Integer Weighted Automata on Infinite Words

Vesa Halava*
Department of Mathematics and Statistics, University of Turku, Turku, Finland
vesa.halava@utu.fi

Tero Harju
Department of Mathematics and Statistics, University of Turku, Turku, Finland
harju@utu.fi

Reino Niskanen
School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, UK
r.niskanen@ljmu.ac.uk

Igor Potapov
Department of Computer Science, University of Liverpool, Liverpool, UK
potapov@liverpool.ac.uk

In this paper we combine two classical generalisations of finite automata (weighted automata and automata on infinite words) into a model of integer weighted automata on infinite words and study the universality and the emptiness problems under zero weight acceptance. We show that the universality problem is undecidable for three-state automata by a direct reduction from the *infinite Post correspondence problem*. We also consider other more general acceptance conditions as well as their complements with respect to the universality and the emptiness problems. Additionally, we build a universal integer weighted automaton with fixed transitions. This automaton has an additional integer input that allows it to simulate any semi-Thue system.

Keywords: Weighted automata; automata on infinite words; undecidability.

1. Introduction

Weighted automata have been extensively studied in recent years [1, 7, 14] and have a wide range of applications, such as speech-recognition [20] and image compression [5]. In weighted automata models a quantitative value (weight) is added to each transition of a finite automaton allowing to enrich the computational model with extra semantics. For example, these weights could be associated with the consumption of resources, time needed for the execution or the probability of the execution. Depending on the semantics (how these weights are used), the acceptance conditions could be defined in various ways, significantly changing the complexity of the weighted automata model.

*Supported by emmy.network foundation under the aegis of the Fondation de Luxembourg.

The acceptance conditions could be defined using various aggregation functions for deterministic or non-deterministic automata that combine weights either on a single path or a set of equivalent paths. For example, consider weighted automata over tropical semirings, i.e., $(\mathbb{Z} \cup \{\infty\}, \min, +, \infty, 0)$. A weight of a word is calculated using the semiring product (i.e., $+$) and the acceptance can be defined using the semiring sum (i.e., min). Furthermore, a word is accepted if its value using the semiring sum is at most a given ν. In [4], the acceptance of infinite words was based on the property that, in the corresponding computation path, a label with the maximal weight is appearing infinitely often in analogy to Büchi automaton. The automata on infinite words have been often motivated for modeling concurrent and communicating systems [23] and more recently infinite words have been used to simulate various processes in computational games [11, 18].

Recently, in [3], the author discussed historical context and the fundamental differences between different weighted automata models based on aggregation functions and acceptance conditions.

In this paper we combine these two fundamental extensions by considering weighted automata on infinite words. The model we consider has weights from the additive group of integers $\mathbb{Z}$ with the zero element 0 and the weights are summed along the path. This model can be seen as a blind one-counter automaton operating on infinite words. Under *the zero acceptance* condition an infinite word w is accepted if there exists a path in the automaton reading w reaching a final state with weight 0 on a finite prefix of w. First we consider two classical decision problems for integer weighted automata on infinite words: the emptiness (checking whether some word is accepted) and the universality problems (checking whether all words are accepted). In contrast to other acceptance conditions with decidable emptiness and universality problems [4], we show that for the zero acceptance, while the emptiness problem is decidable, the universality problem is undecidable.

In this paper we improve the result of [11], where it was shown that the universality problem is undecidable for automata with five states. We prove that the problem remains undecidable for a very minimalistic automaton with only three states. The undecidability result is based on the reductions from the undecidability of the infinite Post correspondence problem (ω PCP) and the state reduction is achieved by proving more restricted form of the ω PCP than in [10]. The idea of proving the undecidability of the universality problem is to construct an automaton that verifies whether a given word is not a solution of a given instance of the infinite Post correspondence problem. This is done by storing the difference of lengths of images in the counter until automaton reaches a symbol that we try to show is different in the images under the morphisms. We store this symbol and let the second morphism catch up after which we verify that the symbols were indeed different. This proof is presented in Section 3.

In Section 4, we investigate variants of zero acceptance in the sense of expanding the condition from the existence of a zero on a path to existence of a weight in a given set. We also modify the acceptance to consider all paths rather than an existence of an accepting path. We call this strong acceptance. This leads to new variants of universality and emptiness problems with emptiness problem being undecidable for strong acceptance for co-zero acceptance.

Finally, in Section 5 we consider a variant of the automaton where all transitions are fixed and the weight given as an input determines whether a word is accepted or not. This automaton can be seen as *universal* in the same sense as a universal Turing machine. That is, for any given semi-Thue system with an input u, the automaton accepts all words if and only if the semi-Thue system terminates on its input.

This paper is an extended version of the conference paper [12] containing additional details of the constructions and full proofs of the results.

2. Notation and definitions

An *infinite word* w over a finite alphabet A is an infinite sequence of letters $w = a_0a_1a_2a_3\cdots$ where $a_i \in A$ is a letter for each $i = 0, 1, 2, \ldots$. We denote the set of all infinite words over A by A^ω. The monoid of all finite words over A is denoted by A^*. The empty word is denoted by ε. A word $u \in A^*$ is a *prefix* of $v \in A^*$, denoted by $u \leq v$, if $v = uw$ for some $w \in A^*$. If u and w are both nonempty, then the prefix u is called *proper*, denoted by $u < v$. A *prefix* of an infinite word $w \in A^\omega$ is a finite word $p \in A^*$ such that $w = pw'$ where $w' \in A^\omega$. This is also denoted by $p \leq w$. The length of a finite word w is denoted by $|w|$. The length of ε is 0. For a word w, we denote by $w(i)$ the ith letter of w, i.e., $w = w(1)w(2)\cdots$. The number of letters a in a word w is denoted by $|w|_a$. The set dA^ω denotes all infinite words starting with d, i.e., $\{dw \mid w \in A^\omega\}$.

Consider a finite (integer) weighted automaton $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$ with the set of states Q, the finite alphabet A, the set of transitions $\sigma \subseteq Q \times A \times Q \times \mathbb{Z}$, the initial state q_0, the set of final states $F \subseteq Q$, and the additive group of integers $\mathbb{Z}$. We write the transitions in the form $t = \langle q, a, p, z\rangle \in \sigma$.

A *configuration* of $\mathcal{A}$ is any triple $(q, u, z) \in Q \times A^* \times \mathbb{Z}$ and it is said to *yield* a configuration $(p, ua, z_1 + z_2)$ if there is a transition $\langle q, a, p, z_2\rangle \in \sigma$.

Let $\pi = t_1t_2t_3\cdots$ be an infinite path of transitions of $\mathcal{A}$ where $t_i = \langle q_{j_i}, a_{k_i}, q_{j_{i+1}}, z_i\rangle$ for $i > 0$ and $q_{j_0} = q_0$. We call such path π a *computation path*. Denote by $\mathcal{R}(\pi)$ the set of all reachable configurations following a path π. That is, for $\pi = \langle q_0, a_{k_0}, q_{j_1}, z_0\rangle \langle q_{j_1}, a_{k_1}, q_{j_2}, z_1\rangle \langle q_{j_2}, a_{k_2}, q_{j_3}, z_2\rangle \cdots$ the set of reachable configurations is

$$\mathcal{R}(\pi) = \{(q_0, \varepsilon, 0), (q_{j_1}, a_{k_0}, z_0), (q_{j_2}, a_{k_0}a_{k_1}, z_0{+}z_1), (q_{j_3}, a_{k_0}a_{k_1}a_{k_2}, z_0{+}z_1{+}z_2), \ldots\}.$$

Further, we denote the path π by π_w if $w = a_{k_0}a_{k_1}a_{k_2}\cdots$. Let $c = (q, u, z) \in \mathcal{R}(\pi)$ for some computation path π. The *weight* of the configuration c is $\gamma(c) = z$. We say that the configuration c *reaches* the state q. If a computation path π reading w is fixed, by the *weight of prefix* $\gamma(p)$ we denote the weight of configuration $(q, p, z) \in \mathcal{R}(\pi)$ where $w = pu$ for some $u \in \mathcal{A}^\omega$.

