NUMÉRO D'ANONYMAT : ....

MCAL – MT – Examen

Durée : 1h30, sans document

- N'oubliez pas d'indiquer votre numéro d'anonymat sur le sujet et glissez le dans votre copie à la fin de l'épreuve.
- Répondez sur votre copie sauf pour les questions avec pointillés.
- Commencez par lire tout le sujet pour repérer les questions faciles.
- Respectez les notations du cours.
- Le sujet est sur 20 points et comporte 5 exercices indépendants.

Le barème est donné à titre indicatif.

Tous les appareils électroniques sont interdits à l'exception des montres.

4 pt

Exercice 1 : Connaissez-vous les définitions du cours ? (10 min)

Complétez les pointillés.

0.5 pt

Q1. Deux MT M_1 et M_2 sont équivalentes si et seulement si ....

....

0.5 pt

Q2. Un langage L est décidable si et seulement si ....

....

0.5 pt

Q3. Un langage L est récursivement énumérable si et seulement si ....

....

0.5 pt

Q4. Un langage L est indécidable si et seulement si ....

....

0.5 pt

Q5. Le langage reconnu par une machine de Turing M est l'..

tels que l'exécution ....

....

0.5 pt

Q6. Si m est le codage binaire de la MT M, ω un mot binaire et U est la machine de Turing universelle alors $\ldots(m) = \ldots\ldots$ et $M(\ldots) = \ldots\ldots\ldots\ldots\ldots\ldots$

1 pt

Q7. Le langage reconnu par la machine universelle U est

$$\mathscr{L}(U) = \{\ \ldots\ldots\ldots\ldots \in \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \mid \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \}$$

C'est l'ensemble des couples $(\ldots, \ldots)$ formés .... tel que .

....

2 pt

Exercice 2 : Utilisation des machines de Turing (10 min)

0.25 pt **Q8.** Expliquez en français le fonctionnement d'une instruction « do M_{body} while M_{cond} ».

0.75 pt **Q9.** Dessinez la MT qui correspond à l'instruction do M_{body} while M_{cond}. Vous représenterez les machines de Turing M_{body} et M_{cond} par des nuages avec un état initial, un état accepteur et si nécessaire un état de rejet.

0.25 pt **Q10.** Donnez une MT qui reconnaît le langage $\{0,1\}^*$.

0.75 pt **Q11.** Donnez une MT déterministe qui reconnaît les écritures binaires (little endian = bits de poids faible à gauche) des entiers naturels sans 0 inutile.

Exemples :

- 3 est représenté en binaire par le mot $^{\infty}\Box \mid \$ \mid 1 \mid 1 \mid \Box^{\infty}$, **pas** par $^{\infty}\Box \mid \$ \mid 1 \mid 1 \mid 0 \mid \dots \mid 0 \mid \Box^{\infty}$
- En particulier l'entier 0 est représenté par le mot $^{\infty}\Box \mid \$ \mid 0 \mid \Box^{\infty}$

3.5 pt

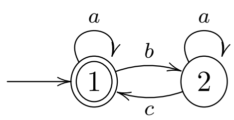
Exercice 3 : Langages réguliers et machines de Turing (20 min)

Justifiez soigneusement vos réponses.

.75 pt **Q12.** Expliquez comment traduire un automate (à nombre) d'états fini A en une machine de Turing M équivalente. Que signifie « équivalente » ?

0.75 pt **Q13.** Que faut-il faire en plus si on souhaite que la MT M décide le langage $\mathscr{L}(A)$? Que signifie « décide » ?

0.5 pt **Q14.** **Application :** On considère l'alphabet $\Sigma = \{a, b, c\}$. Dessinez la machine de Turing M équivalente à l'automate (à nombre) d'états fini suivant

$A =$ 

0.5 pt **Q15.** **Application (suite) :** Complétez (avec un crayon d'une autre couleur) la machine de Turing M de la question précédente afin qu'elle **décide** le langage $\mathscr{L}(A)$.

0.5 pt **Q16.** Existe-t'il des **langages réguliers** qui ne sont pas reconnaissables par les machines de Turing ?
Justifiez soigneusement votre réponse ; une réponse « oui/non » ne donne pas de point.

0.5 pt **Q17.** Existe-t'il des **langages** qui ne sont pas reconnaissables par les machines de Turing ?
Justifiez soigneusement votre réponse ; une réponse « oui/non » ne donne pas de point.

4 pt

Exercice 4 : Formalisation et Théorème de Rice (15 min)

On note $\mathcal{M}$ l'ensemble des codages binaires de machines de Turing.

