Finite Substitutions and Integer Weighted Finite Automata

Vesa Halava

Turku Centre for Computer Science TUCS,
Lemminkäisenkatu 14 A, 4th floor,
FIN-20520 Turku, Finland.
e-mail: vehalava@utu.fi

Turku Centre for Computer Science
TUCS Technical Report No 197
August 1998

ISBN 952-12-0260-2
ISSN 1239-1891

Abstract

In this work we present a new chain of undecidability reductions, which begins from the classical halting problem of Turing machines and ends to the undecidability proof of the equivalence problem for finite substitutions on regular languages in Chapter 4. This undecidability result was originally proved by L. Lisovik in 1997. We present a new proof, which is shorter and more elementary than the original one. Our proof uses the undecidability of the universe problem for the integer weighted finite automata.

An integer weighted finite automaton is a finite automaton, which has integer weights on its edges. This automaton accepts an input word, if there exists a path reading the word such that the sum of the weights of the used edges is zero. In the universe problem we ask whether a given integer weighted finite automaton accepts all its input words. This problem is proved undecidable in Chapter 3. The proof uses the undecidability of a certain modification of the Post Correspondence Problem, proved in Chapter 2.

We also compare integer weighted finite automata to some other counter automata and consider the closure properties of the languages accepted by the integer weighted finite automata. Further, we also compare Lisovik's original chain of undecidability reduction to the one presented in this work.

Keywords: undecidability, weighted finite automata, unimodal, universe problem, finite substitutions, equivalence problem, Post Correspondence Problem

TUCS Research Group
Theory Group: Mathematical Structures in Computer Science

Acknowledgements

This report is a somewhat corrected version of my Licentiate's Thesis accepted in the Department of Mathematics of the University of Turku. I am very grateful to the reviewers, Dr. Alexandru Mateescu and Prof. Paavo Turakainen, for these corrections.

I would like to thank my supervisor Dr. Tero Harju for the help and suggestions he has given to me in our many discussions. His advice have made many improvements to the results in this work. I would also like to thank Prof. Juhani Karhumäki for guiding me to this topic.

Finally, of course, I would like to thank my wife Heli for the support.

August 27, 1998

Vesa Halava

Contents

Preface

In this thesis we shall present a chain of undecidability reductions to show that the equivalence problem of finite substitutions on regular language is undecidable. This result was originally proved by Lisovik in 1997 [17]. The chain we shall present is a new one, which is shorter than the original one and the proofs are elementary. The base of our chain will be the halting problem for Turing machines.

First we prove that a certain modification of the Post Correspondence Problem is undecidable. In this modification we are given two special type of morphisms, and we are asked whether there exists a fixed type of a word such that the image of the word under the first morphism is a prefix of the image of the word under the second morphism. The formal definition of this modification and the undecidability proof is given in Chapter 2.

In the first main chapter of the thesis, Chapter 3, we define integer weighted finite automata, which are finite automata without final states and having integer weights on the transitions. The sum of the weights of the transitions on a given path (or a computation) is the weight of this path and the automaton accepts an input word if there is a path for this word starting from the initial state and having the weight zero.

The main new result in this thesis is that the universe problem, asking whether all input words are accepted by a given integer weighted finite automaton, is undecidable. Actually, we prove that the universe problem is undecidable for a very special kind of integer weighted finite automata, called unimodal, where the weights on the transitions are defined so that all paths of the automaton can be divided into four consecutive parts; first having positive weights on the transitions, second having zero weights on the transitions, third having negative weights on the transitions and fourth having zero weights on the transitions. Any of these parts can also be trivial. The proof we shall present is from Halava and Harju [7], and this proof uses the modification of the Post Correspondence Problem mentioned above.

An integer weighted finite automaton is closely related to some models of counter automata that are known in the literature. Similarities and differences between these automata are considered in Section 3.3.

In Chapter 3 we also consider the closure properties of the family of languages accepted by the integer weighted finite automata.

In Chapter 4 we present a new proof for the undecidability of the equivalence problem of finite substitutions on regular language, a result originally proved by Lisovik in [17]. Our proof uses the undecidability of the universe problem for integer weighted finite automata and it is from Halava and Harju [8]. The proof we present is considerably easier and the chain of undecidability results used in it is much shorter than in Lisovik's proof.

Also a special inclusion problem, asking whether for all words, the image of a word under a given periodic morphism is included in the image of the same word under a given finite substitution, is proved to be undecidable in Chapter 4. This result is also from Halava and Harju [8].

In the last section of Chapter 4 we shall compare the chain of undecidability reduction of the original proof of Lisovik to the one presented in this thesis.

1 Preliminaries

Throughout this work we shall denote the sets of natural numbers, integers, rational numbers and real numbers by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$, respectively.

For a set S we denote by 2^S the *power set* of S, i.e. the collection of all subsets of S.

Sometimes a singleton set $\{u\}$ is written simply u.

1.1 Semigroups, monoids and groups

Let S be a set and assume that $\cdot$ is an associative binary operation on S, that is, for all a, b and c in S,

$$a \cdot b \in S \quad \text{and} \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c.$$

Then $(S, \cdot)$ is called a *semigroup*. Usually, if no confusion is possible, we denote $ab = a \cdot b$, and the semigroup $(S, \cdot)$ is denoted simply by S.

An element ι in a semigroup S is called an *identity* if, for all a in S, $a\iota = \iota a = a$. If there is an identity element in a semigroup S, then S is called a *monoid*. Note that the identity element in a monoid is necessarily unique.

An element z in S is called a *zero* if, for all a in S, $za = az = z$. Note that also the zero element, if it exists, is unique in a semigroup.

Let G be a set. A triple $\mathbf{G} = (G, \cdot, \iota)$ is called *group*, if $(G, \cdot)$ is a monoid with identity ι and, for all a in S, there exists an *inverse* element a^{-1} such that

$$a \cdot a^{-1} = a^{-1} \cdot a = \iota.$$

A group $\mathbf{G}$ is called *Abelian*, if it is commutative, i.e. if for all a and b in G, $ab = ba$.

For example, $(\mathbb{Z}, +, 0)$ is an Abelian group. We shall use this *additive group of integers* quite often in what follows.

1.2 Basics of words

Let A be a finite set of symbols, called an *alphabet*. A *word* over A is a finite sequence of symbols in A. We denote by A^* the set of all words over A. Note that also the *empty word* is in A^*. Throughout this paper we shall denote the empty word by ε.

Let $u = u_1 \dots u_n$ and $v = v_1 \dots v_m$ be two words in A^*, where u_i and v_j are in A for $1 \le i \le n$ and $1 \le j \le m$. Define the binary operation $\cdot$ on A^*, called *concatenation*, in a usual way by

$$u \cdot v = uv = u_1 \dots u_n v_1 \dots v_m.$$

This operation is clearly associative on A^* and, since the empty word is the identity element of A^*, A^* is a monoid. We say that A^* is the *word monoid generated by the alphabet A*. The *word semigroup generated by A* is $A^+ = A^* \setminus \{\varepsilon\}$.

A semigroup S is called *free* or *freely generated* if there is a subset X of S such that each element of S has a unique factorization over X. Such a set X is called the *free generating set of S*. A monoid M with identity element ι is called *free* if $M \setminus \{\iota\}$ is a free semigroup.

Note that a word monoid A^* is free and A is its free generating set.

Let A be an alphabet and assume that $u = u_1 \dots u_n$ is in A^* with $u_i \in A$. The *length* of the word u is n and it is denoted by $|u| = n$. The rth letter of a word v will be sometimes denoted by $(v)_r$ for $1 \leq r \leq |v|$.

Assume that u and v are in A^*. Then u is a *factor* of v, if $v = w_1 u w_2$ for some words w_1 and w_2 in A^*. If $u \neq \varepsilon$, then it is called *proper* factor. Further, we say that u is a *prefix* of v, if $w_1 = \varepsilon$, i.e. $v = uw_2$. If $w_2 = \varepsilon$, i.e. $v = w_1 u$ then u is called a *suffix* of v. If u is a prefix of v, we denote $u < v$. If $u < v$, i.e. $v = uw$ for some $w \in A^*$, then we write $u^{-1}v = w$.

For an alphabet A, a subset L of A^* is called a *language*. The concatenation of languages K and L, where $K, L \subseteq A^*$, is defined as the language

$$K \cdot L = KL = \{uv \mid u \in K, \; v \in L\} \subseteq A^*.$$

Now the concatenation is an associative binary operation on 2^{A^*}, and therefore 2^{A^*} is a monoid of sets, where $\{\varepsilon\}$ is the identity (and $\emptyset$ is the zero).

Finally, we mention that we shall denote the alphabets in this thesis by A, B, Σ, Δ, Γ and Θ, depending on the considerations.

1.3 Morphisms and finite substitutions

Let Σ and Δ be two alphabets. A mapping h from Σ^* into Δ^* is called a *(monoid) morphism* if, for all $u, v \in \Sigma^*$,

$$h(uv) = h(u)h(v).$$

It follows that a morphism becomes defined by the images of the letters of the domain alphabet Σ. Note that for all morphisms, the image of the empty word is the empty word. On the other hand, if $h(a) \neq \varepsilon$ for all $a \in \Sigma$, then h is said to be *nonerasing*.

A mapping $\varphi : \Sigma^* \to 2^{\Delta^*}$ is called a *substitution*, if

1. $\varphi(\varepsilon) = \{\varepsilon\}$ and

2. $\varphi(uv) = \varphi(u)\varphi(v)$ for all u and v in Σ^*.

These conditions mean that a substitution is a morphism from Σ^* into the monoid 2^{Δ^*}. A substitution φ is called *ε–free* if $\varepsilon \notin \varphi(a)$ for all $a \in \Sigma$, and *finite* if, for all $a \in \Sigma$, $\varphi(a)$ is a finite subset of Δ^*.

Let $L \subseteq \Sigma^*$ be a language. Two substitutions $\varphi, \xi : \Sigma^* \to 2^{\Delta^*}$ are *equivalent* on L if for all $w \in L$, $\varphi(w) = \xi(w)$.

The equivalence problem of two finite ε–free substitutions will be considered in Chapter 4.

1.4 Turing Machines

Turing machines were defined by Alan Turing in 1936 as a theoretical model for the notion of *algorithm*, that is, of an effective procedure that produces the value of a function f for each possible argument given as an input.

We shall give here a short account on the well known undecidability result that states that it is undecidable for a given Turing machine whether its computation halts on a given input word. For a more complete treatment the reader is referred to Rozenberg and Salomaa [21].

Our definition of a Turing machine is not the usual one. However, it is rather immediate that this definition of Turing machine has equivalent accepting power than the one given in the literature, see e.g. [21].

A *Turing machine*, TM for short, is a 8-tuple

$$\mathcal{M} = (Q, \Sigma, \Theta, \mathcal{R}, q_0, \#, X, q_h),$$

where Q is the finite set of *states*, Σ and Θ are the finite *input* and *tape alphabets*, respectively, with $\Sigma \subseteq \Theta$, $\mathcal{R}$ is a set of *transition rules*, $q_0 \in Q$ is the *initial state*, $\# \in \Theta$ is the *blank symbol*, X is the symbol of the left–boundary and $q_h \in Q$ is the *halting state*.

In the model that we consider, a Turing machine consists of a *finite control unit*, one *left bounded potentially infinite tape*, where the input is written, and one *read–write head*, which moves to both directions of the tape. The left–bound is denoted by the symbol $X \in \Theta$ and X appears only as this left–bound, which cannot be overwritten. Here the tape being potentially infinite means that the input can be arbitrarily long.

The transition rules in $\mathcal{R}$ are of the form

$$apb \to \begin{cases} qac & \text{for } a \neq X, \\ acq, & \\ aqc, & \end{cases} \tag{1.1}$$

where $a \in \Theta$, $b, c \in \Theta \setminus \{X\}$, $q \in Q$ and $p \in Q \setminus \{q_h\}$. A transition rule means that if the head scans b in the state p, the machine moves to the state q, replaces b with c, and moves the head one square to the left (the first case), to the right (the second case) or stays in the same square (the third case). Note that there is no rules in $\mathcal{R}$ having q_h on the left hand side.

Usually, a TM is defined with a *two–way* infinite tape and the the transitions are defined by a *transitions function* $\delta : Q \times \Theta \to Q \times \Theta \times \{L, R, S\}$, where L, R and S are the moving direction of a transition. Note that our definition of a Turing machine with the above transition rules is, in effect, a *semi–Thue system*, cf. [12, 21].

