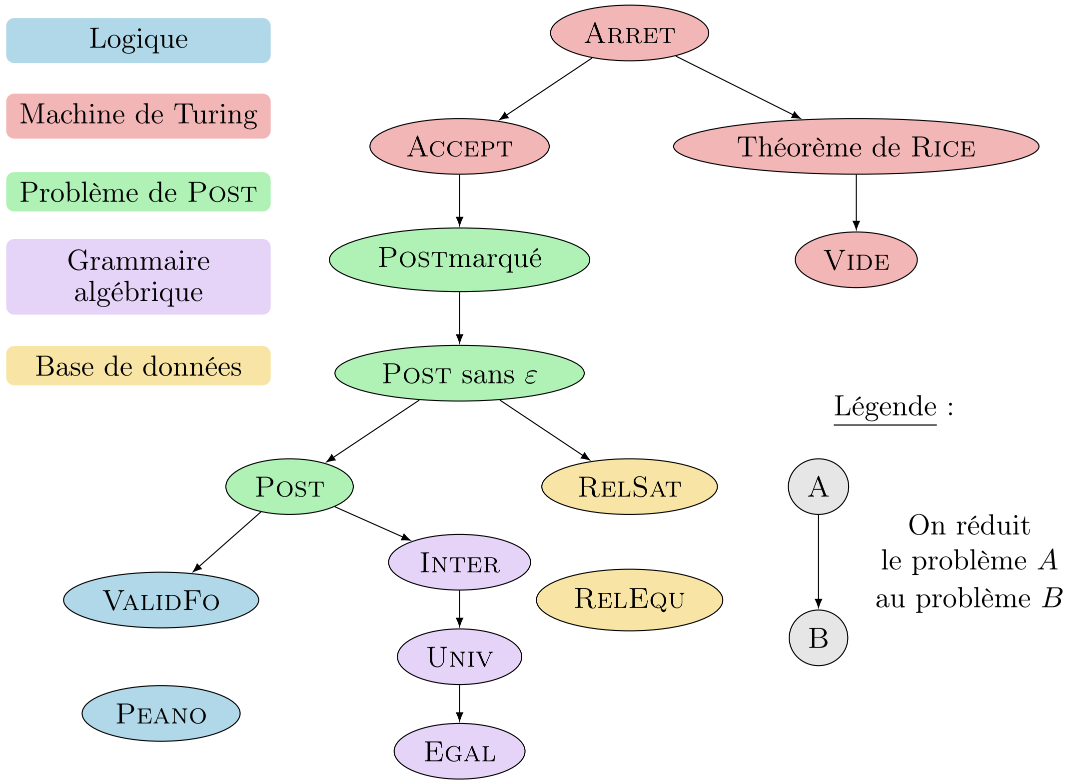
Problèmes indécidables et réduction



Le problème

$\boxed{\textsc{Arret}}$ $\begin{cases} \text{\underline{entrée :} une machine de Turing } M \text{ déterministe, un mot } w \\ \text{\underline{sortie :} oui si } M(w) \text{ s'arrête, non sinon} \end{cases}$

est indécidable (construction d'une machine paradoxale) (voir le polycopié de calculabilité de François Schwarzentruber)

On réduit Arret à

$\boxed{\textsc{Accept}}$ $\begin{cases} \text{\underline{entrée :} une machine de Turing } M\text{, un mot } w \\ \text{\underline{sortie :} oui si } M \text{ accepte } w\text{, non sinon} \end{cases}$

(voir le polycopié de calculabilité de François Schwarzentruber)

On réduit Accept à $\boxed{\textsc{Post}\text{marqué}}$

$\begin{cases} \text{\underline{entrée :} un alphabet fini } \Sigma\text{, un ensemble de tuiles fini } \left(\begin{array}{|c|}\hline u_i \\ \hline v_i \\ \hline \end{array}\right)_{i=1,\dots,n} \text{ avec } u_i, v_i \in \Sigma^* \\ \text{\underline{sortie :} oui s'il existe } i_2, \dots, i_p \in \{1, \dots, n\} \text{ tels que } u_1 u_{i_2} \dots u_{i_p} = v_1 v_{i_2} \dots v_{i_p}\text{, non sinon} \end{cases}$

(voir le polycopié de calculabilité de François Schwarzentruber)

