Problèmes de décision sur les semi-groupes

Auteur : Tihon, Elisa
Promoteur(s) : Charlier, Emilie
Faculté : Faculté des Sciences
Diplôme : Master en sciences mathématiques, à finalité didactique
Année académique : 2019-2020
URI/URL : http://hdl.handle.net/2268.2/9208

Université de Liège
Faculté des Sciences
Département de mathématique

Problèmes de décision sur les semi-groupes

Mémoire de fin d'études présenté en vue de l'obtention du grade de Master en Sciences Mathématiques à finalité

Promoteur : Émilie Charlier

réalisé par

Elisa Tihon

Année académique 2019-2020

Remerciements

« Beaucoup de nos rêves semblent d'abord impossibles, puis improbables, et enfin, en faisant preuve de suffisamment de volonté, ils deviennent rapidement inévitables. »

Christopher Reeve

Je souhaite tout d'abord remercier ma promotrice Émilie Charlier, de m'avoir proposé ce sujet et de m'avoir encadré durant mes deux années de master. Je la remercie également pour sa disponibilité sans limite, ses conseils avisés et ses réponses à mes nombreuses questions.

J'en profite pour remercier toutes les personnes du département qui m'ont soutenue durant la réalisation de ce travail mais surtout durant mes cinq années d'études.

Je tiens évidement à remercier mes parents pour avoir cru en moi depuis toujours et m'avoir permis de réaliser ces si belles études. Leur soutient et leurs encouragements ont contribué, sans aucun doute, à ma réussite. Je remercie également Grégoire pour avoir toujours su trouver les mots pour me réconforter et me faire aller de l'avant. Je lui suis très reconnaissante de m'avoir soutenue et motivée durant ces cinq années.

Finalement, parce que ces études n'auraient pas été si belles sans eux, je tiens à remercier mes amis de math, avec qui les cours et les temps de midi furent plus qu'agréables. Je remercie particulièrement Emeline pour les moments de partage, de doute, de fou rire et de travail. Elle représente sans doute ma plus belle rencontre de ces études.

Table des matières

Introduction

La *calculabilité* est la branche des mathématiques qui s'occupe de modéliser ce qui est effectivement calculable.

Dans les années 1930 s'élabore, sous l'impulsion notamment du mathématicien Alan Turing, une théorie abstraite de la calculabilité, et ce avant même l'avènement de l'ordinateur. Mais avant le développement de l'informatique, la théorie de la calculabilité n'était utilisée et développée que par un cercle restreint de mathématiciens et de logiciens. L'apparition des ordinateurs, auxquels la théorie de la calculabilité s'applique parfaitement, en a fait un sujet en plein développement qui passionne de nombreux chercheurs encore actuellement. On pourrait dire maintenant qu'une connaissance élémentaire de la calculabilité fait partie du bagage scientifique essentiel à l'informaticien. La théorie de la calculabilité est en effet un ingrédient classique de n'importe quel cours de logique mathématique. Même si d'autres disciplines comme la théorie des modèles, la théorie des ensembles et la théorie de la démonstration se sont développées comme des sujets à part entière, les liens avec la théorie de la calculabilité restent nombreux et la découverte de nouveaux liens se poursuit. Plusieurs branche d'informatique ont découlé de la théorie de la calculabilité, comme la théorie de la complexité, les langages formels et la théorie des automates.

Une question fondamentale en informatique théorique est de déterminer si un problème donné peut ou non être résolu au moyen d'un programme exécuté sur un ordinateur. Pour cela, il faut préciser ce qu'on entend par *problème* : un problème est une question générale, qui s'applique à un ensemble d'éléments. Chaque entrée du problème possède une réponse (oui ou non). Un problème serait donc par exemple de déterminer, pour un nombre naturel donné, s'il est premier ou non. Si l'entrée de ce problème est 17, la réponse est "oui", si l'entrée est 18, la réponse est "non". Mais la vraie question à se poser, c'est "peut-on toujours, peu importe l'entrée, déterminer si oui ou non le nombre est premier ?"

En théorie de la calculabilité, formuler un tel problème c'est se poser une question de décidabilité. Il s'agit en fait de rechercher l'existence d'un algorithme qui résout le problème et, s'il existe, de l'expliciter. Un *algorithme* est une procédure finie ou un ensemble fini de règles destiné à résoudre un problème étape par étape. La qualité d'un algorithme ou d'un programme s'évalue au nombre d'opérations de base qu'il exécute pour parvenir à ses fins, reflété par le nombre d'instructions effectuées par l'ordinateur.

Ce présent travail parcourt plusieurs problèmes de décision, principalement appliqués à des semi-groupes, et tente d'en caractériser leur décidabilité. Il s'articule essentiellement autour de l'article [4] "*On the decidability of semigroup freeness*" de Julien Cassaigne et

François Nicolas. Il se décompose en 6 chapitres, de la façon suivante :

Le premier chapitre de ce travail sera consacré aux notations et définitions de base qui seront utilisées dans les chapitre suivants. On définira en particulier la notion de *code*, qui est à la base de ce travail et nous l'illustrerons. Ensuite on introduira les problèmes de décision et plus particulièrement celui traitant du caractère libre d'un semi-groupe. C'est celui-ci qui reviendra le plus souvent et qui sera analysé en long et en large. On le note Free$[S]$ où S est un semi-groupe récursif et on le définit comme suit : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si X est un code. Enfin, on énoncera plusieurs autres problèmes de décision qui sont en lien avec le premier et qui réapparaitront ponctuellement dans ce travail.

Le Chapitre 2 sera consacré aux problèmes de décision relatifs à des semi-groupes engendrés par un unique élément. Nous introduirons un premier concept qui sera utilisé dans tout le chapitre, la notion d'élément *de torsion*. Cela nous permettra de caractériser les matrices qui sont de torsion et nous introduirons un nouveau problème de décision, noté Matrix Torsion, défini comme suit : étant donnés un nombre entier $d \geq 1$ et une matrice $M \in \mathbb{Q}^{d\times d}$, déterminer si M est de torsion. Nous démontrerons, à l'aide de plusieurs résultats préliminaires, que ce problème est décidable en temps polynomial. Enfin, nous introduirons un dernier problème de décision qui revient à se demander si un morphisme donné est de torsion ou non. Nous démontrerons que ce problème est décidable en temps polynomial lui aussi, grâce à une réduction au problème Matrix Torsion.

Nous caractériserons, dans le chapitre 3, les sous-ensembles qui ne sont pas des codes. Nous démontrerons qu'ils satisfont une équation non-triviale, appelée *équation équilibrée*. On introduira ensuite le concept de semi-groupe *simplifiable*. Cela nous permettra de nouveau de caractériser les sous-ensembles qui ne sont pas des codes ainsi que de déterminer le lien entre la décidabilité du problème Free$[S \times T]$ et des problèmes Free$[S]$ et Free$[T]$ lorsque S et T sont deux semi-groupes non vides. Ensuite nous discuterons du cas des matrices entières et des matrices rationnelles pour en conclure que le problème Free$[\mathbb{Q}^{d\times d}]$ est décidable si et seulement si le problème Free$[\mathbb{Z}^{d\times d}]$ l'est. Enfin, nous regarderons le cas des groupes et nous caractériserons la décidabilité du problème Free$[G]$ lorsque G est un groupe.

Dans le chapitre suivant, nous allons nous concentrer sur des problèmes de décision concernant le semi-groupe des matrices 2×2 et plus précisément sur le problème Free$[\mathbb{N}^{2\times2}]$. Nous caractériserons dans un premier temps la décidabilité du problème Free$[K^{2\times2}]$ pour un champ K. On se restreindra en particulier aux matrices triangulaires supérieures et inférieures. L'étude de ces problèmes fera surgir plusieurs questions ouvertes intéressantes à discuter. Enfin, on terminera ce chapitre par le problème de décision Mortal$[S]$ où S est un semi-groupe récursif possédant un zéro. Ce problème se définit comme suit : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si le zéro de S appartient à X^+. Ce problème sera étudié sur le semi-groupe $\mathbb{N}^{2\times2}$.

Le chapitre 5 sera consacré à l'étude du problème de correspondance de Post tel qu'il l'a

lui-même énoncé. On le notera PCP et il peut être défini comme suit, en termes modernes : étant donnés un alphabet Σ et deux morphismes $\sigma, \tau \colon \Sigma^* \to \{0,1\}^*$, déterminer s'il existe un mot $w \in \Sigma^*$ tel que $\sigma(w) = \tau(w)$. On introduira la notion de *système normal* qui nous permettra de démontrer l'indécidabilité de PCP. Ensuite, on s'intéressera à un problème généralisé du problème de correspondance de Post, noté GPCP. On démontrera, entre autres, que si ce problème est décidable alors PCP l'est aussi, en introduisant les *systèmes semi-Thue*.

Dans le dernier Chapitre de ce travail on se concentrera sur des problèmes de décision sur les semi-groupes de matrices carrées de grande dimension. Dans un premier temps, nous tenterons de caractériser la décidabilité du problème $\textsc{Free}[\mathbb{N}^{3\times 3}]$, pour cela nous nous aiderons du problème $\textsc{Free}[\{0,1\}^* \times \{0,1\}^*]$ ainsi que du problème de modification mixée du problème de correspondance de Post. Dans un second temps, nous reviendrons sur le problèmle de décision $\textsc{Mortal}[S]$ discuté au Chapitre 4 mais cette fois pour le semi-groupe $S = \mathbb{Q}^{3\times 3}$. Nous démontrerons que ce problème est indécidable. Enfin, nous regarderons le cas des matrices de plus grande dimension, le but sera de caractériser la décidabilité du problème $\textsc{Free}[\mathbb{N}^{d\times d}]$ pour $d > 3$.

Chapitre 1

Définitions de base et problèmes de décision

Dans ce premier chapitre, nous prendrons soin de définir quelques notions importantes pour la suite, notamment la définition d'un code qui est à la base de ce travail et qui réapparaitra ponctuellement dans celui-ci. De plus, nous parlerons de semi-groupes, en particulier de semi-groupes libres, ainsi que de monoïdes libres et nous discuterons de plusieurs exemples pour mieux comprendre ces notions.

Ensuite, nous introduirons des problèmes de décision et plus particulièrement le problème qui traite du caractère libre de semi-groupes, que nous mettrons en lien avec les morphismes de semi-groupes.

Enfin, nous définirons d'autres problèmes de décision qui seront mis en lien avec le premier problème dans les chapitres qui suivent.

1.1 Notations et définitions de base

Un *semi-groupe* est un ensemble muni d'une opération associative qui est binaire, interne et partout définie. Sans mention explicite, on considèrera les opérations de semi-groupes notées multiplicativement. Dans un semi-groupe S, un élément e est dit *neutre* si

$$s \cdot e = s = e \cdot s$$

pour tout $s \in S$. Un *monoïde* est un semi-groupe qui possède un neutre.

De manière usuelle, nous noterons $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ et $\mathbb{C}$ respectivement le semi-anneau des naturels, l'anneau des entiers, le champ des nombres rationnels, le champ des nombres réels et le champ des nombres complexes.

Pour tous $m, n \in \mathbb{Z}$, on notera $\{m, \dots, n\}$ l'ensemble des naturels k tels que $m \leq k \leq n$.

Un *alphabet* est un ensemble (fini ou infini) de lettres ou de symboles. Un *mot* fini sur un alphabet Σ est une suite finie de lettres. Par exemple, si on a l'alphabet $\Sigma = \{a, b\}$, les mots $aab, abba$ et $aaaa$ sont des mots finis sur Σ. Pour tout mot w, la longueur de w est

notée $|w|$. De plus, pour tout symbole a, on note $|w|_a$ le nombre d'occurrence de a dans le mot w Ainsi, $|aab| = 3$, $|abba| = |aaaa| = 4$ et $|aab|_a = |abba|_a = 2$. Le *mot vide* est le seul mot de longueur 0, noté ϵ.
Enfin, si x et y sont deux mots, on dit que x est un *préfixe* (resp. *suffixe*) de y si il existe un mot z tel que $xz = y$ (rep. $zx = y$). Un préfixe (resp. suffixe) de y sera dit *propre* s'il est distinct de y.

1.1.1 Définitions

Définition 1.1.1. Soient S un semi-groupe et X un sous-ensemble de S. Alors X est un *code* si on a la propriété

$$x_1x_2\cdots x_m = y_1y_2\cdots y_n \iff (x_1, x_2, \ldots, x_m) = (y_1, y_2, \ldots, y_n)$$

pour tous entiers m, $n \geq 1$ et éléments $x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n \in X$.

Cette définition montre en particulier que $m = n$ et que $x_i = y_i$ pour tout $i \in \{1, \ldots, m\}$.

Autrement dit, un sous-ensemble X d'un semi-groupe S est un code si et seulement si aucun élément de S ne possède plus d'une factorisation utilisant des éléments de X.

Pour tout semi-groupe S et tout sous-ensemble $X \subseteq S$, on note X^+ la clôture positive de X pour l'opération du semi-groupe S. Donc X^+ est le sous-semi-groupe de S engendré par X, et de ce fait, il est muni de l'opération de semi-groupe induite par l'opération de S.

Définition 1.1.2. Un semi-groupe S est dit *libre* s'il existe un code $X \subseteq S$ tel que $S = X^+$.

On dira donc qu'un semi-groupe est libre si et seulement si il est engendré par un code.

Pour tout monoïde M et tout sous-ensemble $X \subseteq M$, on note X^* l'ensemble X^+ muni du neutre de M. Un monoïde M est dit *libre* s'il existe un code $X \subseteq M$ tel que $M = X^*$.

Remarquons qu'en particulier aucun monoïde n'est un semi-groupe libre, car un monoïde contient un neutre et ne saurait donc pas être engendré par un code.

1.1.2 Illustrations

Soit Σ un alphabet, on note Σ^* l'ensemble de tous les mots sur Σ. L'ensemble Σ^* est un monoïde libre pour la concaténation, il possède le mot vide ϵ comme neutre et Σ comme code générateur. De même, l'ensemble de tous les mots non-vides sur Σ est noté Σ^+, qui est un semi-groupe libre car il ne possède pas de neutre. On dira qu'un *langage* sur Σ est un sous-ensemble de Σ^*.

Définition 1.1.3. Un *code préfixe* sur l'alphabet Σ est un sous-ensemble $X \subseteq \Sigma^+$ tel que pour tout $x \in X$ et tout $s \in \Sigma^+$, on a $xs \notin X$.

On a comme conséquence directe que tout code préfixe est un code pour la concaténation.

Dans la suite, nous considérerons le semi-groupe $\mathbb{W} = \{0, 1\}^*$.

Exemple 1.1.4. Les sous-ensembles $\{00, 01, 10, 11\}$, $\{01, 011, 11\}$ et $\{0^n1 : n \in \mathbb{N}\}$ du semi-groupe $\mathbb{W}$ sont des codes pour la concaténation, mais $\{01, 10, 0\}$ ne l'est pas car par exemple, l'élément 010 possède deux factorisations en éléments du sous-ensemble : $0(10) = (01)0$.

Pour tous semi-groupes S et S', on définit le *produit direct* de S et S' comme le produit Cartésien $S \times S'$ muni de l'opération de semi-groupe composante à composante, provenant des opérations de S et S' : pour tous éléments (x, x') et (y, y') de $S \times S'$, le produit $(x, x')(y, y')$ est donné par $(xy, x'y')$.

Exemple 1.1.5. Considérons le semi-groupe $\mathbb{W} \times \mathbb{W}$. Les sous-ensembles $\{(0, 1), (1, 0)\}$ et $\{(0, 0), (1, 01), (01, 10)\}$ de $\mathbb{W} \times \mathbb{W}$ sont des codes pour la concaténation composante à composante mais $\{(0, 0), (1, 101), (01, 01)\}$ ne l'est pas car par exemple, l'élément $(0101, 010101)$ possède deux factorisations : $(0, 0)(1, 101)(01, 01) = (01, 01)(0, 0)(1, 101)$.

Définition 1.1.6. Soit D un semi-anneau, notons $D^{d\times d}$ l'ensemble de toutes les matrices $d \times d$ à coefficients dans D. L'ensemble $D^{d\times d}$ est un semi-anneau pour les opérations matricielles usuelles, donc en particulier, $D^{d\times d}$ est un semi-groupe multiplicatif.

Exemple 1.1.7. Considérons le semi-groupe $\mathbb{N}^{2\times 2}$ et un entier $k > 1$, alors les sous-ensembles

$$\left\{ \begin{pmatrix} k & i \\ 0 & 1 \end{pmatrix} : i \in \{0, \ldots, k-1\} \right\} \text{ et } \left\{ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right\}$$

de $\mathbb{N}^{2\times 2}$ sont des codes pour la multiplication matricielle, mais le sous-ensemble

$$\left\{ \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \right\}$$

ne l'est pas car par exemple, l'élément $\left(\begin{smallmatrix} 1 & 6 \\ 0 & 2 \end{smallmatrix}\right)$ possède deux factorisations :

$$\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}.$$

De même, le sous-ensemble

$$\left\{ \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \right\}$$

de $\mathbb{N}^{2\times 2}$ n'est pas un code pour la multiplication matricielle car

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Démontrons que le sous-ensemble $X = \left\{ \begin{pmatrix} k & i \\ 0 & 1 \end{pmatrix} : i \in \{0, \ldots, k-1\} \right\}$ de $\mathbb{N}^{2\times 2}$ est un code pour la multiplication matricielle. Pour cela, considérons deux décompositions en éléments de X :

$$\begin{pmatrix} k & i_0 \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} k & i_{n-1} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} k & j_0 \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} k & j_{m-1} \\ 0 & 1 \end{pmatrix} \tag{1.1}$$

et montrons que $m = n$ et que $i_t = j_t$ pour tout $t \in \{0, \dots, n-1\}$ (avec $i_t, j_t \in \{0, \dots, k-1\}$ pour tout t).

Tout d'abord, l'équation (1.1) peut se réécrire

$$\begin{pmatrix} k^n & k^{n-1}i_{n-1} + \cdots + ki_1 + i_0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} k^m & k^{m-1}j_{m-1} + \cdots + kj_1 + j_0 \\ 0 & 1 \end{pmatrix}. \tag{1.2}$$

En effet, cela peut se démontrer par récurrence sur $t \geq 0$. En utilisant le cas de base $t = 0$, on obtient la matrice $\begin{pmatrix} k & i_0 \\ 0 & 1 \end{pmatrix}$, et pour l'induction on a

$$\begin{aligned} \begin{pmatrix} k & i_0 \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} k & i_{n-2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} k & i_{n-1} \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} k^{n-1} & k^{n-2}i_{n-2} + \cdots + ki_1 + i_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} k & i_{n-1} \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} k^n & k^{n-1}i_{n-1} + \cdots + ki_1 + i_0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

Ensuite, grâce à l'équation (1.2) on obtient le système suivant

$$\begin{cases} k^n = k^m \\ k^{n-1}i_{n-1} + \cdots + ki_1 + i_0 = k^{m-1}j_{m-1} + \cdots + kj_1 + j_0 \end{cases}.$$

Donc $n = m$ et $\mathrm{val}_k(i_{n-1} \cdots i_0) = \mathrm{val}_k(j_{n-1} \cdots j_0)$. Par unicité de la décomposition en base entière, on obtient que $i_t = j_t$ pour tout $t \in \{0, \dots, n-1\}$.

Proposition 1.1.8. *Tout sous-ensemble $X = \{A, B\}$ de $\mathbb{N}^{2\times 2}$ tel que les matrices A et B sont commutatives, n'est pas un code pour la multiplication matricielle.*

Démonstration. On a toujours

$$AB = BA,$$

donc un élément de $\mathbb{N}^{2\times 2}$ possède au moins deux factorisations avec des éléments de X. □

1.2 Un premier problème de décision

En informatique théorique, et plus précisément, en théorie de la calculabilité, *un problème de décision $\mathcal{P}$* est une question mathématique dont la réponse est soit "oui", soit "non". Formuler un problème de décision c'est se poser une question de décidabilité. Il s'agit en fait de rechercher l'existence d'un algorithme résolvant le problème et, s'il existe, de l'expliciter. Une méthode pour résoudre un problème de décision, donnée sous la forme d'un algorithme (procédure ayant un nombre fini d'étapes), est une *procédure de décision* du problème.

On appelle *instance positive* d'un problème toute entrée pour laquelle la réponse au problème de décision est "oui". Ainsi, un alphabet peut être utilisé pour formaliser les instances (positives et négatives) du problème. Les instances positives forment un langage sur cet alphabet. Ce problème sera dit *décidable* si le langage de ses instances positives est décidable, c'est-à-dire si la fonction caractéristique du langage est calculable. Un problème qui n'est pas décidable est dit *indécidable*.

Définition 1.2.1. Tout ensemble de nombres entiers est dit *récursif* s'il est décidable, c'est-à-dire si sa fonction caractéristique est calculable.

Définition 1.2.2. Soient $\mathcal{P}_1$ et $\mathcal{P}_2$ deux problèmes de décision, on dit qu'il existe une *transformation polynomiale* de $\mathcal{P}_1$ vers $\mathcal{P}_2$ s'il existe une fonction f, calculable en un temps polynomial, telle que i est une instance positive de $\mathcal{P}_1$ si et seulement si $f(i)$ est une instance positive de $\mathcal{P}_2$. On dira alors que $\mathcal{P}_1$ est une *réduction* de $\mathcal{P}_2$.

Ainsi, s'il existe une transformation polynomiale du problème $\mathcal{P}_1$ vers un problème $\mathcal{P}_2$ qui est décidable, alors $\mathcal{P}_1$ est lui aussi décidable.

Notre objectif pour le début de ce travail est d'étudier des problèmes de décision qui traitent du caractère libre de semi-groupes. C'est-à-dire étant donné un semi-groupe, déterminer si le problème de savoir si le semi-groupe est libre, est décidable.

Définition 1.2.3. Soit S un semi-groupe récursif, le *problème traitant du caractère libre* de S, que l'on notera $\textsc{Free}[S]$, est le suivant : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si X est un code. Pour tout nombre entier $k \geq 1$, on définit le problème $\textsc{Free}(k)[S]$ de la manière suivante : soit un sous-ensemble $X \subseteq S$ à k éléments, déterminer si X est un code. En particulier, pour tout entier $k \geq 1$, le problème $\textsc{Free}(k)[S]$ est une réduction de $\textsc{Free}[S]$, dans le sens où si on sait décider du problème $\textsc{Free}[S]$ alors on sait décider du problème $\textsc{Free}(k)[S]$.

Le résultat qui suit va nous permettre de caractériser les sous-ensembles qui ne sont pas des codes.

Proposition 1.2.4. *Pour tout alphabet Σ et tous $x, y \in \Sigma^+$ tels que $x \neq y$, les trois assertions suivantes sont équivalentes :*

(1) *$\{x, y\}$ n'est pas un code ;*

(2) *$xy = yx$, et*

(3) *il existe $s \in \Sigma^*$ et $p, q \in \mathbb{N}$ tels que $x = s^p$ et $y = s^q$.*

Démonstration. (3) $\Rightarrow$ (2) Soient $s \in \Sigma^*$ et $p, q \in \mathbb{N}$ tels que $x = s^p$ et $y = s^q$. Alors $xy = s^p s^q = s^{p+q} = s^{q+p} = s^q s^p = yx$.

(2) $\Rightarrow$ (1) Si $xy = yx$ avec $x \neq y$, alors par définition l'ensemble $\{x, y\}$ n'est pas un code.

(1) $\Rightarrow$ (3) Procédons par récurrence sur $|x| + |y|$.

- Cas de base : si $|x| + |y| = 2$, alors x et y sont deux lettres telles que $x \neq y$, donc l'ensemble $\{x, y\}$ est un code.

- Induction : supposons la propriété démontrée pour des mots x', y' tels que $|x'|+|y'| < |x|+|y|$ et montrons-la pour les mots x et y, en supposant (sans perte de généralité) que $|y| > |x|$. Supposons que l'ensemble $\{x, y\}$ n'est pas un code. Il existe alors des nombres naturels $n_1, \ldots, n_k, m_1, \ldots, m_k, n'_1, \ldots, n'_l, m'_1, \ldots, m'_l$ non nuls tels que

Cas 1 : $x^{n_1}y^{m_1}\cdots x^{n_k}y^{m_k} = x^{n'_1}y^{m'_1}\cdots x^{n'_l}y^{m'_l}$ avec $n_1 \neq n'_1$. Si on suppose $n'_1 > n_1$ (l'autre sens se règle de la même façon), alors l'égalité devient

$$\begin{aligned} & y^{m_1}\cdots x^{n_k}y^{m_k} = x^{n'_1-n_1}y^{m'_1}\cdots x^{n'_k}y^{m'_k} && (1.3)\\ \Leftrightarrow\ & yy^{m_1-1}\cdots x^{n_k}y^{m_k} = xx^{n'_1-n_1-1}y^{m'_1}\cdots x^{n'_k}y^{m'_k} && (1.4)\\ \Leftrightarrow\ & y\alpha_1 = x\alpha_2 && (1.5)\end{aligned}$$

en posant $\alpha_1 = y^{m_1-1}\cdots x^{n_k}y^{m_k}$ et $\alpha_2 = x^{n'_1-n_1-1}y^{m'_1}\cdots x^{n'_k}y^{m'_k}$. Comme on a supposé $|y| > |x|$, alors on sait qu'il existe un unique mot w tel que $y = xw$. En remplaçant y par xw dans l'équation (1.3), on obtient l'équation

$$(xw)^{m_1}x^{n_2}\cdots x^{n_k}(xw)^{m_k} = x^{n'_1-n_1}(xw)^{m'_1}\cdots x^{n'_k}(xw)^{m'_k}.$$

Si $w = x$, alors $s = x$ convient. Sinon, le membre de gauche débute par le mot xw alors que le membre de droite commence par le mot xx, donc l'ensemble $\{x, w\}$ n'est pas un code pour la concaténation. De plus, $|x| + |w| < |x| + |y|$. Donc par hypothèse de récurrence, on sait qu'il existe $s \in \Sigma^*$ et $p, q \in \mathbb{N}$ tels que $x = s^p$ et $w = s^q$. En particulier, on obtient que $y = xw = s^{p+q}$, et l'assertion (3) de l'énoncé est démontrée.

Cas 2 : $y^{m_1}x^{n_1}\cdots y^{m_k}x^{n_k} = y^{m'_1}x^{n'_1}\cdots y^{m'_l}x^{n'_l}$ avec $m_1 \neq m'_1$. Si on suppose $m'_1 > m_1$ (l'autre sens se règle de la même façon), alors l'égalité devient

$$\begin{aligned} & x^{n_1}\cdots y^{m_k}x^{n_k} = y^{m'_1-m_1}x^{n'_1}\cdots y^{m'_k}x^{n'_k}\\ \Leftrightarrow\ & xx^{n_1-1}\cdots y^{m_k}x^{n_k} = yy^{m'_1-m_1-1}x^{n'_1}\cdots y^{m'_k}x^{n'_k}\\ \Leftrightarrow\ & x\alpha'_1 = y\alpha'_2\end{aligned}$$

en posant $\alpha'_1 = x^{n_1-1}\cdots y^{m_k}x^{n_k}$ et $\alpha'_2 = y^{m'_1-m_1-1}x^{n'_1}\cdots y^{m'_k}x^{n'_k}$. Ensuite on se ramène au cas 1 car $|y| > |x|$ donc il existe un unique mot w tel que $y = xw$.

□

De façon plus générale, soient $\Sigma_1, \Sigma_2, \ldots, \Sigma_d$ d alphabets, ainsi que $x = (x_1, x_2, \ldots, x_d)$ et $y = (y_1, y_2, \ldots, y_d)$ deux éléments de $\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_d^*$. L'ensemble $\{x, y\}$ n'est pas un code si et seulement si $x_iy_i = y_ix_i$ pour tout $i \in \{1, \ldots, d\}$.

Ainsi, si Σ_i est fini pour tout $i \in \{1, \ldots, d\}$ alors le problème $\textsc{Free}(2)[\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_d^*]$ est décidable.

Dans le Chapitre 6, nous prouverons que le problème $\textsc{Free}[\mathbb{W} \times \mathbb{W}]$ est indécidable. De plus, il sera démontré dans le Chapitre 2 que pour tout entier $d \geq 1$, le problème de décision $\textsc{Free}(1)[\mathbb{Q}^{d\times d}]$ est décidable en temps polynomial.

1.3 Morphismes de semi-groupes

Définition 1.3.1. Soient S et S' deux semi-groupes, une fonction $\sigma\colon S \to S'$ est un *morphisme* de semi-groupes si pour tous x, $y \in S$, on a $\sigma(xy) = \sigma(x)\sigma(y)$.

Remarquons que si S et S' sont des monoïdes, un morphisme $\sigma : S \to S'$ n'enverra pas forcément le neutre de S sur le neutre de S'.

Soit $\sigma\colon S \to S'$ un morphisme de semi-groupes entre monoïdes. On remarque que le morphisme est complètement caractérisé par les images de σ sur les éléments d'un code de S.

Les deux propriétés suivantes seront fréquemment utilisées tout au long de ce travail.

Proposition 1.3.2. *(Propriété universelle). Soient Σ un alphabet et S un semi-groupe. Pour toute fonction $s\colon \Sigma \to S$, il existe un unique morphisme $\sigma\colon \Sigma^+ \to S$ tel que $\sigma(a) = s(a)$ pour toute lettre $a \in \Sigma$.*

Démonstration. Trivialement, pour toute fonction $s\colon \Sigma \to S$, tout morphisme σ de Σ^+ dans S sera tel que $\sigma(a) = s(a)$ pour toute lettre $a \in \Sigma$. Et pour les mots, on n'a pas d'autre choix que de poser $\sigma(ab) = \sigma(a)\sigma(b) = s(a)s(b)$. □

Proposition 1.3.3. *Soient Set S' deux semi-groupes, $\sigma : S \to S'$ un morphisme et X un sous-ensemble de S. Les deux assertions suivantes sont équivalentes :*

(1) *σ est injectif sur X et $\sigma(X)$ est un code ;*

(2) *σ est injectif sur X^+ et X est un code.*

Démonstration. (1) ⇒ (2) Soient $x_1, \ldots, x_n \in X$ et $y_1, \ldots, y_l \in X$ tels que

$$\sigma(x_1 \cdots x_n) = \sigma(y_1 \cdots y_l).$$

Comme σ est un morphisme, l'égalité se réécrit $\sigma(x_1) \cdots \sigma(x_n) = \sigma(y_1) \cdots \sigma(y_l)$. Par hypothèse, $\sigma(X)$ est un code donc l'égalité précédente implique que $n = l$ et $\sigma(x_i) = \sigma(y_i)$ pour tout $i \in \{1, \ldots, n\}$. De plus, σ est injectif sur X par hypothèse, donc $x_i = y_i$ $\forall i \in \{1, \ldots, n\}$. Ainsi, on a au final que $x_1 \cdots x_n = y_1 \cdots y_n$, donc σ est injectif sur X^+. Pour démontrer que X est un code, supposons avoir $x_1, \ldots, x_n \in X$ et $y_1, \ldots, y_l \in X$ tels que $x_1 \cdots x_n = y_1 \cdots y_l$. On a donc $\sigma(x_1 \cdots x_n) = \sigma(y_1 \cdots y_l)$ et comme σ est injectif sur X^+, on retourne à la première étape de cette démonstration.

(2) ⇒ (1) Si σ est injectif sur X^+ alors il l'est aussi sur X car $X \subseteq X^+$. De plus, pour démontrer que $\sigma(X)$ est un code, on considère des éléments $x_1, \ldots, x_n, y_1, \ldots y_l \in X$ tels que

$$\sigma(x_1) \cdots \sigma(x_n) = \sigma(y_1) \cdots \sigma(y_l).$$

Comme σ est un morphisme injectif sur X^+, l'égalité se réécrit $\sigma(x_1 \cdots x_n) = \sigma(y_1 \cdots y_l) \Leftrightarrow x_1 \cdots x_n = y_1 \cdots y_l$. Par hypothèse, X est un code donc l'égalité précédente implique que $n = l$ et $x_i = y_i$ pour tout $i \in \{1, \ldots n\}$. Enfin, on obtient que que $\sigma(x_i) = \sigma(y_i)$ pour tout $i \in \{1, \ldots n\}$. □

1.3.1 Problème de décision sur les morphismes

Soient S un semi-groupe récursif et Σ un alphabet fini. Grâce aux deux propriétés précédentes, on peut reformuler le problème FREE$[S]$ comme suit : étant donnés un alphabet fini Σ et un morphisme $\sigma : \Sigma^+ \to S$, déterminer si σ est injectif. De la même manière, pour tout entier positif k, une formulation alternative de FREE$(k)[S]$ serait : étant donnés un alphabet Σ de cardinalité k et un morphisme $\sigma : \Sigma^+ \to S$, déterminer si σ est injectif.

Un morphisme bijectif est appelé un *isomorphisme*. La fonction inverse de tout isomorphisme est aussi un isomorphisme. Ainsi, on va pouvoir reformuler certaines définitions : Un semi-groupe S est libre si et seulement si il existe un alphabet Σ et un isomorphisme de Σ^+ dans S. Un monoïde M est libre si et seulement si il existe un alphabet Σ et un isomorphisme de Σ^* dans M. Étant donnés un monoïde M et un alphabet Σ, tout morphisme de M dans Σ^* fait correspondre le neutre de M au mot vide.

Ces deux formulations du problème FREE$[S]$ seront utilisées alternativement.

1.4 D'autres problèmes de décision

Les problèmes de décision énoncés dans cette section sont liés à la combinatoire des semi-groupes. Il est intéressant de comparer leurs propriétés à celles du problème traitant du caractère libre.

