Exercice 1: `DECIDABLE` est-il décidable ?

On s'intéresse dans cet exercice à établir si le problème de déterminer si un problème est décidable est lui-même décidable. Autrement dit, soit `DECIDABLE` le langage contenant l'ensemble des descriptions des machines de Turing M qui reconnaissent des langages décidables, `DECIDABLE` est-il décidable ?

1. Soit L un langage décidable, et M la machine de Turing qui reconnait L. Que signifie pour M le fait que L soit décidable ?
2. En déduire l'expression de `DECIDABLE` sous la forme `DECIDABLE` $= \{...\}$.
3. Montrer que `DECIDABLE` est indécidable.
4. Quel théorème du cours permettrait d'arriver aussi à cette conclusion ? On appliquera ce théorème **rigoureusement**.
5. Quelle aurait été la conséquence pour l'informatique si `DECIDABLE` avait été décidable ?

Solution de l'exercice 1:

1. Fait en cours.
2. Fait en cours.
3. Utiliser l'une des deux méthodes suivantes :
   (a) **En utilisant la méthode de réduction :**
   soit la fonction f suivante : $f(< M, w >) =$ si $w \neq x$ alors rejeter
   exécuter $M(w)$.
   accepter
   Clairement, f est une fonction calculable.
   Montrons que f est une réduction du problème de l'arrêt H_{TM} vers `DECIDABLE` :
   – si $M(w)$ s'arrête, alors $N(x) = f(\langle M, w\rangle)$ rejette toujours (=décide) sauf si $x = w$ où le résultat dépend de l'exécution $M(w)$. Si $M(w)$ s'arrête, alors $N(x)$ est décidable (car s'arrête dans tous les cas). Inversement, si $M(w)$ boucle, alors $N(x)$ boucle aussi et N n'est pas décidable.
   – si $N(x)$ est décidable, comme $N(x)$ décide pour tout x, donc également pour $x = w$ qui décide si $M(w)$ décide.
   En conséquence, $\langle M, w\rangle \in H_{\text{TM}} \Leftrightarrow f(\langle M, w\rangle) \in$ `DECIDABLE`. On en déduit donc que f est une réduction de H_{TM} vers `DECIDABLE`.
   Or, le problème de l'arrêt est indécidable, et en conséquence `DECIDABLE` est lui-aussi indécidable.
   (b) **En utilisant un raisonnement par l'absurde :**
   supposons que `DECIDABLE` soit décidable, alors il existe une machine de Turing R qui décide `DECIDABLE`. Alors on peut construire le décideur T suivant :
   $T(< M, w >) =$ construire $N_{<M,w>}(x) =$ si $w \neq x$ alors rejeter
   exécuter $M(w)$.
   accepter
   exécuter $R(\langle N_{<M,w>}\rangle)$
   décider comme R
   Clairement, T est un décideur pour H_{TM} puisque décider si $N_{<M,w>} \in$ `DECIDABLE` avec R est équivalent à décider si $M(w) \in H_{\text{TM}}$ avec T.
   Or, le problème de l'arrêt est indécidable. Donc, ni T, ni R n'existent, et en conséquence `DECIDABLE` est indécidable.
4. Fait en cours.
5. Il aurait permis de décider si un programme s'arrête toujours.