TD 02 – Busy Beaver

Exercice 1. *Le concours du castor affairé*

Cet exercice est basé sur l'article de Tibor Radó, « On Non-Computable Functions », *Bell Systems Technology Journal*, vol. 41, no 3, mai 1962, p. 877-884.

Concours du castor affairé

Considérons des machines de Turing sur l'alphabet binaire $\Gamma = \{1, B\}$ dont la fonction de transition est $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$. Ces machines n'ont pas d'état final, elles s'arrêtent uniquement lorsqu'une transition est indéfinie. Dans cet exercice, les nombres seront représentés en unaire sur le ruban.

Voici les règles du concours du castor affairé.

- **(a)** Le participant sélectionne un entier $n \in \mathbb{N}$, et construit sa propre machine de Turing binaire M à n états.
- **(b)** Il lance sa machine M dans son état initial sur un ruban vide (uniquement des symboles B), et annonce qu'elle s'arrête après s étapes.
- **(c)** Il soumet son entrée (M, s) à un membre du *International Busy Beaver Club*.
- **(d)** Le membre vérifie que la machine M s'arrête après exactement s étapes. Cette vérification est décidable, il suffit de lancer la machine pour s étapes : si la machine ne s'est pas arrêtée après s étapes alors elle est rejetée, et si elle s'arrête en moins de s étapes alors elle est retournée au participant pour correction. Après qu'une machine ait été vérifiée, son score est le nombre de 1 écrits sur le ruban lorsqu'elle s'arrête.

Le champion BB-n est le participant qui obtient le plus grand score avec une machine à n états.

1. Quel score a obtenu le champion BB-2 ?
2. Sauriez-vous obtenir le score 6 dans la catégorie BB-3 ?
3. Comment prouver que 6 est le meilleur score possible de la catégorie BB-3 ?

Le problème du castor affairé consiste à déterminer le plus grand score possible pour BB-n. Soient

- $N(n)$ le nombre de machines de Turing à n états,
- E_n l'ensemble des entrées (M, s) valides pour BB-n,
- $N_e(n)$ la taille de E_n.

4. Que vaut $N(n)$?
5. Argumenter que $0 < N_e(n) < N(n)$.
6. Etant donnée une entrée (M, s), peut-on décider si $(M, s) \in E_n$?

L'ensemble E_n possède une définition claire et précise, est non-vide et fini, pour tout n. Nous verrons à la fin de cet exercice que pourtant, $N_e(n)$, le nombre d'éléments de E_n, n'est pas une fonction calculable.

Croissance de $\Sigma(n)$

Chaque entrée valide (M, s) possède un score bien défini $\sigma(M, s)$. Soit

$$\Sigma(n) = \max \left\{ \sigma(M, s) \mid (M, s) \in E_n \right\}.$$

$\Sigma(n)$ est la fonction du castor affairé, et nous allons voir qu'elle n'est pas calculable. Attention cependant, il est tout a fait possible d'en calculer certaines valeurs, comme $\Sigma(2) = 4$.

7. Regarder sur wikipedia les valeurs de $\Sigma(4)$, $\Sigma(5)$, $\Sigma(6)$.

Soient $f(x)$ et $g(x)$ deux fonctions de $\mathbb{N}$ dans $\mathbb{N}$, nous écrirons

$$f(x) \succ g(x) \text{ si et seulement si } \exists x_0 : \forall x > x_0 : f(x) > g(x).$$

Nous allons maintenant démontrer le théorème suivant :

Théorème. Pour toute fonction calculable $f : \mathbb{N} \to \mathbb{N}$, on a $\Sigma(n) \succ f(n)$.
Donc $\Sigma(n)$ n'est pas calculable.

Démonstration. Soit $f(x)$ une fonction calculable, et soit la fonction $F(x) = \sum_{i=0}^{x} \big(f(i) + i^2\big)$.

8. Argumenter que $F(x)$ est calculable par une machine de Turing.

Soit M_F une telle machine et c son nombre d'états. On considère que M_F, placée sur le premier symbole blanc à droite d'une entrée x (en unaire), s'arrête dans la configuration dont la description instantanée est $xBF(x)qB$ pour un certain état q.