We are ready to define an acceptance condition. An infinite word $w \in A^\omega$ is accepted by $\mathcal{A}$ if there exists an infinite path π such that at least one configuration

c in $\mathcal{R}(\pi)$ reaches a final state and has weight $\gamma(c) = 0$. The language *accepted by* $\mathcal{A}$ is

$$L(\mathcal{A}) = \{w \in A^\omega \mid \exists \pi_w \in \sigma^\omega\ \exists (q, u, 0) \in \mathcal{R}(\pi_w)\colon q \in F\}\,.$$

We call this *zero acceptance*. We discuss other acceptance conditions in Section 4.

The *universality problem* for weighted automata over infinite words is a problem to decide whether the language accepted by a weighted automaton $\mathcal{A}$ is the set of all infinite words. In other words, whether or not $L(\mathcal{A}) = A^\omega$. The problem of *non-universality* is the complement of the universality problem, that is, whether or not $L(\mathcal{A}) \neq A^\omega$ or, for zero acceptance, whether there exists $w \in A^\omega$ such that for every computation path π reading w and every configuration $c \in \mathcal{R}(\pi)$, $\gamma(c) \neq 0$ holds.

An *instance* of the *Post correspondence problem* (PCP, for short) consists of two morphisms $g, h : A^* \to B^*$ where A and B are alphabets. A nonempty word $w \in A^*$ is a solution of an instance (g, h) if it satisfies $g(w) = h(w)$. We say that $g(w)$ (resp. $h(w)$) is the g-image (resp. h-image) of w. It is well-known that it is undecidable whether or not an instance of the PCP has a solution [19]. The problem remains undecidable for A with $|A| \geq 5$; see [17]. The cardinality of the domain alphabet A is said to be the *size* of the instance.

The *infinite Post correspondence problem*, ω PCP, is a natural extension of the PCP. An infinite word w is a *solution* of an instance (g, h) of the ω PCP if for every finite prefix p of w either $h(p) < g(p)$ or $g(p) < h(p)$ holds. In the ω PCP, it is asked whether or not a given instance has a solution or not. Note that in our formulation prefixes have to be proper. It was proven in [10] that the problem is undecidable for domain alphabets A with $|A| \geq 9$ and in [6] the result was improved to $|A| \geq 8$. A more general formulation of the ω PCP was used in both proofs, namely the prefixes did not have to be proper. However, both constructions rule out non-proper prefixes; see [10, 6] for details.

3. Universality problem for zero acceptance

In this section we improve the result of [11], where it was shown that the universality problem is undecidable for automata with five states. We prove that the problem remains undecidable for automata with three states. The tighter bound relies on deriving new properties about the ω PCP instance. In the proof of undecidability of the universality problem for weighted automata, for each instance (g, h) of the ω PCP, we need to construct a weighted automaton $\mathcal{A}$ such that $L(\mathcal{A}) \neq A^\omega$ if and only if the instance (g, h) has an infinite solution.

Theorem 1. *It is undecidable whether or not $L(\mathcal{A}) = A^\omega$ holds for a 3-state integer weighted automaton $\mathcal{A}$ over its alphabet A.*

Let us first focus on constructing the instance of the ω PCP. In [11], a weighted automaton was constructed from an arbitrary instance of the ω PCP. We reiterate the construction of an instance of the ω PCP found in [10], highlighting the properties that simplify the construction of the automaton.

The ω PCP was shown to be undecidable for instances of size 9 in [10]. The proof uses a reduction from the termination problem of the semi-Thue systems proved to be undecidable for 3-rule semi-Thue systems from [15]. We shall now present the construction from [10].

A *semi-Thue system* is a pair $T = (\Sigma, R)$ consisting of an alphabet $\Sigma = \{a_1, \ldots, a_n\}$ and a relation set $R \subseteq \Sigma^* \times \Sigma^*$, the elements of which are called the *rules* of T. For two words $u, v \in \Sigma^*$, we write $u \rightarrow_T v$, if there are words u_1 and u_2 such that $u = u_1 x u_2$ and $v = u_1 y u_2$ where $(x, y) \in R$. Let $\rightarrow_T^*$ be the reflexive and transitive closure of the relation $\rightarrow_T$. Therefore, we have $u \rightarrow_T^* v$ if and only if either $u = v$ or there exists a finite sequence of words $u = v_1, v_2, \ldots, v_n = v$ such that $v_i \rightarrow_T v_{i+1}$ for each $i = 1, 2, \ldots, n-1$.

Let $w_0 \in \Sigma^*$ be a word, and $T = (\Sigma, R)$ a semi-Thue system. We say that T *terminates* on w_0 if there is no infinite sequence of words $w_1, w_2, \ldots$ such that $w_i \rightarrow_T w_{i+1}$ for all $i \geq 0$. Thus, T terminates on w_0 if all derivations starting from w_0 are of finite length. In the *termination problem* we are given a word w_0 called an *input word*, and a semi-Thue system T and it is asked whether or not T terminates on w_0. As mentioned above the termination problem was proved to be undecidable for 3-rule semi-Thue systems in [15]. Observe that the size of the alphabet Σ plays little role as a binary alphabet is sufficient for the undecidability. Let $T_1 = (\Sigma, R_1)$ be a semi-Thue system where $\Sigma = \{a_1, a_2, \ldots, a_k\}$. Define a coding $\varphi\colon \Sigma^* \rightarrow \{a, b\}^*$ with $\varphi(a_i) = ab^i a$ for all i. Then let $R_1' = \{(\varphi(u), \varphi(v)) \mid (u, v) \in R_1\}$ be a new set of rules, and define $T_1' = (\{a, b\}, R_1')$. It follows immediately that $w \rightarrow_{T_1} w'$ in T_1 if and only if $\varphi(w) \rightarrow_{T_1'} \varphi(w')$ in T_1'. Therefore, if T_1 has the undecidable termination problem, then so does the semi-Thue system T_1'.

Let $T = (\{a, b\}, R)$ be an n-rule semi-Thue system with the undecidable termination problem, and let the rules in T be $t_i = (u_i, v_i)$ for $i = 1, 2, \ldots, n$. Let u be the input word.

The domain alphabet of our instance of the ω PCP is $A = \{a_1, a_2, b_1, b_2, d, \#\} \cup R$, where d is for the beginning and synchronisation and $\#$ is a special separator of the words in a derivation. Note that the rules in R are considered as letters in the alphabet. Define two special morphisms for $x \in A^+$. Morphisms l_x and r_x are called the *desynchronising* morphisms, and defined by $\ell_x(a) = xa$ and $r_x(a) = ax$ for each letter a.

In [10] the following construction was given for a semi-Thue system T and an input word u: Define the morphisms $g, h\colon A^* \rightarrow \{a, b, d, \#\}^*$ by (recall that for

$t_i \in R$, we denoted $t_i = (u_i, v_i)$)

$$\begin{array}{lll} h(a_1) = dad, & g(a_1) = add, & \\ h(b_1) = dbd, & g(b_1) = bdd, & \\ h(a_2) = dda, & g(a_2) = add, & \\ h(b_2) = ddb, & g(b_2) = bdd, & \\ h(t_i) = d^{-1}\ell_{dd}(v_i), & g(t_i) = r_{dd}(u_i), & \text{for } t_i \in R, \\ h(d) = \ell_{dd}(u)dd\#d, & g(d) = dd, & \\ h(\#) = dd\#d, & g(\#) = \#dd. & \end{array} \tag{1}$$

Note, that $d^{-1}\ell_{dd}(\cdot)$ means that the image starts with a single d. In the special case, where $v_i = \varepsilon$, we define $h(t_i) = d$.

Let us illustrate the construction with an example.

Example 2. *Let $T = (\{a, b\}, \{t\})$ be a semi-Thue system with the input word $u = a$ and the rule $t = (a, aa)$. The corresponding instance of the ω PCP over alphabet $\{a_1, b_1, a_2, b_2, t, d, \#\}$ is*

$$\begin{array}{llll} h(a_1) = dad, & g(a_1) = add, & h(b_1) = dbd, & g(b_1) = bdd, \\ h(a_2) = dda, & g(a_2) = add, & h(b_2) = ddb, & g(b_2) = bdd, \\ h(t) = dadda, & g(t) = add, & h(d) = ddadd\#d, & g(d) = dd, \\ h(\#) = dd\#d, & g(\#) = \#dd. & & \end{array}$$

It was proved in [10] that the following property holds for this construction:

Property 3. *Let (g, h) be an instance of the ω PCP defined in (1). Each infinite solution of (g, h) is of the form*

$$dw_1\#w_2\#w_3\#\cdots, \quad \text{where } w_j = x_j t_{i_j} y_j \tag{2}$$

for some $t_{i_j} \in R$, $x_j \in \{a_1, b_1\}^*$ and $y_j \in \{a_2, b_2\}^*$ for all j.

