TD de *Calculabilité*
DIU *Enseigner l'informatique au lycée*, bloc 5.3.2
juin 2020
Natacha Portier et Olivier Laurent

Les exercices sont répartis en trois catégories. Les exercices **de base** sont ceux qui vous permettent de savoir si vous avez compris le cours. Les exercices **discussion** sont là pour vous permettre de prendre du recul et d'échanger sur le sujet. Les exercices **avancés** demandent un peu plus d'agilité technique, ils sont là pour que personne ne s'ennuie !

Introduction (video 1)

Exercice 1 (discussion) : Connaissez vous d'autres modèles de calcul ? Sont-ils équivalents à ceux cités dans la video ? Vous pouvez les partager dans mattermost et mettre un lien vers une ressource (cours en ligne, wikipedia...)

L'argument diagonal (video 2)

Exercice 2 (de base) : codage des suites d'entiers

On considère les deux fonctions suivantes de $\mathbb{N}^2 \to \mathbb{N}$.

$\alpha_2(n,m) = (n+m)(n+m+1)/2 + m$

$\beta(n,m) = 2^n(2m+1) - 1$

Expliquer sans faire de preuve formelle pourquoi ce sont des bijections. Pour vous aider pour α_2, vous pouvez sur un plan quadrillé numéroter les couples (n,m) avec leur image $f(n,m)$.

(On note usuellement $\langle x, y\rangle$ pour $\alpha_2(x,y)$.)

Donner une bijection α_3 de $\mathbb{N}^3$ dans $\mathbb{N}$, α_4 de $\mathbb{N}^4$ dans $\mathbb{N}$...

Donner une bijection α de $\mathbb{N}^\omega$ (l'ensemble des suites finies d'entiers) dans $\mathbb{N}$.

Exercice 3 (avancé) :

Montrer que pour tout ensemble E, il n'existe pas de bijection entre E et $\mathcal{P}(E)$.

Exercice 4 (très très avancé) :

Donnez une bijection entre $\mathbb{R}$ et $\mathcal{P}(\mathbb{N})$.

Exercice 5 (discussion) :

L'hypothèse du continu est-elle vraie d'après vous ? Cette question a t-elle un sens ? Est-ce que les entiers existent ? Commaissez-vous l'hypothèse du continu généralisée ?

Exercice 6 (discussion) :

Des parents disent à leur rejeton : si tu mens, tu seras privé de sortie avec tes amis et passeras la soirée à faire la vaiselle. Si tu dis la vérité tu seras privé de sortie et passeras la soirée à passer l'aspirateur. Pour pouvoir sortir, que peut répondre l'ado à la question "Que vas-tu faire ce soir" ?

La machine de Turing (video 3)

Exercice 7 (de base) :

Décrire les machines de Turing qui font les choses suivantes :

1. reconnaitre le langage $L = \{0^n1^m \mid (n,m) \in \mathbb{N}^2\}$
2. écrire 0 1 0 1 0 1 0 ... sur un ruban blanc.
3. multiplier par 2 son entrée qui est un mot binaire.

Exercice 8 (avancé) :

Trouver un codage binaire pour les machines de Turing

Exercice 9 (avancé) :

Décrire les machines de Turing qui font les choses suivantes :

1. ajouter 1 à son entrée binaire.
2. coder en binaire son entrée unaire.

3. additionner deux entrées binaires
4. multiplier deux entrées binaires
5. composer deux fonctions calculées par deux machines différentes
6. calculer α_2

Exercice 10 (discussion) :
Comment simuler une machine de Turing en python ? Comment simuler un programme python avec une machine de Turing ? Et si on remplace python par un autre langage de programmation ?

Exercice 11 (avancé) :
Supposons qu'une machine de Turing s'arrête au bout de t étapes de calcul en consommant s cases mémoires. Quelle(s) relation(s) existent entre t et s ?

