# Integer Weighted Automata on Infinite Words

V. Halava[1[0000−0003−3633−4902]], T. Harju[1[0000−0002−9640−6309]],
R. Niskanen[2[0000−0002−2210−1481]], and I. Potapov[2[0000−0002−7192−7853]]

[1] Department of Mathematics and Statistics, University of Turku, Turku, Finland
{vesa.halava,harju}@utu.fi
[2] Department of Computer Science, University of Liverpool, Liverpool, UK
{reino.niskanen,potapov}@liverpool.ac.uk

**Abstract.** 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 where the automaton is fixed and the word problem is undecidable.

## 1 Introduction

Weighted automata have been extensively studied in recent years [1,6,12] and have a wide range of applications, such as speech-recognition [17] and image compression [4]. 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.

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 for weighted automata over tropical semirings, i.e., $(\mathbb{Z} \cup \{\infty\}, \min, +, \infty, 0)$, where 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) − a word is accepted if its value using the semiring sum is at most $\nu$. In [3], 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 [20] and more recently infinite words have been used to simulate various processes in computational games [10,16].

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 [3], 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 [10], 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 ($\omega$ PCP) and the state reduction is achieved by proving more restricted form of the $\omega$ PCP than in [9]. 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. For this universal automaton it is undecidable whether a given word with an initial integer weight is accepted.

## 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 $A^\omega$. The monoid of all finite words over $A$ is denoted by $A^*$. The empty word is denoted by $\varepsilon$. A word $u \in A^*$ is a *prefix* of $v \in A^*$, denoted by $u \le v$, 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 $\varepsilon$ is 0. For a word $w$, we denote by $w(i)$ the $i$th 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 $dA^\omega$ 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 $t_i = \left\langle q_{j_i}, a_{k_i}, q_{j_{i+1}}, z_i\right\rangle$ for $i > 0$ and $q_{j_0} = q_0$. We call such path $\pi$ a *computation path*. Denote by $\mathcal{R}(\pi)$ the set of all reachable configurations following a path $\pi$. 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 path $\pi$ by $\pi_w$ 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 $\pi$. The *weight* of the configuration $c$ is $\gamma(c) = z$. We say that the configuration $c$ *reaches* state $q$. If computation path $\pi$ 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 $\pi$ 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 $\pi$ 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 $g(w) = h(w)$. It is well known that it is undecidable whether or not an instance of the PCP has a solution. The problem remains undecidable for $A$ with $|A| \geq 5$; see [15]. The cardinality of the domain alphabet $A$ is said to be the *size* of the instance.

The *infinite Post correspondence problem*, $\omega$ PCP, is a natural extension of the PCP. An infinite word $w$ is a *solution* of an instance $(g, h)$ of the $\omega$ PCP if for every finite prefix $p$ of $w$ either $h(p) < g(p)$ or $g(p) < h(p)$ holds. In the $\omega$ 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 [9] that the problem is undecidable for domain alphabets $A$ with $|A| \geq 9$ and in [5] it was improved to $|A| \geq 8$. A more general formulation of the $\omega$ PCP was used in both proofs, namely the prefixes did not have to be proper. However, both constructions rule out non-proper prefixes; see [9,5] for details.

## 3 Universality problem for zero acceptance

In this section we improve the result of [10], 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 $\omega$ PCP instance. In the proof of undecidability of the universality problem for weighted automata, for each instance $(g, h)$ of the $\omega$ 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 3-state integer weighted automaton $\mathcal{A}$ over its alphabet $A$.*

Let us first focus on constructing the instance of the $\omega$ PCP. In [10], a weighted automaton was constructed from an arbitrary instance of the $\omega$ PCP. We reiterate the construction of an instance of the $\omega$ PCP found in [9], highlighting the properties that simplify the construction of the automaton.

The $\omega$ PCP was shown to be undecidable for instances of size 9 in [9]. The proof uses a reduction from the termination problem of the semi-Thue systems proved to be undecidable for the 3-rule semi-Thue systems from [13]. We shall now present the construction from [9].

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 $\omega$ 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 [9] the following construction was given for a semi-Thue system $T$ and an input word $u$: Define the morphisms $g, h\colon A^* \to \{a, b, d, \#\}^*$ by (recall that for $t_i \in R$, we denoted $t_i = (u_i, v_i)$)

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

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 $h(t_i) = d$.

