Correction du partiel.

Pour l' **exercice 1** (3pts) et l'**exercice 2** (3pts) : voir le cours.

Exercice 3.
(3 pts)
Pour montrer que $\leq_{\mathsf{NP}}$ est transitive, supposons que $L_1 \leq_{\mathsf{NP}} L_2$ et $L_2 \leq_{\mathsf{NP}} L_3$, et que $\mathcal{M}_1$ et $\mathcal{M}_2$ sont deux machines de Turing non-déterministes de temps polynomial attestant de ces deux propriétés.
Soit alors $\mathcal{M}$ la machine définie de la manière suivante : sur l'entrée x,
– $\mathcal{M}$ se comporte d'abord comme $\mathcal{M}_1$, et écrit son résultat y sur un ruban de travail R,
– puis $\mathcal{M}$ se comporte comme $\mathcal{M}_2$ en lisant son entrée y sur R et en écrivant son résultat z sur un ruban de sortie R'.
Alors on a :
$(x \in L_1)$ ssi il existe une exécution de $\mathcal{M}_1$ sur x dont le résultat y appartient à L_2.
$(y \in L_2)$ ssi il existe une exécution de $\mathcal{M}_2$ sur y dont le résultat z appartient à L_3.
Donc, comme une exécution de $\mathcal{M}$ est obtenue par une exécution de $\mathcal{M}_1$ suivie d'une exécution de $\mathcal{M}_2$, on a :
$(x \in L_1)$ ssi il existe une exécution de $\mathcal{M}$ sur x dont le résultat z appartient à L_3.
Il nous reste à montrer que $\mathcal{M}$ est de temps polynomial. Soit p_1 (resp. p_2) un polynôme bornant le temps d'exécution de $\mathcal{M}_1$ (resp. $\mathcal{M}_2$).
Considérons une exécution de $\mathcal{M}$ sur une entrée x de longueur n :
– l'exécution de $\mathcal{M}_1$ sur x produit un mot y sur R en un nombre de pas inférieur ou égal à $p_1(n)$; de plus comme chaque écriture de symbole demande au moins une unité de temps on a $|y| \leq p_1(n)$;
– l'exécution de $\mathcal{M}_2$ sur y produit un mot z sur R' en un nombre de pas inférieur ou égal à $p_2(|y|) \leq p_2(p_1(n))$.
Une exécution de $\mathcal{M}$ sur une entrée x est donc faite en un nombre de pas borné par $p_1(n) + p_2(p_1(n))$, qui est un polynôme. Donc $\mathcal{M}$ est de temps polynomial. Donc $L_1 \leq_{\mathsf{NP}} L_3$. La relation $\leq_{\mathsf{NP}}$ est transitive.

Supposons maintenant $L_1 \leq_{\mathsf{NP}} L_2$ et $L_2 \in NP$. Soit $\mathcal{M}_1$ une machine non-déterministe attestant de la première propriété et $\mathcal{M}_2$ une machine non-déterministe de temps polynomial décidant L_2. Soit $\mathcal{M}$ définie comme précédemment par :
– $\mathcal{M}$ se comporte d'abord comme $\mathcal{M}_1$, et écrit son résultat y sur un ruban de travail R,
– puis $\mathcal{M}$ se comporte comme $\mathcal{M}_2$ en lisant son entrée y sur R et accepte (resp. rejette) comme $\mathcal{M}_2$.
Alors si p_1 (resp. p_2) est un polynôme bornant le temps de $\mathcal{M}_1$ (resp. $\mathcal{M}_2$), le temps de $\mathcal{M}$ est borné par $p_1(n) + p_2(p_1(n))$, donc $\mathcal{M}$ est de temps polynomial.
Par ailleurs on a :
$(x \in L_1)$ ssi il existe une exécution de $\mathcal{M}_1$ sur x dont le résultat y appartient à L_2.
$(y \in L_2)$ ssi il existe une exécution de $\mathcal{M}_2$ sur y qui accepte.
Donc :
$(x \in L_1)$ ssi il existe une exécution de $\mathcal{M}$ sur x qui accepte.
Donc $L_1 \in NP$.

