Bordeaux INP ENSEIRB MATMECA

Calculabilité & Complexité
Devoir Maison n° 3

22 mai 2021

Rémi Dehenne
Sébastien Delpeuch
Tom Moënne-Loccoz
Aurélien Moinel

Partie 1 : Problème de Post

Q1 Vérifier que l'instance I de (PCP) a la réponse "oui" si et seulement si :

$$\exists\, w \in X^+, \varphi(w) = \psi(w)$$

Nous procédons à une démonstration en deux temps. Montrons d'abord que si une instance I de (PCP) a la réponse "oui" alors : $\exists\, w \in X^+, \varphi(w) = \psi(w)$.

Supposons une instance I de (PCP) de réponse "oui". Cela signifie l'existence d'un entier $k > 0$ et une suite $i_1, ..., i_k$ telle que $u_{i_1}u_{i_2}\cdots u_{i_k} = v_{i_1}v_{i_2}\cdots v_{i_k}$. Intéressons-nous au mot $w = x_{i_1}x_{i_2}\cdots x_{i_k}$ de taille k. Ce mot appartient à X^+ puisqu'il est de longueur non nulle et n'est constitué que de lettres de l'alphabet X. De plus, $\varphi(w) = u_{i_1}u_{i_2}\cdots u_{i_k}$ et $\psi(w) = v_{i_1}v_{i_2}\cdots v_{i_k}$ par définition de φ et ψ. Enfin, par supposition, $u_{i_1}u_{i_2}\cdots u_{i_k} = v_{i_1}v_{i_2}\cdots v_{i_k}$ donc $\varphi(w) = \psi(w)$.

Donc $\exists\, w \in X^+, \varphi(w) = \psi(w)$.

Montrons maintenant que pour une instance I de (PCP) si $\exists\, w \in X^+, \varphi(w) = \psi(w)$ alors l'instance est de réponse "oui".

Supposons $\exists\, w \in X^+, \varphi(w) = \psi(w)$. Intéressons-nous à un mot w tel que $w \in X^+, \varphi(w) = \psi(w)$. Puisque ce mot appartient à X^+, nous pouvons le décomposer en $w = x_{i_1}x_{i_2}\cdots x_{i_k}$ avec $k = |w| > 0$ et $\forall j \in [1,k], i_j \in [1,n]$. Par définition de φ et ψ, $u_{i_1}u_{i_2}\cdots u_{i_k} = \varphi(x_{i_1}x_{i_2}\cdots x_{i_k})$ et $v_{i_1}v_{i_2}\cdots v_{i_k} = \psi(x_{i_1}x_{i_2}\cdots x_{i_k})$ et par supposition, $\varphi(x_{i_1}x_{i_2}\cdots x_{i_k}) = \psi(x_{i_1}x_{i_2}\cdots x_{i_k})$. L'entier k et la suite $i_1, i_2, ..., i_k$ sont donc tels que $u_{i_1}u_{i_2}\cdots u_{i_k} = v_{i_1}v_{i_2}\cdots v_{i_k}$.

L'instance I a donc la réponse "oui".

Ainsi, nous avons montré que si l'instance I de (PCP) a la réponse "oui", alors $\exists\, w \in X^+, \varphi(w) = \psi(w)$, et que si $\exists\, w \in X^+, \varphi(w) = \psi(w)$, l'instance I de (PCP) a la réponse "oui". Par conséquent, l'instance I de (PCP) a la réponse "oui" si et seulement si $\exists\, w \in X^+, \varphi(w) = \psi(w)$.

Q2 Soit $S(I) = \{w \in X^* | \varphi(w) = \psi(w)\}$ (l'ensemble des solutions au PCP, augmenté du mot vide). $S(I)$ est-il clos par produit ? par facteur ? S'agit-il d'un sous-monoïde de $X^\star$?

Clôture par produit

Nous cherchons à montrer que $S(I)$ est clos par produit.

Soient $w_1,\ w_2 \in S(I)$. Par définition, nous avons $\varphi(w_1) = \psi(w_1)$ et $\varphi(w_2) = \psi(w_2)$ et :

$$\begin{aligned}\varphi(w_1 \cdot w_2) &= \varphi(w_1) \cdot \varphi(w_2) \text{ car } \varphi \text{ est un homomorphisme de } X^\star \text{ dans } \Sigma^\star \\ &= \psi(w_1) \cdot \psi(w_2) \\ &= \psi(w_1 \cdot w_2) \text{ car } \psi \text{ est un homomorphisme de } X^\star \text{ dans } \Sigma^\star\end{aligned}$$

Par conséquent, $\forall w_1, w_2 \in S(I)$, $w_1 \cdot w_2 \in S(I)$, donc $S(I)$ est clos par produit.

Clôture par facteur

Nous prenons l'instance $I = ((a, aa), (aa, a))$. Le mot $w = x_1 \cdot x_2$ est solution au problème de correspondance de Post :

$$\varphi(x_1x_2) = aaa = \psi(x_1x_2)$$

Nous avons donc $w \in S(I)$. Cependant, $\varphi(x_1) = a$ et $\psi(x_1) = aa$, donc $\varphi(x_1) \neq \psi(x_1)$ et $x_1 \notin S(I)$. Par conséquent, $S(I)$ n'est pas clos par facteur.

Sous-monoïde de $X^\star$

$S(I) = \{w \in X^* |\ \varphi(w) = \psi(w)\}$ est un sous-monoïde de $X^\star$ si et seulement si $X^\star$ et $S(I)$ sont des monoïdes, et que $S(I) \subseteq X^*$.

D'une part, X^* est un monoïde par définition de l'étoile de Kleene : son produit $\cdot$ est une loi de composition interne associative, et ε est l'élément neutre de cette loi ($\forall w \in X^*,\ w \cdot \varepsilon = \varepsilon \cdot w = w$).

D'autre part, nous montrons que $S(I)$ est un sous-monoïde de X^*. Nous rappelons que $S(I)$ est un sous-monoïde du monoïde X^* si et seulement si $S(I) \subseteq X^*$, que le produit $\cdot$ est une loi de composition interne dans $S(I)$ et que l'élément neutre du produit ε appartient à $S(I)$. Par définition, $S(I) \subseteq X^*$. Nous avons aussi montré que $S(I)$ est clos par produit, donc le produit est une loi de composition interne dans $S(I)$. Enfin, l'élément neutre ε appartient à $S(I)$ car $\varepsilon \in X^*$ et $\varphi(\varepsilon) = \psi(\varepsilon) = \varepsilon$, puisque φ et ψ sont des homomorphismes.

Ainsi, le produit est une loi de composition interne dans $S(I)$, qui dispose d'un élément neutre ε. Enfin, $S(I) \subseteq X^*$ et X^* est un monoïde, donc $S(I)$ est un sous-monoïde de $X^\star$.

Q3 Montrer que $S(I) = A(I)^\star$.

Nous cherchons à montrer que $S(I) = A(I)^\star$; procédons par double inclusion.

D'une part, montrons que $A(I)^\star \subseteq S(I)$. Soit $w \in A(I)^\star$. Par définition, nous avons la décomposition suivante : $w = w_1 \cdot w_2 \cdot ... \cdot w_n$ avec $\forall i \in [0, n], w_i \in A(I)$. Par définition de $A(I)$, $\forall i \in [0, n], w_i \in S(I)$. Ainsi, w est le produit fini d'éléments de $S(I)$. Comme $S(I)$ est clos par produit, nous avons que $w \in S(I)$. Cela implique $A(I)^\star \subseteq S(I)$.

D'autre part, montrons que $S(I) \subseteq A(I)^\star$. Soit $w \in S(I)$; réalisons une disjonction de cas :

— Soit il existe une décomposition $w = w_1 \cdot w_2$ avec $w_1, w_2 \in S(I)$ tels que $w_1 = \varepsilon$ ou $w_2 = \varepsilon$. Dans ce cas, par définition, $w \in A(I)^\star$.

— Soit $w = w_1 \cdot w_2$ avec $w_1, w_2 \in S(I)$ tels que $w_1 \neq \varepsilon$ et $w_2 \neq \varepsilon$. Dans ce cas, nous réitérons la disjonction avec w_1 et w_2 jusqu'à avoir un nombre fini de solutions atomiques. Cette décomposition ne sera faisable qu'un nombre fini de fois car la longueur de w est entière, les sous-mots ont une longueur strictement inférieure aux mots dont ils sont extraits et $\mathbb{N}$ est bien fondé. *In fine* w est une concaténation d'un nombre de solutions atomiques et donc $w \in A(I)^\star \Rightarrow S(I) \subseteq A(I)^\star$.

En somme, nous avons montré que $A(I)^\star \subseteq S(I)$ et que $S(I) \subseteq A(I)^\star$, donc $S(I) = A(I)^\star$.

Q4 Résoudre le problème PCP sur les instances suivantes :

$$I_1 = ((bbb, bb), (abb, babbb))$$
$$I_2 = ((ba, bab), (abb, bb), (bab, abb))$$
$$I_3 = ((\#\#x, \#\#xaxaxbxbxbx\#), (\#xbxbxbxaxax\#\#, x\#\#), (ax, xa), (bx, xb), (\#x, x\#), (axbx, xbxa))$$

$I_1 = ((bbb, bb), (abb, babbb))$

Il existe une solution en le mot $w = x_1x_2x_1$:

$$\begin{pmatrix} \varphi(w) \\ \psi(w) \end{pmatrix} = \begin{pmatrix} \overbrace{bbb}^{\varphi(x_{i_1})} & \overbrace{abb}^{\varphi(x_{i_2})} & \overbrace{bbb}^{\varphi(x_{i_3})} \\ \underbrace{bb}_{\psi(x_{i_1})} & \underbrace{babbb}_{\psi(x_{i_2})} & \underbrace{bb}_{\psi(x_{i_3})} \end{pmatrix}$$

$I_2 = ((ba, bab), (abb, bb), (bab, abb))$

Par l'absurde, on suppose $\exists\ w \in X^+, \varphi(w) = \psi(w)$. En particulier, w est tel que $|\varphi(w)| = |\psi(w)|$.

