

Le barème est indicatif – Documents autorisés : Notes de cours et TD

La notation attachera une grande importance à la clarté et à la concision des justifications.

Exercice 1

7 Points

Répondez aux questions suivantes en **justifiant brièvement** vos réponses (toute réponse non justifiée vaut zéro)

Rq : X est un ensemble décidable si la question “ $x \in X$?” est un problème décidable.

- a) On considère l’ensemble des fonctions de $\{0, 1, \dots, 41, 42\}$ vers \mathbb{N} . Cet ensemble est-il dénombrable ?

Oui. Une fonction de $\{0, \dots, 42\}$ vers \mathbb{N} est la même chose qu’un tuplets de 43 entiers positifs (les valeurs de la fonction). Or on a montré en TD que \mathbb{N}^k est dénombrable.

- b) On considère l’ensemble des machines de Turing dont le calcul termine quand on leur passe le mot vide en entrée. Cet ensemble est-il dénombrable ?

Oui. L’ensemble de *toutes* les MT est dénombrable. Or tout sous-ensemble d’un ensemble dénombrable est aussi (au plus) dénombrable.

- c) Est-il vrai que toute fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ qui est non bornée est également non calculable ?
Non. L’identité ($f(n) = n$ pour tout n) est calculable, mais non bornée.

- d) On considère le problème de l’arrêt des programmes **LOOP** (étant donné un programme **LOOP** P , est-ce que P s’arrête sur l’entrée 0?). Ce problème est-il décidable ?

Oui. Tout programme LOOP termine toujours, donc le problème est trivial (réponse toujours “OUI”).

- e) Si X est un ensemble décidable, est-il vrai que tout ensemble Y tel que $Y \subseteq X$ est aussi décidable ?

Non. Exemple : $X = \{0, 1\}^*$ et Y est l’ensemble des séquences sur $\{0, 1\}$ qui codent des programmes qui s’arrêtent toujours.

L’inclusion n’est pas une réduction.

- f) On considère les deux problèmes suivants :

(a) **PARITY** : Étant donné un nombre entier n , celui-ci est-il pair ?

(b) $HALT_0$: Étant donné un programme **WHILE** P , celui-ci s’arrête-t’il sur l’entrée 0 ?

Est-il vrai que $HALT_0$ se réduit à **PARITY** ?

Non. **HALT** est indécidable, **PARITY** est décidable, or un problème indécidable ne peut pas se réduire à un problème décidable (sinon il serait décidable).

g) On reprend les deux problèmes de la question précédente. Est-il vrai que *PARITY* se réduit à *HALT*₀ ?

Oui. On peut donner la réduction suivante : si l'entier n est pair, on lui associe le programme qui ne fait rien (et donc termine sur 0) ; si n est impair, on lui associe un programme qui boucle (par exemple WHILE (true) DO skip OD).

Exercice 2

3 Points

Écrire un programme **LOOP** qui prend en entrée un entier $n \in \mathbb{N}$ et retourne 1 si tous les chiffres dans la représentation décimale de n sont les mêmes et 0 sinon. Par exemple, si $n = 2222$ ou $n = 555$, alors le programme retourne 1. Par contre, si $n = 42$ ou $n = 1119$, le programme retourne 0.



Indication :

Vous pouvez réutiliser directement les fonctions arithmétiques vues en TD (addition, multiplication, division, modulo,...) dans votre programme.

P(n)

```
x := n;
res := 1;
loop (n) do
  rest := modulo(x,10);
  y := div (x,10);
  if (y = 0) then skip
  else
    if (rest != modulo(y,10)) then
      res := 0
    fi
  x := y
fi
od
```

Exercice 3

4 Points

On considère l'alphabet $\Sigma = \{a, b\}$. Décrire une machine de Turing à **une seule bande infinie dans les deux sens** qui accepte le langage des palindromes. On rappelle qu'un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche : *ababa* est un palindrome mais *baba* n'en est pas un.

On demande à la fois de donner précisément le code source de la machine (la liste de ses transitions) et de décrire son fonctionnement informellement.



Indication :

Il existe une solution qui utilise 6 états (plus les deux états finaux). Bien qu'il n'est pas demandé de donner précisément cette solution, cela signifie qu'il n'est pas nécessaire d'utiliser beaucoup d'états. De plus, on note qu'il n'est pas demandé de préserver le mot passé en entrée à la machine (on est autorisé à l'écraser si besoin est).

Exercice 4**6 Points**

On considère des variantes des machines de Turing déterministes à une seule bande dont la tête de lecture ne peut **que se déplacer vers la droite**. C'est-à-dire que toutes leurs transitions sont de la forme $(p, a, b, \rightarrow, q)$ (où p, q sont des états et a, b des lettres).

- a) Prouver que le problème de l'arrêt est décidable pour ces machines. On demande donc de donner un algorithme qui prend en entrée la description d'une telle machine M et un mot d'entrée w pour M . Votre algorithme doit tester si le calcul de M sur le mot w termine.
- b) Reprendre la Question 1 dans le cas où on considère des machines qui peuvent également ne pas déplacer leur tête de lecture (c'est-à-dire que toutes leurs transitions ont l'une des deux formes suivantes : $(p, a, b, \rightarrow, q)$ ou $(p, a, b, -, q)$).

Considérons une MT M dont la tête se déplace qu'à droite, et une entrée w .

- a) On simule M sur w tant que la tête reste sur w . Si M s'arrête pendant cette étape, on termine et répond OUI.
- b) Si la tête de M atteint le premier blanc après w : on continue la simulation, mais pas plus que $|Q|$ pas ($Q =$ ensemble d'états de contrôle). Parce que si M fait $|Q| + 1$ pas sur les blancs, alors il y aura 2 configurations avec le même état de contrôle, et donc M serait dans une boucle.

Donc, si M ne s'arrête pas après au plus $|Q|$ pas sur les blancs, on répond NON.

Pour la deuxième partie il faut aussi considérer la possibilité que M boucle en lisant tout le temps la même case du ruban. Ici, il suffit de simuler $|Q||\Gamma|$ pas sur la même case (Γ étant l'alphabet de ruban). Au-delà de $|Q||\Gamma|$ pas sur la même case, M sera dans une boucle en lisant cette case.