It was proved in [9] that the following property holds:

*Property 1.* Let $(g, h)$ be an $\omega$ 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 a $d$, and there is one occurrence of # in $h(d)$ and no occurrences of # in $g(d)$. Later each occurrences of # is produced from # by both $g$ and $h$. Therefore there are infinitely many letters # in each infinite solution.

*Property 2.* Let $(g, h)$ be as in (1). In a solution, the image under $g$ cannot be longer than the image under $h$.

*Property 3.* 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 the *error*) is reached in $h(w)$ at least one letter (of $w$) earlier than it is reached in $g(w)$.

The two properties are illustrated in Figure 1. In the next theorem, we restate and sharpen the result of [9] by improving the undecidability claim of the $\omega$ PCP.

**Theorem 2.** *Let $(g, h)$ be an instance of the $\omega$ PCP defined as in (1) that satisfies Properties 1, 2, 3. It is undecidable whether a solution to $(g, h)$ exists.*

Next, we construct the weighted automaton based on the undecidable instance of the $\omega$ PCP of Theorem 2. This will allow us to prove Theorem 1.

Let $(g, h)$ be a fixed instance of the $\omega$ PCP as defined in (1). Then $g, h\colon A^* \to B^*$ where $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_{s-1}\}$. 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

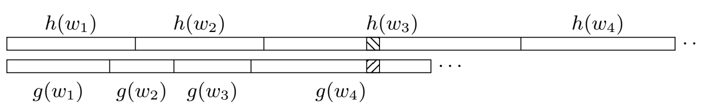


**Fig. 1.** An illustration of a solution candidate to the instance of the $\omega$ PCP satisfying Properties 2 and 3. Here, ⧅ represent the first letter of $h(w_1w_2w_3w_4\cdots)$ that is compared to a letter of $g(w_1w_2w_3w_4\cdots)$ which is represented by ▨.

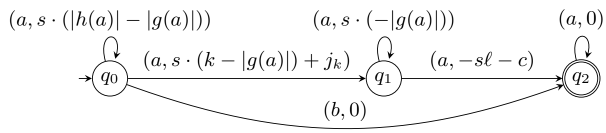


**Fig. 2.** The weighted automaton $\mathcal{A}$. In the figure $a \in A$ and $b \in A \setminus \{d\}$.

$\mathcal{A}$ iff for some finite prefix $p$ of $w$, $g(p) \not< h(p)$. Moreover, by Property 3, 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 $p$, the *error checking*.

Let us begin with the transitions of $\mathcal{A}$. The automaton is depicted in Figure 2. Recall that the cardinality of the alphabet $B$ is $s - 1$. First for each $a \in A$, let $\langle q_0, a, q_0, s(|h(a)| - |g(a)|)\rangle$, $\langle q_1, a, q_1, s(-|g(a)|)\rangle$, $\langle q_2, a, q_2, 0\rangle$ be in $\sigma$ and for all $b \in A \setminus \{d\}$, let $\langle q_0, b, q_2, 0\rangle \in \sigma$. For the error checking we need the following transitions for all letters $a \in A$: Let $h(a) = b_{j_1} b_{j_2} \cdots b_{j_{n_1}}$, where $b_{j_k} \in B$, for each index $1 \leq k \leq n_1$. Then let, for each $k = 1, \ldots, n_1$,

$$\langle q_0, a, q_1, s(k - |g(a)|) + j_k\rangle \in \sigma \tag{3}$$

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

$$\langle q_1, a, q_2, -s\ell - c\rangle \in \sigma. \tag{4}$$

We call the transitions in (3) *error guessing transitions* and in (4) *error verifying transitions*. The next lemma shows a key property about words accepted by $\mathcal{A}$. The proof relies on analysis of weights along computation paths.

**Lemma 1.** *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).*

We are ready to prove the main theorem. By Lemma 1, 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 $\omega$ PCP. By Theorem 2, 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 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 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 1.** *It is undecidable whether or not $\mathcal{L}(\mathcal{A}) = A^\omega$ holds for 3-state integer weighted automaton $\mathcal{A}$ over its alphabet $A$.*

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 3.** *It is decidable whether or not $L(\mathcal{A}) = \emptyset$ holds for 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 [8]. 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 [7] 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 2 (follows from Theorem 1).**

1. *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.*
2. *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 $\mathcal{A}'$.*

## 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 property $\varphi$". We will use notation $\mathbb{Z}$-WA($\exists\,\varphi$) for integer weighted finite automata on infinite words with acceptance condition $\varphi$. Analogously, $\mathbb{Z}$-WA($\forall\,\varphi$) denotes the strong acceptance.

