Clique is NP-complete and some related questions

This document discusses the **Clique** decision problem, shows that it is NP-completeby showing that 3SAT $\leq_p$ Clique

1 Discussion

We will be examing the Clique problem defined below. The description of the encoding used for the instance is in the Encoding section below.

In general we are interested in the complexity of solving problems. To get a feel for this problem, consider the following algorithm which finds a k-Clique in G if one exists

```
Clique(Graph G=(V,E), int k){
    for each V' (subset of V of size k) {
        check that all vertices in V' are connected to eachother
    }
}
```

This simple algorithm actually has terrible running time. The above algorithm has running time worse than n choose k since this is the number of k element subsets of V. If our graph had $10,000$ vertices and $k = 5,000$ the above algorithm would take a LONG time to complete!

2 Proofs

Theorem: Clique is NP-complete
Proof: Follows from the 2 theorems below.
Theorem: Clique $\in$ NP
Proof:
Clique $\in$ NP since

$$\text{Clique} = \{x \in \{0, 1, comma\}^* : \exists y(|y| \leq c|x|^d \text{ and } R(x, y))\}$$

where

1. **Describe R:** R accepts the pair (x, y) if x codes a pair (G, k) and y codes a subset of the vertices of G which form a k-clique in G. Otherwise R rejects.

2. **Show that R runs in polytime:** R operates by taking inputs x (encoding the pair $(G = (V, E), k)$) and y (encoding some $V' \subseteq V$) and

   (a) checks that $|V'| = k$

   (b) checks that all vertices in V' are connected to eachother

   R accepts if this is the case and rejects otherwise. From the description above, it should be clear that R can operate in polytime.

3. **Argue that $|y|$ is polynomial in $|x|$:** Finally, from the description of our encoding of subsets of V (see below), we can take $c = 1$ and $d = 1$. In this case, $|y| \leq |x|$.

Theorem: Clique is NP-hard

Proof: To show that Clique is NP-hard (by a Theorem we proved in class) it is enough to show that 3SAT $\leq_p$ Clique. Note that we are assuming that 3SAT is NP-complete. By definition, to show that 3SAT $\leq_p$ Clique we need to find an $f \in FP$ (polytime computable function) such that

$$(\forall x \in \{0,1\}^*)(x \in \text{3SAT} \Leftrightarrow f(x) \in \text{Clique})$$

Describe f: We describe f by saying what it does to an instance C of 3SAT.
For what follows, i and k are literal indexes (so $i, k \in \{1, \ldots, n\}$). j and l are clause indexes (so $j, l \in \{1, \ldots, m\}$).
Let $C = \{c_1, \ldots, c_m\}$ be a clause set over literals $x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}$ $f(C) = (G = (V, E), k)$ where

$$V = \{x_i^j : x_i \in c_j\} \cup \{\overline{x}_i^j : \overline{x}_i \in c_j\}$$

$$E = \{\{x_i^j, x_k^l\} : j \neq l\} \cup \{\{x_i^j, \overline{x}_k^l\} : j \neq l \text{ and } i \neq k\} \cup \{\{\overline{x}_i^j, \overline{x}_k^l\} : j \neq l\}$$

$$k = m$$

In other words, f maps the clause set C into the graph consisting of m pieces (one piece for each clause in C). Each piece has 3 new vertices, one vertex for each literal appearing in the clause. Vertex x_i^j in piece j corresponds to the literal x_i which appears in clause j. x_i^j is connected to all vertices in **other** components which are **not** labelled like $\overline{x}_i^l$. Similarly $\overline{x}_i^j$ is connected to all vertices in **other** components which are **not** labelled like x_i^l. The intuition is that there is an edge between 2 vertices if it is possible to simultaneously set the corresponding literals in C to $\top$.

Finally $f(C)$ sets $k = m$.

Argue that f is computable in polytime: It should be clear that f is computable in polytime. f is not doing anything complicated.

Show that f really works: Finally, we show that $x \in \text{3SAT} \Leftrightarrow f(x) \in \text{Clique}$
As above, we interpret x as $C = \{c_1, \ldots, c_m\}$ and $f(x) = f(C) = (G = (V, E), m)$.