0.25 pt **Q18.** **Complétez :** Un ensemble de machines de Turing est un ensemble de Rice s'il s'écrit $\{m \in \mathcal{M} \mid \mathcal{C}(\dots\dots\dots\dots\dots\dots)\}$ où la condition $\mathcal{C}$ porte sur le langage reconnu par m

0.25 pt

Q19. **Complétez :** Tout ensemble de Rice non-............. c'est-à-dire différent de ...... et de .. , est ....

1 pt

Q20. **Formalisez en termes mathématiques les ensemble suivants**

1. L'ensemble des machines de Turing qui n'acceptent pas le mot m correspondant à leur codage en binaire :
   $L_1 = \{ m \in \mathcal{M} \mid \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \}$
2. L'ensemble des machines de Turing dont l'exécution sans paramètre est infinie :
   $L_2 = \{ m \in \mathcal{M} \mid \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \}$
3. L'ensemble des machines de Turing qui acceptent tous les mots binaires :
   $L_3 = \{ m \in \mathcal{M} \mid \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \}$
4. L'ensemble des machines de Turing équivalentes à la MT $M_\emptyset$ qui reconnaît le langage vide :
   $L_4 = \{ m \in \mathcal{M} \mid \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \}$

1.5 pt

Q21. Parmi les ensembles L_1 à L_4, trois sont des ensembles de Rice. Lesquels ? Justifiez votre réponse en définissant le critère de Rice, $\mathcal{C}$, correspondant.

— ..... est un ensemble de Rice car ce langage correspond à $\{ m \in \mathcal{M} \mid \varepsilon \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \}$
qui est de la forme $\{ m \in \mathcal{M} \mid \mathcal{C}(\ldots\ldots\ldots\ldots\ldots\ldots) \}$ avec $\mathcal{C}(\ldots..) \stackrel{def}{=} \ldots\ldots\ldots\ldots$

— ..... est un ensemble de Rice car il peut s'écrire $\{ m \in \mathcal{M} \mid \mathcal{C}(\ldots\ldots\ldots\ldots\ldots\ldots) \}$
avec $\mathcal{C}(\ldots..) \stackrel{def}{=} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

— ..... est un ensemble de Rice car il peut s'écrire $\{ m \in \mathcal{M} \mid \mathcal{C}(\ldots\ldots\ldots\ldots\ldots\ldots) \}$
avec $\mathcal{C}(\ldots..) \stackrel{def}{=} \ldots\ldots\ldots\ldots$

1 pt

Q22. Expliquez pourquoi l'ensemble restant n'est pas un ensemble de Rice. Que peut-on en déduire ?

6.5 pt

Exercice 5 : Réduction du PCP au problème de terminaison d'un programme Gamma (40 min)

Le but de cet exercice est de démontrer que la terminaison d'un programme Gamma est un problème indécidable. Pour le prouver on va réduire le Problème de Correspondance de Post (PCP) – connu pour être indécidable – au problème de la terminaison de programmes Gamma.

3.5 pt

Problèmes de Correspondance de Post et exécutions de programmes Gamma (20 min)

On considére l'alphabet $\{ a, c, f, i, \ell \}$ et ε désigne le mot vide (*ie.* le "" des chaînes de caractères).

0.25 pt

Q23. **Complétez :** *Soit* D *un ensemble de domino* PCP(D) *e* .... *de domino* D *telle que* .... *obtenu par concaténation* .... *de* .... *au* .... *obtenu par concaténation* .... *de domino*

Exemple : Considérons l'ensemble de dominos $D_1 = \left\{ \underset{d_1}{\boxed{\dfrac{f}{fac}}}, \underset{d_2}{\boxed{\dfrac{i\ell}{i\ell}}}, \underset{d_3}{\boxed{\dfrac{ac}{\varepsilon}}} \right\}$. La séquence $d_1.d_3.d_2$ est une solution puisqu'on obtient .... en haut et .... en bas. La séquence réduite à ..... est aussi une solution puisqu'on obtient .... en haut et en bas.

Exécutions de programmes Gamma On considére les constructeurs Gamma suivants $\textsc{a}(.)$, $\textsc{i}(.)$, $\textsc{c}(.)$, $\textsc{f}(.)$, $\textsc{l}(.)$ qui prennent tous un argument, ainsi qu'un constructeur Gamma sans argument (*ie.* une constante) ε.
À l'aide des constantes et constructeurs précédents on peut représenter les mots écrits sur l'alphabet $\Sigma = \{a, c, f, i, \ell\}$. On désigne par $[\,\omega\,]_\Gamma$ la représentation en Gamma d'un mot $\omega \in \Sigma^*$.