In [10], 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

| | |
|---|---|
| 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 $S \subseteq \mathbb{Z}$.

asked whether every word is accepted and in the emptiness problem whether at least one infinite word is 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 $\varphi$. We present different acceptances and acceptance conditions in Table 1.

Let us discuss these acceptance properties next. In the already mentioned zero acceptance, 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, 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) proven in [10] 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. We next 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 4.** *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, $A$ can be considered as a $\mathbb{Z}$-VASS for which the reachability relation is effectively semi-linear [2]. Hence, the property can be expressed a sentence in Presburger arithmetics, which is a decidable logic. □

**Corollary 3.** *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 $A$.*

*Proof.* The universality problem for $\mathbb{Z}$-WA(∀ ¬Z) is dual to the emptiness problem for $\mathbb{Z}$-WA(∃ Z), which is decidable by Theorem 3. Analogously, the universality problems for $\mathbb{Z}$-WA(∀ Z) and $\mathbb{Z}$-WA(∃ ¬Z) are dual to the emptiness problems for $\mathbb{Z}$-WA(∃ ¬Z) and $\mathbb{Z}$-WA(∀ Z), respectively, which are decidable. □

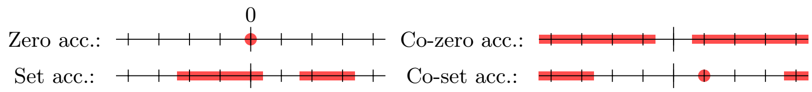


**Fig. 3.** An illustration of different acceptance conditions. In red are weights that are to be reached in an accepting path.

In both zero acceptance and co-zero acceptance, 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 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*. This is summarised in Table 2 where the decidability statuses of the universality and the emptiness problems for the different acceptance conditions.

**Corollary 4.** *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 [10] constructs a weighted automaton that non-deterministically checks for error in a $\omega$ PCP solution candidate. It is possible to construct an automaton with strong co-set acceptance for which the emptiness problem is undecidable and the automaton verifies that the input word is a solution to the $\omega$ PCP instance. The automaton relies on the properties of both strong and co-set acceptance with two intervals to be avoided.

| 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.

## 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 [18] and a universal 2-counter machine [14]. A less well-known fact is that there is also a universal semi-Thue system [19]. In [11], the authors constructed a universal semi-Thue system where the rewriting rules are fixed and the initial word is an encoding of the system to be simulated.

From the details of the $\omega$ PCP construction presented in (1), it is evident that only one of the pairs is not fixed and depends on the input to the given semi-Thue system. Namely, $d$ contains the initial word of the semi-Thue system. Note, that $d$ has to be the first letter of a solution.

We construct a weighted automaton with fixed state structure and transitions. The automaton is constructed using the same idea as in section 3. Namely, that all words but a solution to the $\omega$ 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. Note that due to our approach of storing only partial information about the images of the morphisms in the weight, we do not actually need to know what the input is.

From the previous remark in our weighted automaton 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 $d$ with the weighted automaton. Either the error is in the image of $d$ or not. If there is no error in the 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. That is, the input of our universal weighted automaton is an integer $zs^2 + js + j$ where $z \in \mathbb{N}, j \in \{0, \ldots, s-1\}$ and $s = |B| + 1$. This integer is either $(|h(d)| - |g(d)|)s^2 + 0s + 0$ corresponding to the case when there are no errors in the image of $d$ or $(k - |g(a)|)s^2 + j_k s + j_k$ corresponding to the case where $k$ is the position of the error in $d$ and $j_k$ is the error. For these two cases, we have two paths in the automaton. In the first path the automaton of Section 3 has all the weights multiplied by $s$. In the second path the error verifying part of the

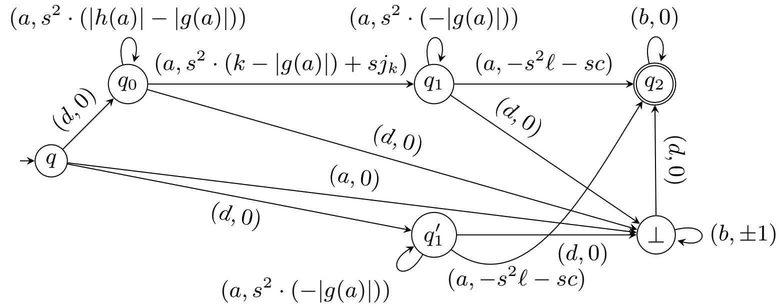


**Fig. 4.** The universal weighted automaton $\mathcal{U}$. In the figure $a \in A \setminus \{d\}$ and $b \in A$.

automaton is used with weights multiplied by $s$ and error verifying transitions have weights $-\ell s^2 - cs - c$ instead of $-\ell s - c$ as in the original automaton.