On réduit PostMarqué à $\boxed{\textsc{Post}}$

$\begin{cases} \text{\underline{entrée :} un alphabet fini } \Sigma\text{, un ensemble de tuiles fini } \left(\begin{array}{|c|}\hline u_i \\ \hline v_i \\ \hline \end{array}\right)_{i=1,\dots,n} \text{ avec } u_i, v_i \in \Sigma^* \\ \text{\underline{sortie :} oui s'il existe } i_1, \dots, i_p \in \{1, \dots, n\} \text{ tels que } u_{i_1} u_{i_2} \dots u_{i_p} = v_{i_1} v_{i_2} \dots v_{i_p}\text{, non sinon} \end{cases}$

(voir le polycopié de calculabilité de François Schwarzentruber)

On réduit ARRET à $P_{\mathcal{P}}$ (Théorème de RICE) où $\emptyset \subsetneq \mathcal{P} \subsetneq RE$

$\boxed{P_{\mathcal{P}}}$ $\begin{cases} \underline{\text{entrée :}} \text{ une machine de Turing } M \\ \underline{\text{sortie :}} \text{ oui si } \mathrm{L}(M) \in \mathcal{P}\text{, non sinon} \end{cases}$

(voir le polycopié de calculabilité de François Schwarzentruber)

En prenant $\mathcal{P} = \{\emptyset\}$, on trouve l'indécidabilité de

$\boxed{\text{VIDE}}$ $\begin{cases} \underline{\text{entrée :}} \text{ une machine de Turing } M \\ \underline{\text{sortie :}} \text{ oui si } \mathrm{L}(M) = \emptyset\text{, non sinon} \end{cases}$

(voir le polycopié de calculabilité de François Schwarzentruber)

On réduit POST à

$\boxed{\text{VALIDFO}}$ $\begin{cases} \underline{\text{entrée :}} \text{ une formule close } \varphi \text{ en logique du premier ordre} \\ \underline{\text{sortie :}} \text{ oui si } \varphi \text{ est valide, non sinon} \end{cases}$

(voir Indécidabilité de la validité d'une formule du premier ordre)

On réduit POST à

$\boxed{\text{INTER}}$ $\begin{cases} \underline{\text{entrée :}} \text{ deux grammaires algébriques } G \text{ et } G' \\ \underline{\text{sortie :}} \text{ oui si } \mathrm{L}(G) \cap \mathrm{L}(G') = \emptyset\text{, non sinon} \end{cases}$

(voir Carton [?, p.166])

On le réduit à

$\boxed{\text{UNIV}}$ $\begin{cases} \underline{\text{entrée :}} \text{ une grammaire algébrique } G \\ \underline{\text{sortie :}} \text{ oui si } \mathrm{L}(G) = \Sigma^* \text{ où } \Sigma \text{ est l'alphabet utilisé, non sinon} \end{cases}$

(voir Carton [?, p.166])

On le réduit à

$\boxed{\text{EGAL}}$ $\begin{cases} \underline{\text{entrée :}} \text{ deux grammaires algébriques } G \text{ et } G' \\ \underline{\text{sortie :}} \text{ oui si } \mathrm{L}(G) = \mathrm{L}(G')\text{, non sinon} \end{cases}$

(voir Carton [?, p.166])

On réduit POST à

$\boxed{\text{POST sans } \varepsilon}$ $\begin{cases} \underline{\text{entrée :}} \text{ des tuiles du problème de POST sans tuile contenant des } \varepsilon \\ \underline{\text{sortie :}} \text{ oui si on peut trouver une suite de tuiles ayant le même mot en haut} \\ \qquad\qquad \text{et en bas, non sinon} \end{cases}$

Je le fais plus bas

On réduit POST sans ε à

$\boxed{\text{REL\_SAT}}$ $\begin{cases} \underline{\text{entrée :}} \text{ une requête } q \text{ en calcul relationnel} \\ \underline{\text{sortie :}} \text{ oui si il existe } \mathcal{I} \text{ telle que } q(\mathcal{I}) \neq \emptyset\text{, non sinon} \end{cases}$

(voir Indécidabilité de la satisfiabilité d'une requête)

