Theoretical Computer Science 1005 (2024) 114637

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On simulating Turing machines with matrix semigroups with integrality tests ☆

Vesa Halava [a,1], Reino Niskanen [b,*]

[a] Department of Mathematics and Statistics, University of Turku, FIN-20014 Turku, Finland
[b] Liverpool John Moores University, UK

ARTICLE INFO

Keywords:
Turing machines
Matrix semigroup
Simulation
Undecidability
Identity problem

ABSTRACT

We present a construction to simulate Turing machines with 3×3 matrices over rationals. The correctness of simulation is guaranteed by testing that the matrices have integral elements during the simulation. This construction implies an undecidability result for a special identity problem for semigroups of 3×3-matrices.

1. Introduction

In this article we prove that a Turing machine can be simulated by a matrix semigroup over rational numbers with integrality tests. That is, the generators of the semigroup are rational matrices, but the product matrices remain integral during a correct simulation. Indeed, multiplying with a matrix such that an element in a simulation matrix turns non-integral is equivalent to usage of an incorrect transition in the Turing machine side. This allows us to faithfully simulate any Turing machine.

As a consequence, we prove that the identity problem, i.e., whether the identity matrix is in the generated semigroup, is undecidable in this setting. Although this result seems to be a quite traditional undecidability result for semigroups generated by rational matrices, our original motivation for this study is quite far from traditional. Our goal is to prove an undecidability result for matrices such that in the products (simulating the computational system reduced to it) the elements of the matrices are significantly smaller than in the traditional reductions. Also, we are interested in pure modelling, i.e., in the question how to simulate a computational system operating with sequences of symbols using matrices.

Indeed, most of the known undecidability reductions for problems in integer matrix semigroups rely on the undecidability of the *Post Correspondence Problem*, PCP for short, or some of its variants. There are also some proofs that use the *Hilbert's Tenth* problem in the reduction; see for example [1]. In the PCP, for given two word morphisms $g, h : A^* \to B^*$, it is asked whether or not there exists a non-empty word w such the g and h agree on it, that is,

$$h(w) = g(w).$$

The traditional reduction from the PCP to integer matrices is based on an injective (n-ary) representation σ of words in $\mathbb{N}$ and the coding γ of pairs of words into 3×3 matrices so that the catenation operation of the word semigroups A^* and B^* are preserved in

☆ This article belongs to Section C: Theory of natural computing, Edited by Lila Kari.

* Corresponding author.
E-mail addresses: vesa.halava@utu.fi (V. Halava), r.niskanen@ljmu.ac.uk (R. Niskanen).

[1] Supported by emmy.network foundation under the aegis of the Fondation de Luxembourg.

https://doi.org/10.1016/j.tcs.2024.114637
Received 30 November 2023; Received in revised form 10 May 2024; Accepted 14 May 2024

Available online 21 May 2024

matrix multiplication. The formal definitions of σ and γ are given in Section 4, but let us give an example how these codings work: for example we may define γ such that for all words $u_1, v_1, u_2, v_2 \in B^*$,

$$\gamma(u_1,v_1)\gamma(u_2,v_2) = \begin{pmatrix} n^{|u_1|} & 0 & 0 \\ 0 & n^{|v_1|} & 0 \\ \sigma(u_1) & \sigma(v_1) & 1 \end{pmatrix}\begin{pmatrix} n^{|u_2|} & 0 & 0 \\ 0 & n^{|v_2|} & 0 \\ \sigma(u_2) & \sigma(v_2) & 1 \end{pmatrix} = \begin{pmatrix} n^{|u_1u_2|} & 0 & 0 \\ 0 & n^{|v_1v_2|} & 0 \\ \sigma(u_1u_2) & \sigma(v_1v_2) & 1 \end{pmatrix},$$

for a large enough $n \in \mathbb{N}$. Here $|u|$ denotes the length of the word u, that is, the number of symbols in u. Now if we simply set $M_i = \gamma(g(a_i), h(a_i))$ for all letters a_i in the alphabet A, we derive from the undecidability of the PCP that it is undecidable for the matrix semigroup generated by the matrices M_i whether or not there exists a matrix M in the semigroup such that $M_{31} = M_{32}$.

In our reduction, we shall use the above mentioned γ, but the mapping σ is modified. The main difference is that our reduction is one step below in the reduction chain of simulation. The key in all undecidability proofs of the PCP is that the pair of morphisms in the PCP can simulate a (universal) computational system such as *Turing machines* [15], *semi-Thue systems* [9], *tag systems* [23], *normal systems* [25], just to mention some of the most well-known systems. In all of these undecidability reductions, except in the Post's original proof from the normal systems [25], the simulation of the computation of the chosen system is done so that there is a nonempty word w for constructed morphisms g and h such that $g(w) = h(w)$ if and only if there exists a computation from a particular configuration u of the system to the configuration v and the word w is a catenation of all configurations (including the used transitions/rules of the used systems) along this computational path.[2] In other words, the word w is very long implying that the words $g(w)$ and $h(w)$ are very long and, therefore, the elements of the matrices $\gamma(g(w), h(w))$ become huge if we consider simulation of the reduced computational system with matrices. For example, the element $(1,3)$ in mapping γ is n^c where c is approximately the sum of lengths of all configurations in the computation of the system. Moreover, some of the elements never decrease when a universal system is simulated through the PCP with the products of matrices.

Our main motivation for this study is the simulation of computational system with matrices directly without remembering the whole history of computation in elements of the matrices implying that the elements are smaller. In our construction for the simulation, the elements of the matrices are integers encoding only the current configuration of the system. Therefore, the elements are much smaller than in a simulation using the PCP as a bridge from a computational system to matrices.

We apply the simulation construction and consider the existence of a particular computation of the simulated Turing machine. As a result we prove undecidability of a variant of the identity problem for matrix semigroups. The identity problem is a long-standing open problem. Unlike most other matrix semigroup problems, the three-dimensional case remains open. It was shown in [5] that the problem is undecidable for integral matrices of dimension four, and in [19], a better bound on the number of matrices in the generator set was given. For two-dimensional matrices, it is known that the identity problem is decidable for integer matrices—the problem is even NP-complete [3]—and undecidable over rational quaternions [2]. Recently, it was shown that there is no embedding of pairs of binary words into $\mathrm{SL}(3,\mathbb{Z})$ [19]. The result suggests that the identity problem is decidable for three-dimensional matrices as the vast majority of undecidability results rely on embeddings of pairs of binary words into matrices. Recently, there has been a surge of interest in the identity problem for different classes of matrices [11,12].

In order to prove undecidability of the identity problem with integrality tests, we use an encoding of pairs of words into matrices that allows us to simulate a Turing machine and, in particular, allows us to use the undecidability of the halting problem for the empty input in a special form. The integrality tests are then used to ensure that a faithful simulation is performed. The integrality test can be performed by checking after each matrix multiplication that the resulting matrix is integral.

Finally, note that simulation of a computational system, such as Turing machines, with integral or rational counters is by no means new. There are famous models such as the *Minsky machine* [22] and the *Fractran* model defined by Conway [10], just to mention two. Our model for the simulation, the integral/rational matrices, is significantly different as the "counters" act on matrices.

On the other hand, there is a vast literature on dynamics of loops of form

$$\texttt{while } (g(\mathcal{X}) = \texttt{true}) \texttt{ do } (\mathcal{X} := f(\mathcal{X})),$$

where $\mathcal{X}$ are the variables, g is a guard condition that the assignment of $\mathcal{X}$ has to satisfy and f is an update function that assigns new values to $\mathcal{X}$. Often, the variables are represented by d-dimensional vectors over $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \ldots$ The guard condition often defines a polytope, i.e., is a system of linear equalities and inequalities, [26,8,17,16,18], but can be defined by, e.g., Presburger formulas [13]. The update function is typically more varied as small tweaks to the function can lead to different results, but is often restricted to linear updates, i.e., multiplying by a matrix [6,7,18,20].