The universal automaton is $\mathcal{U} = (\{q, q_0, q_1, q_2, q'_1, \perp\}, A, \sigma, q, \{q_2\}, \mathbb{Z})$. The states $q_0$, $q_1$, $q_2$ correspond to the first path with transitions for each $a \in A \setminus \{d\}$ and $b \in A$, $\langle q_0, a, q_0, s^2(|h(a)| - |g(a)|)\rangle$, $\langle q_1, a, q_1, s^2(-|g(a)|)\rangle$, $\langle q_2, b, q_2, 0\rangle$ are in $\sigma$. For the error checking we need the following transitions for all letters $a \in A \setminus \{d\}$: Let $h(a) = b_{j_1} b_{j_2} \cdots b_{j_{n_1}}$ where $b_{j_k} \in B$, for each index $1 \leq k \leq n_1$. Then let, for each $k = 1, \ldots, n_1$, $\langle q_0, a, q_1, s^2(k - |g(a)|) + sj_k\rangle \in \sigma$. Let $g(a) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$ where $b_{i_\ell} \in B$, for each index $1 \leq \ell \leq n_2$. For each $\ell = 1, \ldots, n_2$ and letter $b_c \in B$ such that $b_{i_\ell} \neq b_c \in B$, let $\langle q_1, a, q_2, -s^2\ell - sc\rangle \in \sigma$.

The state $q'_1$ corresponds to the second path with transitions, for each $a \in A \setminus \{d\}$, $\langle q'_1, a, q'_1, s^2(-|g(a)|)\rangle$ are in $\sigma$. For the error verification we need the following transitions for all letters $a \in A \setminus \{d\}$. Let $g(a) = b_{i_1} b_{i_2} \cdots b_{i_{n_2}}$ where $b_{i_\ell} \in B$, for each index $1 \leq \ell \leq n_2$. For each $\ell = 1, \ldots, n_2$ and letter $b_c \in B$ such that $b_{i_\ell} \neq b_c \in B$, let $\langle q'_1, a, q_2, -s^2\ell - sc - c\rangle \in \sigma$.

Finally, transitions $\langle q, d, q_0, 0\rangle$, $\langle q, d, q'_1, 0\rangle$ to pick a path, transitions $\langle q, a, \perp, 0\rangle$, for each $a \in A \setminus d$, for words not starting with $d$, transitions $\langle p, d, \perp, 0\rangle$ where $p \in \{q_0, q_1, q'_1\}$, for words that have letter $d$, transitions $\langle \perp, b, \perp, \pm 1\rangle$, $\langle \perp, b, q_2, 0\rangle$ for $b \in A$ and finally $\langle q_2, d, q_2, 0\rangle$.

Let the set of inputs corresponding to the letter $d$ be $\alpha(d)$, defined as the union of $\{(|h(d)| - |g(d)|)s^2\}$ and $\{is^2 + js + j \mid i = |g(d)|, |g(d)|+1, \ldots, |h(d)| \text{ and } b_j = h(d)(i) \in B\}$. Now a word $dw \in A^\omega$ is accepted by $\mathcal{U}$ if and only if for a computation path $\pi$ 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 $zs^2 + j_k s + j_k$ and the first transition is $\langle q, d, q_0, 0\rangle$. Now the automaton is in state $q_0$ with weight $zs^2 + j_k s + j_k$ but none of the weights on this path modify the coefficient of $s^0$ (unless letter $d$ is read) and thus the weight is nonzero in state $q_2$. Assume then that the input is $zs^2 + 0s + 0$ and the first transition is $\langle q, d, q'_1, 0\rangle$. The path reaching $q_2$ (without visiting $\perp$) has $xs^2 - cs - c$ for some $x \in \mathbb{Z}$ and $c \in \{1, \ldots, s-1\}$ which is nonzero. That is for input $zs^2 + j_k s + j_k$ the upper path has to be chosen and for input $zs^2$ the lower path has to be chosen. It is clear that after that the computation follows the corresponding computation of $\mathcal{A}$. If the input is 0, then the only path ending in $q_2$ with weight 0 goes through $\perp$, that is, it is not a solution to the $\omega$ PCP.