On a que :

$$|\varphi(w)| = 2 \times |w|_{x_1} + 3 \times |w|_{x_2} + 3 \times |w|_{x_3}$$
$$|\psi(w)| = 3 \times |w|_{x_1} + 2 \times |w|_{x_2} + 3 \times |w|_{x_3}$$
$$|\varphi(w)| = |\psi(w)| \Rightarrow |w|_{x_1} = |w|_{x_2} (E_1)$$

En particulier,

$$|\varphi(w)|_a = |\psi(w)|_a$$
$$|\varphi(w)|_a = |w|_{x_1} + 2 \times |w|_{x_2} + 2 \times |w|_{x_3}$$
$$|\psi(w)|_a = 2 \times |w|_{x_1} + 2 \times |w|_{w_2} + 2 \times |w|_{w_3}$$
$$|\varphi(w)|_a = |\psi(w)|_a \Rightarrow |w|_{x_1} = 0 \Rightarrow^{(E_1)} |w|_{x_2} = 0$$

Donc, $\exists\ k \in \mathbb{N}^* \mid w = x_3^k$.

On définit l'opérateur ${}^{(1)}u$ sur $\Sigma^\star \to (\Sigma \cup \varepsilon)$ tel que :

$${}^{(1)}u = \begin{cases} x \text{ si } u = xu' \\ \varepsilon \text{ si } u = \varepsilon \end{cases}$$

${}^{(1)}\varphi(w) = b$, ${}^{(1)}\psi(w) = a \Rightarrow \varphi(w) \neq \psi(w)$: c'est absurde, donc I_2 n'a pas de solution.

$$I_3 = ((\#\#x, \#\#xaxaxbxbxbx\#), (\#xbxbxbxaxax\#\#, x\#\#), (ax, xa), (bx, xb), (\#x, x\#), (axbx, xbxa))$$

Il existe une solution en le mot $w = x_1x_3x_6x_4x_4x_5x_6x_6x_4x_5x_4x_6x_6x_5x_4x_4x_6x_3x_2$. En effet, soit :

$$\begin{pmatrix} \varphi(w) \\ \psi(w) \end{pmatrix} = \left(\begin{array}{c} \overbrace{\#\#\text{x}}^{\varphi(x_{i_1})} \\ \underbrace{\#\#\text{xaxaxbxbxbx}\#}_{\psi(x_{i_1})} \end{array} \begin{array}{c} \overbrace{\text{ax}}^{\varphi(x_{i_2})} \\ \underbrace{\text{xa}}_{\psi(x_{i_2})} \end{array} \begin{array}{c} axbx\ bx\ bx\ \#x\ axbx\ axbx\ bx\ \#x\ bx\ axbx\ axbx\ \#x\ bx\ bx\ axbx\ ax \\ xbxa\ xb\ xb\ x\#\ xbxa\ xbxa\ xb\ x\#\ xb\ xbxa\ xbxa\ \#x\ xb\ xb\ xbxa\ xa \end{array} \begin{array}{c} \overbrace{\#\text{xbxbxbxaxax}\#\#}^{\varphi(x_{i_{19}})} \\ \underbrace{\text{x}\#\#}_{\psi(x_{i_{19}})} \end{array} \right)$$

Partie 2 : Systèmes semi-Thuéiens

Q1 Montrer que pour les systèmes S qui conservent la longueur (i.e. $\forall (l, r) \in S,\ |l| = |r|$), le problème (ACC-ST) est *décidable*.

De façon naïve, nous souhaiterions construire et visiter les sommets du graphe orienté $G = (V, E)$ issu des dérivations successives en partant de u, à travers le système S et en cherchant v.

Le graphe est construit par itérations en partant d'un sommet w accessible depuis u (tel que $u \to_S^* w$) et en formant des arêtes entre ce sommet w et tous les sommets atteignables W' depuis w par dérivation immédiate ($u \to_S^* w \wedge (\forall w' \in W', w \to_S w') \Rightarrow u \to_S^* w'$). La construction du graphe débute au sommet u ce qui implique qu'il existe un chemin entre u et tous les sommets du graphe.

Puisque le système S est fini, trouver les dérivations immédiates d'un mot $w \in A^*$ peut être réalisé en temps fini.

Un tel graphe garantit l'équivalence suivante :

$$\forall w \in A^*, u \to_S^* w \Leftrightarrow w \in V$$

En effet,

$$\forall w \in A^*, u \to_S^* w \Rightarrow \exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = w) \in (A^*)^n, \forall i \in [0, n-1], u_i \to_S u_{i+1}$$

Or, par application systématique de la règle de construction,

$$\Rightarrow ((u_i \to_S u_{i+1} \wedge u_i \in V \Rightarrow u_{i+1} \in V) \wedge u_0 = u \in V)$$

$$\Rightarrow \forall i \in [0, n], u_i \in V.$$

En particulier $w \in V$.

A l'inverse, puisqu'il existe un chemin entre u et tous les sommets du graphe,

$$\forall w \in V, \exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = w) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E$$

Et par construction,

$$\forall (s, s') \in E, s \to_S s'$$

Donc,

$$\exists D = (u = u_0, u_1, ..., u_n = w) \in (A^*)^n, \forall i \in [0, n-1], (u_i \to_S u_{i+1}) \Rightarrow \forall i \in [0, n], u \to_S^* u_i$$

En particulier, $u \to_S^* w$.

Donc nous avons bien $\forall w \in A^*, u \to_S^* w \Leftrightarrow w \in V$. En particulier, $u \to_S^* v \Leftrightarrow v \in V$.

Jusque-là, nous avons étudié la formation du graphe sans nous intéresser au fait que le système S préserve la longueur. Cela implique notamment que ce qui a été démontré pourra resservir dans les questions suivantes. Cependant, tel quel, nous ne pouvons pas garanti que le parcours d'un tel graphe à la recherche de v permettra de décider de $u \to_S^* v$. En effet, le graphe ainsi produit pourrait être infini : $|V| = \infty$.

Le système S est tel que $\forall (l, r) \in S, |l| = |r|$. Cela implique que,

$$\forall\ w, w' \in A^*,\ w \to_S w'$$

$$\Rightarrow \exists (l, r) \in S, \exists\ \alpha, \beta \in A^*, w = \alpha \cdot l \cdot \beta \wedge w' = \alpha \cdot r \cdot \beta \wedge |l| = |r|$$

$$\Rightarrow \exists (l, r) \in S, \exists\ \alpha, \beta \in A^*, |w| = |\alpha| + |l| + |\beta| = |\alpha| + |r| + |\beta| = |w'|$$

$$\Rightarrow |w| = |w'|$$

On a donc $\forall w, w' \in A^*, w \to_S w' \Rightarrow |w| = |w'|$. Par induction immédiate, cela implique à son tour que $\forall w, w' \in A^*, w \to_S^* w' \Rightarrow |w| = |w'|$.

Puisque nous avons déjà démontré que $\forall w \in A^*, w \in V \Leftrightarrow u \to_S^* w$. Nous avons que $\forall w \in V, |w| = |u|$. Le nombre de sommets de ce graphe est donc fini, car inférieur au nombre fini de mots de taille $|u|$, $|A|^{|u|}$.

Il s'agit donc de créer un graphe fini, dont la production de chaque sommet et de ses arrêtes a lieu en temps fini, puis de le visiter. Cela prendra donc un temps fini. De plus, nous avons prouvé l'équivalence entre $u \to_S^* v$ et l'existence d'un chemin entre u et v au sein du graphe. Donc puisque nous avons montré que l'algorithme de parcours et construction termine correctement, nous sommes garantis qu'il trouvera v ssi $u \to_S^* v$. La machine de Turing qui met en place cet algorithme et retourne "oui" si le mot v est trouvé et "non" sinon décide donc du problème (ACC-ST) contraint aux systèmes qui conservent la longueur.

En particulier, voici un algorithme écrit en pseudo-code qui décide de $u \to_S^* v$. Nous avons vu en cours que l'expressivité d'une machine de Turing est supérieure à un tel algorithme et il est donc nécessairement possible de créer une machine de Turing qui le simule.

```
Données : u ∈ A*, v ∈ A*, S système semi-Thuéien fini
si longueur(u) ≠ longueur(v) alors
 | retourner Faux

mémoire ← Ensemble()
pile ← Pile()
pile.empiler(u)

tant que (non estVide(pile)) faire
 | w ← pile.dépiler()
 | si w = v alors retourner Vrai
 | si mémoire.contient(w) alors passer
 | mémoire.empiler(w)
 | pour (l, r) ∈ S faire
 | | motsDérivés ← dériverImmédiatement(u, l, r)
 | | pour w' ∈ motsDérivés faire
 | | | pile.empiler(w')

retourner Faux
```

Algorithme 1 : Parcours du graphe des dérivations pour un système semi-Thuéien qui conserve la longueur

Q2 Supposons qu'il existe un ordre total $\sqsubseteq$ sur les mots de A^*, qui est compatible avec le produit i.e. tel que

$$\forall \alpha, u, v, \beta \in A^*, u \sqsubseteq v \Rightarrow \alpha \cdot u \cdot \beta \sqsubseteq \alpha \cdot v \cdot \beta$$

et tel que les ensembles ordonnés $(\mathbb{N}, \leq)$ et $(A^*, \sqsubseteq)$ soient isomorphes. un système S est dit *croissant* pour l'ordre $\sqsubseteq$ si, $\forall (l, r) \in S$, $l \sqsubseteq r$. Montrer que pour les systèmes S croissants pour un ordre vérifiant les hypothèses ci-dessus, le problème (ACC-ST) est décidable.

