Ceci est une collection de vielles questions et vieux examens. Il y a sans doute pas mal de répétition, mais cela donne une idée de la sorte de questions qui apparaissent sur nos examens.

1. Tout langage régulier est décidable. OUI NON

2. Si un langage L est récursif et si son complément est récursivement énumérable alors L est récursivement énumérable. OUI NON

3. Tout langage décidable est régulier. OUI NON

4. Si un langage est accepté par une machine de Turing non-déterministe alors il est accepté par une machine de Turing déterministe. OUI NON

5. Si L_1 est un langage indécidable est si L_2 se réduit à L_1 alors L_2 est NP-complet. OUI NON

6. **P** contient des langages indécidables. OUI NON

7. **NP** contient des langages indécidables. OUI NON

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

9. Une machine de Turing M non-déterministe polynomiale peut être simulée par une machine de Turing déterministe prenant un temps exponentiel. OUI NON

10. On peut considérer **NP** comme la classe des problèmes décidables en temps exponentiel. OUI NON

11. (**9 + 9 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!*

12. (**10 points**) Donnez une formulation précise du lemme d'Ogden et dites à quoi il sert.

13. (**10 points**) Prouvez que tout langage régulier est hors-contexte.

14. (**10 points**) Trouvez le ou les erreur(s) dans la preuve suivante qui devrait montrer que le langage $L = \{a^k b^k : k \geq 0\}$ sur l'alphabet $\Sigma = \{0, 1, a, b, c, \#, \$\}$ n'est pas hors-contexte.

    Preuve. *Soit* n *la constante du lemme d'Ogden et soit* $\mathrm{a^n b^n}$ *un mot du langage* L. *Soit les* a *marqués, c'est-à-dire, soit les positions des* a *distinguées. Alors, par le lemme d'Ogden,* $\mathrm{a^n b^n}$ *peut être écrit comme* uvwxy *de sorte que les conditions du lemme d'Ogden soient vérifiées. Mais alors c'est uniquement les* a *que l'on peut "pomper" puisque ce sont les seuls dans des positions distinguées. Donc on obtiendra un mot de la forme* $\mathrm{a^m b^n}$ *avec* $\mathrm{m > n}$ *qui n'est pas dans* L. *Ceci contredit la conclusion du lemme d'Ogden et on en conclut que* L *n'est pas hors-contexte.*

15. (**10 points**) Donnez la définition de "L est NP-complet".

16. (**10 points**) Pour prouver que le langage $\{(ab)^n c^{2n}\}$ n'est pas régulier certains se servent de la transformation 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,*

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

    Alors L est régulier si et seulement si L_s l'est.

    Expliquez précisément comment ce lemme justifie l'argument indiqué.

17. (**9 points**) Soit $L_1, L_2 \in \mathbf{RE}$. Montrez que $L_1 \cup L_2 \in \mathbf{RE}$.

18. (**5 + 5 points**) Répondez aux questions suivantes en donnant une brève justification.

(a) Soit $L \notin \mathbf{RE}$. Dans quelles classes (précisément) pourrait-on retrouver $\overline{L}$? Encerclez suivant votre réponse et justifiez.

R $\qquad$ $\mathbf{RE} \setminus \mathbf{R}$ $\qquad$ autre

(b) Soit $L_1 \in \mathbf{RE} \setminus \mathbf{R}$ et $L_2 \notin \mathbf{RE}$. Est-ce que $L_1 \cup L_2$ pourrait-être décidable? (Si oui, donnez un exemple, si non, justifiez.)

19. (**10 + 10 point**) Un TP de IFT2102 demande d'écrire une machine de Turing décidant un certain langage. Pour la correction, Louis, 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 $L_X = \{\langle M \rangle |\ \forall w \in X, M(w) \text{ accepte }.\}$

(a) Prouvez que L_X est indécidable (attention aux détails).

(b) Prouvez que $L_X \in \mathbf{RE}$.

20. (**5 + 10 points**) Soit $k \geq 1$. On définit le problème de coloriage de graphe comme suit:

Nom: k-*COL*

Donnée: (V, E) un graphe, k un entier positif

Question: Peut-on colorier les sommets du graphe avec k couleurs de sorte qu'aucune aête $\{u, v\} \in E$ n'ait ses deux sommets de la même couleur? Formellement, existe-il une fonction $f : V \to \{1, \ldots, k\}$ telle que $\forall \{u, v\} \in E,\ f(u) \neq f(v)$?

(a) Soit $k \geq 1$. Prouvez que k-$COL \in \mathbf{NP}$

(b) Prouvez que 3-$COL \propto$ 4-COL.

21. (**9 + 9 points**) Faites DEUX des trois problèmes suivants.

(a) Soit $L_1, L_2 \in \mathbf{NP}$ et L_1 NP-complet. Prouvez que L_2 est NP-complet si et seulement si $L_1 \propto L_2$.

