20 novembre 2017

Feuille d'exercices
Complexité

La taille d'un mot w (nombre de symboles) est notée $\mid w \mid$. Le logarithme en base 2 de x est noté $\log x$.

Machines de Turing

Exercice 1. Définir explicitement une machine de Turing qui n'utilise pas de ruban de travail :

1. qui la teste la parité d'un entioer écrit en binaire ;
2. qui décrémente de 1 un entier ;
3. qui incrémente de 1 un entier (indiquer comment modifier la précédente).

Exercice 2. Indiquer comment construire une machine de Turing qui calcule la longueur de la représentation binaire d'un entier n $(1+\llcorner \log_2 n \lrcorner)$, utiliser l'exercice précédent (on obtient naturellement une machine fonctionnant en espace logarithmique).

Espace logarithmique

Exercice 3.

1. Indiquer comment construire une machine de Turing qui recopie la seconde entrée sur une bande de travail puis calcule l'addition de deux entiers en binaire par l'algorithme du primaire (quelles sont les complexités en temps et en espace de la machine ?)
2. Indiquer comment construire une machine de Turing qui calcule l'addition en espace logarithmique (indication : maintenir sur une bande de travail un indice qui donne la position des bits à additionner).

Exercice 4. Indiquer comment calculer la multiplication de deux entiers en binaire en espace logarithmique.

Exercice 5. On a vu que la composition $g \circ f$ de deux fonctions $f : \Sigma_1^* \to \Sigma_2^a s$ et $g : \Sigma_2^* \to \Sigma_3^*$ calculables en temps polynomial était calculable en temps polynomial. Le but de l'exercice est de montrer qu'on a ce résultat pour l'espace logarithmique. D'après les questions précédentes, les fonctions polynomiales se calculent donc en espace logarithmique.

Soit donc M_f et M_g deux machines qui calculent f et g en espace logarithmique.

1. Pourquoi la méthode utilisée pour le temps polynomial ne fonctionne-t-elle pas pour l'espace logarithmique ? Pourquoi $\log |f(x)| = O(\log |x|)$?
2. Décrire la construction à partir de M_f d'une machine M'_f qui prend en entrée $x \in \Sigma_1^*$ et un entier i et calcule, en espace $O(\log |x|)$, le i-ème symbole de $f(x)$.
3. Utiliser M'_f et M_g pour décrire une machine M qui calcule $g \circ f$ en espace logarithmique.

Classes P et NP

Exercice 6. Montrer directement (sans utiliser le Théorème de Cook) que le problème de l'existence d'un chemin Hamiltonien (chemin passant exactement une fois par chaque nœud) dans un graphe orienté se réduit en temps polynomial au problème SAT . Pour un graphe à n nœuds notés $\{1, \cdots, n\}$. On peut introduire, pour coder une suite de nœuds de longueur n $(u_1, \cdots, u_n)$ une relation binaire "R", "Rij" signifiant "$u_i = j$", et axiomatiser propositionnellement le fait que la suite de nœuds représentée par R est un chemin Hamiltonien.

Exercice 7 (SAT et 3-SAT). On considère divers problèmes de satisfaisabilité,

SAT ' Décider si une formule propositionnelle avec connecteurs dans $\{\wedge, \vee \neg, \to\}$, est satisfaisable.
SAT Décider si un ensemble de clauses est satisfaisable.
3-SAT Décider si un ensemble de clauses, chacune contenant au plus 3 littéraux, est satisfaisable.

On rappelle qu'un problème Pb1 (de décision d'appartenance au langage L_1) se réduit en temps polynomial à un Pb2 (de décision d'appartenance au langage L_2) signifie qu'il existe une fonction f calculable en temps polynomial telle que $x \in L_1$ ssi $f(x) \in L_2$.