Nous souhaiterions réimplémenter un algorithme dont le but serait de construire et inspecter le graphe $G = (V, E)$ exhaustif des dérivations de u, à travers le système S, à la recherche du mot v.

Cependant, il faut montrer que ces nouvelles conditions permettent elles aussi de restreindre ce graphe à un ensemble fini. Sinon, rien ne garantirait la terminaison de notre Machine de Turing.

Nous pouvons substituer à ce graphe le sous-graphe $G' = (V', E')$ constitué exclusivement des sommets représentant un mot $w \in A^*$ tel que $w \sqsubseteq v$ et préservant les arêtes entre les sommets restants.

Nous pouvons démontrer deux propriétés par rapport à un tel graphe :

$$(u \to_S^* v) \iff (\exists\ n \in \mathbb{N}, \exists\ D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E')$$

Autrement dit, il y a équivalence entre la dérivation de u en v par S et l'existence d'un chemin entre u et v dans G'. Et,

$$|V'| < \infty$$

Pour montrer la première propriété, démontrons d'abord que $\forall w \in A^*, v \sqsubseteq w \wedge v \neq w \Rightarrow w \not\to_S^* v$. En effet,

$$\forall (l, r) \in S,$$

$$(l \sqsubseteq r) \Rightarrow (\forall \alpha, \beta \in A^*, \alpha \cdot l \cdot \beta \sqsubseteq \alpha \cdot r \cdot \beta)$$
$$\Rightarrow (\forall w, w' \in A^*, w \to_S w' \Rightarrow w \sqsubseteq w')$$
$$\Rightarrow (\forall w, w' \in A^*, w \to_S^* w' \Rightarrow w \sqsubseteq w')$$
$$\Rightarrow (\forall w, w' \in A^*, w \not\sqsubseteq w' \Rightarrow w \not\to w')$$
$$\Rightarrow (\forall w, w' \in A^*, w \neq w \wedge w' \sqsubseteq w \Rightarrow w \not\to w')$$
$$\Rightarrow (\forall w \in A^*, w \neq v \wedge v \sqsubseteq w' \Rightarrow w \not\to_S^* v)$$

Par contraposée, on a $\forall w \in A^*, w \to_S^* v \Rightarrow w \sqsubseteq v$.

Ensuite, si $u \to_S^* v$ alors $\exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], u_i \to_S u_{i+1}$. En particulier, $\forall i \in [0, n], u \to_S^* u_i \wedge u_i \to_S^* v$. Ainsi,

$$\forall i \in [0, n], u \to_S^* u_i \Rightarrow u_i \in G \wedge (\forall j \in [0, n-1], ((u_i, u_{i+1}) \in E)$$

comme vu dans l'exercice précédent.

De plus,

$$\forall i \in [0, n], u_i \to_S^* v, u_i \sqsubseteq v \Rightarrow u_i \notin V \setminus V' \wedge (\forall j \in [0, n-1], (u_i, u_{i+1}) \notin E \setminus E'$$

.

Ensemble, cela implique que

$$\forall i \in [0, n], u_i \in V' \wedge \forall j \in [0, n-1], (u_i, u_{i+1}) \in E'$$

On a donc bien,

$$(u \to_S^* v) \Longrightarrow (\exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E')$$

Au contraire, si $u \not\to_S^* v$ alors $v \notin V$ et comme $V' \in V$, $v \notin V'$. Donc $u \to_S^* v \Leftrightarrow v \in V'$. Donc il est impossible que $\exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E'$.

Nous avons donc montré qu'il y a équivalence entre $u \to_S^* v$ et l'existence d'un chemin dans G' de u vers v.

Pour montrer la seconde propriété, nous rappelons qu'il existe un isomorphisme entre $(\mathbb{N}, \leq)$ et $(A^*, \sqsubseteq)$. Il existe donc une bijection $\phi : A^* \to \mathbb{N}$ tel que $\forall w, w' \in A^*, w \sqsubseteq w' \Leftrightarrow \phi(w) \leq \phi(w')$. En particulier, toute suite strictement croissante $w_1, w_2, w_3, ..., w_n, \forall n \in \mathbb{N}$ de A^* est telle que $n \leq \phi(w_1) - \phi(w_n) + 1$ (si $w_1 \sqsubseteq w_n$ sinon une telle suite ne peut pas exister). En effet, puisque la suite est strictement croissante dans $(A^*, \sqsubseteq)$, nous avons que $\phi(w_{i+1}) - \phi(w_i) \geq 1$ donc $\phi(w_n) - \phi(w_1) = \sum_{i=1}^{n-1}(w_{i+1} - w_i) \geq \sum_{i=1}^{n-1} 1 = n - 1$. En particulier, puisque tous les sommets du graphe G' représentent des mots distincts w et que pour chacun $w \leq v$, nous pouvons mettre ces mots dans une suite strictement croissante. Nous avons alors que la taille de cette suite est inférieure ou égale à $\phi(v) - \phi(u) + 1$. $\phi(v)$ et $\phi(u)$ représentant des nombres finis, il advient que le cardinal de V' est fini.

Nous avons donc démontré que le sous-graphe G' conserve un chemin vers v et contient un nombre fini de sommets. Ce graphe est constructible en tentant de construire le graphe G mais en refusant les sommets

$w, v \sqsubseteq w \wedge w \neq v$. Comme le graphe G, la production de chaque sommet et de ses arêtes peut donc avoir lieu en temps fini.

Une machine de Turing qui construit et parcourt le graphe G' termine donc forcément en ayant trouvé v si $u \to_S^* v$. Elle peut donc décider du problème (ACC-ST) contraint aux systèmes croissants dans $(A^*, \sqsubseteq)$.

Suit un algorithme en pseudo-code qui décide de $u \to_S^* v$.

```
Données : u ∈ A*, v ∈ A*, S système semi-Thuéien fini
mémoire ← Ensemble()
pile ← Pile()
pile.empiler(u)

tant que (non estVide(pile)) faire
    w ← pile.dépiler()
    si w = v alors retourner Vrai
    si v ⊑ w alors passer
    si mémoire.contient(w) alors passer
    mémoire.empiler(w)
    pour (l, r) ∈ S faire
        motsDérivés ← dériverImmédiatement(u, l, r)
        pour w' ∈ motsDérivés faire
            pile.empiler(w')

retourner Faux
```

Algorithme 2 : Parcours du graphe des dérivations pour un système semi-Thuéien croissant dans $(A^*, \sqsubseteq)$

Q3 En déduire que le problème (ACC-ST) est décidable pour les systèmes S qui n'ont qu'une règle.

Nous pouvons séparer ce problème en deux cas complémentaires.

Si l'unique règle (l, r) de S est telle que $|l| = |r|$, S est un système qui conserve la longueur tel qu'étudié dans la question 1. Nous avons déjà montré que pour un tel système, le problème (ACC-ST) est décidable.

Sinon nous avons nécessairement que $|l| \neq |r|$.

Si l'unique règle est telle que $|l| < |r|$, alors le système entier est croissant pour les longueurs. Dans ce cas, nous pouvons proposer une machine de Turing inspirée de celle obtenue à la question précédente. En effet, l'ordre sur les tailles n'est pas total donc on ne peut pas directement réemployer la solution de la question précédente. Cependant, nous conservons la propriété $\forall w \in A^*, (|v| \leq |w| \wedge w \neq v) \Rightarrow w \not\to_S^* v$.

Cette propriété nous assure que si l'on contraint le graphe $G = (V, E)$ des mots obtenus par dérivation depuis u à la recherche de v à travers S au graphe G' où seuls sont conservés les mots $w \in A^*, |w| \leq |v|$, nous préservons l'équivalence

$$u \to_S^* v \iff (\exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E')$$

De plus, nous pouvons montrer facilement montrer que le graphe G' est fini. En effet, tous les sommets w de V' sont tels que $|w| \leq |v|$ par définition, donc le nombre de sommets de V' est inférieur au nombre de mots de A^* de taille inférieure ou égale à $|v|$. Ce nombre est fini et égal à $\Sigma_{i=0}^{|v|} |A|^i = \frac{1-|A|^{|v|+1}}{1-|A|} < \infty$ si $|A| \neq 1$ et $|v| + 1 < \infty$ sinon.

Les conditions sur G' sont donc réunies pour mettre en place un algorithme de construction et de parcours de G' qui garantit de trouver le sommet v en temps fini si $u \to_S^* v$ et de terminer sans le trouver sinon. Nous pouvons donc implémenter cet algorithme dans une machine de Turing et cette dernière décidera du problème.

Enfin, si l'unique règle est telle que $|l| > |r|$, alors le système entier est décroissant pour les longueurs. Nous pouvons alors nous intéresser au graphe G'' où seuls les sommets w de G tels que $|w| \geq |v|$ sont conservés. Encore une fois, nous avons la propriété $\forall w \in A^*, (|v| \geq |w| \wedge w \neq v) \Rightarrow w \not\to_S^* v$ qui garantit l'équivalence

$$u \to_S^* v \iff (\exists n \in \mathbb{N}, \exists D = (u = u_0, u_1, ..., u_n = v) \in (A^*)^n, \forall i \in [0, n-1], (u_i, u_{i+1}) \in E'')$$

De plus, $\forall w, w' \in A^*, w \to_S^* w' \Rightarrow |w| \geq |w'|$ donc puisque le graphe G'' est construit par dérivation de u, $\forall w \in V'', |u| \geq |w|$. La taille de G'' est donc fini car $|V''| \leq \frac{1-|A|^{|u|+1}}{1-|A|} < \infty$ si $|A| \neq 1$ et $|u| + 1$ sinon.

