Computability and Complexity

Assignment 2

Ari Feiglin

Exercise 2.1:

Given an undirected graph $G = (V, E)$, we say that two points $u, v \in V$ are **nearbors** if they are neighbors or have a common neighbor. We say that $S \subseteq V$ is a **nearbor dominating set** if for every $v \in V$, either $v \in S$ or v has a nearbor in S. Prove that the following decision problem is **NP**-complete:

$$\mathsf{Dom\text{-}Set}' = \{(G, k) \mid G \text{ is an undirected graph which has a nearbor dominating set whose size is at most } k\}$$

$\mathsf{Dom\text{-}Set}'$ is obviously in **NP**, as we can define a verifier $V\big((G, k), S\big)$ which verifies that G is an undirected graph, k is a natural number, and S is a nearbor dominating set of size $\leq k$. Verifying that S is a nearbor dominating set takes polynomial time in $|G|$, as we must iterate over every vertex in G, and every vertex in S (which is a subset of the vertices in G), and check if they have a shared neighbor. So this is a polynomial-time verifier for $\mathsf{Dom\text{-}Set}'$, and there exists a polynomial p such that for every input $(G, k) \in \mathsf{Dom\text{-}Set}'$, there exists a witness S whose length is at most $p(|(G, k)|)$ (this is as $|S| \leq |G|$). So $\mathsf{Dom\text{-}Set}' \in \mathbf{NP}$, as required.

We will define a Karp reduction from $\mathsf{Dom\text{-}Set}$ to $\mathsf{Dom\text{-}Set}'$, and since $\mathsf{Dom\text{-}Set}$ is **NP**-complete this proves that $\mathsf{Dom\text{-}Set}'$ is as well. Given an undirected graph $G = (V, E)$, let us define $G' = (V', E')$ where $V' = \{v_1, v_2 \mid v \in V\}$ consists of two copies of V, and

$$E' = \{\{v_1, u_1\} \mid \{v, u\} \in E\} \cup \{\{v_1, v_2\} \mid v \in V\}$$

So we now claim that $(G, k) \mapsto (G', k)$ is a Karp reduction from $\mathsf{Dom\text{-}Set}$ to $\mathsf{Dom\text{-}Set}'$.

If S is a dominating set in G, then it is a nearbor dominating set in G'. This is because for every $v_i \in V'$,

(1) If $v_i = v_1 \in V$, then since S is a dominating set in G, either $v_1 \in S$ or v_1 has some neighbor in S. In particular, v_1 is in S or has a nearbor in S.

(2) If $v_i = v_2 \in V$, then $v_1 \in V$ and so is in S, or has a neighbor in S. This means that v_2 has a nearbor in V (since if v_1 has a neighbor in S, this neighbor is a nearbor with v_2, as they both neighbor v_1).

So for every $v \in V'$, either $v \in S$ or v has a nearbor in S. This means that S is a nearbor dominating set, as required.

In particular, this means that if G has a dominating set of size $\leq k$, then G' had a nearbor dominating set of $\leq k$. Thus if $(G, k) \in \mathsf{Dom\text{-}Set}$ then $(G', k) \in \mathsf{Dom\text{-}Set}'$.

Now, if S' is a nearbor dominating set of G', let us define S as follows. For $v_i \in S'$:

(1) If $v_i = v_1$, then put v into S.

(2) If $v_i = v_2$, then put v into S.

In other words, if we define $f\colon V' \to V$ by $f(v_i) = v$, then $S = f(S')$. We claim that S is a dominating set of G. Let $v \in V$, then v_2 is either in S' or v_2 has a nearbor in S'.

(1) If $v_2 \in S'$, then by definition $v \in S$.

(2) If v_2 has a nearbor in S', then since v_2's only neighbor is v_1, either $v_1 \in S'$ or v_1 has some neighbor in S'. The first case means $v \in S$. The second case means there is some $u_1 \in S'$ where $\{v_1, u_1\} \in E'$, as v_1's only neighbors are its neighbors in G and v_2. This would mean v and u are neighbors in G. And by definition, $u \in S$, so v has a neighbor in S.