Indeed, the image $g(w)$ is always of the form $r_{d^2}(v)$, and therefore, by the form of h, between two separators # there must occur exactly one letter $t \in R$. Also, the separator # must be followed by words in $\{a_1, b_1\}^*$ before the next occurrence of a letter $t \in R$. By the form of $h(t)$ the following words before the next separator must be in $\{a_2, b_2\}^*$. The form (2) follows when we observe that there must be infinitely many separators # in each infinite solution. Indeed, all solutions begin with the letter d, and there is one occurrence of # in $h(d)$ and no occurrences of # in $g(d)$. Later each occurrence of # is produced from # by both g and h. Therefore there are infinitely many letters # in each infinite solution.

Property 4. *Let (g, h) be as in (1). In a solution, for any finite prefix, the g-image cannot be longer than the h-image.*

Assume towards a contradiction that w is a solution and p its prefix such that $|g(p)| \geq |h(p)|$. Observe that the word $h(p)$ has more occurrences of # than $g(p)$. Indeed, $|h(p)|_\# = |g(p)|_\# + |p|_d \geq |g(p)|_\# + 1$. Therefore, $h(p) \not< g(p)$.

Property 5. *Let (g, h) be as in (1). In a word w beginning with the letter d, the first position where $h(w)$ and $g(w)$ differ (called an* error*) is reached in $h(w)$ at least one letter (of w) earlier than it is reached in $g(w)$.*

Let us restate this property and prove it.

Lemma 6. *Let (g, h) be as in (1) and assume that $w \in dA^\omega$ is not an infinite solution of the instance (g, h). Let $p = u'c$, where $c \in A$, be the shortest prefix of w such that $g(p) \not< h(p)$. Let r be the least position such that $h(p)(r) \neq g(p)(r)$. Then $r \leq |h(u')|$.*

Proof. Note first that $|p| \geq 2$ by the definition of $h(d)$ and $g(d)$. By the minimality of p, we have $g(u') \leq h(u')$.

Let v be the longest prefix of u' of the form in (2), that is,

$$v = dw_1\#w_2\#w_3\# \cdots w_n\#,$$

where

$$w_j = x_j t_{i_j} y_j$$

for some $t_{i_j} \in R, x_j \in \{a_1, b_1\}^*$ and $y_j \in \{a_2, b_2\}^*$ for all $j = 1, 2, \ldots, n$. Now

$$g(v) = v_1\#v_2\# \cdots v_n\#dd$$

and

$$h(v) = v_1\#v_2\# \cdots v_n\#ddv_{n+1}\#d$$

where $v_i \in (B \setminus \{\#\})^+$ for $i = 1, 2, \ldots, n + 1$. More precisely,

$$\begin{aligned} v_1 &= dd\, g(w_1) = l_{d^2}(u)dd, \\ v_j &= dd\, g(w_j) = d\, h(w_{j-1})\, dd \quad \text{for } j = 2, \ldots, n \quad \text{and} \\ v_{n+1} &= d\, h(w_n)\, dd. \end{aligned}$$

We prove that the error must appear within $v_{n+1}\#d$ in the image $h(v)$ which proves the claim. Assume to the contrary that the error is not within $v_{n+1}\#$, i.e., that the error appears after the last occurrence of $\#$ in the h-image of v. To cover $v_{n+1}\#d$ there must exist w_{n+1} such that $w_{n+1} = x_{n+1}t_{i_{n+1}}y_{n+1}$ (where $t_{i_{n+1}} \in R$, $x_{n+1} \in \{a_1, b_1\}^*$ and $y_{n+1} \in \{a_2, b_2\}^*$). By the maximality of v, $vw_{n+1}\#$ is not a prefix of u', and therefore, $u' = vw_{n+1}$ and $c = \#$. But then $g(p) \leq h(p)$; a contradiction. □

The two properties are illustrated in Figure 1. In the next theorem, we restate and sharpen the result of [10] by improving the undecidability claim of the ω PCP.

Theorem 7. *Let (g, h) be an instance of the ω PCP defined as in (1) that satisfies Properties 3, 4, 5. It is undecidable whether a solution to (g, h) exists.*

Next, we construct the weighted automaton based on the undecidable instance of the ω PCP of Theorem 7. This will allow us to prove Theorem 1.

Let (g, h) be a fixed instance of the ω PCP as defined in (1). Then $g, h\colon A^* \to B^*$ where $A = \{a_1, a_2, b_1, b_2, d, \#, t_1, \dots, t_n\}$ and $B = \{a, b, d, \#\}$. Recall, that letters t_i correspond to the rules of an n-rule semi-Thue system. We construct a weighted automaton $\mathcal{A} = (Q, A, \sigma, q_0, F, \mathbb{Z})$, where $Q = \{q_0, q_1, q_2\}$ and $F = \{q_2\}$, corresponding to the instance (g, h) such that an infinite word $w \in A^\omega$ is accepted by $\mathcal{A}$ iff for some finite prefix p of w, $g(p) \not< h(p)$. Moreover, by Property 5, such p exists for all infinite words except for the solutions of the instance (g, h). We call the verification that $g(p) \not< h(p)$, for a prefix p, *error checking*.

Let us begin with the transitions of $\mathcal{A}$. The automaton is depicted in Figure 2. Let us define $\tau\colon B \to \{1, 2, 3, 4\}$ as $\tau(a) = 1$, $\tau(b) = 2$, $\tau(d) = 3$ and $\tau(\#) = 4$. First for each $c \in A$, let $\langle q_0, c, q_0, 5(|h(c)| - |g(c)|)\rangle$, $\langle q_1, c, q_1, 5(-|g(c)|)\rangle$, $\langle q_2, c, q_2, 0\rangle$ be in σ and for all $c' \in A \setminus \{d\}$, let $\langle q_0, c', q_2, 0\rangle \in \sigma$. For the error checking we need the following transitions for all letters $c \in A$: Let $h(c) = b_{j_1} b_{j_2} \cdots b_{j_{n_1}}$, where $b_{j_k} \in B$, for each index $1 \le k \le n_1$. Then let, for each $k = 1, \dots, n_1$,

$$\langle q_0, c, q_1, 5(k - |g(c)|) + \tau(b_{j_k})\rangle \in \sigma. \tag{3}$$

Let $g(c) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$, where $b_{i_\ell} \in B$, for each index $1 \le \ell \le n_2$. For each $\ell = 1, \dots, n_2$ and letter $b_e \in B$ such that $b_{i_\ell} \neq b_e$, let

$$\langle q_1, c, q_2, -5\ell - \tau(b_e)\rangle \in \sigma. \tag{4}$$

We call the transitions in (3) *error guessing transitions* and in (4) *error verifying transitions*.

Before proving the main result, let us illustrate the construction with an example.

Example 8. *Consider the* ω PCP *of Example 2. We will not present the whole weighted automaton as even for such small semi-Thue system and* ω PCP *instance, the automaton has 115 transitions. Recall that* $\tau(a) = 1$ *and* $\tau(\#) = 4$.

Let $w \in da_1a_1A^\omega$*. It is easy to see that this is not a solution of the* ω PCP*. Indeed,* $g(da_1a_1) = ddaddadd \not< ddadd\#ddaddad = h(da_1a_1)$*. First we show that guessing that the error is in the* g*-image of the first* a_1 *does not lead to an accepting computation, since* $g(da_1) = ddadd < ddadd\#ddad = h(da_1)$*, then we show that there is an accepting computation when we guess that the error is in the* g*-image of the second* a_1*.*

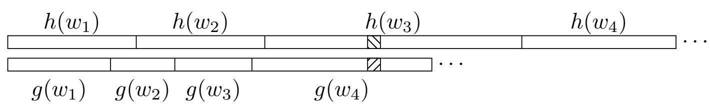

Fig. 1. An illustration of a solution candidate to the instance of the ω PCP satisfying Properties 4 and 5. Here, ⧅ represent the first letter of $h(w_1w_2w_3w_4\cdots)$ that is compared to a letter of $g(w_1w_2w_3w_4\cdots)$ which is represented by ⧄.

