Correction de l'examen – 18 mai 2022

1 Calculabilité

Exercice 1

Soit f une fonction calculable. On appelle $d(f)$ le domaine de définition de f, c'est à dire

$$d(f) = \{x \mid f(x) \text{ est défini}\}$$

Pour chacune des affirmations suivantes, dites si elle est vraie ou fausse et à chaque fois justifier vos réponses.

1. $\exists f$ tel que $d(f)$ est décidable.
2. $\forall f, d(f)$ est décidable.
3. $\exists f$ tel que $d(f)$ est récursivement énumérable et non décidable.
4. $\exists f$ tel que $d(f)$ n'est pas récursivement énumérable.
5. $\exists f$ tel que le complémentaire de $d(f)$ n'est pas récursivement énumérable.
6. $\exists f$ tel que le complémentaire de $d(f)$ est récursivement énumérable et non décidable.
7. $\exists f$ tel que le complémentaire de $d(f)$ est décidable.

Pour n'importe quelle fonction f calculée par la procédure `ProcF`, posons la procédure automatique `sfc` qui calcule la fonction semi-caractéristique de $d(f)$:

```
int sfc(int x) {
    ProcF(x);
    return 1;
}
```

Cette procédure est bel et bien la fonction semi-caractéristique de $d(f)$: si x est dans l'ensemble de définition de f, alors `ProcF`(x) va finir, et ainsi la fonction retourne 1. Sinon, `ProcF`(x) ne fini pas et `sfc`(x) ne fini pas non plus.

1. **Vrai**. Prenons f totale. Son ensemble de définition est $\mathbb{N}$, et comme la procédure automatique `ProcF` termine pour n'importe quelle entrée, la procédure automatique `sfc` termine aussi pour n'importe quelle entrée. Comme `sfc` termine pour toute entrée, cette procédure automatique calcule la fonction caractéristique de f et donc $d(f)$ est décidable.
2. **Faux**. Prenons f une fonction partielle. C'est à dire qu'il existe x tel que `ProcF`(x) ne termine pas. Comme `ProcF`(x) ne termine pas, `sfc`(x) ne termine pas non plus, et `sfc` n'est pas la fonction caractéristique mais semi-caractéristique de $d(f)$. Cet ensemble n'est pas décidable, donc on n'a pas $\forall f.\ d(f)$ décidable.
3. **Vrai**. En prenant la même fonction que pour la question précédente, comme `sfc` calcule la fonction semi-caractéristique de $d(f)$, cet ensemble est récursivement énumérable. Cependant, il n'est pas décidable (on ne peut pas exhiber la fonction caractéristique de $d(f)$).
4. **Faux**. Toute fonction est soit partielle, soit totale. On a vu en (1) qu'une fonction totale donne $d(f)$ décidable et donc récursivement énumérable, et en (2) et (3) qu'une fonction partielle donne $d(f)$ récursivement énumérable. Il n'existe donc pas de fonction telle que $d(f)$ non récursivement énumérable.

5. **Vrai**. On peut prendre n'importe quelle fonction partielle, et on ne peut pas énumérer les éléments dans le complémentaire de $d(f)$. En effet, supposons que le complémentaire est récursivement énumérable. Alors on sait quels éléments sont dans $d(f)$ et quels éléments ne sont pas dedans. Donc $d(f)$ est décidable. Or, $d(f)$ est récursivement énumérable et non décidable d'après la question 3. Donc le complémentaire de $d(f)$ n'est pas récursivement énumérable.
6. **Faux**. On a $\forall f.\ d(f)$ récursivement énumérable au minimum. Donc si le complémentaire $d(f)$ est aussi récursivement énumérable, alors $d(f)$ et son complémentaire seront décidables. Or, on veut que le complémentaire de $d(f)$ ne soit pas décidable, donc on ne peut pas trouver f tel que son complémentaire est récursivement énumérable et non décidable.
7. **Vrai**. On prend f une fonction totale. Comme $d(f)$ est décidable, son complémentaire est aussi décidable (il suffit de renvoyer l'inverse de $d(f)$).

Exercice 2

> Soit f une bijection des suites finies d'entiers dans $\mathbb{N}$ définie $f(x_1, x_2, \ldots, x_k)$ où la liste est $(x_1, x_2, \ldots, x_k)$ et $x_i \in \mathbb{N}$. En déduire une fonction g bijective des suites croissantes (au sens large i.e. $x_i \leq x_{i+1},\ \forall i$) finies d'entiers dans $\mathbb{N}$.