$C \in \text{3SAT} \Rightarrow \exists \tau$ a truth value assignment, such that $C^\tau = \top$ That is, $c_j^\tau = \top$ for $j = 1, \ldots, m$. For each j, *blame* the truth of c_j on some literal $\mathcal{L}^j$ appearing in c_j. In this way we end up with m (not necessarily distinct) literals $\mathcal{L}^1, \ldots, \mathcal{L}^m$. For each j, the literal $\mathcal{L}^j$ is associated with 1 vertex in piece j of G, so the collection $\mathcal{L}^1, \ldots, \mathcal{L}^m$ corresponds to m distinct vertices, each in separate pieces of G. Now each of these m vertices is connected to eachother since τ is able to set them all to $\top$ simultaneously. So $(G, m) = f(x) \in$ Clique . That is, $x \in \text{3SAT} \Rightarrow f(x) \in \text{Clique}$

$(G = (V, E), m) = f(x) \in \text{Clique} \Rightarrow G$ has an m-clique V'. Note that each vertex in V' must come from a separate piece of G (see the definition of f). So V' consists of m vertices, exactly 1 vertex from each piece of G. Then V' looks like $\mathcal{L}^1, \ldots, \mathcal{L}^m$ where each $\mathcal{L}^j$ corresponds to some literal appearing in c_j. Note that by definition of f, no literal and its negation appears in V', so we are free to define τ as the truth assignment which sets all literals appearing in V' to $\top$. τ sets all other x_i to $\perp$ (it actually does not matter what we set the remaining variables to). Then $c_j^\tau = \top$

since $\mathcal{L}^j$ appears in c_j and τ sets this literal to $\top$. So $C^\tau = \top$, that is $C \in$ 3SAT. So we have $f(x) \in$ Clique $\Rightarrow x \in$ 3SAT.

Combining the two results, we have $x \in$ 3SAT $\Leftrightarrow f(x) \in$ Clique so that Clique is NP-hard.

Exercise:

1. Write up a clause set $C \in 3SAT$, draw $f(C)$. What is k in this case? Find a truth value assignment satisfying C and (using the proof above) find the corresponding k-clique in $f(C)$. Conclude that $f(C) \in$ Clique.

2. Write up a clause set C such that $f(C) \in$ Clique. Draw $f(C)$. What is k in this case? Find a k-clique in $f(C)$ and the associated truth value assignment τ (defined in the proof above) such that $C^\tau = \top$. Conclude that $C \in$ 3SAT.

3 Problem Definitions

We define Clique and 3SAT.

Clique

Input: A graph $G = (V, E)$, $k \in \mathbb{N}$

Question: Does G have a k-clique? That is, is there $V' \subseteq V$ such that $|V'| = k$ and every vertex in V' is connected to every other vertex in V'.

3SAT

Input: A set of clauses $C = \{c_1, \ldots, c_m\}$ over literals $x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}$ such that each c_i has size 3. C represents a propositional formulae in CNF (Conjunctive Normal Form).

Question: Is there some truth assignment τ such that $C^\tau = \top$? That is, is there a truth assignment τ such that

$$\wedge_{i=1}^{m} (\vee_{l \in c_i} l)^\tau = \top$$

Exercises:

1. Is the clause set $\{\{x_1, x_2, \overline{x_3}\}\}$ in 3SAT?

2. Is the clause set $\{\{x_1, x_2, \overline{x_3}\}, \{\overline{x_1}, x_4, \overline{x_3}\}, \{x_7, x_2, \overline{x_5}\}, \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}\}$ in 3SAT?

3. Come up with a clause set which is not in 3SAT and explain why this is the case.

4. Come up with an instance I_1 of the Clique problem such that $I_1 \notin$ Clique.

5. Come up with an instance I_2 of the Clique problem such that $I_2 \in$ Clique.

4 Encodings

We assume that a graph G is encoded as an adjacency matrix. That is an n^2 bit string of 0 and 1 representing the adjacency matrix of G. k is written in binary[1]. If we are representing a graph with n vertices and $k \leq n$ then this representation takes no more than $O(n^2)$ bits.

We will require an encoding for subsets of V. We will endcode $V' \subseteq V$ as an n-bit bit string such that the ith bit is 1 if and only if the ith vertex is in V'. If x encodes some instance $(G = (V, E), k)$ of Clique and y encodes $V' \subseteq V$ then $|y| \leq |x|$.

Exercises:

[1] In this case we could even write k in unary

1. Explain why $|y| \leq |x|$ above.