Our setting can be seen as a non-deterministic loop of form $\texttt{while } (g(\mathcal{X}) = \texttt{true}) \texttt{ do } (\mathcal{X} := f_1(\mathcal{X}) \texttt{ or } \mathcal{X} := f_2(\mathcal{X}) \texttt{ or } \ldots \texttt{ or } \mathcal{X} := f_n(\mathcal{X}))$, where $\mathcal{X}$ is a d-dimensional rational matrix, each f_i is a multiplication by a matrix and $g(\mathcal{X})$ returns `true` if and only if every component of $\mathcal{X}$ is integral. That is, the loop is of the form

$$\texttt{while } (M \overset{?}{\in} \mathbb{Z}^{d\times d} = \texttt{true}) \texttt{ do } (M := MM_1 \texttt{ or } M := MM_2 \texttt{ or } \ldots \texttt{ or } M := MM_n).$$

[2] In Post's original undecidability proof the word w consists only of the rules words used in the derivation of a normal system, not the full configurations of the derivation.

Recall, that the termination of non-deterministic loops with linear guards and linear updates is undecidable [26].

2. Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ be the sets of the natural numbers, the integers and the rational numbers. We denote by $\mathbb{P}$ the set of all primes.

A *semigroup* is a set equipped with an associative binary operation. Let S be a semigroup and G be a subset of S. We say that a semigroup S is *generated* by a subset G of S if each element of S can be expressed as a composition of elements of G. In this case, we call G a *generating set* of S and denote $S = \langle G \rangle$. Given an *alphabet* $\Sigma = \{a_1, a_2, \ldots, a_m\}$, a finite *word* u is an element of semigroup Σ^*. The *empty word* is denoted by ε. The length of a finite word u is denoted by $|u|$ and $|\varepsilon| = 0$.

We shall consider semigroups where the generators are $d \times d$ matrices (over rationals) and the composition operation is the matrix multiplication. Denote by $\boldsymbol{I}_d$ the d-dimensional *identity matrix*. If the dimension is clear from context, we denote the identity matrix simply by $\boldsymbol{I}$.

Let $G \subseteq \mathbb{K}^{d\times d}$ for some $d \in \mathbb{Z}_+$ and $\mathbb{K} \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$. Let us define the *integral* set $\langle G \rangle_{\mathbb{Z}} = \langle G \rangle \cap \mathbb{Z}^d$. That is, $\langle G \rangle_{\mathbb{Z}}$ consists of all elements $M \in \langle G \rangle$ such that $M \in \mathbb{Z}^{d\times d}$ even if generators used are not in $\mathbb{Z}^{d\times d}$. It is also possible to define an element to be *integral with respect to* Z for some $Z \subseteq G$. That is, $M \in \langle G \rangle$ is integral with respect to Z if $M = M'N$, for some $M' \in \langle G \rangle$ and $N \in Z$ such that $M'N \in \mathbb{Z}^{d\times d}$. In Section 6, we discuss a modification of our construction that takes the integrality with respect to a set into account.

Let us next prove a simple property regarding a product of primes. This lemma will be useful in upcoming sections when we show that an incorrect product of matrices cannot result in a correct product.

Lemma 1. *Let $p_1, p_2, \ldots, p_n$ be odd pairwise different primes, where $n \geq 2$. Then*

$$\prod_{i=1}^{n} p_i \neq \sum_{j=1}^{n} \left(a_j \prod_{\substack{i=1 \\ i\neq j}}^{n} p_i \right),$$

where $a_j \in \mathbb{N} \setminus \{0\}$ for all j.

Proof. Assume towards a contradiction that the equality holds. Now

$$\sum_{j=1}^{n} \left(a_j \prod_{\substack{i=1 \\ i\neq j}}^{n} p_i \right) = \sum_{j=1}^{n-1} \left(a_j \prod_{\substack{i=1 \\ i\neq j}}^{n} p_i \right) + a_n \prod_{\substack{i=1 \\ i\neq n}}^{n} p_i = p_n \sum_{j=1}^{n-1} \left(a_j \prod_{\substack{i=1 \\ i\neq j}}^{n-1} p_i \right) + a_n \prod_{i=1}^{n-1} p_i.$$

By the assumption, the above is equal to $\prod_{i=1}^{n} p_i$. By rearranging terms, we have the equation

$$(p_n - a_n) \prod_{i=1}^{n-1} p_i = p_n \sum_{j=1}^{n-1} \left(a_j \prod_{\substack{i=1 \\ i\neq j}}^{n-1} p_i \right).$$

The right-hand side is positive and divisible by p_n. On the other hand, the left-hand side is divisible by p_n only if $p_n - a_n$ is. The term $p_n - a_n$ cannot be both divisible by p_n and positive, hence we reach a contradiction. □

3. Halting problem

A Turing machine $\mathcal{M}$ (with a final state), TM for short, is a 7-tuple

$$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \star, h),$$

where Q is a finite set of states, q_0 is the initial state, $h \in Q$ is the final state, Σ is the input alphabet, Γ is the tape alphabet with $\Sigma \subseteq \Gamma$, and δ is a partial function $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ called the *transition function* where L and R are special direction symbols and $\star \in \Gamma$ is the blank symbol. The TM operates on a one-way infinite tape.

Note that the TM's are deterministic, however, we allow δ to be a partial function, i.e., it may be undefined for some values $(q, a) \in Q \times \Gamma$. Therefore, if $\delta(q, a)$ is defined, it is unique. Each *transition* of a TM $\mathcal{M}$ is of the form $\delta(p, a) = (q, b, D)$. Here D refers to "direction". The values L and R refer to "left move" and "right move", respectively. As the tape is one-way infinite, we can assume that the leftmost cell on the tape is $\triangleright$ and $\delta(q, \triangleright) = (q', \triangleright, R)$ for all $q \in Q$, and furthermore, that no other production rule writes $\triangleright$ on the tape.

A *configuration* of the TM $\mathcal{M}$, at some point in its computation, is the current state of the machine and the content of its tape. Let the content of the tape be $\triangleright uav \star \star \cdots$ where $u, v \in \Gamma^*$, assume that $\mathcal{M}$ is in state q reading the symbol $a \in \Gamma$ and assume further that $\triangleright uav$ is the shortest word containing all nonblank letters of the tape. Then the configuration represented by the word $\triangleright u(q, a)v \in \Gamma^*(Q \times \Gamma)\Gamma^*$ where v is either ε or ends with a nonblank letter.

A *step in a computation* or a *move* $\gamma \vdash_{\mathcal{M}} \gamma'$ yielding from one configuration γ of $\mathcal{M}$ to the next one γ' is defined in the usual way. We define here only the right-move, the left-move definition is analogous. Let the configuration be $\triangleright u(q,a)v$ and assume that $\delta(q,a) = (p,b,R)$. Then

$$\triangleright u(q,a)v \vdash_{\mathcal{M}} \begin{cases} \triangleright ub(p,\star), & \text{if } v = \varepsilon, \\ \triangleright ub(p,c)v', & \text{if } v = cv' \text{ and } c \in \Gamma. \end{cases}$$

As the TM's are deterministic and since δ is a (partial) function, for each configuration $\gamma = \triangleright u(q,a)v$, there exists at most one configuration γ' such that $\gamma \vdash_{\mathcal{M}} \gamma'$.

Let $\vdash^*_{\mathcal{M}}$ or $\vdash^*$, for short, be the reflexive and transitive closure of the relation $\vdash_{\mathcal{M}}$. Thus $\gamma \vdash^* \gamma'$ if and only if there exists a finite sequence $\gamma = \gamma_1 \vdash \gamma_2 \vdash \cdots \vdash \gamma_k = \gamma'$ of configurations for some $k \geq 1$ including the possibility that $\gamma = \gamma'$. Such a sequence is called a *computation* of $\mathcal{M}$. It is an *accepting* computation if the state in γ' is the unique final state h.

A seminal result in computability theory states that the *halting problem* of Turing machines on the empty input is undecidable; see, e.g., [21].