Définissons une fonction bijective h entre les suites finies d'entiers et les suites finies d'entiers croissantes. On pose :

$$h(x_1, \ldots, x_k) = \forall i \leq k.\ \sum_{j=1}^{i} x_j$$

C'est à dire que pour tout i, x_i est la somme de x_1 à x_i. Par exemple, pour trois entiers x_1, x_2, x_3, $h(x_1, x_2, x_3) = (x_1, x_1 + x_2, x_1 + x_2 + x_3)$. Cette suite d'entiers est bien croissante ($\forall i.\ x_i < x_{i+1}$) et h est bijective. Prouvons la bijection :

Injection On démontre l'injectivité par induction sur le nombre d'entiers dans la suite.

- **Base** S'il y a un seul entier x_0, alors $h(x_0) = (x_0)$. Prenons $x, y \in \mathbb{N}$ et supposons $h(x) = h(y) \Rightarrow (x) = (y) \Rightarrow x = y$ donc la propriété est bien prouvée pour une suite de taille 1.
- **Induction** On suppose que la propriété est vraie pour tout $i \leq n$. Montrons qu'elle est vraie au rang n. Soit la suite $(x_1, x_2, \ldots, x_i)$ telle que $h(x_1, x_2, \ldots, x_i) = h(y_1, y_2, \ldots, y_i) \Rightarrow (x_1, x_2, \ldots, x_i) = (y_1, y_2, \ldots, y_i)$ et montrons que l'injectivité est encore vraie si on y ajoute un entier. Prenons deux entiers $x, y \in \mathbb{N}$. Supposons $h(x_1, x_2, \ldots, x_i, x) = h(x_1, x_2, \ldots, x_i, y)$, c'est à dire que $(x_1, x_1 + x_2, \ldots, \sum_{j=0}^{i} x_j, \sum_{j=0}^{i} x_j + x) = (x_1, x_1 + x_2, \ldots, \sum_{j=0}^{i} x_j, \sum_{j=0}^{i} x_j + y)$, on a $\sum_{j=0}^{i} x_j + x = \sum_{j=0}^{i} x_j + y \Rightarrow x = y$ donc h est bien injective si on y ajoute un entier.
- **Conclusion** h est injective pour les suites avec un seul entier, et elle est injective pour les suites avec n entiers, donc h est injective pour tout entier.

Surjection On prend $(y_1, y_2, \ldots, y_k)$ une suite d'entiers et on veut démontrer qu'il existe $(x_1, x_2, \ldots, x_k)$ une autre suite d'entiers tel que $(y_1, y_2, \ldots, y_k) = h(x_1, x_2, \ldots, x_k)$. On a $(x_1, x_2, \ldots, x_k) = (y_1, y_2 - y_1, \ldots, y_k - y_{k-1})$. $x_1, x_2, \ldots, x_k \in \mathbb{N}$ car la suite y_i est croissante et donc comme $y_{i+1} > y_i$, on a $y_{i+1} - y_i \geq 0$. h est bien surjective.

Comme h est bijective et que f est bijective, on peut exhiber $g = f \circ h$ qui est également bijective (la composition de deux fonctions bijectives est également bijective).

2 Complexité

Exercice 3

Considérons le problème Clique, ce problème est connu $\mathcal{NP}$-complet.
Clique (Clique)

Données $G = (V, E)$ un graphe non-orienté et $k \in \mathbb{N}$

Question Existe-t-il une clique de taille k ?

1. Montrer que le problème Deux Cliques Disjointes est $\mathcal{NP}$-complet.
   Deux Cliques Disjointes

   Données $G = (V, E)$ un graphe non-orienté et $k \in \mathbb{N}$.

   Question Existe-t-il deux cliques disjointes de taille k ?
2. Montrer que le problème Clique reste $\mathcal{NP}$-complet même quand tous les sommets admettent un même degré.

   Aide : Soit Δ le degré maximum du graphe. Construire Δ copies du graphe G et ajouter des sommets pour obtenir un graphe Δ-régulier. Vous montrerez l'équivalence suivante : il existe une clique de taille k dans G si et seulement s'il existe une clique de taille k dans le graphe construit.

1. Soit $\langle G = (V, E), k \rangle$ une instance de Clique. On construit l'instance $\langle G' = (V', E'), k' \rangle$ de Deux Cliques Disjointes de la façon suivante :

$$\begin{aligned} V' &= V \cup \{u \mid u \in V\} \\ E' &= E \cup \{uv \mid uv \in E \wedge u, v \in V' - V\} \\ k' &= k \end{aligned}$$

   En clair, on prend le graphe G, et on le duplique. Montrons qu'il existe une clique de taille k dans G si et seulement s'il existe deux cliques disjointes de taille k' dans G'.

