TD Théorie n°1

Indécidabilité par réduction

Introduction, Notations.

L'objectif de ce TD est de réaliser des réductions d'un problème à un problème indécidable. Dans ce TD, on ne traitera pas de réductions polynomiale à un problème **NP**-difficile.

Une machine $\mathscr{M}$ pourra être décrite sous forme d'un algorithme, d'un programme C, ou d'un programme OCaml. Quel que soit le format, on prendra soin de n'utiliser *que* des fonctions calculables, ce n'est pas toujours évident lorsqu'on écrit une machine sous forme d'algorithme.

Dans ce TD, la sérialisation d'un objet mathématique x sera noté $\langle x \rangle$. Cette notation est là pour simplifier l'écriture des machines. Cependant, si une sérialisation n'est pas évidente pour un « type d'objet », on pourra préciser une fonction de sérialisation associée à ce « type ». Ainsi, on notera $\langle (x, y) \rangle$ la sérialisation du couple (x, y) (ceci peut être étendu pour tout n-uplet). On notera $\langle \mathscr{M} \rangle$ la sérialisation de la machine $\mathscr{M}$: dans le cas du coude C, ou du code OCaml, il s'agira du code source de $\mathscr{M}$, pour un algorithme, ce sera une description suffisamment précise de l'algorithme.

On notera $\mathscr{U}$ une machine universelle, calculant la fonction `interprète` :

$$\texttt{interprète}(\mathscr{M}, w) = \begin{cases} w' \text{ si } w \xrightarrow[\mathscr{M}]{} w' \\ \text{non défini sinon} \end{cases}.$$

On pourra appeler cette fonction en C ou en OCaml avec la fonction `interprète` ou `exécute`. On utilisera le mot clé « Interprète » ou « Exécute » dans un algorithme.

On définit $w^{\text{rev}} = w_n w_{n-1} \dots w_1$ le mot w renversé où $w = w_1 w_2 \dots w_n$. On notera $L^{\text{rev}} = \{w^{\text{rev}} \mid w \in L\}$ le langage L renversé, et l'opération $L \mapsto L^{\text{rev}}$ est appelée *renversement*.

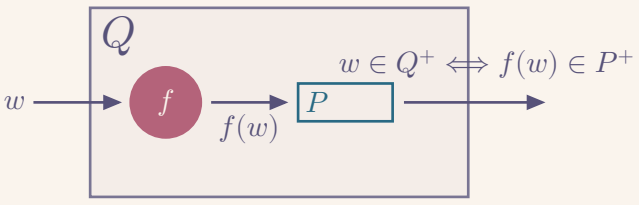
On notera L_P le langage d'un problème P (*i.e.* P^+). On définit le problème $\textsc{Appartient}_L$ qui décide si $w \in L$.

On rappelle la structure d'une réduction.

Définition. On dit qu'un problème Q *se réduit à* P dès lors qu'il existe une fonction $f : \mathscr{E}_Q \to \mathscr{E}_P$ *calculable* telle que :

$$w \in Q^+ \iff f(w) \in P^+.$$

On notera alors $Q \preccurlyeq P$.



Si $Q \preccurlyeq P$ et que Q n'est pas décidable, alors P le n'est pas non plus.

On rappelle le théorème de l'arrêt :

Théorème. Le problème Arrêt est indécidable.

Arrêt : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine, et $w \in \Sigma^\star$
Sortie. La machine $\mathscr{M}$ s'arrête-t-elle sur l'entrée w ?

On pourra également utiliser les propriétés de stabilité des langages décidables par union, par concaténation, par intersection et par complémentaire, même si ça n'est pas le but premier de ce TD.

Par exemple, le problème CoArrêt (sur la page suivante) est aussi indécidable, c'est le complémentaire du problème Arrêt.

CoArrêt :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine, et $w \in \Sigma^\star$
Sortie. Est ce que la machine $\mathscr{M}$ ne s'arrête pas sur l'entrée w ?

On admettra que les problèmes ci-dessous sont indécidables (ils viennent du TD 11). On pourra utiliser ces problèmes pour les réductions, mais pas ceux définis après la ligne.

Parmi ces problèmes, un est décidable. Trouvez lequel !

ArrêtUniv :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Est-ce que la machine $\mathscr{M}$ s'arrête sur toutes ses entrées ?

ArrêtExiste :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Est-ce que la machine $\mathscr{M}$ s'arrête sur une de ses entrées ?

