DM 3 (version finale) - à rendre avant le 31/01/2019

Vous traiterez au choix l'exercice 3 ou l'exercice 4.

Exercice 1.

1. Soit L un langage rationnel qui est reconnu par un automate avec n états. Montrer que L est infini si et seulement s'il existe $w \in L$ tel que $n \leq |w| < 2n$.
2. Un langage rationnel est dit *unitaire* s'il existe un automate fini déterministe avec exactement un état acceptant qui le reconnaît. Montrer que L est unitaire si et seulement si
$$\forall u, v, w \in \Sigma^* \quad u, uv, w \in L \implies wv \in L.$$
3. Soient N, M des automates finis complets avec n et m états respectivement. Montrer que $L(N) = L(M)$ si et seulement si $\forall w \in \Sigma^*$ tel que $|w| < nm$ alors $w \in L(N) \iff w \in L(M)$. Conclure qu'il est possible de décider algorithmiquement si deux automates reconnaissent le même langage.

Exercice 2.
Une machine de Turing est dite *linéairement bornée* si elle n'écrit pas en dehors de l'espace utilisé par la donnée.
Comme on peut toujours supposer qu'une machine de Turing n'écrit pas le symbole blanc $\sqcup$ (en le dupliquant éventuellement en un autre symbole $\sqcup_2$), on peut aussi dire qu'une machine linéairement bornée est une machine telle que si $\delta(p, \sqcup) \ni (q, b, x)$ est une transition alors $b = \sqcup$ et $x = \leftarrow$. Ceci empêche la machine de modifier les symboles blancs présents sur la bande.

1. Etant donné un mot w et une machine linéairement bornée M, peut-on décider si M accepte w ?
2. Etant donné une machine linéairement bornée M, peut-on décider si M n'accepte aucun mot, c'est-à-dire $L(M) = \emptyset$?
Indice : On pourra commencer par montrer que ce problème est indécidable pour les machines de Turing en général.

Exercice 3. *Choix 1*
Soit Σ un ensemble. Un arbre à branchement Σ est un ensemble $A \subseteq \Sigma^*$, possiblement infini, clos par préfixe. Lorsque $A \subseteq \{0,1\}^*$ est un arbre, on parle d'arbre à branchement binaire. Une branche infinie de A est une suite $(u_i)_{i \in \mathbb{N}} \in \Sigma^{\mathbb{N}}$ telle que pour tout $i \in \mathbb{N}$, les éléments de la suite $flat(u) \in \Sigma^{\mathbb{N}}$ définie par $(flat(u))_i = u_0 \dots u_{i-1}$ sont dans A. Notons $B_\infty(A)$ l'ensemble des branches infinies de A.

1. Soit $P : \mathcal{P}(\Sigma^{\mathbb{N}}) \to \mathcal{P}(\Sigma^*)$ la fonction qui, à un ensemble de branches infinies, associe l'arbre de leurs préfixes. P est définie par
$$P(X) = \{u \in \Sigma^* \mid \exists v \in X \; \forall k < |u| \; u_k = v_k\}$$
Montrer que pour tout arbre A, $P(B_\infty(A)) \subseteq A$ et, pour tout $X \in \mathcal{P}(\Sigma^{\mathbb{N}})$, $B_\infty(P(X)) \supseteq X$. Ces inclusions sont-elles toujours des égalités ?

2. Démontrer le lemme de König faible : si Σ est un ensemble fini, tout arbre infini $A \subseteq \Sigma^*$ admet une branche infinie $u \in B_\infty(A)$.

L'objet de cet exercice est de montrer que cet énoncé ne se relativise pas au monde des arbres récursifs. Pour se faire, on exhibe un arbre binaire $K \subseteq \{0,1\}$ appelé *arbre de Kleene* ayant les propriétés suivantes :

1. K est infini.
2. K est récursif, i.e., il existe une fonction récursive totale $\chi_K : \{0,1\}^* \to \{0,1\}$ telle que $\chi_K(p) = 1$ ssi $p \in K$.
3. K n'admet pas de branche infinie récursive.

Avant d'attaquer cette définition, on regarde quelques cas où l'on peut extraire des branches infinies de manière effective.

3. Soit $A \subseteq \{0,1\}^*$ un arbre récursif, i.e., un arbre tel qu'il existe une fonction récursive totale $\chi_A : \{0,1\}^* \to \{0,1\}$ telle que $\chi_A(p) = 1$ ssi $p \in A$. Montrer que si $A = P(X)$ pour un certain $X \subseteq \Sigma^{\mathbb{N}}$ non vide, alors A possède une branche infinie récursive, i.e., il existe une fonction récursive totale $\phi : \mathbb{N} \to \{0,1\}$ telle que $\phi \in B_\infty(A)$.

Une suite $(u_i)_{i\in\mathbb{N}} \in \Sigma^{\mathbb{N}}$ est *ultimement périodique* si il existe $m, p \in \mathbb{N}$ tels que, pour tout $n \geq m$, $u_n = u_{n+p}$.

