the 3 Main Ideas)
How do these 3 topics link?

There exists a link between Exact complexity of NP-Completeness and Meta-algorithmic Problems

Def. Meta-algorithmic Problem: problems whose inputs and/or outputs are representations of algorithms.
Examples: the Halting Problem, Circuit-SAT, Program Verification, Circuit Minimization,...

(Closer Look:) Circuit-SAT
“...we code our problem as a circuit”

Boolean Circuit Definition:
Inputs: Bits \( x_i \in \{0, 1\} \), first \( n \) gates \( \leftarrow \) inputs
Gates perform boolean operations:
\[ g_k = O_{k}(g_i, g_j), \quad i, j < k \]
\( g_m \) is the output gate, where \( m \) is the largest \( O_k \in \{\text{AND}, \text{OR}, ...\} \).

Size \( (f') = \min_m \exists m \text{ gate circuit } C \]
Let \( \forall x_1, \ldots, x_n \quad C(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \)
(Boolean Circuit Definition continued...)

Non-uniform: different circuit for each input length, no guarantee they are related

Reardon & Shannon: \( \forall n \exists f \) on \( n \) bit inputs s.t.
\[
\text{Size}(f) = \Omega \left( \frac{2^n}{n} \right)
\]

- For any function we know how to construct, the best lower bound is \( 3^n \)

More Restrictive Circuits: CNF/DNF
  1. Conjunctive Normal Form (CNF)
  2. Disjunctive Normal Form (DNF)

  \[
  \text{def. literal: } x_i \oplus \overline{x}_i \\
  \text{def. clause: } \bigvee (\text{OR}) \text{ of literals} \\
  \text{def. term: } \bigwedge (\text{AND}) \text{ of literals}
  \]

  CNF formula: \( \bigwedge \) of clauses
  DNF formula: \( \bigvee \) of terms

\[\text{Exercise: Prove that for } F(x_1, \ldots, x_n) = (\sum x_i) \mod 2\]
that the number of terms in any DNF \( \geq \Omega \left( 2^n \right) \)
and that the number of clauses in any DNF \( \geq \Omega \left( 2^n \right) \)

Circuit-SAT

\[
\text{def. SAT: satisfiability of CNF formulation} \\
\text{def. \( k \)-SAT: """" whose clauses are all } \leq k \text{ in size}
\]
K-SAT is NP-Complete \( \forall k \geq 3 \)

(Circuits & Circuit-SAT Continued)

Depth of unbounded Fan-in Circuit:

\[
\text{leAC}^0 \quad O(\log^* n)
\]

- PRAM constant time algorithms

\[d:\]

\[x_i, \overline{x_i}\]

What do we know about constant depth circuit functions?

\[
f(x_1, \ldots, x_n) = (x_1 \oplus \ldots \oplus x_n)
\]

\[
\text{Size}_d(f) \geq \exp \left( \Omega \left( n^{1/(d-1)} \right) \right)
\]

- proof for \( d > 2 \) is non-trivial

**K-\#SAT Problem**

def. Given a K-CNF \( \varphi \), count the number

\[
x_1 \cdots x_n \quad \varphi(x_1, \ldots, x_n) = 1
\]

Best known algorithms for K-SAT run in time

\[
2^{n(1-1/k)} \quad \text{(Schoning, PPSZ, IMP algorithms)}
\]

- different techniques, same running time

(IMP also works for K-\#SAT, we will go over it.)
Random Restriction Method
\[ \rho: x_1, \ldots, x_n \rightarrow \{0, 1, \ast^3\} \]
\[ f \uparrow \rho \]

Question: What happens to the parity of \( f \) if we restrict \( \rho \) in any way?

Höstad Switching Lemma

Let \( \varphi \) be a \( k \)-CNF.

**def.** decision tree: a binary tree where internal vertices are labeled by variable names

- left = assigning var. to 0
- right = assigning var. to 1

leaves are labeled by output vertices \((0, 1)\)

\[ \quad \text{e.g.} \]

\[ \begin{array}{c}
\text{y} \\
\downarrow \\
0 \\
\downarrow \\
0 \\
\downarrow \\
0 \\
\downarrow \\
0 \\
\downarrow \\
1 \\
\downarrow \\
1 \\
\end{array} \]

The decision tree depth of a decision function \( f \) is the tree depth.

If D.T. has depth \( D \) then \( f \) can be written as a \( D \)-CNF (take the AND of the path) or a
D-DNF (take the OR of the path).

Canonical D.T. for CNF

Query all variables in the first unsatisfied clause.
Substitute and simplify.
If any clause becomes 0, stop and output 0.
If all clauses are satisfied, stop and output 1.

Höfted Switching Lemma
("Swiching" comes from switching between CNF & DNF)
Let \( \rho \) be a K-CNF.
Let \( \rho \) be a random restriction leaving \((m)\) variables unset. Then, \[ \text{Prob}[\text{depth}(DT(\rho)) \leq (4e \cdot 1_p \cdot K)^D] \]
every iteration, \[ \frac{\ln(n)}{\log(n)} > 1, \frac{\ln(1/c)}{\log(n)} > n, c > \exp(n^2) \]
(Proof deferred to next class)

Intuition: if \( p > 1 > K \), then each clause has a reasonable chance of contributing to the decision...
\[ p_{K-CNF} \]
\[ p = \frac{1}{8ek} \text{ or some constant in } K \]

\[
\begin{array}{ccc}
(1-p)n & & pm \\
\end{array}
\]

- Divide the inputs at random into chunks of these sizes

Instead of looking at SAT directly, CDT \( \mathcal{R}_p \)
- Probabilistic algorithm

Next class: proof and applications