ArrêtSimult :
Entrée. Deux sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de machines.
Sortie. Est-ce que les machines $\mathscr{M}$ et $\mathscr{N}$ s'arrêtent sur les mêmes entrées ?

Arrêt$_w$:
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. La machine $\mathscr{M}$ s'arrête-t-elle sur l'entrée w ?

Début du TD.

Régulier :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il régulier ?

RÉPONSE. Montrons que Régulier se réduit à CoArrêt. Soit $(\mathscr{M}, w)$ une entrée du problème CoArrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est un élément de $\{a^n b^n \mid n \in \mathbb{N}\}$,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Régulier}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a^n b^n \mid n \in \mathbb{N}\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors Régulier est indécidable aussi.

Fini :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il fini ?

RÉPONSE. Montrons que Fini se réduit à CoArrêt. Soit $(\mathscr{M}, w)$ une entrée du problème CoArrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Fini}^+ &\iff \mathscr{L}(\mathscr{M}') \text{ est fini} \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors Fini est indécidable aussi.

Vide :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il vide ?

RÉPONSE. Montrons que VIDE se réduit à COARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème COARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{VIDE}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{COARRÊT}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème COARRÊT est indécidable, alors VIDE est indécidable aussi.

ACCEPTETOUT :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il $\Sigma^\star$?

RÉPONSE. Montrons que ACCEPTETOUT se réduit à ARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème ARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{ACCEPTETOUT}^+ &\iff \mathscr{L}(\mathscr{M}') = \Sigma^\star \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{ARRÊT}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème ARRÊT est indécidable, alors ACCEPTETOUT est indécidable aussi.

COMMENCEPAR_a :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Est-ce que la machine $\mathscr{M}$ s'arrête sur une entrée commençant par a ?

RÉPONSE. Montrons que COMMENCEPAR_a se réduit à ARRÊT_w. Soit $\mathscr{M}$ une entrée du problème ARRÊT_w. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée commence par a,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{COMMENCEPAR}_a^+ &\iff \mathscr{L}(\mathscr{M}') \cap \{a\}\Sigma^\star \neq \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{ARRÊT}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a\}\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème ARRÊT est indécidable, alors COMMENCEPAR_a est indécidable aussi.

ENTRÉEPAIR :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Est-ce que la machine $\mathscr{M}$ s'arrête sur une entrée de taille paire ?

RÉPONSE. Montrons que ENTRÉEPAIR se réduit à ARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème ARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est de taille paire,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{EntréePair}^+ &\iff \mathscr{L}(\mathscr{M}') = (\Sigma\Sigma)^\star \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $(\Sigma\Sigma)^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors EntréePair est indécidable aussi.

LangagePair : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Est-ce que $\mathscr{L}(\mathscr{M})$ admet un nombre pair de mots ?

RÉPONSE. Montrons que LangagePair se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est a ou aa,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{LangagePair}^+ &\iff \mathscr{L}(\mathscr{M}') = \{a, aa\} \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a, aa\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors LangagePair est indécidable aussi.

Local : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il local ?

RÉPONSE. Montrons que Local se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est aa ou ab
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{Local}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{aa, ab\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors Local est indécidable aussi.

LangageImpair : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Est-ce que $\mathscr{L}(\mathscr{M})$ admet un nombre impair de mots ?

RÉPONSE. Montrons que LangageImpair se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est a,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{LangageImpair}^+ &\iff \mathscr{L}(\mathscr{M}') = \{a\} \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors LangageImpair est indécidable aussi.

Infini : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il infini ?

RÉPONSE. Montrons que Infini se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Infini}^+ &\iff \mathscr{L}(\mathscr{M}') \text{ est infini} \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors Infini est indécidable aussi.

Palindrome : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Les mots acceptés par $\mathscr{M}$ sont-ils tous des palindromes ?

RÉPONSE. Montrons que Palindrome se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est un palindrome,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Palindrome}^+ &\iff \forall s \in \mathscr{L}(\mathscr{M}'), s^{\text{rev}} = s, \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit l'ensemble des palindromes,[1] en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors Palindrome est indécidable aussi.

Renversement : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il stable par renversement[2] ?

RÉPONSE. Montrons que Renversement se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est un palindrome,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Renversement}^+ &\iff \mathscr{L}(\mathscr{M}) = \mathscr{L}(\mathscr{M})^{\text{rev}} \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit l'ensemble des palindromes,[1] en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors Renversement est indécidable aussi.