0.25 pt

Q24. Donnez la représentation en Γ du mot *facil* : $[\,faci\ell\,]_\Gamma =$ ....

0.25 pt

Q25. Donnez l'exécution du programme Gamma Γ_{D_1} ci-dessous sur l'ensemble réduit à l'élément $\mathrm{G}(\,[\,faci\ell\,]_\Gamma\,,\,[\,faci\ell\,]_\Gamma\,)$

$$\Gamma_{D_1} \stackrel{def}{=} \begin{cases} \mathrm{G}(\,\textsc{f}(x)\,,\,\textsc{f}(\textsc{a}(\textsc{c}(y)))\,) & \xrightarrow[r_1]{} \mathrm{G}(\,x\,,\,y\,) \\ \mathrm{G}(\,\textsc{i}(\textsc{l}(x))\,,\,\textsc{i}(\textsc{l}(y))\,) & \xrightarrow[r_2]{} \mathrm{G}(\,x\,,\,y\,) \\ \mathrm{G}(\,\textsc{a}(\textsc{c}(x))\,,\,y\,) & \xrightarrow[r_3]{} \mathrm{G}(\,x\,,\,y\,) \end{cases}$$

0.25 pt

Q26. Donnez l'exécution du programme Γ_{D_1} sur l'ensemble réduit à l'élément $\mathrm{G}(\,[\,acfaci\ell\,]_\Gamma\,,\,[\,acfaci\ell\,]_\Gamma\,)$

0.5 pt

Q27. On souhaite ajouter une règle (r_0) au programme Γ_{D_1} afin qu'il **ne termine pas** lorsqu'on l'exécute sur $\mathrm{G}(\,[\,faci\ell\,]_\Gamma\,,\,[\,faci\ell\,]_\Gamma\,)$. Complétez la régle (r_0).

$$\mathrm{G}(\,\varepsilon\,,\,\varepsilon\,) \xrightarrow[r_0]{} \mathrm{G}(\ \ldots\ldots\ldots\ldots\ldots\ldots\ ,\ \ldots\ldots\ldots\ldots\ldots\ldots\ )$$

Généralisation Dans les questions précédentes on a raisonné sur *la solution « facil »* du $\textsc{pcp}(D_1)$; on veut maintenant généraliser le raisonnement à toute solution du $\textsc{pcp}(D_1)$.

1 pt

Q28. En vous inspirant fortement du programme Γ_{D_1} donnez un programme Γ'_{D_1} qui **ne termine pas** lorsqu'on l'exécute sur $\mathrm{G}'(\,\varepsilon\,,\,\varepsilon\,,\,[\,\omega\,]_\Gamma\,)$ quel que soit le mot ω correspondant à une solution du $\textsc{pcp}(D_1)$. Pour cela on utilise un constructeur $\mathrm{G}'(\,.\,,\,.\,,\,.\,)$ à trois arguments au lieu de $\mathrm{G}(\,.\,,\,.\,)$.

Programme Gamma associé à un PCP : application à un autre ensemble de dominos

0.5 pt

Q29. Donnez une séquence de dominos de l'ensemble D_2 ci-dessous qui soit une solution du $\textsc{pcp}(D_2)$ et donnez le mot correspondant à cette solution.

$$D_2 = \left\{ \underset{d_1}{\boxed{\dfrac{ia}{a}}}, \underset{d_2}{\boxed{\dfrac{acc}{ci}}}, \underset{d_3}{\boxed{\dfrac{f}{fac}}} \right\}$$

0.5 pt

Q30. Donnez un programme Γ'_{D_2} qui **ne termine pas** lorsqu'on l'exécute sur $\mathrm{G}'(\,\varepsilon\,,\,\varepsilon\,,\,[\,\omega\,]_\Gamma\,)$ où ω est le mot correspondant à une solution du problème $\textsc{pcp}(D_2)$.

3 pt

Indécidabilité de la terminaison de programmes Gamma (20 min)

— On note $\mathscr{D}$ l'ensemble de tous les ensembles de dominos possibles sur l'alphabet Σ. Ainsi, $D \in \mathscr{D}$ est un ensemble de dominos.

— PCPSAT désigne l'ensemble des ensembles D de dominos pour lesquels PCP(D) a une solution. Autrement dit, $D \in$ PCPSAT si et seulement si PCP(D) admet une solution.

— *Gamma* désigne l'ensemble de tous les programmes Gamma possibles.