Thus for every $v \in V$, either $v \in S$ or v has a neighbor in S. This means that S is a dominating set in G. And since $S = f(S')$, $|S| \leq |S'|$, so if S' is a nearbor dominating set whose size is $\leq k$, S is a dominating set whose size is $\leq k$.

This means that if $(G', k) \in \mathsf{Dom\text{-}Set}'$, then $(G, k) \in \mathsf{Dom\text{-}Set}$. And since we showed the other direction above, this means that $(G, k) \mapsto (G', k)$ is indeed a Karp reduction from $\mathsf{Dom\text{-}Set}$ to $\mathsf{Dom\text{-}Set}'$, which means $\mathsf{Dom\text{-}Set}'$ is **NP**-complete, as required.

Exercise 2.2:

Prove that the decision problem of determining the existence of a Hamiltonian path in an undirected graph is **NP**-complete.

Let us define this decision problem:

$$\mathsf{HP} = \{G \mid G \text{ is an undirected graph with a Hamiltonian path}\}$$

HP is in **NP**, as we can define a verifier $V(G, P)$ which verifies that G is an undirected graph, and P is a Hamiltonian path. These both take polynomial time, as all we must check that each vertex is visited by P, and only once. We can do this by iterating over P and marking the number of times we've visited each vertex, and then afterward verifying that each vertex has been visited exactly once. This takes $O(|G| + |P|)$ time, which is polynomial in the size of (G, P), so V runs in polynomial time. And since for a Hamiltonian path, $|P| \leq |G|$, V is a polynomial proof system. Thus $\mathsf{HP} \in \mathbf{NP}$ as required.

We will now define a reduction from DHP (the decision problem of Hamiltonian paths in directed graphs) to HP. Suppose $G = (V, E)$ is a directed graph, then let us define the undirected graph $G' = (V', E')$ by

$$V' = \{v_{\text{in}}, v_{\text{mid}}, v_{\text{out}} \mid v \in V\}$$

and

$$E' = \{\{v_{\text{in}}, v_{\text{mid}}\},\ \{v_{\text{mid}}, v_{\text{out}}\} \mid v \in V\} \cup \{\{v_{\text{out}}, u_{\text{in}}\} \mid (v, u) \in E\}$$

Now we claim that the map $G \mapsto G'$ is a Karp reduction. Obviously constructing G' can be constructed in polynomial time from G.

Suppose that G has a Hamiltonian path, suppose the path is

$$P\colon v_0 \to v_1 \to \cdots \to v_{n-1} \to v_n$$

then we claim that

$$P'\colon \underbrace{v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}}}_{v_0} \to \underbrace{v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}}}_{v_1} \to \cdots \to \underbrace{v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}}_{v_n}$$

is a Hamiltonian path in G'. Firstly, every edge here is a valid edge in G' since

(1) For edges of the form $\{v_{i,\text{in}}, v_{i,\text{mid}}\}$ and $\{v_{i,\text{mid}}, v_{i,\text{out}}\}$, these are edges by the definition G'.

(2) For edges of the form $\{v_{i,\text{out}}, v_{i+1,\text{in}}\}$, these are edges in G' since (v_i, v_{i+1}) is an edge in G, since it is an edge in P.

This path visits every vertex in G', since the path P visits every vertex in G. So for every $v \in G$, eventually P visits v, and so P' eventually visits v_{in}, v_{mid}, and v_{out}. This covers all the vertices in G', so P' does indeed visit every vertex in G'. And P' does not visit the same vertex twice, as this would imply that P does, contradicting P being a Hamiltonian path. So P' is a Hamiltonian path in G'.

So if $G \in \mathsf{DHP}$, then $G' \in \mathsf{HP}$. This proves the first direction of showing that $G \mapsto G'$ is a Karp reduction. We must now show that if $G' \in \mathsf{HP}$ then $G \in \mathsf{DHP}$.

Now, suppose P' is a Hamiltonian path in G'. In P' we cannot visit vertices in the order $v_{0,\text{in}} \to v_{1,\text{out}} \to v_{2,\text{in}}$ as then we cannot visit both $v_{1,\text{mid}}$ and $v_{2,\text{mid}}$. This is because if we first visit $v_{1,\text{mid}}$, this must be through the edge $v_{1,\text{in}} \to v_{1,\text{mid}}$ which means that we have visited all of $v_{1,\text{mid}}$'s neighbors and thus cannot visit $v_{2,\text{mid}}$. Similar for if we first were to visit $v_{2,\text{mid}}$.

