CPSC 511 — Fall 2014
Solutions for Question #2 on Midterm Test

In this question you were asked to consider the following ***decision problems***.

k-Clique

Instance:	An undirected graph $G = (V, E)$ and a positive integer k such that $k \leq \lvert V \rvert$
Question:	Does there exist a clique with size at least k in G?

Half-Clique

Instance:	An undirected graph $G = (V, E)$ such that $\lvert V \rvert = 2n$ for some positive integer n
Question:	Does there exist a clique with size at least $n = \lvert V \rvert/2$ in G?

You were asked to use the fact that the ***k-Clique*** problem is $\mathcal{NP}$-complete to prove that the ***Half-Clique*** problem is $\mathcal{NP}$-complete as well.

Assumptions about Encodings

A1. Instances of both problems are encoded over an alphabet Σ_G such that

$$\{\mathtt{v}, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathtt{(}, \mathtt{)}, \mathtt{,}\} \subseteq \Sigma_G$$

so vertices in a graph G can be ordered as

$$V = \{v_0, v_1, \ldots, v_{n-1}\}$$

where $n = \lvert V \rvert$, and vertices can be represented in a straightforward way. For example, the vertex v_i might be encoded using the letter `v` followed by an unpadded decimal representation of the index i. Suppose that encodings of vertices do not include the symbols `(`, `)`, or `,`.

A2. It is possible to decide whether a string $\omega \in \Sigma_G^\star$ is a valid encoding of some instance (G, k) of ***k*-Clique** deterministically, using time that is polynomial in the length $|\omega|$ of ω, in the worst case.

A3. It is possible to decide whether a string $\omega \in \Sigma_G^\star$ is a valid encoding of some instance G of ***Half-Clique*** deterministically, using time that is polynomial in the length $|\omega|$ of ω, in the worst case.

A4. If $\omega \in \Sigma^\star$ is a valid encoding of an instance of either ***k-Clique*** or ***Half-Clique***, that includes an undirected graph $G = (V, E)$, then the length of ω is

- at least linear in the number of vertices in G, and
- at most polynomial in the number of vertices in G.

A5. If ω is a valid encoding of an instance of ***Half-Clique*** — that is, an undirected graph $G = (V, E)$ including an even number of vertices — then every vertex $v \in V$ has an encoding as a string in $\Sigma_G^\star$ whose length is at most polynomial[1] in $|\omega|$.

A6. If ω is a valid encoding of an instance of ***Half-Clique*** — that is, an undirected graph $G = (V, E)$ — and μ is a string in $\Sigma_G^\star$ whose length is at most polynomial in the length of ω, then it is possible to decide whether μ encodes a vertex in V deterministically, using time that is polynomial in the length of ω as well.

A7. If ω is a valid encoding of an instance of either ***k-Clique*** or ***Half-Clique*** that includes an undirected graph $G = (V, E)$, then the decimal representation of the number $|V|$ of vertices in G can be computed deterministically from ω using time that is polynomial in the length of ω.

A8. If ω is a valid encoding of an instance of ***k-Clique*** then the decimal representation of the integer input k can also be computed deterministically from ω using time that is polynomial in the length of ω.

A9. If ω is a valid encoding of an instance of either ***k-Clique*** or ***Half-Clique*** that includes an undirected graph $G = (V, E)$, and $\mu, \nu \in \Sigma_G^\star$ are strings encoding vertices $u, v \in V$ respectively, then it is possible to use ω, μ and ν to do each of the following, deterministically, and using time that is polynomial in the length of ω:

- Check whether $u = v$.
- Check whether $(u, v) \in E$.

Note: It is certainly acceptable if some (or even most) of these assumptions are not mentioned in your answer! However, they all do (very probably) get used in a complete solution for this problem.

[1] Indeed, it is likely that every vertex has an encoding whose length is at most *logarithmic* in $|\omega|$, too, but the above assumption is really all that is needed here.

Proof of Membership in $\mathcal{NP}$

Suppose, now, that $\omega \in \Sigma_G^\star$ is an encoding of a Yes-instance of ***Half-Clique***. That is, ω encodes a graph $G = (V, E)$ such that G has a clique with size at least n, where $|V| = 2n$.

A ***certificate*** for ω will initially be defined to be an encoding of a set $S \subseteq V$ such that $|S| = n$ and S is a clique in G.

This might be encoded as a string over an alphabet Σ_C that includes the symbols listed in Assumption A1, above, so that vertices can also be represented as strings in $\Sigma_C^\star$.