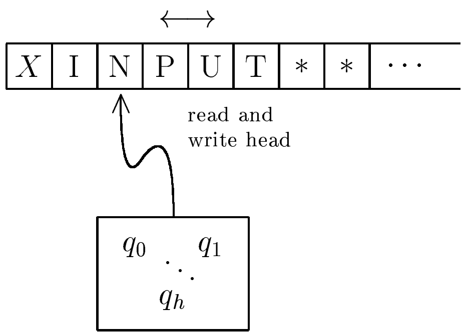

Figure 1: A Turing machine.

A *configuration* (or an *instantaneous description*) of a TM $\mathcal{M}$ is a word

$$X\alpha_1 q \alpha_2 \in X\Theta^* Q \Theta^+,$$

where $\alpha_1\alpha_2$ is a context of the tape. The initial configuration is $Xq_0w\#^k$, where w is the input word and some $k \in \mathbb{N}$. A *one step computation* or a *move* of $\mathcal{M}$ is defined as follows: Let $Xx_1 \ldots x_{i-1}px_ix_{i+1} \ldots x_n$ be the current configuration. For $x_{i-1}px_i \to qx_{i-1}y \in \mathcal{R}$ (where $i \neq 1$), we write

$$Xx_1 \ldots x_{i-1}px_ix_{i+1} \ldots x_n \vdash_{\mathcal{M}} Xx_1 \ldots x_{i-2}qx_{i-1}yx_{i+1} \ldots x_n,$$

for $x_{i-1}px_i \to x_{i-1}yq \in \mathcal{R}$, we write

$$Xx_1 \ldots x_{i-1}px_ix_{i+1} \ldots x_n \vdash_{\mathcal{M}} Xx_1 \ldots x_{i-2}x_{i-1}yqx_{i+1},$$

and, for $x_{i-1}px_i \to x_{i-1}qy$, we write

$$Xx_1 \ldots x_{i-1}px_ix_{i+1} \ldots x_n \vdash_{\mathcal{M}} Xx_1 \ldots x_{i-1}qyx_{i+1} \ldots x_n.$$

Note that there is no transition from the configuration Xuq, since our model of Turing machine cannot extend the tape. But if the computation leads to such configuration, we may of course add one blank symbol to the end already in the initial configuration.

Let $\vdash^*_{\mathcal{M}}$ (or $\vdash^*$ for short) be the reflexive and transitive closure of the relation $\vdash_{\mathcal{M}}$. Then the language *accepted* by $\mathcal{M}$ is

$$L(\mathcal{M}) = \left\{w \in \Sigma^* \mid \exists k \in \mathbb{N}\colon Xq_0w\#^k \vdash^*_{\mathcal{M}} Xq_hv \text{ with } v \in (\Sigma \cup \#)^+\right\}.$$

Note that although we think that the tape is finite we may add any number of blank symbols to the end of the input. Here the word v is sometimes thought as the result of the computation.

A language $L \subseteq \Sigma^*$ is called *recursively enumerable*, if there is a TM $\mathcal{M}$ such that $L(\mathcal{M}) = L$. Moreover, L is called *recursive* if there is a *halting* TM $\mathcal{M}$ such that $L = L(\mathcal{M})$ and $\mathcal{M}$ *halts on all input words* from Σ^*, meaning that the computations stop on every input. If the input word is not in L, then the computation stops on some state not q_h.

As mentioned earlier, Turing machines are used as a formalism for algorithms. A halting Turing machine $\mathcal{M}$ provides an algorithm to *decide* whether or not $w \in \Sigma^*$ is in $L(\mathcal{M})$, in a way that, if the computation on given input w ends in the configuration Xq_hv, then $w \in L(\mathcal{M})$, and if the computation ends in some other configuration, then $w \notin L(\mathcal{M})$. Here $L(\mathcal{M})$ is necessarily recursive, since we assumed that the computation halts on every input.

1.5 Decidability and undecidability

Let P be a well-defined decision problem, i.e. a set of instances each of which either has or does not have a certain property and we are asked to decide whether the instance given as an input has or does not have the property. We say that the problem P is *decidable*, if there exists an algorithm, which for every correct input of an instance of P terminates and gives an answer "yes", if the input has the required property and answers "no" otherwise. If no such algorithm exists, then P is said to be *undecidable*. To prove that a problem P is decidable, we must find an algorithm which decides P, or show that such an algorithm exists.

Note that here "P has an algorithm" means, by Church thesis, that there exists a halting TM $\mathcal{M}$, such that

$$L(\mathcal{M}) = \left\{ I \text{ is an instance of } P \mid Xq_0I\#^k \vdash_{\mathcal{M}}^* Xq_hv \text{ for some } k \in \mathbb{N} \right\} .$$

Here the instance I is encoded to the input alphabet in a specific way.

In this thesis we are mainly interested in the undecidability results. The usual method to prove undecidability of a problem P is to reduce some already known undecidable problem Q to P. This means that we transform effectively an arbitrary instance q of Q to some instance p of P, in a way that the condition "p has the property of P" is equivalent with the condition "q has the property of Q". It follows that if the problem P is decidable, then the instances q of the problem Q can be solved and therefore the problem Q becomes decidable, which leads to a contradiction.

We shall present a chain of undecidability results. We use a classical undecidable problem, the *halting problem for Turing machines*, stated in the next theorem, as a base of this chain. Our form of the halting problem is not the usual one, since our definition of Turing machine is not quite orthodox.

Moreover, let $\mathcal{M} = (Q, \Sigma, \Theta, \mathcal{R}, q_0, \#, X, q_h)$ be a Turing machine. For any correct input $w \in \Sigma^*$ of $\mathcal{M}$ we may easily construct a TM $\mathcal{M}'$ such that it begins with empty tape, writes w to the tape and then simulates $\mathcal{M}$ on the input w. Assume that q_0' is the initial state of $\mathcal{M}'$. It follows that, for some $k, \ell \in \mathbb{N}$,

$$Xq_0'\#^k \vdash_{\mathcal{M}'}^* Xq_0w\#^\ell \vdash_{\mathcal{M}'}^* Xq_hu \iff Xq_0w\#^\ell \vdash_{\mathcal{M}}^* Xq_hu.$$

In the original halting problem we are asked to decide whether a Turing machine halts on a input word, but based on the above considerations, we state

Theorem 1.1. *It is undecidable for a TM whether it halts on the empty tape, i.e. whether, for some $k \in \mathbb{N}$ and $u \in \Theta$,*

$$Xq_0\#^k \vdash^* Xq_h u.$$

This is the form of the halting problem we shall use in the next chapter. Note that we may assume that the tape is "physically" left bounded, not only marked by some symbol.

For the proof of the halting problem, see for example Rozenberg and Salomaa [21].

1.6 Finite automata

We conclude this chapter with the definition of a finite nondeterministic automaton, which is a restricted machine model for accepting languages. As a general reference for finite automata we give Eilenberg [5].

A *finite automaton*, FA for short, is a quintuple

$$\mathcal{A} = (Q, A, \delta, I, F),$$

where Q is a *finite* set of *states*, A is a finite *input alphabet*, δ is a *transition function*,

$$\delta : Q \times A \to 2^Q,$$

I is the set of *initial states* and F is the set of *final states*. A transition $p \in \delta(q, a)$, where $p, q \in Q$ and $a \in A$, will also be written as (q, a, p), in which case $\delta \subseteq Q \times A \times Q$ is regarded as a relation (and sometimes also as an alphabet).

A *path* (or a *computation*) π of $\mathcal{A}$ is a sequence

$$\pi = (q_1, a_1, q_2)(q_2, a_2, q_3) \cdots (q_n, a_n, q_{n+1}), \tag{1.2}$$

where $(q_i, a_i, q_{i+1}) \in \delta$ for all $1 \leq i \leq n$. If we think δ as an alphabet, then we can write $\pi \in \delta^*$. If $q_1 = q_{n+1}$, then π is called a *cycle*. A cycle (q, a, q) is called a *loop*.

The *label* of the path π is the word $\|\pi\| = a_1 \ldots a_n$. A path is called *successful* (or *accepting*) or we say that $\mathcal{A}$ *accepts* the word $\|\pi\|$, if $q_1 \in I$ and $q_{n+1} \in F$ in (1.2). The language accepted by $\mathcal{A}$ (or the *behaviour* of $\mathcal{A}$) is the subset $L(\mathcal{A})$ of A^* consisting of the labels of the successful paths of $\mathcal{A}$. Actually, $\| \cdot \|$ is a morphism from δ^* into A^*, defined by $\|(q, a, p)\| = a$, for all $(q, a, p) \in \delta$, and

$$L(\mathcal{A}) = \{w \in A^* \mid w = \|\pi\|,\ \pi \in \delta^* \text{ is a successful path in } \mathcal{A}, \text{ and}$$
$$w = \varepsilon \text{ if } I \cap F \neq \emptyset\}.$$

A transition $(p, a, q) \in \delta$ is also denoted by $p \xrightarrow{a} q$. A path π in (1.2) can be written in the form

$$\pi = q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \cdots \xrightarrow{a_{n-1}} q_n \xrightarrow{a_n} q_{n+1}.$$

This notation of transitions leads to the *graph presentation* of the finite automata, where Q is a set of *vertices* and there is an *labeled directed edge* from the vertex p to the vertex q with label a, if (p, a, q) is in δ.

An automaton $\mathcal{A}$ is said to be *deterministic*, if δ is a partial function from $Q \times A$ into Q. This means that for all $q \in Q$ and $a \in A$, there is at most one $p \in Q$ such that (q, a, p) is in δ. If this is not the case, then $\mathcal{A}$ is called *nondeterministic*. We shall also use the notation DFA for a deterministic and NFA for a nondeterministic finite automaton.

A language $L \subseteq A^*$ is called *regular* if there is an FA $\mathcal{A}$ such that L is the behaviour of $\mathcal{A}$, i.e. if $L(\mathcal{A}) = L$. We shall denote the family of regular languages by Reg. Note that $\mathcal{A}$ can also be nondeterministic, since we know that each language accepted by an NFA, is also accepted by a DFA. This result is very basic in the formal language theory, see Salomaa [22].

Let $\mathcal{A} = (Q, A, \delta, I, F)$ be a finite automaton. There exists a finite automaton $\mathcal{A}' = (Q \cup \{q_0\}, A, \delta', q_0, F')$ with only one initial state such that $L(\mathcal{A}') = L(\mathcal{A})$. Such an $\mathcal{A}'$ can be constructed for $\mathcal{A}$ as follows: for all $a \in A$,

$$\delta'(q_0, a) = \{p \mid p \in \delta(q, a) \text{ for some } q \in I\},$$

$\delta'(q, a) = \delta(q, a)$ for all $q \in Q$ and $a \in A$ and

$$F' = \begin{cases} F \cup \{q_0\} & \text{if } I \cap F \neq \emptyset, \\ F & \text{otherwise.} \end{cases}$$

There exists a same kind of procedure for nonempty languages to construct an FA with only one final state, if $\varepsilon \notin L(\mathcal{A})$, and with two final states, if $\varepsilon \in L(\mathcal{A})$, and accepting the same language $L(\mathcal{A})$.

A finite automaton is said to be *complete*, if for all input words there is a path in the automaton. It is known that, for every regular language L, there is a complete (nondeterministic) FA $\mathcal{A}$ with only one initial and one (or two) final states such that $L(\mathcal{A}) = L$.

If all states of an FA $\mathcal{A} = (Q, A, \delta, q_0, F)$ are final, i.e. $F = Q$, then we say that $\mathcal{A} = (Q, A, \delta, q_0)$ is a finite automaton *without final states*. Such automata are used as a base of the weighted finite automata defined later in Chapter 3.

2 A modification of the PCP

In this chapter we shall define a modification of the famous undecidability result called *Post Correspondence Problem*, the PCP for short.

In the PCP we are given two morphisms $h, g : \Sigma^* \to \Delta^*$ and it is asked whether their *equality set*

$$E(h,g) = \{w \in \Sigma^+ \mid h(w) = g(w)\}$$

is empty or not.

Theorem 2.1. *The Post Correspondence Problem is undecidable.*

The undecidability of the PCP is a classical result in formal language theory, and it was first proved by Post in [20]. For the proof, see Rozenberg and Salomaa [21].

The *size* of an instance (h, g) of the PCP is defined to be the cardinality of the alphabet Σ. We denote by PCP(n) the subproblem of the PCP for instances of size at most n.

It is also known that if $n \leq 2$, then PCP(n) is decidable, and if $n \geq 7$ then the problem is undecidable. For the proof of the undecidability of PCP(7), see Matiyasevich and Senizerques [18], and for the proof of the decidability of PCP(2), see Ehrenfeucht, Karhumäki and Rozenberg [4]. For n greater than 2 and smaller than 7 the decidability status of the PCP is open.