Theorem 2. *It is undecidable whether a given $\mathcal{M}$ TM halts on the empty input. That is, whether $(q_0, \triangleright\star) \vdash^* \triangleright u(h,a)v$, where $u, v \in \Gamma^*$, $a \in \Gamma$, holds or not.*

It is well-known that there is a myriad of ways to alter the definition of Turing machines or their structure and retain the undecidability of the halting problem. We shall modify any TM $\mathcal{M}$ to an equivalent TM $\mathcal{M}'$ as follows:

First, we may use the second marker ($\triangleleft$) to fully surround the non-empty portions of the infinite tape and additional states that move this marker if extra space is required by the machine. More precisely, when the space needs to be created on the right side of the tape, the right-marker needs to be moved one cell to the right. That is, if the machine is in state p, the current symbol read is the right-marker $\triangleleft$ and there is a right-move for $\delta(p,\star) = (q,b,R)$, then we add a new state $p_\triangleleft$ and transitions

$$\delta(p,\triangleleft) = (p_\triangleleft, \star, R) \quad \text{and} \quad \delta(p_\triangleleft, \star) = (p, \triangleleft, L). \tag{1}$$

Similarly, we need to remove extra $\star$ symbols between the markers. First of all, the extra $\star$ symbols are detected by adding a check for all transitions $\delta(p,a) = (q,\star,D)$ (except for those that were added for adding extra space). If it is then this extra $\star$ is shifted by the right-border marker $\triangleleft$, the machine is in a new state p reading $\triangleleft$ and we add transitions

$$\delta(p,\triangleleft) = (p',\star,L) \quad \text{and} \quad \delta(p',\star) = (p'',\triangleleft,L), \tag{2}$$

for new states p' and p'' and then the machine moves back to where it printed the extra $\star$ and reads the border marker next to it.

Secondly, we may assume that the first step of the TM is to write $\triangleright$ and $\triangleleft$ on the tape. We can further assume that the tape is cleared before meeting the final state h, that is the problem of halting is to decide whether or not $(q_0,\star) \vdash^* (h,\star)$.

It is obvious that the markers may be missing at some particular point of computation, but it is clear that $(q_0,\star) \vdash^*_{\mathcal{M}} \triangleright u(q,a)v$ if and only if $\triangleright(q_0,\star)\triangleleft \vdash^*_{\mathcal{M}'} \triangleright u(q,a)v\triangleleft$. Let us note that the above changes (1) and (2) are done in our matrix simulation with one single matrix in each case and the markers are never missing.

Theorem 3. *Let $\mathcal{M}$ be a Turing machine with delimiters, $\triangleright$ and $\triangleleft$, surrounding non-blank tape content and where the initial configuration $c = (q_0,\star)$. It is undecidable whether the machine reaches configuration c again.*

4. Matrix reachability from Turing machines

In this section, we simulate a Turing machine $\mathcal{M}$ using a matrix semigroup. That is, we will construct a set $\mathcal{M}_\delta = \{M_1, M_2, \ldots, M_k\} \subseteq \mathbb{Q}^{3\times3}$ that simulates $\mathcal{M}$ when the integrality test is performed after each multiplication.

The main idea in an encoding of the computation of a Turing machine is to cut the configuration $u(q,a)v$ into two words $u(q,a)$ and v, embed the pair of words into a matrix, and then to use specific matrices to move one symbol from one word to another.

It is worth highlighting that commonly an n-ary representation of words is done using a simple encoding of letters. Assume that the alphabet is binary, i.e., let $A = \{a,b\}$ and $w \in A^*$. Let $\tau : A \to \mathbb{N}$ be defined as $\tau(a) = 1$ and $\tau(b) = 2$. Then $\sigma' : A^* \to \mathbb{N}$ is defined by $\sigma'(w_1 w_2 \cdots w_k) = \sum_{i=1}^{k} \tau(w_i) \cdot 3^{n-i}$. See, for example, [24,14,4]. We use a different encoding that allows us to construct matrices with smaller elements.

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \star, h)$ be a Turing machine defined in the previous section. Let $C = (Q \times \Gamma) \cup \Gamma \cup \{\#\}$ be the set of symbols of a configuration, where # is a new symbol. Let $m = |C| = |Q| \cdot |\Gamma| + |\Gamma| + 1$ and $p_1, p_2, \ldots, p_m \in \mathbb{P} \setminus \{2\}$. Let $\varphi : C \to \mathbb{N}$ be an encoding defined by

$$\varphi(a_i) = \prod_{\substack{j=1 \\ j \neq i}}^{m} p_j,$$

for all $i = 1, \ldots, m$. That is, for any distinct $a, a' \in C$, $\gcd(a,a') \neq 1$. Denote by $\varphi(C) = \{\varphi(a) \mid a \in C\}$. Let $n > p_1 \cdots p_m$ and let $\sigma : C^* \to \mathbb{N}$ be the injective mapping using the n-ary representation. That is, we associate each letter of $\varphi(C)$ with a unique integer

in $\{1,\dots,n-1\}$ and for a word $w_1w_2\cdots w_k \in C^*$, $\sigma(w_1w_2\cdots w_k) = n^{k-1}\varphi(w_1) + n^{k-2}\varphi(w_2) + \cdots + n^0\varphi(w_k)$. For the below, note that $\sigma(uv) = n^{|v|}\sigma(u) + \sigma(v)$, for all $u, v \in C^*$ and especially that $\sigma((cv)^R) = n\sigma(v^R) + \sigma(c)$ for all $v \in C^*$ and $c \in C$.

Let γ be the mapping

$$\gamma(u,v) = \begin{pmatrix} n^{|u|} & 0 & 0 \\ 0 & n^{|v|} & 0 \\ \sigma(u) & \sigma(v) & 1 \end{pmatrix}. \tag{3}$$

Let $u(q,a)v$, where $\triangleright$ is the first symbol and $\triangleleft$ is the last symbol, be the current configuration of the deterministic TM $\mathcal{M}$. We represent this by

$$\gamma(u(q,a), \#v^R) = \begin{pmatrix} n^{|u(q,a)|} & 0 & 0 \\ 0 & n^{|v\#|} & 0 \\ \sigma(u(q,a)) & \sigma(\#v^R) & 1 \end{pmatrix}. \tag{4}$$

Note that # is a new marker symbol to ensure the element $(3,2)$ of our matrices is nonzero. Also note that the element $(3,1)$ is never zero as it has $\sigma(q,a)$ for some $q \in Q$ and $a \in \Gamma$.

We are ready to define the set of matrices $\mathcal{M}_\delta$. We begin with transitions added by applying (1) or (2) when modifying the TM, we study these cases separately. Consider a transition $\delta(q,a) = (p,b,R)$ of $\mathcal{M}$, we add matrix

$$M_{(q,a),c} = \begin{pmatrix} n & 0 & 0 \\ 0 & n^{-1} & 0 \\ -n\sigma((q,a)) + \sigma(b(p,c)) & -n^{-1}\sigma(c) & 1 \end{pmatrix},$$

for every $c \in \Gamma$ to $\mathcal{M}_\delta$. Note that $\mathcal{M}$ is deterministic, so the state p and symbols b are uniquely determined by (q,a). Similarly, a transition $\delta(q,a) = (p,b,L)$ is represented by a matrix

$$M_{(q,a),c} = \begin{pmatrix} n^{-1} & 0 & 0 \\ 0 & n & 0 \\ -n^{-1}\sigma(c(q,a)) + \sigma((p,c)) & \sigma(b) & 1 \end{pmatrix},$$

for every $c \in \Gamma$ which is also added to $\mathcal{M}_\delta$. Then for the transitions added when applying (1) (originally $\delta(p,\star) = (q,b,R)$) we add

$$M_{(p,\triangleleft),\triangleleft} = \begin{pmatrix} n & 0 & 0 \\ 0 & 1 & 0 \\ -n\sigma((p,\triangleleft)) + \sigma(b(q,\triangleleft)) & 0 & 1 \end{pmatrix}. \tag{5}$$

Similarly, the space removal in (2) is performed by a one special left-move matrix

$$M_{(p,\triangleleft),\star} = \begin{pmatrix} n^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ -n^{-1}\sigma(\star(p,\triangleleft)) + \sigma((p',\star)) & 0 & 1 \end{pmatrix},$$

and left-move matrices for $c \in \Gamma$

$$M_{(p',\star),c} = \begin{pmatrix} n^{-1} & 0 & 0 \\ 0 & n & 0 \\ -n^{-1}\sigma(c(p',\star)) + \sigma((p'',c)) & \sigma(\triangleleft) & 1 \end{pmatrix}.$$

Note also, that there exists at most one matrix in the set M_δ (moving either to the left or to the right) for all combinations of $(q,a) \in Q \times \Gamma$ and $c \in \Gamma$ as the TM $\mathcal{M}$ is deterministic.