In this case a certificate μ for ω might have the form

$$(\mu_1\,,\mu_2\,,\,\ldots\,,\mu_n)$$

where $\mu_1, \mu_2, \ldots \mu_n$ are strings in $\Sigma_C^\star$ that encode vertices $u_1, u_2, \ldots, u_n \in V$, respectively.

Note that, by assumptions A1, A4 and A5, it is possible to choose strings $\mu_1, \mu_2, \ldots, \mu_n$, in the above certificate, whose lengths are at most polynomial in the length of ω — so that there exist integer constants c_1 and c_2 such that every encoding ω of a Yes-instance of ***Half-Clique*** has a certificate with length at most $p(|\omega|)$, where $p : \mathbb{N} \to \mathbb{N}$ is the polynomial function such that $p(n) = c_1 n^{c_2}$ for every integer $n \geq 0$.

The definition of a ***certificate*** will now be slightly modified: A certificate for an encoding ω of a Yes-instance of ***Half-Clique*** is an encoding $\mu \in \Sigma_C^\star$ of a clique in G with size n (where $|V| = 2n$) such that $|\mu| \leq p(|\omega|)$.

Suppose now that # is a symbol that does not belong to either Σ_G or Σ_C, so that it can be used as a separator between an encoding of an instance of ***Half-Clique*** and a certificate for this. A ***verification algorithm*** for ***Half-Clique*** is an algorithm, taking a string $\zeta \in (\Sigma_G \cup \Sigma_C \cup \{\#\})^\star$ as input, that is as shown in Figure 1 on page 4.

It should be clear by an inspection of this algorithm that it accepts a string ζ if and only if ζ is an encoding of a Yes-instance of ***Half-Clique*** along with a certificate for this encoding, as this has been defined above. Thus this is a ***correct*** verification algorithm for ***Half-Clique***. It should also be clear that the above assumptions — including, in particular, assumptions A3, A4, A6 and A9 — can be used to show that this algorithm can be implemented as a deterministic algorithm (or Turing machine) that uses a number of steps that is polynomial in the length of the encoding ω of the Yes-instance that is part of the input string ζ in the worst case. Thus this is a ***polynomial time verification algorithm*** for ***Half-Clique***, as needed to show that ***Half-Clique*** $\in \mathcal{NP}$.

On input $\zeta \in (\Sigma_G \cup \Sigma_C \cup \{\#\})^\star$:

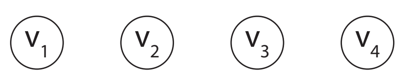
1. If $\zeta = \omega\#\mu$ where $\omega \in \Sigma_G^\star$, $|\mu| \leq p(|\omega|)$, and $\mu \in \Sigma_C^\star$, then go to step 2. Otherwise ***reject***.
2. If ω is a valid encoding of an instance of ***Half-Clique*** including an undirected graph $G = (V, E)$ such that $|V| = 2n$ for some integer $n \geq 0$ then go to step 3. Otherwise ***reject***.
3. If μ is an encoding of a set
$$S = \{u_1, u_2, \ldots, u_n\}$$
of n vertices in V then go to step 4. Otherwise ***reject***.
4. If $u_1, u_2, \ldots, u_n$ are distinct and $(u_i, u_j) \in E$ for all integers i and j such that $1 \leq i < j \leq n$ then ***accept***. Otherwise ***reject***.

Figure 1: A Polynomial-Time Verification Algorithm for ***Half-Clique***

Proof That *Half-Clique* is $\mathcal{NP}$-Hard

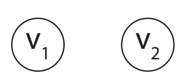
Consider a total function
$$f : \Sigma_G^\star \to \Sigma_G^\star$$
with the following properties.

- If a string $\omega \in \Sigma_G^\star$ is not a valid encoding of an instance of ***k-Clique***, then $f(\omega)$ is an encoding of an undirected graph $G = (V, E)$ with four vertices and no edges —



  — that is, an encoding of a small No-instance of ***Half-Clique***.

- If a string $\omega \in \Sigma_G^\star$ is a valid encoding of an instance of ***k-Clique*** including an integer input k such that either $k = 0$ or $k = 1$ — so that this is an encoding of a Yes-instance of ***k-Clique*** — then $f(\omega)$ is an encoding of an undirected graph $G = (V, E)$ with *two* vertices and no edges —



— that is, an encoding of a small Yes-instance of ***Half-Clique***.