Les conditions sur G'' sont ici encore réunies pour mettre en place un algorithme de construction et de parcours de G'' qui garantit de trouver le sommet v en temps fini si $u \to_S^* v$, et de terminer sans le trouver sinon. Nous pouvons donc implémenter cet algorithme dans une machine de Turing et cette dernière décidera du problème.

Ainsi, dans tous les cas, le problème est décidable donc le problème entier est décidable.

Q4 Résoudre le problème (ACC-ST) sur les instances suivantes :

$$\begin{aligned}
S_1 =&\{(ab, ba)\},\ u_1 = aabbb,\ v_1 = bbbaa\\
S_2 =&\{(ab, ba), (ab, aab), (aa, ba)\},\ u_2 = aab,\ v_2 = ababa\\
S_3 =&\{(x0, x1), (x1, x0)\} \cup \{(x0^n10, x0^n11) | 0 \leq n \leq 5\} \cup \{(x0^n11, x0^n10) | 0 \leq n \leq 5\},\\
&u_3 = x1111111,\ v_3 = x0000000\\
S_4 =&\{(aba, \varepsilon), (\varepsilon, aba)\}, u_4 = a^{4042}b^{2021}, v_4 = (ab)^{2021}a^{2021}
\end{aligned}$$

Soit $S_1 = \{(ab, ba)\},\ u_1 = aabbb,\ v_1 = bbbaa$, on obtient :

$$\begin{aligned}
u_1 =& \text{a}\mathbf{ab}\text{bb}\\
&\mathbf{ab}\text{abb}\\
&\text{ba}\mathbf{ab}\text{b}\\
&\text{b}\mathbf{ab}\text{ab}\\
&\text{bba}\mathbf{ab}\\
&\text{bb}\mathbf{ab}\text{a}\\
&\text{bbbaa} = v_1
\end{aligned}$$

Soit $S_2 = \{(ab, ba), (ab, aab), (aa, ba)\},\ u_2 = aab,\ v_2 = ababa$, on obtient :

$$\begin{aligned}
u_2 =& \text{a}\mathbf{ab}\\
&\mathbf{ab}\text{a}\\
&\text{a}\mathbf{ab}\text{a}\\
&\text{a}\mathbf{aa}\text{ba}\\
&\text{ababa} = v_2
\end{aligned}$$

Soit $S_3 = \{(x0, x1), (x1, x0)\} \cup \{(x0^n10, x0^n11) | 0 \leq n \leq 5\} \cup \{(x0^n11, x0^n10) | 0 \leq n \leq 5\}, u_3 = x1111111,\ v_3 = x0000000$, on obtient :

Soit $S_4 = \{(aba, \varepsilon), (\varepsilon, aba)\}, u_4 = a^{4042}b^{2021}, v_4 = (ab)^{2021}a^{2021}$

$$\begin{aligned}
u_4 =& a^{4042}b^{2021} && (1)\\
&a^{4041}abb^{2020} && (2)\\
&a^{4041}\underbrace{ab\mathbf{a}}\,\mathbf{ba}b^{2020} && (3)\\
&a^{4041}bab^{2020} && (4)\\
&a^{4040}\underbrace{aba}\,b^{2020} && (5)\\
&a^{4040}b^{2020} && (6)
\end{aligned}$$

Entre l'étape 1 et l'étape 6, on a a^{n-2} et b^{n-1}. On répète donc l'opération 2020 fois afin d'obtenir : ε.

$$\begin{aligned}
&\varepsilon && (7)\\
&\mathbf{aba} && (8)\\
&ab\mathbf{aba}a && (9)\\
&ab^2a^2 && (10)
\end{aligned}$$

Ainsi, on répète cette opération 2020 fois afin d'obtenir $(ab)^{2021}a^{2021}$

$u_3 = x1111111$
$x0111111$
$x0101111$
$x1101111$
$x1001111$
$x0001111$
$x0001011$
$x1001011$
$x1101011$
$x0101011$
$x0111011$
$x1111011$
$x1011011$
$x0011011$
$x0010011$
$x1010011$
$x1110011$
$x0110011$
$x0100011$
$x1100011$
$x1000011$
$x0000011$
$x0000010$
$x1000010$
$x1100010$
$x0100010$
$x0110010$
$x1110010$
$x1010010$
$x0010010$
$x0011010$
$x1011010$
$x1111010$
$x0111010$
$x0101010$
$x1101010$
$x1001010$
$x0001010$
$x0001110$
$x1001110$
$x1101110$
$x0101110$
$x0111110$
$x1111110$
$x1011110$
$x0011110$
$x0010110$
$x1010110$
$x1110110$
$x0110110$
$x0100110$
$x1100110$
$x1000110$
$x0000110$
$x0000100$
$x1000100$
$x1100100$
$x0100100$
$x0110100$
$x1110100$
$x1010100$
$x0010100$
$x0011100$
$x1011100$
$x1111100$
$x0111100$
$x0101100$
$x1101100$
$x1001100$
$x0001100$
$x0001000$
$x1001000$
$x1101000$
$x0101000$
$x0111000$
$x1111000$
$x1011000$
$x0011000$
$x0010000$
$x1010000$
$x1110000$
$x0110000$
$x0100000$
$x1100000$
$x1000000$
$x0000000 = v_3$

Partie 3 : Machines de Turing versus Systèmes semi-Thuéiens

Q1 L'application C est-elle injective ? surjective ?

$$\exists \Sigma = \{a\} \cup \{\triangleright, \square\},\ \exists d_1, d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 = a,\ d_2 = a\square,\ d_1 \neq d_2,\ \mathbf{C}(d_1) = a\square^\infty = \mathbf{C}(d_2)$$

Donc $\exists d_1, d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \neq d_2 \wedge \mathbf{C}(d_1) = \mathbf{C}(d_2)$, la fonction **C** n'est pas injective de **CI** sur **CONF**.

$$\forall c \in \mathbf{CONF}(\mathcal{M}),\ \exists u, v \in \Sigma^*,\ \exists q \in Q,\ (u \cdot v) \in \triangleright \cdot (\Sigma \setminus \{\triangleright\}),\ c = uqv\square^\infty,$$
$$\exists d \in \mathbf{DI}(\mathcal{M}),\ d = uqv,\ \mathbf{C}(d) = uqv\square^\infty = c$$

Donc $\forall c \in \mathbf{CONF}(\mathcal{M}),\ \exists d \in \mathbf{DI}(\mathcal{M}),\ \mathbf{C}(d) = c$. **C** est donc une fonction surjective de $\mathbf{DI} \to \mathbf{CONF}$.

Q2 Donner un système semi-Thuéien $S_\mathcal{M}$ sur Σ^* tel que, pour tous $d_1, d_2 \in \mathrm{DI}(\mathcal{M})$ on ait :

$$d_1 \to_{S_\mathcal{M}}^* d_2 \Rightarrow \mathbf{C}(d_1) \vdash_\mathcal{M}^* \mathbf{C}(d_2)$$

et pour tous c_1, $c_2 \in \mathbf{CONF}(\mathcal{M})$ et $d_1 \in \mathbf{DI}(\mathcal{M})$ on ait :

$$(\mathbf{C}(d_1) = c_1 \text{ et } c_1 \vdash_\mathcal{M}^* c_2) \Rightarrow (\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \to_{S_\mathcal{M}}^* d_2 \text{ et } \mathbf{C}(d_2) = c_2)$$

Sous l'hypothèse que $|\Sigma| < \infty$, le système $S_\mathcal{M}$ est défini par les trois propositions suivantes.

$$\forall (q, x, q', y, R) \in \delta,\ (qx, yq') \in S_\mathcal{M}$$

$$\forall (q,x,q',y,L) \in \delta,\ \forall \mu \in \Sigma,\ (\mu q x, q' \mu y) \in S_{\mathcal{M}}$$

$$\forall (l,r) \in S_{\mathcal{M}},\ \exists\ q,\ q' \in Q,\ \exists\ x,y \in \Sigma^*,$$
$$(l = qx \wedge r = yq' \wedge (q,x,q',y,R) \in \delta) \vee (\exists\ \mu \in \Sigma,\ l = \mu q x \wedge r = q' \mu y \wedge (q,x,q',y,L) \in \delta)$$

Démontrons maintenant que les propositions demandées sont respectées par un tel système.

D'abord, montrons que :

$$\forall d_1, d_2 \in DI(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}^* d_2 \Rightarrow \mathbf{C}(d_1) \vdash_{\mathcal{M}}^*\ \mathbf{C}(d_2)$$

Cela s'obtient immédiatement par induction si nous avons :

$$\forall d_1, d_2 \in DI(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \Rightarrow \mathbf{C}(d_1) \vdash_{\mathcal{M}}\ \mathbf{C}(d_2)$$

Or,

$$\forall d_1, d_2 \in DI(\mathcal{M}),$$
$$d_1 \rightarrow_{S_{\mathcal{M}}} d_2$$
$$\Rightarrow \exists (l,r) \in S_{\mathcal{M}},\ \exists \alpha, \beta \in \Sigma^*,\ d_1 = \alpha l \beta,\ d_2 = \alpha r \beta$$
$$\Rightarrow \exists (l,r) \in S_{\mathcal{M}},\ \exists \alpha,\beta,x,y \in \Sigma^*,\ \exists q_1,q_2 \in Q, (l = q_1 x,\ r = y q_2,\ d_1 = \alpha q_1 x \beta,\ d_2 = \alpha y q_2 \beta \wedge \exists (q_1,x,q_2,y,R) \in \delta)$$
$$\vee (\exists \mu \in \Sigma,\ l = \mu q_1 x,\ r = q_2 \mu y,\ d_1 = \alpha \mu q_1 x \beta,\ d_2 = \alpha q_2 \mu y \beta \wedge \exists (q_1,x,q_2,y,R) \in \delta)$$
$$\Rightarrow (\exists \alpha,\beta,x,y \in \Sigma^*,\ \exists q_1,q_2 \in Q,\ d_1 = \alpha q_1 x \beta,\ d_2 = \alpha y q_2 \beta \wedge \exists (q_1,x,q_2,y,R) \in \delta)$$
$$\vee\ (\exists \alpha,\beta,x,y \in \Sigma^*,\ \exists q_1,q_2 \in Q, \exists \mu \in \Sigma,\ d_1 = \alpha \mu q_1 x \beta,\ d_2 = \alpha q_2 \mu y \beta \wedge (q_1,x,q_2,y,L) \in \delta)$$