Concat : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il la concaténation de deux langages décidables ?

[1] Cet ensemble est formellement $\{w\, w^{\text{rev}} \mid w \in \Sigma^\star\} \cup \{w\, x\, w^{\text{rev}} \mid x \in \Sigma, w \in \Sigma^\star\}$.
[2] Un langage L est stable par renversement si $L^{\text{rev}} = L$.

RÉPONSE. Ce langage est **décidable**. En effet, tout langage décidable L peut s'écrire comme $\{\varepsilon\} \cdot L = L$. Le langage $\{\varepsilon\}$ est fini, donc reconnaissable. Ainsi, le langage L est reconnu par la machine $\mathscr{M}'$ ci-dessous.

(1) Renvoyer ***V***.

DISJOINTS : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines.
Sortie. Les langages de $\mathscr{M}$ et $\mathscr{N}$ sont-ils disjoints ?

RÉPONSE. Montrons que DISJOINTS se réduit à COARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème COARRÊT. Construisons les machines $\mathscr{M}'$ et $\mathscr{N}$ ($\mathscr{M}'$ est celle en haut, $\mathscr{N}$ est celle en bas) comme les algorithmes ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner ***V***.
  (b) sinon retourner ***F***.

$$\begin{aligned}\langle (\mathscr{M}', \mathscr{N}) \rangle \in \text{DISJOINTS}^+ &\iff \mathscr{L}(\mathscr{M}') \cap \mathscr{L}(\mathscr{N}) = \emptyset \\ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{COARRÊT}^+\end{aligned}$$

(1) Renvoyer ***V***.

D'où la réduction. Comme le problème COARRÊT est indécidable, alors DISJOINTS est indécidable aussi.

UNIONTOUT : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines.
Sortie. L'union des langages de $\mathscr{M}$ et $\mathscr{N}$ vaut-il $\Sigma^\star$?

RÉPONSE. Montrons que UNIONTOUT se réduit à ARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème ARRÊT. Construisons les machines $\mathscr{M}'$ et $\mathscr{N}$ ($\mathscr{M}'$ est celle en haut, $\mathscr{N}$ est celle en bas) comme les algorithmes ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner ***V***.
  (b) sinon retourner ***F***.

$$\begin{aligned}\langle (\mathscr{M}', \mathscr{N}) \rangle \in \text{UNIONTOUT}^+ &\iff \mathscr{L}(\mathscr{M}') \cup \mathscr{L}(\mathscr{N}) = \Sigma^\star \\ &\iff \mathscr{L}(\mathscr{M}') = \Sigma^\star \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{ARRÊT}^+\end{aligned}$$

(1) Renvoyer ***F***.

D'où la réduction. Comme le problème ARRÊT est indécidable, alors DISJOINTS est indécidable aussi.

CONCATTEST : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$, $\langle \mathscr{N} \rangle$ et $\langle \mathscr{K} \rangle$ de trois machines.
Sortie. Le langage de $\mathscr{M}$ est-il la concaténation du langage de $\mathscr{N}$ et de $\mathscr{K}$?

RÉPONSE. Montrons que CONCATTEST se réduit à ARRÊT. Soit $(\mathscr{Z}, w)$ une entrée du problème ARRÊT. Construisons les machines $\mathscr{M}$, $\mathscr{N}$ et $\mathscr{K}$ ($\mathscr{M}$ est celle en haut, $\mathscr{N}$ est celle au milieu, et $\mathscr{K}$ est celle en bas) comme les algorithmes ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\langle(\mathscr{M},\mathscr{N},\mathscr{K})\rangle \in \textsc{ConcatTest}^+ \iff \mathscr{L}(\mathscr{M}) = \overbrace{\mathscr{L}(\mathscr{N})\cdot\mathscr{L}(\mathscr{N})}^{\Sigma^\star}$$
$$\iff \mathscr{L}(\mathscr{M}) = \Sigma^\star$$
$$\iff \text{l'exécution de } \mathscr{Z} \text{ sur } w \text{ termine}$$
$$\iff \langle(\mathscr{Z},w)\rangle \in \textsc{Arrêt}^+$$

(1) Renvoyer **V**.

(1) Renvoyer **V**.

D'où la réduction. Comme le problème Arrêt est indécidable, alors ConcatTest est indécidable aussi.

Max3 :
Entrée. La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Les mots acceptés par $\mathscr{M}$ sont-ils de taille inférieure ou égale à 3 ?