- Otherwise ω is a valid encoding of an instance of ***k-Clique*** that includes an undirected graph $G = (V, E)$ and an integer k such that $2 \leq k \leq n = |V|$. In this case $f(\omega)$ is an encoding of a graph $\widehat{G} = (\widehat{V}, \widehat{E})$ that is as follows.
  - $\widehat{V} = V \cup \{v_{n+1}, v_{n+2}, \dots, v_{2n}\}$ where the vertices $v_{n+1}, v_{n+2}, \dots, v_{2n}$ are distinct and do not belong of V — so that $|\widehat{V}| = 2n = 2 \cdot |V|$.
  - $\widehat{E}$ includes
    - the edges in E
    - edges (v_{n+i}, v_{n+j}) for all integers i and j such that $1 \leq i < j \leq n - k$, and
    - edges (u, v_{n+i}) for each vertex $u \in V$ and for every integer i such that $1 \leq i \leq n - k$

    — and no others. Note that it follows that none of the vertices $v_{2n-k+1}, v_{2n-k+2}, \dots v_{2n}$ have any neighbours in $\widehat{G}$.

Claim: If ω encodes a Yes-instance (G, k) of ***k-Clique*** then $f(w)$ encodes a Yes-instance of ***Half-Clique***.

Proof: This is certainly true if $k = 0$ or $k = 1$. Suppose, instead, that (G, k) is a Yes-instance of ***k-Clique*** such that $k \geq 2$. Then there exists a subset S of V such that $|S| = k$ and S is a clique in G. It is easy to confirm that the set

$$S \cup \{v_{n+1}, v_{n+2}, \dots, v_{n-k}\}$$

is then a subset of $\widehat{V}$ with size $n = |\widehat{V}|/2$ that is a clique in $\widehat{G}$, as needed to show that $\widehat{G}$ is a Yes-instance of ***Half-Clique***.

Claim: If $f(\omega)$ is an encoding of a Yes-instance $\widehat{G}$ of ***Half-Clique*** then ω is a valid encoding of a Yes-instance (G, k) of ***k-Clique***.

Proof: Once again, the claim is trivial if ω is not a valid encoding of an instance of ***k-Clique*** at all (since $f(\omega)$ encodes a No-instance of ***Half-Clique*** in this case) or if ω is a valid encoding of an instance (G, k) of ***k-Clique*** such that $k = 0$ or $k = 1$ (in which case this is trivially a Yes-instance of this problem). It remains only to consider the case that ω is a valid encoding of an instance (G, k) of ***k-Clique*** such that $k \geq 2$.

Since $f(\omega)$ encodes a Yes-instance $\widehat{G} = (\widehat{V}, \widehat{E})$ of ***Half-Clique*** such that $|\widehat{V}| = 2n \geq 2k \geq 4$, there exists a set $S \subseteq \widehat{V}$ with size $n \geq 2$ such that S is a clique in $\widehat{G}$. As noted above, the vertices $v_{2n-k+1}, v_{2n-k+2}, \dots, v_{2n}$ do not have any neighbours in $\widehat{G}$ so they cannot belong to S, and

$$S \subseteq V \cup \{v_{n+1}, v_{n+2}, \dots, v_{2n-k}\}.$$

Now since $|S| = n$ and

$$|S \cap \{v_{n+1}, v_{n+2}, \ldots, v_{2n-k}\}| \leq |\{v_{n+1} v_{n+2}, \ldots, v_{2n-k}\}| = n - k,$$

it follows that

$$|S \cap V| \geq |S| - (n - k) = k.$$

It is now reasonably easy to check that $S \cap V$ is a clique in G with size at least k, as required to confirm that (G, k) is a Yes-instance of ***k-Clique***.

Finally, it should be reasonably clear that the function $f : \Sigma_G^\star \to \Sigma_G^\star$ that has now been described can be computed, deterministically, using time that is at most polynomial in the length of the input string ω — see assumptions A1, A2, A7 and A8 above. Thus f is a polynomial-time mapping reduction from ***k-Clique*** to ***Half-Clique***, so that

$$\boldsymbol{k}\textbf{\textit{-Clique}} \preceq \textbf{\textit{Half-Clique}}.$$

Since ***k-Clique*** is $\mathcal{NP}$-hard, it follows that ***Half-Clique*** is $\mathcal{NP}$-hard as well.

Conclusion

Since ***Half-Clique*** $\in \mathcal{NP}$ and ***Half-Clique*** is $\mathcal{NP}$-hard, it follows that ***Half-Clique*** is $\mathcal{NP}$-complete.

A Final Note

This solution is *definitely* longer and more detailed than anything needed to receive full marks for this question on the midterm test! However, for full marks it was necessary that

- a ***certificate*** for a Yes-instance of ***Half-Clique*** was clearly described,
- at least a little bit of information about a polynomial-time verification algorithm was provided, and
- a polynomial-time mapping reduction from ***k-Clique*** to ***Half-Clique*** was clearly presented and discussed.