IFT 2105 H21 **Geňa Hahn**

Introduction à l'informatique théorique

FINAL

le 26 avril 2021

Durée: 180 minutes + 30 minutes pour imprimer et scanner

15h30 – 19h00
ESH: 15h30 – 20h00

Valeur: 50% de la note totale

Directives:

- Toute documentation écrites est permise.
- **Consultation par internet ou mobile est interdite.**
- Répondez sur le questionnaire, dans l'espace libre qui suit chaque question. Utilisez les dos des pages comme brouillon. **L'espace aloué n'est aucune indication de la longueur de la réponse! Il est souvent beaucoup trop grand.**
- Sauf indication contraire, aucun point ne sera accordé pour une réponse, correcte ou pas, si elle n'est pas accompagnée d'une justification.
- Notez la différence entre *justifier* (argument rapide et court) et *prouver* ou *démontrer* (argument détaillé).
- Vous pouvez vous servir de résultats vus en cours, en TP ou dans des livres à condition de les énoncer précisément. Bien sûr, si on vous demande de prouver un résultat vu en cours ou ailleurs, il ne suffit pas de citer, il faut faire la preuve!
- Pour répondre à une question, vous pouvez également vous servir de résultats énoncés dans d'autres questions dans l'examen, même si vous ne les avez pas démontrés.
- Rappel: $\mathbb{N}$ est l'ensemble des entiers non négatifs.

Déposez le questionnaire rempli EN UN SEUL PDF sans oublier les pages A et B comme les deux premières pages, dans cet ordre.

Pour que la copie soit corrigée, le nom du fichier doit comprendre votre nom et la mention du final, par exemple nom-final.pdf

Aucun fichier ne sera accepté par courriel, vous devez déposer la copie sur Studium.

Sans cette page la copie ne sera pas corrigée.

Page A

La somme des points est 195, l'examen est noté sur 125.

Les problèmes en bleu et en gras sont obligatoires, les autres au choix. Les questions aux choix ne comptent pas si vous n'avez pas essayez toutes les questions obligatoires et obtenu au moins 30 points.

1. ________	/20	10. ________	/10
2. ________	/10	11. ________	/10
3. ________	/15	12. ________	/20
4. ________	/10	13. ________	/10
5. ________	/10	14. ________	/10
6. ________	/10	15. ________	/10
7. ________	/10	16. ________	/10
8. ________	/10	17. ________	/10
9. ________	/10		

Total : ________ + ________ + ________ = ________

Nom: Vanessa Thibault-Soucy

Code permanent/matricule: 20126808

Sans cette page la copie ne sera pas corrigée.

Page B

Ma signature ci-dessous fait foi de:

- J'ai fait l'examen moi-même, sans l'aide d'autres personnes.
- Pendant l'examen, je n'ai consulté aucune source d'information autre que le livre du cours, les notes de cours, les devoirs et mes notes (ni Web, ni autres livres, ni forums, ni courriels, ni textos, etc.).
- Pendant l'examen le seule connection internet ou mobile était celle du ZOOM du cours prévu pour l'examen.
- J'accepte d'être convoqué, après la correction de mon examen, par sélection aléatoire ou si le professeur le juge opportun, à une entrevue en ligne de validation que ma compréhension de la matière et mes réponses à l'examen sont compatibles.

Votre signature ______________________

1. **(20 x 1 points) Pour chacun des énoncés suivants indiquez s'il est vrai ou non en encerclant OUI ou NON. N'encerclez rien s'il est impossible de répondre à la question.** *AUCUNE JUSTIFICATION N'EST NECESSAIRE.*

   (a) L'ensemble de mots sur un alphabet Σ est dénombrable. OUI NON

   (b) Pour tout langage hors contexte il y un nombre infini d'automates finis déterministes qui le reconnaissent. OUI NON

   (c) Il existe une machine de Turing pour décider si un mot donné est généré par une grammaire hors-contexte donnée. OUI NON

   (d) Il existe une machine de Turing pour décider si un mot donné est reconnu par une machine de Turing déterministe donnée. OUI NON

   (e) Un langage NP-complet est polynomial. OUI NON

   (f) Si L_1 est un langage indécidable et si L_2 se réduit polynomialement à L_1 alors L_2 est NP-complet. OUI NON

   (g) Une machine de Turing M acceptant le langage $L = \{\varepsilon\}$ n'accepte aucune entrée. OUI NON

   (h) Pour toute expression régulière il existe un automate fini déterministe qui la reconnaît. OUI NON