Now, say the configuration of the Turing machine is $u(q,a)cv$ and that there exists a (unique) transition $\delta(q,a) = (p,b,R)$. The move of the TM $\mathcal{M}$ is represented by a product of the two matrices

$$\begin{aligned}
&\gamma(u(q,a), \#(cv)^R) M_{(q,a),c} \\
&= \begin{pmatrix} n^{|u(q,a)|+1} & 0 & 0 \\ 0 & n^{|\#cv|-1} & 0 \\ n\sigma(u(q,a)) - n\sigma((q,a)) + \sigma(b(p,c)) & n^{-1}\sigma(\#(cv)^R) - n^{-1}\sigma(c) & 1 \end{pmatrix} \\
&= \begin{pmatrix} n^{|ub(p,a)|} & 0 & 0 \\ 0 & n^{|\#v|} & 0 \\ \sigma(ub(p,c)) & \sigma(\#v^R) & 1 \end{pmatrix} \\
&= \gamma(ub(p,c), \#v^R),
\end{aligned}$$

since

$$\begin{aligned}
&n\sigma(u(q,a)) - n\sigma((q,a)) + \sigma(b(p,c)) \\
&= n^2\sigma(u) + n\sigma((q,a)) - n\sigma((q,a)) + \sigma(b(p,c)) = n^{|b(p,c)|}\sigma(u) + \sigma(b(p,c))
\end{aligned}$$

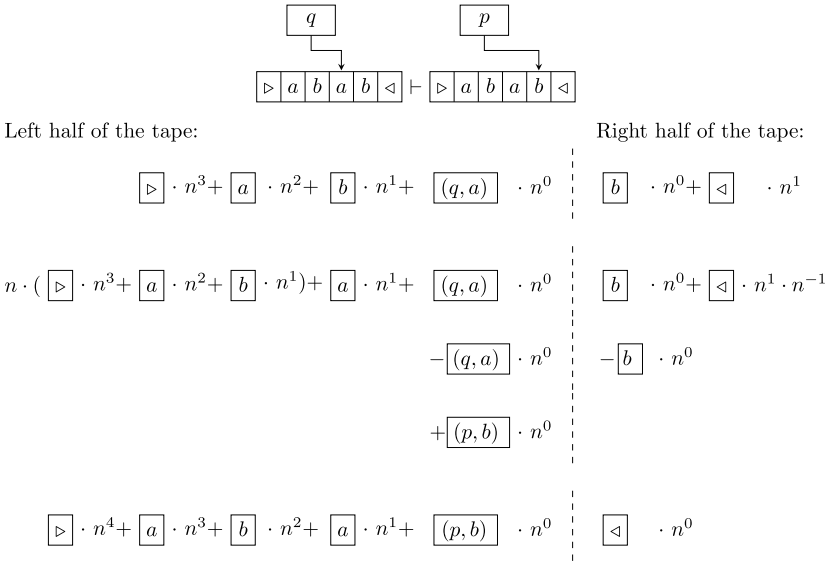

Fig. 1. An illustration of a transition of a TM and the corresponding encoding changes.

$$= \sigma(ub(p,c))$$

and

$$n^{-1}\sigma(\#(cv)^R) - n^{-1}\sigma(c) = n^{-1}(n\sigma(\#v^R) + \sigma(c)) - n^{-1}\sigma(c) = \sigma(\#v^R).$$

Similarly, we can show that if the configuration of the Turing machine is $uc(q,a)v$, using the unique transition $\delta(q,a) = (p,b,L)$ is represented by a product of the two matrices

$$\gamma(uc(q,a), \#v^R))M_{(q,a),c} = \gamma(u(p,c), (\#(bv)^R).$$

For the sake of readability, let us define a mapping $\psi : \mathbb{Q}^{3\times3} \to \mathbb{Q}^4$ by setting

$$\psi(M) = \psi\left(\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}\right) = (m_{11}, m_{31}, m_{22}, m_{32}).$$

We further define

$$\psi(M) \cdot \psi(N) = (m_{11}n_{11}, m_{31}n_{11} + n_{31}, m_{22}n_{22}, m_{32}n_{22} + n_{32}).$$

When restricted to matrices $\gamma(u(q,a), v^R)$ and $M_{(q,a),c}$ as defined above, the mapping is an isomorphism. To further simplify the notation, we will denote $\psi(\gamma(u(q,a), v^R))$ by $\psi(u(q,a), v^R)$.

Example 4. Let $\triangleright ab(q,a)b\triangleleft$ be a configuration of a TM and let us simulate transition $\delta(q,a) = (p,a,R)$. The subsequent configuration is $\triangleright aba(q,b)\triangleleft$. The transition of the TM and the changes in the coefficients in the encoding are depicted in Fig. 1.

Let us present a few observations next. Firstly, the mapping γ is into $\mathbb{N}^{3\times3}$. Secondly, all matrices $M_{(q,a),c}$ are rational, regardless of whether they correspond to the head moving left or right. If a configuration matrix is multiplied by the "correct" matrix, then resulting matrix is also integral, and even in $\mathbb{N}^{3\times3}$. On the other hand, multiplying by an "incorrect" matrix does not guarantee that the resulting matrix is not integral. Let us consider this in details:

For the right transitions, let $\psi(u(q,a), \#(cv)^R)$ be a configuration and $\psi(M_{(q',a'),d})$ correspond to a transition $\delta(q',a') = (p,b,R)$. The resulting vector is

$$\begin{aligned} &\left(n^{|u(q,a)|}, \sigma(u(q,a)), n^{|\#cv|}, \sigma(\#(cv)^R)\right) \cdot \left(n, -n\sigma((q',a')) + \sigma(b(p,d)), n^{-1}, -n^{-1}\sigma(d)\right) \\ &= \left(n^{|u(q,a)+1|}, n\sigma(u(q,a)) - n\sigma((q',a')) + \sigma(b(p,d)), n^{|\#v|}, n^{-1}\sigma(\#(cv)^R) - n^{-1}\sigma(d)\right) \end{aligned} \tag{6}$$

and it is integral as long as $c = d$ by the last element. In other words, the pair (q',a') does not have to match the pair (q,a) of the configuration to ensure that the product is integral. In this case, $\sigma((q,a)) - \sigma((q',a')) \neq 0$ will be in the coefficient of n in the second component of the ψ mapping. It remains to show that this remainder cannot be removed by further applications of "incorrect" matrices. This is proven in the upcoming Lemma 5.

Next we consider left transitions as this case is significantly simpler. Indeed, when $\psi(uc(q,a),\#v^R)$ is multiplied by $\psi(M_{(q',a'),d})$ corresponding to $\delta(q',a')=(p,b,L)$, we have

$$\begin{aligned}&\left(n^{|uc(q,a)|},\sigma(uc(q,a)),n^{|\#v|},\sigma(\#v^R)\right)\cdot\left(n^{-1},-n^{-1}\sigma(d(q',a'))+\sigma((p,d)),n,\sigma(b)\right)\\&=\left(n^{|u(p,d)|},n^{-1}\sigma(uc(q,a))-n^{-1}\sigma(d(q',a'))+\sigma((p,d)),n^{|\#bv|},n\sigma(\#v^R)+\sigma(b)\right),\end{aligned}\tag{7}$$

where the second component is

$$n\sigma(u)+\sigma(c)+\sigma((p,d))-\sigma(d)+n^{-1}(\sigma((q,a))-\sigma((q',a'))).\tag{8}$$

The component remains integral only if $(q,a)=(q',a')$. Note that at this step, there is no requirement that $d=c$ and, in fact, $d=c$ implies that the correct matrix was applied. So the matrix may remain integral, with incorrect matrices, if $(q,a)=(q',a')$ and $d\neq c$.