2.1 The prefix PCP

In this section we shall consider a modification of the PCP, called a *prefix PCP*. We need the undecidability of this problem in the next chapter.

In the undecidability proof we use the halting problem of Turing machines on the empty tape, namely Theorem 1.1. Recall that in Theorem 1.1 we assumed that the tape is left bounded, X is the symbol of the left boundary, we have only one final state q_h and if the machine halts on a given input, it halts on the configuration Xq_hw for some w.

Let $\mathcal{M}$ be a Turing machine, $\mathcal{R}$ be the set of its transition rules and Θ be the tape alphabet. For this given Turing machine $\mathcal{M}$, define two morphisms h and g in a standard way: here a, b and c are new letters.

letter	h	g
a	$\#^2$	$\#$
b	Xq_0	$\#^2$
$x \in \Theta$	x	x
$r\colon u \to v \in \mathcal{R}$	v	u
c	$\#^2$	Xq_h

Here $\#$ is the blank symbol and X is the symbol for the left boundary of the tape. Both $\#$ and X are in Θ.

Denote

$$B = \{b\} \cup \mathcal{R} \cup \Theta,$$

and

$$A = B \cup \{a, c\}.$$

Notice that $|g(w)| \leq |h(w)|$ for all words $w \in A^*$, since our rules are length preserving. Further, let

$$\beta(w) = g(w)^{-1}h(w)$$

be the *overflow* of the word w whenever $g(w) < h(w)$.

Lemma 2.2. *The Turing machine $\mathcal{M}$ halts on the empty tape if and only if there exists a word $w \in a^+B^*c$ such that $g(w) < h(w)$.*

Proof. Assume that there is a halting computation π in $\mathcal{M}$ on the empty tape. Therefore π is of the form

$$\begin{aligned} Xq_0\#^{s-2} &= z_1u_1z_1' \vdash z_1v_1z_1' = z_2u_2z_2' \vdash z_2v_2z_2' = z_3u_3z_3' \vdash \cdots \\ &\vdash z_{m-1}v_{m-1}z_{m-1}' = z_mu_mz_m' \vdash z_mv_mz_m' = Xq_hz, \end{aligned} \tag{2.1}$$

where $s \geq 3$, $|z_iu_iz_i'| = |z_iv_iz_i'| = s$ and $r_i : u_i \to v_i \in \mathcal{R}$ are the rules used, for $1 \leq i \leq m$.

Let

$$w = a^sb\#^{s-2}z_1r_1z_1'z_2r_2z_2' \ldots z_{m-1}r_{m-1}z_{m-1}'z_mr_mz_m'c \in a^+B^*c.$$

By the definitions of h and g,

$$\begin{aligned} h(w) &= \#^{2s}Xq_0\#^{s-2}z_1v_1z_1'z_2v_2z_2' \ldots z_{m-1}v_{m-1}z_{m-1}'Xq_hz\#^2, \\ g(w) &= \#^s\#^2\#^{s-2}z_1u_1z_1'z_2u_2z_2' \ldots z_{m-1}u_{m-1}z_{m-1}'z_mu_mz_m'Xq_h. \end{aligned}$$

By the equalities in (2.1),

$$g(w) = \#^{2s}Xq_0\#^{s-2}z_1v_1z_1'z_2v_2z_2' \ldots z_{m-1}v_{m-1}z_{m-1}'Xq_h,$$

and we get that $g(w) < h(w)$.

Assume next that $g(w) < h(w)$ for some $w \in a^+B^*c$. Note that $w \notin a^+c$, since $g(z) \not< h(z)$ for all $z \in a^+c$.

In order that $g(w) < h(w)$, the word w must start with $w_1 = a^s\#^jb\#^{s-2}$ for some $s > 0$ and any $j \geq 0$. For this prefix we have $\beta(w_1) = Xq_0\#^{s-2}$, which corresponds to the initial configuration of the Turing machine $\mathcal{M}$ for some s. We shall define a sequence w_i, $i \in \mathbb{N}$, of the necessary prefixes of w.

The only way to obtain $Xq_0\#^{s-2}$ as an image of g is to have $z_1 r_1 z_1'$ as a factor of w, where $r : Xq_0\# \to v_1 \in \mathcal{R}$, $z_1 = \varepsilon$ and $z_1' = \#^{s-3}$. Let $w_2 = w_1 z_1 r_1 z_1'$. It follows that $\beta(w_2) = z_1 v_1 z_1'$. Now if Xq_h is not a prefix of this overflow, we must continue by $z_2 r_2 z_2'$, where $r : u_2 \to v_2 \in \mathcal{R}$, $z_1 v_1 z_1' = z_2 u_2 z_2'$. Define $w_3 = w_2 z_2 r_2 z_2'$.

We continue this procedure, and define $w_i = w_{i-1} z_{i-1} r_{i-1} z_{i-1}'$ until for some m, $\beta(w_m) = z_m v_m z_m' = Xq_h z$, $z \in \Theta^*$. Note that this m is unique, since there is no transitions from q_h, and necessarily $w = w_m c$, since the procedure itself is unique. It follows now that

$$Xq_0\#^{s-2} \vdash^* Xq_h z$$

in $\mathcal{M}$, where r_i, for $1 \leq i \leq m$, is the sequence of the rules used in this halting computation. $\square$

Note that the weight $h(c) = \#^2$ is superfluous in the above construction. What matters is that g writes Xq_h at some point – and if this occurrence matches that written earlier by h, then $\mathcal{M}$ has entered the halting state in a proper computation.

As a corollary to the above, we state a rather general undecidability result for instances of morphisms.

Theorem 2.3. *It is undecidable for an instance (h, g) of two nonerasing morphisms whether there exists a word $w \in a^+B^*c$ such that $g(w) < h(w)$, where $h(a) \in g(a)^*$, and $|g(x)| = |h(x)|$ for all $x \in B \cup \{c\}$.*

We shall refer to this modification of the PCP as the *prefix PCP*.

Note that the previous theorem does not imply that it is undecidable for two given morphisms $h, g; \Sigma^* \to \Delta^*$, whether there exists a word $w \in \Sigma^*$ such that $g(w) < h(w)$. For example, for (h, g) in the proof of Lemma 2.2, $g(a^k) < h(a^k)$ for all $k \in \mathbb{N}$.

Indeed, the existence of a word w such that $g(w) < h(w)$ is trivially decidable. Therefore the fixed ending with c is essential in the above undecidability result.

3 Integer weighted finite automata

In this chapter we shall define a restricted type of finite automata with integer weights on the transitions and prove a simple undecidability property for such automata. The type of automata we consider is closely related to the 1-turn one–counter automata as considered by Baker and Book [2], Greibach [6], and especially by Ibarra [11]. In our model the counter is replaced by a weight function of the transitions, and while doing so, the finite automaton becomes independent of the counter.

3.1 Definitions

We shall first give a general definition of a weighted finite automaton that is a restricted case of an extended finite automaton of Mitrana and Stiebe [19].

Consider a (nondeterministic) finite automaton $\mathcal{A} = (Q, A, \delta, q_0)$ *without final states* with the states Q, the alphabet A, the set of transitions $\delta \subseteq Q \times A \times Q$ and the initial state q_0.

We shall redefine the transitions using a *set of edges* $T = \{t_1, t_2, \ldots, t_m\}$ and a transition function $\sigma\colon T \to \delta$. We do this to allow transitions t_i and t_j, where $i \neq j$, but $\sigma(t_i) = \sigma(t_j)$. In other words, there may exist many copies of one transition in this new set of edges. Clearly this new definition of the transitions does not affect the language accepted by the automaton.

Let G, or $(G, \cdot, \iota)$, be a group with identity ι. A (G-)*weighted finite automaton* $\mathcal{A}^\gamma$ consists of a finite automaton $\mathcal{A} = (Q, A, \sigma, q_0)$ as above, and a *weight function* $\gamma\colon\ \{t_1, \ldots, t_n\} \to G$ of the edges. To simplify the notation, we shall write the edges in the form

$$t = \langle q, a, p, z \rangle$$

if $\sigma(t) = (q, a, p)$ and $\gamma(t) = z$. Similarly, we shall write the transition function σ as a set, $\sigma \subseteq Q \times A \times Q \times \mathbb{Z}$, where

$$\sigma = \{\langle q, a, p, z \rangle \mid \exists t \in T\colon\ \sigma(t) = (q, a, p) \text{ and } \gamma(t) = z\}\,.$$

In the figures we shall denote such an edge t by

$$q \xrightarrow{(a,z)} p\,.$$

Note that the function γ does not affect the computations of the finite automaton $\mathcal{A}$.

Define a morphism $\|\cdot\|\colon T^* \to A^*$ by setting $\|t\| = a$ if $\sigma(t) = (q, a, p)$. Let $\pi = t_{i_0} t_{i_1} \dots t_{i_n}$ be a path of $\mathcal{A}$, where $\sigma(t_{i_j}) = (q_{i_j}, a_j, q_{i_{j+1}})$ for $0 \leq j \leq n-1$. The *weight of the path* π is the element

$$\gamma(\pi) = \gamma(t_{i_0})\gamma(t_{i_1}) \cdots \gamma(t_{i_n}) \in G\,.$$

Moreover, we let $L(\mathcal{A}^\gamma) = \|\gamma^{-1}(\iota)\|$, that is,

$$L(\mathcal{A}^\gamma) = \{w \in A^* \mid w = \|\pi\|,\ \gamma(\pi) = \iota\}\,,$$

be the language *accepted by* $\mathcal{A}^\gamma$.

A *configuration* of $\mathcal{A}^\gamma$ is any triple $(q, w, g) \in Q \times A^* \times G$.

A configuration (q, aw, g_1) is said to *yield* in a configuration $(p, w, g_1 g_2)$, denoted by

$$(q, aw, g_1) \models_{\mathcal{A}^\gamma} (p, w, g_1 g_2)\,,$$

if there is an edge t such that $\sigma(t) = (q, a, p)$ with $\gamma(t) = g_2$. Let $\models^*_{\mathcal{A}^\gamma}$, or simply $\models^*$ if $\mathcal{A}^\gamma$ is clear from the context, be the reflexive and transitive closure of the relation $\models_{\mathcal{A}^\gamma}$.

We shall restrict to the case, where the group of the automaton is the additive group of integers, namely $(\mathbb{Z}, +, 0)$. Such automata are called *integer weighted finite automata* and denoted by FA($\mathbb{Z}$).

As mentioned, our definition of the integer weighted finite automata is a restricted case of the extended finite automata of Mitrana and Stiebe [19]. In the extended finite automata the underlying automata has final states $F \subseteq Q$ and the transitions reading the empty word are allowed.

An FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ is said to be *unimodal* (or *1–turn*) if for all paths in $\mathcal{A}^\gamma$ the weights of the edges are first strictly positive, then 0, then strictly negative and finally 0.

To be more precise a weighted $\mathcal{A}^\gamma$ is unimodal, if every path π (not only those that have weight 0) can be partitioned into $\pi = \pi_1\pi_2\pi_3\pi_4$, where the edges in π_1 have positive weight, those in π_3 have negative weight, and those in π_2 and π_4 have zero weight, see Figure 2. Any of these subpaths π_i may also be trivial, for example, we may have only zero weights or only strictly negative weights in our path.

The requirement made for all paths in the unimodality condition is clearly a structural restriction of an FA($\mathbb{Z}$) $\mathcal{A}^\gamma$, and therefore it is not an additional condition imposed on the weight functions, or on the acceptance of a word.

Let $\mathcal{A}^\gamma$ be an FA($\mathbb{Z}$). The empty word is always included in $L(\mathcal{A}^\gamma)$, since in an integer weighted finite automaton all states, including the initial state, are final. Therefore we known that not all regular languages can be accepted by an FA($\mathbb{Z}$). On the other hand,

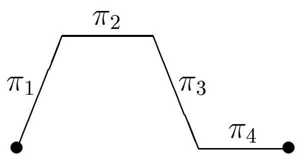

Figure 2: The behaviour of the weights in a unimodal FA($\mathbb{Z}$).

Theorem 3.1. *For each regular language L, there exists an* FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ *such that* $L(\mathcal{A}^\gamma) = L \cup \{\varepsilon\}$.