1.4.1 Mortalité

Soit S un semi-groupe. Un *zéro* de S est un élément $z \in S$ tel que $zs = sz = z$ pour tout $s \in S$. Aucun semi-groupe n'a plus de un zéro. Pour tout semi-groupe récursif S possédant un zéro, notons MORTAL$[S]$ le problème suivant : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si le zéro de S appartient à X^+. Pour tout entier $k \geq 1$, notons MORTAL$(k)[S]$ la réduction de MORTAL$[S]$ pour des instances $X \subseteq S$ de cardinalité k.

Nous étudierons ce problème plus en détails dans le Chapitre 4 relatif aux matrices 2×2, ainsi que dans le Chapitre 6 relatif aux matrices 3×3.

1.4.2 Majoration

Soit d un entier strictement positif. Un sous-ensemble $X \subseteq \mathbb{Q}^{d \times d}$ est dit *borné* s'il existe un nombre positif constant a tel que le module de chacune des entrées de toute matrice de X est inférieur à a. Soit S un sous-ensemble récursif de $\mathbb{Q}^{d \times d}$, notons BOUNDED$[S]$ le problème suivant : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si X^+ est borné. Pour tout entier $k \geq 1$, BOUNDED$(k)[S]$ est la réduction de BOUNDED$[S]$ pour des instances $X \subseteq S$ de cardinalité k.

Nous démontrerons dans le Chapitre 2 que le problème BOUNDED$(1)[\mathbb{Q}^{d \times d}]$ est décidable.

1.4.3 Appartenance à un semi-groupe

Pour tout semi-groupe récursif S, notons MEMBER$[S]$ le problème suivant : étant donnés un sous-ensemble fini $X \subseteq S$ et un élément $a \in S$, déterminer si a appartient à X^+. Pour tout entier $k \geq 1$, MEMBER$(k)[S]$ est la réduction de MEMBER$[S]$ pour les instances (X, a) où la cardinalité de $X \subseteq S$ est égale à k.

Nous démontrerons aussi dans le Chapitre 2 que le problème MEMBER$(1)[\mathbb{Q}^{d\times d}]$ est décidable en temps polynomial.

1.4.4 Caractère fini d'un semi-groupe

Notons FINITE$[S]$ le problème suivant : étant donné un sous-ensemble fini $X \subseteq S$, déterminer si X^+ est fini. Pour tout entier $k \geq 1$, FINITE$(k)[S]$ est la réduction de FINITE$[S]$ pour des instances $X \subseteq S$ de cardinalité k.

Pour tout semi-groupe S récursif, FREE$(1)[S]$ est le problème complémentaire de FINITE$(1)[S]$.

1.4.5 Problème de correspondance de Post généralisé

Notons GPCP le problème suivant : étant donnés un alphabet fini Σ, deux morphismes $\sigma, \tau : \Sigma^* \to \mathbb{W}$ et $s, s', t, t' \in \mathbb{W}$, déterminer s'il existe un mot $w \in \Sigma^*$ tel que

$$s\sigma(w)t = s'\tau(w)t'.$$

Pour tout entier $k \geq 1$, GPCP(k) est la réduction de GPCP pour les instances $(\Sigma, \sigma, \tau, s, s', t, t')$ où la cardinalité de Σ est égale à k.

Ce problème sera largement étudié dans le Chapitre 5.

Chapitre 2

Matrices et morphismes de torsion

Ce chapitre est dédié aux problèmes de décision relatifs à des semi-groupes engendrés par un unique élément. On introduira donc la notion de torsion et dans un premier temps, nous nous consacrerons à l'étude des matrices à coefficients complexes qui sont de torsion.

Ensuite, nous étudierons plusieurs problèmes de décision, notamment le problème Matrix Torsion qui revient à déterminer si une matrice carrée $d \times d$ à coefficients rationnels est de torsion. On pourra démontrer que ce problème est décidable, et même en temps polynomial grâce à une réduction à un autre problème de décision. Cela nous permettra d'établir la décidabilité d'un deuxième problème.

Enfin, on étudiera le problème Morphism Torsion, qui revient à déterminer si un morphisme est de torsion. A la fin de ce chapitre, nous serons capables de démontrer que ce problème est décidable en temps polynomial. Cela nous permettra enfin d'établir que le problème Free(1)[hom(Σ^*)] est décidable lui aussi en temps polynomial.

2.1 Matrices de torsion à coefficients complexes

Nous allons nous intéresser à présent au cas des semi-groupes engendrés par un unique élément.

Définition 2.1.1. Soit S un semi-groupe, un élément $s \in S$ est dit *de torsion* s'il satisfait une des quatre conditions équivalentes suivantes :

(1) le singleton $\{s\}$ n'est pas un code ;

(2) il existe deux entiers p et q avec $0 \leq p < q$ tels que $s^p = s^q$;

(3) le semi-groupe $\{s, s^2, s^3, s^4, ...\}$ a une cardinalité finie ;

(4) la suite $(s, s^2, s^3, s^4, ...)$ finit par être périodique.

Le théorème suivant caractérise les matrices carrées à coefficients complexes qui sont de torsion. Rappelons d'abord un fait élémentaire d'algèbre linéaire.

Lemme 2.1.2. *Soient M une matrice carrée à coefficients complexes et λ une valeur propre de M. La multiplicité de λ comme racine du polynôme minimal de M est égale à l'ordre maximal d'un bloc de Jordan de M correspondant à la valeur propre λ.*

Théorème 2.1.3. *Soient d un entier positif et $M \in \mathbb{C}^{d\times d}$. Les quatre assertions suivantes sont équivalentes :*

(*i*) *la matrice M est de torsion ;*

(*ii*) *il existe $v \in \{0, \dots, d\}$ et un ensemble fini U de racines de l'unité tels que le polynôme minimal de M sur $\mathbb{C}$ est égal à*

$$x^v \prod_{u\in U} (x-u);$$

(*iii*) *il existe une matrice diagonale D et une matrice nilpotente N telles que chaque valeur propre de D est une racine de l'unité et telles que la matrice*

$$\begin{pmatrix} D & O \\ O & N \end{pmatrix}$$

est une forme normale de Jordan de M ;

(*iv*) *il existe un entier $n \geq 2$ tel que $M^d = M^{nd}$.*

Démonstration. (*i*) $\Rightarrow$ (*ii*) : Supposons que la matrice M soit de torsion. Alors il existe deux entiers p et q tels que $0 \leq p < q$ et $M^p = M^q$. Notons $\mu(z)$ le polynôme minimal de M. Comme $M^q - M^p$ est une matrice nulle, alors $\mu(x)$ divise $x^q - x^p = x^p(x^{q-p} - 1)$ car ce polynôme est annulé par la matrice M. Ainsi, les zéros de l'unité apparaissent une seule fois et on peut écrire le polynôme minimal de M sous la forme

$$\mu(x) = x^v \prod_{u\in U} (x-u) \text{ avec } v \in \{0, \dots, p\} \text{ et } U \subseteq \{u \in \mathbb{C} : u^{q-p} = 1\}.$$

De plus, par le théorème de Cayley-Hamilton on sait que toute matrice annule son polynôme caractéristique. Donc cela implique que $\mu(x)$ divise le polynôme caractéristique de M, qui est de degré d, donc v pourra prendre au maximum la valeur d. Ainsi l'assertion (*ii*) est démontrée.

(*ii*) $\Rightarrow$ (*iii*) : Cela découle immédiatement du Lemme 2.1.2 car les zéros du polynôme minimal de M sont 0 et des racines de l'unité (qui sont toutes de multiplicité 1). Ainsi on crée la matrice diagonale D telle que chacune de ses valeurs propres correspond à une racine de l'unité et la matrice N correspondant à la valeur propre 0. On obtient donc une matrice nilpotente

$$N = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}.$$

$(iii) \Rightarrow (iv)$: Comme la matrice $\left(\begin{smallmatrix} D & O \\ O & N \end{smallmatrix}\right)$ est une forme normale de Jordan de M, il existe une matrice inversible P telle que

$$M = P \begin{pmatrix} D & O \\ O & N \end{pmatrix} P^{-1}.$$

Soit m un entier strictement positif tel que $\lambda^m = 1$ pour toute valeur propre λ de D. Alors D^m est la matrice identité, donc on a $D^{(m+1)d} = (D^m)^d D^d = D^d$. De plus, comme la matrice N est nilpotente, $N^{(m+1)d}$ et N^d sont égales à la même matrice nulle. Au final, on a

$$M^{(m+1)d} = P \begin{pmatrix} D^{(m+1)d} & O \\ O & N^{(m+1)d} \end{pmatrix} P^{-1} = P \begin{pmatrix} D^d & O \\ O & N^d \end{pmatrix} P^{-1} = M^d.$$

On obtient donc l'assertion (iv) en posant $n = m + 1$.

$(iv) \Rightarrow (i)$: cela se voit clairement en prenant $p = d$ et $q = nd$ et en utilisant le point (2) de la Définition 2.1.1. □

Dans la suite, nous nous servirons principalement de l'équivalence entre les assertions (i) et (iv) du Théorème 2.1.3.

2.2 Problème de décision sur les matrices de torsion

Concentrons-nous maintenant sur des matrices à coefficients rationnels.

Définition 2.2.1. Notons ϕ la *fonction indicatrice d'Euler* : pour tout entier $n \geq 1$, $\phi(n)$ donne le nombre de $k \in \{1, \ldots, n\}$ tels que k et n sont premiers entre eux.

Définition 2.2.2. Soit n un nombre naturel, on appelle Φ_n le n-ième *polynôme cyclotomique* défini comme étant le polynôme unitaire dont les racines complexes sont les racines primitives n-ièmes de l'unité.

Proposition 2.2.3. *Pour tout entier $n \geq 1$, le degré du n-ième polynôme cyclotomique est égal à $\phi(n)$.*

Démonstration. Soit $S = \{x \in \mathbb{C} : x^n = 1\}$ l'ensemble des racines n-ièmes de l'unité. Le cardinal de S est égal à n et on peut écrire $S = \{\exp(\frac{2ik\pi}{n}) : k \in \{0, \ldots, n\}\}$. On appelle racine primitive nième de l'unité tout élément de S l'engendrant. Ainsi, si k est un entier compris entre 0 et n, on peut dire que $\exp(\frac{2ik\pi}{n})$ est une racine primitive nième de l'unité si et seulement si $\text{pgcd}(k, n) = 1$ (c'est-à-dire si k et n sont premiers entre eux). De part cela et du fait de la simplicité de ses racines, le polynôme cyclotomique de degré n est un polynôme de degré égal au cardinal de l'ensemble $\{k \in \{0, \ldots, n-1\} : \text{pgcd}(k, n) = 1\}$, c'est-à-dire égal à $\phi(n)$. □

Définition 2.2.4. On définit le problème Matrix Torsion comme suit : étant donnés un entier $d \geq 1$ et une matrice $M \in \mathbb{Q}^{d \times d}$, déterminer si M est de torsion.

Remarque 2.2.5. Pour tout entier $d \geq 1$, le problème complémentaire de FREE(1)$[\mathbb{Q}^{d\times d}]$ (c'est-à-dire, étant donné un sous-ensemble $X \subseteq \mathbb{Q}^{d\times d}$ à 1 élément, déterminer si X n'est pas un code) est une réduction de MATRIX TORSION étant donné la Définition 2.1.1.

Théorème 2.2.6. *Soit la fonction calculable* $r\colon \mathbb{N}\setminus\{0\} \to \mathbb{N}\setminus\{0\} : n \mapsto ppcm\{m \in \mathbb{N} : \phi(m) \leq n\}$. *Alors pour tout entier* $d \geq 1$ *et pour toute matrice* $M \in \mathbb{Q}^{d\times d}$, *on a* $M^d = M^{d+r(d)}$ *si et seulement si* M *est de torsion.*

Démonstration. La fonction r est calculable. Pour la condition suffisante, comme M est une matrice de torsion, on sait qu'il existe des nombres naturels $p < q$ tels que $M^p = M^q$. Si on note m_M le polynôme minimum de M, alors m_M divise $x^p(x^{q-p}-1)$. Comme $x^{q-p}-1$ est un produit de polynômes cyclotomiques distincts et qu'ils sont irréductibles sur $\mathbb{Q}$, alors $m_M = x^s\Phi_{n_1}\cdots\Phi_{n_k}$, où $n_1, \ldots, n_k$ sont distincts. Comme m_M est de degré au plus d et que Φ_n est de degré $\phi(n)$, alors m_M divise $x^d(x^{r(d)}-1)$. Donc $M^d = M^{d+r(d)}$.

La réciproque est évidente étant donné la Définition 2.1.1. □

Corollaire 2.2.7. *Le problème* MATRIX TORSION *est décidable. De plus, étant donné la remarque 2.2.5, le problème* FREE(1)$[\mathbb{Q}^{d\times d}]$ *est décidable en temps polynomial pour tout nombre entier* $d \geq 1$.

Par contre on ne peut pas conclure immédiatement que le problème MATRIX TORSION est décidable en temps polynomial car r n'est pas majoré par un polynôme.

Remarque 2.2.8. Grâce au Corollaire 2.2.7, le problème FREE(1)$[\mathbb{Q}^{d\times d}]$ est décidable en temps polynomial, et comme FREE(1)$[S]$ est le problème complémentaire de FINITE(1)$[S]$ pour tout semi-groupe S, alors le problème FINITE(1)$[\mathbb{Q}^{d\times d}]$ est décidable.

Définition 2.2.9. Pour tout nombre $k \in \mathbb{N}$, on définit le problème MP(k) comme suit : étant donnés un nombre entier $d \geq 1$ et deux matrices A, B $\in \mathbb{Q}^{d\times d}$, déterminer s'il existe un nombre $n \in \mathbb{N}$ tel que $A = B^{n+k}$.

Théorème 2.2.10. (KANNAN ET LIPTON *[11]*) *Le problème* MP(0) *est décidable en temps polynomial.*

Corollaire 2.2.11. *Pour tout* $k \in \mathbb{N}$, *le problème* MP(k) *est décidable en temps polynomial.*

Démonstration. Par le théorème précédent, il suffit de montrer qu'il existe une transformation polynomiale du problème MP(k) au problème MP(0). Notons O_k la matrice nulle $k \times k$ et N_k la matrice $k \times k$ définie comme suit : pour tous indices $i, j \in \{1, \ldots, k\}$, la $(i,j)^{\text{ème}}$ entrée de N_k est égale à 1 si $j - i = 1$ et vaut 0 sinon. On voit donc que pour tout $n \in \mathbb{N}$, on a $N_k^n = O_k$ si et seulement si $n \geq k$, car la matrice N_k^{k-1} est composée uniquement de zéros et d'un 1 en position supérieure droite.

Soit (d, A, B) une instance du problème MP(k). Définissons deux matrices $C, D \in \mathbb{Q}^{(k+d)\times(k+d)}$ de la manière suivante :

$$C = \begin{pmatrix} A & O \\ O & O_k \end{pmatrix} \text{ et } D = \begin{pmatrix} B & O \\ O & N_k \end{pmatrix}.$$

Considérons $(k+d, C, D)$ une instance de MP(0) et $(k+d, C, D)$ est calculable à partir de (d, A, B) en temps polynomial. Pour tout $n \in \mathbb{N}$, on a $C = D^n$ si et seulement si $A = B^n$ et $n \geq k$. Donc, (d, A, B) est une instance positive de MP(k) si et seulement si $(k+d, C, D)$ est une instance positive de MP(0). □

Théorème 2.2.12. *Le problème* Matrix Torsion *est décidable en temps polynomial.*

Démonstration. On sait que pour tout $k \in \mathbb{N}$ le problème MP(k) est décidable en temps polynomial, donc il suffit de montrer qu'il existe une transformation polynomiale de Matrix Torsion vers MP(2).

Soit (d, M) une instance de Matrix Torsion. Alors (d, M) est une instance positive si et seulement si l'instance (d, M^d, M^d) du problème MP(2) est une instance positive. En effet, si (d, M) est une instance positive de Matrix Torsion alors la matrice M est de torsion. Par le Théorème 2.1.3, on sait que cette condition est équivalente au fait qu'il existe un entier $n \geq 2$ tel que $M^d = M^{nd}$. Ainsi, l'instance (d, M^d, M^d) est une instance positive du problème MP(2) car le nombre naturel $m = n - 2$ convient : $(M^d)^{m+2} = (M^d)^n = M^d$. Réciproquement, si (d, M^d, M^d) est une instance positive du problème MP(2) alors il existe $m \in \mathbb{N}$ tel que $M^d = (M^d)^{m+2}$. En posant $n = m + 2$ on obtient l'équation $M^d = M^{nd}$, donc on peut appliquer le Théorème 2.1.3. Ainsi, la matrice M est de torsion et (d, M) est une instance positive de Matrix Torsion. On a donc obtenu une réduction polynomiale de Matrix Torsion à MP(2), qui est décidable en temps polynomial vu le Corollaire 2.2.11. □

Proposition 2.2.13. *Pour tout nombre entier $d \geq 1$, le problème* Member$(1)[\mathbb{Q}^{d\times d}]$ *peut être vu comme une réduction du problème* MP(1)*. C'est donc un problème décidable en temps polynomial.*

Démonstration. Soit (d, A, B) une instance positive de MP(1), c'est-à-dire telle que $A, B \in \mathbb{Q}^{d\times d}$ et telle qu'il existe un nombre $n \in \mathbb{N}$ tel que $A = B^{n+1}$. Ainsi, l'entrée $(X = \{B\}, A)$ est une instance positive du problème Member$(1)[\mathbb{Q}^{d\times d}]$ car $A \in X^+ = \{B\}^+$ car on sait qu'il existe $n \in \mathbb{N}$ tel que $A = B^{n+1}$, donc $A \in \{B, B^2, B^3, B^4, \ldots\}$.

Inversement, considérons (B, A) une instance positive de Member$(1)[\mathbb{Q}^{d\times d}]$, c'est-à-dire telle que $A, B \in \mathbb{Q}^{d\times d}$ et $A \in \{B\}^+$. Alors il existe $n > 0$ tel que $A = B^n$. Donc il suffit de poser $m = n - 1 \in \mathbb{N}$ et on a $A = B^{m+1}$. Ainsi, (d, A, B) est une instance positive de MP(1). □

Définition 2.2.14. Une matrice carrée M à coefficients complexes est dite *bornée en puissance* si le semi-groupe $\{M, M^2, M^3, M^4, ...\}$ est borné. On définit ainsi le problème MATRIX POWER BOUNDEDNESS : étant donnés un entier $d \geq 1$ et une matrice $M \in \mathbb{Q}^{d\times d}$, déterminer si M est borné en puissance.

Remarquons que pour tout entier $d \geq 1$, le problème BOUNDED(1)$[\mathbb{Q}^{d\times d}]$ est une réduction de MATRIX POWER BOUNDEDNESS. En effet, soit (d, M) une instance de MATRIX POWER BOUNDEDNESS, (d, M) est une instance positive si et seulement si $M \in \mathbb{Q}^{d\times d}$ et le semi-groupe $\{M, M^2, M^3, M^4, ...\}$ est borné. C'est-à-dire si et seulement si $(X = \{M\})$ est une instance positive de BOUNDED(1)$[\mathbb{Q}^{d\times d}]$ car X^+ est borné.

Lemme 2.2.15. *Considérons une matrice carrée M à coefficients complexes. Notons $\mu(x)$ le polynôme minimum de M et $\nu(x) = pgcd(\mu(x), \mu'(x))$. Alors M est bornée en puissance si et seulement si les deux assertions suivantes sont satisfaites :*

(i) chaque racine de $\mu(x)$ est en module plus petite ou égale à 1 et

(ii) chaque racine de $\nu(x)$ est en module strictement plus petite que 1.

Démonstration. Les racines de μ sont les valeurs propres de M et les racines de ν sont les racines de multiplicité au moins 2 de μ. Si on note $\lambda_1, \ldots, \lambda_p$ les valeurs propres de M alors on peut écrire

$$(M^n)_{ij} = \sum_{k=1}^{p} P_{ij}^{(k)}(n)\lambda_k^n$$

pour tout $n \in \mathbb{N}$, avec deg $(P^{(k)}) < m_k =$ multiplicité de λ_k comme zéro du polynôme minimum μ de M. Si on suppose que les conditions (i) et (ii) de l'énoncé sont satisfaites, alors il existe une constante $C > 0$ telle que pour tous $i, j \in \mathbb{N}$ on a $(M^n)_{ij} < C$. Car les racines simples de μ sont en module plus petites ou égales à 1 et les polynômes $P^{(k)}$ correspondants sont des constantes. De plus, les autres racines sont des racines de ν et sont en module strictement plus petites que 1, et donc le terme $P_{ij}^{(k)}(n)\lambda_k^n$ tend vers 0. L'élément $(M^n)_{ij}$ est donc borné pour tout $n \in \mathbb{N}$.

Réciproquement, supposons qu'il existe une constante $B > 0$ telle que pour tout $n \geq 0$ et pour tous i, j, on a $|(M^n)_{ij}| < B$. Soit J une forme de Jordan associée à M. Il existe donc une matrice inversible S telle que $M = S^{-1}JS$. Soient n, i, j fixés, on a

$$(J^n)_{ij} = (SM^nS^{-1})_{ij} = \sum_{k,l} S_{ik}(M^n)_{kl}S_{lj}.$$

On obtient que

$$|(J^n)_{ij}| \leq \sum_{k,l} |S_{ik}| \cdot B \cdot |S_{lj}|.$$

Les coefficients $(J^n)_{ij}$ sont donc bornés. Maintenant, montrons que (i) et (ii) doivent être vérifiés. Pour chaque valeur propre λ, notons J_λ un bloc de Jordan de dimension maximale dans J.

Soit λ une valeur propre dont la multiplicité comme zéro du polynôme minimum vaut 1. Alors $(J_\lambda)^n = (\lambda^n)$. De ce qui précède, on obtient que λ^n doit être borné, et donc que $|\lambda| \leq 1$. (i) est donc vérifié.

Soit λ une valeur propre dont la multiplicité comme zéro du polynôme minimum est supérieure ou égale à 2. Alors

$$J_\lambda = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda \end{pmatrix}.$$

Par récurrence sur n, on peut montrer que pour tous n, i, j, on a $(J_\lambda^n)_{ij} = \binom{n}{n+i-j}\lambda^{n+i-j}$.

En effet, si $n = 1$, on a $(J_\lambda)_{ij} = \begin{cases} \lambda \text{ si } i-j=0 \\ 1 \text{ si } i-j=-1 \\ 0 \text{ si } i-j>0 \text{ ou } i-j<-1 \end{cases}$

De même,

- si $i-j=0$, alors $\begin{pmatrix} 1 \\ 1+i-j \end{pmatrix}\lambda^{1+i-j} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}\lambda^1 = \lambda$.
- si $i-j=-1$, alors $\begin{pmatrix} 1 \\ 1+i-j \end{pmatrix}\lambda^{1+i-j} = 1$.
- si $i-j>0$ ou $i-j<-1$, alors $\begin{pmatrix} 1 \\ 1+i-j \end{pmatrix}\lambda^{1+i-j} = \dfrac{1}{(1+i-j)!(j-i)!}\lambda^{1+i-j} = 0$.

Donc $(J_\lambda)_{ij} = \begin{pmatrix} 1 \\ 1+i-j \end{pmatrix}\lambda^{1+i-j}$.

Ensuite pour l'induction, on a

$$\begin{aligned} (J_\lambda^{n+1})_{ij} &= (J_\lambda^n J_\lambda)_{ij} \\ &= \sum_k (J_\lambda^n)_{ik}(J_\lambda)_{kj} \\ &= (J_\lambda^n)_{ij}(J_\lambda)_{jj} + (J_\lambda^n)_{i(j-1)}(J_\lambda)_{(j-1)j} \\ &= \begin{pmatrix} n \\ n+i-j \end{pmatrix}\lambda^{n+i-j}\lambda + \begin{pmatrix} n \\ n+i-(j-1) \end{pmatrix}\lambda^{n+i-(j-1)}1 \\ &= \begin{pmatrix} n+1 \\ n+1+i-j \end{pmatrix}\lambda^{n+1+i-j}. \end{aligned}$$

On obtient bien l'égalité attendue.

En particulier, on a $(J_\lambda^n)_{12} = n\lambda^{n-1}$. De ce qui précède, on obtient que $n\lambda^{n-1}$ doit être borné, et donc que $|\lambda| < 1$. (ii) est donc vérifié. □

Proposition 2.2.16. *Le problème* MATRIX POWER BOUNDEDNESS *est décidable.*

Démonstration. Considérons une matrice carrée M à coefficients rationnels, $\mu(x)$ son polynôme minimum et notons $\nu(x) = \text{pgcd}(\mu(x), \mu'(x))$. Alors $\mu(x)$ et $\nu(x)$ sont calculables à partir de M en temps polynomial [11]. En effet, l'algorithme suivant nous permet de calculer le polynôme minimum d'une matrice donnée :

Soit une matrice M de dimension $n \times n$.
Initialisation : $i \leftarrow 1$.
Calculer $M^2, M^3, \ldots, M^n$.
Tant que $i \leq n$, répéter :
Si le système linéaire de n^2 équations

$$\sum_{j=0}^{i} y_j M^j = 0$$

possède une solution $y = (y_0, y_1, \ldots, y_i)$ dans les rationnels, avec $y_i \neq 0$,
rendre $\mu_M(x) = \sum_{j=0}^{i} y_j x^j$
$i \leftarrow i + 1$.
Fin de la procédure.

Cet algorithme nous rend le polynôme de degré minimal, qui est un multiple du polynôme minimum. De plus, la procédure s'effectue en temps polynomial (en la longueur de l'entrée n) car la solution du système d'équations linéaires peut être obtenue en temps polynomial.

Ensuite pour calculer $\nu(x)$, il reste à calculer le pgcd de $\mu(x) = \sum_{i=0}^{n} a_i x^i$ et de sa dérivée, que l'on calcule comme suit

$$\mu'(x) = \sum_{i=1}^{n} i a_i x^{i-1}$$

où les a_i sont les coefficients du polynôme $\mu(x)$.

Ainsi, étant donné le Lemme 2.2.15, déterminer si M est borné en puissance revient à vérifier les conditions (i) et (ii). Cela peut se faire en utilisant la procédure de décision de Tarski. En effet, grâce aux algorithmes précédents on sait calculer les polynômes ν et μ. De plus, les deux assertions (i) et (ii) sont des formules du premier ordre donc grâce à la procédure d'élimination des quantificateurs de Tarski [21], on arrive à décider si les deux formules sont vraies en même temps ou pas (c'est-à-dire déterminer si M est borné en puissance). □

2.3 Problème de décision sur les morphismes de torsion

Définition 2.3.1. Pour tout alphabet Σ, notons $\hom(\Sigma^*)$ l'ensemble de tous les morphismes de Σ^* dans lui-même. On définit ainsi le problème MORPHISM TORSION suivant :

étant donnés un alphabet fini Σ et un morphisme $\sigma \in \hom(\Sigma^*)$, déterminer si σ est de torsion (pour la composition de fonctions).

La taille d'une entrée (Σ, σ) de MORPHISM TORSION est égale à $\sum_{a\in\Sigma}(1 + |\sigma(a)|)$.

Définition 2.3.2. Soient Σ un alphabet fini, d la cardinalité de Σ et $a_1, a_2, ..., a_d$ tels que $\Sigma = \{a_1, a_2, ..., a_d\}$. La *matrice d'incidence* de σ (relative à l'ordre choisi des lettres de Σ) est définie par

$$\begin{pmatrix} |\sigma(a_1)|_{a_1} & |\sigma(a_2)|_{a_1} & \cdots & |\sigma(a_d)|_{a_1} \\ |\sigma(a_1)|_{a_2} & |\sigma(a_2)|_{a_2} & \cdots & |\sigma(a_d)|_{a_2} \\ \vdots & \vdots & \ddots & \vdots \\ |\sigma(a_1)|_{a_d} & |\sigma(a_2)|_{a_d} & \cdots & |\sigma(a_d)|_{a_d} \end{pmatrix} \in \mathbb{N}^{d\times d}.$$

Pour tous $i, j \in \{1, \ldots, d\}$, la $(i, j)^{\text{ème}}$ entrée de la matrice d'incidence est égale au nombre d'occurrence de a_i dans $\sigma(a_j)$.

Proposition 2.3.3. *Soit Σ un alphabet fini. Pour chaque morphisme $\sigma \in hom(\Sigma^*)$, notons P_σ la matrice d'incidence de σ. Alors*

(i) on a l'égalité $P_\sigma P_\tau = P_{\sigma\tau}$ pour tous $\sigma, \tau \in hom(\Sigma^*)$;

(ii) pour tout $P \in \mathbb{N}^{d\times d}$, il existe un nombre fini de morphismes $\tau \in hom(\Sigma^*)$ tels que $P_\tau = P$.

Démonstration. Pour démontrer le point (i), on aimerait obtenir l'égalité

$$|(\sigma\tau)(a_j)|_{a_i} = \sum_{k=1}^{d} |\sigma(a_k)|_{a_i} |\tau(a_j)|_{a_k}$$

pour tous $i, j \in \{1, \ldots, d\}$. On sait qu'il existe des lettres $b_1, \ldots, b_n \in \Sigma$ telles que $\tau(a_j) = b_1 \cdots b_n$. Ensuite $\sigma(\tau(a_j)) = \sigma(b_1 \cdots b_n) = \sigma(b_1) \cdots \sigma(b_n)$. Donc on obtient que

$$|\sigma(\tau(a_j))|_{a_i} = \sum_{l=1}^{n} |\sigma(b_l)|_{a_i} = \sum_{k=1}^{d} \sum_{\substack{l=1 \\ b_l = a_k}}^{n} |\sigma(a_k)|_{a_i} = \sum_{k=1}^{d} |\tau(a_j)|_{a_k} |\sigma(a_k)|_{a_i}.$$

Pour le point (ii), on se convainc facilement que pour toute matrice à coefficients naturels, on trouve un nombre fini de morphismes dont la matrice d'adjacence est égale à la matrice de départ. Par exemple, si on considère l'alphabet fini $\Sigma = \{a_1, a_2\}$ et la matrice

$$P = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix},$$

on observe que les morphismes $\tau \in \hom(\Sigma^*)$ possibles pour avoir $P = P_\tau$ sont tels que $\tau(a_1) = a_1a_2a_2a_2$ et $\tau(a_2) = a_1a_1a_2$ à permutations près. On en a donc un nombre fini. □

Théorème 2.3.4. *Le problème* Morphism Torsion *est décidable en temps polynomial.*

Démonstration. Vu le Théorème 2.2.12, il suffit de montrer qu'il existe une réduction polynomiale de Morphism Torsion à Matrix Torsion. Pour cela, nous allons montrer qu'un morphisme est de torsion si et seulement si sa matrice d'incidence l'est.

Soient (Σ, σ) une instance de Morphism Torsion et d la cardinalité de Σ. Pour chaque morphisme $\tau \in \hom(\Sigma^*)$, notons P_τ la matrice d'incidence de τ. Clairement, (d, P_σ) est une instance de Matrix Torsion calculable en temps polynomial.

Vérifions que (Σ, σ) est une instance positive de Morphism Torsion si et seulement si (d, P_σ) est une instance positive de Matrix Torsion. Vu le point (i) de la Proposition 2.3.3, $P_\sigma^n = P_\sigma P_\sigma \cdots P_\sigma = P_{\sigma^n}$ pour tout $n \in \mathbb{N}$. Donc, si σ est de torsion, alors P_σ le sera aussi, car s'il existe $p < q$ tels que $\sigma^p = \sigma^q$ alors $P_\sigma^p = P_{\sigma^p} = P_{\sigma^q} = P_\sigma^q$. Inversement, si on suppose que P_σ est de torsion, alors l'ensemble de matrices $\mathcal{P} = \{P_\sigma, P_\sigma^2, P_\sigma^3, P_\sigma^4, \cdots\}$ est fini. Par le point (ii) de la Proposition 2.3.3, il existe un nombre fini de morphismes $\tau \in \hom(\Sigma^*)$ tels que $P_\tau \in \mathcal{P}$. Comme $P_{\sigma^n} \in \mathcal{P}$ pour tout entier $n \geq 1$, alors l'ensemble $\{\sigma, \sigma^2, \sigma^3, \sigma^4, ...\}$ est fini, et donc σ est de torsion par définition. □

Corollaire 2.3.5. *Pour tout alphabet Σ, le problème* Free(1)[$hom(\Sigma^*)$] *est décidable en temps polynomial.*

Démonstration. Ce problème se réduit au problème complémentaire de Morphism Torsion car un morphisme $\sigma \in \hom(\Sigma^*)$ est de torsion si et seulement si ce n'est pas un code. □

Pour tout alphabet fini Σ contenant au moins un élément et pour tout nombre entier $k > 1$, la décidabilité du problème Free(k)[hom(Σ^*)] n'est pas encore prouvée. On abordera la question de la décidabilité du problème Free(2)[hom($\mathbb{W}$)] dans le Chapitre 4.

Chapitre 3

Le cas des groupes et du produit direct de semi-groupes

Dans ce chapitre, nous allons tout d'abord démontrer que pour tout semi-groupe S et tout sous-ensemble $X \subseteq S$ contenant au moins un élément, X n'est pas un code si et seulement si les éléments de X satisfont une équation non triviale, qui sera dite *équilibrée*.

On examinera ensuite plusieurs conséquences de ce résultat qui feront appel au caractère simplifiable des semi-groupes ainsi qu'à leur produit direct. Nous nous concentrerons sur des matrices à coefficients rationnels et à coefficients entiers pour établir un premier résultat important : le problème $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$ se réduit au problème $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$ pour tous nombre entiers $k, d \geq 1$.

