Great Ideas in Theoretical CS

Lecture 16:
NP I: Poly-Time Reductions

Anil Ada
Ariel Procaccia (this time)
$k$-COLORING

• Reminder: a $k$-coloring of a graph satisfies:
  o Each node has a color
  o There are at most $k$ different colors
  o Every two nodes connected by an edge have different colors

• A graph is $k$-colorable iff it has a $k$-coloring
2-COLORING

• Is this graph 2-colorable?
2-COLORING

• Given a graph $G$, how can we decide if it is 2-colorable?

• Enumerate all possible $2^n$ colorings to look for a valid one...

• OK, but how can we **efficiently** decide if $G$ is 2-colorable?
  - In polynomial time in the number of vertices $n$
2-COLORING

• Poll 1: $G = (V, E)$ is 2-colorable iff:
  1. $G$ has a Hamiltonian cycle
  2. $|E| \leq |V| - 1$
  3. Every vertex in $G$ has even degree
  4. $G$ has no odd cycles
2-COLORING

- Algorithm (reminder):
  - Choose an arbitrary node, color it red and its neighbors blue
  - Color the uncolored neighbors of the blue vertices red, etc.
  - If $G$ is not connected, repeat for every component
3-COLORING

• Is this graph 3-colorable?
3-COLORABILITY ORACLES

• We can decide 3-colorability by trying all $3^n$ possible colorings
• Let’s say we can ask an oracle...

3-colorability decision oracle
3-COLORABILITY ORACLES

- How do we turn a decision oracle into a search oracle?

NO / YES, here’s how

3-colorability search oracle
3-COLORABILITY ORACLES

What if I gave the oracle partial colorings of $G$? For each partial coloring of $G$, I could pick an uncolored node and try different colors on it until the oracle says “YES”. I would then have a larger partial coloring.

The oracle doesn’t accept partial colorings!
3-COLORABILITY ORACLES

Given: 3-colorability decision oracle

YES
3-COLORABILITY ORACLES

Given:
3-colorability decision oracle

YES
3-COLORABILITY ORACLES

Given:
3-colorability decision oracle

NO
3-COLORABILITY ORACLES

Given:
3-colorability decision oracle

YES
A 3-colorability search oracle can be simulated using a linear number of calls to a decision oracle!
CLIQUE

• Reminder: A $k$-clique is a set of $k$ nodes with all possible edges between them

  1-clique  2-clique  4-clique

• **CLIQUE**: Given a graph $G$ and $k \in \mathbb{N}$, does $G$ contain a $k$-clique?
INDEPENDENT SET

• A *k*-independent set is a set of *k* nodes with no edges between them

\[ G_k \subseteq \mathbb{N} \]

• **INDEPENDENT-SET**: Given a graph \( G \) and \( k \in \mathbb{N} \), does \( G \) contain a *k*-independent set?
**CLIQUE vs. IS**

- Let $G^* = (V, E^*)$ be the complement of $G = (V, E)$
  
  $(u, v) \in E \iff (u, v) \notin E^*$

- Poll 2: $G$ has a $k$-clique for $k \geq 2$ iff:
  1. $G^*$ has an IS of size $k$
  2. $G^*$ has an IS of size $2k$
  3. $G^*$ has an IS of size $k^2$
  4. $G^*$ has an IS of size $n = |V|$
CLIQUE vs. IS

\[ \langle G, k \rangle \rightarrow \text{YES/NO} \]

Construct:
**INDEP.-SET**
decision oracle

\[ \langle G^*, k \rangle \rightarrow \text{YES/NO} \]

Given:
**CLIQUE**
decision oracle
Clique vs. IS

\[ \langle G, k \rangle \quad \text{YES/NO} \]

Construct: Clique oracle

\[ \langle G^*, k \rangle \quad \text{YES/NO} \]

Given: INDEP.-SET oracle
**CLIQUE vs. IS**

- We can quickly reduce an instance of CLIQUE to an instance of INDEPENDENT-SET, and vice versa
- There is a fast method for one iff there is a fast method for the other

CLIQUE and INDEPENDENT-SET are cosmetically different but essentially the same!
Poly-Time Reductions

- $L$ has a polynomial-time reduction to $L'$, denoted $L \leq^P_T L'$, if and only if it is possible to solve $L$ in polynomial time using a polynomial-time algorithm for $L'$.

- If $L \leq^P_T L'$ then:
  1. $L' \in \text{P} \Rightarrow L \in \text{P}$
  2. $L \notin \text{P} \Rightarrow L' \notin \text{P}$
Circuit-Sat

- AND, OR, NOT gates wired together
- **Circuit-Satisfiability:** Given a circuit with $n$ inputs and one output, is there a way to assign 0/1 values to the input wires so that the output value is 1 (true)?
3-COLORABILITY VS. CIRCUIT-SAT

Fundamentally different problems?
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>f(x, y)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>f(x, y)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[
f(x, y) = f(x, y)
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>( f(x, y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( f(x, y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

\[
\begin{array}{c|c|c}
 x & y & f(x, y) \\
 0 & 0 & 0 \\
 0 & 1 & 1 \\
 1 & 0 & 1 \\
 1 & 1 & 1 \\
\end{array}
\]
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( f(x, y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x, y) \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3-COLORABILITY VS. CIRCUIT-SAT

\[ f(x) \]

\[ \begin{array}{c|c}
  x & \text{NOT} \\ 
  \hline 
  0 & 1 \\
  1 & 0 \\
\end{array} \]
3-COLORABILITY VS. CIRCUIT-SAT

AND Gate from OR and NOT
3-COLORABILITY VS. CIRCUIT-SAT

\[ x \quad y \quad z \]

\[ \text{OR} \quad \text{NOT} \quad \text{OR} \]

\[ \begin{array}{c}
\text{OR} \\
\text{NOT} \\
\end{array} \]

\[ \begin{array}{c}
\text{OR} \\
\text{NOT} \\
\end{array} \]
3-COLORABILITY VS. CIRCUIT-SAT

Circuit is satisfiable
Graph is 3-colorable
3-COLORABILITY VS. CIRCUIT-SAT

Circuit

YES/NO

Construct: CIRCUIT-SAT decision oracle

Graph with subgraphs corresponding to gates and edge between output and false

YES/NO

Given: 3-COLORAB. decision oracle
3-COLORABILITY VS. CIRCUIT-SAT

• There is a polynomial-time reduction from CIRCUIT-SAT to 3-COLORABILITY

• Fact: Any of the four problems we discussed polynomial-time reduces to any of the others

But nobody knows how to efficiently solve any of these four problems in the worst case!
SUMMARY

• Terminology:
  o $k$-COLORING, CLIQUE, INDEPENDENT-SET, CIRCUIT-SAT
  o Polynomial-time reduction

• Principles:
  o Computationally efficient reductions between problems!