By symmetry we cannot visit vertices in the order $v_{0,\text{out}} \to v_{1,\text{in}} \to v_{2,\text{out}}$ (the proof is very similar as before). Thus P' must be of one of the following forms:

(1) If P' starts with $v_{0,\text{in}} \to v_{0,\text{mid}}$, then

$$P'\colon v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

as we cannot go from an out vertex to an in vertex to an out vertex, as we showed above. So this is the only valid form.

(2) If P' starts with $v_{0,\text{in}} \to v_{1,\text{out}}$ then since we cannot go from in to out to in, P' must be of the form

$$P'\colon v_{0,\text{in}} \to v_{1,\text{out}} \to v_{1,\text{mid}} \to v_{1,\text{in}} \to \cdots \to v_{n,\text{out}} \to v_{n,\text{mid}} \to v_{n,\text{in}} \to v_{0,\text{out}} \to v_{0,\text{mid}}$$

as $v_{0,\text{mid}}$ must be the final vertex, as once we visit it, it must be through $v_{0,\text{out}}$, and so we cannot continue afterward. But then if we move $v_{0,\text{in}}$ to the end, this remains a valid Hamiltonian path, of the form

$$v_{1,\text{out}} \to v_{1,\text{mid}} \to v_{1,\text{in}} \to \cdots \to v_{n,\text{out}} \to v_{n,\text{mid}} \to v_{n,\text{in}} \to v_{0,\text{out}} \to v_{0,\text{mid}} \to v_{0,\text{in}}$$

Reversing it (and changing the indexes), gives another Hamiltonian path of the form

$$v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

(3) If P' starts with $v_{0,\text{mid}} \to v_{0,\text{out}}$ then P' must be of the form

$$P'\colon v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}} \to v_{0,\text{in}}$$

since $v_{0,\text{in}}$ must be the final vertex, since it must be reached from some out vertex, and we cannot go out-in-out, and since we've already visited $v_{0,\text{mid}}$, we cannot go anywhere after visiting $v_{0,\text{in}}$. But we can also move $v_{0,\text{in}}$ to the beginning and we get another Hamiltonian path of the form

$$v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

(4) Similarly if P' starts with $v_{0,\text{mid}} \to v_{0,\text{in}}$, then by symmetry we can get a Hamiltonian path of the form (since there is no real difference between out and in vertices)

$$v_{0,\text{out}} \to v_{0,\text{mid}} \to v_{0,\text{in}} \to \cdots \to v_{n,\text{out}} \to v_{n,\text{mid}} \to v_{n,\text{in}}$$

and reversing this Hamiltonian path (and changing the indexes) gives another Hamiltonian path of the form

$$v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

(5) In the case where P' starts with $v_{0,\text{out}}$, by symmetry with the cases where it starts with $v_{0,\text{in}}$, we can get a Hamiltonian path of the form

$$v_{0,\text{out}} \to v_{0,\text{mid}} \to v_{0,\text{in}} \to \cdots \to v_{n,\text{out}} \to v_{n,\text{mid}} \to v_{n,\text{in}}$$

and reversing and changing the indexes gives a Hamiltonian path of the form

$$v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

So in every case, we get a Hamiltonian path of the form

$$P'\colon v_{0,\text{in}} \to v_{0,\text{mid}} \to v_{0,\text{out}} \to v_{1,\text{in}} \to v_{1,\text{mid}} \to v_{1,\text{out}} \to \cdots \to v_{n,\text{in}} \to v_{n,\text{mid}} \to v_{n,\text{out}}$$

But this Hamiltonian path can only exist in G' if the path

$$P\colon v_0 \to v_1 \to \cdots \to v_n$$

exists in G, and is Hamiltonian. Obviously if P' exists in G', P must exist in G. And if P' is Hamiltonian, then so too must P be. This is since it must visit every v_i, as P' visits every $v_{i,\text{xx}}$, and it must visit each once, as P' visits each triplet of $v_{i,\text{in}} \to v_{i,\text{mid}} \to v_{i,\text{out}}$ only once.