Exercice 4.
(5 pts)

1. Soit $\mathcal{L} \in \mathsf{P}$ et $\chi_{\mathcal{L}}$ sa fonction caractéristique (calculable en temps polynomial). Alors une fonction de sélection pour $\mathcal{L}$ est définie par $f(x,y) = x$ si $\chi_{\mathcal{L}}(x) = 1$ et $f(x,y) = y$ sinon. On vérifie aisément que cette fonction, calculable en temps polynomial, est effectivement une fonction de sélection pour $\mathcal{L}$.

2. Soit $\mathcal{L} \in \mathsf{P\text{-}Sel}$, et f sa fonction de sélection. Alors soit g définie par $g(x,y) = y$ si $f(x,y) = x$ et $g(x,y) = x$ sinon. La fonction g est calculable en temps polynomial, car on peut évaluer $f(x,y)$ en temps polynomial, et on peut décider en temps polynomial si un mot est égal à x. De plus, si $x \in \bar{\mathcal{L}}$ et $y \notin \bar{\mathcal{L}}$ on a $f(x,y) = y$, donc $g(x,y) = x \in \bar{\mathcal{L}}$. Symétriquement si $x \notin \bar{\mathcal{L}}$ et $y \in \bar{\mathcal{L}}$, $f(x,y) = x$ et donc $g(x,y) = y \in \bar{\mathcal{L}}$. Enfin, si $x \in \bar{\mathcal{L}}$ et $y \in \bar{\mathcal{L}}$, on a aussi $g(x,y) \in \bar{\mathcal{L}}$.
   Donc g est une fonction de sélection pour $\bar{\mathcal{L}}$.

3. Soit $\mathcal{L} \in \mathsf{P\text{-}Sel}$, et f sa fonction de sélection. Soit $L' = \{\langle x,y\rangle : f(x,y) = x\}$. On a $L' \in \mathsf{P}$, car on peut en temps polynomial (par rapport à $|\langle x,y\rangle|$) évaluer $f(x,y)$ et tester si son résultat est égal à x. Soit $\langle x,y\rangle \in \mathcal{L} \times \bar{\mathcal{L}}$, alors $f(x,y) = x$ et donc $\langle x,y\rangle \in L'$. Inversement, si $\langle x,y\rangle \in \bar{\mathcal{L}} \times \mathcal{L}$, alors $\langle x,y\rangle \in \overline{L'}$.
   Réciproquement, étant donné un tel L', on définit la fonction f par : $f(x,y) = x$ si $\langle x,y\rangle \in L'$ et $f(x,y) = y$ sinon. Comme $L' \in \mathsf{P}$, on peut décider en temps polynomial si $\langle x,y\rangle \in L'$, et donc $f \in \mathsf{FP}$. De plus, si $x \in \mathcal{L}$ et $y \notin \mathcal{L}$ on a $\langle x,y\rangle \in \mathcal{L} \times \bar{\mathcal{L}} \subseteq L'$, donc $f(x,y) = x$. Symétriquement si $x \notin \mathcal{L}$ et $y \in \mathcal{L}$ on a $f(x,y) = y$. Enfin si $x \in \mathcal{L}$ et $y \in \mathcal{L}$, comme $f(x,y) = x$ ou y, $f(x,y) \in \mathcal{L}$. Donc f est une fonction de sélection pour $\mathcal{L}$ et $\mathcal{L} \in \mathsf{P\text{-}Sel}$.

4. Soit $\mathcal{L} \in \mathsf{P\text{-}Sel}$ un langage NP-dur (avec f sa fonction de sélection). En particulier, il existe une réduction polynomiale g telle que $x \in \textsc{Sat} \iff g(x) \in \mathcal{L}$. L'algorithme suivant décide Sat en temps polynomial, et donc $\mathsf{P} = \mathsf{NP}$.
   Entrée : $\phi(x_1, \ldots, x_k)$.
   – Pour i de 1 à k, faire
     – $y := f(g(\phi[x_i \leftarrow 0]), g(\phi[x_i \leftarrow 1]))$;
     – Si $y = g(\phi[x_i \leftarrow 0])$, alors $\phi := \phi[x_i \leftarrow 0]$ et $\phi := \phi[x_i \leftarrow 1]$ sinon.
   – Accepter ssi ϕ est vraie (ϕ n'a plus de variable à cette étape).