Séparons les deux propositions reliées par $\vee$ pour faciliter la lecture.

$$\exists \alpha,\beta \in \Sigma^*,\ \exists x,y \in \Sigma,\ \exists q_1,q_2 \in Q,\ d_1 = \alpha q_1 x \beta,\ d_2 = \alpha y q_2 \beta \wedge \exists (q_1,x,q_2,y,R) \in \delta$$
$$\Rightarrow \exists \alpha,\ \beta \in \Sigma^*,\ \exists x,y \in \Sigma,\ \exists q_1,q_2 \in Q, \mathbf{C}(d_1) = \alpha q_1 x \beta \square^\infty,\ \mathbf{C}(d_2) = \alpha y q_2 \beta \square^\infty,\ \beta \square^\infty \in \Sigma^\infty \wedge \exists (q_1,x,q_2,y,R) \in \delta$$
$$\Rightarrow \mathbf{C}(d_1) \vdash_{S_{\mathcal{M}}} \mathbf{C}(d_2)$$

D'autre part,

$$\exists \alpha,\beta \in \Sigma^*, \exists x,y,\mu \in \Sigma, \exists q_1,q_2 \in Q, d_1 = \alpha \mu q_1 x \beta, d_2 = \alpha q_2 \mu y \beta \wedge \exists (q_1,x,q_2,y,L) \in \delta$$
$$\Rightarrow \exists \alpha,\beta \in \Sigma^*, \exists x,y,\mu \in \Sigma, \exists q_1,q_2 \in Q, \mathbf{C}(d_1) = \alpha \mu q_1 x \beta \square^\infty, \mathbf{C}(d_2) = \alpha q_2 \mu y \beta \square^\infty, \beta \square^\infty \in \Sigma^\infty \wedge \exists (q_1,x,q_2,y,R) \in \delta$$
$$\Rightarrow \mathbf{C}(d_1) \vdash_{S_{\mathcal{M}}} \mathbf{C}(d_2)$$

Puisque les deux propositions impliquent que $\mathbf{C}(d_1) \vdash_{S_{\mathcal{M}}} \mathbf{C}(d_2)$ et qu'une au moins est vraie, nous avons que $\forall d_1, d_2 \in DI(\mathcal{M}), d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \Rightarrow \mathbf{C}(d_1) \vdash_{S_{\mathcal{M}}} \mathbf{C}(d_2)$.

Montrons maintenant que,

$$\forall c_1,c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), (\mathbf{C}(d_1) = c_1\ \wedge\ c_1 \vdash_{\mathcal{M}}^*\ c_2) \Rightarrow (\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}^*\ d_2\ \wedge\ \mathbf{C}(d_2) = c_2)$$

Encore une fois, cela se démontre immédiatement par induction si nous avons :

$$\forall c_1,c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), (\mathbf{C}(d_1) = c_1\ \wedge\ c_1 \vdash_{\mathcal{M}}\ c_2) \Rightarrow (\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}\ d_2\ \wedge\ \mathbf{C}(d_2) = c_2)$$

Nous étudions la proposition équivalente,

$$\forall c_1,c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), \mathbf{C}(d_1) = c_1,\ c_1 \vdash_{\mathcal{M}}\ c_2 \Rightarrow (\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}\ d_2\ \wedge\ \mathbf{C}(d_2) = c_2)$$

$$\forall c_1,c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), \mathbf{C}(d_1) = c_1,$$
$$c_1 \vdash_{\mathcal{M}}\ c_2$$
$$\Rightarrow \exists \alpha,\beta \in \Sigma^*, \beta' \in \Sigma^\infty, \beta' = \beta \square^\infty, \exists q_1,q_2 \in Q, \exists x,y \in \Sigma,$$
$$(c_1 = \alpha q_1 x \beta', c_2 = \alpha y q_2 \beta', d_1 = \alpha q_1 x \beta \wedge (q_1,x,q_2,y,R) \in \delta)$$

$$\vee\ (\exists \mu \in \Sigma, c_1 = \alpha\mu q_1 x\beta', c_2 = \alpha q_2 \mu y \beta', d_1 = \alpha q_1 x \beta \wedge (q_1, x, q_2, y, L) \in \delta)$$

Séparons encore une fois le problème entre les deux propositions.

$$\forall c_1, c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), \mathbf{C}(d_1) = c_1, \exists \alpha, \beta \in \Sigma^*, \exists \beta' \in \Sigma^\infty, \beta' = \beta\square^\infty, \exists q_1, q_2 \in Q, \exists x, y \in \Sigma,$$

$$c_1 = \alpha q_1 x \beta', c_2 = \alpha y q_2 \beta', d_1 = \alpha q_1 x \beta, \wedge (q_1, x, q_2, y, R) \in \delta$$
$$\Rightarrow c_2 = \alpha y q_2 \beta', d_1 = \alpha q_1 x \beta \ \wedge\ \exists (l, r) \in S_{\mathcal{M}}, l = q_1 x, r = y q_2$$
$$\Rightarrow \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_2 = \alpha y q_2 \beta, (c_2 = \alpha y q_2 \beta', d_1 = \alpha q_1 x \beta \ \wedge\ \exists (l, r) \in S_{\mathcal{M}}, l = q_1 x, r = y q_2)$$
$$\Rightarrow \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_2 = \alpha y q_2 \beta, (c_2 = \alpha y q_2 \beta', d_1 = \alpha q_1 x \beta \ \wedge\ d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \ \wedge\ c_2 = \mathbf{CI}(d_2))$$
$$\Rightarrow \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \ \wedge\ c_2 = \mathbf{CI}(d_2)$$

D'autre part,

$$\forall c_1, c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), \mathbf{C}(d_1) = c_1, \exists \alpha, \beta \in \Sigma^*, \exists \beta' \in \Sigma^\infty, \beta' = \beta\square^\infty, \exists q_1, q_2 \in Q, \exists x, y \in \Sigma,$$

$$\exists \mu \in \Sigma, c_1 = \alpha\mu q_1 x \beta', c_2 = \alpha q_2 \mu y \beta', d_1 = \alpha\mu q_1 x \beta, \wedge (q_1, x, q_2, y, L) \in \delta$$
$$\Rightarrow \exists \mu \in \Sigma, c_2 = c_2 = \alpha q_2 \mu y \beta', \alpha\mu q_1 x \beta \ \wedge\ \exists (l, r) \in S_{\mathcal{M}}, l = \mu q_1 x, r = q_2 \mu y$$
$$\Rightarrow \exists \mu \in \Sigma, \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_2 = \alpha q_2 \mu y \beta, (c_2 = \alpha q_2 \mu y \beta', d_1 = \alpha\mu q_1 x \beta \ \wedge\ \exists (l, r) \in S_{\mathcal{M}}, l = \mu q_1 x, r = q_2 \mu y)$$
$$\Rightarrow \exists \mu \in \Sigma, \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_2 = \alpha q_2 \mu y \beta, (c_2 = \alpha q_2 \mu y \beta', d_1 = \alpha\mu q_1 x \beta \ \wedge\ d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \ \wedge\ c_2 = \mathbf{CI}(d_2))$$
$$\Rightarrow \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_1 \rightarrow_{S_{\mathcal{M}}} d_2 \ \wedge\ c_2 = \mathbf{CI}(d_2)$$

À nouveau, puisque les deux propositions impliquent que $(\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}\ d_2\ \wedge\ \mathbf{C}(d_2) = c_2)$ et qu'une au moins est vraie, nous avons que :

$$\forall c_1, c_2 \in \mathbf{CONF}(\mathcal{M}), \forall d_1 \in \mathbf{DI}(\mathcal{M}), \mathbf{C}(d_1) = c_1,\ c_1 \vdash_{\mathcal{M}}\ c_2 \Rightarrow (\exists d_2 \in \mathbf{DI}(\mathcal{M}),\ d_1 \rightarrow_{S_{\mathcal{M}}}\ d_2\ \wedge\ \mathbf{C}(d_2) = c_2)$$

Q3 Montrer que la machine $\mathcal{M}$ accepte un mot $u \in \Gamma^*$ si et seulement si,

$$\exists v \in \Sigma^* Q_+ \Sigma^*,\ \ q_{\_\triangleright} u\ \rightarrow^*_{S_{\mathcal{M}}}\ v \tag{11}$$

La machine $\mathcal{M}$ accepte un mot $u \in \Gamma^*$ est équivalent à

$$\exists c_2 \in \mathbf{CONF}(\mathcal{M}), \exists u, v \in \Sigma^*, q_+ \in Q_+, c_2 = u q_+ v \square^\infty, q_{\_\triangleright} u \square^\infty\ \vdash^*_{\mathcal{M}}\ c_2$$