So if G' has a Hamiltonian path, so too must G. Meaning if $G' \in \mathsf{HP}$, then $G \in \mathsf{DHP}$. Thus $G \in \mathsf{DHP}$ if and only if $G' \in \mathsf{HP}$, so $G \mapsto G'$ is a Karp reduction from DHP to HP. Since DHP is **NP**-complete, so too must HP be.

Exercise 2.3:

(1) Assuming that $\mathbf{P} \neq \mathbf{NP}$, is the following problem **NP**-complete?

$$\mathsf{100SubsetSum} = \{A \mid A \text{ is a set of natural numbers containing a subset whose sum is } 100\}$$

(2) How would your answer change if $A \in \mathsf{100SubsetSum}$ can be a set of integers, not just natural numbers?

(1) Notice that if $A \in \mathsf{100SubsetSum}$, and if $S \subseteq A$ has a sum of one hundred then $|S| \leq 100$. This is because the minimum number in A is at least one, and so $\sum S \geq |S|$. If we allow A to contain zeroes, then it still must have a subset of non-zero natural numbers whose sum is 100, as if $\sum S = 100$, let $S' = S \setminus \{0\}$, then $\sum S' = 100$ as well. So

in any case, in order to determine if $A \in \mathsf{100SubsetSum}$ it is sufficient to iterate over all of its subsets whose size is ≤ 100.

The number of subsets of size ≤ 100 is

$$\sum_{k=1}^{100} \binom{|A|}{k}$$

And if k is a constant, then

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n \cdots (n-k+1)}{k!} \in O(n^k)$$

And so there is $O(n^{100})$ subsets of A whose size is ≤ 100. And for each subset, it takes $O(100) = O(1)$ time to check if its sum is one hundred, so we can check if $A \in \mathsf{100SubsetSum}$ in $O(n^{100})$ time. Thus $\mathsf{100SubsetSum} \in \mathbf{P}$, and since $\mathbf{P} \neq \mathbf{NP}$, $\mathbf{NPC}$ and $\mathbf{P}$ are disjoint so $\mathsf{100SubsetSum}$ is *not* $\mathbf{NP}$-complete.

(2) Let us define a Karp reduction from $\mathsf{SubsetSum}$ to $\mathsf{100SubsetSum}$. $\mathsf{100SubsetSum}$ is obviously in $\mathbf{NP}$, as we can define a verifier $V(A, S)$ where V verifies that $S \subseteq A$ and $\sum S = 100$. This can be done in polynomial time, and since $S \subseteq A$, A would have a witness whose size is $\leq |A|$. Thus V is indeed a polynomial proof system for $\mathsf{100SubsetSum}$, meaning $\mathsf{100SubsetSum} \in \mathbf{NP}$.

Suppose (A, k) is an input for $\mathsf{SubsetSum}$, then we define

$$A' = \{200a \mid a \in A\} \cup \{100 - 200k\}$$

This can be constructed in polynomial time from (A, k).

Now, if $(A, k) \in \mathsf{SubsetSum}$, then there exists a subset $B \subseteq A$ such that $\sum B = k$. Then $B' = \{200b \mid b \in B\}$ is a subset of A' and $\sum B' = 200k$ and so $B'' = B' \cup \{100 - 200k\}$ is also a subset of A' and

$$\sum B'' = \sum B' + 100 - 200k = 100$$

and so $A' \in \mathsf{100SubsetSum}$.

And if $A' \in \mathsf{100SubsetSum}$, then suppose $B'' \subseteq A'$ has a sum of 100. Now, suppose that $100 - 200k \notin B''$. But then every element of B'' is of the form $200b$ for some $b \in A$, and so $\sum B''$ must be a multiple of 200, contradicting it being equal to 100. So $100 - 200k \in B''$. Let $B' = B'' \setminus \{100 - 200b\}$, and so B' is of the form $B' = \{200b \mid b \in B\} = 200 \cdot B$ for some subset $B \subseteq A$. And so

$$\sum B'' = \sum B' + 100 - 200k = 200 \sum B + 100 - 200k$$

And since $\sum B'' = 100$, we have

$$200 \sum B - 200k = 0 \implies \sum B = k$$

thus $(A, k) \in \mathsf{SubsetSum}$.

