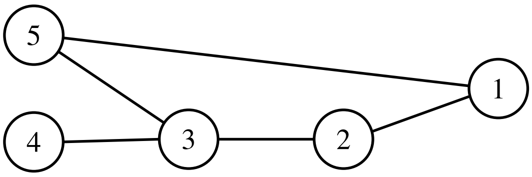
TD nº 1

Graphes et colorations

Un graphe est ici donné par une matrice T de taille n par n : $T[i,j]$ contient 1 s'il y a une arête du sommet i au sommet j, et contient 0 sinon. Un graphe G de n sommets est donc de taille n^2. Un sommet dans ce modèle est un entier compris entre 1 et n : il est donc de taille $\log n$. On voit sur la figure ci-dessous un exemple de graphe à 5 sommets, et la matrice qui lui correspond.



$$T = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

FIGURE 1 – Un graphe et sa représentation comme un tableau

Exercice 1. *Graphes de Mycielski*

Une coloration d'un graphe par k couleurs est la donnée, pour chaque sommet du graphe, d'une des k couleurs. On dit qu'un graphe est k-coloriable si on peut le colorer avec k couleurs de sorte que deux sommets adjacents (reliés par une arête) soient de couleurs différentes. Il est k-chromatique s'il est k-coloriable mais pas $(k-1)$-coloriable.

1. Montrer qu'un graphe à n sommets est n-coloriable. Donner un graphe à 5 sommets qui n'est pas 4-coloriable.

On dit qu'un graphe est biparti si on peut partitionner ses sommets en deux ensembles V_1 et V_2 de sorte qu'il n'y ait aucune arête entre deux sommets de V_1 (resp. de V_2). Les seules arêtes joignent donc un sommet de V_1 à un sommet de V_2.

2. Montrer qu'un graphe est 2-coloriable si et seulement si il est biparti.
3. Donner un algorithme pour décider si un graphe est biparti.
4. On dit qu'un graphe contient une clique de taille k s'il contient k sommets tous reliés les uns aux autres. Montrer que si un graphe est k-coloriable, il n'a pas de clique de taille $k+1$.
5. Donner deux exemples d'un graphe sans clique de taille 3 (sans triangle) mais qui ne sont pas 2-coloriable.

Soit G est un graphe. On note les sommets $u_1 \dots u_n$. On construit un graphe $\theta(G)$ de la façon suivante

- $\theta(G)$ a $2n+1$ sommets : les n sommets du graphe original, des sommets $u'_1 \dots u'_n$ en plus, et un sommet v ;
- Le graphe contient les arêtes suivantes :
  - Les arêtes originales entre les sommets u_i ;
  - S'il y a une arête de u_i à u_j, on ajoute une arête de u'_i à u_j ;
  - Il y a une arête de u'_i à v pour tout i.

6. Construire le graphe $\theta(G)$ lorsque G est un cycle à 5 sommets. Donner sa matrice.

7. Montrer, dans le cas général, que si G n'a pas de triangle, $\theta(G)$ non plus.

8. Montrer, dans le cas général, que si G est k-chromatique, $\theta(G)$ est $(k+1)$-chromatique.

Exercice 2. *Ramsey*

Dans cette partie, on colorie les *arêtes* et non les sommets. On s'intéresse uniquement à une coloration avec deux couleurs de graphe complet (tous les sommets sont reliés les uns aux autres).

1. Montrer qu'un graphe complet (clique) à au moins 6 sommets colorié avec deux couleurs contient soit un triangle rouge soit un triangle bleu. On pourra commencer par montrer qu'il existe un sommet qui, à renommage des couleurs près, a trois arêtes bleues.

2. En déduire que dans tout groupe de six personnes, il y en a soit trois qui se connaissent mutuellement, soit trois qui ne se connaissent pas entre elles.

