TD 01 – Rappels de calculabilité

Exercice 1. *Diagonale de Cantor*

On note $[0,1]$ l'ensemble des réels entre 0 et 1. Remarque : la représentation de certains réels n'est pas unique, mais cela n'a pas d'importance pour cet exercice.

1. Démontrer que $|\mathbb{N}| < |[0,1]|$.
2. En déduire qu'il existe des fonctions de $\mathbb{N}$ dans $\mathbb{N}$ qui ne sont calculées par aucun programme (écrit dans votre langage de programmation favori).
3. Existe-t-il un ensemble X tel que $|\mathbb{N}| < |X| < |[0,1]|$?

Rappel : soient A et B deux ensembles, une fonction $f : A \to B$ est

— injective ssi $\forall\, a, a' \in A : f(a) = f(a') \implies a = a'$ (ou la contraposée),

— surjective ssi $\forall\, b \in B : \exists\, a \in A : b = f(a)$,

— bijective ssi elle est à la fois injective et surjective.

Utile : Théorème de Cantor-Schröder-Bernstein : soient A et B deux ensembles, si il existe une fonction injective de A vers B (intuitivement $|A| \leq |B|$), et une fonction injective de B vers A (intuitivement $|B| \leq |A|$), alors il existe une bijection entre A et B (intuitivement $|A| = |B|$).

Exercice 2. *Ensembles infinis dénombrables*

1. Montrer que $\mathbb{N} = \{0, 1, 2, \dots\}$ et $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ sont en bijection.
2. Montrer que $\mathbb{N}$ et $\mathbb{N} \times \mathbb{N}$ sont en bijection.
3. Peut-on en déduire que $|\mathbb{Q}| = |\mathbb{N}|$?
4. Montrer que $\mathbb{N}$ et $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ sont en bijection.
5. Montrer que Σ^* et $\mathbb{N}$ sont en bijection, pour Σ un alphabet fini.

Exercice 3. *Ensembles infinis indénombrables*

1. Montrer que $\mathcal{P}(\mathbb{N})$ (l'ensemble des parties de $\mathbb{N}$) et $[0,1]$ sont en bijection.
2. Montrer que l'ensemble des langages sur un alphabet fini Σ, et $[0,1]$, sont en bijection.
3. Montrer que $[0,1]$ et $\mathbb{R}$ sont en bijection.

Exercice 4. *Hiérarchie de Chomsky ♡*

Extrait de `https://en.wikipedia.org/wiki/Chomsky_hierarchy` (01/2022)

Grammar	Languages	Automaton	Production rules*
Type-0	Recursively enumerable	Turing machine	$\gamma \to \alpha$ (no constraints)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \to \alpha \gamma \beta$
Type-2	Context-free	Non-det. pushdown automaton	$A \to \alpha$
Type-3	Regular	Finite state automaton	$A \to \mathrm{a}$ et $A \to \mathrm{a}B$

* Meaning of symbols :

— a = terminal, A, B = non-terminal,
— α, β, γ = string of terminals and/or non-terminals, α, β = maybe empty, γ = never empty.

Note that the set of grammars corresponding to recursive languages is not a member of this hierarchy ; these would be properly between Type-0 and Type-1.

Every regular language is context-free, every context-free language is context-sensitive, every context-sensitive language is recursive and every recursive language is recursively enumerable. These are all proper inclusions, meaning that there exist recursively enumerable languages that are not context-sensitive, context-sensitive languages that are not context-free and context-free languages that are not regular.

En français, on dit aussi langage *régulier* ou *rationnel* pour type-3, et *algébrique* pour type-2.

✎ Donner des langages qui distinguent chacun de ces types. (Énoncé du lemme de l'étoile pour les langages réguliers ? Du lemme d'itération pour les langages algébriques ?)

Exercice 5. *MT*

Donner des machines de Turing pour décider les langages suivants.