If we guess that the error will occur in the third position of the images, we need to store the symbol a and position 3 when reading d. This is done by using the transition

$$\langle q_0, d, q_1, 5(k - |g(d)|) + \tau(b_{j_k})\rangle = \langle q_0, d, q_1, 5(3-2)+1\rangle = \langle q_0, d, q_1, 6\rangle .$$

Then we have to verify the error using a transition

$$\langle q_1, a_1, q_2, -5\ell - \tau(b_e)\rangle = \langle q_1, a_1, q_2, -5 - \tau(b_e)\rangle ,$$

where $\tau(b_e) = 2, 3, 4$. *After these two transitions the weight is at most* -1 *and thus* w *is not accepted with this path.*

On the other hand, if we guess that the error will occur in the 6th position of the h*-image, we use the transition*

$$\langle q_0, d, q_1, 5(6-2)+4\rangle = \langle q_0, d, q_1, 24\rangle .$$

Then the first a_1 *is read in the state* q_1 *with the transition* $\langle q_1, a_1, q_1, -5\cdot 3\rangle$ *after which the weight is* 9. *Then we verify the error with the transition* $\langle q_1, a_1, q_2, -5-4\rangle$. *After these three transitions the weight is 0 and the computation has reached the state* q_2. *Thus* w *is accepted by the automaton.*

The next lemma shows a key property about words accepted by $\mathcal{A}$. The proof relies on analysis of weights along computation paths.

Lemma 9. *A word* $w \in A^\omega$ *is accepted by* $\mathcal{A}$ *if and only if* w *is not a solution of the instance* (g,h) *of the* $\omega\,$PCP *as defined in* (1).

Proof. Let $w = c_1c_2\cdots$ with $c_i \in A$ for all $i = 1, 2, \ldots$. Assume first that w is not a solution of the instance (g,h) of the form (1). Now either the first letter is d or not. In the latter case w is accepted by a path starting with $\langle q_0, c_1, q_2, 0\rangle$.

In the first case by Properties 4 and 5, there exists a prefix p of w such that $g(p) \not< h(p)$ and the first error position is reached in the h-image of w at least one letter (of w) before it is reached in the g-image of w. Let r be the minimal position for which $h(w)(r) \neq g(w)(r)$. In other words for $p = dc_2\cdots c_n$, there exists a position $t < n$ such that $r = |h(dc_2\cdots c_{t-1})| + k$ where $k \leq |h(c_t)|$, and $r = |g(dc_2\cdots c_{n-1})| + \ell$ where $\ell \leq |g(c_n)|$. Denote $h(w)(r) = b_{j_k}$. It is the kth letter of the image $h(c_t)$, and $g(w)(r)$ is the ℓth letter of the image $g(c_n)$. By the choice of r, these letters are nonequal.

Now, w is accepted in the state q_2 with the following path: First, the prefix $dc_2\cdots c_{t-1}$ is read in the state q_0 with the weight

$$5(|h(dc_2\cdots c_{t-1})| - |g(dc_2\cdots c_{t-1})|).$$

When reading c_t, the automaton uses the error guessing transition

$$\langle q_0, c_t, q_1, 5(k - |g(c_t)|) + \tau(b_{j_k})\rangle ,$$

and then the word $c_{t+1}\cdots c_{n-1}$ is read in the state q_1 with the weight

$$5(-|g(c_{t+1}\cdots c_{n-1})|).$$

Finally, while reading the letter c_n, the state q_2 is reached by the error verifying edge $\langle q_1, c_n, q_2, -5\ell - \tau(b_{j_k})\rangle$. Note that such an error verifying edge exists as the ℓth letter in $g(c_n)$ is not equal to the kth letter, b_{j_k}, of $h(c_t)$. Naturally, after reaching q_2 the weight does not change as for all letters there are only transitions with zero weight. Now the weight of the above path is

$$\begin{aligned}\gamma(p) &= 5(|h(dc_2 \cdots c_{t-1})| - |g(dc_2 \cdots c_{t-1})|) + 5(k - |g(c_t)|) + \tau(b_{j_k}) \\ &\quad + 5(-|g(c_{t+1} \cdots c_{n-1})|) - s\ell - \tau(b_{j_k}) \\ &= 5\big(|h(dc_2 \cdots c_{t-1})| + k - |g(dc_2 \cdots c_{n-1})| - \ell\big) = 5(r - r) = 0.\end{aligned}$$

Therefore, w is accepted, as claimed.

Next we prove that if w is a solution of the instance (g, h), then it is not accepted by $\mathcal{A}$. Assume contrary to the claim that w is a solution and there is an accepting path of w in $\mathcal{A}$. As stated in (2), we have $w = dw_1\#w_2\#w_3\# \cdots$, where $w_j = x_j t_{i_j} y_j$ for some $t_{i_j} \in R, x_j \in \{a_1, b_1\}^*$ and $y_j \in \{a_2, b_2\}^*$ for all j.

There are two possible computation paths for w. It can be accepted by a path visiting q_1 or not. In the second case, the part of w that is read in q_0 has to have equal lengths under g and h. By Property 4, w is not a solution of the instance of the ω PCP.

If the computation path visits q_1, then we can partition w into different parts according to the state-transition of the automaton. That is, w has a prefix $p = uxvy$, where $x, y \in A$, such that $u \in A^*$ is read in the state q_0 and $v \in A^*$ in the state q_1, and when reading the letter x the path reaches q_1 and when reading the letter y the path reaches q_2. The weight $\gamma(p)$ of p is now

$$\begin{aligned}\gamma(p) &= 5(|h(u)| - |g(u)|) + 5(k - |g(x)|) + \tau(b_{j_k}) + 5(-|g(v)|) + (-5\ell - \tau(b_e)) \\ &= 5\big(|h(u)| + k - |g(uxv)| - \ell\big) + \tau(b_{j_k}) - \tau(b_e)\end{aligned}$$

where $h(x)(k) = b_{j_k}$ and $g(y)(\ell) \neq b_e$. As $\tau(b_{j_k}) < 5$ and $\tau(b_e) < 5$, we have that $\gamma(p) = 0$ if and only if $|h(u)| + k = |g(uxv)| + \ell$ and $\tau(b_{j_k}) = \tau(b_e)$. Denote $r = |h(u)| + k$. Now, $\gamma(p) = 0$ if and only if $h(w)(r) = b_{j_k} \neq b_c = g(w)(r)$, which is a contradiction since w was assumed to be a solution of (g, h). □

We are ready to prove the main theorem. By Lemma 9, a word $w \in A^\omega$ is accepted by the above constructed integer weighted automaton $\mathcal{A}$ iff w is not a solution of a given instance (g, h) of the ω PCP. By Theorem 7, it is undecidable whether or not the instance (g, h) has a solution or not. This proves Theorem 1.

Note that the number of the letters in the alphabet A in Theorem 1 is small. Indeed, $|A| = 9$ by the construction in (1). The number of transitions on the other hand is huge. The number of error guessing and verifying transitions is dependent on the lengths of the images. One of the rules consists of encoding of all the rules of the 83-rule semi-Thue system with an undecidable termination problem. Its image is several hundreds of thousands letters long.

Next, we consider the universality problem for automata, where all states are final. That is, we consider an acceptance condition, where a word is accepted

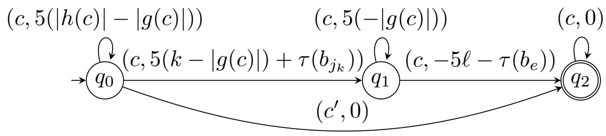

Fig. 2. The weighted automaton $\mathcal{A}$. In the figure $c \in A$ and $c' \in A \setminus \{d\}$.

based solely on weight. Formally, $\mathcal{L}(\mathcal{A}) = \{w \in A^\omega \mid \exists \pi_w \in \sigma^\omega \; \exists (q, u, 0) \in \mathcal{R}(\pi_w)\}$. Relaxing the state reachability condition on the previously defined automaton leads to new accepting paths. For example, an infinite word starting with a_1 is accepted in the state q_0 since $|h(a_1)| - |g(a_1)| = 0$. On the other hand this word can also be accepted in q_2 with the transition $\langle q_0, a_1, q_2, 0\rangle$. So we need to show that no new words are accepted in states q_0 and q_1.