9. Est-ce que $F(x) \geq f(x)$ pour tout x ?

10. Est-ce que $F(x) \geq x^2$ pour tout x ?

11. Est-ce que $F(x+1) > F(x)$ pour tout x ?

12. Donner la fonction de transition d'une machine de Turing M_x à $x+1$ états qui, à partir d'une entrée vide, écrit x symboles 1 consécutifs sur le ruban de façon à être composée avec M_F.

Soit la machine M_{FFx} qui correspond à la composition $M_F \circ M_F \circ M_x$.

13. Combien d'états possède la machine M_{FFx} ?

14. Que peut-on en déduire sur Σ(réponse à la question précédente) ?

15. Démonter que $F(F(x)) \succ F(1 + x + 2c)$.

16. En déduire que $\Sigma(1 + x + 2c) \succ F(1 + x + 2c)$.

17. Conclure la preuve.

<u>Fonction $S(n)$</u>
Remarquons que E_n coïncide avec l'ensemble des machines de Turing à n états qui s'arrêtent sur l'entrée vide. Soit $S(n) = \max\big\{s \mid (M, s) \in E_n\big\}$.

18. La fonction $S(n)$ est-elle calculable ?

<u>Fonction $N_e(n)$</u>
Rappelons que $N_e(n) = |E_n|$. Soit $N(s, n) = \big|\big\{(M, s') \in E_n \mid s' = s\big\}\big|$.

19. La fonction $N(s, n)$ est-elle calculable ?

Soient $G(s, n) = \sum_{i=1}^{s} N(i, n)$ et $\Phi(s, n) = N_e(n) - G(s, n)$.

20. Exprimer avec des mots la fonction $G(n, s)$.

21. Que peut-on dire de $\Phi(s, n)$?

22. Quelle est la valeur minimale de s telle que $\Phi(s, n) = 0$?

23. En déduire que $N_e(n)$ n'est pas calculable.

Remarque
Supposons que pour un entier n_0 nous arrivons à déterminer $N_e(n_0)$.

24. Peut-on alors déterminer $S(n_0)$ et $\Sigma(n_0)$?

25. Montrer que $S(n) \leq \Sigma(20n)$ (ou un peu plus que 20 si vous voulez).

Exercice 2. *Busy beaver et conjectures mathématiques*

La conjecture de Goldbach, formulée en 1742 et toujours ouverte, énonce que tout nombre entier pair supérieur à 3 peut s'écrire comme la somme de deux nombres premiers.

1. Donner un programme qui s'arrête si et seulement si la conjecture de Goldbach est fausse (ce programme n'a pas d'entrée).

On peut déduire de votre réponse précédente que si l'on savait décider le problème de l'arrêt, alors on saurait décider si la conjecture de Goldbach est vraie ou fausse.

2. Peut-on néanmoins semi-décider si cette conjecture est vraie ? ou bien si elle est fausse ?

3. Expliquer la phrase suivante :
« $BB(27)$ **is at least as hard as Goldbach conjecture.** »
Source : `https://bbchallenge.org/story#what-is-known-about-bb` (2024-02-22).

4. Expliquer la phrase suivante :
« $BB(7\,910)$ **is independent of ZFC.** »
Source : `https://bbchallenge.org/story#what-is-known-about-bb` (2024-02-22).

Pour aller plus loin :
- « A Relatively Small Turing Machine Whose Behavior Is Independent of Set Theory », Yedidia, Aaronson, 2016
(`https://arxiv.org/pdf/1605.04343.pdf`)
- « The Busy Beaver Frontier », Aaronson, 2020
(`https://www.scottaaronson.com/papers/bb.pdf`)

La suite de Collatz (ou suite de Syracuse) d'un entier $n > 0$ est la suite infinie $(f^i(n))_{i \in \mathbb{N}}$ avec

$$f: \mathbb{N} \to \mathbb{N}$$
$$n \mapsto \begin{cases} n/2 & \text{si } n \text{ est pair} \\ 3n+1 & \text{si } n \text{ est impair.} \end{cases}$$

La conjecture de Collatz, formulée en 1937 et toujours ouverte, énonce que $\forall\, n > 0 : \exists\, i \geq 0 : f^i(n) = 1$, c'est-à-dire que la suite de Collatz de tout entier converge vers la boucle $1, 4, 2$.

5. Proposer un algorithme liant le problème de l'arrêt et la conjecture de Collatz.