Dans un sens,

$$\Rightarrow (\exists d_1 \in \mathbf{DI}(\mathcal{M}), d_1 = q_{\_\triangleright} u, \mathbf{C}(d_1) = q_{\_\triangleright} u \square^\infty \Rightarrow \exists d_2 \in \mathbf{DI}(\mathcal{M}), d_1 \rightarrow^*_{S_{\mathcal{M}}} d_2 \wedge \mathbf{C}(d_2) = c_2)$$
$$\Rightarrow \exists v \in \mathbf{DI}, q_{\_\triangleright} u \rightarrow^*_{S_{\mathcal{M}}} v \wedge (\exists u, v \in \Sigma^*, q_+ \in Q_+, \mathbf{C}(v) = u q v \square^\infty)$$
$$\Rightarrow \exists v \in \Sigma^* Q_+ \Sigma^*,\ \ q_{\_\triangleright} u\ \rightarrow^*_{S_{\mathcal{M}}}\ v$$

Dans l'autre sens,

$$\exists v \in \Sigma^* Q_+ \Sigma^*,\ \ q_{\_\triangleright} u\ \rightarrow^*_{S_{\mathcal{M}}}\ v$$
$$\Rightarrow q_{\_\triangleright} u \square^\infty \vdash^*_{\mathcal{M}} v \square^\infty$$
$$\Rightarrow \exists c_2 \in \mathbf{CONF}(\mathcal{M}), \exists u, v \in \Sigma^*, q_+ \in Q_+, c_2 = v\square^\infty, q_{\_\triangleright} u \square^\infty\ \vdash^*_{\mathcal{M}}\ c_2$$
$$\Rightarrow \exists c_2 \in \mathbf{CONF}(\mathcal{M}), \exists u, v \in \Sigma^*, q_+ \in Q_+, c_2 = u q_+ v \square^\infty, q_{\_\triangleright} u \square^\infty\ \vdash^*_{\mathcal{M}}\ c_2$$

Q4 On considère l'alphabet $\Sigma^{'} := \Sigma \cup Q \cup \{\#\}$ obtenu en ajoutant un nouveau symbole $\#$ à $\Sigma \cup Q$. Construire un système semi-Thuéien $T_{\mathcal{M}}$ sur l'alphabet $\Sigma^{'}$ tel que, pour tout mot $u \in \Gamma^*$, (13) est vrai si et seulement si

$$q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ \#.$$

$$\forall q \in Q_+, (q, \#) \in S'$$
$$\forall \mu \in \Sigma, (\mu\#, \#) \in S'$$
$$\forall \mu \in \Sigma, (\#\mu, \#) \in S'$$
$$\forall (l, r) \in S', (\exists q \in Q_+, l = q \wedge r = \#) \vee (\exists \mu \in \Sigma, l = \mu\# \wedge r = \#) \vee (\exists \mu \in \Sigma, l = \#\mu \wedge r = \#)$$
$$T_{\mathcal{M}} = S_{\mathcal{M}} \cup S'$$

Nous avons déjà démontré qu'il y a équivalence entre l'acceptation d'un mot $u \in \Gamma^*$ par une machine $\mathcal{M}$ et $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{S_{\mathcal{M}}} \ v$.

Le système $T_{\mathcal{M}}$ garantit toujours cette équivalence dans la mesure où les règles (l, r) qui y sont ajoutées ($(l, r) \in T_{\mathcal{M}} \setminus S_{\mathcal{M}} = S'$) sont telles que $l = q, q \in Q_+$ ou $\exists u, v \in \Sigma^*, l = u\#v$. Autrement dit, le premier type de règle ne permet une dérivation que sur un élément de la forme de v et ne peut donc pas agir rétroactivement sur la dérivation $q_{\_\triangleright} u \ \rightarrow^*_{S_{\mathcal{M}}} \ v$. Le second type de règle ne pouvant s'appliquer qu'après l'usage de la première règle dans la mesure où $\{\#\}$ n'appartient pas à l'alphabet de $\mathcal{M}$, il n'y a pas non plus d'action rétroactive.

De ce fait il y a donc l'équivalence entre l'acceptation d'un mot $u \in \Gamma^*$ par une machine $\mathcal{M}$ et $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ v$.

Il suffit donc de montrer maintenant qu'il y a équivalence entre $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ v$ et $q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ \#$.

Montrer l'implication de gauche à droite est triviale dans la mesure où nous avons $v \rightarrow^*_{T_{\mathcal{M}}} \ \#$ en employant la règle issue de la première proposition de la définition de $S' \subset T_{\mathcal{M}} : \forall q \in Q_+, (q, \#) \in S'$ puis en employant les règles issues de la deuxième et proposition de la définition de S' qui permettent d'enlever les éléments respectivement à gauche et à droite du $\#$.

Il faut maintenant démontrer que $q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ \# \Rightarrow (\exists v \in \Sigma^* Q_+ \Sigma^* q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ v)$. Pour cela, il suffit de remarquer que le symbole $\#$ n'appartient pas à l'alphabet de $\mathcal{M}$ et ne peut apparaître que dans une relation du type $q \in Q_+, (q, \#)$. On a donc $q_{\_\triangleright} u \rightarrow^*_{T_{\mathcal{M}}} \# \Rightarrow \exists v \in (\Sigma \cup Q)^* Q_+ (\Sigma \cup Q \cup \{\#\})^*, q_{\_\triangleright} u \rightarrow^*_{T_{\mathcal{M}}} v \rightarrow^*_{T_{\mathcal{M}}} \#$.

Or, nous avons que $\forall w, w' \in (\Sigma \cup Q \cup \{\#\})^*, w \ \rightarrow_{T_{\mathcal{M}}} \ w' \Rightarrow |w|_{q \in Q} + |w|_{\#} = |w'|_{q \in Q} + |w'|_{\#}$. Puisque, $|q_{\_\triangleright} u|_{q \in Q} + |q_{\_\triangleright} u|_{\#} = 1$ et $q_{\_\triangleright} u \rightarrow^*_{T_{\mathcal{M}}} v$, alors $|v|_{q \in Q} + |v|_{\#} = 1 \wedge |v|_{q \in Q} \geq 1 \rightarrow |v|_{\#} = 0 \wedge |v|_{q \in Q} = 1$. Donc $v \in \Sigma^* Q_* \Sigma^*$.

Donc il y a bien équivalence entre $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ v$ et $q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ \#$. Et puisqu'il y a aussi équivalence entre $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ v$ et $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{S_{\mathcal{M}}} \ v$, il y a équivalence entre $\exists v \in \Sigma^* Q_+ \Sigma^*, \ q_{\_\triangleright} u \ \rightarrow^*_{S_{\mathcal{M}}} \ v$ et $q_{\_\triangleright} u \ \rightarrow^*_{T_{\mathcal{M}}} \ \#$.

Q5 Donner une réduction du problème (ACC-MT) (le problème de l'acceptation pour les machines de Turing) au problème (ACC-ST).

Supposons qu'il existe une machine M_{ST} qui prend en entrée un mot w et un système semi-Thuéiens $S_{\mathcal{M}}$ et qui retourne "oui" en temps fini si $w \rightarrow^*_{S_{\mathcal{M}}} \#$.

Montrons que pour toutes instances (machine de Turing M, mot w) du problème d'acceptation, il est possible de créer une machine $R_{(M,x)}$ qui termine en temps fini tel que $M_{ST}(R_{(M,x)}) =$ "oui" si M termine sur x et "non" sinon. Autrement dit, montrons qu'il existe une réduction de (ACC-MT) à (ACC-ST).

Construisons la machine R qui prend en entrée une machine M et un mot x et qui retourne le système semi-Thuéiens T_M tel que décrit précédemment, ainsi que le mot $q_{\_\triangleright} x$.

Montrons que pour toute entrée, R termine en temps fini. Le retour de $q_{\_\triangleright} x$ à partit de x a lieu en temps fini de façon triviale. La construction de T_M a lieu en temps fini puisqu'il s'agit de construire une règle par transitions "droites", $|\Sigma|$ règles par transitions gauches et un nombre fini de transitions supplémentaires proportionnel à $|Q_+|$ et $|\Sigma|$. Ainsi, dans la mesure où une machine de Turing possède un nombre fini de transitions et d'états acceptants, et sous hypothèse que l'alphabet qu'elle emploie est lui aussi fini, le système T_M est fini et sa création a lieu en temps fini.

Le système T_M est construit de telle sorte à ce qu'il y ait équivalence entre $q_{\_\triangleright} x \rightarrow^*_{T_M} \#$ et acceptation de x par M, donc $M_{ST}(R_{(M,x)})$ répond au problème de (ACC-MT) en temps fini. Il y a donc une réduction du problème (ACC-MT) au problème (ACC-ST). Cela implique que le problème (ACC-ST) est indécidable.

Q6 Montrer que, même en se restreignant aux systèmes semi-Thuéiens sur l'alphabet à 2 lettres $\{a, b\}$, le problème (ACC-ST) reste indécidable.

Nous avons montré en cours que si une fonction $f : \{0,1\}^* \to \{0,1\}^*$ est calculable par une machine de Turing $M = (\Sigma, Q, \delta)$, alors f est calculable pour une machine de Turing $M^{'}$ définie sur un alphabet $\Sigma^{'} = \Gamma \cup \{\triangleright, \square\}$, $\Gamma = \{a, b\}$. Pour cela, on code chaque symbole de Σ en base 2 sur $\log(|\Sigma|)$ bits, en choisissant arbitrairement que a correspond à la valeur 0_2 et b à 1_2 (ou inversement).

Ainsi, si le problème de l'acceptation pour les machines de Turing ayant un alphabet composé de quatre symboles $\Sigma^{'} = \Gamma \cup \{\triangleright, \square\}$ était décidable, alors le problème de l'acceptation pour toute machine de Turing (ACC-MT) serait également décidable. Or, (ACC-MT) est indécidable, donc le problème de l'acceptation pour les machines de Turing $M^{'} = (\Sigma^{'}, Q^{'}, \delta^{'})$ telles que $|\Sigma^{'}| = 4$ est indécidable.