1. Montrer que SAT (et donc 3-SAT) se réduit à SAT ' en temps polynomial.

2. On montre à cette question que le problème SAT ' se réduit au problème 3-SAT en temps polynomial. Soit $\mathcal{F}$ l'ensemble des formules construites sur un ensemble de variables propositionnelles $\mathcal{P}$ avec les connecteurs $\{\wedge, \vee \neg, \rightarrow\}$.
On construit un nouvel ensemble de variables propositionnelles $\tilde{\mathcal{P}} = \{c_F \ / \ F \in \mathcal{F}\}$, où les constantes c_F sont des constantes toutes distinctes entre elles et telles que si $\alpha \in \mathcal{P}$, alors $c_\alpha = \alpha$ (donc $\mathcal{P} \subset \tilde{\mathcal{P}}$ et si F n'est pas atomique, $c_F \notin \mathcal{P}$).
A chaque valuation sur $\mathcal{P}$, on associe une valuation $\tilde{v}$ définie sur $\tilde{\mathcal{P}}$ (et qui prolonge v) définie par :

$$\tilde{v}(c_F) = v(F)\,.$$

a. Montrer que, pour toute formule F, pour toute valuation v sur $\mathcal{P}$, il existe un ensemble $\mathcal{C}_F$ de clauses contenant chacune au plus 3 littéraux, tel que

$$\begin{array}{l} \forall v \text{ valuation définie sur } \mathcal{P}\ \tilde{v}(\mathcal{C}_F) = 1 \\ \forall v' \text{ valuation définie sur } \tilde{\mathcal{P}}\ \big(v'(\mathcal{C}_F) = 1 \Rightarrow v'(c_F) = v'(F)\big) \end{array}$$

On construira $\mathcal{C}_F$ en prenant la réunion pour chaque occurrence de sous-formule "$A\,?\,B$" et "$\neg E$" dans F des équivalences "$c_A\,?\,c_B \leftrightarrow c_{A\,?\,B}$" et $c_{\neg E} \leftrightarrow \neg c_E$ mises sous forme clausale (définir $\mathcal{C}_F$ par induction, si F est une variable propositionnelle, $\mathcal{C}_F$ est vide).

b. Montrer que F est satisfaisable si et seulement si $\mathcal{C}_F \cup \{c_F\}$ est satisfaisable.

c. Montrer que de plus la taille de $\mathcal{C}_F$ est linéaire en la longueur de F, et que $\mathcal{C}_F$ se calcule en temps linéaire à partir de F.

d. En déduire que SAT ' se réduit à 3-SAT en temps linéaire.

3. Montrer que SAT ', SAT , 3-SAT se réduisent l'un à l'autre en temps polynomial, puis que SAT ' et 3-SAT sont NP-complets.

Exercice 8. Montrer que la recherche de la validité d'une formule est un problème coNP-complet.
On a vu à l'exercice 7 un algorithme de mise sous forme clausale (forme normale conjonctive-disjonctive) par *équisatisfaisabilité* qui est linéaire en temps. Que penser de la complexité d'un algorithme de mise sous forme clausale par *équivalence logique* ?

Exercice 9 (cas particuliers NP-complets de 3-SAT).

1. Montrer que le problème de la satisfaisabilité pour un ensemble de clauses, chacune contenant exactement 3 littéraux est NP-complet.

2. Montrer que le problème de la satisfaisabilité pour un ensemble de clauses, chacune contenant exactement 3 littéraux distincts est NP-complet.

3. Montrer que le problème de la satisfaisabilité pour un ensemble de clauses contenant chacune au plus 3 littéraux, et tel que chaque variable possède au plus 3 occurrences distinctes et chaque littéral au plus 2 occurrences distinctes dans l'ensemble considéré est NP-complet.

Exercice 10 (satisfaisabilité des clauses de Horn). La première question est un rappel sur la méthode de résolution en calcul propositionnel. Vous pouvez passer directement à la seconde question.

Soit $\mathcal{P} = \{p_n \ / \ n \in \mathbb{N}\}$ un ensemble de variables propositionnelles. On appelle *clause* un couple de deux ensembles finis de variables propositionnelles, soient Γ et Δ, et on note la clause $\Gamma \longrightarrow \Delta$. Si $\Gamma = \{a_1, \ldots, a_n\}$, et $\Delta = \{b_1, \ldots, b_p\}$, on note aussi la clause : $a_1, \ldots, a_n \longrightarrow b_1, \ldots, b_p$. La clause $\Gamma \longrightarrow \Delta$ s'interprète sémantiquement comme la formule $\bigvee_{p\in\Gamma} \neg p \vee \bigvee_{q\in\Delta} q$, ou encore, $\bigwedge_{p\in\Gamma} p \rightarrow \bigvee_{q\in\Delta} q$. la *clause vide*, $\longrightarrow$, s'interprète sémantiquement par l'absurde.