Nous terminerons ce chapitre par l'étude du problème de décision traitant du caractère libre dans le cas des groupes. Pour cela, nous introduirons le problème d'acceptation qui revient à déterminer si un automate donné accepte un élément d'un groupe donné. Grâce à l'étude de ce problème, on démontrera un deuxième résultat important qui traite de la décidabilité de problème $\textsc{Free}[G]$ où G est un groupe.

3.1 Équations équilibrées

Une partie du chapitre repose sur les conséquences du lemme qui suit :

Lemme 3.1.1. *Soient S un semi-groupe et X un sous-ensemble de S de cardinalité plus grande que 1. L'ensemble X n'est pas un code si et seulement si il existe $x, x' \in X$ et $z, z' \in X^+$ tels que $x \neq x'$ et $zxzx'z' = zx'z'xz$.*

Démonstration. Soient Σ un alphabet et $\sigma : \Sigma^+ \to S$ un morphisme tels que σ induit une bijection de Σ dans X. Supposons que X n'est pas un code. Vu la Proposition 1.3.3, σ n'est pas injectif. Par conséquent, il existe $w, w' \in \Sigma^+$ tels que $w \neq w'$ et $\sigma(w) = \sigma(w')$.

- Si on suppose que w n'est pas un préfixe de w' et que w' n'est pas un préfixe de w, alors il existe $a, a' \in \Sigma$ et $u, v, v' \in \Sigma^*$ tels que $a \neq a'$, $w = uav$ et $w' = ua'v'$. Donc

u est le plus long préfixe commun de w et w'. Il suffit ensuite de voir que choisir $\sigma(a), \sigma(a'), \sigma(vau)$ et $\sigma(v'au)$ pour x, x', z et z' respectivement est un bon choix. En effet, on a

$$\begin{aligned} zxzx'z' &= \sigma(vau)\sigma(a)\sigma(vau)\sigma(a')\sigma(v'au) \\ &= \sigma(vauavaua'v'au) \\ &= \sigma(va)\sigma(uav)\sigma(a)\sigma(ua'v')\sigma(au) \\ &= \sigma(va)\sigma(ua'v')\sigma(a)\sigma(uav)\sigma(au) \\ &= \sigma(vaua'v'auavau) \\ &= \sigma(vau)\sigma(a')\sigma(v'au)\sigma(a)\sigma(vau) \\ &= zx'z'xz. \end{aligned}$$

- Si on suppose à présent que w est un préfixe propre de w' (le cas où w' serait préfixe propre de w se règle de la même façon car w et w' jouent des rôles symétriques). Alors il existe $a \in \Sigma$ tel que wa est un préfixe de w'. Comme $\sigma : \Sigma \to X$ est une bijection, alors Σ et X ont le même nombre d'éléments. De plus, la cardinalité de X est plus grande que 1 donc celle de Σ aussi, ainsi il existe $b \in \Sigma$ tel que $a \neq b$. Clairement, on a $\sigma(wb) = \sigma(w)\sigma(b) = \sigma(w')\sigma(b) = \sigma(w'b)$. Mais wb n'est pas un préfixe de $w'b$ car wa est un préfixe de w' et $a \neq b$. De plus, $w'b$ n'est pas un préfixe de wb. Ainsi, le second cas se réduit au premier.

La réciproque est directe vu la Définition 1.1.1. □

Remarque 3.1.2. Pour tout élément $y \in X^+$, les factorisations de y sur X sont en correspondance une à une avec les pré-images de y par la bijection σ utilisée dans la preuve précédente. Soient $\overline{x}, \overline{x}' \in \Sigma$ et $\overline{z}, \overline{z}' \in \Sigma^+$ tels que $x = \sigma(\overline{x}), x' = \sigma(\overline{x}'), z = \sigma(\overline{z})$ et $z' = \sigma(\overline{z}')$. On a

$$zxzx'z' = \sigma(\overline{z})\sigma(\overline{x})\sigma(\overline{z})\sigma(\overline{x}')\sigma(\overline{z}') = \sigma(\overline{zxzx}'\overline{z}').$$

On dit que l'équation $zxzx'z' = zx'z'xz$ est "équilibrée" dans le sens que le mot $\overline{zxzx}'\overline{z}'$ est une permutation du mot $\overline{zx}'\overline{z}'\overline{xz}$ avec

$$zx'z'xz = zxzx'z' \Leftrightarrow \sigma(\overline{zx}'\overline{z}'\overline{xz}) = \sigma(\overline{zxzx}'\overline{z}').$$

3.2 Simplification

Définition 3.2.1. Soient S un semi-groupe et $s \in S$. On dit que s est *simplifiable à gauche* de S si pour tous $u, v \in S$, $su = sv$ implique $u = v$. De la même façon, on dira que s est *simplifiable à droite* dans S si pour tous $u, v \in S$, $us = vs$ implique $u = v$. Ainsi, on dit que s est *simplifiable* dans S si s est simplifiable à gauche et à droite dans S. Si tout élément de S est simplifiable alors S est un *semi-groupe simplifiable*.

Exemple 3.2.2. Soit X un ensemble (fini ou infini) et notons S l'ensemble de toutes les fonctions $f\colon X \to X$. Clairement, S est un semi-groupe pour la composition de fonctions (si $f, g \in S$, alors $f \circ g \in S$ et $(f \circ g) \circ h = f \circ (g \circ h)$). Les éléments de S simplifiables à gauche sont les fonctions injectives. En effet, si $f\colon X \to X$ est une fonction injective et que g et g' sont des éléments de S, on a

$$\begin{aligned} fg(x) = fg'(x) &\implies f(g(x)) = f(g'(x)) \\ &\implies g(x) = g'(x) \end{aligned}$$

pour tout élément $x \in X$, où la dernière implication est obtenue grâce à l'injectivité de f. Réciproquement, montrons par contraposée que toute fonction simplifiable à gauche est injective. Soit f une fonction qui n'est pas injective, alors il existe $x_1, x_2 \in X$ tels que $f(x_1) = f(x_2)$ et $x_1 \neq x_2$. Considérons les fonctions $g\colon X \to X\colon x \mapsto x_1$ et $g'\colon X \to X\colon x \mapsto x_2$. Alors on a $f(g(x)) = f(g'(x))$ pour tout $x \in X$ mais $g \neq g'$ car $x_1 \neq x_2$. Donc f n'est pas simplifiable à gauche.

De même, les éléments simplifiables à droite sont les fonctions surjectives. En effet, considérons $f, f' \in S$ et $g\colon X \to X$ une fonction surjective telle que $fg = f'g$. Soit $y \in X$, il existe $x \in X$ tel que $y = g(x)$. Alors on a

$$f(y) = f(g(x)) = f'(g(x)) = f'(y).$$

Donc $f = f'$ et g est bien simplifiable à droite. Réciproquement, montrons par contraposée que toute fonction simplifiable à droite est surjective. En effet, si g n'est pas surjectif, alors il existe $x_0 \in X$ tel que x_0 n'appartient pas à l'image de g. Considérons x_1 dans l'image de g (un tel élément existe si X est non vide) et les fonctions $f\colon X \to X$ définie par $f(x) = x_0$ et $f'\colon X \to X$ définie par $f'(x) = x_1$ si $x \in \mathrm{Im}(g)$ et $f'(x) = x_0$ sinon. Alors on a $f(g(x)) = f'(g(x))$ pour tout $x \in X$ mais $f \neq f'$ car $f(x_0) = x_0$ et $f'(x_0) = x_1$. Donc g n'est pas simplifiable à droite.

Ainsi, les éléments simplifiables de S sont les fonctions bijectives.

On peut réénoncer la propriété caractérisant un élément simplifiable en utilisant les fonctions multiplication à gauche $L_a\colon S \to S$ et à droite $R_a\colon S \to S$ définies par les relations suivantes : $L_a(b) = ab$ et $R_a(b) = ba$ pour tout $b \in S$. Un élément $s \in S$ sera dit simplifiable à gauche si et seulement si la fonction L_s est injective et simplifiable à droite si et seulement si la fonction R_s est injective.

Intéressons-nous à présent à un premier corollaire du Lemme 3.1.1.

Lemme 3.2.3. *Soient S un semi-groupe et X un sous-ensemble de S de cardinalité plus grande que 1 et tel que tout élément de X soit simplifiable à gauche dans S. L'ensemble X n'est pas un code si et seulement si il existe $x, x' \in X$ et $z, z' \in X^+$ tels que $x \neq x'$ et $xz = x'z'$.*

Démonstration. Supposons que X n'est pas un code, par le Lemme 3.1.1 on sait qu'il existe alors $x, x' \in X$ et $t, t' \in X^+$ tels que $txtx't' = tx't'xt$. Par hypothèse, t est simplifiable à gauche dans S donc on a $xtx't' = x't'xt$. Ainsi, en posant $z = tx't'$ et $z' = t'xt$ on a bien l'égalité $xz = x'z'$ attendue. La réciproque est immédiate grâce à la Définition 1.1.1. □

Le Lemme 3.2.3 sera beaucoup utilisé par la suite. L'exemple suivant nous montre l'importance de la propriété de simplification dans l'énoncé du Lemme 3.2.3 :

Exemples 3.2.4.

⋆ Posons

$$X = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} \text{ et } X' = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}.$$

L'ensemble $\{X, X'\}$ possède un élément qui n'est pas simplifiable à gauche. Par exemple l'élément X ne l'est pas car $XX'X = \begin{pmatrix} 64 & 32 \\ 32 & 6 \end{pmatrix} \neq \begin{pmatrix} 40 & 20 \\ 80 & 40 \end{pmatrix} = X'XX$ mais on a $X(XX'X) = X(X'XX)$. De plus, l'ensemble $\{X, X'\}$ n'est pas un code pour la multiplication matricielle car $XXX'X = XX'XX$ et $X \neq X'$. En outre, la matrice ligne $L = (-1\ 2)$ est telle que $LX = (-1\ 2) \cdot \left(\begin{smallmatrix} 4 & 2 \\ 2 & 1 \end{smallmatrix}\right) = (0\ 0)$ et $LX' = (-1\ 2) \cdot \left(\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right) = (3\ 6)$. Donc pour tous $Z, Z' \in \{X, X'\}^+$, on a $LXZ = (0\ 0)$ et $LX'Z'$ est une matrice ligne positive. Par conséquent, XZ et $X'Z'$ sont distincts pour tous $Z, Z' \in \{X, X'\}^+$.

⋆ L'ensemble des entiers positifs est un semi-groupe simplifiable pour l'addition (pour tous $a, b \in S$, $s + a = s + b \Rightarrow a = b$).

3.3 Produit direct de semi-groupes

Étant donnés deux semi-groupes S et T tels que T est commutatif, caractérisons les sous-ensembles de $S \times T$ qui sont des codes.

Lemme 3.3.1. *Soient S et T deux semi-groupes où T est commutatif et soit Z un sous-ensemble de $S \times T$ de cardinalité plus grande que 1. On définit l'application $\alpha\colon S \times T \to S$ par $\alpha(s,t) = s$ pour tout $(s,t) \in S \times T$. L'ensemble Z est un code si et seulement si α est injectif sur Z et $\alpha(Z)$ est un code.*

Démonstration. Si on suppose que α n'est pas injectif sur Z, alors il existe $x \in S$ et $y, y' \in T$ tels que $y \neq y'$ et $(x,y), (x,y') \in Z$. On a $(x,y)\cdot(x,y') = (xx, yy') = (xx, y'y) = (x,y')\cdot(x,y)$ car T est commutatif. Donc Z n'est pas un code.

Si on suppose maintenant que $\alpha(Z)$ n'est pas un code, alors par le Lemme 3.1.1 il existe $(x,y), (x',y') \in Z$ et $(u,v), (u',v') \in Z^+$ tels que $x \neq x'$ et $uxux'u' = ux'u'xu$. De plus, on a $vyvy'v' = vy'v'yv$ car y, y', v et v' appartiennent à T, qui est commutatif. Donc, on a

$$\begin{aligned}(u,v)(x,y)(u,v)(x',y')(u',v') &= (uxux'u', vyvy'v') \\ &= (ux'u'xu, vy'v'yv) \\ &= (u,v)(x',y')(u',v')(x,y)(u,v).\end{aligned}$$

Donc Z n'est pas un code car on a trouvé deux factorisations différentes d'un élément de $S \times T$ en des éléments de Z.

La réciproque est immédiate grâce à la Proposition 1.3.3 car l'application α est un morphisme. En effet, pour tous $(s,t),(s',t') \in S \times T$, on a $\alpha(ss',tt') = ss' = \alpha(s,t)\alpha(s',t')$. □

Lemme 3.3.2. *Soient S et T deux semi-groupes et $y \in T$. Pour tout sous-ensemble $X \subseteq S$ de cardinalité plus grande que 1, $X \times \{y\}$ est un code si et seulement si X est un code.*

Démonstration. Comme T n'est pas nécessairement commutatif, on pose $T' = \{y, y^2, \dots\}$, il s'agit d'un sous-semi-groupe commutatif de T tel que $X \times \{y\} \subseteq S \times T'$. Vu le lemme précédent, Z est un code si et seulement si $\{((s,t),\alpha(s,t)) : (s,t) \in Z\}$ est un code. Donc

$$\begin{aligned} X \times \{y\} \text{ est un code } &\Leftrightarrow \{((s,y),\alpha(s,y)) : (s,y) \in X \times \{y\}\} \text{ est un code} \\ &\Leftrightarrow \{((s,\alpha(s)) : s \in X\} \text{ est un code} \\ &\Leftrightarrow X \text{ est un code.} \end{aligned}$$

□

Théorème 3.3.3. *Soient S et T deux semi-groupes récursifs non-vides et soit un entier $k > 1$. Si le problème* Free$(k)[S \times T]$ *est décidable, alors les problèmes* Free$(k)[S]$ *et* Free$(k)[T]$ *le sont aussi.*

Démonstration. Soit y un élément fixé de T. Pour tout sous-ensemble $X \subseteq S$ à k éléments, $X \times \{y\}$ est un sous-ensemble de $S \times T$ à k éléments. De plus, vu le Lemme 3.3.2, X est un code si et seulement si $X \times \{y\}$ est un code. Donc, il existe une réduction du problème Free$(k)[S]$ au problème Free$(k)[S \times T]$. De la même façon, Free$(k)[T]$ se réduit à Free$(k)[S \times T]$. □

La réciproque du Théorème 3.3.3 est fausse en toute généralité : par exemple, le problème Free$[\mathbb{W}]$ est décidable alors que le problème Free$(k)[\mathbb{W} \times \mathbb{W}]$ est indécidable pour tout entier $k \geq 13$ (nous en reparlerons dans le Chapitre 6).

Par contre, on a le lemme suivant :

Lemme 3.3.4. *Soient S et T deux semi-groupes récursifs et un entier $k > 1$. Si le problème* Free$(k)[S]$ *est décidable et que T est commutatif, alors le problème* Free$(k)[S \times T]$ *est décidable.*

Démonstration. Nous allons démontrer que le problème Free$(k)[S \times T]$ se réduit au problème Free$(k)[S]$. Considérons une instance $Z \subseteq S \times T$ à k éléments du problème Free$(k)[S \times T]$ et considérons la transformation $\alpha\colon S \times T \to S\colon (s,t) \mapsto s$. On a les équivalences suivantes : Z est une instance positive de Free$(k)[S \times T]$ si et seulement si Z est un code. Grâce au Lemme 3.3.1, on sait que Z est un code si et seulement si α est injectif sur Z et $\alpha(Z)$ est un code. Comme α est injectif sur Z si et seulement si $|\alpha(Z)| = k$, alors la condition est équivalente à avoir $\alpha(Z)$ qui est un code à k éléments. Enfin, on a cela si et seulement si $\alpha(Z)$ est une instance positive de Free$(k)[S]$ car $\alpha(Z) \subseteq S$. □

Définition 3.3.5. Pour chaque nombre $d \in \mathbb{N}$, notons $\mathbb{W}^{\times d}$ le semi-groupe obtenu à partir du produit direct de d copies de $\mathbb{W}$. En particulier, on a $\mathbb{W}^{\times 0} = \{\epsilon\}$, $\mathbb{W}^{\times 1} = \mathbb{W}$, $\mathbb{W}^{\times 2} = \mathbb{W} \times \mathbb{W}$, $\mathbb{W}^{\times 3} = \mathbb{W} \times \mathbb{W} \times \mathbb{W}$, etc.

Théorème 3.3.6. *Soient n un nombre entier strictement positif et Σ_1, Σ_2,..., Σ_n n alphabets finis. Notons d le nombre de $i \in \{1, \cdots, n\}$ tels que la cardinalité de Σ_i est plus grande que 1. Pour tout entier $k \geq 1$, le problème* $\text{FREE}(k)[\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*]$ *est décidable si et seulement si le problème* $\text{FREE}(k)[\mathbb{W}^{\times d}]$ *est décidable.*

Démonstration. Considérons uniquement le cas où $k > 1$ car si $k = 1$ les deux problèmes sont trivialement décidables. De plus, pour toute permutation $(i_1, i_2, \ldots, i_n)$ de $\{1, \ldots, n\}$, les ensembles $\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*$ et $\Sigma_{i_1}^* \times \Sigma_{i_2}^* \times \cdots \times \Sigma_{i_n}^*$ sont isomorphes car la fonction qui envoie $(w_1, w_2, \ldots, w_n) \in \Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*$ sur $(w_{i_1}, w_{i_2}, \ldots, w_{i_n})$ est un isomorphisme. Ainsi, on peut supposer, sans perte de généralité, que la cardinalité de Σ_i est plus grande que 1 pour tout $i \in \{1, \ldots, d\}$. De manière équivalente, on peut dire que Σ_i^* est un ensemble commutatif pour tout $i \in \{d+1, \ldots, n\}$. Par le Théorème 3.3.3 et le Lemme 3.3.4, on sait que $\text{FREE}(k)[\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*]$ est décidable si et seulement si le problème $\text{FREE}(k)[S]$ est décidable, avec $S = \Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_d^*$.

Pour tout $i \in \{1, \ldots, d\}$, posons $\phi_i : \mathbb{W} \to \Sigma_i^*$ un morphisme injectif. La fonction qui fait correspondre chaque $(u_1, u_2, \ldots, u_d) \in \mathbb{W}^{\times d}$ à $(\phi_1(u_1), \phi_2(u_2), \ldots, \phi_d(u_d))$ est un morphisme injectif de S dans $\mathbb{W}^{\times d}$. Ainsi, le problème $\text{FREE}(k)[\mathbb{W}^{\times d}]$ se réduit au problème $\text{FREE}(k)[\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*]$.

Pour la réciproque, posons $\psi_i : \Sigma_i^* \to \mathbb{W}$ un morphisme injectif pour tout $i \in \{1, \ldots, d\}$. La fonction qui envoie chaque $(v_1, v_2, \ldots, v_d) \in S$ sur $(\psi_1(v_1), \psi_2(v_2), \ldots, \psi_d(v_d))$ est un morphisme injectif de S dans $\mathbb{W}^{\times d}$. Donc le problème $\text{FREE}(k)[\Sigma_1^* \times \Sigma_2^* \times \cdots \times \Sigma_n^*]$ se réduit au problème $\text{FREE}(k)[\mathbb{W}^{\times d}]$. □

3.4 Matrices rationnelles et matrices entières

Lemme 3.4.1. *Soient d un nombre entier strictement positif, $\mathcal{X}$ un sous-ensemble de $\mathbb{C}^{d\times d}$ de cardinalité plus grande que 1 et une fonction $\lambda : \mathcal{X} \to \mathbb{C} \setminus \{0\}$. L'ensemble $\mathcal{X}$ est un code pour la multiplication matricielle si et seulement si les deux conditions suivantes sont satisfaites :*

(i) $\{\lambda(X)X : X \in \mathcal{X}\}$ *est un code pour la multiplication matricielle et*

(ii) pour tous $X, Y \in \mathcal{X}$, $X \neq Y$ implique $\lambda(X)X \neq \lambda(Y)Y$.

Démonstration. La preuve de ce lemme se décompose en trois parties.

1) Posons $\mathcal{Z} = \{(X, \lambda(X)) : X \in \mathcal{X}\}$. Par le Lemme 3.3.1, on sait que $\mathcal{Z}$ est un code si et seulement si $\mathcal{X}$ est un code. En effet, on a $\mathcal{Z}$ qui est un code si et seulement si l'application $\alpha\colon \mathbb{C}^{d\times d} \times \mathbb{C}_0 \to \mathbb{C}^{d\times d} : (X, \lambda(X)) \mapsto X$ est injective sur $\mathcal{Z}$ (ce qui est le cas) et que l'ensemble $\alpha(\mathcal{Z}) = \{X : X \in \mathcal{X}\}$ est un code, c'est-à-dire si et seulement si $\mathcal{X}$ est un code.

2) Posons maintenant $\mathcal{Z}' = \{(\lambda(X)X, \lambda(X)) : X \in \mathcal{X}\}$. Montrons que $\mathcal{Z}'$ est un code si et seulement si les conditions (i) et (ii) de l'énoncé sont satisfaites. Grâce au Lemme 3.3.1, on sait que $\mathcal{Z}'$ est un code si et seulement si l'application

$$(\lambda(X)X, \lambda(X)) \mapsto \lambda(X)X$$

est injective sur $\mathcal{Z}'$ et l'ensemble $\{\lambda(X)X : X \in \mathcal{X}\}$ est un code. Cette dernière condition correspond exactement à l'assertion (i) de l'énoncé. Ainsi, il suffit en fait de démontrer que l'application

$$a_1 : X \mapsto \lambda(X)X$$

est injective sur $\mathcal{X}$ (ce qui correspond à l'assertion (ii) de l'énoncé) si et seulement si l'application

$$a_2 : (\lambda(X)X, \lambda(X)) \mapsto \lambda(X)X$$

est injective sur $\mathcal{Z}'$.

Soient $X, Y \in \mathcal{X}$ tels que $(\lambda(X)X, \lambda(X)) \neq (\lambda(Y)Y, \lambda(Y))$, montrons que $\lambda(X)X \neq \lambda(Y)Y$. Si on suppose par l'absurde que $\lambda(X)X = \lambda(Y)Y$, alors d'une part $X = Y$ car l'application a_1 est injective sur $\mathcal{X}$, et d'autre part, $\lambda(X) \neq \lambda(Y)$ car $(\lambda(X)X, \lambda(X)) \neq (\lambda(Y)Y, \lambda(Y))$. Donc au final, l'ensemble $\{\lambda(X)X : X \in \mathcal{X}\}$ n'est pas un code, on obtient une contradiction.

Réciproquement, si on suppose par l'absurde que $X \neq Y$ et que $\lambda(X)X = \lambda(Y)Y$, alors $\lambda(X) = \lambda(Y)$ car l'application a_2 est injective. De nouveau, l'ensemble $\{\lambda(X)X : X \in \mathcal{X}\}$ n'est pas un code, on obtient une contradiction.

3) Il reste à montrer que $\mathcal{Z}$ est un code si et seulement si $\mathcal{Z}'$ est un code. On définit la fonction $\sigma : \mathbb{C}^{d\times d} \times \mathbb{C} \setminus \{0\} \to \mathbb{C}^{d\times d} \times \mathbb{C} \setminus \{0\}$ par $\sigma(X, a) = (aX, a)$ pour tout $(X, a) \in \mathbb{C}^{d\times d} \times \mathbb{C} \setminus \{0\}$. Remarquons que $\sigma(\mathcal{Z}) = \{\sigma(X, \lambda(X)) : X \in \mathcal{X}\} = \{(\lambda(X)X, \lambda(X)) : X \in \mathcal{X}\} = \mathcal{Z}'$, que σ est injectif car si $(aX, a) = (bY, b)$ alors $a = b$ et $X = Y$, et que σ est un morphisme car $\sigma(XY, ab) = (abXY, ab) = ((aX)(bY), ab) = \sigma(X, a)\sigma(Y, b)$ pour tous $X, Y \in \mathbb{C}^{d\times d}$ et tous $a, b \in \mathbb{C} \setminus \{0\}$. Donc σ est un morphisme injectif sur $\mathbb{C}^{d\times d} \times \mathbb{C} \setminus \{0\}$ tel que $\sigma(\mathcal{Z}) = \mathcal{Z}'$. Donc par la Proposition 1.3.3, on en déduit que $\mathcal{Z}$ est un code si et seulement $\mathcal{Z}'$ en est un.

Ainsi, $\mathcal{X}$ est un code si et seulement si les conditions (i) et (ii) de l'énoncé sont satisfaites. □

Soient $X = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right)$, $Y = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\mathcal{X} = \{X, Y\}$, $\lambda(X) = 1$ et $\lambda(Y) = 2$. Remarquons que l'on a l'égalité suivante : $\lambda(X)X = \lambda(Y)Y$. Clairement, $\{\lambda(X)X, \lambda(Y)Y\} = \{X\}$ est un code pour la multiplication matricielle mais $\mathcal{X}$ ne l'est pas. Cela nous montre que le point (ii) du Lemme 3.4.1 est important pour pouvoir démontrer que $\mathcal{X}$ est un code.

Maintenant on peut démontrer le résultat important de cette section :

Théorème 3.4.2. *Pour tous entiers $k, d \geq 1$, le problème* $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$ *est décidable si et seulement si* $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$ *l'est.*

Démonstration. Si $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$ est décidable alors $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$ l'est aussi car $\mathbb{Z}^{d\times d} \subset \mathbb{Q}^{d\times d}$.

De plus, on a démontré, grâce au Théorème 2.2.6, que le problème $\textsc{Free}(1)[\mathbb{Q}^{d\times d}]$ est décidable. Il reste donc à montrer qu'il existe une réduction du problème $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$ au problème $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$ lorsque $k > 1$.

Pour tout sous-ensemble fini $\mathcal{X} \subseteq \mathbb{Q}^{d\times d}$, notons $t(\mathcal{X})$ le plus petit entier $n \geq 1$ tel que $nX \in \mathbb{Z}^{d\times d}$ pour tout $X \in \mathcal{X}$. Pour toute instance $\mathcal{X}$ du problème $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$, $\mathcal{X}' = \{t(\mathcal{X})X : X \in \mathcal{X}\}$ est une instance du problème $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$. De plus, $\mathcal{X}'$ est calculable à partir de $\mathcal{X}$ et par le Lemme 3.4.1, $\mathcal{X}$ est une instance positive de $\textsc{Free}(k)[\mathbb{Q}^{d\times d}]$ si et seulement si $\mathcal{X}'$ est une instance positive de $\textsc{Free}(k)[\mathbb{Z}^{d\times d}]$. □

Pour conclure cette section, regardons si le Théorème 3.4.2 peut s'appliquer de manière analogue à d'autres problèmes de décision tels que ceux introduits au Chapitre 1.

Premièrement, pour tous entiers k et $d \geq 1$, le problème $\textsc{Mortal}(k)[\mathbb{Q}^{d\times d}]$ est décidable si et seulement si $\textsc{Mortal}(k)[\mathbb{Z}^{d\times d}]$ est décidable car on peut utiliser la même réduction que celle utilisée dans la preuve du théorème précédent. En effet, pour tout sous-ensemble $\mathcal{X} \subseteq \mathbb{Q}^{d\times d}$ à k éléments, $\mathcal{X}$ est une instance positive de $\textsc{Mortal}(k)[\mathbb{Q}^{d\times d}]$ si et seulement si $\mathcal{X}'$ est une instance positive de $\textsc{Mortal}(k)[\mathbb{Z}^{d\times d}]$, où $\mathcal{X}'$ est défini comme dans la preuve du Théorème 3.4.2. Notons en passant que la décidabilité du problème $\textsc{Mortal}(k)[\mathbb{Q}^{d\times d}]$ n'a pas encore été démontrée pour des paires (d, k) de nombres entiers (voir l'article [2]).

Deuxièmement, pour tout nombre entier $d \geq 1$, le problème $\textsc{Bounded}[\mathbb{Z}^{d\times d}]$ est décidable car dans le cas des nombres entiers, ce problème est le même que $\textsc{Finite}[\mathbb{Z}^{d\times d}]$, qui est décidable (voir l'article [13]).

Troisièmement, le problème de savoir s'il existe des nombres strictement positifs k_0 et d_0 satisfaisant les deux propriétés suivantes reste ouvert : $\textsc{Member}(k_0)[\mathbb{Q}^{d_0\times d_0}]$ est indécidable et $\textsc{Member}(k_0)[\mathbb{Z}^{d_0\times d_0}]$ est décidable.

3.5 Le cas des groupes

Concentrons-nous maintenant, non plus sur des semi-groupes mais sur des groupes, et tentons de caractériser la décidabilité du problème $\textsc{Free}[G]$ où G est un groupe.

3.5.1 Rappels de théorie des automates

Définition 3.5.1. Soit X un ensemble, un *automate* sur X est un quadruplet $A = (Q, E, I, F)$ où Q est un ensemble d'*états*, I et F sont des sous-ensembles de Q et E est un sous-ensemble de $Q \times X \times Q$ dont les éléments sont appelés les *transitions* de A.

Les éléments de I sont les états *initiaux* de A et les éléments de F sont les états *finals* de A. On dit que A est *fini* si les ensembles Q et E sont finis. Une transition $(p, s, q) \in E$ est notée $p \xrightarrow{s} q$.

Définition 3.5.2. Soient M un monoïde, A un automate sur M et s un élément de M. On dit que A *accepte* s si pour un entier $n \in \mathbb{N}$ il existe $n+1$ états $q_0, q_1, ..., q_n$ et n éléments $s_1, s_2, ..., s_n \in M$ tels que $s = s_1 s_2 ... s_n$, q_0 est un état initial de A, q_n est un état final de A et $q_{i-1} \xrightarrow{s} q_i$ est une transition de A pour tout $i \in \{1, ..., n\}$. On définit le *comportement* de A comme étant l'ensemble des éléments de M qui sont acceptés par A.

Remarque 3.5.3. Pour tout automate $A = (Q, E, I, F)$ sur un monoïde M tel que $I \bigcap F \neq \emptyset$, A accepte le neutre de M. En effet, il existe au moins un état q qui est initial et final, tel que $q \xrightarrow{e} q$ où e est le neutre de M. Donc l'automate A accepte e.

3.5.2 Le problème d'acceptation

Définition 3.5.4. Pour tout monoïde récursif M, on définit le problème ACCEPT$[M]$ comme suit : étant donnés un automate fini A sur M et un élément $s \in M$, déterminer si A accepte s.

Exemple 3.5.5. Pour tout alphabet fini Σ, ACCEPT$[\Sigma^*]$ est décidable en un temps polynomial. En effet, soient A un automate fini sur Σ^* et $\omega \in \Sigma^*$, si on suppose que A est déterministe, alors on va pouvoir construire un algorithme qui décide du problème ACCEPT$[\Sigma^*]$: on simule l'automate A sur l'entrée ω en commençant par l'état initial, si il s'arrête dans un état final alors la réponse est "oui", sinon la réponse est "non". Cet algorithme est assez rapide, si le mot ω est de longueur n et que A est représenté par sa table de transition, alors chaque transition nécessite le même temps et donc l'algorithme est en $O(n)$.

Définition 3.5.6. Un *groupe* est un monoïde G dans lequel tout élément admet un inverse. Le neutre de G est noté 1_G. La fonction inverse dans G est la fonction de G dans G qui applique à chaque élément $g \in G$ son inverse g^{-1}.

Remarquons que si on considère un groupe G, alors un élément $x \in G$ est de torsion si et seulement si il existe un entier $n \geq 1$ tel que $x^n = 1_G$. Il suffit d'utiliser la Définition 2.1.1.

Lemme 3.5.7.

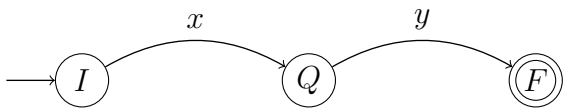
(*i*) *Soit M un monoïde récursif, si le problème* ACCEPT$[M]$ *est décidable alors l'opération de M est calculable.*

(*ii*) *Soit G un groupe récursif, si l'opération de G est calculable, alors l'inversion dans G l'est aussi.*

Démonstration. Soient $x, y \in M$, définissons l'automate $A_{x,y}$ sur $\{x, y\}$ par :

- I, Q et F sont les états de $A_{x,y}$

- $I \xrightarrow{x} Q$ et $Q \xrightarrow{y} F$ sont les transitions de $A_{x,y}$
- I est l'unique état initial
- F est l'unique état final



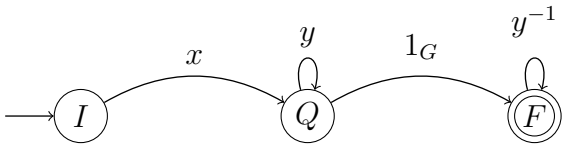
L'ensemble des éléments de M acceptés par $A_{x,y}$ est égal à $\{xy\}$. Supposons dans un premier temps que le problème ACCEPT$[M]$ est décidable. Pour calculer xy à partir de x et y, il suffit de d'abord calculer $A_{x,y}$ et puis d'examiner les éléments de M un à un jusqu'à en trouver un qui soit accepté par $A_{x,y}$. Ainsi on a démontré le point (i).