RÉPONSE. Montrons que Max3 se réduit à Arrêt. Soit $(\mathscr{M},w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée a une taille inférieure ou égale à 3,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\langle\mathscr{M}'\rangle \in \textsc{Max3}^+ \iff \mathscr{L}(\mathscr{M}') = \bigcup_{i=0}^{3}\Sigma^i$$
$$\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine}$$
$$\iff \langle(\mathscr{M},w)\rangle \in \textsc{Arrêt}^+$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\bigcup_{i=0}^{3}\Sigma^i$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors Max3 est indécidable aussi.

NonStop :
Entrée. La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. La machine $\mathscr{M}$ ne s'arrête-t-elle pas, quel que soit l'entrée ?

RÉPONSE. Montrons que NonStop se réduit à CoArrêt. Soit $(\mathscr{M},w)$ une entrée du problème NonStop. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\langle\mathscr{M}'\rangle \in \textsc{NonStop}^+ \iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas}$$
$$\iff \langle(\mathscr{M},w)\rangle \in \textsc{CoArrêt}^+$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors NonStop est indécidable aussi.

Monoïde :
Entrée. La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il un monoïde, pour la concaténation ?

RÉPONSE. Montrons que Monoïde se réduit à Arrêt. Soit $(\mathscr{M},w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est ε,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{Monoïde}^+ &\iff \mathscr{L}(\mathscr{M}') = \{\varepsilon\}\\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine}\\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{\varepsilon\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors Monoïde est indécidable aussi.

OCaml : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. La machine $\mathscr{M}$ reconnaît-elle un programme OCaml ?

RÉPONSE. Montrons que OCaml se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est `3 + 4`,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{OCaml}^+ &\iff \mathscr{L}(\mathscr{M}') = \{\text{« 3 + 4 »}\}\\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine}\\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{\text{« 3 + 4 »}\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors OCaml est indécidable aussi.

StableConcat : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il stable par concaténation ?

RÉPONSE. Montrons que StableConcat se réduit à CoArrêt. Soit $(\mathscr{M}, w)$ une entrée du problème CoArrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est a
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{StableConcat}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset\\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas}\\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors OCaml est indécidable aussi.

TaillePremière : **Entrée.** La sérialisation $\langle\mathscr{M}\rangle$ d'une machine.
Sortie. La machine $\mathscr{M}$ accepte-t-elle les entrées dont la taille est un nombre premier ?

RÉPONSE. Montrons que TaillePremière se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est de taille première,
   (a) alors retourner **V**.
   (b) sinon retourner **F**.

$$\begin{aligned}\langle\mathscr{M}'\rangle \in \textsc{TaillePremière}^+ &\iff \mathscr{L}(\mathscr{M}') = \bigcup_{p \text{ premier}} \Sigma^p\\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine}\\ &\iff \langle(\mathscr{M}, w)\rangle \in \textsc{Arrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\bigcup_{p \text{ premier}} \Sigma^p$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors TaillePremière est indécidable aussi.

Automate :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il reconnu par un automate fini $\mathscr{A}$?

RÉPONSE. Montrons que Automate se réduit à CoArrêt. Soit $(\mathscr{M}, w)$ une entrée du problème CoArrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et l'entrée est un élément de $\{a^n b^n \mid n \in \mathbb{N}\}$,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Automate}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{a^n b^n \mid n \in \mathbb{N}\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors Automate est indécidable aussi.

AutomateLocal :
Entrée. La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il reconnu par un automate local $\mathscr{A}_{\text{loc}}$?

RÉPONSE. Montrons que AutomateLocal se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est aa ou ab
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{Local}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{CoArrêt}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{aa, ab\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors AutomateLocal est indécidable aussi.

AccepteCouple :
Entrée. Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines, et deux mots $(u, v) \in (\Sigma^\star)^2$.
Sortie. La machine $\mathscr{M}$ accepte-t-elle u si et seulement si $\mathscr{N}$ accepte v ?

RÉPONSE. Montrons que AccepteCouple se réduit à Arrêt. Soit $(\mathscr{M}, w)$ une entrée du problème Arrêt. On pose $u = b$, et $v = a$. Construisons les machines $\mathscr{M}'$ et $\mathscr{N}'$ ($\mathscr{M}'$ est en haut, et $\mathscr{N}'$ est au milieu) comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle (\mathscr{M}', \mathscr{N}', u, v) \rangle \in \text{AccepteCouple}^+ &\iff (u \in \mathscr{L}(\mathscr{M}') \iff v \in \mathscr{L}(\mathscr{N}')) \\ &\iff u \in \mathscr{L}(\mathscr{M}') \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{Arrêt}^+\end{aligned}$$

(1) Retourner **V**

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\Sigma^\star$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème Arrêt est indécidable, alors AccepteCouple est indécidable aussi.