Corollary 10. *It is undecidable whether or not $\mathcal{L}(\mathcal{A}) = A^\omega$ holds for a 3-state integer weighted automaton $\mathcal{A}$ over its alphabet A.*

Proof. We show that languages accepted by the previously constructed automaton are the same under both acceptance conditions. It is clear that $L(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A})$. For the other inclusion, we note that an infinite word cannot be accepted in q_1 as $\tau(b_{j_k}) \in \{1, 2, 3, 4\}$ is added to the counter when entering q_1. It is enough to show that a word accepted in q_0 can also be accepted in q_2.

Let $w \in A^\omega$ that is accepted in the state q_0. That is, there is a prefix p that is read in q_0 and $\gamma(p) = \gamma(5(|h(p)| - |g(p)|)) = 0$. Now w is accepted in q_2 when the next letter, say c, is read using the transition $\langle q_0, c, q_2, 0\rangle$. □

It is also natural to consider the emptiness problem for weighted automata. That is, whether for a given weighted automaton $\mathcal{A}$, $L(\mathcal{A}) = \emptyset$. In contrast to the result of Theorem 1, the emptiness problem is decidable.

Theorem 11. *It is decidable whether or not $L(\mathcal{A}) = \emptyset$ holds for an integer weighted automaton $\mathcal{A}$ over its alphabet A.*

Proof. Let $\mathcal{A}$ be a weighted automaton on infinite words. Consider it as a weighted automaton on finite words, $\mathcal{B}$, defined in [9]. Clearly $L(\mathcal{A}) = \emptyset$ if and only if $L(\mathcal{B}) = \emptyset$. Indeed, an infinite word w is accepted by $\mathcal{A}$ if and only if there is a finite prefix u of w with $\gamma(u) = 0$. This u is accepted by $\mathcal{B}$. On the other hand, if some finite word u is accepted by $\mathcal{B}$ then an infinite word starting with u is accepted by $\mathcal{A}$. In [8] it was shown that languages defined by weighted automata on finite words are context-free languages. It is well-known that emptiness is decidable for context-free languages. □

Corollary 12. *For weighted automata $\mathcal{A}$ and $\mathcal{B}$ the following problems are undecidable:*

(i) Language equality: Whether $L(\mathcal{A}) = L(\mathcal{B})$.
(ii) Language inclusion: Whether $L(\mathcal{B}) \subset L(\mathcal{A})$.
(iii) Language union: Whether $L(\mathcal{A}) \cup L(\mathcal{B}) = A^\omega$.
(iv) Language regularity: Whether $L(\mathcal{A})$ is recognised by a Büchi automaton.

Proof. Let $\mathcal{A}$ be the automaton from Theorem 1.

(i) Let $\mathcal{B}$ be a weighted automaton with one state q and transitions $\langle q, c, q, 0\rangle$ for all $c \in A$. Clearly $L(\mathcal{B}) = A^\omega$. Now the automata $\mathcal{A}$ and $\mathcal{B}$ accept the same language if and only if $\mathcal{A}$ accepts A^ω.
(ii) Let $\mathcal{B} = (\{q_0, q_1\}, A, \sigma, q_0, \{q_1\}, \mathbb{Z})$ where

$$\sigma = \{\langle q_0, d, q_1, 0\rangle, \langle q_1, d, q_1, 1\rangle\} \cup \{\langle q_0, c, q_1, 1\rangle, \langle q_1, c, q_1, 0\rangle \mid c \in A \setminus \{d\}\}.$$

Let w be a solution of an instance (g, h) of ω PCP. It is accepted by $\mathcal{B}$ but not by $\mathcal{A}$. Now if the instance (g, h) of ω PCP has a solution, then $L(\mathcal{B}) \not\subset L(\mathcal{A})$. On other hand if the instance (g, h) of ω PCP does not have a solution, then $L(\mathcal{B}) \subset L(\mathcal{A})$.
(iii) Let $\mathcal{B}$ be an automaton accepting the empty language. Now $L(\mathcal{A}) \cup L(\mathcal{B}) = A^\omega$ holds if and only if $L(\mathcal{A}) = A^\omega$.
(iv) The claim follows as A^ω is an omega-regular language. □

Corollary 13. *It is undecidable whether $L(\mathcal{A}) = L(\mathcal{A}')$ for two weighted automata $\mathcal{A}, \mathcal{A}'$ such that there exists a bijective mapping from edges of $\mathcal{A}$ to edges of $\mathcal{A}'$.*

Proof. Let $\mathcal{A}$ be the automaton of Theorem 1. Consider an automaton $\mathcal{A}'$ with a single state q'. Let $T_c = \{\langle q, c, p, z\rangle \in \sigma\}$ be the set of all transitions in $\mathcal{A}$ reading a letter c. Denote by n_c the size of the set T_c. Now, in $\mathcal{A}'$ for each letter c we add a transition $\langle q', c, q', -i\rangle$ where $i = 0, \ldots, n_c - 1$. Clearly $L(\mathcal{A}') = A^\omega$. There is an obvious bijection between transitions of $\mathcal{A}$ and $\mathcal{A}'$. Now the automata accept the same language if and only if $\mathcal{A}$ accepts A^ω. □

4. Different acceptance conditions

We will examine another non-deterministic acceptance that we call *strong acceptance*. It is informally defined as "a word is accepted iff every path in the machine according to this word satisfies the property φ". We will use notation $\mathbb{Z}$-WA($\exists\, \varphi$) for integer weighted finite automata on infinite words with an acceptance condition φ. Analogously, $\mathbb{Z}$-WA($\forall\, \varphi$) denotes the strong acceptance.

In [11], integer weighted automata on infinite words were introduced and it was proven that the universality problem is undecidable for *zero acceptance*. In this section, we investigate other acceptance properties and their effect on the decidability of language theoretic problems. The two problems we study are the universality and the emptiness problems. In the universality problem we are asked whether every word is accepted and in the emptiness problem whether at least one infinite word is

Acceptance (∃):	$w \in L(A) \iff \exists \pi_w \in \sigma^\omega\ \varphi(\pi_w)$
Strong acceptance (∀):	$w \in L(A) \iff \forall \pi_w \in \sigma^\omega\ \varphi(\pi_w)$
Zero acceptance (Z):	$\varphi(\pi_w) = \exists (q, u, z) \in \mathcal{R}(\pi_w)\ (q \in F \wedge z = 0)$
Co-zero acceptance (¬Z):	$\varphi(\pi_w) = \forall (q, u, z) \in \mathcal{R}(\pi_w)\ (q \notin F \vee z \neq 0)$
Set acceptance (S):	$\varphi(\pi_w) = \exists (q, u, z) \in \mathcal{R}(\pi_w)\ (q \in F \wedge z \in S)$
Co-set acceptance (¬S):	$\varphi(\pi_w) = \forall (q, u, z) \in \mathcal{R}(\pi_w)\ (q \notin F \vee z \notin S)$

Table 1. Different acceptances and acceptance conditions. Note that $S \subseteq \mathbb{Z}$.

accepted. That is, we are interested in the universality and the emptiness problems for $\mathbb{Z}$-WA($\exists\,\varphi$) and $\mathbb{Z}$-WA($\forall\,\varphi$) for various φ. We present different acceptances and acceptance conditions in Table 1.

Let us discuss these acceptance properties next. In the already mentioned zero acceptance, a word w is accepted iff on a computation path reading w there is an intermediate configuration where the state is final and the weight is zero. We denote this property by Z. The complementary property, co-zero acceptance, is defined in the obvious way. That is, a word w is accepted iff on a computation path reading w, all configurations are either not in a final state or do not have weight zero. This property is denoted by ¬Z.