On note $R(p,q)$ le plus petit nombre n tel que tout graphe à au moins n sommets contient soit une clique bleue de taille p soit une clique rouge de taille q. Si jamais un tel nombre n'existe pas, on pose $R(p,q) = +\infty$. Evidemment, $R(p,q) = R(q,p)$.
Attention : une clique bleue de taille p est une clique de taille p où toutes les *arêtes* sont bleues (puisqu'on colorie les arêtes et pas les sommets)

3. Calculer la valeur de $R(1,q)$ et $R(p,1)$.

4. Calculer la valeur de $R(2,q)$ et $R(p,2)$.

5. Calculer la valeur de $R(3,3)$.

6. Montrer que $R(p,q) \leq R(p-1,q) + R(p,q-1)$. (Prendre un sommet v au hasard, et séparer les autres en deux parties : ceux pour lesquels l'arête qui les relie à v est bleue, et ceux pour lesquels elle est rouge)

7. En déduire que $R(p,q)$ est toujours bien défini. Donner une borne pour $R(4,4)$.

TD n° 2

Formules propositionnelles

On définit les formules sur n variables $x_1 \ldots x_n$ de la façon suivante :

- Les constantes 0 et 1 peuvent être vues comme des formules
- Une variable x_i peut être vue comme une formule, qu'on appellera temporairement θ. Dans ce cas, $\theta(x_1 \ldots x_n) = 1$ si et seulement si $x_i = 1$.
- Si φ et ψ sont des formules, alors $\theta = \varphi \wedge \psi$ est une formule. $\theta(x_1 \ldots x_n) = 1$ si et seulement si $\varphi(x_1 \ldots x_n) = 1$ et $\psi(x_1 \ldots x_n) = 1$.
- Si φ et ψ sont des formules, alors $\theta = \varphi \vee \psi$ est une formule. $\theta(x_1 \ldots x_n) = 1$ si et seulement si $\varphi(x_1 \ldots x_n) = 1$ ou $\psi(x_1 \ldots x_n) = 1$.
- Si φ est une formule, alors $\theta = \neg\varphi$ est une formule. $\theta(x_1 \ldots x_n) = 1$ si et seulement si $\varphi(x_1 \ldots x_n) = 0$.

Attention, il ne faut pas confondre une formule et sa valeur : les formules $\varphi = x \wedge (x \wedge y)$ et $\psi = x \wedge y$ ne sont pas les mêmes (elles ne s'écrivent pas pareil), mais ont la même valeur : pour tout (x,y), $\psi(x,y) = \varphi(x,y)$. Lorsqu'une telle égalité survient, on dit que ψ et φ sont équivalentes. Une formule φ à n variables est dite satisfaisable si on peut trouver un choix de $x_1 \ldots x_n$ tel que $\varphi(x_1 \ldots x_n) = 1$.

Exercice 1.

1. Montrer que la formule $\psi_0 = (x \vee y) \wedge (\neg(z \wedge \neg y))$ est satisfaisable. Donner tous les x,y,z,w tels que $\psi_0(x,y,z,w) = 1$.

2. Soit $f : \{0,1\}^n \to \{0,1\}$. Montrer que :

$$f(x_1,\ldots,x_{n-1},x_n) = (f(x_1,\ldots,x_{n-1},0) \wedge \neg x_n) \vee (f(x_1,\ldots,x_{n-1},1) \wedge x_n).$$

3. En déduire que toute fonction de $\{0,1\}^n$ dans $\{0,1\}$ est équivalente à une formule.

Par définition, un littéral est soit une variable x_i soit la négation d'une variable $\neg x_i$ qu'on note plus souvent $\overline{x_i}$. Une formule est dite en forme normale disjonctive (DNF) si elle s'écrit comme un "ou" ($\vee$) de "et" ($\wedge$) de littéraux, c'est à dire s'il existe des littéraux $\ell_{i,j}$ tels que $\varphi = \bigvee_i (\bigwedge_j \ell_{i,j})$.

4. Montrer que toute formule est équivalente à une formule sous forme normale disjonctive. Donner une formule normale disjonctive pour la formule ψ_0.

Une formule est dite en forme normale conjonctive (CNF) si elle s'écrit comme un "et" ($\wedge$) de "ou" ($\vee$) de littéraux, c'est à dire s'il existe des littéraux $\ell_{i,j}$ tels que $\varphi = \bigwedge_i (\bigvee_j \ell_{i,j})$.