Exercice 5.
(4 pts)

1. Supposons $\mathsf{SPACE}(n^c) \subseteq NP$. Essayons alors de montrer que $\mathsf{SPACE}(n^{2c}) \subseteq NP$.
   Soit $A \in \mathsf{SPACE}(n^{2c})$, décidé par une MT $\mathcal{M}$ de temps $d.n^{2c}$. Définissons alors :

$$A_{pad} = \{x01^{|x|^2} / x \in A\}.$$

   Soit $\mathcal{M}'$ la machine suivante : sur l'entrée x, elle écrit $x01^{|x|^2}$. Cette machine est de temps polynomial et calcule une réduction de A à A_{pad}, donc $A \leq_P A_{pad}$.
   Maintenant, soit $\mathcal{M}''$ la machine suivante :
   sur l'entrée y,
   – si y n'est pas de la forme $x01^{|x|^2}$, alors elle rejette ;
   – si $y = x01^{|x|^2}$, la machine écrit x sur un ruban de travail, puis se comporte comme $\mathcal{M}$ sur x.

Alors cette machine décide A_{pad}, et son calcul est fait en espace $O(n^c)$, donc $A_{pad} \in \mathsf{SPACE}(n^c)$. Donc $A_{pad} \in \mathsf{NP}$, et comme $A \leq_P A_{pad}$ on en déduit que $A \in NP$. Donc $SPACE(n^{2c}) \subseteq NP$.

En répétant cet argument, on obtient que pour tout $k \geq 1$, $SPACE(n^{2^k.c}) \subseteq \mathsf{NP}$. Or, si $d > 0$, alors il existe k tel que $d \leq 2^k.c$, donc $SPACE(n^d) \subseteq NP$. Donc $\cup_{d>0} SPACE(n^d) = \mathsf{PSPACE} \subseteq \mathsf{NP}$. Or $\mathsf{NP} \subseteq \mathsf{PSPACE}$, donc $\mathsf{NP} = \mathsf{PSPACE}$.

2. S'il existe $c > 0$ tel que $\mathsf{SPACE}(n^c) = \mathsf{NP}$, alors par la question précédente $\mathsf{SPACE}(n^c) = \mathsf{PSPACE}$. Or par le théorème de hiérarchie en espace déterministe, $\mathsf{SPACE}(n^c) \subsetneq \mathsf{SPACE}(n^{c+1}) \subseteq \mathsf{PSPACE}$. C'est une contradiction.

3. On note $\mathsf{E} = \bigcup_c \mathsf{DTIME}(2^{cn})$ [1]. On veut montrer que $\mathsf{DTIME}(2^{cn}) \subseteq \mathsf{NP}$ implique $\mathsf{NP} = \mathsf{E}$.
Supposons $\mathsf{DTIME}(2^{cn}) \subseteq \mathsf{NP}$. On procède comme dans le 1. pour montrer que $\mathsf{DTIME}(2^{2cn}) \subseteq \mathsf{NP}$. Soit $A \in \mathsf{DTIME}(2^{2cn})$. On définit :

$$A_{pad} = \{x01^{|x|} / x \in A\}.$$

On peut montrer alors que $A \leq_P A_{pad}$ et $A_{pad} \in \mathsf{DTIME}(2^{cn})$. De ces deux propriétés on déduit alors, grâce à l'hypothèse, que $A \in \mathsf{NP}$. Donc $\mathsf{DTIME}(2^{2cn}) \subseteq \mathsf{NP}$, et plus généralement, pour tout $k \geq 1$, $\mathsf{DTIME}(2^{2^k cn}) \subseteq \mathsf{NP}$. Donc $\mathsf{E} \subseteq \mathsf{NP}$.
Supposons alors $\mathsf{DTIME}(2^{cn}) = \mathsf{NP}$. D'après le résultat précédent on a $\mathsf{DTIME}(2^{cn}) = \mathsf{NP} = \mathsf{E}$.
Or on a $2^{cn} \log(2^{cn}) = cn2^{cn} = o(2^{(c+1)n})$, donc d'après le théorème de hiérarchie en temps déterministe on a $\mathsf{DTIME}(2^{cn}) \subsetneq \mathsf{DTIME}(2^{(c+1)n})$, d'où une contradiction.