Soient $g, h \in G$, pour déterminer si h est l'inverse de g, il suffit de calculer gh, ce qui est possible par hypothèse, et de déterminer si le résultat est égal à 1_G. Par conséquent, l'inverse de tout élément de G est calculable. □

Théorème 3.5.8. *Soit G un groupe récursif, si le problème* ACCEPT$[G]$ *est décidable, alors* FREE$[G]$ *l'est aussi.*

Démonstration. Considérons un sous-ensemble fini $X \subseteq G$ contenant au moins un élément. Pour tout $x \in G$, notons A_x l'automate sur G défini par :

- I, Q et F sont les états de A_x,
- les transitions de A_x sont $I \xrightarrow{x} Q$, $Q \xrightarrow{1_G} F$ et pour chaque $y \in X$, $Q \xrightarrow{y} Q$ et $F \xrightarrow{y^{-1}} F$,
- I est l'unique état initial,
- F est l'unique état final.



L'ensemble des éléments de G acceptés par A_x est égal à $\{xz_1 \cdots z_n 1_G (z'_1)^{-1} \cdots (z'_l)^{-1} : z_1, \ldots, z_n, z'_1, \ldots, z'_l \in X\} = \{xzz'^{-1} : (z, z') \in X^* \times X^*\}$. Par le Lemme 3.2.3, on sait que X n'est pas un code si et seulement si il existe $x, x' \in X$ et $z, z' \in X^+$ tels que $x \neq x'$ et $xz = x'z'$, c'est-à-dire, dans notre cas, s'il existe $x, x' \in X$ tels que $x \neq x'$ et A_x accepte x', car A_x accepte x' si et seulement si il existe $z, z' \in X^+$ tels que $x' = xzz'^{-1}$.

Si le problème ACCEPT$[G]$ est décidable, alors par le Lemme 3.5.7, l'opération de G est calculable et donc l'inversion dans G est aussi calculable. Ainsi, l'automate A_x est calculable à partir de x et X. Donc on peut décider si X est un code ou non.

Ainsi, le problème FREE$[G]$ est décidable. □

Lorsqu'on considère des instances (A, s) du problème ACCEPT$[G]$ où l'automate A a au plus 3 états, le Théorème 3.5.8 est toujours valable.

Chapitre 4

Le cas des matrices carrées de dimension 2

Dans ce chapitre, nous nous concentrerons sur le semi-groupe des matrices 2×2, c'est à partir de là que vont se former plusieurs questions ouvertes intéressantes. Tout d'abord nous allons caractériser la décidabilité du problème $\textsc{Free}[K^{2\times 2}]$ pour un champ K en se ramenant au problème $\textsc{Accept}[K^{2\times 2}]$ déjà introduit. De plus, nous regarderons quelques cas particuliers du problème $\textsc{Free}(2)[\mathbb{N}^{2\times 2}]$, pour lesquels on sait discuter de la décidabilité. On se restreindra en particulier aux matrices triangulaires supérieures et inférieures.

Ensuite nous étudierons la décidabilité du problème $\textsc{Free}(2)[\hom(\mathbb{W})]$, bien qu'elle soit ouverte, nous y apporterons quelques détails ainsi que des exemples d'instances de ce problème.

Enfin, nous aborderons l'étude du problème $\textsc{Mortal}[\mathbb{Q}^{n\times n}]$ pour $n = 2$ que nous poursuivrons dans le Chapitre 6 relatif aux matrices 3×3. Dans ce chapitre-ci, nous démontrerons qu'un cas particulier de ce problème est décidable en construisant un algorithme.

4.1 Concernant l'indécidabilité

Bien que la décidabilité des problèmes $\textsc{Free}[\mathbb{N}^{2\times 2}]$, $\textsc{Free}[\mathbb{Q}^{2\times 2}]$ et $\textsc{Free}[\mathbb{Z}^{2\times 2}]$ est toujours ouverte, Bell et Potapov [1] ont prouvé que $\textsc{Free}(7)[\mathcal{H}^{2\times 2}]$ est indécidable, où

$$\mathcal{H} = \left\{ \begin{pmatrix} x & y & z & t \\ -y & x & -t & z \\ -z & t & x & -y \\ -t & -z & y & x \end{pmatrix} : x, y, z, t \in \mathbb{Q} \right\}$$

est l'ensemble des quaternions. De plus, on en déduit grâce au Théorème 2.2.6 que le problème $\textsc{Free}(1)[\mathcal{H}^{2\times 2}]$ est décidable : pour tout $M \in \mathcal{H}^{2\times 2}$, M est de torsion si et seulement si $M^8 = M^{8+r(8)}$. Ainsi, une question naturelle se pose :

Question ouverte 1 : Existe-t-il un semi-anneau commutatif récursif D satisfaisant les deux propriétés suivantes : le problème $\textsc{Free}(1)[D^{2\times 2}]$ est décidable et le problème $\textsc{Free}[D^{2\times 2}]$ est indécidable ?

Soit D un semi-anneau récursif tel que FREE(1) $[D^{2\times 2}]$ est décidable. Alors, l'ensemble des éléments de D qui sont de torsion pour la multiplication matricielle est obtenu récursivement : pour chaque $t \in D$, t est de torsion pour la multiplication matricielle si et seulement si $\begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix}$ est de torsion pour la multiplication matricielle. En effet, il existe $p, q \in \mathbb{N}$ tels que $t^p = t^q$ si et seulement si

$$\begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix}^p = \begin{pmatrix} t^p & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} t^q & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix}^q.$$

De plus, l'ensemble des éléments de D qui sont de torsion pour l'addition matricielle est aussi obtenu récursivement : pour chaque $t \in D$, t est de torsion pour l'addition matricielle si et seulement si $\begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$ est de torsion pour la multiplication matricielle. En effet, il existe $p, q \in \mathbb{N}$ tels que $pt = qt$ si et seulement si

$$\begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}^p = \begin{pmatrix} 1 & pt \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & qt \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}^q.$$

Par conséquent, la décidabilité de FREE(1)$[D^{2\times 2}]$ contrôle D.

Soit K un champ d'extension de $\mathbb{Q}$ de degré d. Comme il existe un homomorphisme d'anneaux injectif entre K et $\mathbb{Q}^{d\times d}$ ([10]), alors grâce au Théorème 2.2.6, on sait que pour tout $M \in K^{2\times 2}$, M est de torsion si et seulement si $M^{2d} = M^{2d+r(2d)}$. Ainsi, le problème FREE(1)$[K^{2\times 2}]$ est décidable. Le fait de démontrer l'indécidabilité de FREE$[K^{2\times 2}]$ pour des champs d'extension K de $\mathbb{Q}$ répondrait à la Question 1 et permettrait une belle avancée concernant l'indécidabilité du problème FREE$[\mathbb{Q}^{2\times 2}]$.

Passons maintenant à quelque chose de plus général que la Question 1.

Lemme 4.1.1. *Soient A un anneau commutatif et une matrice $X \in A^{2\times 2}$ de déterminant nul. Alors,*

(i) pour toute matrice $Y \in A^{2\times 2}$, on a l'égalité $XXYX = XYXX$;

(ii) la matrice X est de torsion si et seulement si sa trace est de torsion.

Démonstration. Notons t la trace de la matrice X.

Le polynôme caractéristique de X est donné par $\det(X - zI_2) = z^2 - tr(X)z - \det(X) = z^2 - tz$. Donc par le théorème de Cayley-Hamilton on a l'égalité $X^2 = tX$. Ainsi on obtient que $XXYX = tXYX = XYtX = XYXX$ pour tout $Y \in A^{2\times 2}$ où la deuxième égalité est obtenue grâce au fait que l'anneau A est commutatif. Le point (i) est alors démontré.

Pour le point (ii), on sait d'une part que $X^{n+1} = t^n X$ pour tout $n \in \mathbb{N}$. Donc si t est de torsion alors X l'est aussi. En effet, s'il existe $p, q \in \mathbb{N}$ tels que $t^p = t^q$, alors on a $X^{p+1} = t^p X = t^q X = X^{q+1}$. D'autre part, la trace de X^n est donnée par $tr(t^{n-1}X) = t^{n-1}tr(X) = t^n$ pour tout entier $n \geq 1$. Donc si X est de torsion alors t l'est aussi, car si il existe $p, q \in \mathbb{N}$ tels que $X^p = X^q$ alors $t^p = tr(X^p) = tr(X^q) = t^q$. Ainsi le point (ii) est démontré. □

Soit K un champ, le *groupe linéaire général* de degré d sur K est noté $\mathrm{GL}(d, K)$ et défini comme suit :

$$\mathrm{GL}(d, K) = \left\{ X \in K^{d\times d} : \det(X) \neq 0 \right\}.$$

Supposons K récursif et l'addition et la multiplication dans K calculables, alors par le Lemme 3.5.7 (ii), l'inversion pour l'addition et pour la multiplication dans K et $K \setminus \{0\}$ respectivement sont aussi calculables. En particulier, le déterminant des matrices sur K est calculable, donc $\mathrm{GL}(d, K)$ est un ensemble récursif.

Proposition 4.1.2. *Soit K un champ avec des opérations calculables. Le problème* $\textsc{Free}[K^{2\times 2}]$ *est décidable si et seulement si le problème* $\textsc{Free}[\textsc{Gl}(2, K)]$ *l'est.*

Démonstration. Il est clair que si le problème $\textsc{Free}[K^{2\times 2}]$ est décidable alors $\textsc{Free}[\textsc{Gl}(2, K)]$ l'est aussi car $\mathrm{GL}(2, K)$ est un sous-semi-groupe de $K^{2\times 2}$. Démontrons la réciproque. Supposons que le problème $\textsc{Free}[\textsc{Gl}(2, K)]$ est décidable, soit un sous-ensemble fini $\mathcal{X} \subseteq K^{2\times 2}$, démontrons que l'on peut décider si c'est un code ou non.

- Si $\mathcal{X} \subseteq \mathrm{GL}(2, K)$, on peut directement conclure car le problème $\textsc{Free}[\textsc{Gl}(2, K)]$ est décidable.
- Si $\mathcal{X} \nsubseteq \mathrm{GL}(2, K)$ et si $\mathcal{X}$ contient plus de un élément, alors il existe $X \in \mathcal{X}$ tel que $\det(X) = 0$. Dans ce cas on peut appliquer le Lemme 4.1.1(i) et obtenir l'équation $XXYX = XYXX$ pour toute matrice $Y \in \mathcal{X}$. Donc $\mathcal{X}$ n'est pas un code.
- Si $\mathcal{X} = \{X\}$ avec $X \notin \mathrm{GL}(2, K)$, c'est-à-dire $\det(X) = 0$, alors on va s'aider du Lemme 4.1.1 (ii). Notons t la trace de X. Si $t = 0$ alors X est de torsion, et si $t \neq 0$ alors la matrice $\begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix}$ appartient à $\mathrm{GL}(2, K)$ car le déterminant est non nul. De plus, il est facile de vérifier que la matrice $\begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix}$ est de torsion si et seulement si t est de torsion, ce qui est équivalent à dire que X est de torsion, vu le Lemme 4.1.1(ii). Donc $\mathcal{X}$ n'est pas un code.

□

Corollaire 4.1.3. *Soit K un champ avec des opérations calculables. Si le problème* $\textsc{Free}[K^{2\times 2}]$ *est indécidable alors le problème* $\textsc{Accept}[K^{2\times 2}]$ *aussi.*

Démonstration. Procédons par contraposée et supposons que le problème $\textsc{Accept}[K^{2\times 2}]$ est décidable. Alors le problème $\textsc{Accept}[\mathrm{GL}(2, K)]$ est décidable car c'est une réduction du premier problème. Par le Théorème 3.5.8, comme $\mathrm{GL}(2, K)$ est un groupe, alors le problème $\textsc{Free}[\mathrm{GL}(2, K)]$ est décidable. Donc par la Proposition 4.1.2, le problème $\textsc{Free}[K^{2\times 2}]$ est décidable. □

Remarquons qu'on ne peut pas en conclure directement que la réponse à la Question 1 est "oui", il faudrait d'abord s'attaquer au problème suivant :

Question ouverte 2 : Existe-t-il un semi-anneau commutatif récursif D satisfaisant les deux propriétés suivantes : $\textsc{Free}(1)[D^{2\times 2}]$ est décidable et $\textsc{Accept}[D^{2\times 2}]$ est indécidable ?

4.2 Concernant la décidabilité

Dans cette section, nous allons nous pencher sur la question suivante et tenter d'y apporter des éléments de réponse en se restreignant à des matrices particulières.

Question ouverte 3 : Le problème $\textsc{Free}(2)[\mathbb{N}^{2\times 2}]$ est-t-il décidable ?

Cassaigne et Harju [9] ont étudié ce problème et ont tenté de trouver des sous-problèmes qui seraient décidables. Malheureusement, ces sous-problèmes sont fort restreints et ne découlent que sur des conditions suffisantes.

Remarquons tout de même que si $\textsc{Free}(k)[\mathbb{N}^{2\times 2}]$ est décidable pour un entier $k \geq 2$, alors par le Corollaire 6.2.5 que l'on démontrera dans le Chapitre 6, le problème $\textsc{Free}(2)[\mathbb{N}^{2\times 2}]$ est décidable.

Nous allons regarder plusieurs exemples très restreints du problème $\textsc{Free}(2)[\mathbb{N}^{2\times 2}]$ qui fonctionnent.

4.2.1 Deux matrices triangulaires supérieures

Pour tout semi-anneau D et tout nombre entier $d \geq 1$, notons $\textsc{Tri}(d, D)$ l'ensemble de toutes les matrices triangulaires supérieures $d \times d$ à coefficients dans D. $\textsc{Tri}(d, D)$ est un sous-semi-anneau de $D^{d\times d}$, donc en particulier, $\textsc{Tri}(d, D)$ est un semi-groupe multiplicatif. Par exemple, $\textsc{Tri}(2, D)$ est l'ensemble de toutes les matrices de la forme $\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}$ avec $a, b, c \in D$.

Question ouverte 4 : Le problème $\textsc{Free}[\textsc{Tri}(2, \mathbb{N})]$ est-t-il décidable ?

Pour tous entiers $k, d \geq 1$, le problème $\textsc{Free}(k)[\textsc{Tri}(d, \mathbb{Q})]$ est décidable si et seulement si le problème $\textsc{Free}(k)[\textsc{Tri}(d, \mathbb{Z})]$ est décidable : la démonstration est la même que celle du Théorème 3.4.2. En particulier, le problème $\textsc{Free}(2)[\textsc{Tri}(2, \mathbb{Q})]$ est décidable si et seulement si le problème $\textsc{Free}(2)[\textsc{Tri}(2, \mathbb{Z})]$ l'est.

Pour tenter de répondre à la Question 4, nous allons nous restreindre aux matrices 2×2 triangulaires supérieures du type

$$D_\lambda = \begin{pmatrix} \lambda & 0 \\ 0 & 1 \end{pmatrix} \text{ et } T_\mu = \begin{pmatrix} \mu & 1 \\ 0 & 1 \end{pmatrix}$$

pour tous $\lambda, \mu \in \mathbb{C}$.

Proposition 4.2.1. *L'ensemble $\{D_2, T_2\}$ est un code pour la multiplication matricielle.*

Démonstration. Considérons deux décompositions en éléments de $\{D_2, T_2\}$:

$$D_2^{n_1} T_2^{m_1} \cdots D_2^{n_k} T_2^{m_k} = D_2^{n'_1} T_2^{m'_1} \cdots D_2^{n'_l} T_2^{m'_l} \tag{4.1}$$

et montrons que $k = l$, $n_i = n'_i$ et $m_i = m'_i$ pour tout $i \in \{1, \dots, k\}$. Tout d'abord, on peut montrer par récurrence sur n que

$$(D_2)^n = \begin{pmatrix} 2^n & 0 \\ 0 & 1 \end{pmatrix}.$$

En effet, on a $D_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ et par induction on obtient que

$$(D_2)^{n+1} = \begin{pmatrix} 2^n & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^{n+1} & 0 \\ 0 & 1 \end{pmatrix}.$$

De même, on démontre par récurrence sur m que

$$(T_2)^m = \begin{pmatrix} 2^m & 2^{m-1} + 2^{m-2} + \cdots + 2^1 + 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^m & \mathrm{val}_2(1^m) \\ 0 & 1 \end{pmatrix}.$$

De plus, pour tous $m, n \in \mathbb{N}$ et tout mot $w \in \{0,1\}^*$ de longueur m, on a

$$T_2^n \begin{pmatrix} 2^m & \mathrm{val}_2(w) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^{m+n} & \mathrm{val}_2(w1^n) \\ 0 & 1 \end{pmatrix},$$

et $D_2^n \begin{pmatrix} 2^m & \mathrm{val}_2(w) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^{m+n} & \mathrm{val}_2(w0^n) \\ 0 & 1 \end{pmatrix}$. Ainsi l'équation (4.1) se réécrit

$$\begin{pmatrix} 2^{\sum_{i=1}^{k}(m_i+n_i)} & \mathrm{val}_2(1^{m_k}0^{n_k}\cdots 1^{m_1}0^{n_1}) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^{\sum_{i=1}^{l}(m'_i+n'_i)} & \mathrm{val}_2(1^{m'_l}0^{n'_l}\cdots 1^{m'_1}0^{n'_1}) \\ 0 & 1 \end{pmatrix}.$$

On obtient donc que $k = l$, $m_i = m'_i$ et $n_i = n'_i$ pour tout $i \in \{1, \dots, k\}$ par unicité de la décomposition en base entière. Donc l'ensemble $\{D_2, T_2\}$ est bien un code.

□

Exemple 4.2.2. Les ensembles $\left\{D_2, T_{1/2}\right\}$ et $\left\{D_{2/3}, T_{-3/5}\right\}$ ne sont pas des codes pour la multiplication matricielle car $D_2T_{1/2} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = T_{1/2}D_2T_{1/2}D_2$ et $D_{2/3}T_{-3/5}D_{2/3}T_{-3/5} = \begin{pmatrix} 4/25 & 2/5 \\ 0 & 1 \end{pmatrix} = T_{-3/5}T_{-3/5}D_{2/3}D_{2/3}$.

Exemple 4.2.3. Posons $D = D_{2/3}$ et $T = T_{3/5}$. L'ensemble $\{D, T\}$ n'est pas un code pour la multiplication matricielle car les deux produits

$$DTTTTTTTTTTDDTDDTDDDDDDDDDD$$

et

$$TTDDDDDDTTDDTDTDTDDTTDDTDTT$$

sont égaux à

$$\begin{pmatrix} \dfrac{32768}{6591796875} & \dfrac{242996824}{146484375} \\ 0 & 1 \end{pmatrix}.$$

Considérons le problème de décision $\mathcal{P}$ défini comme suit : étant donnés deux matrices

$$A = \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix} \text{ et } B = \begin{pmatrix} b & 1 \\ 0 & 1 \end{pmatrix}$$

où $a, b \in \mathbb{Q} \setminus \{-1, 0, 1\}$, déterminer si l'ensemble $\{A, B\}$ est un code pour la multiplication matricielle.
Une instance de ce problème est totalement déterminée par la paire (a, b).

Lemme 4.2.4. *Si $\{A, B\}$ n'est pas un code, alors A et B satisfont une équation $U = V$ où U et V sont des produits contenant le même nombre de A et de B. De plus, U commence par A et V commence par B.*

Démonstration. Posons $U = V$ la plus petite équation satisfaite par le semi-groupe $\{A, B\}^+$. Les deux membres de cette équation commencent avec des matrices différentes car s'ils débuttaient tous les deux avec la matrice A par exemple, alors le semi-groupe satisferait une équation plus courte : $A^{-1}U = A^{-1}V$. S'ils n'ont pas le même nombre de A que de B, alors il suffit de considérer l'équation $UV = VU$. □

Lemme 4.2.5. *Les instances (a, b), (b, a), $(\frac{1}{a}, \frac{1}{b})$ et $(\frac{1}{b}, \frac{1}{a})$ du problème $\mathcal{P}$ sont équivalentes.*

On peut maintenant établir deux conditions suffisantes pour déterminer si le semi-groupe $\{A, B\}^+$ est libre. La première utilise une nouvelle notion : si p est un nombre premier et $x \in \mathbb{Q}$, on note $v_p(x)$ le *nombre p-adique* de x, défini par $v_p(p^n \frac{y}{x}) = n$ si $n, y, z \in \mathbb{Z}$ et p ne divise ni z ni y. Par convention, $v_p(0) = +\infty$. De plus, il satisfait les propriétés suivantes : $v_p(mn) = v_p(m) + v_p(n)$ et $v_p(m + n) = \min\{v_p(m), v_p(n)\}$ si $v_p(m) \neq v_p(n)$ pour tous $m, n \in \mathbb{Z}$.

Proposition 4.2.6. *S'il existe un nombre p tel que $v_p(a) > 0$ et $v_p(b) > 0$, alors l'ensemble $\{A, B\}$ est un code.*

Démonstration. Supposons avoir deux décompositions en éléments de $\{A, B\}$ qui soient égales. Par le Lemme 4.2.4, on peut supposer qu'elles ont la forme $AX = BY$. Les matrices A et B peuvent être vues comme des matrices à coefficients dans $\mathbb{Z}_p$. Posons

$$X = \begin{pmatrix} x & x' \\ 0 & 1 \end{pmatrix} \text{ et } Y = \begin{pmatrix} y & y' \\ 0 & 1 \end{pmatrix}.$$

Ainsi, la matrice $AX = \begin{pmatrix} ax & ax' \\ 0 & 1 \end{pmatrix}$ est telle que $v_p(ax') = v_p(a) + v_p(x') \geq v_p(a) > 0$ alors que la matrice $BY = \begin{pmatrix} by & by' + 1 \\ 0 & 1 \end{pmatrix}$ est telle que $v_p(by' + 1) = 0$ car $v_p(by') > 0$ et $v_p(1) = 0$. Donc la relation $AX = BY$ ne tient pas. □

Cela montre en particulier que l'ensemble $\{D_2, T_2\}$ de la Proposition 4.2.1 est un code.

La deuxième condition suffisante est la suivante :

Lemme 4.2.7. *Soit (a, b) une instance du problème $\mathcal{P}$ avec $|a| < 1$ et $|b| < 1$. Si $X = \begin{pmatrix} x & x' \\ 0 & 1 \end{pmatrix} \in \{A, B\}^+$, alors $x' \in I$ où I est un intervalle défini par :*

- *si $a > 0$ et $b > 0$, alors $I = [0; \frac{1}{1-b}[$;*
- *si $a < 0$ et $b > 0$, alors $I =]\frac{a}{1-b}; \frac{1}{1-b}[$;*
- *si $a > 0$ et $b < 0$, alors $I = [0, 1]$;*
- *si $a < 0$ et $b < 0$, alors $I =]\frac{a}{1-ab}; \frac{1}{1-ab}[$.*

Démonstration. Dans les quatre cas on a clairement $0, 1 \in I$. De plus, comme $|a| < 1$ on a $aI \subset I$, et comme $|b| < 1$, on a $bI + 1 \subset I$. □

Proposition 4.2.8. *Si $|a| + |b| \leq 1$, alors l'ensemble $\{A, B\}$ est un code.*

Démonstration. Supposons qu'il existe une relation $AX = BY$ avec $X \in \{A, B\}^+$. L'élément dans le coin supérieur droit de $AX = BY$ doit appartenir à $aI \cap (bI + 1)$. Mais comme $|a| + |b| \leq 1$ alors cette intersection est vide sauf si $a > 0$, $b < 0$, $I = [0, 1]$ et $|a| + |b| = 1$. Ainsi on a $aI \cap (bI + 1) = \{a\} = \{1 + b\}$. Dans ce cas, grâce au Lemme 4.2.4, on peut supposer qu'il y a le même nombre de fois la matrice A dans chaque membre de l'équation. Ainsi, les seules matrices de $\{A, B\}^+$ ayant un 1 en position supérieure-droite sont de la forme BA^k, or X et Y ne peuvent pas être tous les deux de cette forme. □

En utilisant la Proposition précédente ainsi que le Lemme 4.2.5, on peut conclure que l'ensemble $\{A, B\}$ est un code pour la multiplication matricielle si $\frac{1}{|a|} + \frac{1}{|b|} \leq 1$.

Par exemple, l'ensemble $\{D_2, T_3\} = \left\{ \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix} \right\}$ est un code.

4.2.2 Une matrice triangulaire supérieure et une matrice triangulaire inférieure

Posons

$$A_\lambda = \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} \text{ et } B_\lambda = \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix}$$

pour tout $\lambda \in \mathbb{C}$. Notons Λ l'ensemble des $\lambda \in \mathbb{C}$ tels que $\{A_\lambda, B_\lambda\}$ n'est pas un code pour la multiplication matricielle.

L'étude de Λ a été initiée par Brenner et Charnow [3], et notre motivation à la poursuivre réside dans le fait que Free(2)$[\mathbb{Q}^{2\times 2}]$ est décidable si et seulement si $\Lambda \cap \mathbb{Q}$ est récursif. On démontre tout d'abord que $\Lambda = -\Lambda$.

Lemme 4.2.9. *Pour tout $\lambda \in \mathbb{C}$, l'ensemble $\{A_\lambda, B_\lambda\}$ est un code pour la multiplication matricielle si et seulement si $\{A_{-\lambda}, B_{-\lambda}\}$ en est un.*

Démonstration. Pour tout groupe G et tout sous-ensemble $X \subseteq G$, X est un code si et seulement si $\{x^{-1} : x \in X\}$ est un code. De plus, on a $A_\lambda^{-1} = A_{-\lambda}$ et $B_\lambda^{-1} = B_{-\lambda}$ pour tout $\lambda \in \mathbb{C}$ donc on a bien le résultat attendu. □

Montrons maintenant que chaque élément de $\Lambda \cap \mathbb{R}$ appartient à $]-1, 1[$.

Proposition 4.2.10. *Pour tout nombre réel λ tel que $|\lambda| \geq 1$, l'ensemble $\{A_\lambda, B_\lambda\}$ est un code pour la multiplication matricielle.*

Démonstration. Par le Lemme 4.2.9, on sait qu'il suffit de démontrer que l'ensemble $\{A_\lambda, B_\lambda\}$ est un code pour tout nombre réel $\lambda \geq 1$. Notons $\mathcal{A}$ l'ensemble des vecteurs colonnes $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{2\times 1}$ tels que $0 < y < x$ et $\mathcal{B}$ l'ensemble des vecteurs colonnes $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{2\times 1}$ tels que $0 < x < y$. On peut remarquer que pour tout nombre réel $x, y > 0$, on a $A_\lambda \begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{A}$ car

$$\begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + \lambda y \\ y \end{pmatrix}$$

avec $y < x + \lambda y$. De même, on a $B_\lambda \begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{B}$ car

$$\begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ \lambda x + y \end{pmatrix}$$

avec $x < \lambda x + y$. Ensuite, posons $M, N \in \{A_\lambda, B_\lambda\}^*$. En appliquant successivement la remarque ci-dessus, on obtient que $A_\lambda M \begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \mathcal{A}$ et $B_\lambda N \begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \mathcal{B}$. Comme $\mathcal{A} \cap \mathcal{B} = \emptyset$, alors on a $A_\lambda M \neq B_\lambda N$. On conclut, grâce à la contraposée du Lemme 3.2.3, que l'ensemble $\{A_\lambda, B_\lambda\}$ est un code pour la multiplication matricielle. □

Remarque 4.2.11. L'ensemble Λ contient aussi des nombres complexes de module ≥ 1. Par exemple, il contient le nombre imaginaire i car

$$A_i B_i A_i = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} = B_i A_i B_i.$$

Finalement, démontrons le résultat important de cette section, à savoir : la borne supérieure de l'ensemble $\Lambda \cap \mathbb{Q}$ est égale à 1.

Lemme 4.2.12. *Soit λ un nombre réel. S'il existe deux nombres entiers $m, n \geq 1$ tels que*

$$\lambda^2 = \frac{mn - m - n - 1}{mn} \tag{4.2}$$

alors l'ensemble $\{A_\lambda, B_\lambda\}$ n'est pas un code pour la multiplication matricielle.

Démonstration. Soient $m, n \in \mathbb{N}$. On montre par récurrence sur m et n respectivement que

$$A_\lambda^m = A_{m\lambda} \tag{4.3}$$

et

$$B_\lambda^n = B_{n\lambda}. \tag{4.4}$$

En effet, si $m = 0$ on a $A_\lambda^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = A_{0\lambda}$. Ensuite, on suppose l'équation (4.3) vérifiée pour $m \in \mathbb{N}$ et on le montre pour $m+1$:

$$A_\lambda^{m+1} = \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix}^{m+1} = \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix}^m \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & m\lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & (m+1)\lambda \\ 0 & 1 \end{pmatrix} = A_{(m+1)\lambda}.$$

On effectue une preuve similaire pour démontrer l'égalité (4.4).

Ensuite, on effectue les calculs suivants :

$$\begin{aligned} A_\lambda B_\lambda^n A_\lambda^m B_\lambda &= \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix}^n \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix}^m \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ n\lambda & 1 \end{pmatrix} \begin{pmatrix} 1 & m\lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1+n\lambda^2 & \lambda \\ n\lambda & 1 \end{pmatrix} \begin{pmatrix} 1+m\lambda^2 & m\lambda \\ \lambda & 1 \end{pmatrix} \\ &= \begin{pmatrix} (1+n\lambda^2)(1+m\lambda^2)+\lambda^2 & (1+n\lambda^2)m\lambda+\lambda \\ n\lambda(1+m\lambda^2)+\lambda & nm\lambda^2+1 \end{pmatrix} \\ &= \begin{pmatrix} mn\lambda^4+(m+n+1)\lambda^2+1 & mn\lambda^3+(m+1)\lambda \\ mn\lambda^3+(n+1)\lambda & mn\lambda^2+1 \end{pmatrix}. \end{aligned}$$

De même, on obtient

$$B_\lambda A_\lambda^m B_\lambda^n A_\lambda = \begin{pmatrix} mn\lambda^2+1 & mn\lambda^3+(m+1)\lambda \\ mn\lambda^3+(n+1)\lambda & mn\lambda^4+(m+n+1)\lambda^2+1 \end{pmatrix}.$$

Ainsi, on a l'égalité $A_\lambda B_\lambda^n A_\lambda^m B_\lambda = B_\lambda A_\lambda^m B_\lambda^n A_\lambda$ si et seulement si

$$\begin{aligned} & mn\lambda^2+1 = mn\lambda^4+(m+n+1)\lambda^2+1 \\ \Leftrightarrow\ & nm\lambda^4+(n+m+1-nm)\lambda^2 = 0 \\ \Leftrightarrow\ & \lambda^2(nm\lambda^2+(n+m+1-nm)) = 0, \end{aligned}$$

c'est-à-dire si et seulement si m et n satisfont l'équation (4.2) avec $mn \neq 0$. □

Proposition 4.2.13. *La borne supérieure de $\Lambda \cap \mathbb{R}$ est égale à 1.*

Démonstration. Considérons $\lambda_n = \sqrt{1 - 2n^{-1} - n^{-2}}$ pour tout nombre entier $n \geq 3$. Alors λ_n tend vers 1 lorsque n tend vers l'infini. De plus, grâce au Lemme 4.2.12 on a $\lambda_n \in \Lambda$ lorsqu'on considère le cas particulier où $m = n$ dans l'équation (4.2). □

Proposition 4.2.14. *Pour tout nombre réel $\delta > 0$ il existe $\lambda \in \mathbb{Q}$ tel que $1 - \delta < \lambda < 1$ et $\{A_\lambda, B_\lambda\}$ n'est pas un code pour la multiplication matricielle.*

Démonstration. Considérons la suite de nombre $(n_0, n_1, n_2, n_3, \dots)$ définie de manière récursive par

$$\begin{cases} n_0 = 3, \\ n_1 = 6 \text{ et} \\ n_{k+2} = 6n_{k+1} - n_k - 6 \ \forall k \in \mathbb{N}\,. \end{cases}$$

On peut démontrer par récurrence sur k que

$$n_k = \frac{3}{4}\left[(3 + 2\sqrt{2})^k + (3 - 2\sqrt{2})^k\right] + \frac{3}{2}$$

pour tout nombre $k \in \mathbb{N}$. En effet, si $k = 0$ on obtient bien $n_0 = 3$ et si $k = 1$ on obtient $n_1 = 6$. Ensuite, si on suppose que la relation est vérifiée pour tout nombre $k \in \mathbb{N}$, démontrons-la pour $k + 1$. Par définition, on a $n_{k+1} = 6n_k - n_{k-1} - 6$. En appliquant l'hypothèse de récurrence plusieurs fois, on obtient

$$\begin{aligned} n_{k+1} &= \frac{18}{4}\left[(3 + 2\sqrt{2})^k + (3 - 2\sqrt{2})^k\right] + \frac{18}{2} - \frac{3}{4}\left[(3 + 2\sqrt{2})^{k-1} + (3 - 2\sqrt{2})^{k-1}\right] - \frac{3}{2} - 6 \\ &= \frac{3}{2}\left(3 + 2\sqrt{2}\right)^{k-1}\left(3(3 + 2\sqrt{2}) - \frac{1}{2}\right) + \frac{3}{2}\left(3 - 2\sqrt{2}\right)^{k-1}\left(3(3 - 2\sqrt{2}) - \frac{1}{2}\right) + \frac{3}{2} \\ &= \frac{3}{4}\left[\left(3 + 2\sqrt{2}\right)^{k+1} + \left(3 - 2\sqrt{2}\right)^{k+1}\right] + \frac{3}{2}. \end{aligned}$$

Donc n_k est positif pour tout $k \in \mathbb{N}$. De plus, le nombre

$$\lambda_k = 1 - \frac{n_{k+1} + n_k + 3}{2n_{k+1}n_k}$$

est un nombre rationnel qui tend vers 1 lorsque k tend vers l'infini.