5. Montrer que toute formule est équivalente à une formule sous forme normale conjonctive. Donner une formule normale conjonctive pour la formule ψ_0.

6. Montrer qu'on peut construire pour toute formule φ à n variables, un polynôme P tel que $\varphi(x_1 \ldots x_n) = 1$ si et seulement si $P(x_1 \ldots x_n) = 0$.

7. Montrer qu'on peut construire pour toute formule φ à n variables, un polynôme P tel que φ est satisfaisable si et seulement si le polynôme P a une racine dans $\mathbb{R}$, plus précisément tels que φ soit satisfaisable d'autant de façons que P n'a de racines. (Attention, ce n'est pas une conséquence directe de la question précédente. Par exemple, pour $\varphi(x,y) = x \wedge y$, on peut choisir dans la question précédente $P(x,y) = xy - 1$. Dans ce cas, φ est satisfaisable uniquement pour $x = y = 1$. En revanche $P(x,y) = 0$ pour $x = y = 1$ mais aussi pour $x = 2$ et $y = 1/2$. Dans $\mathbb{R}$, les variables ne valent pas seulement 0 ou 1 !)

Exercice 2.
On considère ici une formule ψ à 32 variables. Les variables s'appellent $x_0 \ldots x_{15}$ et $y_0 \ldots y_{15}$. Pour $i \neq j$, on note $\varphi_{i,j}$ la formule : $\varphi_{i,j} = (x_i \wedge \overline{x_j}) \vee (\overline{x_i} \wedge x_j) \vee (y_i \wedge \overline{y_j}) \vee (\overline{y_i} \wedge y_j)$.

1. Vérifier que $\varphi_{i,j}(x_i, x_j, y_i, y_j) = 1$ si et seulement si $(x_i, y_i) \neq (x_j, y_j)$

On considère l'ensemble suivant :

$$\begin{aligned} I = \quad & \{(i,j) \text{ t.q. } i \neq j \text{ et } i \equiv j \, mod \, 4]\} \bigcup \{(i,j) | i \neq j \text{ et } i/4 = j/4\} \bigcup \\ & \{(0,5),(2,7),(8,13),(10,15),(4,1),(3,6),(9,12),(11,14)\} \end{aligned}$$

(Ici, la division $(i/4 = j/4)$ est à comprendre comme une division entière. Par exemple, on a bien $11/4 = 9/4$.)

2. Montrer que la formule ψ suivante est satisfaisable (on calculera *toutes* les solutions) :

$$\psi = \bigwedge_{(i,j) \in I} \varphi_{i,j} \wedge (x_0 \wedge y_0) \wedge (\neg y_3)) \wedge (\neg x_6) \wedge x_{12}.$$

3. On pose $z_i = x_i + 2 * y_i$. Pour chacune des solutions de la formule ψ écrire les nombres z_i qui correspondent dans la grille suivante :

z_0	z_1	z_2	z_3
z_4	z_5	z_6	z_7
z_8	z_9	z_{10}	z_{11}
z_{12}	z_{13}	z_{14}	z_{15}

Que constatez-vous ? Pouvez-vous l'expliquer ?

TD n° 3

Machines de Turing

Exercice 1. *Trois résultats du cours*

On note $\sim$ la relation d'équipotence entre ensembles et $\mathscr{P}(X)$ l'ensemble des parties d'un ensemble X.

1. Montrer que pour tout ensemble, $X \not\sim \mathscr{P}(X)$.

Soit Σ un alphabet fini.

2. Montrer que Σ^* est dénombrable (c'est-à-dire, équipotent à $\mathbb{N}$).
3. Montrer que l'ensemble $\mathscr{P}(\Sigma^*)$ des langages sur Σ n'est ni fini, ni dénombrable.

Exercice 2. *Codage de* SAT

Construire un codage du problème SAT (satisfaisabilité d'une formule propositionnelle) sur l'alphabet $\{0,1\}$.

Exercice 3.