   (i) Le langage des palindromes n'est jamais régulier. OUI NON

   (j) Le problème d'arrêt peut être formalisé comme le langage $\{\langle M, w\rangle : \langle M\rangle$ *est le code d'une machine de Turing* M *et* $w \in \Sigma^*$ *tels que* M *décide* $w\}$ et la question *Est-ce que ce langage est reconnu par une machine de Turing?* OUI NON

   (k) Si L est un langage régulier sur l'alphabet Σ, alors pour tout mot $w \in L$ il existe une constante p telle que $|w| \geq p$ et il existe $x, y, z \in \Sigma^*$ vérifiant (1) $w = xyz$, (2) $|xy| \leq p$, (3) $|y| \geq 1$, et (4) pour tout $i \in \mathbb{N}$, $xy^iz \in L$. OUI NON

   (l) Un langage L sur un alphabet Σ et décidable si est seulement si au moins un de L et $\Sigma^* \setminus L$ est reconnaissable. OUI NON

   (m) L'ensemble d'expressions régulières sur un alphabet Σ est un langage hors-contexte. OUI NON

   (n) Une machine de Turing dont le langage est décidable s'arrête sur toute entrée. OUI NON

   (o) Soit L_1, L_2 deux langages sur le même alphabet. Si $L_1 \subseteq L_2$ et si L_1 est régulier, alors L_2 est régulier. OUI NON

   (p) Si L est un langage hors-contexte alors $\overline{L}$ est hors-contexte. OUI NON

   (q) Toute propriété de langages reconnaissables est indécidable. OUI NON

   (r) Le langage Σ^* accepte tout mot sur Σ. OUI NON

   (s) Le complément d'une expression régulière est une expression régulière. OUI NON

   (t) L'ensemble des automates finis est dénombrable. OUI NON

2. **(5 + 5 points) Justifiez DEUX de vos réponses à la question 1. Votre réponse ne vaut rien si elle ne comprend pas l'énoncé de ce que vous prouvez!**

- Question ....O)....

Prenons 2 langages :

- $L_1 = \emptyset \to L_1 \in \mathcal{L}_R$
- $L_2 = \{a^n b^n \mid n \in \mathbb{N}\} \to L_2 \notin \mathcal{L}_R$

Pourtant :

$L_1 \subseteq L_2$ car $\emptyset \in \forall L$

* Copier de la correction du devoir 5 de la démo du 22 Fevrier

- Question ....K)....

Ceci étant le Lemme du Pompisle ($\mathcal{L}_R$), il est insufisant pour prouver qu'un langage est régulier. Cependant, sa contreposé prouver si un langage $\notin \mathcal{L}_R$.

Car $\forall p \in \mathbb{N}$, si $w \in L$, avec $|w| \geq p$, alors $\forall x, y, z \in \Sigma^*$ verifiant :
- $w = xyz$
- $|xy| \leq p$
- $|y| > 0$
- $\exists i \in \mathbb{N}$ tq $xy^iz \notin L$

Donc le théorem est vrai, mais il ne peut seulement être utiliser pour prouver qu'un langage n'est pas régulier.

3. **(5 + 5 + 5 points) Pour chacun des langages suivants, dites s'il est régulier (R), hors-contexte mais pas régulier (HC), ou pas hors-contexte (A). Justifiez brièvement.**

(a) $\{0^n1^m0^n1^m : 28736 < m < 183738627, 28736 < n < 183738627\}$

$\boxed{\text{HC}}$ Il existe une grammaire Hors-Context qui peut généré Le langage.

(b) $\{0^n1^m0^n1^m : m, n \in \mathbb{N}\}$

$\boxed{\text{A}}$ Avec le lemme du pompiste pour les langages Hors contexte, nous pouvons trouver une constante P, un $w \in L$ tq $|w| \geqslant P$, $\forall u, v, x, y, z$ verifiant :

- $w = uvxyz$
- $|vxy| \leq p$
- $|vy| > 0$
- $\forall i \in \mathbb{N}$, $uv^ixy^iz \notin L$

Le mot que nous pomperions n'aurais pas la forme $0^n \underbrace{1^m 0^n}_{\text{serait}} 1^m$, donc ne serait pas dans le langage

(c) $\{0^n1^m0^m1^n : m, n \in \mathbb{N}\}$

$\boxed{\text{A}}$ Comme nous n'avons pas de detail sur la difference ente m et n, nous ne pouvons pas crée de grammaire qui génère ce mot.