Let us consider the above case of multiplying by a wrong matrix with transition to the left a bit further. It turns out that the matrix with $d\neq c$ was used, then the next matrix in the product has to correspond to a move of the head to the right. Indeed, assume that we are in the case (7) with the second component integral and according to (8) it equals to $n\sigma(u)+\sigma(c)+\sigma((p,d))-\sigma(d)=\sigma(u(p,d))+\sigma(c)-\sigma(d)$. Assume further that the vector is multiplied by a vector corresponding to a transition $\delta(r,e)=(p',b',L)$, i.e., by the vector $((n^{-1},-n^{-1}\sigma(f(r,e))+\sigma((p',d')),n,\sigma(b'))$. As previously, the second component becomes

$$n^{-1}(\sigma(u(p,d))+\sigma(c)-\sigma(d))-n^{-1}\sigma(f(r,e))+\sigma((p',d'))$$

and this is an integer for some $(r,e)\in Q\times\Gamma$ if and only if $\sigma(c)+\sigma((p,d))-\sigma(d)-\sigma((r,e))=0$. This is not possible due to the way embedding φ is defined. Denote $\sigma(c)=\varphi(a_i)$, $\sigma((p,d))=\varphi(a_j)$, $\sigma(d)=\varphi(a_k)$ and $\sigma((r,e))=\varphi(a_\ell)$, where i,j,k and ℓ are distinct. (Recall, that if $j=\ell$, then $(p,d)=(r,e)$ and the sum is non-zero as $d\neq c$.) Now,

$$\begin{aligned}&\varphi(a_i)+\varphi(a_j)=\varphi(a_k)+\varphi(a_\ell)\\\Leftrightarrow&\prod_{\substack{x=1\\x\neq i}}^{m}p_x+\prod_{\substack{x=1\\x\neq j}}^{m}p_x=\prod_{\substack{x=1\\x\neq k}}^{m}p_x+\prod_{\substack{x=1\\x\neq \ell}}^{m}p_x\\\Leftrightarrow&\frac{p_1\cdots p_m}{p_ip_jp_kp_\ell}(p_jp_kp_\ell)+\frac{p_1\cdots p_m}{p_ip_jp_kp_\ell}(p_ip_kp_\ell)=\frac{p_1\cdots p_m}{p_ip_jp_kp_\ell}(p_ip_jp_\ell)+\frac{p_1\cdots p_m}{p_ip_jp_kp_\ell}(p_ip_jp_k)\\\Leftrightarrow&\,p_jp_kp_\ell+p_ip_kp_\ell=p_ip_jp_\ell+p_ip_jp_k\\\Leftrightarrow&\,p_kp_\ell(p_i+p_j)=p_ip_j(p_k+p_\ell),\end{aligned}$$

and thus $p_kp_\ell=(p_k+p_\ell)$ and $p_ip_j=(p_i+p_j)$. It is trivial that $p_kp_\ell\neq(p_k+p_\ell)$. Indeed, p_kp_ℓ is odd, being a product of two odd primes, while p_k+p_ℓ is even, being a sum of two odd primes. Alternatively, the equation does not hold as it is a special case of Lemma 1.

So we have showed that in both directions it is possible to get an integral matrix with an incorrect transition matrix, but if an incorrect matrix with transition to the left is applied, then the next matrix in the product has to be to the right to keep the matrix integral.

Next we shall show that if the configuration matrix is multiplied by a wrong matrix, it creates an integer that cannot be removed and thus leads to a rational number if the head is moved beyond this point. This means that, in a way, a multiplication by an incorrect matrix results in a coefficient that cannot be removed. Thus, this makes the tape content on the other side inaccessible.

Before formally proving the above, let us define a useful notation. We say that matrix $M\in\mathbb{N}^{3\times3}$ is *valid* if $M=\gamma(u(q,a),\#v^R)$ for some configuration $(u(q,a),v)$ of the TM $\mathcal{M}$. This means that the words u and v do not contain symbols from the set $Q\times\Gamma$. Note that this does not imply that $(u(q,a),v)$ can be reached by the TM. Analogously, we say that $\psi(u(q,a),\#v^R)$ is valid if it corresponds to a valid configuration of $\mathcal{M}$.

Let us consider the step where the matrix becomes invalid as in our previous considerations. For that assume that $\psi(M)$ is valid and $\psi(MN)$ is invalid for some matrix $N\in M_\delta$, but $\psi(MN)\in\mathbb{N}^4$. Since by our assumption $\psi(MN)\in\mathbb{N}^4$, the component that invalidates the vector is in the second component by our considerations in (7) and (6). Namely, if N simulates a move to the right, then we have the case in (6) with $c=d$, where the element in the second component is

$$\sigma(u(q,a))n-\sigma((q',a'))n+\sigma(b(p,c))$$

for some $(q,a),(q',a'),b,(p,c)\in C$ and $\sigma((q,a))n-\sigma((q',a'))n+\sigma(b(p,c))\neq\sigma(f(r,e))$ for all $f,(r,e)\in C$. Necessarily, $(q,a)\neq(q',a')$, and actually, the condition is that

$$\sigma((q,a))-\sigma((q',a'))+\sigma(b)\neq\sigma(f)$$

for all $f\in\Gamma$.

Similarly, if N simulates a move to the left as in (7), then the element in the second component is as in (8) for some $d,c,(q,a),(q',a'),(p,d)\in C$ and $u\in C^+$. As $\psi(MN)\in\mathbb{N}^4$ holds, then $(q,a)=(q',a')$ which implies that $\sigma(c)-\sigma(d)+\sigma((p,d))\neq\sigma((r,e))$ for all $(r,e)\in C$.

The following lemma shows that once a matrix product becomes invalid it cannot be transformed into a valid one with matrices in M_δ.

Lemma 5. *Let $\psi(M)$ be valid, i.e., $M = \gamma(u(q,a),\#v^R)$ for some configuration. Let $N \in \mathcal{M}_\delta$ such that $\psi(MN) \in \mathbb{N}^4$ is no longer valid. Then there is no sequence $M_{i_1}, M_{i_2}, \dots, M_{i_s} \in \mathcal{M}_\delta$ such that $\psi(MNM_{i_1}M_{i_2}\cdots M_{i_s})$ is valid.*

Proof. Let us assume that there exists a sequence $M_{i_1}, M_{i_2}, \dots, M_{i_s} \in \mathcal{M}_\delta$ such that $\psi(MNM_{i_1}M_{i_2}\cdots M_{i_s})$ is valid and moreover that s is the smallest index such that the product is valid. Assume first that the final matrix, M_{i_s}, corresponds to the head moving to the right

$$\psi(M_{i_s}) = \left(n, -n\sigma((q'',a'')) + \sigma(b''(p'',c'')), n^{-1}, -n^{-1}\sigma(c'')\right)$$

and

$$\psi(MNM_{i_1}M_{i_2}\cdots M_{i_{s-1}}) = \left(n^{|u|}, \sum_{j=0}^{|u|}\alpha_j n^j, n^{|\#v|}, \sigma(\#v^R)\right),$$

where each α_j is a sum of images of letters under σ with both positive and negative coefficients. Note that by our assumption on the minimality of the sequence, at least one α_j is not an image of a letter in Γ under σ for $j = 1, \dots, n$ or α_0 is not an image of a letter in $Q \times \Gamma$. After multiplying $\psi(MNM_{i_1}M_{i_2}\cdots M_{i_{s-1}})$ with the final matrix M_s, we have

$$\begin{aligned}&\left(n^{|u|}, \sum_{j=0}^{|u|}\alpha_j n^j, n^{|\#v|}, \sigma(\#v^R)\right)\cdot\left(n, -n\sigma((q'',a'')) + \sigma(b''(p'',c'')), n^{-1}, -n^{-1}\sigma(c'')\right)\\ &= \left(n^{|ux|+1}, \sum_{j=0}^{|u|}\alpha_j n^{j+1} - n\sigma((q'',a'')) + \sigma(b''(p'',c'')), n^{|\#v|-1}, n^{-1}(\sigma(\#v^R) - \sigma(c''))\right).\end{aligned} \tag{9}$$

Since the product matrix is valid, we observe that, for $j = 2, \dots, |u|$, each $\alpha_j = \sigma(a_{\ell_j})$ for some $a_{\ell_j} \in \Gamma$. The multiplication can only affect α_0 directly and α_1 indirectly via carries.