It is straightforward to see that since the universality problem is undecidable for $\mathbb{Z}$-WA(∃ Z) as proven in [11] and Theorem 1, the emptiness problem is undecidable for $\mathbb{Z}$-WA(∀ ¬Z). Indeed, the universality and the emptiness problems are complementary and so are zero acceptance and strong co-zero acceptance. Next we show the decidability of the other combinations. That is, that the emptiness problem is decidable for $\mathbb{Z}$-WA(∃ Z), $\mathbb{Z}$-WA(∃ ¬Z), $\mathbb{Z}$-WA(∀ Z), $\mathbb{Z}$-WA(∀ ¬Z) and that the universality problem is decidable for $\mathbb{Z}$-WA(∃ ¬Z), $\mathbb{Z}$-WA(∀ Z), $\mathbb{Z}$-WA(∀ ¬Z).

Theorem 14. *Let $\mathcal{A}$ be a $\mathbb{Z}$-WA(∃ ¬Z) or $\mathbb{Z}$-WA(∀ Z). It is decidable whether $L(\mathcal{A}) = \emptyset$ holds.*

Proof. Let us consider $\mathbb{Z}$-WA(∃ ¬Z) as the proof for the other class is analogous. Let $\mathcal{A}$ be a $\mathbb{Z}$-WA(∃ ¬Z). Now the question can be restated as

$$\exists w \in A^\omega\ \exists \pi_w \in \sigma^\omega\ \forall (q, u, z) \in \mathcal{R}(\pi_w)(q \notin F \vee z \neq 0).$$

As we are interested in an existence of such path, we can ignore the letters. Indeed, if we find a path, there is a corresponding word that is accepted and hence $L(\mathcal{A})$ is not empty. That is, $\mathcal{A}$ can be considered as a $\mathbb{Z}$-VASS for which the reachability relation is effectively semi-linear [2]. Hence, the property can be expressed as a sentence in Presburger arithmetics, which is a decidable logic. □

Corollary 15. *Let $\mathcal{A}$ be a $\mathbb{Z}$-WA(∀ ¬Z), $\mathbb{Z}$-WA(∀ Z)or $\mathbb{Z}$-WA(∃ ¬Z). It is decidable whether $L(\mathcal{A}) = A^\omega$ holds, where $\mathcal{A}$ is over alphabet A.*

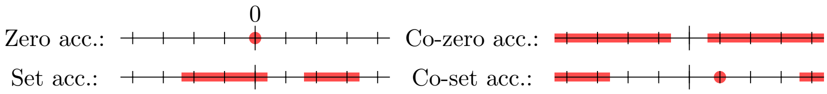

Fig. 3. An illustration of different acceptance conditions. In red are weights that are to be reached in an accepting path.

Proof. The universality problem for $\mathbb{Z}$-WA($\forall\neg$Z) is dual to the emptiness problem for $\mathbb{Z}$-WA($\exists$Z), which is decidable by Theorem 11. Analogously, the universality problems for $\mathbb{Z}$-WA($\forall$Z) and $\mathbb{Z}$-WA($\exists\neg$Z) are dual to the emptiness problems for $\mathbb{Z}$-WA($\exists\neg$Z) and $\mathbb{Z}$-WA($\forall$Z), respectively, which are decidable. □

In both zero acceptance and co-zero acceptance, the integer 0 seems to play an important role. This is not true. One can alter some of the transitions to have acceptance for any fixed integer. For example, by introducing a new initial state q_0' and transitions $\langle q_0', a, q, z+1\rangle$ for every transition $\langle q_0, a, q, z\rangle \in \sigma$. Furthermore, one can multiply all the weights in the transitions by some constant N to ensure that in the interval $\{0, \ldots, N-1\}$ only 0 is actually reachable. This leads to an acceptance condition for intervals with the same decidability statuses. Note that due to the construction, no weights $1k, \ldots, (N-1)k$ are reachable for any integer k. This leads us to an observation that we can consider finite or infinite sets and retain the decidability statuses. For example, multiplying all the weights in the transitions by an even N, we can specify an acceptance condition where "a word is accepted iff upon reaching a final state, weight is either in interval $\{0, \ldots, \frac{N}{2}-1\}$ or interval $\{\frac{N}{2}+1, N-1\}$". Let us call this acceptance condition *set acceptance*. Figure 3 illustrates the differences between zero, co-zero, set and co-set acceptances with respect to weights that are reached on accepting paths.

Let $S \subseteq \mathbb{Z}$. In the set acceptance, a word w is accepted iff on a computation path reading w there is an intermediate configuration where the state is final and the weight is in S. For the dual co-set acceptance, a word w is accepted iff on a computation path reading w all intermediate configurations are either not in a final state or the weight is not in S.

It is straightforward to see that the undecidability of the universality problem follows from the undecidability of the universality problem for zero acceptance. Likewise, the emptiness problem is decidable due to the decidability of the emptiness problem for zero acceptance. The other decidability results for variants of set acceptance can be proven *mutatis mutandis*. That is, in the emptiness problems, the automaton can be considered as a Z-VASS with effectively semi-linear reachability relation. While the decidability of the universality problems for $\mathbb{Z}$-WA($\exists\neg$S), $\mathbb{Z}$-WA($\forall$S) and $\mathbb{Z}$-WA($\forall\neg$S) follows as they are complementary to the emptiness problems for $\mathbb{Z}$-WA($\forall$S), $\mathbb{Z}$-WA($\exists\neg$S) and $\mathbb{Z}$-WA($\exists$S), respectively.

This is summarised in Table 2 where the decidability statuses of the universality

Acceptance	Universality	Emptiness
Zero	Undecid.	Decid.
Co-zero	Decid.	Decid.
Set	Undecid.	Decid.
Co-set	Decid.	Decid.

Strong acc.	Universality	Emptiness
Zero	Decid.	Decid.
Co-zero	Decid.	Undecid.
Set	Decid.	Decid.
Co-set	Decid.	Undecid.

Table 2. Decidability status of the universality and emptiness problems under different acceptances The result in blue implies other undecidability results.

and the emptiness problems for the different acceptance conditions.

Corollary 16. *The following hold:*

- *The universality problem is decidable for* $\mathbb{Z}$-*WA*$(\exists\,\neg\mathrm{S})$, $\mathbb{Z}$-*WA*$(\forall\,\mathrm{S})$ *and* $\mathbb{Z}$-*WA*$(\forall\,\neg\mathrm{S})$ *and undecidable for* $\mathbb{Z}$-*WA*$(\exists\,\mathrm{S})$.
- *The emptiness problem is decidable for* $\mathbb{Z}$-*WA*$(\exists\,\mathrm{S})$, $\mathbb{Z}$-*WA*$(\exists\,\neg\mathrm{S})$ *and* $\mathbb{Z}$-*WA*$(\forall\,\mathrm{S})$ *and undecidable for* $\mathbb{Z}$-*WA*$(\forall\,\neg\mathrm{S})$.

It is worth highlighting that the construction of [11] constructs a weighted automaton that non-deterministically checks for errors in a ω PCP solution candidate. It is possible to construct an automaton that instead verifies that the input word is a solution to the ω PCP instance. This would give an alternative proof to the undecidability of the emptiness problem for strong co-set acceptance from which other undecidability results would follow. The construction is similar to the construction in this section and relies heavily on the properties of both strong and co-set acceptance with two intervals to be avoided.

5. A universal weighted automaton

In this section we consider a universal weighted automaton. The goal is to construct a universal weighted automaton similar to a universal machine which has fixed rules and can simulate any machine that is given as an input. It is well-known that there exists a universal Turing machine [21] and a universal 2-counter machine [16]. A less well-known fact is that there is also a universal semi-Thue system [22]. Note that this universal semi-Thue system does not have fixed rules. To be precise, all but one rules are fixed. The final rule can be effectively constructed from a given semi-Thue system that the universal semi-Thue system simulates. That is, there exists a semi-Thue system T_U with fixed rules R_U such that for any given semi-Thue system T and an input word u, there is a semi-Thue system T'_U with rules $R_U \cup \{(x_T, y_T)\}$. This T'_U terminates on an input word u_T if and only if T terminates on u. Furthermore, words x_T, y_T and u_T are effectively constructable from T and u.

In [13], the authors constructed a universal semi-Thue system where all rewriting rules are fixed and the initial word is an encoding of the system to be simulated. This construction has 24 rules and its alphabet has 8 letters. Note that this semi-Thue system is universal with respect to the termination problem. Other undecidable problems for semi-Thue systems have their own universal semi-Thue systems.