Par le lemme du pompiste nous pouvons montré que le langage n'est pas Hors-contexte car pour toute des decomposition $uvxyz$ de $w \in L$, pour $\forall i \in \mathbb{N}$, uv^ixy^iz, $|vxy| \leq p$, $|w| \geqslant p$, $|uy| > 0$, $uv^ixy^iz \notin L$

Donc $L \notin \mathcal{L}_{HC}$

4. **(10 points) Prouvez que si $L \in NP - complet$ et $L <_p L'$ alors $L' \in NP - complet$, à condition**

que $L' \in NP$ **(ajoutez ce qui manque avant de prouver l'énoncé complété).**

On sait que $\forall L'' \in NP$, $L'' \leq_p L$. On sait également que $L \leq_p L'$.
Donc $L'' \leq_p L'$ et $L' \in NP\text{-}C$. On peut montrer la transitivité :

- $L'' \leq_p L' \iff \exists f''$ calculable en temps polynomial tq $w \in L''$ ssi $f''(w) \in L$.
- $L \leq_p L' \iff \exists f'$ calculable en temps polynomial tq $f'(w) \in L'$ ssi $w \in L$.

Donc $f = f'f''(w)$ et $w \in L''$ ssi $f'f''(w) \in L$.

Preuve venant des notes de cours

5. (**10 points**) Est-ce que $\cup_{i \in \mathbb{N}} L_i$ est un langage régulier si L_i est régulier pour tout $i \in \mathbb{N}$? Prouvez votre réponse.

Oui — Nous utiliserons k comme constant, avec $0 < k \leq i$.

Preuve par récurence:
- Cas de bas : $k = 1 \rightarrow \bigcup_{i=0}^{k} L_i = L_0$: par hypothèse $L_0 \in \mathcal{L}_R$

Hypothèse de récurence : Pour $k \geq 2$, supposons que $\bigcup_{i \in \mathbb{N}} L_i \in \mathcal{L}$
- $k-1 \Rightarrow \bigcup_{i=0}^{k-2} L_i$ est régulier.

- Pas de récurence:
  - $k > 2 \Rightarrow \bigcup_{i=0}^{k-1} L_i = \left(\bigcup_{i=0}^{k-2} L_i\right) \cup L_k$ est également régulier car

    ① $\bigcup_{i=0}^{k-2} L_i$ est régulier par preuve de récurence

    ② L_k est régulier selon l'énoncé.

    ③ L'union de 2 langage régulier est régulier.

Nous avons donc prouvé que $\cup_{i \in \mathbb{N}} L_i \in \mathcal{L}_R$.

- Preuve venant du devoir 2 Q1 (1er Fevrier)

6. (**10 points**) Soit M une machine de Turing. Supposons que sur l'entrée w, $|w| = n$, M fait au plus n^3 transitions. Prouvez que sur l'entrée w, M utilise au plus un nombre polynomial (en $|w| = n$) de cases de son ruban.

7. **(10 points) Prouvez que tout langage fini est hors-contexte.**

Sachant qu'un langage hors-context est un langage qui est définit par une grammaire hors context.
Une grammaire Hors-context contient un nombre fini de règle qui génère, après un nombre fini d'application, des mots fini (concatenation de variables terminales.)

Donc $\forall$ langage fini pourrait être généré par une G_{HC}. Pour $w \in L$, L un langage fini et $G = (V, \Sigma, S, R)$ une grammaire HC.

- Si $w = \varepsilon$, alors $S \to \varepsilon$, génère w
- Si $w = a$, avec $a \in \Sigma^*$, $|w| \geq 1$, alors $S \overset{k}{\Rightarrow} a$, $k \in \mathbb{N}$

  Si on regarde avec une étape de moins $(k-1)$, $\exists u \in (V \cap \Sigma)^*$

  $$u \overset{1}{\Rightarrow} a$$
  $$S \overset{k-1}{\Rightarrow} u$$

  Notre mot w sera alors généré par G_{HC}

Ainsi, Nous avons que tout langage fini peut-être générer par une Grammaire Hors context en un nombre fini d'étapes.

8. (**10 points**) Prouvez que $STABLE$ (voir l'annexe) est dans NP.

9. (**5 + 5 points**) Trouvez le(s) erreur(s) dans la preuve suivante qui prétend démontrer que $L = \{0^s 1^s : s \geq 1\}$ sur $\Sigma = \{0, 1\}$ est hors contexte.