4. Soit $A \subseteq \Sigma^*$ un arbre infini. En utilisant le lemme d'Ogden, montrer que si A est également un langage algébrique, alors A a une branche ultimement périodique.

Appelons $T : \mathbb{N}^3 \to \mathbb{N}$ la fonction qui sur l'entrée (e, m, n) renvoie

- 0 si la machine de Turing de code e ne termine pas en moins de n étapes sur l'entrée m
- $k+1$ si la machine de Turing de code e termine sur l'entrée m en renvoyant k en moins de n étapes.

On admet que T est récursive primitive.

Grâce à ce prédicat T, on définit K comme suit

$$K = \{v \in \{0,1\}^* \mid \forall e < |v| \; (v_e = 0 \Rightarrow T(e,0,|v|) \leq 1) \wedge (T(e,0,|v|) = 1 \Rightarrow v_e = 0)\}$$

5. Vérifier que K est un arbre récursif.

Si $e \in \mathbb{N}$, on appelle $\varphi_e : \mathbb{N} \rightharpoonup \mathbb{N}$ la fonction partielle calculée par la machine de Turing de code e. Notez qu'en particulier, $x \notin dom(\varphi_e)$ ssi $\forall n \in \mathbb{N}\; T(e,x,n) = 0$ et $\varphi_e(x) = k$ ssi $\exists n \in \mathbb{N}\; T(e,x,n) = k+1$.

6. Montrer que $B_\infty(K) \neq \emptyset$. En déduire que K est infini.

7. Montrer que toute branche $u \in B_\infty(K)$ n'est pas récursive.

Indication : utiliser le second théorème de récursion de Kleene

L'espace de Cantor $\mathcal{C}$ est l'espace des suites binaires $\{0,1\}^{\mathbb{N}}$ et l'espace de Baire $\mathcal{N}$ celui des suites d'entiers $\mathbb{N}^{\mathbb{N}}$. Un résultat classique est qu'il existe des bijections $\mathcal{C} \simeq \mathcal{N}$. Dans la suite de l'énoncé, on établit une version récursive de ce résultat. On pose

$$\mathcal{C}_{eff} = \{p \in \mathcal{C} \mid \exists e \in \mathbb{N}\; \varphi_e = p\} \qquad \text{et} \qquad \mathcal{N}_{eff} = \{p \in \mathcal{N} \mid \exists e \in \mathbb{N}\; \varphi_e = p\}$$

Notre but dans la vie sera désormais de montrer l'existence

— de fonctions $g : \mathcal{N}_{eff} \to \mathcal{C}_{eff}$ et $g^{-1} : \mathcal{C}_{eff} \to \mathcal{N}_{eff}$ telles que $g^{-1} \circ g = id$ et $g \circ g^{-1} = id$.

— de codes $e_{g^{-1}}, e_g \in \mathbb{N}$ tels que pour tout $e \in \mathbb{N}$, $g(\varphi_e) = \varphi_{e_g}(e)$ dès que $g(\varphi_e)$ est définie et, similairement, $g^{-1}(\varphi_e) = \varphi_{e_{g^{-1}}}(e)$ dès que $g^{-1}(\varphi_e)$ est définie.

On va utiliser l'arbre de Kleene K pour construire cette bijection. À noter que la définition précise de K n'est pas nécessaire pour la suite ; on pourrait utiliser n'importe quel arbre satisfiant les propriétés 1, 2 et 3 pour obtenir le même résultat.

8. Soit $O \subseteq \{0,1\}^*$ l'ensemble des chemins minimaux sortant de K. Plus formellement, on pose $O = \{pa \notin K \mid p \in K, a \in \{0,1\}\}$. Montrer que O est récursif et donc qu'il existe une bijection récursive $h : O \to \mathbb{N}$.
Indication : on pourra commencer par construire une suite récursive $(\{0,1\}^*)^{\mathbb{N}}$ *correspondant à un parcours en largeur de l'arbre binaire complet* $\{0,1\}^*$.

9. Soit $flat : (\Sigma^+)^{\mathbb{N}} \to \Sigma^{\mathbb{N}}$ la fonction de concaténation infinie mettant bout à bout tous les mots de sa suite en entrée (formellement, on a $flat(u)_{x+\sum_{i=0}^{k-1}|u_i|} = (u_k)_x$ pour tout $k \in \mathbb{N}$ et $x < |u_k|$). Montrer que la composée

$$\mathcal{N} = \mathbb{N}^{\mathbb{N}} \xrightarrow{h\circ -} O^{\mathbb{N}} \subseteq (\{0,1\}^*)^{\mathbb{N}} \xrightarrow{flat} \mathcal{C}$$

bien qu'elle-même pas surjective, se restreint à une bijection $g : \mathcal{N}_{eff} \to \mathcal{C}_{eff}$. Montrer que g et sa réciproque sont récursives.
Indication : si vous êtes bloqués, vous pouvez examiner au brouillon une situation analogue où l'on prend un arbre binaire fini à la place de K et où $flat$ *se restreint à une bijection* $O^{\mathbb{N}} \to \mathcal{C}$. *Vous pourrez par la suite remarquer que* $f \in \mathcal{C}$ *est récursive si et seulement si, pour tout* $k \in \mathbb{N}$, *la fonction* $n \mapsto f(k+n)$ *est récursive.*