Theorem 17 ([13]) *Let T be a semi-Thue system with input u. There exists a fixed semi-Thue system T_U, and a morphism φ, such that T_U terminates on $\varphi(T, u)$ if and only if T terminates on u.*

In this section we observe that by applying the previous construction to T_U, we obtain a weighted automaton that has fixed transitions and accepts all words if and only if T_U terminates on its input. That is, we revisit all previous steps of our construction. The automaton is constructed using the same idea as in Section 3. Namely, that all words but a solution to the ω PCP are accepted. Unlike the previous definition, where the initial weight was 0, in the universal weighted automaton, there is an additional initial weight. This weight is used to store the information on the input word of the semi-Thue system.

Our aim is to construct a pair of morphisms g and h that map all letters to fixed finite words. From the details of the ω PCP construction presented in (1), it is evident that most of the images are already fixed. Indeed, images of $a_1, b_1, a_2, b_2, \#$ under both g and h do not depend on the given semi-Thue system, while images of $t_i \in R$ and d are dependant. Now, by considering a semi-Thue system with *fixed* rules, also the images of $t_i \in R$ are fixed. The remaining non-fixed image is the h-image of d, which contains the input word of the semi-Thue system T_U. Note, that by Property 3, d has to be the first letter of a solution and that is the only time letter d appears in the word.

Following the construction of the weighted automaton from an instance of the ω PCP, we observe that only transitions corresponding to the letter d are not fixed. We use the fact that d has to be the first letter by fixing weight for d to be 0 and having the input, i.e., the initial weight, depend on d.

There are two cases that can happen when reading the first letter, d, which contains the input of the underlying semi-Thue system T_U. Either the error is in the h-image of d or not. If there is no error in the h-image of d, then the difference of lengths of the images is given as an input. If there is an error, then its position and letter are given. For these two cases, we have two paths in the automaton; see a depiction in Figure 4. In the first path the automaton of Section 3 has all the weights multiplied by 5. In the second path the error verifying part of the automaton is used with weights multiplied by 5 and error verifying transitions have weights $-\ell \cdot 5^2 - \tau(b_e) \cdot 5 - \tau(b_e)$ instead of $-\ell \cdot 5 - \tau(b_e)$ as in the original automaton. That is, the input of our universal weighted automaton is an integer $z \cdot 5^2 + j \cdot 5 + j$ where $z \in \mathbb{N}$ and $j \in \{0, 1, 2, 3, 4\}$. This integer is either $(|h(d)| - |g(d)|)5^2 + 0 \cdot 5 + 0$ corresponding to the case when there are no errors in the h-image of d or $(k - |g(c)|)5^2 + \tau(b_{j_k}) \cdot 5 + \tau(b_{j_k})$ corresponding to the case where k is the position of the error in d and $\tau(b_{j_k})$ is the error.

Let us define the automaton. Recall that we defined $\tau \colon B \to \{1, 2, 3, 4\}$ as $\tau(a) = 1$, $\tau(b) = 2$, $\tau(d) = 3$ and $\tau(\#) = 4$. The universal automaton is $\mathcal{U} = (\{q, q_0, q_1, q_2, q_1', \bot\}, A, \sigma, q, \{q_2\}, \mathbb{Z})$. The states q_0, q_1, q_2 correspond to the first

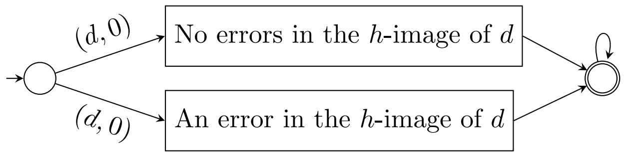

Fig. 4. A illustration of the idea of the construction of the universal weighted automaton. Here, the two blocks contain modified parts of the automaton constructed in Section 3.

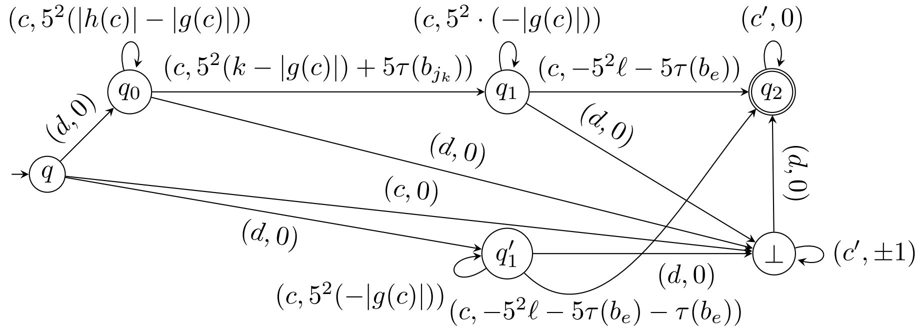

Fig. 5. The universal weighted automaton $\mathcal{U}$. In the figure $c \in A \setminus \{d\}$ and $c' \in A$.

path with transitions for each $c \in A \setminus \{d\}$ and $c' \in A$, $\langle q_0, c, q_0, 5^2(|h(c)| - |g(c)|)\rangle$, $\langle q_1, c, q_1, 5^2(-|g(c)|)\rangle$, $\langle q_2, c', q_2, 0\rangle$ are in σ. For the error checking we need the following transitions for all letters $c \in A \setminus \{d\}$: Let $h(c) = b_{j_1} b_{j_2} \cdots b_{j_{n_1}}$ where $b_{j_k} \in B$, for each index $1 \leq k \leq n_1$. Then let, for each $k = 1, \ldots, n_1$, $\langle q_0, c, q_1, 5^2(k - |g(c)|) + 5\tau(b_{j_k})\rangle \in \sigma$. Let $g(c) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$ where $b_{i_\ell} \in B$, for each index $1 \leq \ell \leq n_2$. For each $\ell = 1, \ldots, n_2$ and letter $b_e \in B$ such that $b_{i_\ell} \neq b_e \in B$, let $\langle q_1, c, q_2, -5^2\ell - 5\tau(b_e)\rangle \in \sigma$.

The state q_1' corresponds to the second path with transitions, for each $c \in A \setminus \{d\}$, $\langle q_1', c, q_1', 5^2(-|g(c)|)\rangle$ are in σ. For the error verification we need the following transitions for all letters $c \in A \setminus \{d\}$. Let $g(c) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$ where $b_{i_\ell} \in B$, for each index $1 \leq \ell \leq n_2$. For each $\ell = 1, \ldots, n_2$ and letter $b_e \in B$ such that $b_{i_\ell} \neq b_e \in B$, let $\langle q_1', c, q_2, -5^2\ell - 5\tau(b_e) - \tau(b_e)\rangle \in \sigma$.

Finally, transitions $\langle q, d, q_0, 0\rangle, \langle q, d, q_1', 0\rangle$ to pick a path, transitions $\langle q, c, \bot, 0\rangle$, for each $c \in A \setminus \{d\}$, for words not starting with d, transitions $\langle p, d, \bot, 0\rangle$ where $p \in \{q_0, q_1, q_1'\}$, for words that have more than one occurrence of the letter d, transitions $\langle \bot, c, \bot, \pm 1\rangle, \langle \bot, c, q_2, 0\rangle$ for $c \in A$ and finally $\langle q_2, d, q_2, 0\rangle$.

Let (g, h) be an instance of the ω PCP as in (1). We define the inputs next. Note that the possible input values correspond to weights that can be added when reading d in the automaton of Section 3 with slight modifications as discussed above. Let the set of inputs corresponding to the letter d be $\alpha(d)$, defined as the union of

$$\{(|h(d)| - |g(d)|)5^2\} \text{ and} \tag{5}$$

$$\{(i - |g(d)|)5^2 + \tau(b_j)5 + \tau(b_j) \mid i = |g(d)|+1, \dots, |h(d)| \text{ and } b_j = h(d)(i) \in B\}. \tag{6}$$

Note that $(|h(d)| - |g(d)|)5^2 \neq 0$ as per construction of the instance. Now a word $dw \in A^\omega$ is accepted by $\mathcal{U}$ if and only if for a computation path π of dw there exists a prefix $p \leq \pi$ that reaches q_2 with weight 0. That is, $\gamma(p) + \beta = 0$ where $\beta \in \alpha(d)$.