There are two cases to consider. In the first case α_0 is $\sigma(e)$ for some $e \in Q \times \Gamma$ which implies that α_1 is not an image of a letter. In this case, for the product to be valid,

$$\sigma(e) - \sigma((q'',a'')) + \sigma(b'') = \sigma(f) + yn$$

for some $f \in \Gamma$ and $y \in \mathbb{Z}$ is such that $\alpha_1 + y = \sigma(g)$ for some $g \in \Gamma$, must hold. This is not true due to the definition of n. Indeed, the largest image under φ is at most $\frac{n}{3}$. Hence $y = 0$ and there are no carries. Thus α_1 prevents the result from being a valid matrix.

In the second case, α_0 does not correspond to an image of a letter from $Q \times \Gamma$ under σ. Our goal is to show that there does not exist some $d \in \Gamma$ such that

$$\alpha_0 - \sigma((q'',a'')) + \sigma(b'') = \sigma(d).$$

We prove this as a separate lemma, Lemma 6, after this proof.

The case where matrix M_{i_s} corresponds to the head moving to the left is proven in analogous way. Let M_{i_s} correspond to the head moving to the left, i.e.,

$$\psi(M_{i_s}) = \left(n^{-1}, -n^{-1}\sigma(c''(q'',a'')) + \sigma((p'',c'')), n, \sigma(b'')\right)$$

and

$$\psi(MNM_{i_1}M_{i_2}\cdots M_{i_{s-1}}) = \left(n^{|u|}, \sum_{j=0}^{|u|}\alpha_j n^j, n^{|\#v|}, \sigma(\#v^R)\right),$$

where each α_j is a sum images of letters under σ with both positive and negative coefficients. Again, we multiply the latter matrix by the former and obtain

$$\left(n^{|u|-1}, \sum_{j=0}^{|u|}\alpha_j n^{j-1} - n^{-1}\sigma(c''(q'',a'')) + \sigma((p'',c'')), n^{|\#v|+1}, \sigma(\#(b''v)^R)\right).$$

There are two subcases to consider. Either $\alpha_0 = \sigma((q'',a''))$ or α_0 is not an image of a letter under σ. The first subcase implies that α_j is not an image of a letter under σ for some $j = 1, 2, \dots, |u|$. That is, the analogous considerations as above show that these two coefficients do not become images of some letter under σ.

Hence there is no sequence $M_{i_1}, M_{i_2}, \dots, M_{i_s}$ such that $MNM_{i_1}M_{i_2}\cdots M_{i_s}$ is a valid configuration. □

Lemma 6. *Let $\alpha = \sum_{i=1}^{r} z_i\sigma(s_i)$, where $z_i \in \mathbb{Z}$ and $s_i \in C$. Let $(q,a) \in Q \times \Gamma$ and let $b, c \in \Gamma$. The equation*

$$\alpha - \sigma((q,a)) + \sigma(b) = \sigma(c)$$

does not hold.

Proof. Assume the contrary. That is,

$$\alpha - \sigma((q,a)) + \sigma(b) = \sigma(c) \tag{10}$$

holds. Let $\mathcal{I} \subseteq C$ be the set of all letters that appear in α, and let us partition α into two sets, α^+ and α^-, where the coefficients of letters are positive or negative, respectively. That is,

$$\alpha^+ = \{(a_i, z_i) \mid a_i \in \mathcal{I} \text{ and } z_i \in \mathbb{Z}, z_i > 0\};$$

$$\alpha^- = \{(a_i, z_i) \mid a_i \in \mathcal{I} \text{ and } z_i \in \mathbb{Z}, z_i < 0\}.$$

It is clear that if $(a_i, z_i) \in \alpha^+$, then $(a_i, z) \notin \alpha^-$ (for any $z \in \mathbb{Z}$) and vice versa. Now $\alpha = \sum_{(s,x)\in\alpha^+} x \cdot \sigma(s) + \sum_{(t,y)\in\alpha^-} y \cdot \sigma(t)$. We can rewrite (10) as

$$\sum_{(s,x)\in\alpha^+} x \cdot \sigma(s) + \sigma(b) = -\sum_{(t,y)\in\alpha^-} y \cdot \sigma(t) + \sigma(c) + \sigma((q,a)). \tag{11}$$

Let $\sigma((q,a)) = \prod_{\substack{j=1 \\ j\neq k_1}}^{m} p_j$, $\sigma(b) = \prod_{\substack{j=1 \\ j\neq k_2}}^{m} p_j$ and $\sigma(c) = \prod_{\substack{j=1 \\ j\neq k_1}}^{m} p_j$. Without loss of generality, we assume that $k_1, k_2, k_3 \notin \mathcal{I}$. If this is not the case, then some coefficients of terms of α are different but the same reasoning applies. Let us next define the sets of indexes of letters appearing in α^+ and α^-, together with k_1, k_2 and k_3. That is, $P^+ = \{i_j \mid (s_{i_j}, z_{i_j}) \in \alpha^+\} \cup \{k_1\}$ and $P^- = \{i_j \mid (t_{i_j}, y_{i_j}) \in \alpha^-\} \cup \{k_2, k_3\}$. Equation (11) is equivalent to

$$\begin{aligned} &x_{i_1} \prod_{\substack{j=1 \\ j\neq i_1}}^{m} p_j + x_{i_2} \prod_{\substack{j=1 \\ j\neq i_2}}^{m} p_j + \cdots + x_{i_q} \prod_{\substack{j=1 \\ j\neq i_q}}^{m} p_j + \prod_{\substack{j=1 \\ j\neq k_1}}^{m} p_j \\ &= -y_{\ell_1} \prod_{\substack{j=1 \\ j\neq \ell_1}}^{m} p_j - y_{\ell_2} \prod_{\substack{j=1 \\ j\neq \ell_2}}^{m} p_j - \cdots - y_{\ell_r} \prod_{\substack{j=1 \\ j\neq \ell_r}}^{m} p_j + \prod_{\substack{j=1 \\ j\neq k_2}}^{m} p_j + \prod_{\substack{j=1 \\ j\neq k_3}}^{m} p_j, \end{aligned}$$

where $(s_{i_j}, x_{i_j}) \in \alpha^+$ and $(t_{i_j}, y_{i_j}) \in \alpha^-$. We can divide both sides of the equation by primes not corresponding to indexes of P^+ and P^- as every product contains those primes:

$$\begin{aligned} &x_{i_1} \prod_{\substack{j\in P^+\cup P^- \\ j\neq i_1}}^{m} p_j + x_{i_2} \prod_{\substack{j\in P^+\cup P^- \\ j\neq i_2}}^{m} p_j + \cdots + x_{i_q} \prod_{\substack{j\in P^+\cup P^- \\ j\neq i_q}}^{m} p_j + \prod_{\substack{j\in P^+\cup P^- \\ j\neq k_1}}^{m} p_j \\ &= -y_{\ell_1} \prod_{\substack{j\in P^+\cup P^- \\ j\neq \ell_1}}^{m} p_j - y_{\ell_2} \prod_{\substack{j\in P^+\cup P^- \\ j\neq \ell_2}}^{m} p_j - \cdots - y_{\ell_r} \prod_{\substack{j\in P^+\cup P^- \\ j\neq \ell_r}}^{m} p_j \\ &\quad + \prod_{\substack{j\in P^+\cup P^- \\ j\neq k_2}}^{m} p_j + \prod_{\substack{j\in P^+\cup P^- \\ j\neq k_3}}^{m} p_j. \end{aligned}$$

Next, we take the common factors on both sides. Namely, those primes corresponding to indexes in P^- on the left-hand side and to indexes in P^+ on the right-hand side.

$$\begin{aligned} &p_{\ell_1} p_{\ell_2} \cdots p_{\ell_r} p_{k_2} p_{k_3} \left(x_{i_1} \prod_{\substack{j\in P^+ \\ j\neq i_1}}^{m} p_j + \cdots + x_{i_q} \prod_{\substack{j\in P^+ \\ j\neq i_q}}^{m} p_j + \prod_{\substack{j\in P^+ \\ j\neq k_1}}^{m} p_j \right) \\ &= p_{i_1} p_{x_2} \cdots p_{i_q} p_{k_1} \left(-y_{\ell_1} \prod_{\substack{j\in P^- \\ j\neq \ell_1}}^{m} p_j - \cdots - y_{\ell_r} \prod_{\substack{j\in P^- \\ j\neq \ell_r}}^{m} p_j + \prod_{\substack{j\in P^- \\ j\neq k_2}}^{m} p_j + \prod_{\substack{j\in P^- \\ j\neq k_3}}^{m} p_j \right). \end{aligned}$$