Exercice 4. *Choix 2*

Certains problèmes sont indécidables.. Mais à quel point ? Le but de cet exercice est d'introduire la notion de **Hiérarchie Arithmétique**, qui est une classification des problèmes en fonction de leur *degré d'indécidabilité*.
On commence par expliciter la notion d'oracle et de Turing-réduction.

Définition (Oracle)**.** *Un oracle pour un langage B est une machine hypothétique capable de décider pour toute chaîne w si* $w \in B$. *Une Machine de Turing avec oracle est une machine modifiée qui peut effecteur des transitions en faisant appel à un oracle. On notera* M^B *une machine de Turing ayant accès à un oracle pour le langage B.*

Définition (Turing-reduction)**.** *Un langage A est Turing-réductible à un langage B s'il est décidable par une machine de Turing utilisant un oracle pour B. On note alors* $A \leq_T B$.

Dans tout cet exercice, un *problème* désigne l'ensemble des entrée pour lesquelles une propriété est vraie. Par exemple, le problème de l'arrêt se définit informellement comme "La machine M s'arrête-t-elle ?". Formellement, en supposant qu'on appelle M_σ la machine de code $\sigma \in \Sigma^*$ pour un schéma de codage raisonnable :

$$\text{HALT} = \{\sigma \in \Sigma^* \mid M_\sigma \text{ s'arrête (sur l'entrée vide)}\}$$

1. (échauffement) Montrer que si $A \leq_T B$ et B décidable, alors A est décidable.
Les oracles sont des machines qui peuvent apporter beaucoup de puissance, et on les utilise pour définir la hiérarchie arithmétique.

Définition (Hiérarchie Arithmétique)**.** *On note $\Delta_0 = \Sigma_0 = \Pi_0$ la classe des problèmes récursifs (décidables). On définit alors :*

— *Σ_n la classe des problèmes récursivement énumérables avec un oracle d'un ensemble de Π_{n-1}.*

— *Π_n la classe complémentaire de Σ_n, ou encore la classe des problèmes co-récursivement énumérables (coRE) avec un oracle d'un ensemble de Π_{n-1}.*

— *$\Delta_n = \Sigma_n \cap \Pi_n$ la classe des problèmes récursifs avec un oracle d'un ensemble de Π_{n-1}.*

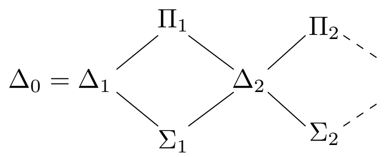

FIGURE 1 – Hiérarchie arithmétique

Un problème B est dit Σ_n**-dur** (resp. Π_n-dur) si pour tout problème $A \in \Sigma_n$, $A \leq_T B$. Si de plus $B \in \Sigma_n$, alors il est dit Σ_n**-complet**.

2. À quoi correspondent les classes Σ_1 et Π_1 ?

3. Prouver que $HALT$ est Σ_1-complet. En déduire pour chaque $n \in \mathbb{N}$ un problème Σ_n-complet.

4. Où se trouve $\{\sigma \in \Sigma^* \mid \forall x, M_\sigma(x) \text{ termine}\}$ dans la hiérarchie arithmétique ?

5. Prouver que si $A \in \Delta_2$, alors sa fonction caractéristique est limite d'une fonction récursive g, c'est à dire que

$$\chi_A(x) = \lim_{s \to \infty} g(x, s)$$

On suppose maintenant A quelconque tel que sa fonction caractéristique est limite d'une fonction récursive g.

6. Expliciter une formule logique de la forme $\exists s \forall t, \phi(s,t,x)$ équivalente à "$x \in A$", avec $\phi(s,t,x)$ sans quantificateurs.

7. En déduire que $A \in \Sigma_2$.

8. Avec la même méthode, prouver que $\overline{A} \in \Sigma_2$. En déduire le théorème suivant :

Théorème (Limit Lemma, Shoenfield (1959))**.** *$A \in \Delta_2$ si et seulement si sa fonction caractéristique est limite d'une fonction récursive.*

Ce théorème est généralisable à n'importe quel niveau de la hiérarchie arithmétique. Il permet d'expliciter la relation entre les niveaux : chaque niveau peut être vu comme la limite du niveau inférieur. En effet, dans le cas du théorème, une fonction récursive est capable de décider des ensembles de Σ_1 ou Π_1, et lorsque l'on en prend la limite on peut décider les ensembles du niveau supérieur : Σ_2 et Π_2.