# Tutorial 9

Gidon Rosalki

2025-05-28

## 1 Verifiers

Reminder: We define the language $NP$ as follows:

$$NP = \bigcup_{k=0}^{\infty} NTIME\left(n^k\right)$$

**Definition 1.1** (Polynomial verifier)**.** *A verifier for a language $L$ is a deterministic TM $V$ which given $(w, c)$, with $c$ polynomial in $|w|$, runs in polynomial time in $|w|$ and*

$$\forall w \in L,\ \exists c \in \Sigma^* : V(w, c) = q_{acc}$$
$$\forall w \notin L,\ \forall c \in \Sigma^*\ V(w, c) = q_{rej}$$

*This $c$ is called a witness.*

**Theorem 1** (Equivalent definition of NP)**.** *A language $L$ can be recognized by an NTM in polynomial time* ***if and only if*** *there is a polynomial verifier for $L$.*

*Proof .* Proved in lecture 9 □

**Example 1.** *Let*

$$\textit{U-ST-HAMPATH} = \{\langle G, s, t\rangle : G \text{ is an undirected graph, and there exists a Hamiltonian path from } s \text{ to } t\}$$

*Show that U-ST-HAMPATH is NP-Complete:*

*Solution.* We first need to show that U-ST-HAMPATH is in NP: We will show that there exists a polynomial time verifier, $V$. Let $V$ take as input $w = \langle G, s, t\rangle$, and a witness $c = \langle v_1, \ldots, v_n\rangle$, a sequence of $n = |V|$ vertices. $V$ then performs the following:

1. Check if the first vertex is $s$, and the last is $t$
2. Check that all nodes are different (since if this is not the case, then there is a repetition, and this is not a Hamiltonian path)
3. Validate the existence of an edge between every two adjacent vertices

$\langle G, s, t\rangle \in U - ST - HAMPATH$ **if and only if** There is a Hamiltonian path in $G$, from $s$ to $t$ **if and only if** There exists $c \in \Sigma^*$ such that $V$ accepts $(w, c)$.
The witness $c$ is polynomial in the size of the input $|\langle G, s, t\rangle|$, and each operation can be performed in polynomial time.

We now need to show that U-ST-HAMPATH is NP-Hard. In the previous tutorial, it was shown that $3 - SAT \leq_p D - ST - HAMPATH$. As 3-SAT is NP-Hard, then so is D-ST-HAMPATH. We can thus show that U-ST-HAMPATH is NP-Hard, by performing the reduction $D - ST - HAMPATH \leq_p U - ST - HAMPATH$. Let there be the reduction $f(\langle G, s, t\rangle) = \langle G', s_{in}, t_{out}\rangle$, where $G'$ is defined as follows:

- For every vertex $v \in V$, our function will write the new vertices $v_{in}, v_{mid}, v_{out}$, and connect the nodes with the following edges: $\{v_{in}, v_{mid}\}, \{v_{mid}, v_{out}\}$.
- For every edge $(u, v) = e \in E$, we will define the new edge $u_{out}, v_{in}$.

Correctness: Let us assume that $\langle G, s, t \rangle \in D - ST - HAMPATH$. Then, let there be the directed Hamiltonian path $s, u^1, u^2, \ldots, u^k, t$ in $G$. Therefore, there is the path in $G'$:

$$s_{in}, s_{mid}, s_{out}, u^1_{in}, u^1_{mid}, u^1_{out}, \ldots u^k_{in}, u^k_{mid}, u^k_{out}, t_{in}, t_{mid}, t_{out}$$

Since the original path contained all the vertices, then so too does the new path in $G'$, thus

$$\langle G', s, t \rangle \in U - ST - HAMPATH$$

Let there be a Hamiltonian path in $G'$ from $s_{in}$ to $t_{out}$. Thus, there is a Hamiltonian path in $G'$, but we do not know what this path "looks like". We proceed with the following claim. A Hamiltonian path that starts at $s_{in}$ and ends at $t_{out}$ does not contain a directed traversal of the form $(v_{in}, u_{out})$. That is to say, we never travel "backwards" along an edge. Let us assume that there is such a directed traversal, and let $(v_{in}, u_{out})$ be the first such one in this path. Since the path is Hamiltonian, we must visit $v_{mid}$ at some point on the path. If we have already visited it, then we must have visited it from $v_{out}$, since we are now located at $v_{in}$, and it has no other edges, but in that case, in order to reach $v_{out}$, we must have arrived from some $x_{in}$, which is a contradiction to $(v_{in}, u_{out})$ being the first such pair. Therefore, we must visit $v_{mid}$ after visiting $u_{out}$, and therefore we must reach it from $v_{out}$, at which point we get stuck, since we have already visited $v_{in}$. Therefore, we cannot reach $t_{out}$. We can therefore conclude that all the edges in the path are of the formats $(v_{in}, v_{mid}), (v_{mid}, v_{out}), (v_{out}, u_{in})$. Thus, the Hamiltonian path is of the form

$$s_{in}, s_{mid}, s_{out}, u^1_{in}, u^1_{mid}, u^1_{out}, \ldots u^k_{in}, u^k_{mid}, u^k_{out}, t_{in}, t_{mid}, t_{out}$$

which can be easily mapped to a Hamiltonian path in $G$.