From the construction, it is evident that only a solution to the $\omega$ PCP instance does not have a path that ends in $q_2$ with weight 0. 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 $w \in A^\omega$ and $\beta \in \mathbb{Z}$. If $w$ is accepted by $\mathcal{U}$ with input $\beta$, we denote it by $(w, \beta) \in L(\mathcal{U})$. From the previous consideration we get:

**Theorem 5.** *Let $w \in A^\omega$ and $\beta \in \mathbb{Z}$. It is undecidable whether $(w, \beta) \in L(\mathcal{U})$, where $\mathcal{U}$ is a fixed weighted automaton on infinite words under zero acceptance.*

## References


1. Almagor, S., Boker, U., Kupferman, O.: What's decidable about weighted automata? Inf. Comput. (2020). https://doi.org/10.1016/j.ic.2020.104651
2. Blondin, M., Haase, C., Mazowiecki, F.: Affine extensions of integer vector addition systems with states. In: Proceedings of CONCUR 2018. LIPIcs, vol. 118, pp. 14:1–14:17 (2018). https://doi.org/10.4230/lipics.concur.2018.14
3. Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. ACM Trans. Comput. Log. **11**(4) (2010). https://doi.org/10.1145/1805950.1805953
4. Culik, K., Kari, J.: Image compression using weighted finite automata. Computers & Graphics **17**(3), 305–313 (1993). https://doi.org/10.1016/0097-8493(93)90079-O
5. Dong, J., Liu, Q.: Undecidability of infinite Post correspondence problem for instances of size 8. ITA **46**(3), 451–457 (2012). https://doi.org/10.1051/ita/2012015
6. Droste, M., Kuich, W., Vogler, H.: Handbook of weighted automata. Springer Science & Business Media (2009)
7. Halava, V.: Finite Substitutions and Integer Weighted Finite Automata. Licentiate thesis, University of Turku, Turku, Finland (1998)
8. Halava, V., Harju, T.: Undecidability in integer weighted finite automata. Fundam. Inform. **38**(1-2), 189–200 (1999). https://doi.org/10.3233/FI-1999-381215
9. Halava, V., Harju, T.: Undecidability of infinite post correspondence problem for instances of size 9. RAIRO - ITA **40**(4), 551–557 (2006)
10. Halava, V., Harju, T., Niskanen, R., Potapov, I.: Weighted automata on infinite words in the context of Attacker-Defender games. Inf. Comput. **255**, 27–44 (2017)
11. Halava, V., Matiyasevich, Yu., Niskanen, R.: Small semi-thue system universal with respect to the termination problem. Fundam. Inform. **154**(1-4), 177–184 (2017)
12. Kiefer, S., Murawski, A., Ouaknine, J., Wachter, B., Worrell, J.: On the complexity of equivalence and minimisation for Q-weighted automata. LMCS **9**(1) (2013)
13. Matiyasevich, Yu., Sénizergues, G.: Decision problems for semi-Thue systems with a few rules. Theor. Comput. Sci. **330**(1), 145–169 (2005)
14. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc. (1967)
15. Neary, T.: Undecidability in binary tag systems and the Post correspondence problem for five pairs of words. In: STACS 2015. LIPIcs, vol. 30, pp. 649–661 (2015)
16. Niskanen, R., Potapov, I., Reichert, J.: On decidability and complexity of low-dimensional robot games. J. Comput. Syst. Sci. **107**, 124–141 (2020)
17. Roche, E., Schabes, Y.: Speech recognition by composition of weighted finite automata. In: Finite-state language processing, pp. 431–454. MIT press (1997)

18. Rogozhin, Yu.: Small universal Turing machines. Theor. Comput. Sci. **168**(2), 215–240 (1996). https://doi.org/10.1016/S0304-3975(96)00077-1
19. Sénizergues, G.: Some undecidable termination problems for semi-Thue systems. Theor. Comput. Sci. **142**(2), 257–276 (1995)
20. Thomas, W.: Automata on infinite objects. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics, pp. 133–192. MIT Press (1990)