Proof. Let L be a regular language and let $\mathcal{A} = (Q, A, \delta, q_0, F)$ be a (nondeterministic) finite automaton such that $L(\mathcal{A}) = L$, where $\delta \subseteq Q \times A \times Q$. We may assume that $\mathcal{A}$ has one initial state q_0 and one final state q_f (or two final states q_0 and q_f, if $\varepsilon \in L$) such that there are no transitions to q_0 and no transitions from q_f. It is well known that such an $\mathcal{A}$ exists for all regular languages L.

We shall define an FA($\mathbb{Z}$) $\mathcal{A}^\gamma = (Q, A, \sigma, q_0)$, where σ is a bijection and therefore we may define $T = \delta$. One such required weight function γ is defined, for $(p, a, q) \in \delta$, by

$$\gamma(p,a,q) = \begin{cases} 0 & \text{if } p = q_0 \text{ and } q = q_f, \\ 1 & \text{if } p = q_0 \text{ and } q \neq q_f, \\ -1 & \text{if } p \neq q_0 \text{ and } q = q_f, \\ 0 & \text{if } p \neq q_0 \text{ and } q \neq q_f. \end{cases} \tag{3.1}$$

It is obvious that $\gamma(\pi) = 0$ only in the case where π is an accepting path of $\mathcal{A}$. This proves our claim. $\square$

Note that actually the above definition of γ defines a unimodal FA($\mathbb{Z}$) that accepts $L \cup \{\varepsilon\}$.

As we mentioned, not all regular languages are accepted by a (unimodal) FA($\mathbb{Z}$). On the other hand, it is easy to show that not all languages accepted by integer weighted finite automata are regular, since the language

$$L = \{a^n b^n \mid n \geq 0\}$$

is accepted by a unimodal FA($\mathbb{Z}$) of Figure 3, but L is not regular.

Our main result in this chapter is the undecidability of the *universe problem* for the unimodal integer weighted automata. In other words, we

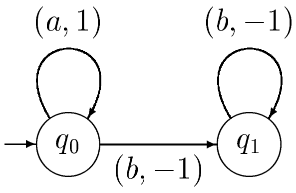

Figure 3: A unimodal FA($\mathbb{Z}$) accepting the language $\{a^n b^n \mid n \geq 0\}$.

shall prove that it is undecidable for a given unimodal FA($\mathbb{Z}$) $\mathcal{A}^\gamma$, whether $L(\mathcal{A}^\gamma) = A^*$.

The undecidability property of the universe problem for the unimodal weighted finite automata turns out to hold in a more restrictive case. Indeed, we shall show that it is sufficient to have only four states, and that the finite automata are *acyclic*, that is, the states can be ordered, $Q = \{q_0, q_1, \ldots, q_s\}$, in such a way that $(q_i, a, q_j) \in \delta$ implies $i \leq j$.

3.2 The universe problem

In this section we shall consider the universe problem for the integer weighted finite automata.

Theorem 3.2. *The universe problem is undecidable for 4-state unimodal integer weighted finite automata.*

Proof. Let h and g be two nonerasing morphisms from A^* as constructed in Section 2.1, where $A = \{a, c\} \cup B$, $h(a) \in g(a)^*$ and $|g(y)| = |h(y)|$ for all $y \in B \cup \{c\}$.

We shall construct an FA($\mathbb{Z}$) $\mathcal{A}^\gamma$, where $= (Q, A, \sigma, q_0)$ such that $L(\mathcal{A}^\gamma) \neq A^*$ if and only if there exists a word $w \in a^+B^*c$ with $g(w) < h(w)$. For this we must show that $\mathcal{A}^\gamma$ accepts all words for which

1. w in a^+B^*c such that $g(w) \not< h(w)$ and
2. in $A^* \setminus a^+B^*c$.

Case 1 is equivalent to the fact that $w \in a^+B^*c$ and for some r, $1 \leq r \leq |g(w)|$, $g(w)_r \neq h(w)_r$. A picture of the full $\mathcal{A}^\gamma$ is given at the end of the proof, see Figure 4 on page 25.

Let $A = \{a_1, a_2, \ldots, a_{m-2}\}$, $Q = \{q_0, q_1, q_2, q_3\}$, and note that by the construction of Section 2.1, if y is a letter in A then $|h(y)| \leq 3$. We shall first define the edges for the acceptance of the words of Case 1. For $a(= a_k)$, define the edges

$$\langle q_0, a, q_0, 5m\rangle\,, \qquad \langle q_0, a, q_2, 5m(-|g(a)|+3)-5k+1\rangle\,,$$

and for all $a_k \in B$, define the edges

$$\begin{aligned}
&\langle q_0, a_k, q_1, 15m+1\rangle\,, && \langle q_0, a_k, q_2, 5m(-|g(a_k)|+3)-5k+1\rangle\,,\\
&\langle q_1, a_k, q_1, 0\rangle\,, && \langle q_1, a_k, q_2, -5m|g(a_k)|-5k\rangle\,,\\
&\langle q_0, a_k, q_3, 5m(i-j)\rangle\,, && \langle q_1, a_k, q_3, 5m(i-j-3)-1\rangle\,,
\end{aligned}$$

where $1 \le i \le |h(a_k)|$, $1 \le j \le |g(a_k)|$ and $h(a_k)_i \ne g(a_k)_j$. For c, define the edges

$$\langle q_1, c, q_3, 5m(i-j-3)-1\rangle\,,$$

where $1 \le i \le |h(c)|$, $1 \le j \le |g(c)|\}$ and $h(c)_i \ne g(c)_j$. Finally, for all $y \in A$, define the edges

$$\begin{aligned}
&\langle q_3, y, q_3, 0\rangle\,, && \langle q_2, y, q_2, -5m|g(y)|\rangle\,,\\
&\langle q_2, y, q_3, 5m(i-j-3)+5\ell-1\rangle\,,
\end{aligned}$$

where $1 \le \ell \le m-2$, $1 \le i \le |h(a_\ell)|$, $1 \le j \le |g(y)|$ and $h(a_\ell)_i \ne g(y)_j$.

We denote by $L(q_i)$ the subset of $L(\mathcal{A}^\gamma)$ that is accepted with zero weight in state q_i, for $i = 0, 1, 2, 3$.

It is clear by the definitions that $L(q_0) = \{\varepsilon\}$ and that $L(q_1) = \emptyset = L(q_2)$, since the paths that end in q_1 or q_2 have weights 1 modulo 5, and thus certainly the weights are nonzero.

Next we shall consider the language $L(q_3)$.

Claim. Let w be a word in a^+B^*c. Then $w \in L(q_3)$ if and only if $g(w) \not< h(w)$.

Proof. Assume $w \in a^+B^*c$ is such that $g(w) \not< h(w)$. This means that for some r, $1 \le r \le |g(w)|$, $h(w)_r \ne g(w)_r$.

We have two possibilities.

(a) Let $a_k \in B \cup \{c\}$, and

$$\begin{aligned}
w = ua_kv \text{ with } & |h(u)| < r \le |h(ua_k)| \text{ and}\\
& |g(u)| < r \le |g(ua_k)|.
\end{aligned}$$

Then for some $1 \le i \le |h(a_k)|$ and $1 \le j \le |g(a_k)|$,

$$|h(u)| + i = r = |g(u)| + j. \tag{3.2}$$

Note that $a_k \neq a$, since $h(a) \in g(a)^*$.

(b) Let $a_k \in B \cup \{a\}$, $y \in A$, and

$$w = ua_kvyz \text{ with } |h(u)| < r \leq |h(ua_k)| \text{ and } |g(ua_kv)| < r \leq |g(ua_kvy)|.$$

Then for some $1 \leq i \leq |h(a_k)|$ and $1 \leq j \leq |g(y)|$,

$$|h(u)| + i = r = |g(ua_kv)| + j. \tag{3.3}$$

If we count the difference of the images by these morphisms step by step for all prefixes of w, we notice that when reading the symbol a, the difference increases by 1, and when reading symbols from $B \cup \{c\}$ the difference does not change. It follows that the above factorization is unique for a given r.

In both of these cases, $u \in a^+B^*$, i.e. $u = ef$, where $e \in a^+$ and $f \in B^*$. By the definitions of h and g this implies that $|h(u)| - |g(u)| = |e|$. Note also that the words $w \in a^+c$ belong to the case (b).

In the case (a) there are two possibilities depending on whether $f = \varepsilon$ or not. If $f = \varepsilon$ then our path goes from q_0 straight to q_3. If $f \neq \varepsilon$ then $u = ef$ is read in a way that ends up in the state q_1, then goes to the state q_3 while reading a_k and finally reads v in q_3. The weight of such a path is, for i and j as in (3.2), in the case $f = \varepsilon$,

$$5m|e| + 5m(i - j) = 5m(|e| + i - j) = 5m(|h(u)| + i - |g(u)| - j) = 0,$$

and, in the case $f \neq \varepsilon$,

$$\begin{aligned} &5m|e| + 15m + 1 + 5m(i - j - 3) - 1 = 5m(|e| + i - j) \\ &= 5m(|h(u)| + i - |g(u)| - j) = 0. \end{aligned}$$

Therefore the path has weight 0.

In the case (b) we have two subcases. First, if $f = \varepsilon$, then we end up in the state q_0 by reading u. Second, if $f \neq \varepsilon$ we end up in the state q_1 by reading u; then we move to the state q_2 by reading a_k, read v in q_2, go to the state q_3 while reading y with the weight $5m(i - j - 3) + 5k - 1$, for i and j as in (3.3), and finally z is read in q_3. The weight of the path of the first subcase is

$$\begin{aligned} &5m|u| + 5m(-|g(a_k)| + 3) - 5k + 1 - 5m|g(v)| + 5m(i - j - 3) + 5k - 1 \\ &= 5m(|u| - |g(a_kv)| + i - j) = 5m(|h(u)| + i - |g(ua_kv)| - j) = 0. \end{aligned}$$

In the second subcase our path has the weight

$$\begin{aligned}&5m|e| + 15m + 1 - 5m|g(a_k)| - 5k - 5m|g(v)| + 5m(i - j - 3) + 5k - 1\\&= 5m(|e| - |g(a_k v)| + i - j) = 5m(|h(u)| + i - |g(ua_k v)| - j) = 0.\end{aligned}$$

We have proved that words w in a^+B^*c with $g(w) \not< h(w)$ are accepted by our automaton.

To prove the claim in the other direction, we show that for all $w \in A^*$ accepted in q_3, there exists an r, $1 \le r \le |g(w)|$ such that $h(w)_r \neq g(w)_r$.

Assume that $w \in A^*$ is accepted in q_3. Then we have four cases.

(1) If the accepting path of w goes from q_0 straight to q_3. Then $w = ea_k v$, where $e \in a^*$, $a_k \in B$ and $v \in A^*$, and this path gives

$$\begin{aligned}(q_0, w, 0) &\models^* (q_0, a_k v, 5m|e|) \models (q_3, v, 5m|e| + 5m(i - j))\\&\models^* (q_3, \varepsilon, 5m(|e| + i - j)),\end{aligned}$$

where $1 \le i \le |h(a_k)|$, $1 \le j \le |g(a_k)|$, and $h(a_k)_i \neq g(a_k)_j$. Since the weight of this path is 0, and $|h(e)| - |g(e)| = |e|$, we get that $5m(|h(e)| + i - |g(e)| - j) = 0$, and therefore $|h(e)| + i = r = |g(e)| + j$, which shows that $g(w) \not< h(w)$

(2) If the accepting path of w goes from q_0 to q_1 and from there to q_3, then $w = edfa_k v$, where $e \in a^*$, $d \in B$, $f \in B^*$, $a_k \in A$ and $v \in A^*$, and this path gives

$$\begin{aligned}(q_0, w, 0) &\models^* (q_0, dfa_k v, 5m|e|) \models (q_1, fa_k v, 5m|e| + 15m + 1)\\&\models^* (q_1, a_k v, 5m|e| + 15m + 1)\\&\models (q_3, v, 5m|e| + 15m + 5m(i - j - 3) - 1)\\&\models^* (q_3, \varepsilon, 5m|e| + 5m(i - j)),\end{aligned}$$

where $1 \le i \le |h(a_k)|$, $1 \le j \le |g(a_k)|$, and $h(a_k)_i \neq g(a_k)_j$. Since the weight of this path is 0, and $|h(edf)| - |g(edf)| = |e|$, we get that $5m(|h(edf)| + i - |g(edf)| - j) = 0$, and therefore again $|h(edf)| + i = r = |g(edf)| + j$, and so $g(w) \not< h(w)$.