Runtime: Clearly polynomial, since $|G'| = 3|G|$, and computing it is trivial. □

## 2 3-COLOURING

Given a graph $G = (V, E)$, the $k$-COLOURING problem is to validate whether or not it is possible to colour all vertices in $k$ different colours such that no 2 neighbouring vertices have the same colour.

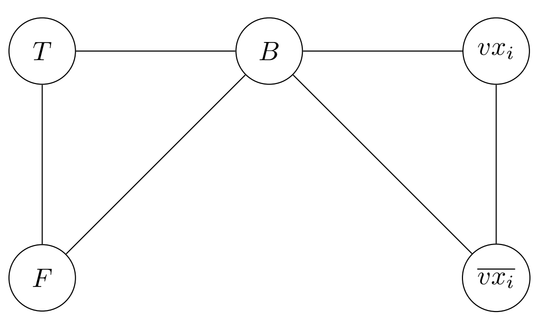
$$\text{3-COLOURING} = \{\langle G \rangle : \text{ There's a valid 3-colouring of } G\}$$

One way to check if a graph $\langle G \rangle$ is not in 3-COLOURING is by validating the existence of a clique of size 4 in $G$. This can be done in polynomial tie, since there are $\binom{|V|}{4} = O\left(|V|^4\right)$ sets of size 4, and it takes polynomial time to verify if each is a clique.

**Theorem 2** (3-COLOURING is NP-Complete)**.** *We will first show that $3C \in NP$ by using a verifier $V$. Given a graph $\langle G \rangle$, and a colouring $c$, $V$ checks if $c$ is a valid colouring of $G$, by going over all the edges in $E$, and verifying that the vertices at each edge have different colours. This can be done in polynomial time.*
*We will now prove that $3 - SAT \leq_p 3C$, thus proving hardness in NP. Let $\langle \varphi \rangle$ be a 3-CNF formula, with $x_1, \ldots, x_n$ variables, and $c_1, \ldots, c_m$ clauses. The reduction creates the following graph:*

1. *Create 3 vertices, $T, F, B$, and connect them all together.*

2. *For each variable $x_i \in \varphi$, create 2 vertices, $vx_i, \overline{vx_i}$, add an edge between them, and an edge from each of them to $B$.*




3. *For each clause $c_j = (l_1 \vee l_2 \vee l_3)$, add triple or gadget graph, connect to relevant literals vertices, and connect the output to $B$ and to $F$ (forcing it to be $T$)*

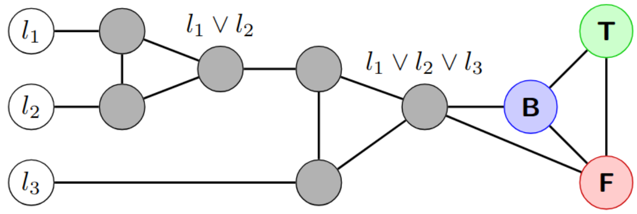



Figure 1:

*Notice, if $l_1, l_2, l_3$ are all false, then the triple or gadget output must be coloured false, but it is also connected to $F$ , so it is an invalid colouring. On the other hand, if at least one of them is coloured true then there exists a valid colouring.*

*Correctness: $\varphi \in 3SAT \implies \varphi$ is satisfiable. Let $A(x_i) \to \{T, F\}$ be a satisfying assignment. If $x_i$ is assigned $T$, we colour $vx_i$ green $= T$, and $\overline{vx_i}$ red $= F$. Otherwise, we assign the opposite. This is valid as they are connected to $B$, and to each other, 3 different colours so far. Since $\varphi$ is satisfiable, each clause $c_j$ is satisfiable, so at least one of the literals is true, and from the gadget observation, it is 3-colourable.*
*Let $\langle G \rangle \in 3C$. We construct a satisfying assignment through the colour of the vertices $vx_i$. If it is coloured true we assign true, otherwise false. If by contradiction this is not a satisfying assignment, then there is a clause $c_j$ which is not satisfied. As such, $l_1^j, l_2^j, l_3^j$ are all false, and are thus all coloured false in the graph. In this case this means the clause triple or gadget is not 3 colourable, thus the entire graph is not 3 colourable in contradiction. So the assignment must be a satisfying assignment.*
*Runtime: For each variable we construct 2 nodes, and connect 3 edges, this is polynomial in $n$. For each clause we create an or gadget which is a fixed size of nodes and edges. There are $m$ gadgets, so this too is polynomial in the input size.*