Complexité et Calculabilité

Groupe D , L3-Informatique

October 17, 2023

1 Partition en Triangles

1.1 Introduction :

1.1.1 Langage :

Toute traduction elle s'opère entre un langage formel σ et un langage θ . Un langage est un ensemble de mot bien formé , pour savoir si un mot est bien formé nous devons doté notre langage de règle de grammaire .

Un mot est un ensemble de symbole , les symboles sont définit sur un alphabet .

1.1.2 Problème de décision:

Un type de problème est un ensemble de phrase obéissant à une certaine construction , par exemple : toute les phrases qui obéissent à la forme "Fait il beau dans le pays x ? , avec x $\in$ Pays"

Un problème de décision est un problème qui demande de répondre à la phrase par "oui" ou "non" .

Ainsi un problème de décision peut être vue comme un langage L tel que l'ensemble des phrases appartenant au langage L est associé à une réponse positive .

1.1.3 Traduction :

Une réduction est appelé communément appelé une "traduction" .

Ainsi l'on dit qu'on a réduit un problème A à un problème B si l'on arrive à trouver une fonction qui traduit tout instance d'un problème A en une instance du problème B . La réduction se note : $A \leq B$.

Comme dit précédemment un problème est un langage d'où le mot traduction, ainsi f se définit par :

Soit A un problème définit sur un langage σ et B un problème NP-complet définit sur un langage θ .

Si f est une fonction de traduction de A vers B alors on a :

$f : \sigma^* \rightarrow \theta^*$ tel que : $A \leq B \leftrightarrow w \in A \leftrightarrow f(w) \in B$

En francais ça donne :

Soit f tel que "w décrit une instance du problème A " $\leftrightarrow$ "on peut traduire w en une instance du problème B par f " .

Une autre façon de voir les choses est de dire que <u>tout problème de type A</u> est mis en correspondance par f avec une partie des problèmes de type B .

1.1.4 Réduction

Remarque : f est supposé est calculable en temps polynomial .

Si nous avons un algorithme : Solve_B nous répondant "oui" ou "non" pour chaque instance d'un problème de type B et que nous avons f <u>une fonction totale</u>, traduisant un problème de A vers un problème de type B . Alors comme f est totale , "$\forall a \in L(A)$ Solve_B ($f(I_a)$)" résout tous les problème de A .

Ainsi on pourrait dire que f(A) est inclu dans L(B) et comme f est totale L(A) est inclu dans L(B) à une traduction près . Il serait ainsi naturel de dire que B est au moins aussi dur que A car B "contient la complexité de A" (à un temps polynomial près) .

2 Exerice:

Notre objéctif va être de prouver que le problème de la partition en Triangles est NP-complet .

2.1 Probleme Enonce:

Partition en Triangles:

Entrée : Un graphe G = (V,E) avec $|V| = 3q, q \in \mathbb{N}$.

Question : Est-ce qu'il existe une partition de V en q ensembles disjoints V1, V2,..., Vq de trois sommets chacun tel que pour chaque $Vi = \{v_{i1}, v_{i2}, v_{i3}\}$ les trois arêtes appartiennent à E ?

Autrement dit peut-on couvrir le graphe par des triangles de taille trois ?

2.2 Préliminaire

Pour prouver que le problème P1 est NP-complet on doit réduire un problème connu comme étant NP-Complet à P1 , ainsi on prouvera que P1 est au moins aussi dur qu'un problème NP et par transitivité qu'il est NP-complet*

*: En effet , tout problème de NP est réductible à un problème NP-complet grâce à une certaine fonction de traduction: f_i :

Soit PC1 un problème NP-Complet et $L(PC1)$ son langage , alors nous avons : $\forall P_i \in NP \exists f_i \in poly_function \forall I_i \in L(P_i)$ tq $f_i(I_i) \in L(PC1)$

Si un problème NP-complet PC1 se réduit à un problème NP1 par une fonction de traduction g1 :

$\exists g1 \in poly_function \forall IC_i \in L(PC1)$ tq $g1(IC_i) \in L(NP1)$

Alors on a :

$\forall I_i \in L(P_i)$ $g1(f_i(I_i)) \in L(NP1)$

Donc NP1 est aussi NP-complet .

3 Résolution:

Nous allons utilisé le problème 3-SAT pour la réduction.Ainsi nous devons traduire une instance de 3-SAT en une instance de la partition Triangle.

3.1 Traduction :

3.1.1 Vocabulaire:

Soit $\mathbb{P}$ un ensemble de variables porpositionnelle .

Un littéral est de la forme : $\forall l_i, \exists pv_i \in \mathbb{V}$ where $l_i \in \{\neg pv_i, pv_i\}$

Une clause est de la forme : $\forall C_i, C_i = \bigvee_{i\in[r]} l_i$ où $l_i \in \mathbb{L}$ et $r \in \mathbb{N}$

Et soit $taille : \mathbb{C} \to \mathbb{N}$ qui associe à chaque clause le nombre de littéraux dans celle ci .