Now the left-hand side is equal to the right-hand side if and only if both

$$p_{i_1}p_{i_2}\cdots p_{i_q}p_{k_1} = x_{i_1}\prod_{\substack{j\in P^+\\ j\neq i_1}}^{m} p_j + \cdots + x_{i_q}\prod_{\substack{j\in P^+\\ j\neq i_q}}^{m} p_j + \prod_{\substack{j\in P^+\\ j\neq k_1}}^{m} p_j \text{ and}$$

$$p_{\ell_1}p_{\ell_2}\cdots p_{\ell_r}p_{k_2}p_{k_3} = -y_{\ell_1}\prod_{\substack{j\in P^-\\ j\neq \ell_1}}^{m} p_j - \cdots - y_{\ell_r}\prod_{\substack{j\in P^-\\ j\neq \ell_r}}^{m} p_j + \prod_{\substack{j\in P^-\\ j\neq k_2}}^{m} p_j + \prod_{\substack{j\in P^-\\ j\neq k_3}}^{m} p_j$$

hold at the same time. But neither equation holds by Lemma 1. □

We have proved that the computation of a TM can be simulated with matrix semigroup with integrality test. We state this as a theorem.

Theorem 7. *Let $\mathcal{M}$ be a TM and let $G = \{M_{(q,a),c} \mid \delta(q,a)$ is defined and $c \in \Gamma\}$, and assume that $u(q,a)v$ is a valid configuration of $\mathcal{M}$. Then $\triangleright(q_0,\star)\triangleleft \vdash^*_{\mathcal{M}} u(q,a)v$ if and only if there exist matrices $M_1,\ldots,M_k \in G$ such that*

$$\gamma(\triangleright(q_0,\star),\#\triangleleft)\cdot M_1\cdot M_2\cdots M_k = \gamma(u(q,a),\#v^R)$$

and

$$\gamma(\triangleright(q_0,\star),\#\triangleleft)\cdot M_1\cdot M_2\cdots M_j \in \mathbb{Z}^{3\times 3}$$

for all $j = 1,\ldots,k$.

Our first undecidability result follows from the halting problem.

Theorem 8. *Let $G \subseteq \mathbb{Q}^{3\times 3}$ be a finite set of matrices and $M \in \mathbb{Z}^{3\times 3}$. Let S be the matrix semigroup generated by G. It is undecidable whether or not there exists a nonempty sequence of matrices $M_{i_1}, M_{i_2},\ldots,M_{i_k} \in S$ such that*

$$M\cdot M_{i_1}\cdot M_{i_2}\cdot\ldots\cdot M_{i_k} = M$$

and

$$M\cdot M_{i_1}\cdot M_{i_2}\cdot\ldots\cdot M_{i_j} \in \mathbb{Z}^{3\times 3}$$

for all $j = 1,\ldots,k$.

Proof. Let G be as in the previous theorem and let $M = \gamma(\triangleright(q_0,\star),\#\triangleleft)$, where $\triangleright(q_0,\star)\triangleleft$ is the initial configuration of Turing machine $\mathcal{M}$ with undecidable halting problem. It is clear, with help of Lemma 5, that G simulates $\mathcal{M}$ and that the two properties of the claim hold if and only if $\mathcal{M}$ halts. □

5. The identity problem for rational matrix semigroups with integrality tests

In this section, we apply Theorem 8 to show that the identity problem is undecidable in this setting. Let us first define the identity problem for a generating set G of a d-dimensional matrix semigroup with entries from $\mathbb{K} \in \{\mathbb{Z},\mathbb{Q},\mathbb{R},\mathbb{C}\}$, i.e., $G \subseteq \mathbb{K}^{d\times d}$.

Problem 9 *(Identity problem). Given a finite set of matrices $G \subseteq \mathbb{K}^{d\times d}$. Does the identity matrix $\boldsymbol{I}_d$ belong to the semigroup $\langle G\rangle$?*

Recall that it is known that the identity problem is decidable for $\mathbb{K} = \mathbb{Z}$ and $d = 2$ [3], undecidable for $\mathbb{K} = \mathbb{H}$ (rational quaternions) and $d = 2$ [2], and $\mathbb{K} = \mathbb{Z}$ and $d = 4$ [5]. We are studying the identity problem for three-dimensional matrices, which is a well-known open problem.

Let us introduce a variant of the identity problem, where there is an additional *integrality test*. Let $G \subseteq \mathbb{Q}^{d\times d}$ be a finite set of rational matrices. Consider $M \in \langle G\rangle$ but $M \notin \mathbb{Z}^{d\times d}$. Then $M \notin \langle G\rangle_{\mathbb{Z}}$.

Problem 10 *(Identity problem with integrality test). Given a finite set of matrices $G \subseteq \mathbb{K}^{d\times d}$. Does the identity matrix $\boldsymbol{I}_d$ belong to $\langle G\rangle_{\mathbb{Z}}$? That is, does there exist a sequence of matrices $M_i \in G$ such that*

$$M_1M_2\cdots M_r = \boldsymbol{I},$$

and

$$M_1M_2\cdots M_j \in \mathbb{Z}^{d\times d}$$

for all $j \in \{1, \ldots, r\}$.

Naturally, we can also ask the other standard matrix semigroup questions for our scenario. But apart from the identity problem, hardly any new results can be derived as most of the problems are undecidable already for integral matrices (i.e., with no integrality tests required).

Theorem 11. *The identity problem with integrality test is undecidable for* $G \subseteq \mathbb{Q}^{3\times 3}$.

Proof. Let G be the set $\mathcal{M}_\delta$ constructed in the previous section together with additional matrices N_1, N_2 used to embed the initial configuration and N_3 to remove the final configuration. More precisely, the matrices are

$$N_1 = \begin{pmatrix} n^2 & 0 & 0 \\ 0 & n^2 & 0 \\ \sigma(\triangleright(q_0,\star)) & \sigma(\#\triangleleft) & n \end{pmatrix},$$

$$N_2 = \begin{pmatrix} n & 0 & 0 \\ 0 & n^{-1} & 0 \\ -\sigma((q_0,\star)) + n^{-1}\sigma(b(p,\triangleleft)) & -n^{-2}\sigma(\triangleleft) & n^{-1}, \end{pmatrix},$$

and

$$N_3 = \begin{pmatrix} n^{-2} & 0 & 0 \\ 0 & n^{-2} & 0 \\ -n^{-2}\sigma(\triangleright(q_0,\star)) & -n^{-2}\sigma(\#\triangleleft) & 1 \end{pmatrix},$$

where $\triangleright(q_0,\star)\triangleleft$ is the initial configuration and the second configuration is $\triangleright b(p,\triangleleft)$.

It is straightforward to see that a non-empty product resulting in the identity matrix has to start with N_1 or with a matrix of form (5). Indeed, these are the only matrices in $\mathbb{Z}^{3\times 3}$. Let us first consider the case where N_1 is the first matrix. We can further observe that multiplying N_1 with any other matrix beside N_1 or N_2 or of form (5) result in a matrix that violates integrality. Indeed, for example, when $M = \begin{pmatrix} n & 0 & 0 \\ 0 & n^{-1} & 0 \\ -n\sigma((q,a))+\sigma(b(p,c)) & -n^{-1}\sigma(c) & 1 \end{pmatrix} \in \mathcal{M}_\delta$. Then

$$N_1 M = \begin{pmatrix} n^2 & 0 & 0 \\ 0 & n^2 & 0 \\ \sigma(\triangleright(q_0,\star)) & \sigma(\#\triangleleft) & n \end{pmatrix} \begin{pmatrix} n & 0 & 0 \\ 0 & n^{-1} & 0 \\ -n\sigma((q,a)) + \sigma(b(p,c)) & -n^{-1}\sigma(c) & 1 \end{pmatrix}$$

$$= \begin{pmatrix} n^3 & 0 & 0 \\ 0 & 1 & 0 \\ n\sigma(\triangleright(q_0,\star)) - n^2\sigma((q,a)) + n\sigma(b(p,c)) & n^{-1}\sigma(\#\triangleleft) - \sigma(c) & n \end{pmatrix}.$$

Normally, $n^{-1}\sigma(\triangleleft)$ would be removed by the correct choice of a matrix with $c = \triangleleft$, but as the bottom right corner is not 1, this does not happen.