(b) Supposez que $\mathbf{P} \neq \mathbf{NP}$. Supposons que l'on sait que $L \in \mathbf{NP} \setminus \mathbf{NPC}$ mais que l'on ne sait pas s'il est dans $\mathbf{P}$. Si l'on trouve un algorithme polynomial pour décider L, quelle serait l'influence de ce résultat sur $\mathbf{P}$? Justifiez-le.

(c) On sait que $\mathbf{P} \subseteq \mathbf{NP}$. Soit L un langage NP-complet. Prouvez en détail que si $L \in \mathbf{P}$ alors $\mathbf{P} = \mathbf{NP}$.

BONIS

1. (**15 points**) Prouvez le lemme de la question 16. *Indication: Une direction ne demande qu'une ou deux lignes. L'autre est un peu plus longue mais pas compliquée.*

2. (**5 + 5 points**) Le bilan de notre exploration des limites de calculabilité n'est pas très reluisant:

   - Le problème d'arrêt (et toutes ses variantes) pour une machine de Turing est indécidable.
   - Déterminer si un langage donné possède une propriété non-triviale est indécidable.

   Même lorsque nous ne considérons que les problèmes décidables, les choses ne sont guère plus reluisantes puisque pour beaucoup de problèmes importants nous ne connaissons pas de solution efficace.

   (a) Expliquez pourquoi ces résultats ne signifient pas la mort de l'informatique.

   (b) Justifiez l'étude de l'indécidabilité et de la calculabilité

ANNEXE

Vous pouvez considérer que les langages suivants sont indécidables.

- $L_0 = \{w|\ w = w_i \text{ et } M_i(w_i) \text{ refuse/boucle à l'infini } \}$
- $\overline{L_0} = \{w|\ w = w_i \text{ et } M_i(w_i) \text{ accepte } \}$
- $L_{vide} = \{\langle M\rangle|\ L(M) = \emptyset\ \}$
- $L_\varepsilon = \{\langle M\rangle|\ L(M) = \{\varepsilon\}\ \}$
- $L_w = \{\langle M\rangle|\ M(w) \text{ accepte } \}$ pour un $w \in \Sigma^*$ fixé.
- $L_{\Sigma^*} = \{\langle M\rangle|\ L(M) = \Sigma^*\ \}$
- $L_{\overline{\Sigma^*}} = \{\langle M\rangle|\ L(M) \neq \Sigma^*\ \}$
- $LU = \{\langle M, w\rangle|\ M(w) \text{ accepte } \}$
- $\overline{LU} = \{\langle M, w\rangle|\ M(w) \text{ refuse/boucle à l'infini } \}$
- $L_{reg} = \{\langle M\rangle|\ L(M) \text{ est régulier } \}$
- $L_{rec} = \{\langle M\rangle|\ L(M) \text{ est décidable } \}$
- $L_{\overline{rec}} = \{\langle M\rangle|\ L(M) \text{ est indécidable } \}$
- $L_{\overline{équiv}} = \{\langle M_1, M_2\rangle|\ \exists w \in \Sigma^* \text{ tq } M_1(w) \text{ et } M_2(w) \text{ n'arrêtent pas en même temps } \}$

1. (**15 x 2 points**) Pour chacun des énoncés suivants indiquez s'il est vrai ou non en encerclant OUI ou NON. *AUCUNE JUSTIFICATION N'EST NECESSAIRE*

   (a) Il n'y a aucun langage régulier indécidable. OUI NON

   (b) Tout automate à pile non-déterministe est équivalent à un automate à pile déterministe. OUI NON

   (c) Tout langage décidable est régulier. OUI NON

   (d) Si un langage est accepté par une machine de Turing non-déterministe alors il est accepté par une machine de Turing déterministe. OUI NON

   (e) Tout langage accepté par un automate fini est récursif. OUI NON

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

   (g) Le complément d'un langage généré par une grammaire hors-contexte est accepté par un automate à pile non-déterministe. OUI NON

   (h) Toute machine de Turing qui accepte un langage récursif s'arrête sur toute entrée. OUI NON

   (i) **P** contient un langage accepté par une seule machine de Turing déterministe. OUI NON

   (j) L'hypothèse de Church dit, que pour reconnaître un langage qui n'est pas récursivement énumérable, il faut prier. OUI NON

   (k) Une machine de Turing M non-déterministe polynomiale peut être simulée par une machine de Turing déterministe prenant un temps exponentiel. OUI NON

   (l) Tout problème **NP**-complet est décidable. OUI NON

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

   (n) Toute expression régulière est acceptée par une machine de Turing. OUI NON

   (o) Si X est un problème **NP**-complet et Y peut être polynomialement réduit à X alors Y est également **NP**-complet. 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 ....

- Question ....

3. (**5 + 5 points**) Répondez aux questions suivantes en donnant une brève justification.

   (a) Soit $L \notin \mathbf{RE}$. Dans quelles classes (précisément) $\overline{L}$ ne pourrait-il jamais appartenir ? Encerclez suivant votre réponse et justifiez.