La *partie positive* de $\Gamma \longrightarrow \Delta$ désigne Δ, la *partie négative* $\Gamma \longrightarrow \Delta$ désigne Γ. Un littéral de la clause $\Gamma \rightarrow \Delta$ est une variable propositionnelle de Δ, ou la négation d'une variable propositionnelle de Γ.

Une *clause de Horn* est une clause dont la partie positive contient au plus une variable propositionnelle. Une clause positive est une clause dont la partie négative est vide.

Une valuation v satisfait une clause $\Gamma \longrightarrow \Delta$ signifie qu'il existe une variable propositionnelle p telle que : $p \in \Gamma$ et $v(p) = 0$, ou $p \in \Delta$ et $v(p) = 1$.

Un ensemble de clauses $\mathcal{S}$ est *satisfaisable* signifie qu'il existe une valuation qui satisfait toutes les clauses de $\mathcal{S}$.

La *règle de coupure* associe à deux clauses une nouvelle clause selon le schéma suivant :

$$\frac{\Gamma_1 \cup \{p\} \longrightarrow \Delta_1 \quad \Gamma_2 \longrightarrow \Delta_2 \cup \{p\}}{\Gamma_1 \cup \Gamma_2 \longrightarrow \Delta_1 \cup \Delta_2}\ .$$

On définit inductivement la déduction par coupures. une clause C se déduit par coupures d'un ensemble de clauses $\mathcal{S}$ ssi :

— $C \in \mathcal{S}$

— C est obtenue par une règle de coupure à partie de clauses qui se déduisent par coupures de $\mathcal{S}$.

Un ensemble de clauses est *réfutable par coupures* signifie que la clause vide se déduit de $\mathcal{S}$.

1. Quelques généralités.

- **a.** Montrer que la règle de coupure est sémantiquement correcte : si C se déduit par coupures à partir de $\mathcal{S}$ est $\mathcal{S}$ est satisfaisable, alors C est satisfaisable. En déduire que si la clause vide se déduit par coupures de $\mathcal{S}$, alors $\mathcal{S}$ n'est pas satisfaisable.
- **b.** Montrer que si un ensemble de clauses n'est pas satisfaisable il contient nécessairement une clause positive.
- **c.** Soit $\mathcal{S}$ est un ensemble de clauses et p une variable propositionnelle. On suppose que p n'apparaît à la fois positivement et négativement dans aucune clause de $\mathcal{S}$. On note $res(\mathcal{S}, p)$ l'ensembles des clauses obtenues par une seule règle de coupure sur p entre deux clauses de $\mathcal{S}$. On note $\mathcal{S}_p$ l'ensemble de clauses qui contiennent une occurrence de la variable propositionnelle p (positivement ou négativement).
  - i. Montrer que $\mathcal{S}$ est satisfaisable ssi $(\mathcal{S} - \mathcal{S}_p) \cup res(\mathcal{S}, p)$ est satisfaisable.
  - ii. En déduire que la méthode de réfutation par coupures est complète (pour un ensemble fini de clauses) : si un ensemble fini de clauses n'est pas satisfaisable, alors il est réfutable par coupures.
  - iii. La preuve ci-dessus induit un algorithme pour la satisfaisabilité d'un ensemble fini de clauses. Montrer qu'il est en général en temps exponentiel (évaluer la taille de $res(\mathcal{S}, p)$ en fonction de la taille de $\mathcal{S}$). On verra que la satisfaisabilité d'un ensemble de clauses est un problème NP-complet.

2. Le cas des clauses de Horn : $\mathcal{S}$ désigne un ensemble de clauses de Horn.

- **a.** Montrer que le fait d'être une clause de Horn est stable par coupures.
- **b.** On reprend les notations de la question précédente. Soit $\longrightarrow p$ une clause positive de $\mathcal{S}$. Soit $res^+(\mathcal{S}, p)$ l'ensembles des clauses obtenues par une seule règle de coupure entre $\longrightarrow p$ et une clause de $\mathcal{S}$ qui ne contient p qu'en position négative. Montrer que $\mathcal{S}$ est satisfaisable ssi $(\mathcal{S} - \mathcal{S}_p) \cup res^+(\mathcal{S}, p)$ est satisfaisable.
- **c.** En déduire un algorithme pour la satisfaisabilité d'un ensemble de clauses de Horn en temps quadratique en fonction de l'entrée.