Notation : Soit C_i une clause quelconque $taille(C_i) = |C_i|$

Une CNF est une conjonction de clause :

$\bigwedge_{i\in[r]} C_i$ avec $C_i \in \mathbb{C}$ et $|C_i| <= r$

Une 3-CNF est définit comme suit: $\bigwedge_{i\in[3]} C_i$ avec $C_i \in \mathbb{C}$ et $|C_i| = 3$

3.2 3-SAT

Soit ϕ une 3-CNF définit sur $\mathbb{V}$ (on suppose que l'intepprétation des symboles de ϕ est hérité de la logique classique , on notera J cette structure).

Une instance du problème 3-SAT est définit comme suit :

Existe-t-il une assignation qui satisfasse ϕ ?

$\phi^{\mathbb{J}}[l_0 \to b_0, l_1 \to b_1, ..., l_n \to b_n] = 1$ avec $l_i \in \mathbb{L}$ et $b_i \in 0, 1$

Ou de manière fonctionnelle avec $\theta : \mathbb{P}^n \rightarrow \{0,1\}^n$ tel que $\forall l_i \in \{l_0, l_1, ..., l_p\}, \theta_{b_0,b_1,...,b_p}(l_i) = b_i$ et $\theta^{\phi}_{b_0,b_1,...,b_p}(\phi) = 1$

(Cette fonction pourrait nous être utile pour résoudre FSAT).

3.3 Traduction :

Soit l'instance d'un prolbème 3-SAT avec n clauses et k variables . On note ϕ l'expression à satisfaire , $\mathbb{V}^{\phi}$ l'ensemble des symboles de variables de ϕ, $\mathbb{L}^{\phi}$ l'ensemble des littéraux associer à $\mathbb{V}^{\phi}$,$\mathbb{C}^{\phi}$ l'ensemble des clauses appareceant dans ϕ .

(On utilisera : $lp_i et ln_i \in \mathbb{L}^{\phi}$, $v_i \in \mathbb{V}^{\phi}$ et $C_i \in \mathbb{C}^{\phi}$)

Les littéraux de $\mathbb{L}^{\phi}$ seront traduits par les sommets $\{s_1, ..., s_2 kn\}$. En sommes on crée pour chaque variable de $\mathbb{V}^{\phi}$ (k) 2 sommets (2) pour chacune des n clauses de ϕ(n) (=2kn).

Le premier sommet s_{2i} représente la pontentielle occurence du littéral positif $lp_i = v_i$ dans la clause C_i et l'autre s_{2i+1} la pontentielle occurence du littéral négatif : $ln_i = \neg v_i$ dans la clause C_i .

Soit un ensemble de sommets : $\{a_1, ..., a_{kn}\}$, taille = kn car il y a kn sommets invalidé par l'assignation parmis les 2kn .

La distribution de valeurs de vérité θ sera traduite par : si ${}^{\phi}(lp_i) = 1$ avec $lp_i \in \mathbb{L}$ un littéral positif alors on ajoutera le triplet $(s_{2i+1}, a_{2i}, a_{2i+1})$ à la couverture Θ . sinon on ajoutera le triplet $(s_{2i}, a_{2i}, a_{2i+1})$ à la couverture Θ . Ces triplets représente les littéraux/clauses qui ont été exclu par la nature de l'assignation .

Soit un ensemble de variable : $\{a_0, a_1, ..., a_{2n}\}$

Soit un ensemble de variable : $\{c_1, ..., c_{3n}\} \in 0, 1^{3n}$, $c_{3i} = 1$ si la première variable de la clause C_i est un littéral positif sinon $c_{3i} = 0$.

La clause C_i sera traduite par 3 triplets , dont chacun possède 2 sommets dans $\{a_1, ..., a_{2n}\}$ et un sommet dans $\{s_1, ..., s_{2kn}\}$. Ainsi C_i se transforme en :

$CT_i = \{(s_{2n\times n_1+i+c_{3i}}, a_{2i}, a_{2i+1}), (s_{2n\times n_2+i+c_{1+3i}}, a_{2i}, a_{2i+1}), (s_{2n\times n_3+i+c_{2+3i}}, a_{2i}, a_{2i+1})\}$

avec n_1 ,n_2 et n_3 les indexes des variables apparaissant dans la clauseC_i . (* voir exemple).

Si $\exists e \in [2kn]$ tel que s_e n'est pas encore couvert alors $\forall CT_i$ tel que $s_e \in CT_i$ on ajoute les triplets de CT_i à Θ .

Soit $C_i = (l^i_1, l^i_2, l^i_3), \theta^{C_i}(C_i) = 1$ sera traduit par $\exists e \in [2kn]$ tel que $(s_e, a_{2i}, a_{2i+1}) \in \Theta$

Soit $\theta^{\phi}(\phi) = 1$ sera traduit par $\forall C_i \in \mathbb{C}, \exists e \in [2kn]$ tel que $(s_e, a_{2i}, a_{2i+1}) \in \Theta$ ou plus simplement Θ couvre tous les points du graphe .

*:Ainsi par exemple $2n \times n_1$ nous place sur le premier sommet de la variable numéro n_1 puis $+i$ sur l'instance positive destiné à la clause C_i et $+c_{3i}$ sur l'instance appareceant dans C_i (+0 si positif +1 si négatif).

3.4

There is also a new problem, these model required new intensity level to store the information in the image.