> **Preuve.** Soit n la constante du lemme du pompiste (gonflement) et soit $0^n 1^n$ un mot dans L. Soit $0^n 1^n = uvxyz$ une décomposition du mot garantie par le lemme. Si $v = 0^k,\ x = \varepsilon,\ y = 1^k$ avec $1 \leq k \leq \frac{n}{2}$, on aura que $|vxy| \leq n$ et $|vy| \geq 1$. En plus $uv^i xy^i z \in L$ pour tout $i \geq 0$. Donc L est hors-contexte.

Prouvez correctement que L est hors-contexte.

Erreur(s):

Preuve:

10. (**10 points**) Utilisez le théorème de Myhill-Nerode pour prouver que le langage $\{(aab)^n bba(bcb)^n : n \in \mathbb{N}\}$ sur $\{a, b, c\}$ n'est pas régulier.

11. (**10 points**) VRAI OU FAUX ? Soit $L, L' \subseteq \Sigma^*$ deux langages sur un alphabet Σ tels que $L \subseteq L'$. Si L' est régulier alors L est régulier. Prouvez votre réponse.

Faux

Si $L' = \Sigma^*$, $\Sigma^* \in \mathcal{L}_R$

et $L = \{a^n b^n \mid n \in \mathbb{N}\} \notin \mathcal{L}_R$

$$\{a^n b^n \mid n \in \mathbb{N}\} \subseteq \Sigma^* \subseteq \Sigma^*$$

* Σ^* englobe tout les langage définit sur Σ

12. (**10 + 10 point**) Un TP de IFT2105 demande d'écrire une machine de Turing décidant un certain langage. Pour la correction, Rémi, votre charmant démonstrateur, choisit un ensemble X de chaînes sur lesquelles vos machines seront testées. Ce problème de correction se formalise comme suit :

soit $X = \{w_1, \ldots, w_k\}$ et soit $L_X = \{\langle M \rangle \mid \forall w \in X, w \in L(M)\}$.

(a) Prouvez que L_X est indécidable.

Supposons que Lx est décidable, alors il existe une MT M qui décide Lx.
Contruisons une MT Mx qui s'arrète sur l'entrée <M,w>, avec w ∈ Lx. Mx accepte <M,w> si M accepte w, sinon refuse.

On construit maintenant une autre MT M_{Mx} qui utilise Mx comme sous-machine. Cette machine M_{Mx} appellera Mx pour determiner le comportement de M quand l'entré sur M est son propre descriptif <M>. M_{Mx} refuse si M accepte et accepte si M refuse. M_{Mx} se décrit comme suit:

M_{Mx} = a l'entrée <M>, avec M une MT:
1. Envoit l'entrée à <M,<M>>
2. retourne l'inverse de lareponce de Mx

$$\left(M_{Mx}(\langle M \rangle) = \begin{cases} \text{accepte si } M \text{ refuse} \langle M \rangle \\ \text{refuse si } M \text{ accepte } \langle M \rangle \end{cases} \right)$$

Cependans si on envoit à M_{Mx} l'entré $\langle M_{Mx} \rangle$

$$M_{Mx}(\langle M_{Mx} \rangle) = \begin{cases} \text{accepte si } M_{Mx} \text{ refuse} \\ \text{refuse si } M_{Mx} \text{ accepte} \end{cases}$$

Cela implique donc que M_{Mx} et M_x n'existe pas.

On a donc une contradiction, alors Lx est indécidable.

Preuve inspirer de A_{TM} p.207 du manuel.

(b) Prouvez que L_X est reconnaissable.

Selon la définition d'un langage reconnaissable, nous savons que :

Un langage L est reconnaissable s'il existe une MT qui, sur un entrée w :

- accepte si $w \in L$
- Refuse ou boucle si $w \notin L$.

- Si on construit une MT M qui reconnait L_x, et prend en entré un mot $w \in L_x$, si $w \in L_x$ elle accepte et refuse ou boucle sinon.

- On construit une MT M_M qui prend en entré $<M,w>$ et qui appelle M sur w. Donc

M_M = 1. prend en entré $<M,w>$
2. appelle M sur w et attend une reponce
Si M accepte, M_M accepte
Si M refuse, M_M refuse
Si M boucle, M_M attend à l'infini.

On voit donc qu'une telle machine (M_M) peut exister
Donc M reconnait L_x, alors L_x est reconnaissable.

13. (**10 points**) Prouvez que pour tout Σ et tout $L \subseteq \Sigma^*$, $(L^*)^* = L^*$.