So $(A, k) \in \mathsf{SubsetSum}$ if and only if $A' \in \mathsf{100SubsetSum}$, meaning $(A, k) \mapsto A'$ is a Karp reduction from $\mathsf{SubsetSum}$ to $\mathsf{100SubsetSum}$. Since $\mathsf{SubsetSum}$ is $\mathbf{NP}$-complete, so is $\mathsf{100SubsetSum}$.

Exercise 2.4:

Given an undirected graph $G = (V, E)$, and a vertex coloring $\sigma_V : V \longrightarrow \mathbb{N}$ and an edge coloring $\sigma_E : E \longrightarrow \mathbb{N}$, we say that an edge $\{u, v\} \in E$ is **happy** if

$$\sigma_V(v) = \sigma_V(u) = \sigma_E(\{u, v\})$$

Prove that the following problem is $\mathbf{NP}$-complete:

$$\mathsf{HappyEdges} = \left\{ (G, \sigma_E, k) \;\middle|\; \begin{array}{l} G \text{ is an undirected graph and } \sigma_E \text{ is an edge coloring, such that there exists a vertex} \\ \text{coloring } \sigma_V \text{ such that there exist at least } k \text{ happy edges.} \end{array} \right\}$$

Firstly, $\mathsf{HappyEdges}$ is in $\mathbf{NP}$, as we can define a verifier $V\big((G, \sigma_E, k), \sigma_V\big)$ which verifies that G is an undirected graph, σ_E is an edge coloring, and σ_V is a vertex coloring such that there are at least k happy edges. This can be accomplished by iterating over every edge and checking if it is happy or not, and if so incrementing some counter. We can also require that σ_V's values are the same as σ_E's as otherwise, an edge connected to that vertex cannot be happy (meaning

$(G, \sigma_E, k) \in \mathsf{HappyEdges}$ if and only if there exists a vertex coloring using the same colors as σ_E which has at least k happy edges). Then

$$|\sigma_V| \leq |\sigma_E| \cdot |V|$$

as the length of σ_V is less than the maximum value in σ_E's length times $|V|$ (ie. σ_V's length is less than the length of the coloring where every vertex colored is colored with the maximum value), and the maximum value in σ_E has a length less than that of σ_E itself. And so the length of this witness is less than a polynomial of the length of (G, σ_E, k) (the polynomial being $|(G, \sigma_E, k)|^2$).

This means that V is a polynomial-time verifier, and every input in $\mathsf{HappyEdges}$ has a witness whose length is less than some polynomial of the length of the input. And so V is a polynomial proof system for $\mathsf{HappyEdges}$, so $\mathsf{HappyEdges} \in \mathbf{NP}$.

Now, to show that $\mathsf{HappyEdges}$ is $\mathbf{NP}$-complete, we will construct a Karp reduction from SAT to $\mathsf{HappyEdges}$. Suppose we are given a boolean formula in CNF φ. We define a graph G as follows:

(1) For every variable x_i in φ, we define a vertex x_i in G. So if the variables in φ are $x_1, \dots, x_n$, then G has vertices $x_1, \dots, x_n$.

(2) For every disjunction in φ, we define another vertex D_i. So if φ is of the form $\varphi = \bigwedge_{i=1}^m D_i$, then G has vertices $D_1, \dots, D_m$.

(3) For every variable x_i, we define an edge $\{x_i, D_j\}$ between x_i and every disjunction D_j in which x_i appears.

Now let us define

$$\mathsf{T}_i = 2i, \quad \mathsf{F}_i = 2i + 1$$

this just means that T_i and F_i are all distinct. Now, we define a coloring of the edges in G as follows: for every edge $\{x_i, D_j\}$, if x_i appears in D_j as-is then set $\sigma_E(\{x_i, D_j\}) = \mathsf{T}_i$. Otherwise $\neg x_i$ appears in D_j, and so set $\sigma_E(\{x_i, D_j\}) = \mathsf{F}_i$.

And finally set k to m, the number of disjunctions in φ. Constructing G, σ_E, and k all take polynomial time.