1. $L = \{w \in \Sigma^* \mid |w| \equiv 3 \mod 4\}$ avec $\Sigma = \{a\}$.
2. $L = \{a^i b^j c^k \mid i = j$ ou $j = k\}$ avec $\Sigma = \{a, b, c\}$ (*inherently ambiguous*).
3. $L = \{ww \mid w \in \Sigma^*\}$ avec $\Sigma = \{a\}$.
4. $L = \{ww \mid w \in \Sigma^*\}$ avec $\Sigma = \{a, b, c\}$.

Exercice 6. *Propriétés de clôture ?*

Démontrer ou réfuter chacune des propriétés de clôture suivantes.
Indication : 6 sont correctes, et 3 sont incorrectes.

1. Un langage $L \subseteq \Sigma^*$ est décidable si et seulement si L et $\Sigma^* \setminus L$ sont semi-décidables.
2. La famille des langages décidables est close par intersection et union.
3. La famille des langages semi-décidables est close par intersection et union.
4. La famille des langages non décidables est close par intersection et union.
5. La famille des langages non semi-décidables est close par intersection et union.
6. La famille des langages décidables est close par complémentation.
7. La famille des langages semi-décidables est close par complémentation.
8. La famille des langages non décidables est close par complémentation.
9. La famille des langages non semi-décidables est close par complémentation.

Exercice 7. *L'arrêt*

Indiquer si chacun des énoncés qui suit est vrai ou faux, en justifiant.

1. $\nexists M_{halt}, \forall M, \forall w : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$.
2. $\forall M, \forall w, \nexists M_{halt} : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$.

Exercice 8. *Espace et temps*

Supposons qu'une machine de Turing s'arrête au bout de t étapes de calcul en visitant (possiblement plusieurs fois chacune) s cases du ruban. Quelle(s) relation(s) existe(nt) entre t et s ?

Exercice 9. *MT : multi-ruban*

Objectif : montrer que le modèle des machines de Turing à plusieurs rubans et plusieurs têtes de lecture (une tête de lecture indépendante par ruban, et un seul état pour toute la machine) est équivalent au modèle des machines de Turing. Le multi-ruban est très pratique !

En fonction de l'état et du symbole lu sur chacun des rubans, la machine peut

— changer d'état,

— écrire un symbole sur chaque ruban,

— déplacer chaque tête vers la droite ou la gauche indépendamment les unes des autres.

1. Donner le type de la fonction de transition δ, et donner un exemple de transition.

Dans l'état initial, l'entrée est écrite sur le premier ruban et tous les autres rubans sont vides. Un mot est accepté si et seulement si la machine entre dans l'état final q_F au cours du calcul.

2. Comment simuler une MT *multi-ruban* avec une MT *mono-ruban* ?

Exercice 10. *MT : non-déterministes*

Objectif : montrer que le modèle des machines de Turing non-déterministes est équivalent au modèle des machines de Turing.

Une machine de Turing non-déterministe peut, à une étape de temps donnée (c'est à dire dans un état et pour un symbole lu), avoir plusieurs transitions possibles. Un calcul d'une machine de Turing est une succession de choix parmi l'ensemble des transitions possibles. Un mot est accepté si et seulement si il existe au moins un calcul qui mène à l'état final.

1. Donner le type de la fonction de transition δ, et donner un exemple de transition.

2. Donner une machine de Turing non déterministe qui reconnaisse le langage $L = \{ww \mid w \in \{a,b\}^*\}$.

3. Soit $r = \max\{|\delta(q,x)| \mid q \in Q \text{ et } x \in \Gamma\}$. Que représente r ?

4. Que pourrait représenter une suite finie de lettres sur l'alphabet $R = \{1,\ldots,r\}$?

5. Peut-on construire une machine de Turing qui énumère (c'est-à-dire écrit un à un et se place dans un état particulier q_e à chaque fois que son ruban contient un mot à énumérer) tous les mots finis sur l'alphabet R ?

6. Montrer que tout langage reconnu par une MT non-déterministe est reconnu par une MT déterministe (indication : on pourra utiliser trois rubans, le premier pour l'entrée, le second pour énumérer les choix non-déterministes, le troisième pour simuler une exécution particulière de la machine non-déterministe).