Exercice 11 (2-SAT). On appelle 2-SAT le problème de la satisfaisabilité d'un ensemble de clauses contenant chacune *au plus 2* littéraux. On montre que ce problème est P. On utilise en particulier la stabilité par règle de coupure de l'ensemble des clauses à 2 littéraux.

1. Combien peut-on écrire de clauses distinctes contenant au plus 2 littéraux avec n variables ?

2. Donner un algorithme polynomial en fonction du nombre de variables pour 2-SAT (on peut procéder par élimination successive de chaque variable, s'inspirer de l'algorithme donné pour les clauses de Horn).

Exercice 12. Dans cet exercice tous les graphes sont non-orientés. Soit G un graphe, $G = (S, A)$, où S est l'ensemble des sommets du graphe, A l'ensemble des arêtes du graphe, on doit avoir $A \subset \mathcal{P}_2(S)$ où $\mathcal{P}_2(S)$ est l'ensemble des paires de sommets (sous-ensembles de S à un ou deux éléments).

On étudie les trois problèmes suivants, pour un graphe $G = (S, A)$ et un entier k ($k \leq |S|$).

i. Existe t-il un ensemble de nœuds indépendant de cardinalité k ?
Un sous-ensemble I de S est indépendant signifie que

$$\forall i,\, j \in I\, \{i,\, j\} \notin A\,.$$

ii. Existe t-il une clique de cardinalité k ?
Un sous-ensemble C de S est une clique signifie que

$$\forall i,\, j \in C\, \{i,\, j\} \in A\,.$$

iii. Existe t-il un recouvrement de G de cardinalité k ?
Un sous-ensemble R de S est un recouvrement (par sommets) de G signifie que

$$\forall\{i,\, j\} \in\ A\ (i \in R \text{ ou } j \in R).$$

On va montrer que ces 3 problèmes sont NP-complets.

1. Montrer que ces 3 problèmes se réduisent l'un à l'autre (entre (i) et (ii) passer au complémentaire pour A, entre (i) et (iii) passer au complémentaire pour l'ensemble cherché).

2. Montrer que le problème 3-SAT se réduit en temps polynomial au problème de la recherche d'un ensemble indépendant.
Indication. Soit $\mathcal{C}$ un ensemble de clauses d'au plus 3 littéraux, chaque clause est vue comme un ensemble de variables propositionnelles ou négations de variable propositionnelle. On note $\bar{x}$ le littéral « opposé » de x (si $x = \alpha$ alors $\bar{x} = \neg\alpha$, si $x = \neg\alpha$ alors $\bar{x} = \alpha$, α étant une variable propositionnelle).
On associe à $\mathcal{C}$ le graphe G dont l'ensemble de sommets est

$$S = \big\{(C, x)\ /\ C \in \mathcal{C} \text{ et } x \in C\big\}$$

dont l'ensemble des arêtes est

$$A = \{\{(C, x),\, (D, y)\}\ /\ (C = D \text{ et } x \neq y) \text{ ou } x = \bar{y}\}$$

En déduire que les 3 problèmes introduits au début de l'exercice sont NP-complets.

Exercice 13 (NAESAT). On s'interesse au problème de décision suivant nommé NAESAT (pour Not All Equal SAT). Cet exercice utilise des définitions et résultats de l'exercice 7.

Donnée : Un ensemble fini de clauses d'exactement 3 littéraux $\mathcal{C}$ sur un ensemble de variables propositionnelles $\mathcal{P}$.
Question : Existe t-il une valuation v telle que $v(\mathcal{C}) = 1$ et $\overline{v}(\mathcal{C}) = 1$ (où $\overline{v}$ est définie par $\forall p \in \mathcal{P}\ \overline{v}(p) = 1 - v(p)$) ?

Nous dirons que $\mathcal{C}$ est NAE-*satisfaisable* s'il existe une telle valuation.