Un langage L^* est le resultat de la concatenation de L avec lui même.

- si $L = \{\varepsilon\} \to L^* = \{\varepsilon\} = (L^*)^*$
- si $L = \{a, b\} \to L^* = \{\varepsilon, a, b, aa, ab, bb, \ldots\} = (L^*)^*$

$L^* = (L^*)^*$ car quans on concatene L^* avec lui même, nous avons toujours Le mot vide (ε) qui fait partie du langage.
Comme un langage est fermer sur l'étoile de Kleene, qu'on concatene L avec lui-même une infinité de fois, ou L^* avec lui-même, le résulat final sera le même.

Donc $L^* = (L^*)^*$.

14. (**10 points**) Soit Σ un alphabet et soit E_Σ l'ensemble d'expressions régulières sur Σ. Soit $\Gamma_\Sigma = \{(,), \emptyset, \varepsilon, +, \cdot, ^*\} \cup \Sigma$ (notons que $\emptyset$ et ε sont ici des *lettres* de l'alphabet différents de $\boldsymbol{\varepsilon}$ et $\boldsymbol{\emptyset}$, le mot vide et l'ensemble vide, respectivement). On a alors que $E_\Sigma \subseteq \Gamma_\Sigma^*$. Est-ce que le langage E_Σ est décidable? Hors contexte? Régulier? Expliquez (ou, mieux, prouvez) votre réponse.

15. (**10 points**) Pour prouver que le langage $\{(ab)^n c^{2n}\}$ n'est pas régulier certains se servent de la tranformation de $(ab)^n c^{2n}$ en $e^n d^n$ et du fait que l'on sait déjà que ce dernier langage n'est pas régulier. Pour justifier cet argument formellement, on peut prouver le lemme suivant.

Lemme. *Soit Σ et Δ deux alphabets et soit $s : \Sigma \to \Delta^*$ une application. Soit $L \subseteq \Sigma^*$. Définissons L_s comme le langage obtenu à partir de L en remplaçant chaque occurrence de $a \in \Sigma$ par $s(a)$, c'est-à-dire,*

$$L_s = \{w \in \Delta^* : w = s(a_1)s(a_2)\dots s(a_k) \text{ tel que } a_1 \dots a_k \in L\}.$$

Alors si L est régulier, L_s l'est également.

Expliquez précisément comment ce lemme justifie l'argument indiqué.

16. **(10 points) Soit *k-COL* le problème défini dans l'ANNEXE. En supposant que *3-COL* est NP-complet, prouvez que $k - COL$ est NP-completpour $k \geq 3$.** Soit 3-Col et k-COL des Graph,

- Si k=3. Comme nous supposons que 3-Col est NP-Complet, nous savons que 3-Col $\leq_p$ 3-Col est trivial. Donc par simple logic si k=3, k-Col $\in$ NP-Complet

- Si k = 4 : Montrons que 4-Col $\in$ NP. On peut généré de facon non-deterministe tout les 4-Coloriages possible pour un coloriage donné (Fait en temps polynomial) On peut ensuite verifier s'il est valide ou non, Si oui: $\to$ accepte, Sinon $\to$ refuse, en boclont sur les arrête du graph. (4-Col $\in$ NP)

3-Col $\leq_p$ 4-Col • On crée $f(\langle G \rangle)$: $V' = V \cup V_{new}$, $E' = E \cup \{uV_{new} \mid u \in V\}$, $G' = (V', E')$
En ajoutant un sommet adjacent à $\forall$ les autre sommet du graph 3-Col, on s'assure que la 4 em couleur (differente des 3 autres) qu'on utilise pour ce sommet transforme le graph 3-Col en 4-Coloriable. Donc $\langle G' \rangle$ est 4-Col.
pour confirmee le tout, allons dans l'autre sens: Si $\langle G' \rangle$ est 4-Coloriable, alors il existe un 4 coloriage G'. Soit c, la couleur utilisee pour V_{new} dans le 4-coloriage. Sachant que V_{new} est adjacent à $\forall$ Les sommet de G', aucun autre sommets ne contient c. Les sommet de G sont donc 3-Coloriable.
$\Rightarrow G \in$ 3-COL.

k-Col = $\{\langle G \rangle \mid G$ est k-coloriable$\}$: On peut utiliser la même méthode pour montré que k-col $\leq_p (k+1)$-Col. La preuve reste la même.

k-Col $\in$ NP-Complet.