On considère la machine de Turing $\mathscr{M} = (\Sigma, \Gamma, Q, q_0, q_a, q_r, \delta)$ définie par : $\Sigma = \{0,1\}$; $\Gamma = \{0,1,\mathrm{B}\}$; $Q = \{q_0, q_1 q_a, q_r\}$ et :

$$\begin{array}{l|l}
\delta(q_0,B) = (q_0,B,\rightarrow) & \delta(q_1,B) = (q_a,B,\downarrow) \\
\delta(q_0,0) = (q_0,0,\downarrow) & \delta(q_1,0) = (q_0,0,\downarrow) \\
\delta(q_0,1) = (q_1,1,\rightarrow) & \delta(q_1,1) = (q_1,1,\rightarrow)
\end{array}$$

1. La suite de configurations $(q_0, \underline{\mathrm{B}}110)$; $(q_0, \mathrm{B}\underline{1}10)$; $(q_1, \mathrm{B}1\underline{1}0)$; $(q_1, \mathrm{B}11\underline{0})$; $(q_0, \mathrm{B}11\underline{0})$ est-elle une dérivation de $\mathscr{M}$?
2. Quel est le langage reconnu par $\mathscr{M}$?
3. Ce langage est-il décidable ?

Exercice 4. *Trois machines de Turing*

Construire des machines de Turing qui calculent :

1. L'ensemble des mots sur $\{0,1\}$ comportant un nombre pair de 1 ;
2. la fonction successeur qui envoie la représentation binaire d'un entier n sur la représentation binaire de $n+1$;
3. le langage $\{a^n b^n c^n,\ n \in \mathbb{N}\}$.

Exercice 5. *Composition de MT*

Soient $\mathscr{M}_1$, $\mathscr{M}_2$ deux machines de Turing de même alphabet d'entrée Σ et $f_1, f_2 : \Sigma^* \rightarrow \Sigma^*$ les fonctions partielles calculées par ces machines. Construire une MT qui calcule $f_2 \circ f_1$.

Exercice 6. *Machines "oublieuses"*

Dans cet exercice, on n'est pas obligés de toujours donner la table de transition des machines présentées : une explication rigoureuse suffit. Trouver la manière et estimer le temps nécessaire pour simuler une machine de Turing :

1. à k rubans par une machine à un ruban (bonus : peut-on faire mieux avec 2 rubans ?) ;
2. travaillant sur un alphabet Γ par une machine travaillant sur l'alphabet $\{0,1,\square\}$;
3. à ruban bi-infini par une machine à ruban mono-infini.

Une machine de Turing est dite *oublieuse*[1] si sur une entrée x, la position de la tête de lecture à l'instant i ne dépend que de i et de $|x|$.

4. Expliquer comment simuler une machine de Turing par une machine oublieuse, et estimer le temps de la simulation. Comment peut-on simuler une machine de Turing fonctionnant en temps (constructible) T par une machine oublieuse fonctionnant en temps $O(T^2)$? Et en $O(T\log T)$?

Exercice 7. *Ensembles dénombrables*

1. Soient A et B deux ensembles **infinis** et $f : A \to B$. Démontrer les implications suivantes :
   (a) ($A \sim \mathbb{N}$ et f surjective) $\Rightarrow$ ($B \sim \mathbb{N}$) ;
   (b) ($B \sim \mathbb{N}$ et f injective) $\Rightarrow$ ($A \sim \mathbb{N}$) ;
   (c) ($B \sim \mathbb{N}$ et $A \subseteq B$) $\Rightarrow$ ($A \sim \mathbb{N}$).
2. Montrer que l'ensemble des parties finies de $\mathbb{N}$ est dénombrable.

Exercice 8. *Dieu est-il décidable ?*

Soit L le langage restreint à l'unique mot u, défini comme suit :

$$u = \begin{cases} 1 \text{ si Dieu existe ;} \\ 0 \text{ sinon.} \end{cases}$$

Ce langage est-il décidable ?

1. *Oblivious*, en anglais.

TD n° 4

Le temps polynômial

Exercice 1. *Une machine à shift*

1. **1.** Décrire une machine de Turing d'alphabet d'entrée $\{a,b\}$ qui "shift" son entrée (*i.e.* qui remplace le mot $\mathrm{B}w_1\dots w_n$ initialement écrit sur son ruban par le mot $\mathrm{BB}w_1\dots w_n$).
2. **2.** Évaluer la complexité de cette machine.