In any product resulting in the identity matrix, there must be an equal number of matrices N_1 and N_2 as multiplying by N_2 is the only way to produce a matrix with 1 in the bottom right corner.

Let $M \in \{N_1, N_2\}^*$. M is valid if and only if $M = N_1 N_2$. If M is not valid, then analogously to the proof of Lemma 5, it can be proven that a valid matrix cannot be obtained using matrices from $\mathcal{M}_\delta \cup \{N_1, N_2, N_3\}$.

Assume then that the first matrix is of form (5), i.e., is $\begin{pmatrix} n & 0 & 0 \\ 0 & 1 & 0 \\ -n\sigma((p,\triangleleft))+\sigma(b(q,\triangleleft)) & 0 & 1 \end{pmatrix}$ for some $p, q \in Q$ and $b \in \Gamma$. It is straightforward to see that matrices N_2, N_3 and those corresponding to moving the head to the right cannot be applied as the element $(2,2)$ would become rational. If N_1, a matrix corresponding to moving the head to the left or of the form (5) is applied, then the resulting matrix is not valid and by Lemma 5 cannot be made valid.

Finally, observe that, similarly to how N_1 had to be the first matrix, N_3 has to be the last matrix and, more specifically, can only multiply $\gamma(\triangleright(q_0,\star),\triangleleft)$. The resulting matrix is the identity matrix. The matrix $\gamma(\triangleright(q_0,\star),\triangleleft)$ is in the semigroup if and only if the TM halts. Thus the identity problem is undecidable. □

6. Future work

In the previous sections, we constructed a generator set G that allow us to simulate a Turing machine when the partial products are tested to be integers. It would be interesting to see if it is possible to simulate a TM with a matrix semigroup where the integrality test is not performed after every multiplication. That is, there is a set $Z \subseteq G$ such that the integrality is tested only after a matrix from Z appears in the product. In other words, the model has fewer integrality checks or even a fixed number of integrality checks. This can be achieved by constructing a universal TM with special properties that ensure that a computation consists of some special transitions. These special transitions would then be transformed into matrices with integrality checks. This would require a careful analysis similar to Lemma 5 of "incorrect" simulations to make sure that a valid configuration cannot be obtained.

CRediT authorship contribution statement

Vesa Halava: Writing – review & editing, Writing – original draft, Methodology, Investigation. **Reino Niskanen:** Writing – review & editing, Writing – original draft, Methodology, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Vesa Halava reports financial support was provided by emmy.network foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] Paul Bell, Vesa Halava, Tero Harju, Juhani Karhumäki, Igor Potapov, Matrix equations and Hilbert's tenth problem, Int. J. Algebra Comput. 18 (8) (2008) 1231–1241, https://doi.org/10.1142/S0218196708004925.
[2] Paul Bell, Igor Potapov, Reachability problems in quaternion matrix and rotation semigroups, Inf. Comput. 206 (11) (2008) 1353–1361, https://doi.org/10.1016/j.ic.2008.06.004.
[3] Paul C. Bell, Mika Hirvensalo, Igor Potapov, The identity problem for matrix semigroups in $SL(2,\mathbb{Z})$ is NP-complete, in: Proceedings of Symposium on Discrete Algorithms 2017, 2017.
[4] Paul C. Bell, Igor Potapov, On undecidability bounds for matrix decision problems, Theor. Comput. Sci. 391 (1–2) (2008) 3–13, https://doi.org/10.1016/j.tcs.2007.10.025.
[5] Paul C. Bell, Igor Potapov, On the undecidability of the identity correspondence problem and its applications for word and matrix semigroups, Int. J. Found. Comput. Sci. 21 (6) (2010) 963–978, https://doi.org/10.1142/S0129054110007660.
[6] Michael Blondin, Christoph Haase, Filip Mazowiecki, Mikhail A. Raskin, Affine extensions of integer vector addition systems with states, Log. Methods Comput. Sci. 17 (3) (2021), https://doi.org/10.46298/LMCS-17(3:1)2021.
[7] Michael Blondin, Mikhail A. Raskin, The complexity of reachability in affine vector addition systems with states, Log. Methods Comput. Sci. 17 (3) (2021), https://doi.org/10.46298/LMCS-17(3:3)2021.
[8] Mark Braverman, Termination of integer linear programs, in: Proceedings of Computer Aided Verification 2006, in: LNCS, vol. 4144, Springer, 2006, pp. 372–385.
[9] Volker Claus, Some remarks on PCP(k) and related problems, Bull. Eur. Assoc. Theor. Comput. Sci. 12 (1980) 54–61.
[10] John H. Conway, FRACTRAN: A Simple Universal Programming Language for Arithmetic, Springer New York, New York, NY, 1987, pp. 4–26.
[11] Ruiwen Dong, The identity problem in the special affine group of Z^2, in: Proceedings of LICS 2023, 2023, pp. 1–13.
[12] Ruiwen Dong, The identity problem in nilpotent groups of bounded class, in: SODA 2024, 2024, https://doi.org/10.1137/1.9781611977912.138, in press, arXiv:2208.02164.
[13] Alain Finkel, Jérôme Leroux, How to compose Presburger-accelerations: applications to broadcast protocols, in: Proceedings of FST 2002, in: LNCS, vol. 2556, Springer, 2002, pp. 145–156.
[14] Vesa Halava, Tero Harju, Mika Hirvensalo, Undecidability bounds for integer matrices using Claus instances, Int. J. Found. Comput. Sci. 18 (5) (2007) 931–948, https://doi.org/10.1142/S0129054107005066.
[15] John E. Hopcroft, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, Mass., 1979.
[16] Mehran Hosseini, Joël Ouaknine, James Worrell, Termination of linear loops over the integers, in: Proceedings of ICALP 2019, in: LIPIcs, vol. 132, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 118:1–118:13.
[17] Radu Iosif, Arnaud Sangnier, How hard is it to verify flat affine counter systems with the finite monoid property?, in: Proceedings of ATVA 2016, in: LNCS, vol. 9938, Springer, 2016, pp. 89–105.
[18] Toghrul Karimov, Engel Lefaucheux, Joël Ouaknine, David Purser, Anton Varonka, Markus A. Whiteland, James Worrell, What's decidable about linear loops?, Proc. ACM Program. Lang. 6 (POPL) (2022) 1–25, https://doi.org/10.1145/3498727.
[19] Sang-Ki Ko, Reino Niskanen, Igor Potapov, On the identity problem for the special linear group and the Heisenberg group, in: Proceedings of ICALP 2018, 2018.
[20] Laura Kovács, Anton Varonka, What else is undecidable about loops?, in: Proceedings of RAMiCS 2023, in: LNCS, vol. 13896, Springer, 2023, pp. 176–193.
[21] Zohar Manna, Mathematical Theory of Computation, McGraw-Hill Book Co, 1974.
[22] Marvin L. Minsky, Recursive unsolvability of post's problem of "tag" and other topics in theory of Turing machines, Ann. Math. 74 (3) (1961) 437–455, http://www.jstor.org/stable/1970290.
[23] Turlough Neary, Undecidability in binary tag systems and the post correspondence problem for five pairs of words, in: 32nd International Symposium on Theoretical Aspects of Computer Science, in: LIPIcs. Leibniz Int. Proc. Inform., vol. 30, Schloss Dagstuhl. Leibniz-Zent. Inform, Wadern, 2015, pp. 649–661.
[24] Michael S. Paterson, Unsolvability in 3×3 matrices, Stud. Appl. Math. 49 (1) (1970) 105, https://doi.org/10.1002/sapm1970491105.
[25] Emil Leon Post, A variant of a recursively unsolvable problem, Bull. Am. Math. Soc. 52 (1946) 264–268, https://doi.org/10.1090/S0002-9904-1946-08555-9.
[26] Ashish Tiwari, Termination of linear programs, in: Proceedings of Computer Aided Verification 2004, in: LNCS, vol. 3114, Springer, 2004, pp. 70–82.