Exercice 6.
(5 pts)
Notons IPD le langage $IN\_PLACE\_DIVERGE$.

1. Soit $\mathcal{M}$ la machine utilisant l'alphabet $\{0, 1, \triangleright, \#, \square\}$ (à 5 symboles donc) et définie par : sur l'entrée α,
- sur un ruban de travail, elle écrit $|\alpha|$ symboles $\square$, suivis d'un #, puis ramène la tête de lecture au début du ruban ;
- sur un deuxième ruban de travail, elle initialise un compteur en binaire, à la valeur $5^{|\alpha|}$;
- elle simule ensuite l'exécution de la machine $\mathcal{M}_\alpha$ (à un ruban) sur le premier ruban de travail, avec :
  - à chaque pas simulé, elle décrémente le compteur,
  - si sur le premier ruban de travail la tête lit le symbole #, alors $\mathcal{M}$ rejette,
  - si la machine $\mathcal{M}_\alpha$ s'arrête, alors $\mathcal{M}$ rejette,
  - si le compteur arrive à 0, alors $\mathcal{M}$ accepte.

Alors : la repésentation de $5^{|\alpha|}$ en binaire est de longueur polynomiale en $|\alpha|$; la description de la machine $\mathcal{M}_\alpha$ est de taille inférieure à α, et donc la simulation de $\mathcal{M}_\alpha$ effectuée ici est en espace polynomial par rapport à $|\alpha|$. Donc le calcul de $\mathcal{M}$ est en espace polynomial.

Par ailleurs, le nombre de contenus possibles pour le premier ruban de travail est majoré par $5^{|\alpha|}$, donc : si $\mathcal{M}_\alpha$ effectue plus de $5^{|\alpha|}$ pas, alors elle diverge, et si elle diverge alors elle effectue plus de $5^{|\alpha|}$ pas ... Donc $\mathcal{M}$ décide bien le langage IPD, donc $IPD \in \mathsf{PSPACE}$.

1. A ne pas confondre avec $\mathsf{EXP} = \bigcup_c \mathsf{DTIME}(2^{n^c})$.

2. Soit $L \in$ PSPACE, décidé par une machine déterministe $\mathcal{N}$ en espace $d.n^c$. En particulier, $\mathcal{N}$ termine sur toutes les entrées. On peut supposer $c, d \geq 1$. Soit $a = \llcorner \mathcal{N} \lrcorner$.

Soit $x \in \{0,1\}^\star$. Soit alors $\mathcal{M}^x$ la machine suivante :

1. elle écrit x sur son premier ruban de travail,
2. elle simule ensuite l'exécution de $\mathcal{N}$ sur l'entrée x :
   - si $\mathcal{N}$ accepte, alors : $\mathcal{M}^x$ efface le contenu des rubans et revient en (a),
   - si $\mathcal{N}$ s'arrête mais n'accepte pas, alors $\mathcal{M}^x$ rejette.

Alors on a : $\mathcal{M}^x$ diverge sur l'entrée vide ssi $\mathcal{N}$ accepte x.

De plus on a : $\llcorner \mathcal{M}^x \lrcorner \leq k_1|x| + \llcorner \mathcal{N} \lrcorner + k_2$, où k_1, k_2 sont des constantes.

Soit alors α_x un mot obtenu en concaténant $\llcorner \mathcal{M}^x \lrcorner$ avec des symboles inutiles, de façon à avoir $|\alpha_x| \geq d.|x|^c$ et $\mathcal{M}_{\alpha_x} \equiv \mathcal{M}^x$.

Alors on a : l'application $x \rightarrowtail \alpha_x$ est calculable en temps polynomial, et

$x \in L$ ssi $\alpha_x \in IPD$.

Donc $L \leq_P IPD$. Donc IPD est PSPACE-dur, donc d'après la question précédente il est PSPACE-complet.