Nous verrons bientôt que selon une conjecture communément admise, le problème SAT *(satisfaisabilité d'une formule propositionnelle) n'est pas polynômial. Dans les deux exercices qui suivent, nous allons montrer que deux sous-problèmes de* SAT *sont dans* P*. Rappelons d'abord quelques définitions :*
- *Un* littéral *est une formule propositionnelle de l'une des forme p ou $\overline{p}$, où p est une variable propositionnelle. (On note $\overline{p}$ pour $\neg p$.)*
- *Une* clause *est une disjonction de littéraux.*

Exercice 2. HORN-SAT $\in$ P

1. **1.** Une *clause de Horn* est une clause comportant au plus un littéral positif. Autrement dit, c'est une formule propositionnelle de la forme $\overline{p}_1 \vee \cdots \vee \overline{p}_{n-1} \vee p_n$ (ce qui s'écrit aussi : $(p_1 \wedge \cdots \wedge p_{n-1}) \rightarrow p_n$) ou de la forme p (simple variable propositionnelle). Montrez que le problème suivant est dans P :

   HORN-SAT
   Entrée : Une conjonction Φ de clauses de Horn ;
   Question : Φ est-elle satisfaisable ?

Exercice 3. 2-SAT $\in$ P

Une *2-clause* est une clause comportant au plus deux littéraux (*e.g.*, $p \vee q$, $\neg p \vee q$, $\neg p$, etc). Montrez que le problème suivant est dans P :

2-SAT
Entrée : Une conjonction Φ de 2-clauses ;
Question : Φ est-elle satisfaisable ?

Exercice 4.

Montrer que les problèmes suivants sont dans P.

PREMIERS 2 À 2
Entrée : deux entiers n et p.
Question : n et p sont-ils premiers entre eux ?

VERIF-SAT

Entrée :	une formule propositionnelle φ et une valuation v sur les variables de φ.
Question :	v satisfait-elle φ ?

Exercice 5. P *et la complémentation*

Montrer que la classe P est close par complémentation (*i.e.* si un langage L est dans P alors son complémentaire $\overline{L}$ est dans P).

Exercice 6. 3-COLOR $\leq_{pol}$ CLAUSE-SAT

Un graphe $G = (S, A)$ est dit *3-coloriable* s'il existe une application $c : S \to \{R, V, B\}$ telle que pour toute arête $(x, y) \in A$, $c(x) \neq c(y)$. On considère les problèmes suivants :

3-COLOR

Entrée :	Un graphe G ;
Question :	G est-il 3-coloriable ?

CLAUSE-SAT

Entrée :	Une conjonction Φ de clauses propositionnelles ;
Question :	Φ est-elle satisfaisable ?

Montrer que 3-COLOR $\leq_{\text{pol}}$ CLAUSE-SAT.

Exercice 7. 3-SAT $\leq_{pol}$ CLIQUE

Une *clique* dans un graphe $G = (S, A)$ est un ensemble de sommets deux à deux reliés dans G. Une k-clique est une clique de taille k. On considère les problèmes suivants :

CLIQUE

Entrée :	Un graphe G et un entier $k \geq 1$;
Question :	G admet-il une k-clique ?

3-SAT

Entrée :	Une conjonction Φ de 3-clauses propositionnelles ;
Question :	Φ est-elle satisfaisable ?

Montrer que 3-SAT $\leq_{\text{pol}}$ CLIQUE.

Exercice 8. DNF-SAT $\in$ P

On appelle DNF-SAT la restriction de SAT aux formules écrites sous forme normale disjonctive :

DNF-SAT

Entrée :	Une formule propositionnelle DNF Φ ;
Question :	Φ est-elle satisfaisable ?

1. Montrer DNF-SAT est polynômial.
2. La procédure standard de mise sous forme normale disjonctive d'une formule propositionnelle est-elle une réduction polynômiale de SAT à DNF-SAT ?