

NOM:

# Calculabilité - complexité

*0h45, sans document sauf une feuille A4, 22 avril 2022*

---

Toutes les questions ont le même poids. Les réponses doivent être inscrites à la suite de chacune des questions, dans le cadre prévu à cet effet. Dans le cas (exceptionnel), où vous estimeriez qu'une question nécessite un développement plus important, utilisez une feuille séparée, en précisant le numéro de la question.

---

## 1. Calculabilité

### 1. Comparer l'expressivité de $TLA^+$ et du langage C

- $TLA^+$  est plus expressif que C car il contient des théories mathématiques évoluées (théorie des nombres, théorie des ensembles) non présentes dans C.
- C est un langage de programmation donc plus expressif que  $TLA^+$  qui est un langage de spécification.
- Ils sont équivalents car Turing-complets tous les deux.
- $TLA^+$  est Turing-parfait, mais pas C.
- On ne sait pas, personne n'a envisagé de comparer les deux.

### 2. On considère le castor affairé (*busy beaver*) à 10 états. Lesquelles des propositions suivantes sont vraies (plusieurs choix possibles)

- Il existe un castor affairé à 9 états qui fait plus de transitions que lui.
- L'exécution du castor affairé prendrait plusieurs millénaires sur un ordinateur actuel.
- Il peut parfois écrire plus de 1 qu'il ne fait de transitions.
- On sait qu'il va s'arrêter un jour.
- Toute machine de Turing à 10 états qui s'arrête fait moins de transitions que ce castor affairé.
- Il ne sert à rien.

### 3. Savoir si deux codes calculent la même chose est indécidable. Savoir si deux étudiants ont rendu le même projet est :

- Indécidable.
- Décidable pour un projet dans un langage de programmation, indécidable pour un langage de spécification de type  $TLA^+$ .
- Décidable dans tous les cas.

4. Une machine de Turing universelle peut-elle simuler l'exécution d'une machine non déterministe? Justifier la réponse.
5. Savoir si une machine de Turing a un nombre pair d'états est-il décidable ? Si oui, donner le principe de l'algorithme; si non, argumenter (montrer une réduction ou une contradiction).
6. Savoir si une machine de Turing avec un argument  $a$  s'arrête en moins de  $t$  transitions est-il décidable? Si oui, donner le principe de l'algorithme; si non, argumenter (montrer une réduction ou une contradiction).
7. Savoir si une machine de Turing avec un argument  $a$ , si elle s'arrête, a effectué un nombre pair de transitions est-il décidable?
8. Étant donné une machine de Turing  $\mathcal{M}$  sur l'alphabet  $\{0, 1\}$ , savoir si l'exécution de  $\mathcal{M}$  sur un ruban initialement vide écrit (au moins une fois) deux 1 consécutifs sur le ruban est-il décidable? Si oui, donner le principe de l'algorithme, sinon argumenter.
9. Existe-t-il des fonctions récursives primitives dont on ne peut pas prouver qu'elles terminent, ou est-on toujours sûr qu'elles finiront par donner un résultat ? Justifier la réponse.
10. Existe-t-il des fonctions récursives dont on ne peut pas prouver qu'elles terminent, ou est-on toujours sûr qu'elles finiront par donner un résultat ? Justifier la réponse.

## 2. Complexité

11. La complexité d'un problème dans  $P$  dépend du langage de programmation utilisé pour le résoudre.

- Oui, c'est pour cela que nous disposons de multiples langages de programmation, chacun étant plus efficace sur certains types de problèmes.
- Oui sauf si le problème est aussi dans **PSPACE**.
- Non, tous les modèles de calcul sont équivalents pour **P**.
- Ça dépend si le problème est dans **NP** ou pas.

12. Donner l'intuition qui justifie que l'opinion générale pense que  $P \subsetneq NP$ .

13. Quelles propriétés doit satisfaire la classe des calculs "efficaces" en temps?

14. Est-il vrai que, comme SAT est  $NP$ -complet, tout problème de décision peut être résolu en le réduisant à SAT et en utilisant un solveur SAT?

15. Comment est définie la complexité en espace d'un algorithme ?

16. Quelle est la différence entre l'espace et le temps en tant que ressources à mesurer ?

17. Pourquoi peut-on parler de la classe  $LSPACE$  (espace logarithmique) alors que ça n'a pas de sens de parler d'une classe  $L$  (temps logarithmique)?

18. Existe-il des problèmes dans  $P$  qui nécessitent un nombre exponentiel de cases?

19. Existe-t-il des problèmes  $PSPACE$ -complets qui ne soient pas aussi  $NP$ -complets?

**20. Un problème soluble en temps probabiliste polynomial (dans  $RP$ ) peut ne pas avoir de solution déterministe polynomiale (pas dans  $P$ ).**

- Oui, c'est l'apport des tirages aléatoires.
- Non, on peut simuler un tirage aléatoire en visitant les deux cas.
- On ne sait pas mais on soupçonne que les deux classes sont égales.
- C'est l'inverse, il y a des problèmes solubles déterministiquement en temps polynomial et qui n'ont pas de solution polynomiale avec des tirages aléatoires car ceux-ci peuvent mener l'algorithme dans une mauvaise branche du calcul.