DM8 (éléments de réponses)

Question 1. Le problème Appartient s'apparente au problème de l'arrêt ! Ce problème est semi-décidable, la fonction suivant le résout partiellement.

```
let appartient <f> x =
  universel <f> x
```

En effet, si l'appel $f(x)$ termine et renvoie true, alors la fonction renvoie `true`, sinon la fonction ne termine pas ou renvoie `false`.

On suppose qu'il existe une fonction `appartient : string -> string -> bool` qui résout ce problème pour toutes les instances. On pose :

```
let paradoxe <f> x =
  if appartient <f> <f> then paradoxe <f>
  else true
```

Alors :

- si `paradoxe` termine et renvoie `true`, cela signifie que `appartient` renvoie `false`, donc que `paradoxe` ne termine pas ou renvoie `false` ;
- si `paradoxe` ne termine pas (elle ne peut pas renvoyer `false`), cela signifie que `appartient` renvoie `true`, donc que paradoxe termine et renvoie `true`.

Dans les deux cas, il y a contradiction donc la fonction `appartient` ne peut pas exister. Le problème est indécidable.

Question 2. Supposons le problème semi-décidable. Soit A un algorithme qui le résout partiellement.

- Si $\langle A\rangle$ est une instance positive de Diagonal alors $A(\langle A\rangle)$ termine et renvoie `true`. Mais comme c'est une instance positive, cela veut aussi dire que $\langle A\rangle \notin L(A)$.
- Si $\langle A\rangle$ est une instance négative de Diagonal alors $A(\langle A\rangle)$ ne termine pas ou renvoie `false`. Mais comme c'est une instance négative, cela veut aussi dire que $\langle A\rangle \in L(A)$.

Dans les deux cas on aboutit à une contradiction.

Question 3. Soit $\langle f\rangle \in \Sigma^*$. Alors $\langle f\rangle$ est une instance positive de Diagonal si et seulement si $\langle f\rangle \notin L(f)$ si et seulement si $(\langle f\rangle, \langle f\rangle)$ est une instance négative de Appartient si et seulement si $(\langle f\rangle, \langle f\rangle)$ est une instance positive de coAppartient. Comme la fonction $\langle f\rangle \mapsto (\langle f\rangle, \langle f\rangle)$ est calculable, on en déduit bien la réduction voulue.
On en déduit d'une part que coAppartient n'est pas semi-décidable (on aurait déjà pu le déduire de la question 1), d'autre part que coDiagonal $\leqslant$ Appartient. Par conséquent, coDiagonal est semi-décidable.

Question 4. Diagonal est une propriété des langages de fonctions car $\{\langle f\rangle \in \Sigma^* \mid \langle f\rangle \notin L(f)\}$ est bien une partie de Σ^*.

Question 5. Comme on cherche à montrer que P est indécidable, on peut soit montrer que P est indécidable, soit que co P est indécidable. L'un de ces deux problèmes n'a pas $\varnothing$ comme instance positive. Par ailleurs, on remarque que si P est une propriété non triviale des langages semi-décidables, alors co P l'est également (les rôles de L_1 et L_2 sont inversés).

Question 6. Distinguons les cas :

- si $(\langle f\rangle, x)$ est une instance positive de Appartient alors `universel <f> x` renvoie `true`, donc pour toute entrée y l'appel $g(y)$ a le même résultat que l'appel $f_L(y)$. On en déduit que $L(g) = L(f_L) = L \in P$, donc $\langle g\rangle$ est une instance positive de P ;
- si $(\langle f\rangle, x)$ est une instance négative de Appartient alors l'appel `universel <f> x` ne termine pas ou renvoie `false`. On en déduit que pour toute entrée y, l'appel $g(y)$ ne renvoie jamais true, soit que $L(g) = \varnothing$. Comme on a supposé lors de la question précédente que $\varnothing \notin P$, cela signifie que $\langle g\rangle$ est une instance négative de P.

Par ailleurs, la construction de g est clairement calculable, ce qui montre bien que Appartient $\leqslant P$.

Question 7. Comme Appartient n'est pas décidable, on en déduit que P non plus, ce qui conclut le théorème de Rice.

Question 8. On commence par montrer que la propriété « être non vide » est bien une propriété non triviale des langages semi-décidables. En effet :

- le langage $\varnothing$ est vide et est semi-décidable (et même décidable), par la fonction :

  ```
  let f 1 x = false
  ```

- le langage Σ^* est non vide et est semi-décidable (et même décidable), par la fonction :

  ```
  let f 2 x = true
  ```

Par le théorème de Rice, on en déduit que le problème est bien indécidable. On a ici $P = \mathcal{P}(\Sigma^*) \setminus \{\varnothing\}$.

Question 9.
□ **9.1.** Ce problème est décidable car il suffit de « lire » le code source et de compter le nombre de boucles `while`. Aucune exécution de ce code source n'est nécessaire.
□ **9.2.** Ce résultat ne contradit pas le théorème de Rice car ce dernier concerne les propriétés des langages de fonctions, et non les propriétés sur les fonctions elles-mêmes (leur code source).