Enfin, puisque l'acceptation d'une machine de Turing $M^{'}$ d'alphabet $\Sigma^{'} = \Gamma \cup \{\triangleright, \square\}$, $\Gamma = \{a, b\}$, est indécidable, et que cette machine peut être simulée par un système semi-Thuéiens sur Γ^*, le problème (ACC-ST) est indécidable pour les systèmes semi-Thuéiens sur $\{a, b\}^*$.

Partie 4 : Système semi-Thuéiens versus Problème de Post

Q1 Montrer que, si $u = u_0 \to_S \ u_1 \cdots \to_S \ u_p = v$, alors $\Phi(S, u, v)$ a une solution utilisant p occurrences de couples de l'ensemble $\{l_i^x, {}^x r_i \mid 1 \leq i \leq n\}$.

Nous avons que $\forall i \in [0, p-1], u_i \to_S u_{i+1}$. Pour tout $u_i \to_S u_{i+1}$, une règle de S est employée. On peut donc réaliser une suite (s_i) telle que $\forall i \in [0, p-1], s_i \in [1, n], \exists\ \alpha_i, \beta_i \in \{a, b\}^*, u_i = \alpha_i\ l_{s_i}\ \beta_i \wedge u_{i+1} = \alpha_i\ r_{s_i}\ \beta_i$.

Nous proposons la notation $\bar{w}$ qui envoie tout mot $w \in \{a, b\}^*$ vers une suite d'entiers λ, telle que $\forall i \in [0, |w| - 1]$:

$$\lambda_i = \begin{cases} 3 & \text{si } w_i = a \\ 4 & \text{sinon.} \end{cases}$$

On considérera qu'une suite λ placée dans la suite de réponse correspond en réalité à placer tous les éléments de λ successivement.

Nous proposons la notation $\tilde{\mu}$ qui envoie tout entier $\mu \in [1, n]$ vers $\mu + 5$.

On propose la solution $k = 1 + \sum_{i=0}^{p-1}(2 + |\alpha_i| + |\beta_i|)$ et la suite $1, \bar{\alpha_1}, \tilde{s_1}, \bar{\beta_1}, 5, \bar{\alpha_2}, \tilde{s_2}, \bar{\beta_2}, 5, ..., \bar{\alpha_{p-1}}, \tilde{s_{p-1}}, \bar{\beta_{p-1}}, 2$. Cette solution emploie les couples $\tilde{s_i}, \forall i \in [0, p-1]$; autrement dit, elle emploie p occurrences de l'ensemble $\{(l_i, r_i) | i \in [1, n]\}$.

Cela donne comme mots construits (nous n'emploierons pas les notations $u_{i_1} u_{i_2} u_{i_k}$ et $v_{i_1} v_{i_2} v_{i_k}$ car la suite u a été redéfinie dans l'exercice) :

$$\#\#x\ \alpha_0^x\ l_{s_0}^x\ \beta_0^x \#x\ \alpha_1^x\ l_{s_1}^x\ \beta_1^x\ \#x\ ...\ \alpha_{p-1}^x\ l_{s_{p-1}}^x\ \beta_{p-1}^x\ \#xv^x\#\#$$

et

$$\#\#xu^x\#\ {}^x\alpha_0\ {}^x r_{s_0}\ {}^x\beta_0\ x\#\ {}^x\alpha_1\ {}^x r_{s_1}\ {}^x\beta_1\ x\#\ ...\ {}^x\alpha_{p-1}\ {}^x r_{s_{p-1}}\ {}^x\beta_{p-1}\ x\#\#$$

Or, $\forall i \in [0, p-1], u_i = \alpha_i\ l_{s_i}\ \beta_i\ \wedge\ u_{i+1} = \alpha_i\ r_{s_i}\ \beta_i$ donc $\forall i \in [0, p-1], x\ \alpha_{i+1}^x\ l_{s_{i+1}}^x\ \beta_{i+1}^x = xu_{i+1}^x =^x \alpha_i\ {}^x r_{s_i}\ {}^x\beta_i\ x$.

Donc les deux mots sont égaux à :

$$\#\#xu_0^x\ \#\ xu_1^x\ \#\ ...\ \#xu_p^x\#\#$$

Nous avons donc montré que si $u = u_0 \to_S \ u_1 \cdots \to_S \ u_p = v$, alors $\Phi(S, u, v)$ a une solution utilisant p occurrences de couples de l'ensemble $\{l_i^x, {}^x r_i \mid 1 \leq i \leq n\}$.

Q2 Montrer que, si $H \in \{D, F, L_a, L_b, L_\#, R_1, \ldots, R_n\}^*$ est une solution atomique, alors H est de forme :

$$H = D \cdot H_0 \cdot F$$

avec $H_0 \in \{L_a, L_b, L_\#, R_1, ..., R_n\}^*$ et

$$\varphi(H_0)\#^x v = u^x \# \psi(H_0)$$

Si H est une solution, en particulier on a que ${}^{(1)}\varphi(H) =^{(1)} \psi(H)$ (les premiers symboles sont égaux) et $\varphi(H)^{(-1)} = \psi(H)^{(-1)}$ (les derniers symboles le sont aussi). Puisqu'aucun des couples de $\Phi(S, u, v)$ ne possède un élément nul, cela implique que ${}^{(1)}\varphi(H_1) =^{(1)} \psi(H_1)$ et $\varphi(H_{|H|})^{(-1)} = \psi(H_{|H|})^{(-1)}$.

Or, seul D est tel que ${}^{(1)}\varphi(D) =^{(1)} \psi(D)$ et seul F est tel que $\varphi(F)^{(-1)} = \psi(F)^{(-1)}$. Donc toute solution est forcément décomposable en $D \cdot I_0 \cdot F$ avec $I_0 \in X^*$. En particulier, une solution atomique est une solution, donc elle doit être pareillement décomposable.

Montrons maintenant qu'une solution atomique H est décomposable en $D \cdot H_0 \cdot F$ avec en particulier $H_0 \in (X \backslash \{D, F\})^*$. Pour cela, montrons que pour toute solution I de la forme $D \cdot I_0 \cdot F$ (comme montré précédemment), si $|I_0|_{D,F} \geq 1$, alors la solution ne peut pas être atomique.

Pour cela, remarquons d'abord que les symboles D et F sont les seuls associés à des couples contenant deux symboles # consécutifs. De plus, il n'est pas possible de créer deux symboles # consécutifs sans employer D ou F. En effet, il n'existe pas de lettres de $X \setminus \{D, F\}$ qui à travers φ donne un mot se terminant par #, et, à l'inverse pour ψ, il n'en existe pas qui donne un mot commençant par #. Il est donc impossible de placer deux # consécutifs par φ comme par ψ sans employer les symboles D ou F.

Supposons qu'il y ait un ou plusieurs symboles D ou F dans I_0. Nous nous intéressons au premier symbole D ou F de I_0.

Pour qu'il y ait égalité de $\varphi(I)$ et $\psi(I)$, les doubles # par φ issus du symbole D ou F sont rencontrés par des doubles # par ψ aux mêmes indices. Ces doubles # doivent venir du même symbole puisque nous nous intéressons à la première occurrence de #, à l'exception du D d'origine dont les # par φ et ψ coïncident obligatoirement entre eux. De ce fait, la chaîne extraite de $\varphi(I)$ du départ jusqu'aux doubles # issus du second D ou du premier F doit être égale à la chaîne extraite de la même façon sur ψ.

Supposons d'abord qu'il s'agit d'un symbole D. Puisque les doubles # débutent $\varphi(D)$ et $\psi(D)$, nous avons que le mot I' extrait de I jusqu'à ce second symbole D non compris répond à $\varphi(I') = \psi(I')$. Il s'agit donc d'une solution. Puisque nous avons réussi à décomposer I en une solution et un autre mot différent de ε, il ne peut pas s'agir d'une solution atomique.

Au contraire, si le symbole est F, puisque les doubles # terminent $\varphi(D)$ et $\psi(D)$, nous avons que le mot I' extrait de I jusqu'à ce premier symbole F compris répond à $\varphi(I') = \psi(I')$. Il s'agit donc d'une solution. Puisque nous avons réussi à décomposer I en une solution et un autre mot différent de ε, il ne peut pas s'agir non plus d'une solution atomique.

Ainsi, quel que soit le symbole D ou F contenu dans I_0, I ne peut être atomique. Au contraire, puisque toute solution atomique est solution et est donc de la forme $H = D \cdot H_0 \cdot F$, nous avons forcément que $H_0 \in X \backslash \{D, F\}$.

Si H est une solution, alors $\varphi(H) = \psi(H)$, ce qui implique que si l'on coupe $\varphi(H)$ et $\psi(H)$ aux mêmes indices, les sous-chaînes extraites doivent rester égales.

En particulier, nous avons vu que $H = D \cdot H_0 \cdot F$, ce qui implique que $\varphi(H) = \#\#x\ \varphi(H_0)\ \#xv^x\#\#$ et $\psi(H) = \#\#xu^x\ \psi(H_0)\ x\#\#$. Si nous retirons de $\varphi(H)$ et $\psi(H)$ les trois premiers symboles ainsi que les trois derniers, nous obtenons que $\varphi(H_0)\ \#^x v = u^x\ \psi(H_0)$.

Q3 Montrer que, si $H \in \{L_a, L_b, R_1, \ldots, R_n\}^*$, alors $\exists u_0, v_0 \in \{a, b\}^*$ tels que :

$$\varphi(H) = u_0^x, \phi(H) = {}^x v_0, u_0 \rightarrow_S^p v_0 \text{ où } p = |H|_{R_1,\ldots,R_n}.$$

Nous pouvons remarquer que $\forall \mu \in \{L_a, L_b, R_1, R_2, ..., R_n\}, \exists \lambda_1, \lambda_2 \in \{a, b\}^*, \varphi(\mu) = \lambda_1^x \wedge \psi(\mu) = {}^x\lambda_2$ par observation des couples de $\Phi(S, u, v)$. Donc, puisque φ et ψ sont des homomorphismes et que $\forall w_1, w_2 \in \{a, b\}^*, w_1^x \cdot w_2^x = (w_1 \cdot w_2)^x \wedge {}^x w_1 \cdot {}^x w_2 = {}^x(w_1 \cdot w_2)$, nous avons que $\forall H \in \{L_a, L_b, R_1, R_2, ..., R_n\}^*, \exists u_0, v_0 \in \{a, b\}^*, \varphi(H) = u_0^x \wedge \psi(H) = {}^x v_0$.

