TD

Théorème de Rice

Hasley William

Question 1. Ce problème est bien semi-décidable : Il suffit d'exécuter f sur x et de renvoyer la valeur de retour : Formellement, $\textbf{univ}\langle f\rangle\ x = \top \iff f$ termine et renvoie vrai sur x.

Cependant, ce problème n'est pas décidable : Supposons qu'il existe $\mathscr{A}$, algorithme répondant à **Appartient**. Soit $(\langle f\rangle, x)$, entrée de **Appartient**. Construisons un algorithme $f_\top$ via le code suivant :

```
let f_top (f : string -> bool) : string -> bool = fun x -> let val = f x in true
```

Alors $f_\top(x)$ renvoie vrai si et seulement si $f_\top(x)$ termine, si et seulement si $f(x)$ termine. Ainsi, via cette transformation (notée φ) de $(\langle f\rangle, x)$ en $(\langle f_\top\rangle, x)$, nous réduisons le problème de l'arrêt à **Appartient** :

Soit $(\langle f\rangle, x)$, instance de **Halts**. Alors $\textbf{Halts}((\langle f\rangle, x)) = \top \iff f(x)$ termine $\iff f_\top(x)$ termine $\iff f_\top(x)$ termine et renvoie vrai $\iff \textbf{Appartient}((\langle f_\top\rangle, x)) = \top$.

Or, le problème de l'arrêt est indécidable, donc **Appartient** l'est aussi, sinon nous pourrions décider **Halts** via φ.

Question 2. Supposons que **Diagonal** soit semi-décidable via $\mathscr{A}$. Intéressons-nous à $\textbf{Diagonal}(\langle\mathscr{A}\rangle)$:

La question se réduit à "$\mathscr{A}(\langle\mathscr{A}\rangle) = \top$?". Supposons que $\langle\mathscr{A}\rangle \notin \mathscr{L}(\mathscr{A})$, alors $\mathscr{A}$ renvoie vrai lorsqu'appliqué à $\langle\mathscr{A}\rangle$ (du fait de sa semi-décidabilité). Alors $\langle\mathscr{A}\rangle \in \mathscr{L}(\mathscr{A})$: Ceci est absurde.

Supposons maintenant que $\langle\mathscr{A}\rangle \in \mathscr{L}(\mathscr{A})$, alors par définition de $\langle\mathscr{A}\rangle \in \mathscr{L}(\mathscr{A})$, $\mathscr{A}(\langle\mathscr{A}\rangle)$ renvoie faux : Ceci est également absurde. Alors **Diagonal** n'est pas semi-décidable, donc indécidable (s'il était décidable, l'algorithme associé réaliserait une semi-décision).

Question 3. Soit $\langle f\rangle$, entrée de **Diagonal**. Alors posons $\varphi(\langle f\rangle) = (\langle f\rangle, \langle f\rangle)$.

Il vient alors que $\textbf{Diagonal}(\langle f\rangle) = \top \iff \langle f\rangle \notin \mathscr{L}(f) \iff \textbf{coAppartient}(\langle f\rangle, \langle f\rangle) = \top$.

Ainsi, **Diagonal** $\leq_m$ **coAppartient** via φ (le fait que φ) soit polynomiale en $|\langle f\rangle|$ est évident : nous copions simplement $\langle f\rangle$.

Nous en déduisons donc que **coAppartient** est non semi-décidable (ce qui est cohérent, sans quoi **Appartient** serait décidable). De plus, nous déduisons que **coDiagonal** $\leq_m$ **Appartient** (via la même transformation), donc **coDiagonal** est semi-décidable (par cette réduction à **Appartient**, semi-décidable).

Question 4. Posons $P = \{\mathscr{L}(f) \in \mathscr{P}(\Sigma^*) \mid \langle f\rangle \in \Sigma^*, \langle f\rangle \notin \mathscr{L}(f)\} \subseteq \mathscr{P}(\Sigma^*)$: Alors par construction, le problème associé à P est exactement **Diagonal**.

Question 5. Si $\emptyset \in P$, il suffit de démontrer que P^{c} est indécidable, auquel cas P l'est aussi.

Question 6. Soit $(\langle f \rangle, x)$, entrée de **Appartient**.

Remarquons que $g(y) = \top \iff f(x) = \top = f_L(y)$. Ainsi, en supposant $\emptyset \notin P$, il vient que $\mathscr{L}(g) \in P \iff f(x) = \top$, car f_L est un algorithme décidant partiellement $L \in P$. Dès lors, posons la transformation :

```
let g (f : string) (x : string) : string -> bool = fun y -> univ f x && univ f_l y
```

Alors cette transformation est telle que $x \in \mathscr{L}(f) \iff \forall y \in L,\ \ y \in \mathscr{L}(g) \iff \mathscr{L}(g) \in P$, car si $f(x) = \bot,\ \ \mathscr{L}(g) = \emptyset \notin P$. Ainsi, **Appartient** $\leq_m P$ via cette transformation. (Qui est construite en temps $\mathscr{O}(1)$).

Question 7. La transformation précédente nous permet de déduire le Théorème de Rice : La Question 1. nous donne l'indécidabilité de **Appartient**. Or, ce problème se réduit au problème P (dont on peut supposer que $\emptyset \notin P$ car P est non Triviale). Ainsi, P est indécidable, car si P l'était, la transformation précédente nous permettrait de décider **Appartient**.

Question 8. Ce problème est indécidable d'après le théorème de Rice, car en posant $P = \{\emptyset\}^c$ (tout langage sauf le vide), il vient que P est une propriété des Langages non triviale, et notre problème correspond exactement à P, donc est à ce titre indécidable.

Question 9.

a) Ce problème est bien décidable, il suffit de parser le code pour compter le nombre d'occurences du token WHILE (avec espace à gauche et droite, et hors string), ce que tout compilateur est en théorie capable de faire.

b) Le fait que ce problème soit décidable n'entre pas en conflit avec le Théorème de Rice, car la propriété "s'écrire avec au moins 5 boucles while" n'est pas une propriété sémantique mais Syntaxique ! Cette propriété ne porte pas sur le langage de la fonction passée en paramètre, car tout langage décidable / semi-décidable peut s'écrire via un algorithme auquel nous ajoutons 5 boucles while inutiles (via `while(false){}`).