   R $\qquad$ **RE** $\setminus$ **R** $\qquad$ autre

   (b) Soit $L_1 \in \mathbf{RE} \setminus \mathbf{R}$ et $L_2 \notin \mathbf{RE}$. Est-ce que $L_1 \cup L_2$ pourrait être décidable? (Si oui, donnez un exemple, si non, justifiez.)

4. (**10 points**) Dessinez un automate fini déterministe qui accepte le langage $\{a^m b^m c^k : 0 \leq m \leq 2,\ k \geq 2\}$. *Aucune justification n'est nécessaire.*

5. (**10 points**) Justifier que votre automate de la question 4 accepte le langage qu'il est censé accepter.

6. (**10 points**) Est-ce que l'automate construit en question 4 est minimal? Prouvez votre réponse.

7. (**10 + 5 points**)

   (a) Prouvez que le langage $\{a^n b^m : n \neq m\}$ sur l'alphabet $\{a, b, c\}$ n'est pas régulier.

   (b) Donnez une grammaire hors-contexte qui génère le langage de la première partie de la question. Justifiez votre réponse.

8. (**10 points**) Soit $L_1, L_2 \in \mathbf{RE}$. Montrez que $L_1 \cap L_2 \in \mathbf{RE}$

9. (**10 points**) Donnez la description formelle d'une machine de Turing décidant le langage $L = \{\epsilon\}$ sur l'alphabet $\Sigma = \{a, b\}$.

10. (**5 + 10 + 5 points**) Soit $k \geq 1$. On définit le problème de coloriage de graphe comme suit:

    Nom: k-*COL*

    Donnée: (V, E) un graphe, k un entier positif

    Question: Peut-on colorier les sommets du graphe avec k couleurs de sorte qu'aucune arête $\{u, v\} \in E$ n'ait ses deux sommets de la même couleur? Formellement, existe-il une fonction $f : V \rightarrow \{1, \ldots, k\}$ telle que $\forall \{u, v\} \in E,\ f(u) \neq f(v)$?

    Des descriptions des algorithmes sont suffisantes (par l'hypothèse de Church!) dans les deux premières parties de cette question.

    (a) Soit $k \geq 1$. Prouvez que k-*COL* $\in \mathbf{NP}$

    (b) Prouvez que k-*COL* $\propto (k+1)$-*COL* pour tout $k \geq 1$.

    (c) Déduisez que 3-*COL*$\propto$ k-*COL* pour tout $k \geq 3$.

11. **(7 + 8 points)**

    (a) Trouvez la ou les erreurs dans la preuve suivante du fait (?) que pour tout alphabet Σ, le langage Σ^* est hors-contexte.

    Preuve: Soit n *la constante du lemme du pompiste pour les langages hors-contexte et soit* $w \in \Sigma^*$, $|w| \geq n$. *On peut facilement écrire* w *comme* $uvxyz$ *où* $|vxy| \leq n$ *et* $|vy| \geq 1$. *On a que pour tout* $i \geq 0$, $uv^ixy^iz \in \Sigma^*$. *Donc, par le lemme du pompiste pour les langages hors-contexte,* Σ^* *est un langage hors-contexte.*

    (b) Prouvez rigoureusement que Σ^* est un langage hors-contexte pour tout alphabet Σ.

12. **(20 points)** Un de vos amis - qui est démonstrateur en IFT 1010 - vous fait part de son projet pour aider à corriger les TP dans le cours. Il a commencé à faire un programme qui prend en entrée le code-source de deux fonctions quelconques en Pascal standard et qui détermine si elles sont équivalentes. Il vous explique que deux fonctions sont équivalentes si elles donnent la même réponse dans tous les cas, plantent toutes les deux ou bouclent toutes les deux. Grâce à son programme, il pourra tester facilement les TP des étudiants en comparant leurs programmes à sa solution personnelle (qu'il suppose bonne). Il veut que vous l'aidiez dans son projet. Que lui dites vous? Prouvez votre réponse.

13. **(20 points)** Une machine de Turing M déterministe décide un langage L sur Σ en espace polynomial s'il existe un polynôme $p(n)$ tel que le nombre de cases sur le ruban utilisées par M dans son calcul sur x est borné par $p(|x|)$, pour tout $x \in \Sigma^*$. Soit **PSPACE** la classe de langages (problèmes) décidés par une machine de Turing déterministe en espace polynomial. Prouvez que $Poly \subseteq$ **PSPACE**

BONI (**10 + 10 point**) Un TP de IFT2102 demande d'écrire une machine de Turing décidant un certain langage. Pour la correction, Louis, 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 $L_X = \{\langle M \rangle |\ \forall w \in X, M(w) \text{ accepte .}\}$

1. Prouvez que L_X est indécidable (attention aux détails).
2. Prouvez que $L_X \in$ **RE**.

ANNEXE

Vous pouvez considérer que les langages suivants sont indécidables.