We claim that $\varphi \mapsto (G, \sigma_E, k)$ is a Karp reduction from SAT to $\mathsf{HappyEdges}$. If φ is in SAT, then there exists a boolean vector τ which satisfies φ. Now, for every disjunction D_j in φ, D_j must be satisfied by some x_i which occurs in D_j (meaning τ_i is true and x_i appears in D_j, or τ_i is false and $\neg x_i$ appears in D_j). So set $\sigma_V(x_i) = \sigma_V(D_j) = \tau_i$ (by τ_i what I mean is that if τ_i is true, then set these to T_i, and otherwise to F_i). Notice that this means that $\sigma_V(x_i) = \sigma_V(D_j) = \sigma_E(\{x_i, D_j\})$, as if x_i occurs in D_j then since x_i satisfies D_j, τ_i is true, and $\sigma_E(\{x_i, D_j\}) = \mathsf{T}_i$. And if $\neg x_i$ occurs in D_j, then this means $\sigma_E(\{x_i, D_j\}) = \mathsf{F}_i$ and since x_i satisfies D_j, this means τ_i is false. The rest of the vertices are variables, and we can also color them by $\sigma_V(x_i) = \tau_i$ (this means that if τ_i is true, then we color it as T_i, and if τ_i is false then it is colored as F_i).

This is a valid coloring as for every disjunction D_j, we are choosing a single variable which satisfies it. So each disjunction is only being set once. And each variable is being set to the same value, τ_i (ie. T_i if τ_i is true, and F_i if τ_i is false). So this coloring is well-defined.

Suppose we chose x_i to be the variable which is satisfying D_j. Then as we said above, $\sigma_V(x_i) = \sigma_V(D_j) = \sigma_E(\{x_i, D_j\})$, so $\{x_i, D_j\}$ is happy. So every disjunction has a happy edge connected to it, and since there are $k = m$ disjunctions and none of them are connected, this means there are at least k happy edges. Thus if $\varphi \in \mathsf{SAT}$, then $(G, \sigma_E, k) \in \mathsf{HappyEdges}$.

Now we must show the converse; suppose $(G, \sigma_E, k) \in \mathsf{HappyEdges}$. Notice that every disjunction can be connected to at most one happy edge. Suppose that $\{x_i, D_j\}$ is a happy edge, then for every other edge connected to D_j, it is of the form $\{x_{i'}, D_j\}$ and $\sigma_E(\{x_i, D_j\}) \neq \sigma_E(\{x_{i'}, D_j\})$ since the F_i and T_i are all distinct. And since $\sigma_V(D_j) = \sigma_E(\{x_i, D_j\})$, $\sigma_V(D_j) \neq \sigma_E(\{x_{i'}, D_j\})$ so $\{x_{i'}, D_j\}$ cannot be happy.

So every disjunction can be connected to at most one happy edge, and there are m happy edges. Since m is the number of disjunctions, this means that each disjunction is connected to exactly one happy edge. So let us define a boolean vector τ as follows:

(1) For every disjunction D_j, it is connected to a happy edge $\{D_j, x_i\}$. Set $\tau_i = \sigma_E(\{D_j, x_i\})$ (by this I mean that if the edges's color is equal to T_i set τ_i to true, and if the edge's color is F_i, set τ_i to false).

(2) For the rest of the variables, set τ_i arbitrarily.

Let D_j be a disjunction, and let its happy edge be $\{x_i, D_j\}$. Then if $\sigma_E(\{x_i, D_j\}) = \mathsf{T}_i$ then by its definition x_i occurs in D_j and since we've then set τ_i to be true, this means that D_j is satisfied (since it is a disjunction containing x_i). And if $\sigma_E(\{x_i, D_j\}) = \mathsf{F}_i$ then by its definition $\neg x_i$ occurs in D_j, and since we've then set τ_i to be false, D_j is satisfied (since it is a disjunction containing $\neg x_i$). So every disjunction is satisfied and so φ is satisfied.

This means that if $(G, \sigma_E, k) \in \mathsf{HappyEdges}$, then $\varphi \in \mathsf{SAT}$. So $\varphi \in \mathsf{SAT}$ if and only if $(G, \sigma_E, k) \in \mathsf{HappyEdges}$, so $\varphi \mapsto (G, \sigma_E, k)$ is indeed a Karp reduction from SAT to $\mathsf{HappyEdges}$. Since SAT is $\mathbf{NP}$-complete, so is $\mathsf{HappyEdges}$, as required.