Le problème

$\boxed{\text{REL\_EQU}}$ $\begin{cases} \underline{\text{entrée :}} \text{ deux requêtes } q \text{ et } q' \text{ du calcul relationnel} \\ \underline{\text{sortie :}} \text{ oui si } q \text{ et } q' \text{ sont sémantiquement équivalentes, non sinon} \end{cases}$

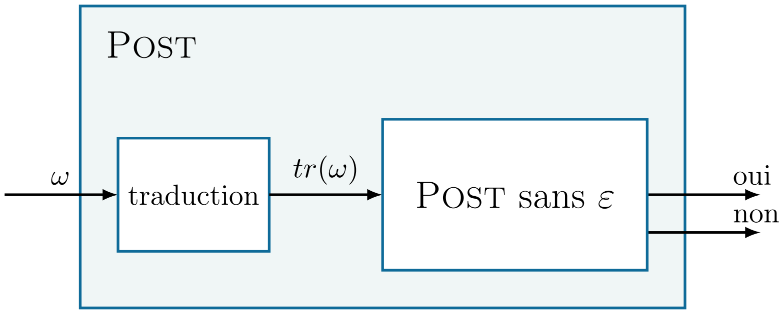
est indécidable. (Je n'ai pas pris le temps de prouver ce résultat)

Le problème

$\boxed{\text{PEANO}} \begin{cases} \underline{\text{entrée :}} \text{ une formule } \varphi \text{ close sur } L = \{0, 1, +, \times, =\} \\ \underline{\text{sortie :}} \text{ oui si } \mathbb{N} \vDash \varphi \text{, non sinon} \end{cases}$

est indécidable. (preuve non triviale, non exigible à l'agrégation)

Réduction de POST à POST sans ε



Soit ω une instance de POST, i.e un nombre fini de tuiles $\begin{array}{|c|}\hline u_1 \\ \hline v_1 \\ \hline\end{array}, \dots, \begin{array}{|c|}\hline u_N \\ \hline v_N \\ \hline\end{array}$ qui peut contenir des ε seulement en haut.

On veut trouver une instance $tr(\omega)$ de POST sans ε qui soit positive SSI ω est positive.

Pour toute tuile $\begin{array}{|c|}\hline \varepsilon \\ \hline a_1 \dots a_k \\ \hline\end{array}$, on va mettre dans $tr(\omega)$ les tuiles :

$$\begin{array}{|c|}\hline \star \\ \hline \star a_1 \dots a_k \\ \hline\end{array}, \begin{array}{|c|}\hline \star \\ \hline a_1 \star a_2 \dots a_k \\ \hline\end{array}, \begin{array}{|c|}\hline \star \\ \hline a_1 a_2 \star a_3 \dots a_k \\ \hline\end{array}, \dots, \begin{array}{|c|}\hline \star \\ \hline a_1 a_2 \dots a_k \star \\ \hline\end{array}$$

Et pour chaque tuile $\begin{array}{|c|}\hline b_1 \dots b_l \\ \hline a_1 \dots a_k \\ \hline\end{array}$, on va mettre dans $tr(\omega)$ les tuiles :

$$\begin{array}{|c|}\hline b_1 \dots b_l \\ \hline \star a_1 \dots a_k \\ \hline\end{array}, \begin{array}{|c|}\hline b_1 \dots b_l \\ \hline a_1 \star a_2 \dots a_k \\ \hline\end{array}, \begin{array}{|c|}\hline b_1 \dots b_l \\ \hline a_1 a_2 \star a_3 \dots a_k \\ \hline\end{array}, \dots, \begin{array}{|c|}\hline b_1 \dots b_l \\ \hline a_1 \dots a_k \star \\ \hline\end{array}$$

et on met aussi la tuile $\begin{array}{|c|}\hline b_1 \dots b_l \\ \hline a_1 \dots a_k \\ \hline\end{array}$ dans $tr(\omega)$.

$\boxed{\Longrightarrow}$ Si $tr(\omega)$ est une instance positive de POST sans ε, on prend une succession de tuiles telle que le mot du haut est identique au mot du bas.