- $L_0 = \{w |\ w = w_i \text{ et } M_i(w_i) \text{ refuse/boucle à l'infini }\}$
- $\overline{L_0} = \{w |\ w = w_i \text{ et } M_i(w_i) \text{ accepte }\}$
- $L_{vide} = \{\langle M \rangle |\ L(M) = \emptyset\ \}$
- $L_\varepsilon = \{\langle M \rangle |\ L(M) = \{\varepsilon\}\ \}$
- $L_w = \{\langle M \rangle |\ M(w) \text{ accepte }\}$ pour un $w \in \Sigma^*$ fixé.
- $LU = \{\langle M, w \rangle |\ M(w) \text{ accepte }\}$
- $\overline{LU} = \{\langle M, w \rangle |\ M(w) \text{ refuse/boucle à l'infini }\}$

RAPPELS

1. Une machine de Turing est un septuplet $M = (Q, \Gamma, \Sigma, \delta, s, B, F)$. Elle commence dans l'état initial s sur le premier symbole du mot d'entrée et elle accepte si elle arrive à un état acceptant $q \in F$. *Si vous utilisez une autre convention, précisez-la!!!*q
2. 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 *tranformation 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 en temps polynomial par une machine de Turing déterministe
   (b) pour tout $w \in \Sigma_1^*$, $w \in L_1$ si et seulement si $f(w) \in L_2$.

   On écrit $L_1 \propto L_2$.
3. Rappel de certaines abréviations utilisées dans le texte:
   - **NPC** est la classe de problèmes (ou, plus précisément, de langages) **NP**-complets.
   - **R** est la classe de langages récursifs
   - **RE** est la classe de langages récursivement énumérables

1. (**9 points**) Pour chacun des langages suivants, dites s'il est régulier (R), hors-contexte mais pas régulier (HC), ou pas hors-contexte (A). Aucune justification n'est nécessaire.
   (a) $\{(a^2b^4ab)^{3k} : k \geq 0\}$
   (b) $\{a^{2n}b^{3n} : n \geq 7\}$
   (c) $\{a^{2n}b^{3n} : n \leq 7\}$
2. (**10 points**) Soit $L_1, L_2 \in$ **RE**. Montrez que $L_1 \cap L_2 \in$ **RE**

3. (**15 points**) Répondez par **vrai** ou **faux** aux questions suivantes et donnez une brève justification.

   (a) Soit une machine de Turing $M = (Q, \Gamma, \Sigma, s, B, F, \delta)$ et soit deux configurations C_1 et C_2 telle que $C_1 \vdash_M^* C_2$ et C_2 est une configuration acceptante. Alors il existe au moins une exécution de M qui terminera en C_2.

   (b) Soit M et M' deux machines de Turing. Soit L le langage accepté par M. Si M' décide L alors ces deux machines sont forcément différentes.

   (c) Soit L un langage accepté par un automate fini déterministe. Aucune machine de Turing acceptant L n'aura d'exécutions infinies.

4. (**14 points**) Donnez une grammaire pour générer le langage $L = \{w \in \{a,b\}^* : w \text{ est un palindrome}\}$ (un palindrome est un mot qui est pareil lu de chaque côté, par exemple aaabbaabaabaabbaaa or aababaa). *Indication: trouvez d'abord une définition récursive d'un palindrome.*

5. (**12 points**) Donnez une description formelle d'un automate à pile non-déterministe qui accepterait le langage $\{(ab)^n c^{2n} : n \geq 1\}$.

6. (**30 points**) Selon notre définition de la machine de Turing, la tête de lecture/écriture doit se déplacer *et* écrire à chaque étape. Certains auteurs préfèrent des machines qui *soit* déplacent la tête sans écrire, *soit* écrivent sans déplacer la tête. Ces auteurs utilisent donc la même définition de machines de Turing que nous sauf que leur fonction de transition est de la forme

$$\delta : Q \times \Gamma \to Q \times (\Gamma \cup \{L, R\})$$

   Evidemment, les notions de dérivation, de dérivation éventuelle, de langage décidé et accepté demeurent les mêmes.

   (a) Prouvez que si L est décidé par une machines de Turing définie selon notre définition, alors L est aussi décidé par une machine de Turing définie selon cette définition alternative.

   (b) Soit L décidé par une machine de Turing M définie selon cette définition alternative. **Ne prouvez pas** qu'il existe une machine de Turing M' selon notre définition décidant L, mais donnez la construction de la **fonction de transition de** M' qui servirait dans une telle preuve.

7. (**10 points**) Donnez la description formelle d'une machine de Turing décidant le langage $L = \{\epsilon\}$ sur l'alphabet $\Sigma = \{a, b\}$.

BONI (**10 points**) Prouvez que votre réponse à la question 3 est correcte.

1. (**30 points**) Pour chacun des langages suivants, donnez soit une grammaire hors contexte qui le génère, soit une preuve qu'il n'est pas hors-contexte.