On construit une réduction du problème SAT ' introduit à l'exercice 7 à NAESAT en associant à une formule propositionnelle F un ensemble de clauses $\mathcal{C}'_F$, obtenu à partir des clauses de l'ensemble $\mathcal{C}_F \cup \{c_F\}$ construit à l'exercice 7 en ajoutant à toutes les clauses de cet ensemble qui ont deux littéraux une même variable propositionnelle z qui n'apparaît pas dans $\mathcal{C}_F \cup \{c_F\}$, et en remplaçant la seule clause c_F qui a un seul littéral, par les deux clauses $c_F \vee z \vee z'$ et $c_F \vee z \vee \neg z'$ pour une nouvelle variable z' qui n'apparaît pas dans $\mathcal{C}_F \cup \{c_F\} \cup \{z\}$.

1. Montrer que si $\mathcal{C}'_F$ est NAE-satisfaisable par v alors v ou $\overline{v}$ satisfait $\mathcal{C}_F \cup \{c_F\}$ et donc F.

2. Montrer que si F est satisfaisable par v, alors $\mathcal{C}'_F$ est satisfaisable par une valuation v' obtenue en prolongeant v en z par $v(z) = 0$ (la valeur en z' n'a pas d'importance).

3. En déduire une réduction de SAT ' à NAESAT en temps polynomial, puis montrer que NAESAT est NP-complet.

Exercice 14 (3-colorabilité). On veut montrer que le problème suivant dit de la *3-colorabilité* est NP-complet.

Donnée : Un graphe G ($G = (E, R)$ avec $R \subset E^2$).
Question : Existe t-il une fonction f de E dans $\{0, 1, 2\}$ telle que $\forall(a, b) \in R\ f(a) \neq f(b)$?

On veut montrer que NAESAT (voir exercice 13) se réduit à la 3-colorabilité en temps polynomial.

Pour cela on associe à un ensemble de 3-clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ construit sur l'ensemble de variables propositionnelles $\mathcal{P} = \{p_1, \ldots, p_n\}$ le graphe G suivant. Soit a arbitraire (non dans $\mathcal{P}$). L'ensemble de ses sommets est :

$$E = \{a\} \cup \mathcal{P} \cup \{\neg x\ /\ x \in \mathcal{P}\} \cup \{(C, \alpha)\ /\ C \in \mathcal{C} \text{ et } \alpha \text{ est un littéral de } C\}.$$

L'ensemble des arêtes est (remarquer que l'orientation n'a pas d'importance) $R = R_1 \cup R_2 \cup R_3$ avec :

$$R_1 = \{(a, x)\ /\ x \in \mathcal{P}\} \cup \{(a, \neg x)\ /\ x \in \mathcal{P}\} \cup \{(x, \neg x)\ /\ x \in \mathcal{P}\}$$

$$R_2 = \{((C, \alpha), (C, \beta))\ /\ C \in \mathcal{C} \text{ et } \alpha, \beta \text{ sont deux littéraux distincts de } C\}$$

$$R_3 = \{((C, \alpha), \alpha)\ /\ (C, \alpha) \in E\}.$$

1. Montrer que si $\mathcal{C}$ est NAE-satisfaisable pour une valuation v alors celle-ci se prolonge sur E en une fonction de coloration convenable pour la 3-colorabilité de G (prendre $f(a) = 2$).

2. Montrer que si G 3-coloriable par f, alors en permutant éventuellement les couleurs de façon que $f(a) = 2$, f restreinte à P est une valuation convenable pour la NAE-satisfaisabilité de $\mathcal{C}$.

3. En deduire que NAESAT se réduit à la 3-colorabilité en temps polynomial, puis montrer que la 3-colorabilité est un problème NP-complet.

Exercice 15 (Problème des mariages). Soit p un entier. On se donne p ensembles finis $E_1, \dots, E_p$ de même cardinal n, et un sous-ensemble A de $E_1 \times \cdots \times E_p$. Un *mariage à* p de $(E_1, \cdots, E_p, R)$ est un sous-ensemble M de A vérifiant

$$\forall (u_1, \dots, u_p) \in M \, \forall (v_1, \dots, v_p) \in M \;\; [(u_1, \dots, u_p) \neq (v_1, \dots, v_p) \Rightarrow u_1 \neq v_1 \text{ et} \dots u_p \neq v_p]$$

Un mariage à p de $(E_1, \cdots, E_p, A)$ est *parfait* si sa cardinalité est n.