Exercice 11. *Utilité des réductions*

Démontrer les quatre énoncés suivants.

1. Si $A \leq_m^T B$ et B est décidable alors A est décidable.

2. Si $A \leq_m^T B$ et B est semi-décidable alors A est semi-décidable.

3. Si $A \leq_m^T B$ et A n'est pas décidable alors B n'est pas décidable.

4. Si $A \leq_m^T B$ et A n'est semi-décidable alors B n'est pas semi-décidable.

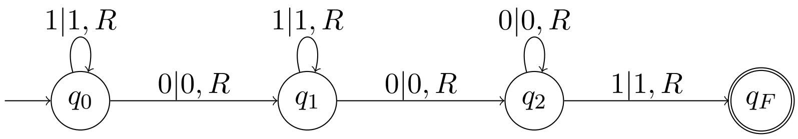
Exercice 12. *Réductions Turing many-one*

Rappel : $L_{\overline{halt\epsilon}} = \{\langle M\rangle \mid M$ ne s'arrête pas quand on la lance sur ϵ $\}$ n'est pas semi-décidable.

1. Déduire du rappel sur $L_{\overline{halt\epsilon}}$ que $\{\langle M\rangle \mid aab \notin L(M)\}$ n'est pas semi-décidable.
2. Déduire du rappel sur $L_{\overline{halt\epsilon}}$ que $L_u = \{\langle M\rangle\#w \mid w \in L(M)\}$ n'est pas décidable.
3. Pour tout langage L, réduire L à $aL = \{aw \mid w \in L\}$.
4. Pour tout langage L, réduire aL à L.
5. Réduire $L_{stupide} = \{a\}$ à L_u.
6. Déduire du rappel sur $L_{\overline{halt\epsilon}}$ que $L_\infty = \{\langle M\rangle \mid M(w) \uparrow$ pour toute entrée $w\}$ n'est pas semi-décidable ($M(w)\uparrow$ signifie que la calcul de M sur l'entrée w ne s'arrête pas).
7. Le complémentaire de L_∞ est-il semi-décidable ? Justifier.

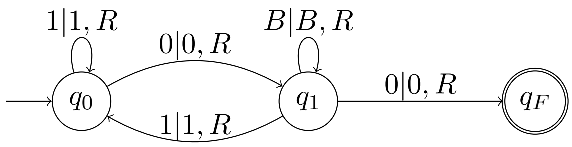
Exercice 13. *Complémentaire*

Soit la machine de Turing M_1 suivante sur l'alphabet d'entrée $\Sigma = \{0, 1\}$.



1. Cette machine de Turing s'arrête-elle sur toute entrée ?
2. Construire une machine de Turing M_1' telles que $L(M_1') = \Sigma^* \setminus L(M_1)$.

Soit la machine de Turing M_2 suivante sur l'alphabet d'entrée $\Sigma = \{0, 1\}$.



3. Construire une machine de Turing M_2' telles que $L(M_2') = \Sigma^* \setminus L(M_2)$.
4. Plus généralement, que penser d'une procédure pour effectuer cette transformation ? (de M à M' telle que $L(M') = \Sigma^* \setminus L(M)$)

Exercice 14. *Lancer deux MT en parallèle*

Soient deux machines mono-ruban $M_A = (Q_A, \Gamma_A, \Sigma_A, q_0^A, B_A, q_F^A, \delta_A)$,
et $M_B = (Q_B, \Gamma_B, \Sigma_B, q_0^B, B_B, q_F^B, \delta_B)$.

1. Définir une machine M à deux rubans, telle que $L(M) = L(M_A) \cup L(M_B)$.
2. Dans le cas où $\Sigma_A = \Sigma_B$ et $L(M_A) = \Sigma_A^* \setminus L(M_B)$, comment adapter la construction pour créer une machine de Turing M' à deux rubans qui décide le langage $L(M_A)$?

Exercice 15. *Semi-décidable mais pas décidable*

✎ Donner un exemple de langage semi-décidable, mais pas décidable (justifier).