Exercice 12 (discussion) :
Peut-on imaginer des machines de Turing à plusieurs rubans ? Sont-elles plus puissantes en terme de puissance de calcul ? De temps de calcul ? Que perd-on en limitant la taille de l'alphabet au minimum (0, 1 et B) ? Et en limitant le nombre d'états internes ?

Les problèmes indécidables (video 4)

Exercice 13 (base) :
Il est possible en Python d'exécuter un programme donné sous forme de chaîne de caractères.

```
progtxt='''print(1 + 1)'''
exec(progtxt)
```

Il est possible de généraliser cela au cas où le programme utilise des variables que l'on spécifie à l'exécution via un dictionnaire :

```
progtxt2='''print(x + x)'''
exec(progtxt2, {"x":2})
```

On cherche à écrire un programme Python arret qui prend 2 arguments : prog (qui est une chaîne de caractère contenant un programme qui utilise la variable x) et input. C'est-à-dire quelque chose de la forme :

```
def arret(prog, input):
    [...]
```

De manière à ce que : si exec(prog, "x" :input) termine, alors arret(prog, input) rend True, et sinon arret(prog, input) rend False.

Est-ce possible ? Si oui comment l'écrire, si non pourquoi ne peut-on pas l'écrire ?

Exercice 14 (base) :
Montrer par réduction que l'ensemble des machines de Turing dont le calcul boucle sur l'entrée 101010 est indécidable. Montrer que l'ensemble des programmes python dont le calcul boucle sur l'entrée 101010 est indécidable (on pourra expliciter la réduction)

Exercice 15 (base) :
La conjecture de Goldbach affirme que tout entier pair est la somme de deux nombres premiers. Montrer que si on se donne une machine résolvant le problème de l'arrêt, on peut l'utiliser pour construire une machine résolvant la conjecture de Goldbach (on ne donnera pas les règles de la machine, expliquer comment on fait). Si halt est la fonction python qui décide si le calcul du code x sur l'entrée vide s'arrête, écrire un programme python pour résoudre la conjecture de Goldbach.

Exercice 16 (un peu plus avancé) :
La conjecture de Syracuse affirme que pour tout entier a la suite définie par $u_0 = a$ et

$$u_{n+1} = \begin{cases} u_n/2 & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$$

atteint 1 pour un certain n.

Montrer que si on se donne une machine résolvant le problème de l'arrêt, on peut l'utiliser pour construire une machine résolvant la conjecture de Syracuse. Si halt est la fonction python qui décide si le calcul du code x sur l'entrée vide s'arrête, écrire un programme python pour résoudre la conjecture de Syracuse.

Exercice 17 (discussion) :

Qu'ont en commun les conjectures de Golbach et de Syracuse ? Comment utiliser cette méthode plus généralement pour trouver si une conjecture est un théorème ?

Exercice 18 (discussion) :

Peut-on être NP-dur et indécidable ? NP-dur et décidable ?

Le théorème de Rice (video 5)

Exercice 19 (base) :

Étant donné une fonction python F à un argument (entier binaire) et un entier n, dire si les problèmes suivants sont décidables ou non :

1. F décide d'un langage infini (on suppose que la fonction retourne toujours 0 ou 1)
2. F calcule la fonction constante n (on suppose que la fonction retourne toujours un entier)
3. F contient l'instruction "while"

Exercice 20 (avancé) :

Les problèmes suivants sont-ils décidables :

1. L'ensemble des machines de Turing qui s'arrêtent en moins de 17 étapes de calcul pour tout mot ω
2. L'ensemble des fonctions python F à un argument (entier binaire) qui s'arrêtent sur l'entrée 101010.
3. L'ensemble des fonctions python F à un argument (entier binaire) qui sur l'entrée 101010 retournent 101010
4. L'ensemble des fonctions python F à un argument (entier binaire) dont le calcule s'arrête sur un nombre infini d'entrées

Exercice 21 (discussion) :

Donnez des propriétés syntaxiques et des propriétés sémantiques des machines, dire lesquelles sont décidables ou indécidables.