(3) If the accepting path goes from q_0 to q_2 and from there to q_3 then

$w = ua_kvyz$, where $u \in a^*$, $a_k \in B \cup \{a\}$ and $y \in A$, and this path gives

$$\begin{aligned}(q_0, w, 0) &\models^* (q_0, a_kvyz, 5m|u|)\\ &\models (q_2, vyz, 5m|u| + 5m(-|g(a_k)| + 3) - 5k + 1)\\ &\models^* (q_2, yz, 5m(|u| - |g(a_k)| + 3) - 5k + 1 - 5m|g(v)|)\\ &\models (q_3, z, 5m(|u| - |g(a_kv)| + 3) - 5k + 1 + 5m(i - j - 3) + 5\ell - 1)\\ &\models^* (q_3, \varepsilon, 5m(|u| + i - |g(a_kv)| - j) + 5(\ell - k)),\end{aligned}$$

where $\ell \in \{1, \dots, m-2\}$, $1 \le i \le |h(a_\ell)|$, $1 \le j \le |g(y)|$, and $h(a_\ell)_i \ne g(y)_j$. Since the path is accepting, necessarily $5(\ell - k) \equiv 0 \pmod{5m}$, which implies that $k = \ell$, and $|u| + i - |g(a_kv)| - j = |h(u)| - |g(u)| + i - |g(a_kv)| - j = 0$ and therefore $|h(u)| + i = r = |g(ua_kv)| + j$, and so $g(w) \not< h(w)$.

(4) If the path goes through both q_1 and q_2. Then $w = udva_kzyf$, where $u \in a^*$, $v \in B^*$, $d, a_k \in B$, $y \in A$ and $z, f \in A^*$, and this path gives

$$\begin{aligned}(q_0, w, 0) &\models^* (q_0, dva_kzyf, 5m|u|) \models (q_1, va_kzyf, 5m|u| + 15m + 1)\\ &\models^* (q_1, a_kzyf, 5m|u| + 15m + 1)\\ &\models (q_2, zyf, 5m|u| + 15m + 1 - 5m|g(a_k)| - 5k)\\ &\models^* (q_2, yf, 5m(|u| - |g(a_k)|) + 15m - 5k - 5m|g(z)| + 1)\\ &\models (q_3, f, 5m(|u| - |g(a_kz)|) + 15m + 1 - 5k + 5m(i - j - 3) + 5\ell - 1)\\ &\models^* (q_3, \varepsilon, 5m(|u| - |g(a_kz)| + i - j) + 5(\ell - k)),\end{aligned}$$

where $\ell \in \{1, \dots, m-2\}$, $1 \le i \le |h(a_\ell)|$, $1 \le j \le |g(y)|$, and $h(a_\ell)_i \ne g(y)_j$. Since the weight of the path is 0, we get that $5(\ell - k) \equiv 0 \pmod{5m}$ and therefore $k = \ell$. Since $|u| = |h(udv)| - |g(udv)|$, we have that $|u| + i - |g(a_kz)| - j = |h(udv)| + i - |g(udva_kz)| - j = 0$, and $|h(udv)| + i = r = |g(udva_kz)| + j$. Again, $g(w) \not< h(w)$. This completes the proof of the present claim.

We have shown that $w \in a^+B^*c$ is accepted if and only if $g(w) \not< h(w)$. This implies that $a^+B^*cA^* \subseteq L(\mathcal{A}^\gamma)$ if and only if for all $w \in a^+B^*c$, $g(w) \not< h(w)$, since the automaton can read any word in A^* while in the state q_3. There are also other words accepted in q_3, but by the proof of the claim, we know that all accepted words w do satisfy the condition $g(w) \not< h(w)$.

It remains to make sure that all words of the case 2 are accepted by the (final) $\mathcal{A}^\gamma$, i.e. that the words not in $a^+B^*cA^*$ should also be accepted by our automaton. This is done by introducing new edges. To get the languages BA^* and cA^* accepted in the state q_3, we define, for $x \in B$,

$$\langle q_0, x, q_3, 0\rangle \qquad \text{and} \qquad \langle q_0, c, q_3, 0\rangle\,.$$

To get the language $a^+(B \cup \{a\})^*$ accepted in the state q_1, we define the edges

$$\langle q_0, a, q_1, 0\rangle\,, \qquad \text{and} \qquad \langle q_1, a, q_1, 0\rangle\,.$$

For the language $a^+(B \cup \{a\})^* BaA^*$, we define new edges, for $x \in B$,

$$\langle q_0, a, q_1, 4\rangle\,, \qquad \langle q_1, x, q_2, -2\rangle \qquad \text{and} \qquad \langle q_2, a, q_3, -2\rangle\,.$$

Then the language is accepted in q_3, since for $w = avdaz$, where $v \in (B \cup a)^*$, $d \in B$ and $z \in A^*$, we have

$$\begin{aligned}(q_0, w, 0) &\models (q_1, vdaz, 4) \models^* (q_1, daz, 4) \models (q_2, az, 4-2) \models (q_3, z, 2-2)\\ &\models^* (q_3, \varepsilon, 0).\end{aligned}$$

Therefore all words not in $a^+B^*cA^*$ are accepted by the automaton.

It now remains to be shown that these new edges do not interfere with the ones defined earlier for case 1, that is, we show that no words from a^+B^*c are accepted with a path, which uses these new edges together with the edges defined earlier.

It is obvious that in q_0 or q_1 no such thing can happen. In q_2 and q_3 the weight of the path is always of the form $5mr + 5s + t$, where $r, s, t \in \mathbb{Z}$, $-(m-2) \le s \le m-2$ and $-4 \le t \le 4$, and since a path is accepting if and only if the weight is 0, necessarily $t = 0$. The only new possible path with $t = 0$ and ending in q_2 is for $w = uava_kz$, where $u \in a^*$, $v \in (B \cup \{a\})^*$, $a_k \in B$ and $z \in A^*$, and

$$\begin{aligned}(q_0, w, 0) &\models^* (q_0, ava_kz, 5m|u|) \models (q_1, va_kz, 5m|u|)\\ &\models^* (q_1, a_kz, 5m|u|) \models (q_2, z, 5m|u| - 5m|g(a_k)| - 5k)\\ &\models^* (q_2, \varepsilon, 5m(|u| - |g(a_kz)|) - 5k),\end{aligned}$$

and to get the weight equal with 0, necessarily $5k \equiv 0 \pmod{5m}$, which is impossible by the choise of m.

There are two new ways to get to q_3 with $t = 0$. First, if the only new edge is $\langle q_1, a, q_1, 0\rangle$, then we first have used the edge reading a letter in B with weight $15m+1$ from q_0 to q_1, but our word is then in $a^*B^+a^+A^*$ and not in $a^+B^+cA^*$. The second case is that we use edges $\langle q_0, a, q_1, 4\rangle$, $\langle q_1, x, q_2, -2\rangle$ for $x \in B$ and $\langle q_2, a, q_3, -2\rangle$ together with edges defined earlier. But then the word is in $a^+(B \cup \{a\})^*BA^*aA^*$ and not in $a^+B^*cA^*$, and no new words are accepted.

Finally we show that $\mathcal{A}^\gamma$ is unimodal. Note that for all $a \in A$,

$$\min\{|h(a)|, |g(a)|\} > 0.$$

In q_0 the weight of the loop edge when reading a is strictly positive, and the edges to q_1 have weight 0 or positive. From q_1 the loops have weight 0, and since $i - j - 3 < 0$, $1 \leq i, j \leq 3$, we see that the leaving edges from q_1 and all edges from q_2 have strictly negative weight. The edges from q_0 to q_2 have either positive, 0 or negative weight, but this causes no harm, because before that we have only positively and after that we have only negatively weighted edges. In q_3 all edges have weight 0. This shows that $\mathcal{A}^\gamma$ is unimodal, and we have proved the theorem. □

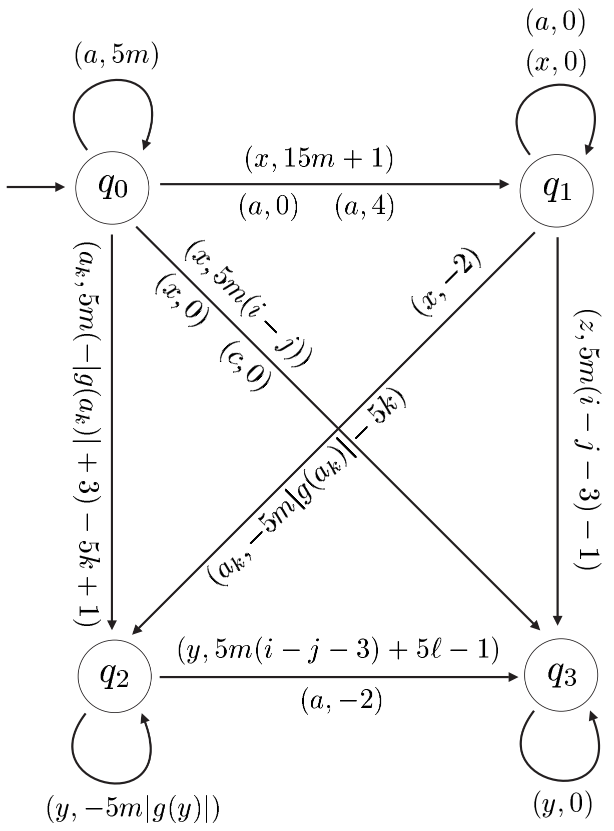

Figure 4: A picture of $\mathcal{A}^\gamma$ constructed in the proof of Theorem 3.2. Here we denote $y \in A$, $x \in B$ and $z \in B \cup \{c\}$.

Note that in the FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ of the proof of Theorem 3.2 the transition function σ is not bijective. It is easy to see that $\mathcal{A}^\gamma$ can be transformed to one having a bijective transition function. Since we have edges $\langle q_0, a, q_1, 0\rangle$

and $\langle q_0, a, q_1, 4\rangle$, we have to make one copy of the state q_1. Similarly we have to make copies of the state q_3, since there might be many edges with different weights reading same letter from q_0 q_1 and q_2 to the state q_3. In other words, if we want to have a bijective transition function, we need much more states.

Next we state few corollaries of Theorem 3.2.

Let $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, A, \sigma, q_0)$, be the FA($\mathbb{Z}$) constructed in the proof of Theorem 3.2, see Figure 4. Let α be a new weight function for the underlying automaton $\mathcal{A}$ such that all edges in $\mathcal{A}$ have weight zero. It is now obvious that

$$L(\mathcal{A}^\alpha) = A^*,$$

and we can state the next corollary considering the equivalence problem for two integer weighted finite automaton.

Corollary 3.3. *It is undecidable for finite automata $\mathcal{A}$ with edge set T, transition function σ and two weight functions $\alpha, \gamma \colon T \to \mathbb{Z}$ whether $L(\mathcal{A}^\alpha) = L(\mathcal{A}^\gamma)$.*

We may extend the above by stating

Corollary 3.4. *It is undecidable for two given* FA($\mathbb{Z}$)*'s $\mathcal{A}^\gamma$ and $\mathcal{B}^\alpha$ whether $L(\mathcal{A}^\gamma) = L(\mathcal{B}^\alpha)$. In fact, $\mathcal{B}$ can be restricted to one state automata.*

Since A^* is a regular language we have also

Corollary 3.5. *It is undecidable for* FA($\mathbb{Z}$) *$\mathcal{A}^\gamma$ and regular languages R whether $L(\mathcal{A}^\gamma) = R$.*

Recall that a group G is *locally finite*, if its finitely generated subgroups are finite. It is clear that if G is locally finite, then $L(\mathcal{A}^\gamma)$ is effectively regular, and thus in this case the universe problem is always decidable. On the other hand, if G is an abelian group that is not locally finite, then it has a finitely generated infinite subgroup, and as is well known, G contains a subgroup isomorphic to $\mathbb{Z}$. Therefore

Corollary 3.6. *Let G be an abelian group. Then the universe problem is decidable for finite automata with weights from G if and only if G is locally finite.*

In the next lemma we state a simple closure property of the languages accepted by the integer weighted finite automata. This collection of languages is referred to as the *family of* FA($\mathbb{Z}$) *languages.*

Lemma 3.7. *The family of* FA($\mathbb{Z}$) *languages is closed under union.*

Proof. Let L_1 and L_2 be two FA($\mathbb{Z}$) languages accepted by the FA($\mathbb{Z}$)'s $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q_1, A, \sigma_1, q_1)$, and $\mathcal{B}^\alpha$, where $\mathcal{B} = (Q_2, A, \sigma_2, q_2)$, respectively. Assume that $Q_1 \cap Q_2 = \emptyset$. We define an FA($\mathbb{Z}$) $\mathcal{C}^\beta$, where

$$\mathcal{C} = (Q_1 \cup Q_2 \cup \{q_0\}, A, \sigma, q_0),$$

and $q_0 \notin Q_1 \cup Q_2$ is the new initial state. The edges in $\mathcal{C}^\beta$ are as in $\mathcal{A}^\gamma$ and $\mathcal{B}^\alpha$, except that we define new edges from q_0 in a way that there is an edge $\langle q_0, a, p, z\rangle$ for $p \in Q_1 \cup Q_2$ in $\mathcal{C}^\beta$ if the edge $\langle q_1, a, p, z\rangle$, where $p \in Q_1$, is in $\mathcal{A}^\gamma$ or the edge $\langle q_2, a, p, z\rangle$, where $p \in Q_2$, is in $\mathcal{B}$. Clearly $L(\mathcal{C}^\beta) = L(\mathcal{A}^\gamma) \cup L(\mathcal{B}^\alpha) = L_1 \cup L_2$. This proves the claim. □

The next corollary will be needed in Chapter 4.