Définissons maintenant le polynôme à deux variables suivant

$$p(x, y) = x^2 + y^2 - 6xy + 6x + 6y + 9.$$

Il satisfait les deux équations suivantes :

$$p(6x - y - 6, x) = p(x, y) \text{ et} \tag{4.5}$$

$$\left(1 - \frac{x + y + 3}{2xy}\right)^2 - \frac{xy - x - y - 1}{xy} = \frac{p(x, y)}{4x^2y^2}. \tag{4.6}$$

En effet,

$$\begin{aligned} p(6x-y-6,x) &= (6x-y-6)^2 + x^2 - 6(6x-y-6)x + 6(6x-y-6) + 6x + 9 \\ &= 36x^2 - 12x(y+6) + y^2 + 12y + 36 - 36x^2 + 6xy + 36x + 36x - 6y \\ &\quad - 36 + x^2 + 6x + 9 \\ &= x^2 + y^2 - 6xy + 6x + 6y + 9 = p(x,y) \end{aligned}$$

et

$$\begin{aligned} \left(1 - \frac{x+y+3}{2xy}\right)^2 - \frac{xy-x-y-1}{xy} &= 1 - \frac{x+y+3}{xy} + \left(\frac{x+y+3}{2xy}\right)^2 - \frac{xy-x-y-1}{xy} \\ &= \frac{4x^2y^2 - 4xy(x+y+3) + (x+y+3)^2}{4x^2y^2} \\ &\quad - \frac{4xy(xy-x-y-1)}{4x^2y^2} \\ &= \frac{p(x,y)}{4x^2y^2}. \end{aligned}$$

Enfin, en se basant sur l'équation (4.5), on démontre par induction que

$$p(n_{k+1}, n_k) = 0 \tag{4.7}$$

pour tout $k \in \mathbb{N}$. En effet, si $k = 0$, alors $p(n_1, n_0) = p(6,3) = 6^2+3^2-6\cdot6\cdot6+6\cdot3+6\cdot3+9 = 0$. Maintenant supposons que l'équation (4.7) est satisfaite pour tout $k \in \mathbb{N}$ et montrons-le pour $k+1$. On a $p(n_{k+2}, n_{k+1}) = p(6n_{k+1} - n_k - 6, n_{k+1}) = p(n_{k+1}, n_k) = 0$.

Par conséquent, l'équation (4.6) implique

$$\left(1 - \frac{n_{k+1}+n_k+3}{2n_{k+1}n_k}\right)^2 - \frac{n_{k+1}n_k - n_{k+1} - n_k - 1}{n_{k+1}n_k} = 0,$$

c'est-à-dire $\lambda_k^2 = \dfrac{n_{k+1}n_k - n_{k+1} - n_k - 1}{n_{k+1}n_k}$. Ainsi, par le Lemme 4.2.12, on en conclut que l'ensemble $\{A_{\lambda_k}, B_{\lambda_k}\}$ n'est pas un code pour la multiplication matricielle. □

Un question persiste néanmoins :

Question ouverte 5 : Existe-t-il un nombre rationnel λ tel que $|\lambda| < 1$ et tel que l'ensemble $\{A_\lambda, B_\lambda\}$ est un code pour la multiplication matricielle ?

4.3 Substitutions sur l'alphabet binaire

Dans cette section, nous allons nous pencher sur l'étude du problème FREE(2)[hom($\mathbb{W}$)].

Pour cela, considérons la fonction de hom($\mathbb{W}$) dans $\mathbb{N}^{2\times 2}$ qui envoie chaque $\sigma \in \hom(\mathbb{W})$ sur la matrice

$$P_\sigma = \begin{pmatrix} |\sigma(0)|_0 & |\sigma(1)|_0 \\ |\sigma(0)|_1 & |\sigma(1)|_1 \end{pmatrix}.$$

Comme vu précédemment, P_σ est la matrice d'incidence de σ relative à 01. La fonction considérée est clairement surjective et c'est un morphisme grâce à la Proposition 2.3.3. Ainsi, les semi-groupes hom($\mathbb{W}$) et $\mathbb{N}^{2\times 2}$ ont des structures très similaires donc se poser la question de la décidabilité du problème Free(2)[hom($\mathbb{W}$)] est très proche de la question de la décidabilité du problème Free(2)[$\mathbb{N}^{2\times 2}$]. Néanmoins, on ne sait pas dire laquelle des deux questions est la plus facile à résoudre.

La proposition suivante nous donne une façon de générer une instance positive de Free(2)[hom($\mathbb{W}$)] à partir d'une instance positive de Free(2)[$\mathbb{N}^{2\times 2}$].

Proposition 4.3.1. *Soient $\sigma, \tau \in hom(\mathbb{W})$ tels que $P_\sigma \neq P_\tau$. Si l'ensemble $\{P_\sigma, P_\tau\}$ est un code pour la multiplication matricielle, alors l'ensemble $\{\sigma, \tau\}$ est un code pour la composition de fonctions.*

Démonstration. C'est un corollaire immédiat de la Propriété 2.3.3 (i). □

Nous allons construire quatre instances positives de Free(2)[hom($\mathbb{W}$)] :

Exemple 4.3.2. Pour chaque $p \in \mathbb{N}$, on définit $\delta_p, \tau_p, \tau'_p \in \hom(\mathbb{W})$ par :

$$\begin{cases} \delta_p(0) = 0^p \\ \delta_p(1) = 1 \end{cases}, \quad \begin{cases} \tau_p(0) = 0^p \\ \tau_p(1) = 10 \end{cases}, \quad \begin{cases} \tau'_p(0) = 0^p \\ \tau'_p(1) = 01 \end{cases}.$$

Ainsi, grâce aux matrices définies à la section 4.2.1, on obtient que pour tout $p \in \mathbb{N}$ on a

$$P_{\delta_p} = \begin{pmatrix} |\delta_p(0)|_0 & |\delta_p(1)|_0 \\ |\delta_p(0)|_1 & |\delta_p(1)|_1 \end{pmatrix} = \begin{pmatrix} p & 0 \\ 0 & 1 \end{pmatrix} = D_p \text{ et } P_{\tau_p} = \begin{pmatrix} p & 1 \\ 0 & 1 \end{pmatrix} = T_p = P_{\tau'_p}.$$

Par conséquent, en utilisant l'exemple 4.2.1 et la Propriété 4.3.1, on en déduit que $\{P_{\delta_2}, P_{\tau_2}\}$, $\{P_{\delta_2}, P_{\tau'_2}\}$, $\{P_{\delta_2}, P_{\tau_3}\}$ et $\{P_{\delta_2}, P_{\tau'_3}\}$ sont des codes pour la multiplication matricielle. Donc les ensembles $\{\delta_2, \tau_2\}$, $\{\delta_2, \tau'_2\}$, $\{\delta_2, \tau_3\}$ et $\{\delta_2, \tau'_3\}$ sont des codes pour la composition de fonctions.

Les deux instances positives suivantes de Free(2)[hom($\mathbb{W}$)] ne seront pas obtenues par la méthode précédente, mais on va plutôt s'aider du Lemme 3.2.3. Pour cela, il faut que notre ensemble hom($\mathbb{W}$) soit simplifiable à gauche, ce qui est direct grâce à l'exemple 3.2.2, car les fonctions injectives sont simplifiables à gauche pour la composition, et que vérifier l'injecivité d'un élément donné $\sigma \in \hom(\mathbb{W})$ est trivial. En effet, σ injectif si et seulement si $\sigma(01) \neq \sigma(10)$.

Exemple 4.3.3. Posons $v, v' \in \hom(\mathbb{W})$ définis par

$$\begin{cases} v(0) = 01 \\ v(1) = 011 \end{cases} \text{et} \begin{cases} v'(0) = 10 \\ v'(1) = 110 \end{cases}.$$

Pour tout $x \in \{0,1\}^+$, $v(x)$ commence par un 0 alors que $v'(x)$ commence par un 1. Par conséquent, pour tous $\alpha, \alpha' \in \hom(\mathbb{W})$ on a $v\alpha \neq v'\alpha'$ sauf si $\alpha(0) = \alpha(1) = \alpha'(0) = \alpha'(1) = \epsilon$. On peut maintenant utiliser le Lemme 3.2.3 et il en découle que l'ensemble $\{v, v'\}$ est un code pour la composition de fonctions.

Remarquons que $P_v = P_{v'} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$.

Exemple 4.3.4. Soient ϕ et $\mu \in \hom(\mathbb{W})$ définis par

$$\begin{cases} \phi(0) = 01 \\ \phi(1) = 0 \end{cases} \text{et} \begin{cases} \mu(0) = 01 \\ \mu(1) = 10 \end{cases}.$$

Les morphismes ϕ et μ sont habituellement appelés *le morphisme de Fibonacci* et *le morphisme de Thue-Morse* respectivement. Posons $\alpha, \beta \in \{\phi, \mu\}^+$, on voit clairement que $\alpha(0)$ et $\beta(0)$ commencent par 01. Donc on a $(\phi\alpha)(0) = \phi(01\dots) = \phi(0)\phi(1)\cdots = 010\dots$ et $(\mu\beta)(0) = \mu(01\dots) = \mu(0)\mu(1)\cdots = 0110\dots$. Par conséquent, $\phi\alpha \neq \mu\beta$; et par le Lemme 3.2.3 on sait que l'ensemble $\{\phi, \mu\}$ est un code pour la composition de fonctions. Remarquons quand même que

$$P_\phi = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \neq \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = P_\mu \text{ et } P_\mu P_\mu P_\phi P_\mu = P_\mu P_\phi P_\mu P_\mu.$$

Par conséquent, l'ensemble $\{P_\phi, P_\mu\}$ n'est pas un code pour la multiplication matricielle.

La ressemblance entre la proposition suivante et le Lemme 4.1.1 (i) nous montre d'autres ressemblances entre $\hom(\mathbb{W})$ et $\mathbb{N}^{2\times 2}$.

Proposition 4.3.5. *Soit $\sigma \in hom(\mathbb{W})$ un homomorphisme non-injectif. Pour tout $\tau \in hom(\mathbb{W})$, on a l'égalité $\sigma\sigma\tau\sigma = \sigma\tau\sigma\sigma$.*

Démonstration. Posons $\alpha = \sigma\sigma\tau\sigma$ et $\beta = \sigma\tau\sigma\sigma$. Comme σ n'est pas injectif, on a $\sigma(01) = \sigma(10)$ et donc il existe $s \in \mathbb{W}$ et $p, q \in \mathbb{N}$ tels que $\sigma(0) = s^p$ et $\sigma(1) = s^q$. Tout d'abord, la matrice

$$P_\sigma = \begin{pmatrix} p|s|_0 & q|s|_0 \\ p|s|_1 & q|s|_1 \end{pmatrix}$$

est singulière. Donc par le Lemme 4.1.1 on a $P_\sigma P_\sigma P_\tau P_\sigma = P_\sigma P_\tau P_\sigma P_\sigma$. Ainsi grâce à la Proposition 2.3.3, on a $P_\alpha = P_{\sigma\sigma\tau\sigma} = P_{\sigma\tau\sigma\sigma} = P_\beta$.

Ensuite, posons $x \in \mathbb{W}$. Ainsi pour tout $\rho \in \hom(\mathbb{W})$ on a

$$\begin{aligned} P_\rho \begin{pmatrix} |x|_0 \\ |x|_1 \end{pmatrix} &= \begin{pmatrix} |\rho(0)|_0 & |\rho(1)|_0 \\ |\rho(0)|_1 & |\rho(1)|_1 \end{pmatrix} \begin{pmatrix} |x|_0 \\ |x|_1 \end{pmatrix} \\ &= \begin{pmatrix} |\rho(0)|_0|x|_0 + |\rho(1)|_0|x|_1 \\ |\rho(0)|_1|x|_0 + |\rho(1)|_1|x|_1 \end{pmatrix} \\ &= \begin{pmatrix} |\rho(x)|_0 \\ |\rho(x)|_1 \end{pmatrix}, \end{aligned}$$

et donc

$$\begin{pmatrix} 1 & 1 \end{pmatrix} P_\rho \begin{pmatrix} |x|_0 \\ |x|_1 \end{pmatrix} = |\rho(x)|_0 + |\rho(x)|_1 = |\rho(x)|.$$

Cette dernière égalité est valable en particulier pour $\rho = \alpha$ et $\rho = \beta$. On a donc

$$|\alpha(x)| = \begin{pmatrix} 1 & 1 \end{pmatrix} P_\alpha \begin{pmatrix} |x|_0 \\ |x|_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix} P_\beta \begin{pmatrix} |x|_0 \\ |x|_1 \end{pmatrix} = |\beta(x)|.$$

Enfin, comme σ envoie chaque élément de $\mathbb{W}$ sur une puissance de s, on sait que l'image $\alpha(x) = \sigma(\sigma(\tau(\sigma(x))))$ est une puissance de s. De même, $\beta(x) = \sigma(\tau(\sigma(\sigma(x))))$ est aussi une puissance de s. Comme de plus $\alpha(x)$ et $\beta(x)$ sont de même longueur, alors ils sont égaux à la même puissance de s. □

4.4 Problème de mortalité sur les matrices 2×2

Nous allons nous concentrer sur le problème de décision MORTAL[$\mathbb{Q}^{n\times n}$] défini dans la Section 1.4. On montrera dans le Chapitre 6 que pour $n \geq 3$, le problème MORTAL[$\mathbb{Q}^{n\times n}$] se réduit à un autre problème de décision, qui est indécidable. Pour $n = 1$, le problème MORTAL[$\mathbb{Q}^{n\times n}$] est trivialement décidable. Il nous reste donc à regarder le cas où $n = 2$. Nous allons effectuer une discussion sur les matrices 2×2 réalisée par MILLER [15]. Pour cela, commençons par redéfinir le problème de décision MORTAL[$\mathbb{Q}^{2\times 2}$] : étant donné un ensemble fini de matrices $P \subseteq \mathbb{Q}^{n\times n}$, déterminer s'il existe un produit fini de matrices de P égal à la matrice nulle.

Ensuite, introduisons une définition qui nous sera utile dans la démonstration :

Définition 4.4.1. Un ensemble fini de matrices possède un *produit d'égalisation* s'il existe un produit de matrices de l'ensemble égal à une matrice dont certaines entrées pré-spécifiées sont égales.

Considérons des matrices carrées 2×2 à coefficients entiers. Certains cas particuliers du problème MORTAL[$\mathbb{Q}^{2\times 2}$] ont déjà été démontrés. Par exemple, si les matrices sont toutes triangulaires supérieures ou triangulaires inférieures, il suffit de vérifier qu'il existe, pour toute position diagonale, au moins une matrice de l'ensemble avec un zéro à cette position.

KROM et KROM ([12]) ont démontré que le problème suivant était équivalent au problème MORTAL[$\mathbb{Q}^{2\times 2}$] :

(1) Étant donné un ensemble fini P de matrices 2×2 inversibles à coefficients entiers, déterminer s'il existe un produit d'éléments de P égal à une matrice quelconque $\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ où $c_{21} = c_{22}$.

Les mêmes auteurs ont ensuite proposé un sous-problème du problème (1) :

(2) Étant donné un ensemble fini P de matrices triangulaires inférieures 2×2 inversibles et à coefficients entiers, déterminer s'il existe un produit d'éléments de P égal à une matrice quelconque $\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ où $c_{21} = c_{22}$.

Si le problème (2) est indécidable, alors le problème (1) l'est aussi et donc il en est de même du problème Mortal[$\mathbb{Q}^{2\times 2}$]. Par contre, si le problème (1), ou même le problème (2), est indécidable, il peut exister des cas particuliers dans lesquels le problème Mortal[$\mathbb{Q}^{2\times 2}$] est décidable. C'est ce que nous allons monter. On va pour cela imposer des restrictions supplémentaires au problème (2) :

Soit un ensemble fini P de matrices triangulaires inférieures 2×2 inversibles et à coefficients entiers. Supposons que les termes de toutes les matrices sont de même signe, disons positif, et que le terme dans le coin supérieur gauche est plus grand ou égal au terme dans le coin inférieur droit. (On peut en effet supposer que toutes les entrées sont positives car multiplier par -1 ne modifiera en rien l'existence d'un produit d'égalisation). On considère donc l'ensemble

$$P = \left\{ \begin{pmatrix} m_{11} & 0 \\ m_{21} & m_{22} \end{pmatrix} : m_{ij} \in \mathbb{N}, m_{11} \geq m_{22} \text{ et } m_{11}m_{22} \neq 0 \right\}.$$

Proposition 4.4.2. *Le problème de déterminer s'il existe un produit de matrices de P égal à une matrice quelconque de la forme*

$$E = \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{pmatrix}$$

où $e_{21} = e_{22}$, est décidable.

Démonstration. Nous allons construire un algorithme pour démonter que le problème est décidable. Comme multiplier une matrice de P par un scalaire ne change en rien le fait d'obtenir un tel produit d'égalisation, on peut multiplier chaque matrice par l'inverse du terme dans le coin inférieur droit, c'est-à-dire par $1/m_{22}$ respectivement. Cela nous permet d'obtenir un ensemble P' de matrices de la forme $\begin{pmatrix} a & 0 \\ b & 1 \end{pmatrix}$ où $a \geq 1$.

Remarquons que lorsqu'on multiplie entre-elles deux matrices de ce type, on obtient

$$\begin{pmatrix} a_1 & 0 \\ b_1 & 1 \end{pmatrix} \begin{pmatrix} a_2 & 0 \\ b_2 & 1 \end{pmatrix} = \begin{pmatrix} a_1a_2 & 0 \\ b_1a_2 + b_2 & 1 \end{pmatrix},$$

donc $e_{21} = a_2b_1 + b_2$ et $e_{22} = 1$.

Ensuite si on multiplie trois de ces matrices, on obtient

$$e_{21} = a_3a_2b_1 + a_3b_2 + b_3 = a_3(a_2b_1 + b_2) + b_3 \text{ et } e_{22} = 1.$$

Donc si on multiplie m matrices de ce type, on a

$$e_{12} = a_mx_m + b_m \text{ et } e_{22} = 1$$

où

$$x_1 = 0 \text{ et } x_{m+1} = a_mx_m + b_m.$$

Il reste donc à déterminer s'il existe un ensemble de matrices $\begin{pmatrix} a_i & 0 \\ b_i & 1 \end{pmatrix} \in P'$ pour $i \in \{1, \ldots, m\}$ telles que

$$a_mx_m + b_m = 1, \text{ c'est-à-dire } x_m = \frac{1 - b_m}{a_m}. \tag{4.8}$$

Pour l'algorithme, pour tester si un ensemble fini P de matrices possède un produit d'égalisation, on forme d'abord le nouvel ensemble P', ensuite on attribue un ordre à P' et on teste chaque matrice de P' dans cet ordre pour savoir si elle pourrait être la dernière matrice (la m-ième) dans un produit d'égalisation de m matrices. On construit l'algorithme comme suit :

1. Si toutes les matrices de P' sont diagonales alors l'algorithme s'arrête et rend la réponse "non" (il n'y a pas de produit d'égalisation). Sinon, on pose $i = 0$.
2. On pose $i = i + 1$ et $m = 0$.
3. Si la $i$ème matrice est $\begin{pmatrix} a & 0 \\ b & 1 \end{pmatrix}$, on pose $d = \frac{1-b}{a}$.
4. Si $d = 0$, l'algorithme s'arrête et rend la réponse "oui" (la $i$ème matrice de P est déjà un produit d'égalisation). Si $d < 0$, on retourne à l'étape 2. Sinon, on passe à l'étape 5.
5. On pose $m = m + 1$.
6. Considérons la $i$ème matrice de P' comme étant le $m$ème facteur d'un produit de m matrices. Si $m = 1$, on retourne à l'étape 5.
7. Passer en revue les n^{m-1} options qu'il y a pour former les $m-1$ matrices du produit, calculer les x_m correspondant et les retenir.
8. S'il existe $x_m = d$, alors l'algorithme s'arrête et rend la réponse "oui".
   Si pour tout $x_m \neq 0$, $x_m > d$, alors si $i < n$, on retourne à l'étape 2 et sinon l'algorithme s'arrête et rend la réponse "non" car toutes les entrées des matrices sont positives et les x_m sont non décroissants.
   Sinon, on retourne à l'étape 5.

L'algorithme s'arrête car il y a un nombre fini de matrices à passer en revue. □

Regardons deux exemples pour illustrer cet algorithme.

Exemple 4.4.3. Considérons l'ensemble de matrices

$$P = \left\{ \begin{pmatrix} 9 & 0 \\ 3 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}, \begin{pmatrix} 9 & 0 \\ 8 & 6 \end{pmatrix}, \begin{pmatrix} 8 & 0 \\ 9 & 4 \end{pmatrix} \right\}.$$

On forme d'abord l'ensemble P' comme décrit plus haut :

$$P' = \left\{ \begin{pmatrix} \frac{9}{7} & 0 \\ \frac{3}{7} & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}, \begin{pmatrix} \frac{3}{2} & 0 \\ \frac{4}{3} & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ \frac{9}{4} & 1 \end{pmatrix} \right\}.$$

Ensuite on remarque que seule la première matrice de P pourrait convenir comme dernier facteur d'un produit d'égalisation car $d = \frac{4}{9} > 0$ pour cette matrice et $d < 0$ pour les trois autres. En avançant dans l'algorithme avec cette matrice, on se rend compte que d est strictement plus petit que tous les x_3 possibles (x_3 étant défini à l'équation (4.8) pour un produit de trois matrices de l'ensemble P'). Dans ce cas, l'algorithme rend la réponse "non" et l'ensemble P ne fournit pas de produit d'égalisation.

Exemple 4.4.4. Considérons l'ensemble de matrices

$$P = \left\{ \begin{pmatrix} 9 & 0 \\ 3 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 8 & 0 \\ 1 & 5 \end{pmatrix} \right\}.$$

On forme l'ensemble

$$P' = \left\{ \begin{pmatrix} \frac{9}{7} & 0 \\ \frac{3}{7} & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{pmatrix}, \begin{pmatrix} \frac{8}{5} & 0 \\ \frac{1}{5} & 1 \end{pmatrix} \right\}.$$

Ensuite, les deux premières matrices de P sont à éliminer directement, car pour la deuxième on a $d < 0$ et pour la première on va se retrouver dans le même cas qu'à l'exemple précédent. Lorsqu'on se concentre sur la troisième matrice, on se rend compte, en effectuant l'algorithme, que cette matrice au carré est un produit d'égalisation. Remarquons que l'algorithme peut nous fournir un produit d'égalisation, mais ne va pas nous les donner tous. Par exemple, la troisième matrice multipliée par la quatrième est aussi un produit d'égalisation.

Chapitre 5

Le problème de correspondance de Post

Dans l'histoire de la calculabilité, le problème de correspondance de Post et ses variantes ont joué un rôle important en tant que problèmes indécidables qui peuvent être utilisés pour prouver l'indécidabilité d'autres problèmes. Dans un premier temps, on se concentrera dans ce chapitre sur le problème de correspondance de Post (PCP) tel qu'il l'a initialement défini [19] et nous y apporterons quelques détails tirés de l'article de Halava [7]. Nous démontrerons son indécidabilité en se ramenant à des instances sous forme d'un système normal. Ce problème de décision sera aussi exploité dans le Chapitre 6 relatif aux matrices de plus grande dimension.

Dans un second temps, nous étudierons une version généralisée du problème de correspondance de Post (GPCP) en se basant sur l'article de Nicolas [16]. Pour cela nous introduirons les systèmes semi-Thue qui nous permettront de lier la décidabilité des problèmes PCP et GPCP, à l'aide d'un troisième problème de décision important.

5.1 Le problème de correspondance de Post

En 1946, Emil Leon Post définit le célèbre problème de correspondance [19] qui porta son nom par la suite et dont il démontra l'indécidabilité. La formulation originale du problème de correspondance de Post est la suivante :

Soit $B = \{a, b\}$ un alphabet binaire. Étant donné un ensemble fini de n paires de mots

$$W = \{(u_i, u'_i) \mid u_i, u'_i \in B^*, i \in \{1, \ldots, n\}\},$$

déterminer s'il existe une suite d'indices $i_1, i_2, \ldots, i_k$ tels que $i_j \in \{1, \ldots, n\}$ pour $1 \leq j \leq k$ et

$$u_{i_1} u_{i_2} \cdots u_{i_k} = u'_{i_1} u'_{i_2} \cdots u'_{i_k}. \tag{5.1}$$

Par exemple dans le cas où $n = 3$ et si on considère les paires de mots $(u_1, u'_1) = (bb, b)$, $(u_2, u'_2) = (ab, ba)$ et $(u_3, u'_3) = (b, bb)$ sur $\{a, b\}$, alors

$$u_1 u_2 u_2 u_3 = bbababb = u'_1 u'_2 u'_2 u'_3$$

et l'équation (5.1) possède une solution. Mais nous allons démontrer qu'en toute généralité, le problème de correspondance de Post est indécidable.

Remarquons que si pour tout $i \in \{1, \ldots, n\}$ on a $|u_i| > |u'_i|$, ou que chaque mot u_i commence avec une lettre différente de celle du mot correspondant u'_i, alors l'équation (5.1) n'a pas de solution.

Pour démontrer l'indécidabilité de ce problème, nous allons le réduire à un autre problème de décision dont on connait l'indécidabilité.

Définition 5.1.1. Soient $A = \{a, b\}$ un alphabet binaire et X une variable s'étendant sur des mots de A^*. Un *système normal* $S = (w, P)$ est constitué d'un mot initial $w \in A^+$ et d'un ensemble fini P de règles de la forme $\alpha X \mapsto X\beta$ où $\alpha, \beta \in A^*$.
On dit qu'un mot v est un *successeur* d'un mot u s'il existe une règle $\alpha X \mapsto X\beta$ dans P telle que $u = \alpha u'$ et $v = u'\beta$. On notera cela $u \to v$ et $\to^*$ sera la clôture réflexive et transitive de $\to$. Ainsi, on a $u \to^* v$ si et seulement si $u = v$ ou s'il existe une suite finie de mots $u = v_1, v_2, \ldots, v_n = v$ tels que $v_i \to v_{i+1}$ pour $i \in \{1, \ldots, n-1\}$.

L'*assertion* d'un système normal $S = (w, P)$ est l'ensemble

$$\mathcal{A}_S = \{v \in A^* \mid w \to^* v\}.$$

Le résultat suivant est cité par Post [19] et démontré par Church [5].

Théorème 5.1.2. *Le problème de déterminer, pour un système normal donné $S = (w, P)$ et un mot $u \in A^+$, si $u \in \mathcal{A}_S$, est indécidable.*

En fait, le problème reste indécidable même si on suppose que dans chaque règle $\alpha X \mapsto X\beta$ de P, les mots α et β sont non vides. On pourra donc le supposer dans la suite.

La démonstration du théorème qui suit est tirée de l'article [7] de Halava. La démonstration initialement réalisée par Post se trouve dans l'article [19].

Théorème 5.1.3. *Le problème de correspondance de Post est indécidable.*

Démonstration. Supposons avoir les équations suivantes

$$w = \alpha_{i_1} x_1,\ x_1\beta_{i_1} = \alpha_{i_2} x_2, \ldots, x_{k-1}\beta_{i_{k-1}} = \alpha_{i_k} x_k,\ , x_k\beta_{i_k} = u, \tag{5.2}$$

pour un système normal $A = (w, P)$ et un mot donné u où $\alpha_{i_j} X \mapsto X\beta_{i_j} \in P$ pour $j \in \{1, \ldots, k\}$. Ensuite on forme l'équation suivante

$$w x_1 \beta_{i_1} x_2 \beta_{i_2} \cdots x_k \beta_{i_k} = \alpha_{i_1} x_1 \alpha_{i_2} x_2 \cdots \alpha_{i_k} x_k u, \tag{5.3}$$

où l'on a concaténé les équations de (5.2) membre à membre.
Considérons deux nouvelles lettres c et f et supposons que la cardinalité de P est égale à t, ainsi on note $P = \{p_1, \cdots, p_t\}$ où $p_j = \alpha_j X \mapsto X\beta_j$ pour $j \in \{1, \ldots, t\}$. De plus, soit d une nouvelle lettre, on définit deux applications $l_d, r_d \colon \{a, b, c\}^* \to \{a, b, c, d\}^*$ telles que pour tout mot $v = a_1 a_2 \cdots a_p$ avec $a_i \in \{a, b, c\}$, on a

$$l_d(v) = d a_1 d a_2 \cdots d a_t \text{ et } r_d(v) = a_1 d a_2 d \cdots a_t d.$$

Ainsi , pour chaque $p_j \in P$, on définit deux paires de mots

$$p_j^\alpha = (l_d(c^j f), r_d(f\alpha_j)) \text{ et } p_j^\beta = (l_d(\beta_j), r_d(c^j)).$$

En fait, on sépare chaque élément de P en deux paires de mots. Le mot $c^j f$ fait office d'indicateur qui nous oblige à choisir ces paires de mots ensemble pour une solution d'une instance du problème PCP, que l'on définit par les paires de mots

$$W = \{(dl_d(fw), dd), (dd, r_d(fu)d), (da, ad), (db, bd)\} \cup \{p_j^\alpha, p_j^\beta \mid j \in \{1, \ldots, t\}\}. \quad (5.4)$$

Il est ensuite simple de montrer que $u \in \mathcal{A}_S$ si et seulement si PCP possède une solution. En effet, on remarque que toutes les solutions d'une instance de PCP sont de la forme

$$\begin{aligned}
&dl_d(fwc^{i_1} f x_1 \beta_{i_1} c^{i_2} f \cdots c^{i_k} f x_k \beta_{i_k}) dd \\
&= dl_d(fw) l_d(c^{i_1} f) l_d(x_1) l_d(\beta_{i_1}) l_d(c^{i_2} f) \cdots l_d(c^{i_k} f) l_d(x_k) l_d(\beta_{i_k}) dd \\
&= dl_d(f\alpha_{i_1} x_1) l_d(c^{i_1} f) l_d(x_1 \beta_{i_1}) l_d(c^{i_1} f) \cdots l_d(c^{i_k} f) l_d(x_k \beta_{i_k}) dd \\
&= dl_d(f\alpha_{i_1} x_1) l_d(c^{i_1} f) l_d(\alpha_{i_2} x_2) l_d(c^{i_1} f) \cdots l_d(c^{i_k} f) l_d(u) dd \\
&= ddr_d(f\alpha_{i_1}) r_d(x_1) r_d(c^{i_1}) r_d(f\alpha_{i_2}) r_d(x_2) \cdots r_d(f\alpha_{i_k}) r_d(x_k) r_d(c^{i_k}) r_d(fu) d \\
&= ddr_d(f\alpha_{i_1} x_1 c^{i_1} f \alpha_{i_2} x_2 c^{i_2} f \cdots \alpha i_k x_k c^{i_k} f u) d
\end{aligned}$$

en utilisant les équations de (5.2) pour le système normal donné S.

Enfin, remarquons que l'on est obligé de couper les règles en deux paires car les mots x_i apparaissent à droite des mots α_i et à gauche des mots β_i dans les équations (5.2). Donc α_i et β_i ne peuvent pas être utilisés dans une même paire de mots. □

Pour terminer cette section, nous allons redéfinir le problème de correspondance de Post en termes modernes. Post définit le problème pour des mots sur un alphabet binaire. En fait, la cardinalité de l'alphabet n'est pas importante car toute instance du problème avec un alphabet de taille arbitraire possède une instance équivalente sur un alphabet binaire en utilisant une application injective de B^* dans $\{a, b\}^*$. Par exemple, si on a l'alphabet $B = \{a_1, \ldots, a_k\}$ alors ϕ défini par $\phi(a_i) = a^i b$ est une application injective de B^* dans $\{a, b\}^*$.

On reformulera donc l'énoncé du problème comme suit, et il sera ainsi utilisé dans la suite.

Définition 5.1.4. Le problème de correspondance de Post, noté PCP, est défini comme suit : étant donnés un alphabet Σ et deux morphismes $\sigma, \tau \colon \Sigma^* \to \{0, 1\}^*$, déterminer s'il existe un mot $w \in \Sigma^+$ tel que $\sigma(w) = \tau(w)$. Pour tout nombre entier $k \geq 1$, le problème PCP(k) est la réduction de PCP aux instances (Σ, σ, τ) où la cardinalité de Σ est égale à k.

Nous allons maintenant étudier d'autres problèmes de décision que l'on mettra en lien avec le problème de correspondance de Post.

5.2 Le problème de correspondance de Post généralisé

Définition 5.2.1. Notons GPCP le problème suivant : étant donnés un alphabet fini Σ, deux morphismes $\sigma, \tau : \Sigma^* \to \mathbb{W}$ et $s, s', t, t' \in \mathbb{W}$, déterminer s'il existe un mot $w \in \Sigma^*$ tel que

$$s\sigma(w)t = s'\tau(w)t'.$$

Remarquons que si $st = s't'$, alors ϵ est une solution de GPCP sur $(\Sigma, \sigma, \tau, s, t, s', t')$, alors que toute solution de PCP est différente du mot vide.

Remarque 5.2.2. Pour toute instance (Σ, σ, τ) de PCP, (Σ, σ, τ) est une instance positive si et seulement si il existe $a \in \Sigma$ tel que $(\Sigma, \sigma, \tau, \sigma(a), \epsilon, \tau(a), \epsilon)$ est une instance positive de GPCP. En effet, (Σ, σ, τ) est une instance positive de PCP si et seulement si il existe $w \in \Sigma^+$ tel que $\sigma(w) = \tau(w)$, c'est-à-dire si et seulement si il existe $a \in \Sigma$ tel que $w = aw'$ avec $w' \in \Sigma^*$ tel que $\sigma(aw') = \tau(aw') \Leftrightarrow \sigma(a)\sigma(w') = \tau(a)\tau(w')$. Donc si et seulement si il existe $a \in \Sigma$ tel que $(\Sigma, \sigma, \tau, \sigma(a), \epsilon, \tau(a), \epsilon)$ est une instance positive de GPCP.