$\forall p \in \mathbb{N}, \forall H \in \{L_a, L_b, R_1, R_2, ..., R_n\}^*, |w|_{\{R_1,R_2,\ldots,R_n\}} = p, H = (\{L_a, L_b\}^*\{R_1, R_2, ..., R_n\})^p\{L_a, L_b\}^*$. Cela implique $\forall i \in [1, p], \exists \alpha_i \in \{a, b\}^*, \exists \alpha_{p+1} \in \{a, b\}^*, \exists s_i \in [1, n], \varphi(H) = (\alpha_1 l_{s_1} \alpha_2 l_{s_2} ... \alpha_p l_{s+1} \alpha_{p+1})^x \wedge \psi(H) = {}^x(\alpha_1 r_{s_1} \alpha_2 r_{s_2} ... \alpha_p r_{s+1} \alpha_{p+1})$. Dans ce cas, $u_0 = \alpha_1 l_{s_1} \alpha_2 l_{s_2} ... \alpha_p l_{s+1} \alpha_{p+1}$ et $v_0 = \alpha_1 r_{s_1} \alpha_2 r_{s_2} ... \alpha_p r_{s+1} \alpha_{p+1}$.

Nous pouvons alors procéder à la dérivation de u_0 selon la règle (l_{s_1}, r_{s_1}), qui existe forcément car $s_1 \in [1, n]$, sur le premier sous-mot l_{s_1}.Une fois le résultat obtenu, nous pouvons réitérer une dérivation sur le résultat sur le premier mot l_{s_2} et la règle (l_{s_2}, r_{s_2}) et ce jusqu'à l_{s_p} et la règle (l_{s_p}, r_{s_p}). Le mot ainsi obtenu sera u_0 où tous les $l_{s_i}, \forall i \in [1, p]$ auront été remplacés par r_{s_i} le rendant égal à v_0. De plus, ce procédé a nécessairement lieu en p dérivations donc nous avons prouvé ce qu'il fallait démontrer.

Q4 Montrer que, pour tous mots $H \in \{L_a, L_b, L_\#, R_1, \ldots, R_n\}^*, u_1 \in \{a,b\}^*, v_1 \in \{a,b\}^*$,

$$\varphi(H)\#\ {}^x v_1 = u_1^x \# \psi(H) \Rightarrow [u_1 \rightarrow_S^p\ v_1 \text{ où } p = |H|_{R_1,\ldots,R_n}].$$

Nous procédons par récurrence sur $|H|_{L_\#}$.

D'abord, pour $H_{L_\#} = 0$, $H \in \{L_a, L_b, R_1, R_2, ..., R_n\}^*$. Si nous avons $\varphi(H)\#\ {}^x v_1 = u_1^x \# \psi(H)$, et que $|\varphi(H)|_\# = 0$, nous avons $\varphi(H) = u_1^x \wedge\ {}^x v_1 = \psi(H)$. Nous pouvons utiliser directement la réponse à la question précédente pour montrer que $u_1 \rightarrow_S^p v_1$ avec $p = |H|_{R_1,R_2,\ldots,R_n}$.

Pour $|H|_{L_\#} = n+1$, nous découpons H en une partie du début jusqu'au premier symbole $L_\#$ (non compris), que nous nommons H_1 avec $|H_1|_{R_1,R_2,\ldots,R_n} = 0$, puis $L_\#$ et enfin le reste que nous nommons H'.

En particulier, si nous avons $\varphi(H)\#\ {}^x v_1 = u_1^x \# \psi(H)$, nous avons $\varphi(H_1)\#x\varphi(H')\#^x v_1 = u_1^x \# \psi(H_1)x\#\psi(H')$. Puisqu'il y a égalité, nous pouvons supprimer tous les éléments jusqu'au premier symbole $\#$ (compris) des deux côtés de l'égalité. Nous obtenons alors $x\varphi(H')\#^x v_1 = \psi(H_1)x\#\psi(H')$.

$\exists u_2 \in \{a,b\}^*,\ {}^x u_2 = \psi(H_1)$, donc nous obtenons $x\varphi(H')\#^x v_1 =\ {}^x u_2 x\#\psi(H') \wedge |H'|_{L_\#} = n$.
Or, $x\varphi(H')\#^x v_1 =\ {}^x u_2 x\#\psi(H') \Rightarrow \varphi(H')\#^x v_1 = u_2^x \#\psi(H')$. Nous pouvons donc appliquer l'hypothèse de récurrence qui nous donne $u_2 \rightarrow_S^{p'} v_1$ où $p' = |H'|_{R_1,R_2,\ldots,R_n}$.

Cependant, nous pouvons aussi supprimer tous les éléments à partir du premier symbole # compris pour obtenir : $\varphi(H_1) = u_1^x$. Nous avons déjà défini la variable u_2 telle que ${}^x u_2 x\# = \psi(H_\#)$. Puisque $|H_1|_{R_1,R_2,\ldots,R,} = 0$, nous avons donc $u_1 \rightarrow_S^{p"} u_2$ où $p" = |H_\#|_{R_1,R_2,\ldots,R_n}$, selon la question 2.

Nous avons alors $u_1 \rightarrow_S^{p"} u_2 \rightarrow_S^{p'} v_1$ avec $p" = |H_1|_{R_1,R_2,\ldots,R_n} \wedge p' = |H'|_{R_1,R_2,\ldots,R_n} \Rightarrow p' + p" = |H|_{R_1,R_2,\ldots,R_n} = p$. Donc $u_1 \rightarrow_S^p v_1$.

Par récurrence nous avons bien pour $\forall H \in \{L_a, L_b, L_\#, R_1, ..., R_n\}^*, u_1 \in \{a,b\}^*, v_1 \in \{a,b\}^*$,

$$\varphi(H)\#\ {}^x v_1 = u_1^x \# \psi(H) \Rightarrow [u_1 \rightarrow_S^p\ v_1 \text{ où } p = |H|_{R_1,\ldots,R_n}]$$

Q5 Montrer que $u \rightarrow_S^* v$ si et seulement si $\Phi(S,u,v)$ a une solution.

Nous avons montré dans la question 1 que $\forall p \in \mathbb{N}, \forall u, v \in \{a,b\}^*, u \rightarrow_S^p v$ implique que $\Phi(S,u,v)$ a une solution en p occurrences. Cela garantit que pour toute valeur de p, $u \rightarrow_S^p v$ implique que $\Phi(S,u,v)$ a une solution finie. Cette généralité de $p \in \mathbb{N}$ nous donne que "$u \rightarrow_S^* v$ implique que $\Phi(S,u,v)$ a une solution finie".

Dans l'autre sens, si $\Phi(S,u,v)$ a une solution finie, il en existe forcément une atomique et selon la réponse à la question 2, celle-ci est nécessairement de la forme $H = D \cdot H_0 \cdot F, H_0 \in (X \setminus \{D,F\})^*$. Dans la question 4, nous avons montré que pour un tel H_0, $\varphi(H)\#\ {}^x v_1 = u_1^x \# \psi(H) \Rightarrow [u_1 \rightarrow_S^p\ v_1 \text{ où } p = |H|_{R_1,\ldots,R_n}]$. Or, puisque H est solution et H est de la forme $D \cdot H_0 \cdot F$, en retirant les trois symboles de départ et les trois symboles de fin de $\varphi(H)$ et de $\psi(H)$, nous devons obtenir le même sous-mot. Il y a donc une égalité de la forme $\varphi(H_0)\#^x v = u^x \# \psi(H_0)$. Cela implique que $u \rightarrow_S^p v$ avec $p = |H_0|_{R_1,R_2,\ldots,R_n}$. Or, puisque que H est fini, H_0 l'est aussi et donc $|H_0|_{R_1,R_2,\ldots,R_n} \leq |H_0| < \infty$. Nous avons donc montré qu'il existe toujours un nombre fini de dérivations telles que $u \rightarrow_S^* v$ si $\Phi(S,u,v)$ a une solution finie.

Il y a donc équivalence.

Q6 Conclure que Φ est une réduction du problème (ACC-ST) au problème (PCP) et que (PCP) est indécidable.

S'il existe une machine qui décide du problème de Post, il est possible de l'employer sur $\Phi(S,u,v)$ pour tout S système semi-Thuéien fini, $u, v \in a, b^*$. L'équivalence démontrée en question 5 implique qu'en faisant cela, nous sommes alors en mesure de décider de $u \rightarrow_S^* v$, qui correspond au problème (ACC-ST). Il y a donc réduction du problème (ACC-ST) (contraint à l'alphabet $\{a,b\}$) au problème (PCP).

Cependant, puisque le problème (ACC-ST) est indécidable (même sur l'alphabet $\{a,b\}$), comme nous l'avons démontré aux questions 6 et 7 de la partie 3, cela implique que le problème (PCP) est lui-même indécidable.

Q7 Pouvez-vous maintenant résoudre (PCP) sur l'instance I_3 de la partie I, question 4 ?

Il est maintenant clair que l'instance I_3 du problème (PCP) de la partie I, question 4 est équivalente au problème S_1 du problème (ACC-ST) de la partie II, question 4. La résolution demeure peu esthétique mais la méthodologie pour l'obtenir en devient triviale.