COMMUTATIF : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
Sortie. Le langage de $\mathscr{M}$ est-il commutatif,[3] pour la concaténation ?

RÉPONSE. Montrons que COMMUTATIF se réduit à COARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème COARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(1) Exécute $\mathscr{M}$ sur w.
(2) Si l'exécution termine et que l'entrée est ab,
  (a) alors retourner **V**.
  (b) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{COMMUTATIF}^+ &\iff \mathscr{L}(\mathscr{M}') = \emptyset \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ ne termine pas} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{COARRÊT}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit $\{ab\}$, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème COARRÊT est indécidable, alors COMMUTATIF est indécidable aussi.

Fin du TD.

Les preuves par réductions se ressemblent étrangement. Pourquoi ? Ceci est dû au théorème de Rice. Ce théorème permet de prouver chacune des réductions ci-dessous. Voici une preuve de ce théorème. On définit tout d'abord le problème ci-dessous, pour $\mathscr{P} \subseteq \wp(\Sigma^\star)$:

VÉRIFIE$_{\mathscr{P}}$: **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine
Sortie. Est-ce que le langage de $\mathscr{M}$ vérifie la propriété $\mathscr{P}$?

On dit que L vérifie $\mathscr{P}$ si $L \in \mathscr{P}$. On dit qu'un prédicat $\mathscr{P}$ est trivial sur les langages machines s'il est toujours vrai ou toujours faux.

Théorème (Rice). Soit $\mathscr{P}$ un prédicat décidable ne dépendant pas de la machine, mais uniquement de son langage. Le problème VÉRIFIE$_{\mathscr{P}}$ est indécidable si $\mathscr{P}$ n'est pas un prédicat trivial sur les langages machines.

Comme $\mathscr{P}$ est non trivial sur les langages machines, soient K et L deux langages décidables tels que K vérifie $\mathscr{P}$, mais pas L.

(1) Supposons $K \neq \emptyset$. Montrons que VÉRIFIE$_{\mathscr{P}}$ se réduit à ARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème ARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

(a) Exécute $\mathscr{M}$ sur w.
(b) Si l'exécution termine et que l'entrée est un élément de K
  (i) alors retourner **V**.
  (ii) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \text{VÉRIFIE}^+_{\mathscr{P}} &\iff \mathscr{L}(\mathscr{M}') = K \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \text{ARRÊT}^+\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit K, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème ARRÊT est indécidable, alors VÉRIFIE$_{\mathscr{P}}$ est indécidable aussi.

(2) Supposons $K = \emptyset$. Ainsi, $L \neq \emptyset$. On notera $\neg\mathscr{P}$ le prédicat inversé, il est également décidable. Montrons que VÉRIFIE$_{\neg\mathscr{P}}$ se réduit à COARRÊT. Soit $(\mathscr{M}, w)$ une entrée du problème COARRÊT. Construisons la machine $\mathscr{M}'$ comme l'algorithme ci-dessous.

[3] Un langage L est commutatif si, pour $w = uv \in L$, on a $vu \in L$, quel que soit la découpe de w.

(a) Exécute $\mathscr{M}$ sur w.
(b) Si l'exécution termine et que l'entrée est un élément de L
  (i) alors retourner **V**.
  (ii) sinon retourner **F**.

$$\begin{aligned}\langle \mathscr{M}' \rangle \in \textsc{Vérifie}^{+}_{\neg\mathscr{P}} &\iff \mathscr{L}(\mathscr{M}') = L \\ &\iff \text{l'exécution de } \mathscr{M} \text{ sur } w \text{ termine} \\ &\iff \langle (\mathscr{M}, w) \rangle \in \textsc{CoArrêt}^{+}\end{aligned}$$

En effet, le langage de $\mathscr{M}'$ est, soit $\emptyset$, soit K, en fonction de si l'exécution termine ou non. D'où la réduction. Comme le problème CoArrêt est indécidable, alors $\textsc{Vérifie}_{\neg\mathscr{P}}$ est indécidable aussi.

Dans tous les cas, on en conclut que $\textsc{Vérifie}_{\mathscr{P}}$ est indécidable.