Pour tout entier $k \geq 1$, GPCP(k) est la réduction de GPCP pour les instances $(\Sigma, \sigma, \tau, s, t, s', t')$ où la cardinalité de Σ est égale à k.

Définition 5.2.3. Un *système semi-Thue* est une paire $T = (\Sigma, R)$ où Σ est un alphabet et R est un sous-ensemble de $\Sigma^* \times \Sigma^*$. Les éléments de R sont appelés les *règles* de T. Pour tous mots $x, y \in \Sigma^*$, on dit que y est *immédiatement dérivable* de x dans T, ce que l'on écrit $x \underset{T}{\longrightarrow} y$, s'il existe $s, t, z, z' \in \Sigma^*$ tels que $x = zsz', y = ztz'$ et $(s, t) \in R$.

Pour tous mots $u, v \in \Sigma^*$, on dit que u est *dérivable* de v dans T, que l'on note $u \overset{*}{\underset{T}{\longrightarrow}} v$, s'il existe un nombre entier $n \geq 0$ et des mots $x_0, x_1, \ldots, x_n \in \Sigma^*$ tels que $x_0 = u, x_n = v$ et $x_{i-1} \underset{T}{\longrightarrow} x_i$ pour tout $i \in \{1, \ldots, n\}$:

$$u = x_0 \underset{T}{\longrightarrow} x_1 \underset{T}{\longrightarrow} x_2 \underset{T}{\longrightarrow} \cdots \underset{T}{\longrightarrow} x_n = v. \tag{5.5}$$

En d'autres mots, $\overset{*}{\underset{T}{\longrightarrow}}$ est la clôture réflexive et transitive de la relation binaire $\underset{T}{\longrightarrow}$.

Définition 5.2.4. Notons Accessibility le problème suivant : étant donnés un système semi-Thue $T = (\Sigma, R)$ et deux mots $u, v \in \Sigma^*$, déterminer si u est dérivable à partir de v dans T, c'est-à-dire si on a $u \overset{*}{\underset{T}{\longrightarrow}} v$.
Pour tout nombre entier $k \geq 1$, le problème Accessibility(k) est la réduction du problème Accessibility aux instances (T, u, v) où T possède k règles.

Soit $k \in \mathbb{N}_0$, la décidabilité des problèmes Accessibility, PCP et GPCP sont liées par les quatre propositions suivantes.

Proposition 5.2.5. *Si* GPCP(k) *est décidable, alors* PCP(k) *est décidable.*

Proposition 5.2.6. *Si* GPCP($k+2$) *est décidable, alors* Accessibility(k) *est décidable.*

Proposition 5.2.7. *Si* PCP$(k+2)$ *est décidable, alors* GPCP(k) *est décidable.*

Proposition 5.2.8. *Si* PCP$(k+4)$ *est décidable, alors* Accessibility(k) *est décidable.*

Remarquons que l'on peut déduire directement la Proposition 5.2.8 des Propositions 5.2.6 et 5.2.7.

Le but ici est donc de démontrer la Proposition 5.2.8. Pour cela, nous allons d'abord donner une preuve des Propositions 5.2.6 et 5.2.7. Remarquons que la Proposition 5.2.5 découle de la remarque 5.2.2.

5.2.1 Quelques mots sur le problème Accessibility

Définition 5.2.9. On note C le langage $\{010^n101 : n \geq 2\}$. Pour tout nombre entier $k \geq 1$, on définit $\mathcal{C}_k$ l'ensemble des instances (T, u, v) de Accessibility telles que $u, v \in C^*$ et $T = (\{0,1\}, R)$ pour un sous-ensemble $R \subseteq C^* \times C^*$ à k éléments.

Le but de cette sous-section sera de démontrer la propriété suivante :

Proposition 5.2.10. *Pour tout nombre entier $k \geq 1$, le problème général* Accessibility *est décidable si et seulement si sa réduction aux instances de $\mathcal{C}_k$ est décidable.*

Pour cela nous construirons une réduction basée sur la transformation suivante :

Définition 5.2.11. Soient $T = (\Sigma, R)$ un système semi-Thue, Δ un alphabet et une application $\alpha\colon \Sigma^* \to \Delta^*$. On définit l'image de T par α, noté $\alpha(T)$, comme étant le système semi-Thue $(\Delta, \{(\alpha(s), \alpha(t)) : (s,t) \in R\})$.

Lemme 5.2.12. *Considérons $\Sigma, \widehat{\Sigma}$ deux alphabets, T un système semi-Thue sur Σ, $\widehat{T}$ un système semi-Thue sur $\widehat{\Sigma}$ et une application $\alpha\colon \Sigma^* \to \widehat{\Sigma}^*$ telle que pour tous mots $x, y \in \Sigma^*$, $x \underset{T}{\longrightarrow} y$ implique $\alpha(x) \underset{\widehat{T}}{\longrightarrow} \alpha(y)$. Alors, pour tous mots $u, v \in \Sigma^*$, $u \xrightarrow[T]{*} v$ implique $\alpha(u) \xrightarrow[\widehat{T}]{*} \alpha(v)$.*

Démonstration. Supposons avoir $u \xrightarrow[T]{*} v$. Alors il existe un entier $n \geq 1$ et $n+1$ mots $x_0, x_1, \ldots, x_n \in \Sigma^*$ qui satisfont l'équation (5.5). Ensuite on obtient de proche en proche que

$$\alpha(u) = \alpha(x_0) \underset{\widehat{T}}{\longrightarrow} \alpha(x_1) \underset{\widehat{T}}{\longrightarrow} \alpha(x_2) \underset{\widehat{T}}{\longrightarrow} \cdots \underset{\widehat{T}}{\longrightarrow} \alpha(x_n) = \alpha(v), \tag{5.6}$$

donc $\alpha(u) \xrightarrow[\widehat{T}]{*} \alpha(v)$. □

Lemme 5.2.13. *Considérons les notations de la Définition 5.2.11. Si α est un morphisme alors pour tous mots $u, v \in \Sigma^*$, $u \xrightarrow[T]{*} v$ implique $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$.*

Démonstration. On applique le Lemme 5.2.12 avec $\widehat{T} = \alpha(T)$. Soient $x, y \in \Sigma^*$ tels que $x \underset{T}{\longrightarrow} y$, c'est-à-dire tels qu'il existe $s, t, z, z' \in \Sigma^*$ tels que $x = zsz', y = ztz'$ et $(s, t) \in R$. Si α est un morphisme, alors on a $\alpha(x) = \alpha(zsz') = \alpha(z)\alpha(s)\alpha(z')$ et $\alpha(y) = \alpha(z)\alpha(t)\alpha(z')$. De plus, $(\alpha(s), \alpha(t))$ est une règle de $\alpha(T)$. Par conséquent, $\alpha(x) \underset{\alpha(T)}{\longrightarrow} \alpha(y)$. □

Définition 5.2.14. Soit (s, t) une règle d'un système semi-Thue, c'est-à-dire une paire de mots. On dit que (s, t) est une *règle d'insertion* si $s = \epsilon$ et une *règle de suppression* si $t = \epsilon$. Un système semi-Thue est dit *sans epsilon* s'il n'a ni règle d'insertion, ni règle de suppression.

Par extension, une instance (T, u, v) du problème Accessibility est dite *sans epsilon* si le système semi-Thue T est sans epsilon.

Définition 5.2.15. Soient Σ un alphabet et d une lettre. On définit λ_d et ρ_d comme étant des morphismes de Σ^* dans $(\Sigma \cup \{d\})^*$ tels que $\lambda_d(a) = da$ et $\rho_d(a) = ad$ pour toute lettre $a \in \Sigma$.

Par exemple, $\lambda_d(1101) = d1d1d0d1$ et $\rho_d(1101) = 1d1d0d1d$.

Lemme 5.2.16. *Pour tout nombre entier $k \geq 1$, le problème* Accessibility(k) *est décidable si et seulement si il est décidable sur des instances sans epsilon.*

Démonstration. Nous allons construire une réduction de problème Accessibility(k) général au problème Accessibility(k) sur des instances sans epsilon.

Soit (T, u, v) une instance de Accessibility(k). Notons Σ l'alphabet de T, d un symbole tel que $d \notin \Sigma$ et $\mu \colon \Sigma^* \to (\Sigma \cup \{d\})^* : w \mapsto \lambda_d(w)d = d\rho_d(w)$. Par définition de μ, on a que $(\mu(T), \mu(u), \mu(v))$ est une instance sans epsilon de Accessibility(k). De plus, $(\mu(T), \mu(u), \mu(v))$ est calculable à partir de (T, u, v). Il reste donc à démontrer que $u \underset{T}{\overset{*}{\longrightarrow}} v$ si et seulement si $\mu(u) \underset{\mu(T)}{\overset{*}{\longrightarrow}} \mu(v)$.

Soient $x, y \in \Sigma^*$ tels que $x \underset{T}{\longrightarrow} y$. Alors il existe $s, t, z, z' \in \Sigma^*$ tels que $x = zsz', y = ztz'$ et (s, t) est une règle de T. Par définition de μ, λ_d et ρ_d, on a

$$\mu(x) = \mu(zsz') = \lambda_d(zsz')d = \lambda_d(z)\lambda_d(s)\lambda_d(z')d = \lambda_d(z)\lambda_d(s)d\rho_d(z') = \lambda_d(z)\mu(s)\rho_d(z')$$

et

$$\mu(y) = \mu(ztz') = \lambda_d(z)\lambda_d(t)\lambda_d(z')d = \lambda_d(z)\mu(t)\rho_d(z').$$

De plus, $(\mu(s), \mu(t))$ est une règle de $\mu(T)$. Par conséquent, on a $\mu(x) \underset{\mu(T)}{\longrightarrow} \mu(y)$. Il suffit d'appliquer le Lemme 5.2.12, avec $\alpha = \mu$ et $\widehat{T} = \mu(T)$, et on obtient que $u \underset{T}{\overset{*}{\longrightarrow}} v$ implique $\mu(u) \underset{\mu(T)}{\overset{*}{\longrightarrow}} \mu(v)$.

Pour la condition suffisante, notons $\widehat{\mu} \colon (\Sigma \cup \{d\})^* \to \Sigma^*$ le morphisme défini par $\widehat{\mu}(a) = a$ pour toute lettre $a \in \Sigma$ et $\widehat{\mu}(d) = \epsilon$. On remarque que $\widehat{\mu}(\mu(w)) = w$ pour tout mot $w \in \Sigma^*$. Donc $T = \widehat{\mu}(\mu(T))$. Par conséquent, pour tous mots $\widehat{u}, \widehat{v} \in (\Sigma \cup \{d\})^*$ on a que $\widehat{u} \underset{\mu(T)}{\overset{*}{\longrightarrow}} \widehat{v}$

implique $\widehat{\mu}(\widehat{u}) \xrightarrow[T]{*} \widehat{\mu}(\widehat{v})$ en appliquant le Lemme 5.2.13 à $\widehat{\mu}(T)$ et $\mu(T)$. En particulier, $\mu(u) \xrightarrow[\mu(T)]{*} \mu(v)$ implique $u = \widehat{\mu}(\mu(u)) \xrightarrow[T]{*} \widehat{\mu}(\mu(v)) = v$. □

Définition 5.2.17. Soient Σ un alphabet, T un système semi-Thue sur Σ et un langage $L \subseteq \Sigma^*$. On dit que L est *fermé pour la dérivation* dans T si pour tout $x \in L$ et pour tout $y \in \Sigma^*$, $x \xrightarrow[T]{} y$ implique $y \in L$.

Le Lemme suivant est en quelque sorte la réciproque du Lemme 5.2.12.

Lemme 5.2.18. *Considérons $\Sigma, \widehat{\Sigma}$ deux alphabets, T un système semi-Thue sur Σ, $\widehat{T}$ un système semi-Thue sur $\widehat{\Sigma}$ et une application $\alpha\colon \Sigma^* \to \widehat{\Sigma}^*$ tels que*

(i) l'image de α est fermée pour la dérivation dans $\widehat{T}$, et

(ii) pour tous mots $x, y \in \Sigma^*$, $\alpha(x) \xrightarrow[\widehat{T}]{} \alpha(y)$ implique $x \xrightarrow[T]{} y$.

Alors pour tous mots $u, v \in \Sigma^*$, $\alpha(u) \xrightarrow[\widehat{T}]{*} \alpha(v)$ implique $u \xrightarrow[T]{*} v$.

Démonstration. Supposons que $\alpha(u) \xrightarrow[\widehat{T}]{*} \alpha(v)$. Alors il existe un nombre entier $n \geq 0$ et $n+1$ mots $\widehat{x}_0, \widehat{x}_1, \ldots, \widehat{x}_n$ sur $\widehat{\Sigma}$ tels que

$$\alpha(u) = \widehat{x}_0 \xrightarrow[\widehat{T}]{} \widehat{x}_1 \xrightarrow[\widehat{T}]{} \widehat{x}_2 \xrightarrow[\widehat{T}]{} \cdots \xrightarrow[\widehat{T}]{} \widehat{x}_n = \alpha(v).$$

Étant donné le point (i), on sait que $\widehat{x}_i$ appartient à l'image de α pour tout $i \in \{0, \ldots, n\}$. Posons $x_0 = u, x_n = v$ et pour tout $i \in \{1, \ldots, n-1\}$, posons $x_i \in \Sigma^*$ tel que $\widehat{x}_i = \alpha(x_i)$. Ainsi, l'équation (5.6) est satisfaite. De plus, on obtient l'équation (5.5) grâce au point (ii) de l'énoncé. Ainsi on obtient bien que $u \xrightarrow[T]{*} v$. □

Définition 5.2.19. Un code X est dit *sans virgule* si pour tous mots x, z et z',

$$(x \in X \text{ et } zxz' \in X^*) \implies (z \in X^* \text{ et } z' \in X^*).$$

Lemme 5.2.20. *Considérons les notations de la Définition 5.2.11 et supposons que*

(i) α est un morphisme injectif,

(ii) $\alpha(\Sigma)$ est un code sans virgule, et que

(iii) T n'a pas de règle d'insertion.

Alors, pour tous mots $u, v \in \Sigma^*$, on a $u \xrightarrow[T]{*} v$ si et seulement si $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$.

Démonstration. Étant donné le Lemme 5.2.13, on sait que $u \xrightarrow[T]{*} v$ implique $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$. Démontrons l'autre sens en utilisant le Lemme 5.2.18.

Soient $\widehat{x}, \widehat{y} \in \Delta^*$ tels que $\widehat{x}$ appartient à l'image de α et $\widehat{x} \xrightarrow[\alpha(T)]{} \widehat{y}$. Alors il existe $\widehat{z}, \widehat{z}' \in \Delta^*$ et $(s, t) \in R$ tels que $\widehat{x} = \widehat{z}\alpha(s)\widehat{z}'$ et $\widehat{y} = \widehat{z}\alpha(t)\widehat{z}'$. Comme α est un morphisme, l'image de

α est égale à $\alpha(\Sigma^*)$. En particulier, $\alpha(s), \widehat{z}\alpha(s)\widehat{z}' \in \alpha(\Sigma^*)$. Par ailleurs, $\alpha(s)$ appartient à $\alpha(\Sigma^*)$ car comme T n'a pas de règle d'insertion, le mot s est non vide. Donc son image par le morphisme injectif α est non vide aussi. Ainsi, $\widehat{z}$ et $\widehat{z}'$ appartiennent à $\alpha(\Sigma^*)$ car $\alpha(\Sigma)$ est un code sans virgule. Donc, il existe $z, z' \in \Sigma^*$ tels que $\alpha(z) = \widehat{z}$ et $\alpha(z') = \widehat{z}'$. Ainsi, on peut maintenant écrire $\widehat{x}$ et $\widehat{y}$ sous la forme

$$\widehat{x} = \alpha(zsz') \text{ et } \widehat{y} = \alpha(ztz').$$

Par conséquent, $\widehat{y}$ appartient à l'image de α, ce qui démontre que l'image de α est fermée pour la dérivation dans $\alpha(T)$. De plus, on obtient aussi que

$$\alpha^{-1}(\widehat{x}) = zsz' \underset{T}{\longrightarrow} ztz' = \alpha^{-1}(\widehat{y}).$$

Ainsi, on peut appliquer le Lemme 5.2.18 avec $\widehat{T} = \alpha(T)$ et on obtient que $u \xrightarrow[T]{*} v$. □

Regardons l'importance des hypothèses du Lemme 5.2.20. L'hypothèse (*iii*) peut facilement être remplacée par "T n'a pas de règle de suppression". Il suffirait d'appliquer le Lemme 5.2.20 à $T' = (\Sigma, \{(t, s) : (s, t) \in R\})$.
Par contre, les deux contre-exemples suivants nous montrent que les hypothèses (*ii*) et (*iii*) sont indispensables.

Exemple 5.2.21. Considérons $T = (\{a, b\}, \{(a, aa)\})$ et $\alpha\colon \{a, b\}^* \to \{0, 1\}^*$ le morphisme défini par $\alpha(a) = 01$ et $\alpha(b) = 011$. On a $\alpha(T) = (\{0, 1\}, \{(01, 0101)\})$. De plus, α est injectif et T est sans epsilon. Par contre, $\alpha(\{a, b\})$ n'est pas un code sans virgule, car si on considère les mots $x = 01, z = 01, z' = 1$, on a $x = \alpha(a) \in \alpha(\{a, b\})$, $zxz' = 01011 = \alpha(ab) \in \alpha(\{a, b\})^*$. De plus, $z \in \alpha(\{a, b\})^*$, mais $z' = 1 \notin \alpha(\{a, b\})^*$. Enfin, $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$ n'implique pas $u \xrightarrow[T]{*} v$ pour tous $u, v \in \{a, b\}^*$ car on a $\alpha(b) \underset{\alpha(T)}{\longrightarrow} \alpha(ab)$, mais pas $b \underset{T}{\longrightarrow} ab$.

Exemple 5.2.22. Considérons $T = (\{a, b, c\}, \{(\epsilon, a), (b, \epsilon)\})$ et $\alpha\colon \{a, b, c\}^* \to \{0, 1\}^*$ le morphisme défini par $\alpha(a) = 101$, $\alpha(b) = 1001$ et $\alpha(c) = 10001$. On a $\alpha(T) = (\{0, 1\}, \{(\epsilon, 101), (1001, \epsilon)\})$. On remarque que α est injectif et que $\alpha(\{a, b, c\})$ est un code sans virgule. Par contre, T admet clairement des règles d'insertion et de suppression. De plus, $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$ n'implique pas $u \xrightarrow[T]{*} v$ pour tous $u, v \in \{a, b, c\}^*$, car on a

$$\alpha(c) \underset{\alpha(T)}{\longrightarrow} 10101001 \underset{\alpha(T)}{\longrightarrow} 10101001011 \underset{\alpha(T)}{\longrightarrow} 1010011 \underset{\alpha(T)}{\longrightarrow} \alpha(a)$$

mais pas $c \xrightarrow[T]{*} a$.

Démonstration. (Proposition 5.2.10) Nous allons construire une réduction du problème Accessibility(k) sur des instances sans epsilon au problème Accessibility(k) sur $\mathcal{C}_k$ pour que l'on puisse appliquer le Lemme 5.2.16.

Considérons (T, u, v) une instance sans epsilon de ACCESSIBILITY(k) et Σ l'alphabet de T. On construit une application injective $\alpha\colon \Sigma \to C$ et on donne le même nom au morphisme de Σ^* dans $\{0,1\}$, qui étend α. Ainsi, l'ensemble $(\alpha(T), \alpha(u), \alpha(v))$ appartient à $\mathcal{C}_k$ et $(\alpha(T), \alpha(u), \alpha(v))$ est calculable à partir de (T, u, v). De plus, grâce au Lemme 5.2.20, on sait que $u \xrightarrow[T]{*} v$ est équivalent à $\alpha(u) \xrightarrow[\alpha(T)]{*} \alpha(v)$. □

5.2.2 Réduction du problème GPCP au problème Accessibility

Définition 5.2.23. Un mot f est dit *bordé* s'il existe trois mots non vides u, v et x tels que $f = xu = vx$.

Définition 5.2.24. On dit que deux mots x et y *se chevauchent* si au moins une des assertions suivantes est satisfaite :

(i) x apparait dans y,

(ii) y apparait dans x,

(iii) un préfixe non vide de x est un suffixe de y, ou

(iv) un préfixe non vide de y est un suffixe de x.

Si on convient que le mot vide apparait dans tout mot, alors le mot vide et tout autre mot se chevauchent toujours.

Lemme 5.2.25. *Soient $T = (\Sigma, R)$ un système semi-Thue et $f, u, v \in \Sigma^*$ vérifiant :*

(i) *f n'est pas bordé,*

(ii) *f n'apparait pas dans u,*

(iii) *f n'apparait pas dans v, et*

(iv) *pour chaque règle $(s,t) \in R$, s et f ne se chevauchent pas.*

Alors, on a $u \xrightarrow[T]{*} v$ si et seulement si il existe $x, y \in \Sigma^*$ satisfaisant $xfv = ufy$ et $x \xrightarrow[T]{*} y$.

Démonstration. La condition nécessaire est directe car si $u \xrightarrow[T]{*} v$, les mots $x = u$ et $y = v$ sont tels que $xfv = ufy$ et $x \xrightarrow[T]{*} y$.

Pour la condition suffisante, supposons qu'il existe $x, y \in \Sigma^*$ tels que $xfv = ufy$ et $x \xrightarrow[T]{*} y$ et notons n le nombre d'occurrence de f dans xfv. Comme f n'est pas bordé, ses occurrences ne se chevauchent pas deux à deux :

$$xfv = ufy = w_0 f w_1 f w_2 \cdots f w_n$$

pour des mots $w_0, w_1, \ldots, w_n \in \Sigma^*$. Comme f n'apparait pas dans v, on a $v = w_n$ et

$$x = w_0 f w_1 f w_2 \cdots f w_{n-1}.$$

De la même façon, comme f n'apparait pas dans u non plus, alors on a $u = w_0$ et

$$y = w_1 f w_2 \cdots f w_n.$$

Enfin, grâce à l'hypothèse (iv), on remarque que f joue le rôle de délimiteur mais en respectant la dérivation dans T. Donc $x \xrightarrow[T]{*} y$ implique $w_{i-1} \xrightarrow[T]{*} w_i$ pour tout $i \in \{1, \ldots, n\}$. Ainsi on a bien $u \xrightarrow[T]{*} v$. □

Pour le reste de cette sous-section, on notera f le mot 0011.

Lemme 5.2.26. *Soit $k \in \mathbb{N}_0$. Pour tout $(T, u, v) \in \mathcal{C}_k$, (T, u, v) est une instance positive de* Accessibility(k) *si et seulement si il existe $x, y \in \{0,1\}^*$ satisfaisant $xfv = ufy$ et $x \xrightarrow[T]{*} y$.*

Démonstration. Le mot f n'est pas bordé et pour tout $s \in C^+$, s et f ne se chevauchent pas. Donc on peut appliquer le Lemme 5.2.25. □

Définition 5.2.27. Une instance $(\Sigma, \sigma, \tau, s, t, s', t')$ de GPCP est appelée *instance de Claus* si $\sigma(\Sigma) \cup \tau(\Sigma) \subseteq \{0,1\}^+$.

On peut maintenant démontrer une version légèrement plus forte de la Proposition 5.2.6.

Théorème 5.2.28. *Soit $k \in \mathbb{N}_0$. Si* GPCP$(k+2)$ *est décidable sur des instances de Claus, alors* Accessibility(k) *est décidable.*

Démonstration. Nous allons construire une réduction de Accessibility(k) sur $\mathcal{C}_k$ à GPCP$(k+2)$ sur des instances de Claus dans le but d'appliquer la Proposition 5.2.10.

Soit (T, u, v) un élément de $\mathcal{C}_k$. Alors il existe $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k \in C^+$ tels que $T = (\{0,1\}, \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\})$. Considérons k symboles $a_1, a_2, \ldots, a_k$ tels que $\Sigma = \{0, 1, a_1, a_2, \ldots, a_k\}$ est un alphabet de cardinalité $k+2$. Soient $\sigma, \tau \colon \Sigma^* \to \{0,1\}^*$ les morphismes définis par :

$$\begin{cases} \sigma(0) = 0 \\ \sigma(1) = 1 \\ \sigma(a_i) = s_i \end{cases} \text{ et } \begin{cases} \tau(0) = 0 \\ \tau(1) = 1\,. \\ \tau(a_i) = t_i \end{cases}$$

pour $i \in \{1, \ldots, k\}$. Enfin, notons $J = (\Sigma, \sigma, \tau, \epsilon, fv, uf, \epsilon)$ l'instance de GPCP$(k+2)$.

On remarque que J est calculable à partir de (T, u, v) et que J est une instance de Claus. En effet, $\sigma(\Sigma) \cup \tau(\Sigma) = \{0, 1, s_1, \ldots, s_k\} \cup \{0, 1, t_1, \ldots, t_k\} \subseteq \{0,1\}^+$ car $s_1, s_2, \ldots, s_k$, $t_1, t_2, \ldots, t_k \in C^+ = \{010^n101 : n \geq 2\}^+$. Il reste donc à démontrer que (T, u, v) est une instance positive de Accessibility(k) si et seulement si J est une instance positive de GPCP$(k+2)$. La condition nécessaire repose sur le Lemme suivant :

Lemme 5.2.29. *Pour tous mots $x, y \in \{0,1\}^*$ tels que $x \xrightarrow[T]{} y$, il existe $z \in \Sigma^*$ tel que $x = \sigma(z)$ et $y = \tau(z)$.*

Démonstration. Soient $z', z'' \in \{0,1\}^*$ et $i \in \{1,\dots,k\}$ tels que $x = z's_iz''$ et $y = z't_iz''$. Alors il suffit de prendre $z = z'a_iz''$ et on a bien que $\sigma(z'a_iz'') = z'\sigma(a_i)z'' = z's_iz'' = x$ et $\tau(z'a_iz'') = z't_iz'' = y$. □

Si on suppose que (T,u,v) est une instance positive de Accessibility(k), alors on a $u \xrightarrow[T]{*} v$. Donc il existe un nombre entier $n \geq 0$ et $n+1$ mots $x_0, x_1, \dots, x_n \in \{0,1\}^*$ qui satisfont l'équation (5.5). Grâce au Lemme 5.2.29, on sait qu'il existe $z_i \in \Sigma^*$ tel que $x_{i-1} = \sigma(z_i)$ et $x_i = \tau(z_i)$ pour tout $i \in \{1,\dots,n\}$. Ainsi, le mot $w = z_1fz_2fz_3\cdots fz_n$ satisfait

$$\begin{aligned}\sigma(w)fv &= \sigma(z_1)f\sigma(z_2)f\sigma(z_3)\cdots f\sigma(z_n)fv\\ &= x_0fx_1fx_2\cdots fx_{n-1}fx_n\\ &= ufx(z_1)fx(z_2)\cdots fx(z_n)\\ &= ufx(w).\end{aligned}$$

Donc J est une instance positive de GPCP$(k+2)$.

Pour la condition suffisante, supposons qu'il existe un mot $w \in \Sigma^*$ tel que $\epsilon\sigma(w)fv = ufx(w)\epsilon$. Par définition des morphismes σ et τ, on a $\sigma(z) \xrightarrow[T]{*} \tau(z)$ pour tout $z \in \Sigma^*$. En particulier, les mots $x = \sigma(w)$ et $y = \tau(w)$ satisfont $xfv = ufy$ et $x \xrightarrow[T]{*} y$ pour tout $z \in \Sigma^*$. On peut donc appliquer le Lemme 5.2.26, et on obtient que (T,u,v) est une instance positive de Accessibility(k). □

5.2.3 Réduction de PCP à GPCP

Définition 5.2.30. Une instance $(\Sigma,\sigma,\tau,s,t,s',t')$ de GPCP est dite (ϵ,ϵ)-*free* si pour toute lettre $a \in \Sigma$ on a $(\sigma(a),\tau(a)) \neq (\epsilon,\epsilon)$.

Lemme 5.2.31. *Pour tout nombre entier $k \geq 1$, GPCP(k) est décidable si et seulement si le problème est décidable sur des instances (ϵ,ϵ)-free.*

Démonstration. Nous allons construire une réduction de GPCP(k) à GPCP(k) sur des instances (ϵ,ϵ)-free.

Soit $I = (\Sigma,\sigma,\tau,s,t,s',t')$ une instance de GPCP(k). On calcule l'ensemble $\widehat{\Sigma}$ des lettres $a \in \Sigma$ telles que $(\sigma(a),\tau(a)) \neq (\epsilon,\epsilon)$. On peut supposer sans perte de généralité, que $\widehat{\Sigma} \neq \emptyset$ car si $\widehat{\Sigma}$ était vide, alors résoudre le problème GPCP(k) sur I reviendrait à déterminer si $st = s't'$. Notons $\widehat{\sigma}$ et $\widehat{\tau}$ les restrictions à $\widehat{\Sigma}^*$ de σ et τ respectivement. Enfin, notons J le septuple $(\widehat{\Sigma},\widehat{\sigma},\widehat{\tau},s,t,s',t')$. On remarque que J est calculable à partir de I et que J est une instance (ϵ,ϵ)-free. De plus, I est une instance positive de GPCP(k) si et seulement si J est une instance positive de GPCP(k). □

Remarquons que toute instance de Claus de GPCP est une instance (ϵ,ϵ)-free, mais que l'inverse n'est pas vrai en général.

De manière similaire, on dira qu'une instance (Σ,σ,τ) de PCP est une instance *de Claus* si $\sigma(\Sigma) \cup \tau(\Sigma) \subseteq \{0,1\}^+$.

On peut maintenant démontrer la Proposition 5.2.7.

Théorème 5.2.32. *Soit $k \in \mathbb{N}_0$.*

(i) Si PCP$(k+2)$ *est décidable alors* GPCP(k) *l'est aussi.*

(ii) Si PCP$(k+2)$ *est décidable sur des instances de Claus, alors* GPCP(k) *aussi.*

Démonstration. Nous allons construire une réduction de GPCP(k) sur des instances (ϵ, ϵ)-free au problème PCP$(k+2)$, dans le but d'appliquer le Lemme 5.2.31.

Soit $I = (\Sigma, \sigma, \tau, s, t, s', t')$ une instance (ϵ, ϵ)-free de GPCP(k). Sans perte de généralité, on peut supposer que $b, e \notin \Sigma$ et poser $\widehat{\Sigma} = \Sigma \cup \{b, e\}$ un alphabet de cardinalité $k+2$. Notons $\lambda = \lambda_d$ et $\rho = \rho_d$ introduits à la Définition 5.2.15 et $\widehat{\rho}, \widehat{\tau} \colon \widehat{\Sigma}^* \to \{0, 1, d, b, e\}^*$ les deux morphismes définis par :

$$\begin{cases} \widehat{\sigma}(b) = b\lambda(s) \\ \widehat{\sigma}(e) = \lambda(t)de \\ \widehat{\sigma}(a) = \lambda(\sigma(a)) \end{cases} \text{ et } \begin{cases} \widehat{\tau}(b) = bd\rho(s') \\ \widehat{\tau}(e) = \rho(t')e \\ \widehat{\tau}(a) = \rho(\tau(a)) \end{cases}$$

pour toute lettre $a \in \Sigma$. Enfin, notons $j \colon \{0, 1, d, b, e\}^* \to \{0, 1\}^*$ un morphisme injectif. Par exemple, j peut être défini par $j(0) = 000$, $j(1) = 111$, $j(d) = 101$, $j(b) = 100$ et $j(e) = 001$.

L'ensemble $J = (\widehat{\Sigma}, j \circ \widehat{\sigma}, j \circ \widehat{\tau})$ est une instance de PCP$(k+2)$ calculable à partir de I et J est une instance de Claus lorsque I en est une. Par conséquent, pour démontrer les points (i) et (ii) de l'énoncé, il faut vérifier que I est une instance positive de GPCP(k) si et seulement si J est une instance positive de PCP$(k+2)$.

Pour la condition nécessaire, nous allons nous aider du Lemme suivant :

Lemme 5.2.33. *Pour tout mot $x \in \Sigma^*$, on a $s\sigma(w)t = s'\sigma(w)t'$ si et seulement si $\widehat{\sigma}(bwe) = \widehat{\tau}(bwe)$.*

Démonstration. Soit $w \in \Sigma^*$, on a

$$\widehat{\sigma}(bwe) = \widehat{\sigma}(b)\widehat{\sigma}(w)\widehat{\sigma}(e) = b\lambda(s)\lambda(\sigma(w))\lambda(t)de = b\lambda(s\sigma(w)t)de$$

et

$$\widehat{\tau}(bwe) = \widehat{\tau}(b)\widehat{\tau}(w)\widehat{\tau}(e) = bd\rho(s')\rho(\tau(w))\rho(t')e = bd\rho(s'\tau(w)t')e.$$

Par définition de λ et ρ, on a $\lambda(x)d = d\rho(x)$ pour tout mot $x \in \{0, 1\}^*$. Donc $s\sigma(w)t = s'\sigma(w)t'$ implique $\widehat{\sigma}(bwe) = \widehat{\tau}(bwe)$. De plus, $\widehat{\sigma}(bwe) = \widehat{\tau}(bwe)$ implique $\lambda(s\sigma(w)t) = \lambda(s'\sigma(w)t')$. Comme λ est injectif, on obtient que $\widehat{\sigma}(bwe) = \widehat{\tau}(bwe)$ implique $s\sigma(w)t = s'\sigma(w)t'$. □

Ainsi, si I est une instance positive de GPCP(k), alors il existe un mot $w \in \Sigma^*$ tel que $s\sigma(w)t = s'\tau(w)t'$. Grâce au Lemme 5.2.33, on sait que le mot bwe appartenant à $\widehat{\Sigma}^*$ est tel que $j \circ \widehat{\sigma}(bwe) = j \circ \widehat{\tau}(bwe)$. Donc J est une instance positive de PCP$(k+2)$.