S'il y a une $\star$ en haut, alors il suffit de placer la tuile $\begin{array}{|c|}\hline \varepsilon \\ \hline a_1 \dots a_k \\ \hline\end{array}$ correspondante au bon endroit dans la succession de tuiles de ω. Sinon, il suffit de faire la même succession puisque toutes les tuiles sans $\star$ sont dans ω. Ainsi ω est une instance positive du problème POST.

$$\begin{array}{|c|}\hline a \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline ba \\ \hline a \star b \\ \hline\end{array} \begin{array}{|c|}\hline \star \\ \hline ba \\ \hline\end{array} \begin{array}{|c|}\hline bbaa \\ \hline a \\ \hline\end{array} \text{ devient } \begin{array}{|c|}\hline a \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline ba \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline \varepsilon \\ \hline ba \\ \hline\end{array} \begin{array}{|c|}\hline bbaa \\ \hline a \\ \hline\end{array}$$

$\boxed{\Longleftarrow}$ Si ω est une instance positive de POST, on prend une succession de tuiles telle que le mot du haut est identique au mot du bas. Si on a utilisé une tuile $\begin{array}{|c|}\hline \varepsilon \\ \hline a_1 \dots a_k \\ \hline\end{array}$ alors on la remplace par $\begin{array}{|c|}\hline \star \\ \hline a_1 \dots a_k \\ \hline\end{array}$ et on remplace la tuile où le ε serait en bas, par la même mais avec le $\star$ au bon endroit. Ainsi $tr(\omega)$ est une instance positive de POST sans ε.

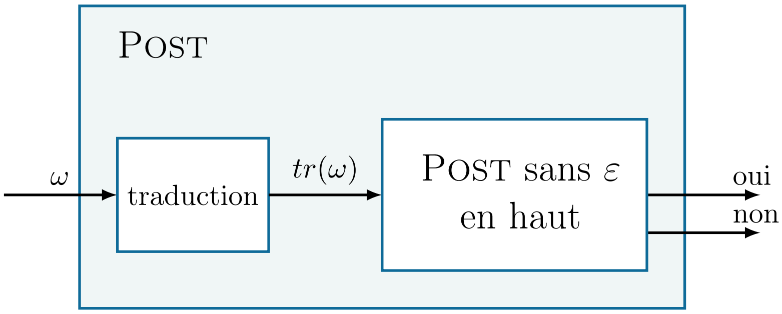
$$\begin{array}{|c|}\hline a \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline ba \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline \varepsilon \\ \hline ba \\ \hline\end{array} \begin{array}{|c|}\hline bbaa \\ \hline a \\ \hline\end{array} \text{ devient } \begin{array}{|c|}\hline a \\ \hline ab \\ \hline\end{array} \begin{array}{|c|}\hline ba \\ \hline a \star b \\ \hline\end{array} \begin{array}{|c|}\hline \star \\ \hline ba \\ \hline\end{array} \begin{array}{|c|}\hline bbaa \\ \hline a \\ \hline\end{array}$$

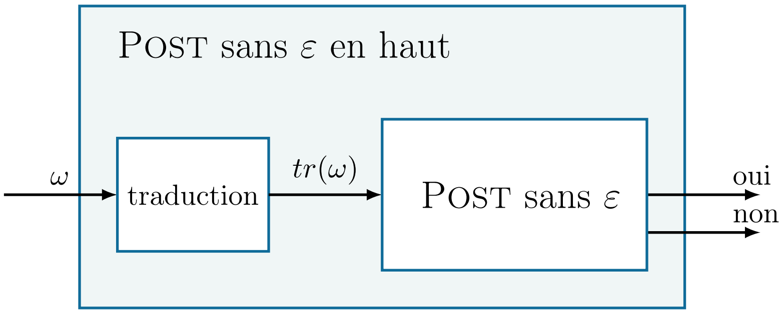
Ensuite si on a des ε en haut et en bas, on peut faire la réduction en deux temps, pour enlever les ε du haut puis ensuite ceux du bas.

Tout cela montre que Post sans ε est indécidable.

Remarques :

En fait, on peut voir ça comme deux réductions successives





Remarques :

En fait, si on regarde plus attentivement la réduction de Accept à Postmarqué et Postmarqué à Post, on observe que l'on a pas de tuiles avec des ε ni en haut ni en bas, donc finalement, on avait déjà l'indécidabilité de Post sans ε, cependant c'est un bon exercice de savoir faire des réductions comme ça à partir de variantes de problème que l'on connait.