Le *problème* des mariages à p est celui de l'existence d'un mariage parfait pour $(E_1, \cdots, E_p, A)$ tel que ci-dessus. Le nom du problème se comprend bien pour $p = 2$. Pour $p = 3$ on peut penser à un couple et son logement.

On va montrer que pour $p \geq 3$ ce problème est NP-complet (le problème se résout en temps polynomial pour $p = 2$ voir exercice 16).

1. Montrer que ce problème est NP, et que si pour $p = 3$ il est NP-complet, alors il est NP-complet pour $p \geq 3$.

2. On montre maintenant que le problème des mariages à 3 est NP-complet, par réduction du problème SAT . Soit $\mathcal{C} = \{C_1 \cdots, C_k\}$ un ensemble de clauses, soit $\{\alpha_1, \cdots, \alpha_l\}$ l'ensemble de toutes les variables qui apparaissent dans les clauses de $\mathcal{C}$.

On construit 3 ensembles $G,\ F,\ L$ disjoints de la façon suivante.

L contient pour chaque clause C_i une copie de chaque littéral x noté x^i :

$$L = \bigcup_{i=1}^{k} \bigcup_{j=1}^{l} \{\alpha_j^i, \neg\alpha_j^i\}\,.$$

G contient une copie de chaque clause C_i, notons la g_i, et une copie de chaque variable α_j pour chaque clause c_i notée g_j^i, complété de façon à avoir un cardinal égal à celui de L :

$$G = \bigcup_{i=1}^{l} \{g_i\} \cup \bigcup_{i=1}^{k} \bigcup_{j=1}^{l} \{g_j^i\} \cup \bigcup_{i=1}^{k-1} \bigcup_{j=1}^{l} \{g_j^{'i}\}$$

F a la même structure que G :

$$F = \bigcup_{i=1}^{l} \{f_i\} \cup \bigcup_{i=1}^{k} \bigcup_{j=1}^{l} \{f_j^i\} \cup \bigcup_{i=1}^{k-1} \bigcup_{j=1}^{l} \{f_j^{'i}\}$$

L'ensemble de triplets A est la réunion des ensembles A_1, A_2 et A_3,

$A_1 = \bigcup_{i=1}^{k} \bigcup_{j=1}^{l} \{(g_j^i, f_j^i, \alpha_j^i), (g_j^{i+1}, f_j^i, \neg\alpha_j^i)\}$ (on pose $g_j^{k+1} = g_j^1$)

$A_2 = \bigcup_{i=1}^{k} \{(g_i, f_i, x^i)$ / où x est un littéral qui apparaît dans $C_i\}$

$A_3 = \bigcup_{i=1}^{k} \bigcup_{j=1}^{l-1} \{(g_j^{'i}, f_j^{'i}, x)$ / $x \in L\}$

(cela peut aider de faire un dessin dans un cas particulier, avec $k = 2$ ou $k = 3$).

Soit M un mariage à 3 de (G, F, K, A).

a. Montrer que pour chaque variable α_j, apparaissent dans $M \cap A_1$, soit tous les littéraux $\alpha_j^i, i = 1 \cdots n$ et aucun des littéraux $\neg\alpha_j^i, i = 1 \cdots k$, soit tous les littéraux $\neg\alpha_j^i, i = 1 \cdots k$ et aucun des littéraux $\alpha_j^i, i = 1 \cdots k$.

b. Montrer que M induit une valuation (utiliser la question précédente) qui satisfait $\mathcal{C}$ (utiliser A_2).

c. Montrer que réciproquement, chaque valuation qui satisfait $\mathcal{C}$ induit un mariage à 3 dans (G, F, K, A).

d. En déduire que le problème des mariages à 3 est NP-complet.

Exercice 16 (couplages). On montre dans cet exercice que le problème des couplages ou mariages à 2, se résout en temps polynomial (on verra que ce problème généralisé à plus de 2 est NP-complet).

Soit $P = (G, F, A)$ un graphe bipartite, avec G et F de même cardinal n, (A est un sous-ensemble de $G \times F$). On appelle couplage un sous-ensemble de A tel que

$$\forall (u, v) \in M \,\forall (u', v') \in M \;\; [(u, v) \neq (u', v') \Rightarrow (u \neq u' \text{ et } v \neq v')]$$

On dit que le couplage M est *maximal* si tout couplage a un cardinal inférieur ou égal au cardinal de M. On dit que le couplage M est *parfait* si $\operatorname{card} M = \operatorname{card} G = \operatorname{card} F$.