La condition suffisante est un peu plus difficile à démontrer. On va considérer les deux lemmes suivants.

Lemme 5.2.34. *Pour tout mot $w \in \widehat{\Sigma}^*$, les trois assertions suivantes sont équivalentes :*

1. $\widehat{\sigma}(we)$ *est un préfixe de* $\widehat{\tau}(we)$,
2. $\widehat{\tau}(we)$ *est un préfixe de* $\widehat{\sigma}(we)$, *et*
3. $\widehat{\tau}(we) = \widehat{\sigma}(we)$.

Démonstration. La lettre e apparait une fois dans $\widehat{\sigma}(e)$ et $\widehat{\tau}(e)$ alors que pour toute lettre $a \in \Sigma \cup \{b\}$, e n'apparait pas dans $\widehat{\sigma}(a)$ et $\widehat{\tau}(a)$. Par conséquent, on a $|\widehat{\sigma}(x)|_e = |x|_e = |\widehat{\tau}(x)|_e$ pour tout mot $x \in \widehat{\Sigma}^*$. Comme e est la dernière lettre de $\widehat{\sigma}(e)$, tout préfixe propre de $\widehat{\sigma}(we)$ contient moins de e que $\widehat{\tau}(we)$. Ainsi, $\widehat{\tau}(we)$ ne peut pas être un préfixe propre de $\widehat{\sigma}(we)$. De la même façon, $\widehat{\sigma}(we)$ ne peut pas être préfixe propre de $\widehat{\tau}(we)$. □

Lemme 5.2.35. *Pour tout mot* $w \in \widehat{\Sigma}^*$, *les trois assertions suivantes sont équivalentes :*

1. $\widehat{\sigma}(bw)$ *est un suffixe de* $\widehat{\tau}(bw)$,
2. $\widehat{\tau}(bw)$ *est un suffixe de* $\widehat{\sigma}(bw)$, *et*
3. $\widehat{\tau}(bw) = \widehat{\sigma}(bw)$.

Démonstration. La démonstration est strictement la même que la démonstration du Lemme 5.2.34, avec cette fois-ci la lettre b qui apparait une unique fois dans $\widehat{\sigma}(b)$ et $\widehat{\tau}(b)$. □

Soit $a \in \widehat{\Sigma}$ tel que $\widehat{\sigma}(a) \neq \epsilon$. Remarquons que la première lettre de $\widehat{\sigma}(a)$ est égale à d si $a \in \Sigma \cup \{e\}$ ou bien à b si $a = b$. De plus, la dernière lettre de $\widehat{\sigma}(a)$ est différente de d. De même, si maintenant $a \in \widehat{\Sigma}$ est tel que $\widehat{\tau}(a) \neq \epsilon$, alors la première lettre de $\widehat{\tau}(a)$ est différente de d et la dernière lettre est égale à e si $a = e$ et à d si $a \in \Sigma \cup \{b\}$.

Supposons maintenant que J est une instance positive de $\mathrm{PCP}(k+2)$, alors il existe un mot $w \in \widehat{\Sigma}^+$ tel que $\widehat{\sigma}(w) = \widehat{\tau}(w)$. Notons x le mot $\widehat{\sigma}(w)$.

Comme I est une instance (ϵ, ϵ)-free de GPCP, alors $(\widehat{\sigma}(a), \widehat{\tau}(a)) \neq (\epsilon, \epsilon)$ pour toute lettre $a \in \widehat{\Sigma}$, donc x est un mot non vide. Vu ce qu'on a dit précédemment, on sait que la première lettre de x est égale à b. Par définition de $\widehat{\sigma}$ et $\widehat{\tau}$, b est aussi la première lettre de w. De la même façon, on obtient que e est la dernière lettre de x et donc aussi de w. Par conséquent on peut réécrire le mot w sous la forme $bw'e$ avec $w' \in \widehat{\Sigma}^*$.

Supposons que w est le plus petit mot non vide sur $\widehat{\Sigma}$ tel que $\widehat{\sigma}(w) = \widehat{\tau}(w)$. Démontrons que $w' \in \Sigma^*$. Pour cela il faut montrer que les lettres e et b n'apparaissent pas dans w' : Si on suppose que e apparait dans w', alors il existe des mots $w_1, w_2 \in \widehat{\Sigma}^*$ tels que $w' = w_1 e w_2$. Ensuite on a

$$x = \widehat{\sigma}(w) = \widehat{\sigma}(bw'e) = \widehat{\sigma}(bw_1ew_2e) = \widehat{\sigma}(bw_1e)\widehat{\sigma}(w_2e)$$

et

$$x = \widehat{\tau}(w) = \widehat{\tau}(bw'e) = \widehat{\tau}(bw_1ew_2e) = \widehat{\tau}(bw_1e)\widehat{\tau}(w_2e).$$

Donc soit $\widehat{\sigma}(bw_1e)$ est un préfixe de $\widehat{\tau}(bw_1e)$, soit $\widehat{\tau}(bw_1e)$ est un préfixe de $\widehat{\sigma}(bw_1e)$. Par le Lemme 5.2.34 on en déduit que $\widehat{\sigma}(bw_1e) = \widehat{\tau}(bw_1e)$. On obtient donc une contradiction car le mot bw_1e est plus court que le mot w. Donc la lettre e n'apparait pas dans w'. De la même manière on montre, en utilisant le Lemme 5.2.35, que la lettre b n'apparait pas dans w'.

Ainsi, w' est un mot sur Σ, et donc d'après le Lemme 5.2.33, on a $s\sigma(w')t = s'\tau(w')t'$. Donc I est bien une instance positive de $\mathrm{GPCP}(k)$. □

En combinant le Théorème 5.2.28 et le Théorème 5.2.32(ii), on obtient une version un peu plus forte de la Proposition 5.2.8.

Corollaire 5.2.36. *Soit $k \in \mathbb{N}_0$. Si* PCP($k+4$) *est décidable sur des instances de Claus, alors* Accessibility(k) *est décidable.*

Comme Matiyasevich et Sénizergues [14] ont démontré que le problème Accessibility(3) est indécidable, alors grâce à la Proposition 5.2.8 on sait que le problème PCP(7) est indécidable. De plus, grâce à la Proposition 5.2.6, on sait que le problème GPCP(5) est indécidable.

Pour terminer, la décidabilité des huit problèmes suivants est encore ouverte : Accessibility(1), Accessibility(2), GPCP(3), GPCP(4), PCP(3), PCP(4), PCP(5) et PCP(6). La décidabilité des problèmes PCP(2) et GPCP(2) a été démontrée par Ehrenfeucht et Rozenberg [6].

Chapitre 6

Le cas des matrices de plus grandes dimensions

Dans ce chapitre, nous allons nous concentrer sur les semi-groupes de matrices carrées de dimension plus grande que 2. Tout d'abord, en considérant le cas des matrices 3×3, nous allons tenter de démontrer l'indécidabilité des problèmes $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ et $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$. Pour cela nous introduirons un nouveau problème de décision dont nous caractériserons la décidabilité sur les instances de Claus.

Ensuite, on reparlera du problème de décision $\textsc{Mortal}[\mathbb{Q}^{n\times n}]$ pour $n = 3$, et on démontrera son indécidabilité en se ramenant au problème de correspondance de Post.

Enfin, on considérera le cas des matrices de dimension plus grande que 3 et le but sera de démontrer que les problèmes $\textsc{Free}(7+h)[\mathbb{N}^{6\times 6}]$, $\textsc{Free}(5+h)[\mathbb{N}^{9\times 9}]$, $\textsc{Free}(4+h)[\mathbb{N}^{12\times 12}]$, $\textsc{Free}(3+h)[\mathbb{N}^{18\times 18}]$ et $\textsc{Free}(2+h)[\mathbb{N}^{36\times 36}]$ sont indécidables pour tout $h \in \mathbb{N}$. Afin d'obtenir ces résultats, nous prouverons d'abord que $\textsc{Free}(kd+1)[D]$ est décidable si $\textsc{Free}(k+1)[D^{d\times d}]$ est décidable pour tous $k, d \in \mathbb{N}_0$ et D un semi-anneau récursif. Nous terminerons par un tableau récapitulatif de la décidabilité du problème $\textsc{Free}(k)[\mathbb{N}^{d\times d}]$ pour $k, d \in \mathbb{N}_0$.

6.1 Les matrices carrées de dimension 3

L'objectif de cette section est de prouver que pour tout nombre entier $k \geq 13$, les deux problèmes $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ et $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$ sont indécidables.

Proposition 6.1.1. *Soit k_0 un nombre naturel. Si le problème $\textsc{Free}(k_0)[\mathbb{W} \times \mathbb{W}]$ est décidable, alors pour tout entier $k \in \{1, \ldots, k_0\}$, le problème $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ l'est aussi.*

Démonstration. Considérons un code $\{u, v, w\}$ à trois éléments avec $u, v, w \in \mathbb{W}$. Par exemple, on pourrait prendre $u = 1$, $v = 01$ et $w = 001$. Considérons le morphisme $\sigma : \mathbb{W} \to \mathbb{W}$ défini par

$$\begin{cases} \sigma(0) = u \\ \sigma(1) = v. \end{cases}$$

Ainsi, pour tout ensemble $X = \{(x_1, x'_1), \ldots, (x_k, x'_k)\} \subseteq \mathbb{W} \times \mathbb{W}$ à k éléments, l'ensemble

$$Y = \{(\sigma(x_1), \sigma(x'_1)), \ldots, (\sigma(x_k), \sigma(x'_k)), (w, w)\} \subseteq \mathbb{W} \times \mathbb{W}$$

à $(k+1)$ éléments est tel que X est un code si et seulement si Y est un code. En effet, supposons dans un premier temps que X est code, et considérons deux décompositions en éléments de Y telles que

$$\begin{aligned}(w,w)^{k_0}\left(\sigma(x_{\mu_1}),\sigma(x'_{\mu_1})\right)(w,w)^{k_1}\left(\sigma(x_{\mu_2}),\sigma(x'_{\mu_2})\right)\cdots\left(\sigma(x_{\mu_n}),\sigma(x'_{\mu_n})\right)(w,w)^{k_n} \qquad (6.1)\\ = (w,w)^{l_0}\left(\sigma(x_{\nu_1}),\sigma(x'_{\nu_1})\right)(w,w)^{l_1}\left(\sigma(x_{\nu_2}),\sigma(x'_{\nu_2})\right)\cdots\left(\sigma(x_{\nu_m}),\sigma(x'_{\nu_m})\right)(w,w)^{l_m},\end{aligned}$$

où $\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_m \in \{1, \ldots, k\}$ et $k_0, \ldots, k_n, l_0, \ldots, l_m \geq 0$, et montrons que $n = m$, $\mu_i = \nu_i$ et $k_j = l_j$ pour tous $i \in \{1, \ldots, n\}$ et $j \in \{0, \ldots, n\}$.
L'égalité (6.1) se réécrit

$$\begin{aligned}&\left(w^{k_0}\sigma(x_{\mu_1})w^{k_1}\sigma(x_{\mu_2})\cdots\sigma(x_{\mu_n})w^{k_n}, w^{k_0}\sigma(x'_{\mu_1})w^{k_1}\sigma(x'_{\mu_2})\cdots\sigma(x'_{\mu_n})w^{k_n}\right)\\ &= \left(w^{l_0}\sigma(x_{\nu_1})w^{l_1}\sigma(x_{\nu_2})\cdots\sigma(x_{\nu_m})w^{l_m}, w^{l_0}\sigma(x'_{\nu_1})w^{l_1}\sigma(x'_{\nu_2})\cdots\sigma(x'_{\nu_m})w^{l_m}\right),\end{aligned}$$

ou bien encore

$$\begin{cases} w^{k_0}\sigma(x_{\mu_1})w^{k_1}\sigma(x_{\mu_2})\cdots\sigma(x_{\mu_n})w^{k_n} = w^{l_0}\sigma(x_{\nu_1})w^{l_1}\sigma(x_{\nu_2})\cdots\sigma(x_{\nu_m})w^{l_m} \\ w^{k_0}\sigma(x'_{\mu_1})w^{k_1}\sigma(x'_{\mu_2})\cdots\sigma(x'_{\mu_n})w^{k_n} = w^{l_0}\sigma(x'_{\nu_1})w^{l_1}\sigma(x'_{\nu_2})\cdots\sigma(x'_{\nu_m})w^{l_m}. \end{cases}$$

Ces deux égalités peuvent se décomposer en trois équations, comme suit

$$\begin{cases} w^{k_0}\cdots w^{k_n} = w^{l_0}\cdots w^{l_m} \\ \sigma(x_{\mu_1})\sigma(x_{\mu_2})\cdots\sigma(x_{\mu_n}) = \sigma(x_{\nu_1})\sigma(x_{\nu_2})\cdots\sigma(x_{\nu_m}) \\ \sigma(x'_{\mu_1})\sigma(x'_{\mu_2})\cdots\sigma(x'_{\mu_n}) = \sigma(x'_{\nu_1})\sigma(x'_{\nu_2})\cdots\sigma(x'_{\nu_m}). \end{cases} \qquad (6.2)$$

Occupons-nous d'abord des deux dernières équations, comme σ est un morphisme injectif sur $\{u, v, w\}^+$ alors on obtient les équations suivantes

$$\begin{cases} x_{\mu_1}x_{\mu_2}\cdots x_{\mu_n} = x_{\nu_1}x_{\nu_2}\cdots x_{\nu_m} \\ x'_{\mu_1}x'_{\mu_2}\cdots x'_{\mu_n} = x'_{\nu_1}x'_{\nu_2}\cdots x'_{\nu_m}. \end{cases}$$

qui peuvent se réécrire en une unique équation

$$(x_{\mu_1}, x'_{\mu_1})\cdots(x_{\mu_n}, x'_{\mu_n}) = (x_{\nu_1}, x'_{\nu_1})\cdots(x_{\nu_m}, x'_{\nu_m}).$$

Comme X est un code, alors on obtient que $m = n$ et $\mu_i = \nu_i$ pour tout $i \in \{1, \ldots, n\}$. Ainsi grâce à la première équation de (6.2), on a de plus, que $k_j = l_j$ pour tout $j \in \{0, \ldots n\}$. Donc Y est bien un code.

Réciproquement, supposons maintenant que Y est un code et considérons deux décompositions en éléments de X telles que

$$(x_{\mu_1}, x'_{\mu_1}) \cdots (x_{\mu_n}, x'_{\mu_n}) = (x_{\nu_1}, x'_{\nu_1}) \cdots (x_{\nu_m}, x'_{\nu_m}) \tag{6.3}$$

où $\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_m \in \{1, \ldots, k\}$, et montrons que $n = m$ et $\mu_i = \nu_i$ pour tout $i \in \{1, \ldots, n\}$.
L'égalité (6.3) se réécrit

$$\begin{cases} x_{\mu_1} x_{\mu_2} \cdots x_{\mu_n} = x_{\nu_1} x_{\nu_2} \cdots x_{\nu_m} \\ x'_{\mu_1} x'_{\mu_2} \cdots x'_{\mu_n} = x'_{\nu_1} x'_{\nu_2} \cdots x'_{\nu_m}. \end{cases}$$

Ensuite, comme σ est un morphisme, on a

$$\begin{cases} \sigma(x_{\mu_1})\sigma(x_{\mu_2}) \cdots \sigma(x_{\mu_n}) = \sigma(x_{\nu_1})\sigma(x_{\nu_2}) \cdots \sigma(x_{\nu_m}) \\ \sigma(x'_{\mu_1})\sigma(x'_{\mu_2}) \cdots \sigma(x'_{\mu_n}) = \sigma(x'_{\nu_1})\sigma(x'_{\nu_2}) \cdots \sigma(x'_{\nu_m}), \end{cases}$$

ou encore

$$\begin{aligned} &\left(\sigma(x_{\mu_1}), \sigma(x'_{\mu_1})\right) \left(\sigma(x_{\mu_2}), \sigma(x'_{\mu_2})\right) \cdots \left(\sigma(x_{\mu_n}), \sigma(x'_{\mu_n})\right) \\ &= \left(\sigma(x_{\nu_1}), \sigma(x'_{\nu_1})\right) \left(\sigma(x_{\nu_2}), \sigma(x'_{\nu_2})\right) \cdots \left(\sigma(x_{\nu_m}), \sigma(x'_{\nu_m})\right). \end{aligned}$$

Comme Y est un code, alors $m = n$ et $\mu_i = \nu_i$ pour tout $i \in \{1, \ldots, n\}$. Donc X est bien un code.

On a donc montré qu'il existait une réduction du problème $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ au problème $\textsc{Free}(k+1)[\mathbb{W} \times \mathbb{W}]$. □

6.1.1 Modification du problème de correspondance de Post

Nous allons introduire un nouveau problème de décision, qui est un mélange des problèmes $\textsc{Free}[\mathbb{W} \times \mathbb{W}]$ et PCP.

Définition 6.1.2. On note MMPCP (Mixed Modification of the PCP) le problème suivant : étant donnée une instance (Σ, σ, τ) de PCP, déterminer s'il existe un entier $n \geq 1$, n symboles $a_1, \ldots, a_n \in \Sigma$ et $2n$ morphismes $\sigma_1, \ldots, \sigma_n, \tau_1, \ldots, \tau_n \in \{\sigma, \tau\}$ tels que

$$\sigma_1(a_1)\sigma_2(a_2) \cdots \sigma_n(a_n) = \tau_1(a_1)\tau_2(a_2) \cdots \tau_n(a_n)$$

et

$$(\sigma_1, \ldots, \sigma_n) \neq (\tau_1, \ldots, \tau_n).$$

Pour tout nombre entier $k \geq 1$, le problème MMPCP(k) est une réduction de MMPCP aux instances (Σ, σ, τ) où Σ contient k éléments.

Nous présentons maintenant la propriété fondamentale du problème MMPCP.

Proposition 6.1.3. *Soit* (Σ, σ, τ) *une instance de* MMPCP *telle que* $\sigma(a) \neq \tau(a)$ *pour tout* $a \in \Sigma$. *Alors* (Σ, σ, τ) *est une instance positive de* MMPCP *si et seulement si l'ensemble* $X = \{(\sigma(a), a) : a \in \Sigma\} \cup \{(\tau(a), a) : a \in \Sigma\}$ *n'est pas un code pour la concaténation composante à composante.*

Démonstration. Pour la condition nécessaire, on sait que si (Σ, σ, τ) est une instance positive de MMPCP, alors il existe n symboles $a_1, \ldots, a_n \in \Sigma$ et $2n$ morphismes $\sigma_1, \ldots, \sigma_n$, $\tau_1, \ldots, \tau_n \in \{\sigma, \tau\}$ tels que

$$\sigma_1(a_1)\sigma_2(a_2)\cdots\sigma_n(a_n) = \tau_1(a_1)\tau_2(a_2)\cdots\tau_n(a_n) \tag{6.4}$$

et $(\sigma_1, \ldots, \sigma_n) \neq (\tau_1, \ldots, \tau_n)$. Donc il suffit de considérer l'égalité

$$(\sigma_1(a_1), a_1)(\sigma_2(a_2), a_2)\cdots(\sigma_n(a_n), a_n) = (\tau_1(a_1), a_1)(\tau_2(a_2), a_2)\cdots(\tau_n(a_n), a_n). \tag{6.5}$$

Si on a cette égalité, alors l'ensemble X n'est pas un code pour la concaténation composante à composante car par hypothèse on a $\sigma(a) \neq \tau(a)$ pour tout $a \in \Sigma$ donc par exemple $(\sigma(a_1), a_1) \neq (\tau(a_1), a_1)$. Réciproquement, si X n'est pas un code pour la concaténation, alors il existe un $n > 0$ tel qu'on a l'égalité (6.5) avec le même indice n de chaque coté de l'égalité et avec $\sigma_1 = \cdots = \sigma_n = \sigma$ et $\tau_1 = \ldots \tau_n = \tau$. Ainsi, pour ce n on obtient l'équation (6.4) avec $(\sigma_1, \ldots, \sigma_n) \neq (\tau_1, \ldots, \tau_n)$. Donc (Σ, σ, τ) est une instance positive de MMPCP. □

Il est clair que toute instance positive de PCP est aussi une instance positive de MMPCP. La proposition suivante nous montre que l'inverse est vraie pour des instances de Claus.

Proposition 6.1.4. *Soient* (Σ, σ, τ) *une instance de Claus de* PCP *et les lettres* $e, b \in \Sigma$, (Σ, σ, τ) *est une instance positive de* MMPCP *si et seulement si il existe* $w \in (\Sigma \setminus \{b, e\})^*$ *tel que* $\sigma(bwe) = \tau(bwe)$.

Démonstration. La condition suffisante est directe étant donnée la Définition 6.1.2.

La condition nécessaire est plus longue à obtenir, mais se démontre de la même façon que la démonstration du Théorème 5.2.32. Pour tout nombre entier $n \geq 1$, on définit $\mathcal{D}_n$ l'ensemble $\{(a_i, \sigma_i, \tau_i) \in \Sigma \times \{\sigma, \tau\} \times \{\sigma, \tau\}$ pour tout $i \in \{1, \ldots, n\}\}$ des triplets satisfaisant l'équation (6.4), ainsi que $\mathcal{D}'_n$ l'ensemble des $(a_i, \sigma_i, \tau_i)_{i\in\{1,\ldots,n\}} \in \mathcal{D}_n$ satisfaisant l'équation (6.5). On remarque qu'il existe un nombre entier $n \geq 1$ tel que $\mathcal{D}'_n \neq \emptyset$ si et seulement si (Σ, σ, τ) est une instance positive de MMPCP.

De plus, pour tout $(a_i, \sigma_i, \tau_i)_{i\in\{1,\ldots,n\}} \in \mathcal{D}'_n$, si $\sigma_1 = \tau_1$ alors $(a_i, \sigma_i, \tau_i)_{i\in\{2,\ldots,n\}} \in \mathcal{D}'_{n-1}$. De même, si $\sigma_n = \tau_n$, alors $(a_i, \sigma_i, \tau_i)_{i\in\{1,\ldots,n-1\}} \in \mathcal{D}'_{n-1}$.

Lemme 6.1.5. *Considérons* $(a_i, \sigma_i, \tau_i)_{i\in\{1,\ldots,n\}} \in \mathcal{D}'_n$ *et* $k \in \{1, \ldots, n-1\}$. *Si* $a_k = e$ *ou* $a_{k+1} = b$, *alors* $(a_i, \sigma_i, \tau_i)_{i\in\{1,\ldots,k\}}$ *appartient à* $\mathcal{D}'_k$ *ou* $(a_i, \sigma_i, \tau_i)_{i\in\{k+1,\ldots,n\}}$ *appartient à* $\mathcal{D}'_{n-k}$.

Démonstration. Supposons que $a_k = e$. Posons $s = \sigma_1(a_1)\sigma_2(a_2)\cdots\sigma_k(a_k)$ et $t = \tau_1(a_1)\tau_2(a_2)\cdots\tau_k(a_k)$. On a $|\sigma(e)|_e = |\tau(e)|_e = 1$ et $|\sigma(a)|_e = |\tau(a)|_e = 0$ pour toute lettre $a \in \Sigma \setminus \{e\}$. Ainsi, on obtient que

$$|s|_e = |a_1a_2\cdots a_k|_e = |t|_e.$$

Comme $a_k = e$, on sait que s et t se terminent avec la lettre e. De plus, si t était un préfixe propre de s alors on aurait la contradiction $|t|_e < |s|_e$. De la même manière, s ne peut pas être préfixe propre de t. Par conséquent, $s = t$. Ainsi, $(a_i, \sigma_i, \tau_i)_{i\in\{1,\dots,k\}} \in \mathcal{D}_k$ et $(a_i, \sigma_i, \tau_i)_{i\in\{k+1,\dots,n\}} \in \mathcal{D}_{n-k}$, et au moins un des deux satisfait l'équation (6.5).

Le cas où $a_{k+1} = b$ se traite de la même façon. En effet, posons $u = \sigma_{k+1}(a_{k+1})\cdots\sigma_n(a_n)$ et $v = \tau_{k+1}(a_{k+1})\cdots\tau_n(a_n)$. On a $|\sigma(b)|_b = |\tau(b)|_b = 1$ et $|\sigma(a)|_b = |\tau(a)|_b = 0$ pour toute lettre $a \in \Sigma \setminus \{b\}$. Ainsi, on obtient

$$|u|_b = |a_{k+1}\cdots a_n|_b = |v|_b.$$

Comme $a_k + 1 = b$, on sait que u et v commencent avec la lettre b. De plus, v ne peut pas être préfixe propre de u et inversement, donc $u = v$. Ainsi, $(a_i, \sigma_i, \tau_i)_{i\in\{1,\dots,k\}} \in \mathcal{D}_k$ et $(a_i, \sigma_i, \tau_i)_{i\in\{k+1,\dots,n\}} \in \mathcal{D}_{n-k}$, et au moins un des deux satisfait l'équation (6.5) □

Considérons n le plus petit nombre naturel strictement positif tel que $\mathcal{D}'_n \neq \emptyset$. Soit $(a_i, \sigma_i, \tau_i)_{i\in\{1,\dots,n\}}$ un élément de $\mathcal{D}'_n$. Vu ce qu'on a fait précédemment, on sait que $\sigma_1 \neq \tau_1$, et comme $\sigma_1(a_1)$ et $\tau_1(a_1)$ commencent par la même lettre, on a $a_1 = b$. De la même manière, $\sigma_n \neq \tau_n$ et comme $\sigma_n(a_n)$ et $\tau_n(a_n)$ finissent par la même lettre, alors $a_n = e$. De plus, le Lemme 6.1.5 nous assure que $w = a_2a_3\cdots a_{n-1}$ appartient à $(\Sigma \setminus \{b, e\})^*$.

Sans perte de généralité, on peut supposer que $\sigma_1 = \sigma$ et $\tau_1 = \tau$. Pour terminer la démonstration il faut prouver que pour tout $i \in \{2, \dots, n\}$, $\sigma_i = \sigma$ et $\tau_i = \tau$, ainsi (Σ, σ, τ) sera une instance positive de MMPCP. Procédons par induction. Soient $i, j \in \{1, \dots, n\}$ tels que $\sigma = \sigma_1 = \sigma_2 = \cdots = \sigma_i$ et $\tau = \tau_1 = \tau_2 = \cdots = \tau_j$. Si $\sigma(a_1a_2\cdots a_i) = \tau(a_1a_2\cdots a_j)$ alors $\sigma(a_i)$ et $\tau(a_j)$ finissent par la même lettre, donc $a_i = a_j = e$. Ainsi, on a $a_1a_2\cdots a_i = a_1a_2\cdots a_j = bwe$. Enfin, si $\sigma(a_1a_2\cdots a_i)$ est un préfixe propre de $\tau(a_1a_2\cdots a_j)$, alors $\sigma_{i+1} = \sigma$. En effet, si on suppose qu'il existe un mot non vide s tel que $\sigma(a_1a_2\cdots a_i)s = \tau(a_1a_2\cdots a_j)$. Alors d'un coté, s commence avec la même lettre que $\sigma_{i+1}(a_{i+1})$ et d'un autre coté, s appartient à $\{d0, d1\}^*d$ car $\sigma(a_1a_2\cdots a_i)$ appartient à $b\{d0, d1\}^*$ alors que $\tau(a_1a_2\cdots a_j) \in bd\{0d, 1d\}^*$. Par conséquent, $\sigma_{i+1}(a_{i+1})$ commence par d. Donc $\sigma_{i+1} = \sigma$. On prouve de la même façon que si $\tau(a_1a_2\cdots a_j)$ est préfixe propre de $\sigma(a_1a_2\cdots a_i)$, alors $\tau_{i+1} = \tau$. □

Théorème 6.1.6. *Le problème* MMPCP(7) *est indécidable sur les instances de Claus.*

Démonstration. Grâce au Théorème 6.1.4, on sait que les problèmes PCP et MMPCP sont équivalents sur les instances de Claus et comme le problème PCP(7) est indécidable, alors le problème MMPCP(7) l'est aussi. □

Notons que la décidabilité des problèmes MMPCP(k) pour $k \in \{2, \dots, 6\}$ est toujours inconnue.

6.1.2 Quelques résultats importants

Nous allons démontrer que le problème FREE[$\mathbb{W} \times \mathbb{W}$] est indécidable pour tout nombre entier $k \geq 13$. Pour cela, nous allons construire une réduction du problème MMPCP(7) sur des instances de Claus au problème FREE(13)[$\mathbb{W} \times \mathbb{W}$].

Lemme 6.1.7. *Soient S un semi-groupe, X un sous-ensemble de S, $s_1, t_1, s_2, t_2 \in X$ et $Y = (X \setminus \{s_1, t_1, s_2, t_2\}) \cup \{t_2s_1, s_2t_1, t_2t_1\}$. Si X est un code, alors Y est un code.*

Démonstration. L'ensemble Y est un code préfixe, c'est-à-dire que aucun de ses éléments n'est préfixe propre d'un autre. De plus, tout élément de Y appartient à X^+ donc Y est un code préfixe sur X. Ainsi, si X est un code, Y aussi. □

La réciproque de ce Lemme n'est pas vraie. En effet, si on considère le cas où $S = \{0, 1, 2\}^+$, $s_1 = 01$, $t_1 = 2$, $s_2 = 0$, $t_2 = 12$ et $X = \{s_1, t_1, s_2, t_2\}$. Ainsi, $Y = \{t_2s_1, s_2t_1, t_2t_1\}$ $= \{1201, 02, 122\}$ est un code préfixe, alors que X n'est pas un code car $s_1t_1 = 012 = s_2t_2$.

Théorème 6.1.8. *Soit k un nombre naturel. Si* FREE$(2k-1)[\mathbb{W} \times \mathbb{W}]$ *est décidable alors les deux problèmes* PCP(k) *et* MMPCP(k) *sont décidables sur des instances de Claus.*

Démonstration. Soient un alphabet Σ contenant les lettres e et b et (Σ, σ, τ) une instance de Claus du problème PCP(k). Pour tout mot $w \in \Sigma^*$, on pose

$$s_w = (\sigma(w), w) \text{ et } t_w = (\tau(w), w).$$

Soient X et Y deux sous-ensembles de $\{0, 1, b, e, d\}^* \times \Sigma^*$ définis par :

$$X = \{s_a : a \in \Sigma\} \cup \{t_a : a \in \Sigma\}$$

et

$$Y = (X \setminus \{s_b, t_b, s_e, t_e\}) \cup \{t_es_b, s_et_b, t_et_b\}.$$

Soient $\phi\colon \Sigma^* \to \mathbb{W}$ et $\psi\colon \{0, 1, b, e, d\}^* \to \mathbb{W}$ deux morphismes injectifs et posons

$$Z = \{(\psi(y), \phi(y')) : (y, y') \in Y\}.$$

Les ensembles X, Y et Z sont calculables à partir de (Σ, σ, τ). De plus, la cardinalité de l'ensemble X est égale à deux fois la cardinalité de Σ, c'est-à-dire $2k$ et la cardinalité de l'ensemble Y est égale à $2k - 1$. Ainsi, l'ensemble Z est une instance du problème FREE$(2k-1)[\mathbb{W} \times \mathbb{W}]$. Enfin, il nous reste à démontrer que les cinq assertions suivantes sont équivalentes :

1) (Σ, σ, τ) est une instance positive de PCP(k).
2) (Σ, σ, τ) est une instance positive de MMPCP(k).
3) X n'est pas un code.
4) Y n'est pas un code.
5) Z n'est pas un code.

Grâce à la Proposition 6.1.4, on sait que les problèmes PCP et MMPCP sont équivalents sur des instances de Claus. Ainsi les assertions 1) et 2) sont équivalentes. La Proposition 6.1.3 assure que les assertions 2) et 3) sont équivalentes étant donné la définition des ensembles s_w et t_w. Ensuite, le Lemme 6.1.7 entraine que 4) implique 3). De plus, comme l'ensemble Z est l'image de l'ensemble Y par les morphismes injectifs ϕ et ψ, alors les assertions 4) et 5) sont équivalentes.

Enfin, il reste à démontrer que 2) implique 4). Si on suppose que (Σ, σ, τ) est une instance positive de MMPCP(k), alors par la Proposition 6.1.4, il existe un mot $w \in (\Sigma \setminus \{b, e\})^*$ tel que $\sigma(bwe) = \tau(bwe)$. De ce fait, le mot w satisfait l'égalité $s_b s_w s_e = t_b t_w t_e$. En effet, on a

$$\begin{aligned} s_b s_w s_e &= (\sigma(b), b)(\sigma(w), w)(\sigma(e), e) \\ &= (\sigma(b)\sigma(w)\sigma(e), bwe) \\ &= (\sigma(bwe), bwe) \\ &= (\tau(bwe), bwe) \\ &= (\tau(b), b)(\tau(w), w)(\tau(e), e) \\ &= t_b t_w t_e. \end{aligned}$$

Et donc on obtient l'équation

$$t_e s_b s_w s_e t_b = t_e t_b t_w t_e t_b,$$

où $t_e s_b, t_e t_b \in Y$, $s_w s_e t_b, t_w t_e t_b \in Y^+$ et $t_e s_b = (\tau(e), e)(\sigma(b), b) \neq (\tau(e), e)(\tau(b), b) = t_e t_b$. Ainsi, l'ensemble Y n'est pas un code. □

Corollaire 6.1.9. *Pour tout nombre entier $k \geq 13$, le problème* Free$(k)[\mathbb{W} \times \mathbb{W}]$ *est indécidable.*

Démonstration. On sait que le problème PCP(7) est indécidable sur les instances de Claus. Donc grâce au Théorème 6.1.8, on obtient que le problème Free$(13)[\mathbb{W} \times \mathbb{W}]$ est indécidable. Ensuite, on conclut grâce à la Proposition 6.1.1. □

Il nous reste enfin à démontrer l'indécidabilité du problème Free$(k)[\mathbb{N}^{3\times 3}]$ pour tout entier $k \geq 13$.

