Informatics 2 – Introduction to Algorithms and Data Structures

Tutorial 9 - NP-completeness, Approximation

1. A propositional formula ϕ over the logical variables $\{x_1, \ldots, x_n\}$ is in *Conjunctive Normal Form (CNF)* if it is of the form

$$\phi = C_1 \wedge \ldots \wedge C_m,$$

such that each clause C_j is a *disjunction* of literals over the x_i variables (ie, every C_j is $(\ell_{j,1} \vee \ldots \ell_{j,r(j)})$ for some $r(j)$ with each $\ell_{j,h}$ being x_i or $\bar{x}_i$ for some i).

The formula ϕ is said to be in *3-CNF* when it is also the case that every clause C_j contains 3 distinct literals (referring to three different variables).

We can consider the following two decision problems:

SAT: Given a *CNF* formula ϕ, determine whether there is a logical assignment to the variables $\{x_1, \ldots, x_n\}$ which simultaneously satisfies all clauses of ϕ.

3-SAT: Given a *3-CNF* formula ϕ, determine whether there is a logical assignment to the variables $\{x_1, \ldots, x_n\}$ which simultaneously satisfies all clauses of ϕ.

   (a) First show that *both* SAT and 3-SAT belong to the complexity class NP, by describing a polynomial-time algorithm which can *verify* a given formula against a solution/certificate.

   (b) Then show that SAT $\leq_P$ 3-SAT (ie, that there is a polynomial-time reduction from SAT to 3-SAT).

   Hint: Think about introducing extra "dummy" variables to design with a *3-CNF* formula with the same constraints as the initial *CNF* formula.

Note that given the status of SAT as the canonical NP-complete problem, this gives the proof that 3-SAT is also NP-complete.

We had already relied on the NP-completeness of 3-SAT in Lecture 26, therefore we can see this question as "filling in a gap" in our coverage of NP-completeness so far.

2. In our lectures on dynamic programming, we considered the "coin changing" problem. We can cast this as a decision problem, by asking whether the optimum solution is $\leq h$ for some given h.

COINS: Given the coin system with denominations $c_1 = 1, \ldots, c_k \in \mathbb{N}$, and a target value $v \in \mathbb{N}$, and a threshold count $h \in \mathbb{N}$, is it the case that the minimum-cardinality multiset of coins (summing to v) has cardinality $\leq h$?

(a) We have already seen a dynamic programming algorithm which will take inputs $c_1 = 1, \ldots, c_k \in \mathbb{N}$ and $v \in \mathbb{N}$, and return the minimum-cardinality multiset of coins that sums to v.

Hence we can use this algorithm to solve the COINS decision problem, simply by comparing its returned value against h.

Will this approach be *polynomial-time* in the size of the inputs to COINS? (these being $\lg(v), \lg(h)$ and $\lg(\max_i c_i)$).

(b) We would like to show that COINS belongs to the class NP. Is it the case that a "certificate" (multiset of coins of cardinality $\leq h$) can be verified in time polynomial in $\lg(v), \lg(\max_i c_i)$ and $\lg(h)$?

3. In Lecture 27 we covered an algorithm to *derandomize* the naïve randomized algorithm (generate a uniform random assignment from $\{0,1\}^n$) for computing an assignment which is expected to satisfy $\geq \frac{7}{8}m$ clauses of a given 3-CNF formula with m clauses.

(a) Execute the derandomization algorithm on the following input formula, to find an assignment which will satisfy at least $\frac{7}{8}9 = 7.875$ of the clauses (meaning 8 in practice, as the number of clauses satisfies for a specific assignment must be an integer value).

$$\begin{aligned}\Phi \quad = \quad & (x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee \bar{x_2} \vee \bar{x_3}) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge \\ & (x_1 \vee \bar{x_2} \vee \bar{x_3}) \wedge (x_1 \vee \bar{x_2} \vee x_4) \wedge (x_2 \vee x_3 \vee \bar{x_4}) \wedge \\ & (\bar{x_1} \vee \bar{x_3} \vee \bar{x_4}) \wedge (\bar{x_2} \vee \bar{x_3} \vee x_4) \wedge (\bar{x_1} \vee x_3 \vee x_4).\end{aligned}$$

(b) Suppose we wanted to apply this same derandomization method to more general CNF formulas than 3-CNF. What main differences we would need to take care of?

4. Given an undirected graph $G = (V, E)$, and sets $\mathcal{I}, \mathcal{K} \subseteq V$, we say that

- $\mathcal{I}$ is an *Independent Set* of G if for every $u \in \mathcal{I}, v \in \mathcal{I} \setminus \{u\}$, that $(u, v) \notin E$.
- $\mathcal{K}$ is a *Vertex Cover* of G if for every $e = (u, v) \in E$, either $u \in \mathcal{K}$ or $v \in \mathcal{K}$.

We have already seen the following two Decision Problems in our lectures on NP-completeness and approximation algorithms.

INDEPENDENT SET: "Does the given graph have an Independent Set of size $\geq k$?"

VERTEX COVER: "Does the given graph have a Vertex Cover of size $\leq \ell$?".

(a) Show that $\mathcal{I}$ is an *Independent Set* of $G \Leftrightarrow (V \setminus \mathcal{I})$ is a *Vertex Cover* of G.

What does this relationship tell us about the relationship between the two Decision problems above?

(b) Can we use the $\Leftrightarrow$ in (a) to infer any relationship about *approximation algorithms* for the optimisation variants ("max" for INDEPENDENT SET, "min" for VERTEX COVER) of these problems?