   (a) (15 points) $L_1 = \{w \in \{x, y, z\}^* : \text{ le nombre d'occurrences de } x \text{ est égal au nombre d'occurences de } y \text{ et } z\}$.

   (b) (15 points) $L_2 = \{(a^n b^n)^m : m, n \geq 0\}$

2. (**20 points**)*Soit $L = \{(ab)^n : n \geq 0\}$ un langage sur $\{a, b, c\}$. Les classes d'équivalence de la relation R_L sont C_i, $i \geq 0$, définies par $C_i = [(ab)^i]$. Le nombre de ces classes étant infini, le langage L n'est pas régulier.*

   Complétez la preuve ci dessus en montrant que le nombre de classes d'equivalence est, en effet, infini. Alternativement, trouvez l'erreur (les erreurs?) dans la "preuve", expliquez la (les), et prouvez que L est régulier.

3. (**30 points**) Vous passez un examen qui vous demande d'écrire une machine de Turing qui décide un certain langage L. Vous le faites, mais tout d'un coup vous vous rendez compte que tout ce que vous avez fait suppose que le mot d'entré soit au début du ruban infini en un sens (à droite) tandis que la machine que vous avez définie suppose l'entrée enfermée entre deux \$ au début du ruban. Au lieu de recommencer, vous décidez de modifier la machine. Comment? C'est-à-dire, en supposant que vous ayez construit $M = (Q, \Gamma, \Sigma, \delta, s, B, F)$, construisez M' équivalente mais avec le mot d'entrée où il le faut.

   (Il ne suffit pas de décrire la construction de M', il faut donner les détails.)

4. (**20 points**) Donnez une définition précise de chacun des concepts suivants

   (a) (5 points) un langage récursif

   (b) (5 points) un langage récursivement énumérable

   (c) (5 points) un langage décidé par une machine de Turing

   (d) (5 points) un langage diagonal L_d (donnez également les suppositions nécessaires)

1. (**10 points**) Expliquez ce que veut dire $(q, w) \overset{*}{\vdash} (p, v)$.

2. (**10 points**) Donnez une définition *précise* du langage accepté par un automate fini. Est-ce qu'il est nécessaire de distinguer entre un automate fini déterministe et un automate fini non-déterministe? Justifiez votre réponse.

3. (**9 points**) Lesquelles des expressions suivante sur $\Sigma = \{0, 1, a, b, c, \%\}$ sont régulières **d'après la définition formelle** (aucune justification nécessaire).

   (a) (**3 points**) $a^*bbbbb0011$

   (b) (**3 points**) $(((0)^5)((1)^3))$

   (c) (**3 points**) $((((a)^*bbbbbbb) \cup (00 \cup 1))(\%^*))$

4. (**15 points**) Prouvez que tout langage fini est régulier.

5. (**14 points**) Pour chacun des langages suivants sur $\Sigma = \{a, b, c\}$ dites s'il est régulier ou non et *prouvez votre réponse.* Les paramètres k sont des entiers non-négatifs.

   (a) (**7 points**) $\{(abab)^k c^{2k} : k \geq 3\}$

   (b) (**7 points**) $\{(abab)^k c^{2k} : 0 \leq k \leq 3\}$

6. (**30 points**) Soit $L = \{w \in \{0, 1\}^* :$ *chaque paire de 1 consécutifs suit au moins une paire de 0 consécutifs*$\}$.

   (a) (**15 points**) Donnez un automate fini qui accepte L.

   (b) (**15 points**) Donnez une expression régulière qui représente L.

7. **(12 points)** Prouvez que pour tout langage L sur un alphabet Σ, $(L^*)^* = L^*$.

BONI (**5 points**) Dans le théorème de Myhill-Nerode sont définies deux relations sur Σ^*, R_M et R_L.

1. (**4 points**) Définissez l'une des deux.

2. (**1 point**) L'automate fini déterministe acceptant le langage donné L et tel que le nombre de ses états est minimum est défini en prenant comme l'ensemble d'états les classes d'équivalence de laquelle de ces deux relations?

1. (**10 points**) Donnez une définition *formelle* et précise d'un automate fini déterministe.

2. (**20 points**) Prouvez que tout langage non-régulier contient un nombre infini de langages réguliers.

3. (**15 points**) Construisez un automate fini capable de décider si l'usager a donné un nom correct à un fichier DOS, c'est-à-dire, un nom de la forme $N.E$ où N est un nom de longueur au moins 1 et au plus 8, E un mot de longueur au plus 3, aucun ne contenant des symboles de l'ensemble I qui comprend les 17 symboles `| \ , . < > ? / : ; " [ ] * + =` **B**
   (ici **B** est le symbole qui représente le "blanc" ou l'espace). Le reste de l'alphabet contient les lettres de l alphabet normal (26 en tout) plus les chiffres $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. *C'est plus simple que cela n'ait l'air. Il ne faut pas plus que 15 états.*
   Aucune justification nécessaire.