Corollary 3.8. *The universe problem is undecidable for* FA($\mathbb{Z}$) *over a binary alphabet.*

Proof. We shall construct an FA($\mathbb{Z}$) $\mathcal{B}^\alpha$ with a weight function α over a binary alphabet $\{a, b\}$ from the FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ of the proof of Theorem 3.2 such that $L(\mathcal{B}^\alpha) = \{a, b\}^*$ if and only if $L(\mathcal{A}^\gamma) = A^*$.

First we encode the alphabet $A = \{a_1, \ldots, a_n\}$ used in Theorem 3.2 to a binary alphabet. One such encoding is provided by $\psi(a_i) = a^i b$ for $a_i \in A$.

Each edge in $\mathcal{A}^\gamma$ is divided in a way that, for each $q \in \{q_0, q_1, q_2\}$, we have new states $q^1, q^2, \ldots, q^n$ and in $\mathcal{B}^\alpha$ we there are edges,

$$\left\langle q, a, q^1, 0\right\rangle, \quad \left\langle q^k, a, q^{k+1}, 0\right\rangle,$$

for $1 \le k \le n-1$, and, for all $1 \le i \le n$ and $\langle q, a_i, p, z\rangle$ in $\mathcal{A}^\gamma$,

$$\left\langle q^i, b, p, z\right\rangle.$$

For the state q_3, we shall define loops for both letters with weight zero, i.e. $\langle q_3, y, q_3, 0\rangle$ for $y \in \{a, b\}$.

Assume that there is a path π for a word $w \in \psi(A^*)$ in $\mathcal{B}^\alpha$. Then π ends in some of the states $\{q_0, q_1, q_2, q_3\}$ and we can transform π to a path in $\mathcal{A}^\gamma$ having the same same weight and reading the word $\psi^{-1}(w)$. Similarly, if we have a path in $\mathcal{A}^\gamma$ reading a word $v \in A^*$, we can transform it to a path in $\mathcal{B}^\alpha$ having the same weight and reading the word $\psi(v)$. So we get that a word $w \in \psi(A^*)$ is in $L(\mathcal{B}^\alpha)$ if and only if $w \in L(\mathcal{A}^\gamma)$.

Finally, we add a part that accepts the words in the regular language $\{a, b\}^* \setminus \psi(A^+)$. This can be done, since, by Lemma 3.7, the family of FA($\mathbb{Z}$) languages is closed under union. □

Note that the number of states in the FA($\mathbb{Z}$) $\mathcal{B}^\alpha$ is less than or equal to $4n + 7$, where $n = |A|$, since the part which accepts $\{a, b\}^* \setminus \psi(A^*)$ can be done by $n + 3$ additional states.

3.3 A comparison with one–counter automata

As we mentioned, the integer weighted finite automata are closely related to the *one–counter automata* that we considered in this section.

A *pushdown automaton* $\mathcal{P}$, *PDA* for short, is a 7-tuple

$$\mathcal{P} = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F),$$

where Q is the set of *states*, Σ and Γ are the finite *input* and *stack* alphabets, respectively, q_0 is the *initial state*, $z_0 \in \Gamma$ is the *initial stack symbol* and δ is a *transition function*,

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}.$$

A PDA consists of a *tape*, where the input is written, and of a *pushdown stack*, which initially has z_0 as the topmost symbol. The input is read from the tape in a way that if ε is read, then we stay in the same position of the tape, and if $a \in \Sigma$ is read, then we move one position to the right. The transitions are of the form

$$(p, a, z, q, v) \in Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*,$$

which can be explained as follows. When in state p, reading a, and z being the topmost symbol of the stack, then $\mathcal{P}$ moves to the state q, replaces z by v, and stays in the same position or moves one position to the right, depending on whether $a = \varepsilon$ or not. If $z = z_0$, then $v = uz_0$ for some $u \in (\Gamma \setminus \{z_0\})^*$, i.e. the initial stack symbol z_0 cannot be erased. We also assume that z_0 is used only as a mark for empty stack, in other words, the context of the stack is always of the form vz_0, where $v \in (\Gamma \setminus \{z_0\})^*$. The stack is said to be empty, if $v = \varepsilon$.

A *configuration* of $\mathcal{P}$ is a triple $(q, u, v) \in Q \times \Sigma^* \times \Gamma^*$, where q is the current state, u the suffix of the input word, which is not yet read, and v is the current content of the stack.

The acceptance of the input words can be defined in three ways: by final states, by empty stack, or both. These lead to the same family of languages accepted by the pushdown automata, see Harrison [9]. We shall use the acceptance by the final states only.

We say that the word $w \in \Sigma^*$ is *accepted* by a PDA $\mathcal{P}$ if, as usual, there is a sequence of transitions in $\mathcal{P}$ that lead from the initial configuration (q_0, w, z_0) to a final configuration (q_f, ε, z), where $q_f \in F$ and $z \in \Gamma^*$. As usual, a sequence of compatible transitions is called a *computation*. The language accepted by a PDA $\mathcal{P}$ is denoted by $L(\mathcal{P})$.

It is known that the family of languages accepted by pushdown automata and the family of *context–free languages*, denoted by $\mathcal{L}_{CF}$, coincide. For a proof of this, see [1, 9, 22].

A *one–counter automaton* is a pushdown automaton, where the stack alphabet consists of only one symbol other than the initial stack symbol, say $\Gamma = \{1, z_0\}$.

The stack of a one–counter automata can be thought of as a counter, where the value of the counter is the number of 1's in the stack. Actually, we may extend the value of the counter to be an integer instead of a natural number, since the sign of an integer can be remembered in the states. We defined accepting on final states because in this way we may use the counter so that not all configurations with an empty stack contents are accepting.

We shall next compare the integer weighted finite automata and one–counter automata. Here we think that the weights of the paths in an FA($\mathbb{Z}$) are counted in the stack.

In a one–counter automaton the transitions depend on the topmost element of the stack, and thus the transitions depend on the counter, but in an FA($\mathbb{Z}$) this is not the case. Also, in a one–counter automaton the ε–transitions are allowed. In an FA($\mathbb{Z}$), on the other hand, we may add any negative number to the counter on one transition, but in a one–counter automaton we can delete only one symbol at a time from the stack. However, using the ε–transitions, we may delete as many we want. We conclude that the significant differences are that in a one–counter automaton the ε–transitions are allowed and the transitions depend on the topmost element of the stack. The latter is equivalent to the fact that the transitions depend on the emptiness of the stack.

Theorem 3.9. *Let $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, A, \sigma, q_0)$, be an* FA($\mathbb{Z}$). *There exists a one–counter automaton $\mathcal{P}$ such that $L(\mathcal{P}) = L(\mathcal{A}^\gamma)$.*

Proof. First transform $\mathcal{A}^\gamma$ into the form, where each edge has weight -1, 0 or 1 by using ε–edges. This can be done by cutting each edge $\langle q, a, p, z\rangle$ into $|z|$ pieces by introducing $|z| - 1$ new states. Denote this new state set by Q', the set of all edges by σ' and call a sequence of such a cut transition a *modular sequence*. Each of these modular sequences can be thought to be ordered.

Define $\bar{Q}' = \{\bar{q} \mid q \in Q'\}$, where $\bar{q}$ is a new state. Let

$$\mathcal{P} = (Q' \cup \bar{Q}' \cup \{q_f\}, A, \{1, z_0\}, \delta, q_0, z_0, q_f),$$

where q_f is a new state, be a one–counter automaton with a transition function δ, that is defined in the following. For $\langle q, a, p, z\rangle \in \sigma'$, where

$z \in \{-1, 0, 1\}$, $p, q \in Q'$ and $a \in A \cup \{\varepsilon\}$, and $u \in \{1, z_0\}$, define

$$\delta(q, a, u) = \begin{cases} (p, u) & \text{if } z = 0, \\ (p, 1u) & \text{if } z = 1, \\ (p, \varepsilon) & \text{if } z = -1 \text{ and } u = 1, \\ (\bar{p}, 1u) & \text{if } z = -1 \text{ and } u = z_0, \end{cases}$$

$$\delta(\bar{q}, a, u) = \begin{cases} (\bar{p}, u) & \text{if } z = 0, \\ (\bar{p}, 1u) & \text{if } z = -1, \\ (\bar{p}, \varepsilon) & \text{if } z = 1 \text{ and } u = 1, \\ (p, 1u) & \text{if } z = 1 \text{ and } u = z_0. \end{cases}$$

For all $q \in Q \cup Q'$, define

$$\delta(q, \varepsilon, z_0) = (q_f, \varepsilon). \tag{3.4}$$

This clearly means that the sign of the weight is remembered in a way that if the one–counter automaton is in the state $q \in Q'$, then the weight is greater than or equal to zero and if it is in the state $\bar{q} \in \bar{Q}'$, then the weight is less than or equal to zero. The transition in (3.4) implies that we can get to the final state q_f only from the states in $Q \cup Q'$. Note that there is no transitions from the state q_f.

If a word is accepted by $\mathcal{A}^\gamma$ in the state q, then it is accepted in $\mathcal{P}$ in the state q_f, since the weight of the path is simulated on the stack, and from q or $\bar{q}$ there is a transition to q_f with empty stack. On the other hand, if a word is accepted by $\mathcal{P}$ in the state q_f, then by the definition of the transitions function δ we get that the word is accepted by automaton with modular sequences in some state state q such that the state before q_f in $\mathcal{P}$ was q or $\bar{q}$. Therefore it is accepted also by $\mathcal{A}^\gamma$. $\square$

It follows that the family of FA($\mathbb{Z}$) languages is included in the one–counter languages, which we denote by $\mathcal{L}_{OC}$ for short. Actually, this inclusion is proper, since the language

$$\{a^n b^n a^m b^m \mid n, m \geq 0\}$$

is clearly a one–counter language, but not an FA($\mathbb{Z}$) language by the forthcoming proof of Theorem 3.13. We shall denote the family of FA($\mathbb{Z}$) languages by $\mathcal{L}_{\mathrm{FA}(\mathbb{Z})}$ and write that

$$\mathcal{L}_{\mathrm{FA}(\mathbb{Z})} \subset \mathcal{L}_{OC} \subset \mathcal{L}_{CF}.$$

A one–counter automaton is called *1-turn* if the stack is used in a way that first we do not change the content of the stack, then we write to it, then, again we do not change the content of the stack, then we delete from it, and finally the content is not changed. In other words, the value on the counter is changed like in the unimodal FA($\mathbb{Z}$) defined earlier, except that we now allow noncounting transitions at the beginning.

It is obvious that for a unimodal FA($\mathbb{Z}$) the procedure of changing it to a one–counter automaton defined in the proof of Theorem 3.9 yields a 1-turn one–counter automaton. It follows that the family of unimodal FA($\mathbb{Z}$) languages is properly included in the family of 1-turn one–counter languages.

It is known that for a 1-turn one–counter automata the universe problem is undecidable. This is a result of Baker and Book [2] and Greibach [6]. By the above inclusion property, the undecidability of the universe problem for a unimodal FA($\mathbb{Z}$) is a somewhat stronger result.

Ibarra proved in [11] that the universe problem is undecidable for a 1-turn one–counter automaton without ε–transitions. Actually, Ibarra proved this undecidability for two restricted forms of ε–free 1-turn one–counter automata; first, having nondeterminism only in deciding when the counter is started to be used, and second, having nondeterminism only in deciding when the counter goes to the decreasing mode. The automata in the Ibarra's proofs use quite many states. Note that these automata accept with final states.