5. (optional) Finally, we consider the problem 3-Colourable, which asks whether a given graph $G = (V, E)$ has a *proper* 3-colouring or not.

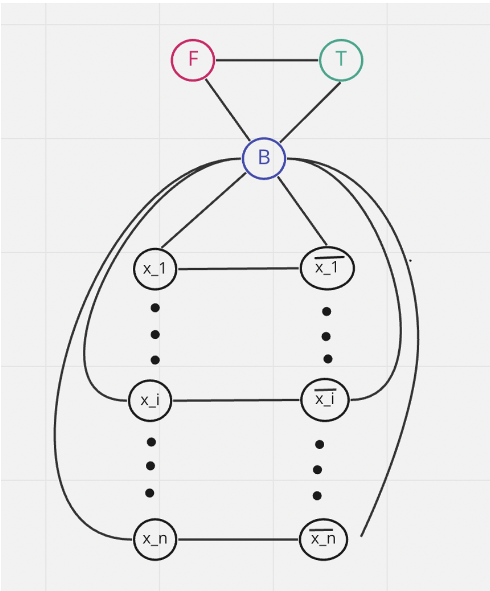
   3-Col: Given an undirected graph $G = (V, E)$, is there is an assignment $c : V \to \{blue, red, green\}$ such that for every $(u, v) \in E$, $c(u) \neq c(v)$?

   (Note the k-Col question can be defined for any $k \geq 2$, and in fact the 2-Col question is equivalent to asking whether a graph is bipartite.)

   We will prove NP-completeness of 3-Col in this question in 3 steps:

   (a) First show that 3-Col is in the class NP, by describing how we could verify a certificate (proposed colouring $c : V \to \{blue, red, green\}$) for $G = (V, E)$ in time polynomial in $n = |V|$ and $m = |E|$.

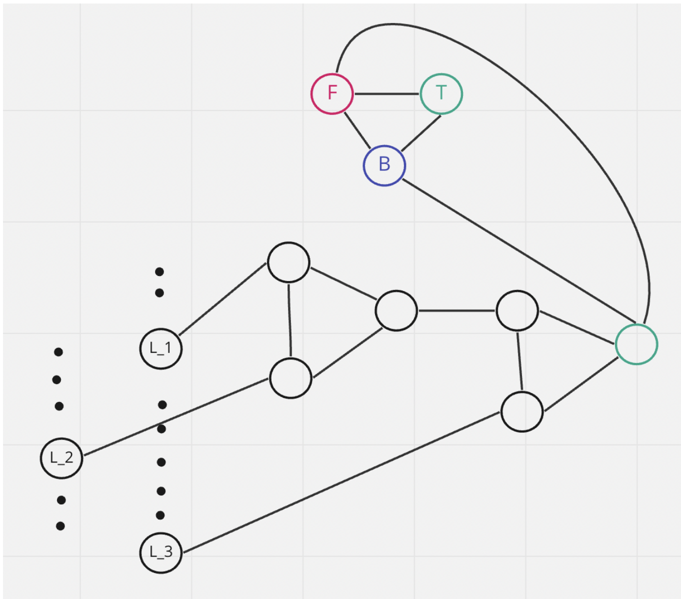
   (b) Next we consider the task of *reducing* the known 3-Sat problem to 3-Col. Our first step for this reduction will be to build a graph which can "encode" the conditions of a truth assignment for the logical variables $\{x_1, x_2, \ldots, x_n\}$.
   Step 1 is as follows: we define first a "central triangle" on the special vertices $\{T, F, B\}$, with the connecting edges $(T, F), (T, B), (F, B)$, where nodes T and F will be used to set the "colour for True" and "colour for False".
   (*We are ensured that if we have a* proper 3-colouring*, each of T, F, B must get a different colour - I have labelled T as green, F as red and B as blue, but it doesn't matter which way they are assigned … what will be important in the argument is "T's colour" and "F's colour".*)
   Also, for every variable x_i appearing in the 3-CNF formula Φ, we add the "x_i triangle" consisting of vertices x_i and $\neg x_i$ together with the central vertex B. This means that there is a specific node defined for *every possible literal on the n variables*. The resulting graph is shown here:



   Your first step is to justify the following claim:
   Claim: The *proper 3-colourings* of the subgraph above are the colourings where for every $i = 1, \ldots, n$, *exactly* one of $x_i, \neg x_i$ has "T's colour" and the other one has "F's colour".

(c) Step 2 of our reduction adds extra "gadgets" onto the assignment-setting subgraph from (b), to encode constraints equivalent to each clause C_j in Φ.

Consider any clause C_j of the formula Φ, made up of the three literals $L_1, L2$ and L_3 (so C_j was $(L_1 \vee L_2 \vee L_3)$). Consider the following "2-triangle gadget" connected to the 3 relevant literal nodes (set up in (b)), with the end node fixed to "T's colour" (this achieved by connecting up to the "central triangle").



Justify the following claim:

Claim: If we know that each of L_1, L_2, L_3 can only be "T's colour" (green) or "F's colour" (red), then there will be a proper 3-colouring of the "gadget" *if and only if* at least one of L_1, L_2 and L_3 is green.

(it will probably help to give names to the unlabelled vertices, to write your argument)

(d) How can we combine (b) and (c) to achieve a full $\leq_P$ reduction from 3-SAT to 3-COL?

Mary Cryan, 7th March 2025