4. (**15 points**) Donnez une expression régulière pour le langage accepté par l'automate construit en question 3. Des abbréviations expliquées sont acceptables.
   Aucune justification nécessaire.

5. (**20 points**) Les langages suivants sont-ils réguliers? Prouvez vos réponses.
   (a) Les mots sur $\{a, b\}$ contenant au moins autant de a que de b.
   (b) Les mots sur $\{a, b\}$ dans lesquels la différence entre le nombre de a et le nombre de b est divisible par 3.

6. (**20 points**) Prouvez que le complément d'un langage régulier est un langage régulier.

BONI (**10 points**). Prouvez qu'il est impossible d'écrire un programme qui décide si un programme quelconque s'arrète.

1. (**20 points**) Soit M un automate fini et soit w un mot sur l'alphabet Σ. Définissez *précisément* ce que veut dire M *accepte* w. Distinguez entre un automate déterministe et un automate non déterministe et expliquez pourquoi une telle distinction est, ou n'est pas, nécessaire.

2. (**20 points**) Quelles sont les classes d'équivalence de la relation R_L sur $\{a, b, c\}^*$ pour le langage décrit par l'expression régulière $(((a)(a \cup b)^*) \cup (b))$?. Utilisez-les pour construire un automate fini acceptant $L(((a)(a \cup b)^*) \cup (b))$

3. (**10 points**) Construisez un automate fini qui accepte les mots sur $\Sigma = \{0, 1, 2\}$ dans lesquels pour chaque paire de symbols consécutifs leur somme est au plus deux (addition habituelle).

4. (**10 points**) Donnez une expression régulière pour le langage accepté par l'automate construit en question 3.

5. (**20 points**) Voici une question du livre:
   Pour chacun des langage suivants, donner une expression régulière représentant son complément.
   (a) $(a \cup b)^* b$
   (b) $((a \cup b)(a \cup b))^*$

   Pourquoi est-elle mal posée? Réparez-la et répondez à une des parties corrigées.

6. (**20 points**) Prouvez que l'intersection de deux langages réguliers est un langage régulier.

BONI (**10 points**) Etant donné un automate fini déterministe $M = (Q, \Sigma, \delta, q_0, F)$ avec
$\delta : Q \times \Sigma \longrightarrow Q$, expliquez comment étendre δ aux mots sur Σ. C'est-à-dire, définissez
$\delta^* : Q \times \Sigma^* \longrightarrow Q$.

1. (**15 points**) Soient $\mathcal{L}_R$, $\mathcal{L}_{HC}$, $\mathcal{L}_{MT}$ les classes de langages acceptées, respectivement, par des automates finis, des automates à pile et des machine de Turing. Classez-les dans l'ordre croissant par inclusion et donnez un exemple pour chaque inclusion d'un langage qui appartient à l'une des classes et pas à l'autre. Aucune preuve n'est demandée.

2. (**15 points**) Donnez une grammaire hors-contexte pour le langage $\{a^m b^n : m, n \geq 0, m \neq n\}$.

3. (**15 points**) En supposant la question 2 faite, prouvez que le langage $\{a^m b^n c^p : m, n, p \geq 0$ et $m \neq n$ ou $n \neq p\}$ est hors-contexte (il est possible de le faire en se servant simplement des propriétés des langages hors-contexte).

4. (**20 points**) Pour chacun des langages suivants dites s'il est ou n'est pas régulier. Justifiez votre réponse.

   (a) $\{a^{3k} b^{4k} : 0 \leq k \leq 10\}$
   (b) $\{a^{3k} b^{4k} : k \geq 10\}$
   (c) $\{(a^3 b^4)^k : k \geq 10\}$

5. (**20 points**) Prouvez que le langage $\{a^{2k} b^{3k} c^{4k} : k \geq 0\}$ n'est pas hors-contexte.

6. (**15 points**) Décrivez une machine de Turing qui accepte le langage de la question 5.

BONI (**10 points**) Prouvez que, si, dans la définition d'une machine de Turing, on permet que la tête reste sur place alors, pour chaque machine obtenue par cette nouvelle définition, il existe une machine de Turing définie par la définition du cours qui lui est équivalente (expliquez le sens d'*équivalente* dans votre réponse).

1. **(5 + 5 points)**

   (a) Trouvez la ou les erreurs dans la preuve suivante du fait (?) que pour tout alphabet Σ, le langage Σ^* est régulier.

   Preuve: Soit n la constante du lemme du pompiste et soit $w \in \Sigma^*$, $|w| \geq n$. On peut facilement écrire w comme uxy où $|ux| \leq n$ et $|x| \geq 1$. On a que pour tout $i \geq 0$, $ux^i y \in \Sigma^*$. Donc, par le lemme du pompiste, Σ^* est un langage régulier.

   (b) Prouvez que Σ^* est un langage régulier pour tout alphabet Σ.

2. **(10 points)** Prouvez que la relation $\leq_m^p$ est transitive.