Next, we show that an input defines the path that needs to be chosen. Assume first that the input is $z5^2 + \tau(b_{j_k})5 + \tau(b_{j_k})$ and the first transition is $\langle q, d, q_0, 0\rangle$. Now the automaton is in state q_0 with weight $z5^2 + b_{j_k}5 + \tau(b_{j_k})$ but none of the weights on this path modify the coefficient of 5^0 (unless letter d is read) and thus the weight is nonzero in the state q_2. Assume then that the input is $z5^2 + 0 \cdot 5 + 0$ and the first transition is $\langle q, d, q_1', 0\rangle$. The path reaching q_2 (without visiting $\perp$) has $x5^2 - e \cdot 5 - e$ for some $x \in \mathbb{Z}$ and $e \in \{1,2,3,4\}$ which is nonzero. That is, for an input $z5^2 + \tau(b_{j_k})5 + \tau(b_{j_k})$ the upper path has to be chosen and for an input $z5^2$ the lower path has to be chosen. It is clear that after that the computation follows the corresponding computation of $\mathcal{A}$.

From the construction, it is evident that only a solution to the ω PCP instance does not have a path that ends in q_2 with weight 0 for all $\beta \in \alpha(d)$. Note that in $\mathcal{U}$ all transitions are fixed as, regardless of $h(d)$ and $g(d)$, the transitions are always $\langle p, d, p', 0\rangle$ or $\langle \perp, d, \perp, \pm 1\rangle$.

Let $\beta \in \mathbb{Z}$. If $w \in \mathcal{A}^\omega$ is accepted by $\mathcal{U}$ with the input β, we denote it by $w \in L(\mathcal{U}_\beta)$. From the previous consideration we obtain that for an instance of the ω PCP of form (1), we can compute a finite set $\alpha(d)$ such that $\bigcup_{z \in \alpha(d)} L(\mathcal{U}_z) = A^\omega$ if and only if (g, h) has no solution.

Example 18. *Let us consider a universal weighted automaton constructed from the* ω PCP *instance of Example 2. As in Example 8, the resulting automaton has too many transitions to be presented in a sensible manner. However, the set* $\alpha(d)$ *and its effect on the computation can be presented. Recall, that the input word of the semi-Thue system of Example 2 is* $\boldsymbol{a}$ *and it is encoded by letter* d *in the* ω PCP *in the following manner* $h(d) = dd\boldsymbol{a}dd\#d$ *and* $g(d) = dd$. *Recall also that* τ *encodes* a *as 1,* b *as 2,* d *as 3 and* $\#$ *as 4. Now as discussed before, the set* $\alpha(d)$ *containing all possible input values is*

$$\begin{aligned}\alpha(d) = \{&(7-2)\cdot 5^2, (3-2)\cdot 5^2 + 1\cdot 5 + 1, (4-2)\cdot 5^2 + 3\cdot 5 + 3,\\ &(5-2)\cdot 5^2 + 3\cdot 5 + 3, (6-2)\cdot 5^2 + 4\cdot 5 + 4, (7-2)\cdot 5^2 + 3\cdot 5 + 3\}\\ = \{&125, 31, 68, 93, 124, 143\},\end{aligned}$$

where the first integer is of form (5) *and the rest are of form* (6). *A word starting with* db_1 *is accepted with input 6 as there the transition* $\langle q, d, q_1', 0\rangle$ *is followed by the transition* $\langle q_1', b_1, q_2, -5^2\cdot 1 - 5\cdot 1 - 1\rangle$, *reaching the final state* q_2 *with zero weight.*

Finally, by applying the construction of the ω PCP instance presented in Section 3 to a universal semi-Thue system of Theorem 17, we obtain the following result.

Theorem 19. *Let T_U be a universal semi-Thue system and let u be its input. There exists a computable finite set $Z \subseteq \mathbb{Z}$ such that $\bigcup_{z\in Z} L(\mathcal{U}_z) = A^\omega$ if and only if T_U terminates on u.*

By following the chain of reductions one step further, we note that the above result holds for any semi-Thue system. That is, for a given semi-Thue system T and its input u, we first construct T_U and $\varphi(T, u)$ and then apply Theorem 19.

Corollary 20. *Let T be a semi-Thue system and let u be its input. There exists a computable finite set $Z \subseteq \mathbb{Z}$ such that $\bigcup_{z\in Z} L(\mathcal{U}_z) = A^\omega$ if and only if T terminates on u.*

Corollary 21. *Given a finite set of integers, $Z \subseteq \mathbb{Z}$, it is undecidable whether or not $\bigcup_{z\in Z} L(\mathcal{U}_z) = A^\omega$ holds for a fixed integer weighted automaton $\mathcal{U}$ over its alphabet A.*

Acknowledgements

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

References

[1] S. Almagor, U. Boker and O. Kupferman, What's decidable about weighted automata?, *Inf. Comput.* (2020).
[2] M. Blondin, C. Haase and F. Mazowiecki, Affine extensions of integer vector addition systems with states, *Proceedings of CONCUR 2018*, *LIPIcs* **118** (2018), pp. 14:1–14:17.
[3] U. Boker, Quantitative vs. weighted automata, *Proceedings of RP 2021*, *LNCS* **13035** (2021), pp. 3–18.
[4] K. Chatterjee, L. Doyen and T. A. Henzinger, Quantitative languages, *ACM Trans. Comput. Log.* **11**(4) (2010).
[5] K. Culik and J. Kari, Image compression using weighted finite automata, *Computers & Graphics* **17**(3) (1993) 305–313.
[6] J. Dong and Q. Liu, Undecidability of infinite Post correspondence problem for instances of size 8, *ITA* **46**(3) (2012) 451–457.
[7] M. Droste, W. Kuich and H. Vogler, *Handbook of weighted automata* (Springer Science & Business Media, 2009).
[8] V. Halava, Finite substitutions and integer weighted finite automata, licentiate thesis, University of Turku, Turku, Finland (1998).
[9] V. Halava and T. Harju, Undecidability in integer weighted finite automata, *Fundam. Inform.* **38**(1-2) (1999) 189–200.
[10] V. Halava and T. Harju, Undecidability of infinite post correspondence problem for instances of size 9, *RAIRO - ITA* **40**(4) (2006) 551–557.
[11] V. Halava, T. Harju, R. Niskanen and I. Potapov, Weighted automata on infinite words in the context of Attacker-Defender games, *Inf. Comput.* **255** (2017) 27–44.

[12] V. Halava, T. Harju, R. Niskanen and I. Potapov, Integer weighted automata on infinite words, *Proceedings of DLT 2021*, *LNCS* **12811** (2021), pp. 167–179.
[13] V. Halava, Yu. Matiyasevich and R. Niskanen, Small semi-thue system universal with respect to the termination problem, *Fundam. Inform.* **154**(1-4) (2017) 177–184.
[14] S. Kiefer, A. Murawski, J. Ouaknine, B. Wachter and J. Worrell, On the complexity of equivalence and minimisation for Q-weighted automata, *LMCS* **9**(1) (2013).
[15] Yu. Matiyasevich and G. Sénizergues, Decision problems for semi-Thue systems with a few rules, *Theor. Comput. Sci.* **330**(1) (2005) 145–169.
[16] M. L. Minsky, *Computation: finite and infinite machines* (Prentice-Hall, Inc., 1967).
[17] T. Neary, Undecidability in binary tag systems and the Post correspondence problem for five pairs of words, *STACS 2015*, *LIPIcs* **30** (2015), pp. 649–661.
[18] R. Niskanen, I. Potapov and J. Reichert, On decidability and complexity of low-dimensional robot games, *J. Comput. Syst. Sci.* **107** (2020) 124–141.
[19] E. L. Post, A variant of a recursively unsolvable problem, *Bull. Amer. Math. Soc.* **52**(4) (1946) 264–268.
[20] E. Roche and Y. Schabes, Speech recognition by composition of weighted finite automata, *Finite-state language processing*, (MIT press, 1997), pp. 431–454.
[21] Yu. Rogozhin, Small universal Turing machines, *Theor. Comput. Sci.* **168**(2) (1996) 215–240.
[22] G. Sénizergues, Some undecidable termination problems for semi-Thue systems, *Theor. Comput. Sci.* **142**(2) (1995) 257–276.
[23] W. Thomas, Automata on infinite objects, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, (MIT Press, 1990), pp. 133–192.