Le problème des couplages est celui de l'existence d'un couplage parfait. Ce problème se ramène évidemment à la recherche d'un couplage maximal.

Un sous-ensemble C de A est un *chemin* de P signifie qu'il existe $\{c_1, \cdots, c_p\}$ un sous-ensemble à p éléments ($i \neq j \Rightarrow c_i \neq c_j$) de $G \cup F$ tel que

$$(u, v) \in C \quad \text{si et seulement si} \quad \exists i \in \{1, \cdots, p-1\} \; \{u, v\} = \{c_i, c_{i+1}\}\,.$$

On appellera *suite des nœuds* de C l'une des deux suites $(c_1, \cdots, c_p)$ ou $(c_p, \ldots, c_1)$; c_1 et c_p sont les *extrémités* de C.

Soit $M \subset A$ est un couplage. On dit qu'un chemin C est *alternant* pour M s'il a un nombre de nœuds pair et si une arête du chemin sur deux est dans M. C'est à dire que, la suite de ses nœuds étant $(c_1, \cdots, c_{2p})$ ordonnée de façon que $c_1 \in G$,

- soit pour tout i impair tel que $0 < i < 2p$, $(c_i, c_{i+1}) \in M$ (donc par définition d'un couplage, si i est pair $(c_{i+1}, c_i) \notin M$) ;
- soit pour tout i pair tel que $0 < i < 2p$, $(c_{i+1}, c_i) \in M$ (donc par définition d'un couplage, si i impair $(c_i, c_{i+1}) \notin M$).

Un nœud est *célibataire* pour un couplage M s'il n'apparaît dans aucune arête du couplage.

Un chemin C alternant pour M d'extrémités c_1 et c_{2p} *étend* M si ses deux extremités sont des nœuds célibataires.

1. Montrer que si C est un chemin alternant pour M qui étend M, alors $M \triangle C$ est un couplage de cardinalité $\operatorname{card} M + 1$.
2. Montrer que si M et M' sont deux couplages, alors toute partie connexe de cardinal impair de $M \triangle M'$ est un chemin alternant pour M.
3. Montrer qu'un couplage M est maximal si et seulement s'il n'existe pas de chemin alternant qui étend M.

On va maintenant montrer que l'algorithme de recherche d'un couplage maximal par itération de la recherche d'un chemin alternant qui étend le couplage déja construit est polynomial. On suppose que l'ensemble A et les couplages sont représentés par des matrices n, n.

4. Quel est le temps de recherche d'un couplage M tel que, pour tout $(u, v) \in A - M$, $M \cup \{(u, v)\}$ n'est pas un couplage ?
5. Montrer que, si C est un chemin alternant qui étend M dont la suite des nœuds est $c_1, \ldots, c_{2p}$, et que si C' est un chemin alternant dont la suite des nœuds s'écrit $c_1, \ldots c_{2i}, c'_{2i+1}, \ldots, c'_{2(i+j)}, \ldots, c_{2q}$ avec $i < q < p$, alors il existe un chemin alternant qui étend M et dont la suite des nœuds commence par la suite des nœuds de C'.
6. Montrer que, pour un couplage donné, on peut trouver l'ensemble des nœuds célibataires en temps $O(n^2)$.
7. Montrer que, étant donné un nœud célibataire $c \in F$ et un couplage M tel qu'à la question **4**, la recherche d'un chemin alternant qui étend M dont l'une des extrémités est c peut se faire en temps $O(n^2 . \operatorname{card} A)$ [1] ("marquer" au fur et à mesure les arêtes visitées).
8. Montrer que la construction du couplage $M \triangle C$ pour un couplage M et un chemin alternant C peut se faire en temps $O(n^2)$.
9. En déduire que la recherche d'un couplage maximum peut se faire en temps $O(n^3 . \operatorname{card} A)$ [2], et que le problème des couplages et polynomial.

1. n pour le temps d'accès aux entrées de la matrice. On peut considérer que ce temps est constant dans un langage de programmation où les matrices sont codées comme des tableaux à double entrée.
2. en fait $O(n . \operatorname{card} A)$ en faisant un peu attention et voir note précédente