Next we compare an unimodal FA($\mathbb{Z}$) and an ε–free 1-turn one–counter automaton. It seems to be obvious that the language

$$\left\{a^n b^m c^k \mid 3n + 5m = 7k,\ n, m, k \in \mathbb{N}\right\}$$

cannot be accepted by an ε–free 1-turn one–counter automaton, since we need to delete more than one element from the stack when reading a single letter. But clearly this language can be accepted by a unimodal FA($\mathbb{Z}$), see Figure 5.

On the other hand the language

$$a^* \left\{a^n b^n \mid n \geq 0\right\}$$

can accepted by an ε–free 1-turn one–counter automaton, since in these we allow transitions, which do not change the stack at the beginning of the calculation. But in the definition of unimodality we did not allow the zero weighted edges before the positively weighted edges.

We conclude that the families of languages accepted by 1-turn one–counter automata and unimodal integer weighted finite automata are incomparable. Note that their intersection contains nonregular languages, since

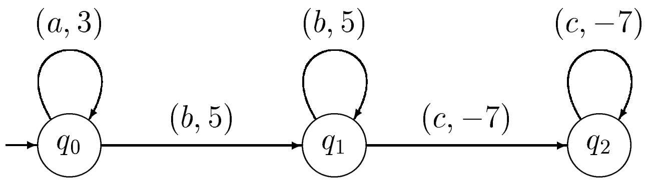

Figure 5: A unimodal FA($\mathbb{Z}$) accepting the language $\{a^n b^m c^k \mid 3n + 5m = 7k,\ n, m, k \in \mathbb{N}\}$.

the language

$$\{a^n b^n \mid n \geq 0\}$$

is in both of these families.

3.4 Closure properties

In this section we consider the closure properties of the family of FA($\mathbb{Z}$) languages under various operations.

We have proved already in Lemma 3.7 that the family of FA($\mathbb{Z}$) languages is closed under union. Actually, this also holds for the family of unimodal FA($\mathbb{Z}$) languages, since the proof of Lemma 3.7 preserves the unimodality. We state this as a corollary:

Corollary 3.10. *The family of unimodal* FA($\mathbb{Z}$) *languages is closed under unions.*

Next we consider the intersection. For this, let

$$L_1 = a^* \{b^n c^n \mid n \geq 0\}$$

and

$$L_2 = \{a^n b^n \mid n \geq 0\} c^*.$$

These languages can be accepted by integer weighted finite automata, but

$$L = L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$$

cannot be, since we know that L is not even a context-free language.

Theorem 3.11. *The family of* FA($\mathbb{Z}$) *languages is not closed under intersections.*

The language L_1 defined above cannot be accepted by a unimodal FA($\mathbb{Z}$), since we did not allow zero weights at the beginning of the computation. But if we redefine

$$L_1 = \{a^n b^m c^n \mid n, m \geq 0\},$$

then we may state

Theorem 3.12. *The family of unimodal* FA($\mathbb{Z}$) *languages is not closed under intersections.*

Next we consider the concatenation operation on languages.

Theorem 3.13. *The family of* FA($\mathbb{Z}$) *languages is not closed under concatenations.*

Proof. Let $L = \{a^n b^n \mid n \geq 0\}$. We shall show that the language

$$L^2 = L \cdot L = \{a^n b^n a^m b^m \mid n, m \geq 0\}$$

cannot be accepted by an FA($\mathbb{Z}$).

Assume on the contrary that there is an FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ with states Q and weight function γ such that it accepts L^2. Since L^2 is an infinite language and it contains words that have length greater than $|Q|$, necessarily there is a cycle in $\mathcal{A}$. We have now two cases to consider:

1) Suppose $\mathcal{A}$ has only one cycle. Assume that the weight of the cycle is d. To accept all words in L^2 with only one cycle, necessarily $d = 0$, since L^2 is infinite. Let $uv \in L^2$, $u, v \in L$ and $|uv| > |Q|$. Then we have a factorization $uv = xyz$, where y is read during the cycle. Since $uv = xyz$ gives a path of weight 0, we get that $xy^k z \in L(\mathcal{A}^\gamma)$ for all $k \in \mathbb{N}$. But L^2 does not contain any words $xy^k z$ for $k > 2$ and $y \neq \varepsilon$.

2) Suppose that $\mathcal{A}^\gamma$ has more than one cycle. If there exists a cycle of weight zero in any accepting path, then we get a contradiction as in the case 1, and on the other hand, the weights of all cycles cannot be of the same sign, since L^2 is an infinite language. It follows that there must be $u, v \in L$ such that $uv = xyzrs$, and, in a accepting path $\pi = \pi_x \pi_y \pi_z \pi_r \pi_s$ of uv, y is read during the cycle π_y and r is read during the cycle π_r, and $\gamma(\pi_y)$ and $\gamma(\pi_r)$ are of different sign. But now the word

$$xy^{k|\gamma(\pi_r)|+1} z r^{k|\gamma(\pi_y)|+1} s \in L(\mathcal{A}^\gamma)$$

for each $k > 0$ is accepted by path

$$\pi' = \pi_x \pi_y^{k|\gamma(\pi_r)|+1} \pi_z \pi_r^{k|\gamma(\pi_y)|+1} \pi_s,$$

since

$$\begin{aligned}
\gamma(\pi') &= \gamma(\pi_x) + \gamma(\pi_y)(k|\gamma(\pi_r)| + 1) + \gamma(\pi_z) + \gamma(\pi_r)(k|\gamma(\pi_y)| + 1) + \gamma(\pi_s) \\
&= \gamma(\pi_x) + \gamma(\pi_y) + \gamma(\pi_z) + \gamma(\pi_r) + \gamma(\pi_s) \\
&\quad + k(\gamma(\pi_y)|\gamma(\pi_r)| + \gamma(\pi_r)|\gamma(\pi_y)|) = 0.
\end{aligned}$$

Since these new words are not in L^2 we get a contradiction that proves the claim. $\square$

Since the above language $L = \{a^n b^n \mid n \geq 0\}$ can be accepted by a unimodal FA($\mathbb{Z}$), we have also

Corollary 3.14. *The family of unimodal* FA($\mathbb{Z}$) *languages is not closed under concatenations.*

Note that although the family of FA($\mathbb{Z}$) languages is not closed under intersections and concatenation, it is closed under these operations with regular languages (that contain the empty word). This is stated in the next theorem.

Theorem 3.15. *Let R be a regular language with $\varepsilon \in R$ and L be an* FA($\mathbb{Z}$) *language. Then $R \cap L$, RL and LR are* FA($\mathbb{Z}$) *languages.*

Proof. Assume that R is accepted by the FA $\mathcal{B} = (Q, A, \delta, q_0, F)$ and L with the FA($\mathbb{Z}$) $\mathcal{A}^\gamma$, where $\mathcal{A} = (P, A, \sigma_1, p_0)$. Assume also that $Q \cap P = \emptyset$. We can transform $\mathcal{B}$ into a $\mathcal{B}^\alpha$ accepting R, where α is defined as in (3.1) on page 18. We may also assume that in $\mathcal{A}^\gamma$ all edges have even weight, since we can multiply each weight by 2, and the accepted language remains the same.

For $R \cap L$, we define $\mathcal{C}^\beta$, where $\mathcal{C} = (Q \times P, A, \sigma, (q_0, p_0))$, and there is an edge

$$\langle (q,p), a, (r,s), (z_1 + z_2) \rangle$$

in $\mathcal{C}^\beta$, for all edges $\langle q, a, r, z_1 \rangle$ in $\mathcal{B}^\alpha$ and $\langle p, a, s, z_2 \rangle$ in $\mathcal{A}^\gamma$. By the definition in (3.1) of α, the weight of a path starting from (q_0, p_0) and ending in a (q, p) is 0 (mod 2) if and only if $q \in F$, which is equivalent to the fact that the word is in R. It follows that $w \in A^*$ is in $R \cap L$ if and only if it is in $L(\mathcal{C}^\beta)$.

For RL we construct an FA($\mathbb{Z}$) $\mathcal{C}^\beta$, where $\mathcal{C} = (Q \cup P, A, \sigma, q_0)$, by connecting the two FA($\mathbb{Z}$)'s by introducing new edges

$$\langle f, a, p, z \rangle ,$$

for $f \in F$ and $a \in A$, if $\langle p_0, a, p, z \rangle$ is an edge in $\mathcal{A}^\gamma$. It is then clear that $L(\mathcal{C}^\beta) = RL$.

For LR the same construction can be used, but this time we connect the two FA($\mathbb{Z}$) in the opposite order. Now the new edges are defined for all $p \in P$, that is,

$$\langle p, a, q, z \rangle ,$$

for $p \in P$ and $a \in A$, if $\langle q_0, a, q, z \rangle$ is an edge in $\mathcal{B}^\beta$. Again, it is clear that $L(\mathcal{C}^\beta) = LR$. □

Next we shall consider the star operation. Let K be a language and define the *star of* K by

$$K^* = \bigcup_{i=0}^{\infty} K^i,$$

where K^i is the concatenation of the language with itself i times, defined recursively by

$$K^0 = \{\varepsilon\}, \quad K^1 = K \quad \text{and} \quad K^i = K^{i-1}K.$$

K^i is also referred to as the ith power of K.

By using again the same language $L = \{a^n b^n \mid n \geq 0\}$ and the proof of Theorem 3.13 we can show

Theorem 3.16. *The family of (unimodal)* FA($\mathbb{Z}$) *languages are not closed under star.*

Proof. Assume that there is an (unimodal) FA($\mathbb{Z}$) $\mathcal{A}^\gamma$ accepting L^* for $L = \{a^n b^n \mid n \geq 0\}$. It follows by Theorem 3.15 that the language

$$L^* \cap a^* b^* a^* b^* = L^2$$

can be accepted by an FA($\mathbb{Z}$), since $a^* b^* a^* b^*$ is a regular language. But this is a contradiction by the proof of Theorem 3.13. □

Next we consider the complement of an FA($\mathbb{Z}$) language. The family of FA($\mathbb{Z}$) languages is not closed under complement, since each FA($\mathbb{Z}$) language contains the empty word, and therefore it is not in the complement. But let us consider the complement modulo ε. For any language $L \subseteq \Sigma^*$, the *complement modulo* ε of L is

$$\bar{L}_\varepsilon = (\Sigma^* \setminus L) \cup \{\varepsilon\}.$$

For example, let $\Sigma = \{a, b\}$ and $L = \{a^n b^n \mid n \geq 0\}$. Now there is a partition

$$\begin{aligned}\bar{L}_\varepsilon = a^* \cup b\Sigma^* \cup a\Sigma^* ba\Sigma^* \\ \cup \{a^n b^m \mid 0 < m < n\} \cup \{a^m b^n \mid m > n > 0\} \, ,\end{aligned} \tag{3.5}$$

and this language can be accepted by the unimodal FA($\mathbb{Z}$) in Figure 6. In other words to prove that the family of (unimodal) FA($\mathbb{Z}$) languages is not closed under complement modulo ε, we have to use some other language than L. It is also clear that this new language cannot be regular, since the family of regular languages is closed under complement.

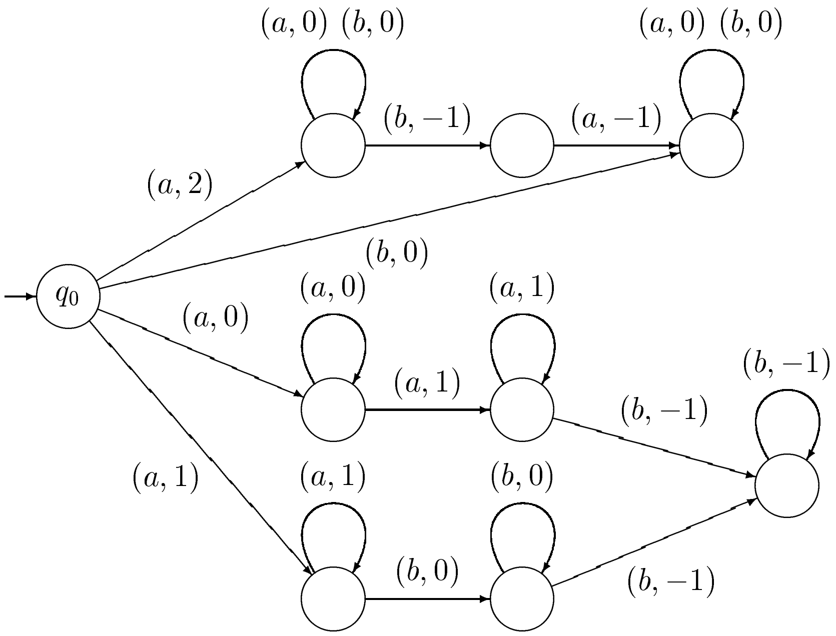

Figure 6: The unimodal FA($\mathbb{Z}$) accepting the complement of the language $\{a^n b^n \mid n \geq 0\}$ modulo ε.