17. (**10 points**) Prouvez le lemme de la question 15.

ANNEXE

1. Une machine de Turing est un septuplet $M = (Q, \Gamma, \Sigma, \delta, s, q_A, q_R)$. Elle commence dans l'état initial s sur le premier symbole du mot d'entrée et elle accepte si elle arrive à l'état acceptant q_A. Elle s'arrête sur acceptation ainsi que sur refus. Ce dernier arrive si la machine entre l'état q_R. *Si vous utilisez une autre convention, précisez-la.*

2. On dit qu'une machine de Turing M est polynomiale s'il existe une constante $c \in \mathbb{R}^{>0}$ et un $k \in \mathbb{N}$ tels que M décide tout mot de longueur n en au plus $c \cdot n^k$ étapes (une étape est simplement une application de la fonction de transition de M).

3. Un langage est polynomial s'il est décidé par une machine de Turing déterministe polynomiale.

4. Soit $L_1 \subseteq \Sigma_1^*$ et $L_2 \subseteq \Sigma_2^*$ deux langages. On dit que L_1 *se réduit polynomialement* à L_2 s'il existe une *transformation polynomiale* de L_1 vers L_2, c'est-à-dire, une application $f : \Sigma_1^* \longrightarrow \Sigma_2^*$ telle que

   (a) f peut être calculée une machine de Turing déterministe polynomiale, et
   (b) pour tout $w \in \Sigma_1^*$, $w \in L_1$ si et seulement si $f(w) \in L_2$.

   On écrit $L_1 \leq_p L_2$.

5. *VERTEX COVER* est le problème suivant :

   - **DONNEE** : un graphe $G = (V, E)$, un naturel k
   - **QUESTION** : existe-t-il $S \subseteq V$, $|S| \leq k$, tel que toute arête ait au moins une extremité dans S?

6. *CLIQUE* est le problème suivant :

   - **DONNEE** : un graphe $G = (V, E)$, un naturel k
   - **QUESTION** : existe-t-il $S \subseteq V$, $|S| \geq k$, tel que $uv \in E$ pour tout $u, v, \in S,\ u \neq v$?

7. *STABLE* est le problème suivant :

   - **DONNEE** : un graphe $G = (V, E)$, un naturel k
   - **QUESTION** : existe-t-il $S \subseteq V$, $|S| \geq k$, tel que $uv \notin E$ pour tout $u, v, \in S,\ u \neq v$?

8. *k-COL* est le problème suivant :

   - **DONNEE** : un graphe $G = (V, E)$, un naturel k
   - **QUESTION** : existe-t-il $c : V(G) \longrightarrow \{0, 1, \ldots, k-1\}$ telle que $c(u) \neq c(v)$ si $uv \in E$?

9. Vous pouvez admettre les langages suivants sur $\Sigma = \{0, 1\}$ comme NP-complet(sauf si on vous demande de le prouver).

   - $L_{SAT} = \{\langle \phi \rangle : \phi$ est une formule booléenne en forme normale conjonctive satisfaisable$\}$
   - $L_{3SAT} = \{\langle \phi \rangle : \phi$ est une formule booléenne en forme normale conjonctive dont chaque clause contient exactement trois litéraux et qui est satisfaisable$\}$
   - $L_{CLIQUE} = \{\langle G, k \rangle : G$ est un graphe qui possède une clique de taille au moins $k\}$

10. Vous pouvez admettre les langages suivants sur $\Sigma = \{0, 1\}$ comme indécidables (sauf si on vous demande de le prouver).

    - $L_d = \{w : w = w_i \text{ et } w_i \notin L(M_i)\} = \{\langle M \rangle : \langle M \rangle \notin L(M)\}$
    - $\overline{L_d} = \{w : w = w_i \text{ et } w_i \in L(M_i)\} = \{\langle M \rangle : \langle M \rangle \in L(M)\}$
    - $L_e = \{\langle M \rangle : L(M) = \emptyset\} = E_{TM}$
    - $L_* = \{\langle M \rangle : L(M) = \Sigma^*\}$

- $L_u = \{\langle M, w\rangle : w \in L(M)\} = A_{TM} = L_{M,w}$
- $\overline{L_u} = \{\langle M, w\rangle : w \notin L(M)\} = \overline{A_{TM}} = \overline{L_{M,w}}$
- $L_r = \{\langle M\rangle : L(M) \text{ est décidable}\}$
- $L_{nr} = \{\langle M\rangle : L(M) \text{ n'est pas décidable}\}$