3. **(20 points)** Pour chacune des questions suivantes encerclez la bonne réponse (OUI ou NON)

   (a) Tout langage régulier est accepté par une machine de Turing non-déterministe. OUI NON
   (b) Soit L_1, L_2 deux langages de problème. Si $L_1 \leq_T^p L_2$ alors $L_1 \leq_m^p L_2$. OUI NON
   (c) Si X est un problème $NP-$complet et si $Y \leq_m^p X$ alors Y est $NP-$complet. OUI NON
   (d) Tout langage récursif est récursivement énumérable. OUI NON
   (e) Un langage L sur l'alphabet Σ est récursif si et seulement si L et son complément (dans Σ^*) $\overline{L}$ sont récursivement énumérables. OUI NON
   (f) Si un langage est accepté par une machine de Turing non-déterministe alors il est accepté par une machine de Turing déterministe. OUI NON
   (g) Toute machine de Turing déterministe acceptant un langage récursif s'arrête sur toute entrée. OUI NON
   (h) Il n'y a aucun langage régulier indécidable. OUI NON
   (i) Le langage $L_1 = \{a^n b^n c^n : n \geq 0\}$ est hors-contexte si et seulement si le langage $L_2 = \{0^k (01)^k 1^k : k > 0\}$ l'est. OUI NON
   (j) Pour n'importe quel alphabet, le langage des palindromes est décidable. OUI NON

4. **(10 + 10 points)** Justifiez DEUX de vos réponses aux questions en 3, c'est-à-dire, prouvez vrais ou faux deux des énoncés de la question 3. *Votre réponse ne vaut rien si elle ne comprend pas l'énoncé de ce que vous prouvez!*

5. **(10 points)** Un de vos amis - qui est démonstrateur en IFT 1010 - vous fait part de son projet pour aider à corriger les TP dans le cours. Il a commencé à faire un programme qui prend en entrée le code-source de deux fonctions quelconques en Pascal standard et qui détermine si elles sont équivalentes. Il vous explique que deux fonctions sont équivalentes si elles donnent la même réponse dans tous les cas, plantent toutes les deux ou bouclent toutes les deux. Grâce à son programme, il pourra tester facilement les TP des étudiants en comparant leurs programmes à sa solution personnelle (qu'il suppose bonne). Il veut que vous l'aidiez dans son projet. Que lui dites vous? Prouvez votre réponse.

6. **(10 + 10 + 5 points)** Une de vos amies vous apprend qu'elle a fait un programme qui résout le problème TAUTOLOGIE (défini plus bas) en temps polynomial. Elle vous fournit le code de son programme en vous demandant votre avis (voir l'ANNEXE si vous désirez revoir les définitions formelles de *formule*, *fonction de vérité* et *satisfaisable*)

   TAUTOLOGIE.
   (Une formule est une tautologie si elle est *toujours* vraie.)
   Donnée: Une formule booléenne ψ avec n variables $x_1, \ldots, x_n$.
   Question: Est-ce que $v(\psi) = 1$ pour TOUTE fonction de vérité v?

   (a) Prouvez que si son programme fonctionne bien, alors vous pouvez résoudre le problème SATISFAISABILITE (défini plus bas) en temps polynomial.
   SATISFAISABILITE.
   (Une formule est satisfaisable si elle *peut* être vraie.)
   Donnée: Une formule booléenne ψ avec n variables $x_1, \ldots, x_n$.
   Question: Est-ce que ψ est satisfaisable?

   (b) Vous dites à un professeur de IFT 2102 que vous pouvez résoudre SATISFAISABILITE en temps polynomial. Il vous regarde d'un air sceptique et vous dit "Mais ce problème est $NP-$complet!". En supposant que $\mathbf{P} \neq \mathbf{NP}$, prouvez que le programme de votre amie ne peut pas fonctionner comme elle le prétend.

   (c) En supposant que $\mathbf{P} = \mathbf{NP}$, montrez que SATISFAISABILITE est effectivement dans $\mathbf{P}$.

7. **(10 points)** Soit **kC** le problème :
   Donnée: un graphe G
   Question: Est-il possible de colorier les sommets de G avec k couleurs de façon à ce que des sommets adjacents aient des couleurs différentes?

   En supposant que $5C$ est $NP-$complet, prouvez que $6C$ l'est aussi.

8. **(10 points)** Voici une "preuve" que $\mathbf{P} = \mathbf{NP}$.