   $\Rightarrow$ Supposons qu'il existe une clique de taille k dans G. Cette clique est retrouvée dans G' avec deux ensembles de sommets disjoints : $V' \cap V$ et $V' - V$. On a $(V' \cap V) \cap (V' - V) = \emptyset$ et on a bien deux cliques de taille k' (car il existe une clique de taille k dans V, que $V' \cap V = V$ et $V' - V = V$ et que $k' = k$).

   $\Leftarrow$ Supposons qu'il existe deux cliques disjointes de taille k' dans G'. Il y a donc une clique dans $V' \cap V$ et une autre dans $V' - V$ car ces deux ensembles de sommets sont disjoints. Or, $V' \cap V = V$ et $V' - V = V$, donc il existe une clique de taille k' dans V et par extension dans G. Comme $k = k'$, il y a bien une clique de taille k dans G s'il y a deux cliques disjointes de taille k' dans G'.

   Clique se réduit à Deux Cliques Disjointes, donc ce problème est $\mathcal{NP}$-dur. Montrons qu'il est aussi dans $\mathcal{NP}$ pour montrer qu'il est $\mathcal{NP}$-complet. Le certificat de Deux Cliques Disjointes est polynomial. En connaissant E', il suffit de donner en certificat deux ensembles de sommets X, X' qui sont les deux cliques. Deux Cliques Disjointes est bien dans $\mathcal{NP}$, donc ce problème est $\mathcal{NP}$-complet.
2. Montrons qu'il existe une clique de taille k dans G si et seulement s'il existe une clique de taille k dans le graphe construit comme dit dans l'énoncé.

   $\Rightarrow$ On suppose qu'il existe une clique de taille k dans G. Dupliquer Δ fois G pour que tous les sommets soient de degré Δ n'influence pas la présence de la clique de taille k dans G, donc il y a bel et bien une clique de taille k dans G'.

   $\Leftarrow$ Par contraposée, on suppose qu'il n'existe pas de clique de taille k dans G. Ainsi, en faisant des copies du graphe, aucune clique de taille k n'apparait en chemin. En effet, G ne contient pas de clique, et une clique de taille k ne peut pas être créée non plus en ajoutant des arêtes. Donc G' ne contiendra pas de clique de taille k.

Clique reste donc bien $\mathcal{NP}$-complet même quand tous les sommets admettent un même degré.

Exercice 4

Soit $\phi = (a \lor b) \land (b \lor \neg c) \land (\neg b \lor \neg d) \land (b \lor d) \land (d \lor a)$.

1. Est-ce que ϕ est satisfiable ?
2. Proposer un algorithme glouton qui permet de satisfaire au moins la moitié des clauses pour une formule 2-SAT.

1. ϕ est satisfiable. Prenons l'affectation α suivante :

$$\alpha(a) = \top,\ \alpha(b) = \top,\ \alpha(c) = \bot,\ \alpha(d) = \bot$$

Cette affectation rend bien ϕ vraie :

$$\begin{aligned}\alpha(\phi) &= (\top \lor \top) \land (\top \lor \top) \land (\neg\top \lor \neg\bot) \land (\top \lor \bot) \land (\bot \lor \top) \\ &= \top \land \top \land \top \land \top \land \top \\ &= \top\end{aligned}$$

2. L'algorithme est le suivant :
   - $\alpha \leftarrow$ affectation de taille du nombre de littéraux de ϕ.
   - Pour chaque clause littéral l de ϕ :
     - $\alpha(l) = \top$
   - Si le nombre de clauses satisfaites par α est moins de la moitié : Renvoyer α inversé
   - Sinon : Renvoyer α

Exercice 5

Double-Sat

Données Soit ϕ une formule logique sous forme conjonctive.

Question Existe-t-il deux affectations possibles pour ϕ ?

Montrer que Double-Sat est $\mathcal{NP}$-complet.

Soit $\langle \phi \rangle$ une instance de Sat. On construit l'instance $\langle \phi' \rangle$ de Double-Sat de la façon suivante :

$$\phi' = \phi \land (z \lor \bar{z})$$

avec z un littéral qui n'apparait pas dans ϕ. Très clairement, Double-Sat est dans $\mathcal{NP}$, il suffit de donner les deux affectations différentes pour tester. Ensuite, Double-Sat est $\mathcal{NP}$-dur, car Sat se réduit à ce problème. La correction de la réduction est évidente : s'il existe une affectation pour ϕ, alors il y aura deux affectations pour ϕ' : une où $\alpha(z) = \top$ et l'autre où $\alpha(z) = \bot$. D'un autre côté, si ϕ' est satisfiable, alors ϕ sera aussi satisfiable car $(z \lor \bar{z})$ est une tautologie et n'influence pas sur la satisfiabilité de la formule. Double-Sat est donc bien $\mathcal{NP}$-complet.