— On note GTT l'ensemble de programmes $\Gamma \in$ *Gamma* dont l'exécution se termine quel que soit l'ensemble E sur lequel on l'exécute. Autrement dit, $\Gamma \in$ GTT signifie que Γ termine toujours.

$$\text{GTT} = \{\Gamma \in \mathcal{G}amma \mid \forall E,\ \Gamma(E) \not\to \infty\} \quad \textit{et} \quad \overline{\text{GTT}} = \mathcal{G}amma \setminus \text{GTT}.$$

0.25 pt

Q31. **Complétez :** $\Gamma \in \overline{\text{GTT}}$ signifie qu' .... ensemble E tel que $\Gamma(E)$........... autrement dit l'exécution de Γ sur E ....

0.25 pt

Q32. **Complétez le diagramme de réduction et les « on doit montrer »**

$$\begin{array}{rcl} D \in \mathscr{D} & \xrightarrow[\textit{traduction}]{M_R} & \Gamma'_D \in \mathcal{G}amma \\ \underbrace{D \in \text{PCPSAT}} & \underset{(\ddagger)}{\ldots\ldots\ldots} & \underbrace{\Gamma'_D \in \ldots\ldots\ldots\ldots\ldots} \\ \textit{indécidable} & \ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{array}$$

Pour montrer l'indécidabilité de la terminaison de programmes Gamma que doit-on faire ?

(a) On doit montrer qu'il .... tout ensemble de domino D en un programme Γ'_D

(b) On doit montrer .... ($\ddagger$).

0.5 pt

Q33. **(a) Complétez la définition de la traduction**

Un domino $\boxed{\frac{u}{v}} \in D$ peut être vu comme un .... que l'on peut inscrire sur la bande 1 d'une MT. Un ensemble D de domino re ....
.... Il e MT M_R à .... qui effectue la .... : elle parcourt le chaque domino d_i re (u, v), elle inscrit la règle Gamma .... $\xrightarrow[r_i]{}$ .... sur la bande 2.

On ajoute ensuite à la bande 2, la règle (r_0) ; ainsi la bande 2 contient le programme Γ'_D tel qu'on l'a défini dans la première partie de cet exercice.

0.5 pt

Q34. **(b) Complétez la preuve de ($\ddagger$)**

($\Rightarrow$) On doit montrer que si $D \in$ PCPSAT alors le programme Γ'_D construit par $M_R(D)$ .... Il suffit donc de trouver un ensemble E sur lequel l'exécution de Γ'_D ....
Puisque $D \in$ PCPSAT, le problème PCP(D) ....

.... ; notons ω le mot corre

montré dans la première partie de l'exercice que Γ'_D ....

si on l'exécute sur l'ensemble .... Il suffit donc de prendre cet ensemble pour E.

Q35. **(b) Complétez la preuve de (‡)**

($\Leftarrow$) Montrons la .... : On doit montrer que si ....

tel que $\Gamma'_D(E)$ .... alors il existe une solution au PCP(D) (et donc $D \in$ PCPSAT).

À l'exce (r$_0$), toute Γ'_D appliquée $G'(\ \omega_1, \omega_2, \omega\ )$ font

.... le mot ω_1 ou le mot ω_2 (ou le

d'appliquer ce .... fois. Ainsi, si $\Gamma_D(E)$ a une exécution infinie il e .... régulièrement

car c'e .... le ω_1 et ω_2. Une exécution infinie contient donc .... applications .... Elle aura

donc forcément la forme $\ldots \xrightarrow[r_0]{} G'(\ \ldots, \ldots, \ldots\ ) \xrightarrow[r_{i_1}]{} \ldots \xrightarrow[r_{i_k}]{} G'(\ \epsilon, \epsilon, \omega\ ) \xrightarrow[r_0]{} \ldots$

On constate sur cette portion d'exécution que le $r_{i_1}, \ldots, r_{i_k}$ consomment

.... le ω. Puisque chacune

.... domino cela signifie que ....

.... solution au PCP(D).

Q36. **Complétez la conclusion :** On a montré par réduction que la terminaison de programmes Gamma est indécidable.

SUPPOSONS LE CONTRAIRE ET MONTRONS UNE CONTRADICTION : Suppo

existe un algorithme capable de décider ....

.... alors on pourrait savoir si

un .... solution. Pour cela, étant donné

un .... il suffirait de construire le programme .... au

moyen de .... e

- si ....
  alors .... solution
- s'il existe ....
  alors .... solution

On serait donc capable de .... le

Po CONTRADICTION.