Définition 6.1.10. Soit un alphabet A. Nous utiliserons dans la preuve qui suit l'opérateur *miroir* défini récursivement par $\epsilon^R = \epsilon$ et $(wa)^R = aw^R$ pour tout mot $w \in A^*$ et lettre $a \in A$. Le miroir du mot $a_1 \cdots a_n$ est simplement le mot $(a_1 \cdots a_n)^R = a_n \cdots a_1$ [20].

Lemme 6.1.11. *Il existe un morphisme injectif de $\mathbb{W} \times \mathbb{W}$ dans $\mathbb{N}^{3\times 3}$.*

Démonstration. Notons $\beta \colon \mathbb{W} \to \mathbb{N}$ l'application définie par : $\beta(w) = \mathrm{val}_2(w^R)$ pour tout mot $w \in \mathbb{W}$. Notons $\Phi \colon \mathbb{W} \times \mathbb{W} \to \mathbb{N}^{3\times 3}$ l'application définie par :

$$\Phi(u, v) = \begin{pmatrix} 2^{|u|} & 0 & \beta(u) \\ 0 & 2^{|v|} & \beta(v) \\ 0 & 0 & 1 \end{pmatrix}$$

pour tous $u, v \in \mathbb{W}$. On remarque facilement que Φ est un morphisme. En effet,

$$\begin{aligned}
\Phi(u,v)\Phi(u',v') &= \begin{pmatrix} 2^{|u|} & 0 & \mathrm{val}_2(u^R) \\ 0 & 2^{|v|} & \mathrm{val}_2(v^R) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{|u'|} & 0 & \mathrm{val}_2(u'^R) \\ 0 & 2^{|v'|} & \mathrm{val}_2(v'^R) \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 2^{|u|+|u'|} & 0 & 2^{|u|}\mathrm{val}_2(u'^R) + \mathrm{val}_2(u^R) \\ 0 & 2^{|v|+|v'|} & 2^{|v|}\mathrm{val}_2(v'^R) + \mathrm{val}_2(v^R) \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 2^{|uu'|} & 0 & \mathrm{val}_2((uu')^R) \\ 0 & 2^{|vv'|} & \mathrm{val}_2((vv')^R) \\ 0 & 0 & 1 \end{pmatrix} \\
&= \Phi(uu', vv')
\end{aligned}$$

pour tous $u, u', v, v' \in \mathbb{W}$. On remarque que β n'est pas injectif car $\beta(u) = \beta(u0)$ pour tout $u \in \mathbb{W}$. Par contre, la fonction qui envoie chaque $u \in \mathbb{W}$ sur $(|u|, \beta(u))$ est injective car $(|u|, \beta(u)) = (|u'|, \beta(u')) \Rightarrow |u| = |u'|$ et $\beta(u) = \beta(u') \Rightarrow u = u'$ pour tous $u, u' \in \mathbb{W}$. Ainsi, l'application Φ est elle aussi injective car

$$\begin{aligned}
\Phi(u,v) = \Phi(u',v') &\Rightarrow \begin{pmatrix} 2^{|u|} & 0 & \beta(u) \\ 0 & 2^{|v|} & \beta(v) \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2^{|u'|} & 0 & \beta(u') \\ 0 & 2^{|v'|} & \beta(v') \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{cases} 2^{|u|} = 2^{|u'|} \\ 2^{|v|} = 2^{|v'|} \\ \beta(u) = \beta(u') \\ \beta(v) = \beta(v') \end{cases} \\
&\Rightarrow (u,v) = (u',v')
\end{aligned}$$

pour tous $(u,v), (u',v') \in \mathbb{W} \times \mathbb{W}$. □

Théorème 6.1.12. *Soit k un nombre naturel, si le problème* $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$ *est décidable alors* $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ *est décidable.*

Démonstration. Grâce au Lemme 6.1.11, on sait qu'il existe un morphisme injectif $\sigma\colon \mathbb{W} \times \mathbb{W} \to \mathbb{N}^{3\times 3}$. Donc $S \subseteq \mathbb{W} \times \mathbb{W}$ est une instance positive de $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ si et seulement si S est un code à k éléments. Comme de plus, σ est injectif sur S, la condition précédente est équivalente au fait que $\sigma(S)$ est un code, étant donnée la Propriété 1.3.3. Comme $\sigma(S) \subseteq \mathbb{N}^{3\times 3}$, $\sigma(S)$ est une instance positive de $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$. On a donc construit une réduction de $\textsc{Free}(k)[\mathbb{W} \times \mathbb{W}]$ à $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$. □

A partir du Corollaire 6.1.9 et du Théorème 6.1.12 on déduit directement le corollaire suivant.

Corollaire 6.1.13. *Pour tout entier $k \geq 13$, le problème* $\textsc{Free}(k)[\mathbb{N}^{3\times 3}]$ *est indécidable.*

6.1.3 Problème de mortalité sur les matrices 3×3

De la même manière que dans la Section 4.4, on définit le problème de décision MORTAL$[\mathbb{Q}^{3\times3}]$ de la façon suivante : étant donné un ensemble $P \subseteq \mathbb{Q}^{3\times3}$, déterminer s'il existe un produit fini de matrices de P égal à la matrice nulle.

Nous allons démontrer, en se basant sur l'article de PATERSON [17], que le problème MORTAL$[\mathbb{Q}^{3\times3}]$ est indécidable. La démonstration repose sur l'indécidabilité du problème de correspondance de Post dont nous avons discuté dans le Chapitre 5.

Théorème 6.1.14. *Le problème* MORTAL$[\mathbb{Q}^{3\times3}]$ *est indécidable.*

Démonstration. Considérons un ensemble $H \subseteq \mathbb{Q}^{3\times3}$ constitué des deux matrices particulières

$$S = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ et } T = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

ainsi que des matrices du type

$$W_j = \begin{pmatrix} p_j & 0 & 0 \\ 0 & r_j & 0 \\ q_j & s_j & 1 \end{pmatrix}$$

pour $j \in \{1, \dots m\}$, où $p_j > q_j \geq 0$ et $r_j > s_j \geq 0$.

Remarquons que pour tous nombres entiers a, b, c, on a

$$\begin{pmatrix} a & b & c \end{pmatrix} S = a \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$$

et

$$\begin{pmatrix} a & b & c \end{pmatrix} T = (a-b) \begin{pmatrix} 1 & -1 & 0 \end{pmatrix}.$$

De plus, les matrices W_j sont toutes inversibles.

Ainsi, une condition nécessaire et suffisante pour obtenir un produit de matrices égal à la matrice nulle est donnée par l'existence d'un produit X de matrices W_j tel que

(a) $\begin{pmatrix} 1 & -1 & 0 \end{pmatrix} X = \begin{pmatrix} 0 & h & k \end{pmatrix}$, ou

(b) $\begin{pmatrix} 1 & -1 & 0 \end{pmatrix} X = \begin{pmatrix} h & h & k \end{pmatrix}$, ou

(c) $\begin{pmatrix} 1 & 0 & 1 \end{pmatrix} X = \begin{pmatrix} 0 & h & k \end{pmatrix}$, ou

(d) $\begin{pmatrix} 1 & 0 & 1 \end{pmatrix} X = \begin{pmatrix} h & h & k \end{pmatrix}$,

pour $h, k \in \mathbb{Q}$. En effet, si la condition (a) est satisfaite, le produit de matrices

$$TXS = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} X \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & h & k \\ 0 & -h & -k \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

rend la matrice nulle. Si la condition (b) est satisfaite on a $TXT = 0$. Si la condition (c) est satisfaite, on a $SXS = 0$ et enfin si (d) est satisfait on a $SXT = 0$.

Cependant, si on a

$$\begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix} W_j = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}$$

pour un certain j, alors

$$(u_1 > 0, u_2 < 0 \text{ et } u_3 = 0) \implies (v_1 > 0, v_2 < 0 \text{ et } v_3 = 0)$$

et

$$(u_1 > 0, u_2 \geq 0 \text{ et } u_3 = 1) \implies (v_1 > 0, v_2 \geq 0 \text{ et } v_3 = 1).$$

En effet,

$$\begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix} \begin{pmatrix} p_j & 0 & 0 \\ 0 & r_j & 0 \\ q_j & s_j & 1 \end{pmatrix} = \begin{pmatrix} u_1 p_j + u_3 q_j & u_2 r_j + u_3 s_j & u_3 \end{pmatrix}.$$

Donc si $u_1 > 0, u_2 < 0$ et $u_3 = 0$, alors $u_1 p_j + u_3 q_j > 0, u_2 r_j + u_3 s_j < 0$ et $u_3 = 0$ car $p_j, r_j > 0$. De même, si $u_1 > 0, u_2 \geq 0$ et $u_3 = 1$ alors $u_1 p_j + u_3 q_j > 0, u_2 r_j + u_3 s_j \geq 0$ et $u_3 = 1$.

Cela implique qu'il ne reste que la condition (d) qui peut être satisfaite. Donc il existe un produit de matrices de H égal à la matrice nulle si et seulement si il existe un produit X de matrices W_j tel que

$$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix} X = \begin{pmatrix} h & h & k \end{pmatrix},$$

pour $h > 0$. Ainsi, on aura $SXT = 0$.

Voici un exemple pour illustrer le lien entre la multiplication matricielle et la concaténation de mots (sur l'alphabet $\{1, 2, 3\}$ ici) :

$$\begin{pmatrix} 123 & 12 & 1 \end{pmatrix} \begin{pmatrix} 1000 & 0 & 0 \\ 0 & 100 & 0 \\ 223 & 32 & 1 \end{pmatrix} = \begin{pmatrix} 123223 & 1232 & 1 \end{pmatrix}.$$

De manière générale, étant donnée une paire de mots (U, V) sur l'alphabet $\{1, 2, 3\}$, on définit la matrice

$$W(U, V) = \begin{pmatrix} p & 0 & 0 \\ 0 & r & 0 \\ q & s & 1 \end{pmatrix}$$

où q, s sont les nombres représentés par les symboles des mots U, V respectivement (ou 0 pour le mot vide) et p, r sont les nombres entiers obtenus en écrivant 1 puis autant de zéros qu'il y a de symboles dans les mots U, V respectivement. Illustrons cela par un exemple pour comprendre : si on considère la paire de mots $(U, V) = (3112, 231)$ alors

$$W(U, V) = \begin{pmatrix} 10000 & 0 & 0 \\ 0 & 1000 & 0 \\ 3112 & 231 & 1 \end{pmatrix}.$$

On remarque alors que si X, Y, U, V sont des mots sur l'alphabet $\{1, 2, 3\}$, on obtient l'égalité

$$\begin{pmatrix} X & Y & 1 \end{pmatrix} W(U, V) = \begin{pmatrix} \overline{XU} & \overline{YV} & 1 \end{pmatrix}$$

où $\overline{\ }$ attire l'attention sur le fait que la concaténation des deux mots doit être interprétée comme un nombre entier. Remarquons, de plus, que $W(U,V)$ satisfait les conditions imposées sur les matrices W_j de H.

Posons maintenant $K = \{(U_1,V_1),(U_2,V_2),\ldots,(U_n,V_n)\}$ un ensemble de paires de mots sur l'alphabet $\{2,3\}$. Ainsi, on peut définir l'ensemble $H(K)$ de matrices par

$$\{S,T\} \cup \bigcup_{j=1}^{n} W(U_j,V_j) \cup \bigcup_{j=1}^{n} W(U_j,1V_j).$$

Par conséquent, l'ensemble $H(K)$ possède un produit de matrices égal à la matrice nulle si et seulement si il existe un produit X des matrices W de $H(K)$ tel que

$$\begin{pmatrix} 1 & 0 & 1 \end{pmatrix} X = \begin{pmatrix} h & h & 1 \end{pmatrix}$$

pour $h \in \mathbb{Q}$. Un tel h devrait commencer par 1 et puis être suivi uniquement de 2 et de 3 car les mots U_i, V_i sont construits sur l'alphabet $\{2,3\}$, ainsi, un tel produit existe si et seulement si le problème de correspondance de Post pour K tel que introduit au début de la Section 5.1 possède une solution. On conclut à l'aide de la Proposition 5.1.3. □

6.2 Matrices carrées de plus grande dimension

L'objectif principal de cette section est de démontrer que le problème $\textsc{Free}(2)[\mathbb{N}^{d\times d}]$ est indécidable pour certains nombres entiers $d \geq 1$.

Définition 6.2.1. Soit S un semi-groupe. Pour tous nombre entier $d \geq 1$, élément $x \in S$ et sous-ensemble $Y \subseteq S$, on définit

$$C_d(x,Y) = \{x^d\} \cup \bigcup_{i=0}^{d-1} x^i Y.$$

Lemme 6.2.2. *Soient $d \in \mathbb{N}_0$, a un symbole et Σ un alphabet tel que $a \notin \Sigma$.*

(i) Supposons que Σ est fini et de cardinalité k. La cardinalité de $C_d(a,\Sigma)$ est égale à $kd+1$.

(ii) Le langage $C_d(a,\Sigma)$ est un code préfixe.

(iii) Tout mot non vide sur $\{a\} \cup \Sigma$ qui ne se termine pas par a, appartient à $C_d(a,\Sigma)^+$.

Démonstration. Le langage $C_d(a,\Sigma)$ est l'ensemble $\{a^d, a^0\Sigma, a\Sigma, aa\Sigma, \ldots, a^{d-1}\Sigma\}$. Donc si Σ est de cardinalité k alors $C_d(a,\Sigma)$ est de cardinalité $kd+1$. De plus, pour tout mot $x \in C_d(a,\Sigma)$ et tout $s \in \Sigma^+$, on a bien $xs \notin C_d(a,\Sigma)$.

Il reste donc à démontrer le point *(iii)*. Considérons $(n,b) \in \mathbb{N}\times\Sigma$ et notons n sous la forme $n = qd+r$ avec $q \in \mathbb{N}$ et $r \in \{0,\ldots,d-1\}$. Comme a^d et $a^r b$ appartiennent à $C_d(a,\Sigma)$, alors $a^n b = (a^d)^q(a^r b)$ est un élément de $C_d(a,\Sigma)^+$. Si on pose

$$L = \{a^n b : (n,b) \in \mathbb{N}\times\Sigma\},$$

alors on vient juste de démontrer que $L \subseteq C_d(a,\Sigma)^+$. Il s'en suit que $L^+ \subseteq C_d(a,\Sigma)^+$. Donc comme L^+ est l'ensemble des mots non vides sur $\{a\} \cup \Sigma$ qui ne finissent pas par a, le point (iii) est démontré. □

Lemme 6.2.3. *Soient S un semi-groupe, $d \in \mathbb{N}_0$, x un élément de S et Y un sous-ensemble fini de S de cardinalité k tel que $x \notin Y$. Alors la cardinalité de $C_d(x,Y)$ est égale à $kd+1$ et l'ensemble $\{x\} \cup Y$ est un code si et seulement si $C_d(x,Y)$ est un code.*

Démonstration. La condition nécessaire découle du Lemme 6.2.2(i) et (ii) avec $a = x$ et $\Sigma = Y$. Démontrons la condition suffisante.

Soient Σ un alphabet de cardinalité k, a un symbole tel que $a \notin \Sigma$, et un morphisme $\sigma\colon (\{a\}\cup\Sigma)^+ \to S$ tel que $\sigma(a) = x$ et $\sigma(\Sigma) = Y$. On remarque que $\sigma(C_d(a,\Sigma)) = C_d(x,Y)$. Supposons que la cardinalité de $C_d(x,Y)$ est égale à $kd+1$ et que $C_d(x,Y)$ est un code. Alors, par le Lemme 6.2.2, σ induit une bijection de $C_d(a,\Sigma)$ dans $C_d(x,Y)$ car les deux ensembles ont la même dimension et

$$\sigma(C_d(a,\Sigma)) = C_d(x,Y).$$

Donc par la Proposition 1.3.2, σ est injectif sur $C_d(a,\Sigma)^+$. Soient $u, v \in (\{a\} \cup \Sigma)^+$ tels que $\sigma(u) = \sigma(v)$ et soit $b \in \Sigma$. Alors $\sigma(ub) = \sigma(vb)$ et $ub, vb \in C_d(a,\Sigma)^+$ par le Lemme 6.2.2(iii). Comme σ est injectif sur $C_d(a,\Sigma)^+$, on a $ub = vb$, et donc $u = v$. On obtient que σ est injectif. Donc par la Proposition 1.3.2, l'ensemble $\{x\} \cup Y$ est un code. □

Théorème 6.2.4. *Soient S un semi-groupe récursif et k, d des nombres entiers strictement positifs. Si* Free$(kd+1)[S]$ *est décidable, alors* Free$(k+1)[S]$ *est décidable.*

Démonstration. Pour tous élément $x \in S$ et sous-ensemble $Y \subseteq S$ à k éléments, $C_d(x,Y)$ est calculable à partir de x et Y car l'opération de S est calculable. Donc grâce au Lemme 6.2.3, on obtient une réduction du problème Free$(kd+1)[S]$ au problème Free$(k+1)[S]$. □

Si Free$(k_0)[S]$ est indécidable pour un nombre entier $k_0 \geq 2$, alors par le Théorème 6.2.4, Free$((k_0-1)d+1)[S]$ est indécidable pour tout nombre entier $d \geq 1$.

Corollaire 6.2.5. *Soit S un semi-groupe récursif.*

- (i) *S'il existe un nombre entier $k \geq 2$ tel que* Free$(k)[S]$ *est décidable, alors* Free$(2)[S]$ *est décidable.*
- (ii) *S'il existe un nombre entier impair $k \geq 3$ tel que* Free$(k)[S]$ *est décidable, alors* Free$(3)[S]$ *est décidable.*

Démonstration. Tout nombre entier $k \geq 2$ peut s'écrire sous la forme $1\cdot(k-1)+1$. Donc par le Théorème 6.2.4, si le problème Free$(k)[S]$ est décidable alors le problème Free$(2)[S]$ aussi. Ainsi on a démontré le point (i).

Pour le point (ii) on procède de la même façon, sachant que tout nombre entier impair $k \geq 3$ s'écrit sous la forme $2(\frac{k-1}{2})+1$. □

Revenons maintenant au sujet qui nous intéresse, à savoir les matrices carrées de dimension plus grande que 2.

Théorème 6.2.6. *Soient D un semi-anneau récursif et k, d des nombres entiers positifs. Si le problème* $\textsc{Free}(k+1)[D^{d\times d}]$ *est décidable, alors le problème* $\textsc{Free}(kd+1)[D]$ *l'est aussi.*

Démonstration. Nous allons montrer qu'il existe une réduction du problème $\textsc{Free}(kd+1)[D]$ au problème $\textsc{Free}(k+1)[D^{d\times d}]$.

Pour tout $n \in \mathbb{N}$, notons I_n la matrice identité $n\times n$ sur D. Considérons le sous-ensemble

$$X = \{a\} \cup \{b_{i,j} : (i,j) \in \{1,\dots,d\} \times \{1,\dots,k\}\}$$

de D à $(kd+1)$ éléments. Pour tout $j \in \{1,\dots,k\}$, notons B_j la matrice $d\times d$ à coefficients dans D définie par : la dernière colonne de B_j est égale à la transposée de $(b_{d,j}\cdots b_{2,j}b_{1,j})$ et toutes les autres entrées sont égales à 0. Enfin, posons

$$A = \begin{pmatrix} O & a \\ I_{d-1} & O \end{pmatrix}$$

et $\mathcal{X} = \{A, B_1, B_2, \dots, B_k\}$. Par exemple, si $d = 4$ et $k = 3$ on a

$$\mathcal{X} = \left\{ \begin{pmatrix} 0&0&0&a \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix} \begin{pmatrix} 0&0&0&b_{4,1} \\ 0&0&0&b_{3,1} \\ 0&0&0&b_{2,1} \\ 0&0&0&b_{1,1} \end{pmatrix} \begin{pmatrix} 0&0&0&b_{4,2} \\ 0&0&0&b_{3,2} \\ 0&0&0&b_{2,2} \\ 0&0&0&b_{1,2} \end{pmatrix} \begin{pmatrix} 0&0&0&b_{4,3} \\ 0&0&0&b_{3,3} \\ 0&0&0&b_{2,3} \\ 0&0&0&b_{1,3} \end{pmatrix} \right\}.$$

On remarque que $\mathcal{X}$ est un sous-ensemble de $D^{d\times d}$ à $(k+1)$ éléments et que $\mathcal{X}$ est calculable à partir de X. Il nous reste donc à vérifier que X est un code pour l'opération multiplicative de D si et seulement si $\mathcal{X}$ est un code pour la multiplication matricielle induite par les opérations de D.

Notons $\mathcal{C}$ comme dans la Définition 6.2.1 :

$$\mathcal{C} = C_d(A, \{B_1,\dots,B_k\}) = \{A^d\} \cup \{A^{i-1}B_j : (i,j) \in \{1,\dots,d\} \times \{1,\dots,k\}\}$$

et l'application $\phi\colon D^{d\times d} \to D : M \mapsto (M)_{dd}$. On se souvient de l'ensemble $\textsc{Tri}(d, D)$ introduit dans la Section 4.1, c'est le semi-anneau des matrices triangulaires supérieures $d \times d$ à coefficients dans D.

Lemme 6.2.7.

i) ϕ *induit un morphisme de* $\textsc{Tri}(d, D)$ *dans* D.

ii) $\mathcal{C}$ *est un sous-ensemble de* $\textsc{Tri}(d, D)$.

iii) ϕ *induit une bijection de* $\mathcal{C}$ *dans* X.

Démonstration. Le point i) est évident car pour tous $T, T' \in \textsc{Tri}(d, D)$ on a

$$\phi(TT') = (TT')_{dd} = (T)_{dd}(T')_{dd} = \phi(T)\phi(T').$$

Ensuite on peut démontrer les points ii) et iii) simultanément. On voit facilement que

$$A^i = \begin{pmatrix} 0 & aI_i \\ I_{d-i} & 0 \end{pmatrix}$$

pour tout $i \in \{0, \dots, d\}$. En particulier, on a $A^d = aI_d$. Ainsi $A^d \in \textsc{Tri}(d, D)$ et $\phi(A^d) = a$. Maintenant considérons $i \in \{1, \dots, d\}$ et $j \in \{1, \dots, k\}$. Toutes les entrées non nulles de $A^{i-1}B_j$ sont situées dans la dernière colonne, donc $A^{i-1}B_j \in \textsc{Tri}(d, D)$. De plus, on a $\phi(A^{i-1}B_j) = b_{i,j}$. □

Ainsi, grâce au Lemme 6.2.7 iii), on sait que la cardinalité de $\mathcal{C}$ est égale à $kd + 1$ et donc, par le Lemme 6.2.3, $\mathcal{X}$ est un code si et seulement si $\mathcal{C}$ est un code. De plus, comme $\phi\colon \textsc{Tri}(d, D) \to X$ est injectif sur $\mathcal{C}$, alors grâce à la Proposition 1.3.3, X est un code implique que $\mathcal{C}$ est un code. Démontrons que l'inverse est aussi vrai. On a l'égalité $B_1M = B_1\phi(M)$ pour toute matrice $M \in \textsc{Tri}(d, D)$. Alors pour tous $M, M' \in \textsc{Tri}(d, D)$,

$$\phi(M) = \phi(M') \text{ implique } B_1M = B_1M'.$$

Si on suppose que $\mathcal{C}$ est un code, alors l'égalité $B_1M = B_1M'$ pour des matrices $M, M' \in \mathcal{C}$ nous donne $M = M'$. Ainsi, B_1 est simplifiable dans $\mathcal{C}^+$. De plus, comme $\mathcal{C}^+$ est un sous ensemble de $\textsc{Tri}(d, D)$ vu le Lemme 6.2.7 ii), alors ϕ est injectif sur $\mathcal{C}^+$. Ainsi, par la Proposition 1.3.3, X est un code.

La preuve du théorème est donc terminée car on a démontré que les trois assertions suivantes sont équivalentes : $\mathcal{X}$ est un code, X est un code et $\mathcal{C}$ est un code. □

Corollaire 6.2.8. *Pour tout $h \in \mathbb{N}$, les problèmes* $\textsc{Free}(7+h)[\mathbb{N}^{6\times6}]$, $\textsc{Free}(5+h)[\mathbb{N}^{9\times9}]$, $\textsc{Free}(4+h)[\mathbb{N}^{12\times12}]$, $\textsc{Free}(3+h)[\mathbb{N}^{18\times18}]$ *et* $\textsc{Free}(2+h)[\mathbb{N}^{36\times36}]$ *sont indécidables.*

Démonstration. Soient $k, d \in \mathbb{N}_0$. En appliquant le Théorème 6.2.6 avec $D = \mathbb{N}^{3\times3}$, on obtient que si le problème $\textsc{Free}(kd + 1)[\mathbb{N}^{3\times3}]$ est indécidable alors le problème $\textsc{Free}(k + 1)[\mathbb{N}^{3d\times3d}]$ est aussi indécidable. Grâce au Corollaire 6.1.13, on sait que le problème $\textsc{Free}(k)[\mathbb{N}^{3\times3}]$ est indécidable pour $k \geq 13$. Ainsi, on obtient les résultats désirés car 13 peut s'écrire sous la forme

$$6 \cdot 2 + 1, 2 \cdot 6 + 1, 4 \cdot 3 + 1, 3 \cdot 4 + 1 \text{ et } 1 \cdot 12 + 1$$

donc si $\textsc{Free}(13)[\mathbb{N}^{3\times3}]$ est indécidable alors $\textsc{Free}(7)[\mathbb{N}^{3\cdot2\times3\cdot2}]$, $\textsc{Free}(3)[\mathbb{N}^{3\cdot6\times3\cdot6}]$, $\textsc{Free}(5)[\mathbb{N}^{3\cdot3\times3\cdot3}]$, $\textsc{Free}(4)[\mathbb{N}^{3\cdot4\times3\cdot4}]$ et $\textsc{Free}(2)[\mathbb{N}^{3\cdot12\times3\cdot12}]$ sont indécidables. □

Lemme 6.2.9. *Pour tout semi anneau D récursif et pour tout nombre entier $d \geq 1$, il existe un morphisme injectif et calculable de $D^{d\times d}$ dans $D^{(d+1)\times(d+1)}$.*

Démonstration. Il suffit de considérer l'application

$$\sigma\colon D^{d\times d} \to D^{(d+1)\times(d+1)} : M \mapsto \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}.$$

Cette application est calculable, elle est injective car pour tous $M, N \in D^{d\times d}$, si $\sigma(M) = \sigma(N)$, alors $\begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} N & 0 \\ 0 & 1 \end{pmatrix}$. Donc $M = N$. De plus, σ est un morphisme car pour tous $M, N \in D^{d\times d}$, on a

$$\sigma(MN) = \begin{pmatrix} MN & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} N & 0 \\ 0 & 1 \end{pmatrix} = \sigma(M)\sigma(N).$$

□

Proposition 6.2.10. *Soient $k, d \in \mathbb{N}_0$. Si le problème* $\textsc{Free}(k)[\mathbb{N}^{d\times d}]$ *est indécidable, alors* $\textsc{Free}(k)[\mathbb{N}^{e\times e}]$ *est indécidable pour tout entier $e \geq d$.*

Démonstration. Cela se démontre de proche en proche grâce au Lemme 6.2.9. En effet, comme il existe un morphisme injectif de $\mathbb{N}^{d\times d}$ dans $\mathbb{N}^{(d+1)\times(d+1)}$ alors si le problème $\textsc{Free}(k)[\mathbb{N}^{d\times d}]$ est indécidable, le problème $\textsc{Free}(k)[\mathbb{N}^{(d+1)\times(d+1)}]$ l'est aussi. □

La Table 6.1 résume les résultats que l'on a obtenu sur la décidabilité du problème $\textsc{Free}(k)[\mathbb{N}^{d\times d}]$ lorsque $k \in \mathbb{N}\setminus\{0\}$ et $d \in \mathbb{N}\setminus\{0, 1\}$.

		k															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$\cdots$
d	2	D	○	○	○	○	○	○	○	○	○	○	○	○	○	○	$\cdots$
	3	D	○	○	○	○	○	○	○	○	○	○	○	U	U	U	$\cdots$
	4	D	○	○	○	○	○	○	○	○	○	○	○	U	U	U	$\cdots$
	5	D	○	○	○	○	○	○	○	○	○	○	○	U	U	U	$\cdots$
	6	D	○	○	○	○	○	U	U	U	U	U	U	U	U	U	$\cdots$
	7	D	○	○	○	○	○	U	U	U	U	U	U	U	U	U	$\cdots$
	8	D	○	○	○	○	○	U	U	U	U	U	U	U	U	U	$\cdots$
	9	D	○	○	○	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	10	D	○	○	○	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	11	D	○	○	○	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	12	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	13	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	14	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	15	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	16	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	17	D	○	○	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	18	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	19	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	20	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	21	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	22	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	23	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	24	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	25	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	26	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	27	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	28	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	29	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	30	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	31	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	32	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	33	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	34	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	35	D	○	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	36	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	37	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	38	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	39	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	40	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	$\cdots$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

TABLE 6.1 – Connaissances actuelles sur la décidabilité de $\text{FREE}(k)[\mathbb{N}^{d\times d}]$ pour toute paire (k, d).

Bibliographie

[1] BELL, P. et I. POTAPOV. Reachability problems in quaternion matrix and rotation semigroups. *Information and Computation*. 2008, 206(11), p. 1353–1361.

[2] BOURNEZ, O. et M. BRANICHKY. The mortality problem for matrices of low dimensions. *Theory of Computing Systems*. 2002, 35(4), p. 433–448.

[3] BRENNER, J. L. et A. CHARNOW. Free semigroups of 2×2 matrices. *Pacific Journal of Mathematics*. 1978, 77(1), p. 57–69.

[4] CASSAIGNE, J. et F. NICOLAS. On the decidability of semigroup freeness. *RAIRO Theor. Inform. Appl.* 2012, 46(3), p. 355–399.

[5] CHURCH, A. Review of [18]. *Journal of Symbolic Logic*. 1943, 8, p. 50–52.

[6] EHRENFEUCHT, A. et G. ROZENBERG. On the (generelized) Post corrspondance problem with lists of lenths 2. *In :* EVEN, S. et O. KARIV, éds, *Proceedings of the 8th International Colloquium on Automata, Languages and Programming (ICALP'81), vol.115*, Springer, 1981, pp. 408-416.

[7] HALAVA, V. Another proof of undecidability for the correspondence decision problem. *CoRR*. 2014. Disponible via l'URL `<http://arxiv.org/abs/1411.5197>`.

[8] HORN, R. A. et C. R. JOHNSON. *Matrix analysis*. Cambridge University Press, 1990. Corrected reprint of the 1985 original.

[9] J. CASSAIGNE, T. HARJU et J. KARHUMAKI. On the undecidability of freeness of matrix semigroups. *International Journal of Algebra and Computation*. 1999, 9(3-4), p. 295–305.

[10] JANUSZ, G. J. *Algebraic number fields*. Vol. 55. Pure and Applied Mathematics, 1973.

[11] KANNAN, R. et R. J. LIPTON. Polynomial-time algorithm for the orbit problem. *Journal of the Association for Computing Machinery*. 1986, 33(4), p. 808–821.

[12] KROM, M. et M. KROM. More on mortality. *The American Mathematical Monthly*. 1990, 97, p. 37–38.

[13] MANDEL, A. et I. SIMON. On finite semigroups of matrices. *Theoretical Computer Science*. 1977, 5(2), p. 101–111.

[14] MATIYASEVICH, Yu. et G. SENIZERGUES. Decision problems for semi-thue systems with a few rules. *Theoretical Computer Science*. 2005, 330(1), p. 145–169.

[15] MILLER, M.A. Mortality for sets of 2×2 matrices. *Mathematics Magazine*. 1994, 67, p. 210–213.

[16] NICOLAS, F. (Generalized) Post correspondence problem and semi-Thue systems. 2008. Disponible via l'URL <http://arxiv.org/abs/0802.0726>.

[17] PATERSON, M. S. Unsolvability in 3×3 matrices. *Studies in Applied Mathematics*. 1970, 49, p. 105–107.

[18] POST, E. L. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*. 1943, 65, p. 197–215.

[19] POST, E. L. A variant of recursively unsolvable problem. *Bulletin of the American Mathematical Society*. 1946, 52(4), p. 264–268.

[20] RIGO, M. *Algorithmique et Calculabilité*. Département de Mathématiques de l'Université de Liège, 2009-2010, notes du cours de troisième bachelier en sciences mathématiques.

[21] TARSKI, A. *A decision method for elementary algebra and geometry*. seconde edition éd. University of California Press, 1951.

[22] V. HALAVA, T. HARJU et M. HIVENSALO. Undecidability bounds for integer matrices using claus instances. *International Journal of Foundations of Computer Science*. 2007, 18(5), p. 931–948.

[23] WOLPER, P. *Introduction à la calculabilité*. 2006.