   Soit $\mathbf{NP}$ *partitionné en deux classes,* $\mathbf{P}$ *et* $\mathbf{NP} \setminus \mathbf{P}$. *On prouve que tous les éléments de* $\mathbf{NP}$ *sont dans la même classe et on en déduit - parce qu'on sait que* $\mathbf{P}$ *n'est pas vide - que* $\mathbf{P} = \mathbf{NP}$. *La preuve du fait que tous les problèmes dans* $\mathbf{NP}$ *sont dans la même classe de la partition en* $\mathbf{P}$ *et* $\mathbf{NP} \setminus \mathbf{P}$ *est par induction. Pour tout ensemble* $S \subset \mathbf{NP}$ *ne contenant qu'un seul élément l'énoncé* tous les éléments de S sont dans la même classe *est évidemment vrai. Supposons donc l'énoncé vrai pour tout ensemble avec au plus* $n-1$ *éléments. Soit* $S \subseteq \mathbf{NP}$ *un ensemble avec* n *éléments. Soit* $X \in S$ *et soit* $S_1, S_2 \subset S$ *tels que* $S = S_1 \cup S_2$, $X \in S_1 \cap S_2$ *et* $|S_1|, |S_2| \leq n-1$. *Par l'hypothèse d'induction, tous les éléments de* S_1 *et de* S_2 *sont dans la même classe que* X, *donc tous les éléments de* S *le sont également.*

   Qu'est ce qui ne va pas?

9. **(BONI: 20 points)** On a vu que la classe $\mathcal{L}_R$ de langages réguliers est fermée par rapport à la réunion, l'intersection et la complémentation. Soit $\mathcal{L}$ la plus petite classe de langages contenant tous les langages finis qui est fermée par rapport à ces opérations. Quelle est la relation entre $\mathcal{L}$ et $\mathcal{L}_R$?

ANNEXE

DEFINITIONS. Soit $X_n = \{x_1, x_2, \ldots, x_n\}$ un ensemble de *variables booléennes* (i.e. qui prennent des valeurs 0 ou 1). Soit $\vee, \wedge$ et $^-$des symboles pour "ou", "et" et "non", respectivement.

- Une *formule booléenne* est définie récursivement:
  - x_i et $\overline{x_i}$ sont des formules,
  - si ψ_1 et ψ_2 sont des formules alors $(\psi_1 \wedge \psi_2)$, $(\psi_1 \vee \psi_2)$, et $(\overline{\psi_1})$ sont également des formules,
  - aucune autre combinaison des symboles $x_i, \wedge, \vee, \bar{}, (,)$ n'est une formule.
- Une fonction de *vérité* est une fonction $v : X_n \to \{0, 1\}$ étendue aux formules par:

- $v((\overline{\psi})) = 0$ si $v(\psi) = 1$ et $v((\overline{\psi})) = 0$ sinon,
- $v((\psi_1 \wedge \psi_2)) = 1$ si et seulement si $v(\psi_1) = v(\psi_2) = 1$,
- $v((\psi_1 \vee \psi_2)) = 0$ si et seulement si $v(\psi_1) = v(\psi_2) = 0$.

- Une formule ψ est *satisfaisable* **s'il existe** une fonction de vérité v telle que $v(\psi) = 1$.
- Une formule ψ est une *tautologie* si $v(\psi) = 1$ **pour toute** fonction de vérité $v : X_n \to \{0, 1\}$

1. (**10 points**) Expliquez ce que veut dire $(q, w) \overset{*}{\vdash} (p, v)$.
2. (**10 points**) Donnez une définition *précise* du langage accepté par un automate fini. Est-ce qu'il est nécessaire de distinguer entre un automate fini déterministe et un automate fini non-déterministe? Justifiez votre réponse.
3. (**9 points**) Lesquelles des expressions suivante sur $\Sigma = \{0, 1, a, b, c, \%\}$ sont régulières **d'après la définition formelle** (aucune justification nécessaire).
   (a) (**3 points**) $a^*bbbbb0011$
   (b) (**3 points**) $(((0)^5)((1)^3))$
   (c) (**3 points**) $((((a)^*bbbbbbb) \cup (00 \cup 1))(\%^*))$
4. (**15 points**) Prouvez que tout langage fini est régulier.
5. (**14 points**) Pour chacun des langages suivants sur $\Sigma = \{a, b, c\}$ dites s'il est régulier ou non et *prouvez votre réponse*. Les paramètres k sont des entiers non-négatifs.
   (a) (**7 points**) $\{(abab)^k c^{2k} : k \geq 3\}$
   (b) (**7 points**) $\{(abab)^k c^{2k} : 0 \leq k \leq 3\}$
6. (**30 points**) Soit $L = \{w \in \{0, 1\}^* :$ *chaque paire de 1 consécutifs suit au moins une paire de 0 consécutifs*$\}$.
   (a) (**15 points**) Donnez un automate fini qui accepte L.
   (b) (**15 points**) Donnez une expression régulière qui représente L.
7. **(12 points)** Prouvez que pour tout langage L sur un alphabet Σ, $(L^*)^* = L^*$.

BONI (**5 points**) Dans le théorème de Myhill-Nerode sont définies deux relations sur Σ^*, R_M et R_L.

1. (**4 points**) Définissez l'une des deux.
2. (**1 point**) L'automate fini déterministe acceptant le langage donné L et tel que le nombre de ses états est minimum est défini en prenant comme l'ensemble d'états les classes d'équivalence de laquelle de ces deux relations?