Let $\Sigma = \{a, b, c\}$ and $S = \{a^n b^n c^n \mid n \geq 0\}$. Now S is not an FA($\mathbb{Z}$)

language, since it is not a context–free language. But

$$\begin{aligned}\bar{S}_\varepsilon =& a^* \cup \{b,c\}\,\Sigma^* \cup \Sigma^* ba\Sigma^* \cup \Sigma^* c\,\{a,b\}\,\Sigma^* \\ & \cup \left\{a^n b^m c^k \mid n \neq k,\ k,m,n \in \mathbb{N}\right\} \\ & \cup \left\{a^n b^m c^k \mid n \neq m,\ k,m,n \in \mathbb{N}\right\}.\end{aligned} \tag{3.6}$$

is a unimodal FA($\mathbb{Z}$) language, since the languages on the first row are regular and the languages on the second row are easily seen to be unimodal FA($\mathbb{Z}$) languages. Since

$$\overline{(\bar{S}_\varepsilon)}_\varepsilon = S,$$

we may write

Theorem 3.17. *The family of (unimodal)* FA($\mathbb{Z}$) *languages is not closed under complement modulo ε.*

It is obvious that the family of FA($\mathbb{Z}$) languages is closed under taking the image of a nonerasing morphism $h : A^* \to B^*$, since each transition reading a letter $a \in A$ can be replaced by a new path, which reads the image $h(a)$ and has the same weight as the original transition. To do this replacing so that it preserves unimodality, we first have to multiply each weight in the original FA($\mathbb{Z}$) by the maximal length of an image of a letter under h, and then do the above construction. Note that the morphism must be nonerasing, since otherwise we would get ε–transitions.

For inverse morphisms, we prove the next theorem.

Lemma 3.18. *Let $h : B^+ \to A^+$ be a morphism and $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, A, \sigma, q_0)$, be an* FA($\mathbb{Z}$)*. Then there exists an* FA($\mathbb{Z}$) *$\mathcal{B}^\alpha$ such that*

$$L(\mathcal{B}^\alpha) = h^{-1}(L(\mathcal{A}^\gamma)) = \left\{h^{-1}(w) \mid w \in L(\mathcal{A}^\gamma)\right\}.$$

Proof. Let $\mathcal{B}^\alpha$, where $\mathcal{B} = (Q, B, \sigma', q_0)$. We define the edges of $\mathcal{B}^\alpha$, for all $q \in Q$ and $b \in B$, by

$$\langle q, b, p, z\rangle \quad \text{if} \quad (q, h(b), 0) \models_{\mathcal{A}^\gamma} (p, \varepsilon, z), \tag{3.7}$$

where $p \in Q$ and $z \in \mathbb{Z}$. Note that if $h(b) = \varepsilon$, then there is a loop

$$\langle q, b, q, 0\rangle\,, \tag{3.8}$$

for all $q \in Q$. Now it is straightforward to show that $(q_0, v, 0) \models_{\mathcal{B}^\alpha} (q, \varepsilon, 0)$ if and only if $(q_0, h(v), 0) \models_{\mathcal{A}^\gamma} (q, \varepsilon, 0)$. This proves the claim. ☐

The above construction does not preserve unimodality in the case, where the morphism is erasing, i.e. the image of some letter is the empty word. This is obvious, since we have a loop with weight zero in each state, defined in (3.8).

Next we claim that if the morphism h in Lemma 3.18 is nonerasing and $\mathcal{A}^\gamma$ is unimodal, then the constructed FA($\mathbb{Z}$) $\mathcal{B}^\alpha$ is also unimodal. We shall prove this by assuming on the contrary that this is not the case. Then we have defined two edges $\langle q_1, a, q_2, z_1\rangle$ and $\langle q_2, b, q_3, z_2\rangle$ in (3.7), where $z_1 \leq 0$ and $z_2 > 0$ (or $z_1 < 0$ and $z_2 \geq 0$). But, since z_1 and z_2 are sums of weights of paths of $\mathcal{A}^\gamma$ reading $h(a)$ and $h(b)$, respectively, it follows that necessarily on the path from q_1 to q_3 reading $h(ab)$ on $\mathcal{A}^\gamma$ we also have two consecutive edges with, the first having weight less than or equal to zero and, the second having weight larger than zero. In other words $\mathcal{A}^\gamma$ is not unimodal, which is a contradiction proving the claim.

We have proved

Theorem 3.19. *The family of* FA($\mathbb{Z}$) *languages is closed under taking the images of nonerasing morphisms and under taking the image of inverse morphisms.*

The family unimodal of FA($\mathbb{Z}$) *languages is closed under taking the images of nonerasing morphisms and inverse morphisms.*

Next we shall consider the *shuffle* operation, denoted by $\sqcup\!\sqcup$. For $u, v \in A^*$ and $a, b \in A$ the shuffle is defined recursively by

$$(au \sqcup\!\sqcup bv) = a(u \sqcup\!\sqcup bv) \cup b(au \sqcup\!\sqcup v),$$

and

$$(\varepsilon \sqcup\!\sqcup u) = (u \sqcup\!\sqcup \varepsilon) = \{u\}\,.$$

For two languages $K, L \subseteq A^*$, define the shuffle by

$$K \sqcup\!\sqcup L = \bigcup_{u\in K,\ v\in L} (u \sqcup\!\sqcup v).$$

Theorem 3.20. *The family of (unimodal)* FA($\mathbb{Z}$) *languages is not closed under shuffle.*

Proof. Let $L_1 = \{a^n b^n \mid n \geq 0\}$ and $L_2 = \{c^m d^m \mid m \geq 0\}$. We shall prove that $L_1 \sqcup\!\sqcup L_2$ is not an FA($\mathbb{Z}$) language.

Assume on the contrary that $L_1 \sqcup\!\sqcup L_2$ is an FA($\mathbb{Z}$) language. Then by Theorem 3.15 the language

$$(L_1 \sqcup\!\sqcup L_2) \cap (a^+b^+c^+d^+ \cup \{\varepsilon\}) = L_1 \cdot L_2$$

is also an FA($\mathbb{Z}$) language, since $a^+b^+c^+d^+$ is a regular language. Since $L_1 \cdot L_2$ has a morphic image L_1^2, also L_1^2 is an FA($\mathbb{Z}$) language. But this is contradicts the proof of Theorem 3.13. This proves the claim. □

3.5 Deterministic integer weighted finite automata

An integer weighted finite automata $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, A, \sigma, q_0)$ is said to be *deterministic*, if the underlying finite automata $\mathcal{A}$ is deterministic. This means that for all $q \in Q$ and $a \in A$ there is at most one edge form the state q reading the letter a. This necessarily implies that the transitions function $\sigma \colon T \to \delta$ of $\mathcal{A}^\gamma$ is bijective, and $\delta \subseteq Q \times A \times Q$ defines a deterministic finite automaton. We shall denote a deterministic integer weighted finite automaton by DFA($\mathbb{Z}$).

Note that each regular language containing the empty word can be accepted by a DFA($\mathbb{Z}$), since we know that each regular language modulo ε is accepted by a deterministic finite automaton with only one initial state, and by using the weight function γ in (3.1) on page 18, we get a DFA($\mathbb{Z}$) accepting the given regular language.

It is obvious that we can accept by DFA($\mathbb{Z}$)'s more than regular languages. This follows, since the FA($\mathbb{Z}$) accepting $L = \{a^nb^n \mid n \geq 0\}$ in Figure 3 on page 19 is deterministic.

Next theorem is obvious, since in the proofs of the previous section for the FA($\mathbb{Z}$) languages, the languages used are DFA($\mathbb{Z}$) languages.

Theorem 3.21. *The family of (unimodal)* DFA($\mathbb{Z}$) *languages is not closed under intersections, concatenations, star or shuffle.*

For nonclosures under the union and the complement modulo ε, we have to give new proofs.

Theorem 3.22. *The family of (unimodal)* DFA($\mathbb{Z}$) *languages is not closed under unions or complement modulo* ε.

Proof. The counter example in the case of union is the language

$$a^* \cup \{a^nb^n \mid n \geq 0\}.$$

Assume on the contrary that this language can be accepted be a DFA($\mathbb{Z}$). We see that to get all words in language accepted we must have path for all powers of a with weight zero. On the other hand, this path is unique for each power of a. To do this with finitely many states the path reading the a's must be of the form, where first we have states in a row and then at the and we have a cycle, see Figure 7. The lengths of the row and cycle are fixed. In other words, there must be a state q such that $(q, a^k, 0) \models^* (q, \varepsilon, 0)$ for some minimal $k \geq 1$. Let q be the state where the cycle begins. There is also a unique path from the initial state to q not using any cycles, reading a^{n_0} and having weight 0, for some $n_0 \in \mathbb{N}$. Now for all $m = n_0 + tk$, where $t \geq 0$, we must have a unique path from q reading b^m with weight 0. So we have also an accepting path for $a^k b^\ell$, where $k = n_0 + tk$ and $\ell = n_0 + (t+1)k$ for some $t \geq 0$. Since this word is not in the language, this is a contradiction, that proves the claim.

In the case of the complement, we consider the complement of the language $L = \{a^n b^n \mid n \geq 0\}$ modulo ε. The FA($\mathbb{Z}$) accepting

$$\begin{aligned}\bar{L}_\varepsilon = {} & a^* \cup b\Sigma^* \cup a\Sigma^* ba\Sigma^* \\ & \cup \{a^n b^m \mid 0 < m < n\} \cup \{a^m b^n \mid m > n > 0\}\end{aligned}$$

was given in Figure 6 on page 36. The proof goes now similarly to the case of union. Since this is straightforward, we shall omit the further details. $\square$

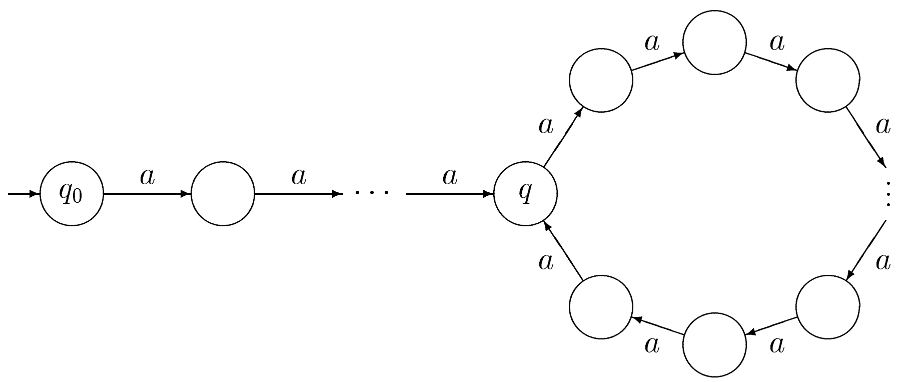

Figure 7: A deterministic path reading a^*.

Although the family of DFA($\mathbb{Z}$) languages is not closed under union, the union of two DFA($\mathbb{Z}$) languages is an FA($\mathbb{Z}$) language, since the latter languages are closed under union.

Next we consider the closure of the family of DFA($\mathbb{Z}$) languages under taking the image of a nonerasing morphism and under taking the inverse image of a morphism.

Theorem 3.23. *The family of (unimodal)* DFA($\mathbb{Z}$) *languages is not closed under taking the image of a morphism, but it is closed under taking the inverse image of a morphism.*

Proof. For the nonclosure under taking the image of a morphism, we give a counter example. Let

$$L = c^* \cup \{a^n b^n \mid n \in \mathbb{N}\} .$$

Clearly this language can be accepted by a DFA($\mathbb{Z}$). Let then $h : \{a, b, c\}^* \to \{a, b\}^*$, $h(a) = h(c) = a$ and $h(b) = b$. Then

$$h(L) = a^* \cup \{a^n b^n \mid n \in \mathbb{N}\} ,$$

and this languages can not be accepted by a DFA($\mathbb{Z}$). This follows by the proof of Theorem 3.22

The closure under the inverse image follows similarly than in the case of FA($\mathbb{Z}$) languages, since the proof of Lemma 3.18 holds also in the deterministic case. Indeed, in the deterministic case for all $h(a)$ there exists at most one path from every state. $\square$

Note that the image of a DFA($\mathbb{Z}$) language under a nonerasing morphism is necessarily an FA($\mathbb{Z}$) language, since the latter languages are closed under the image of a nonerasing morphism.

We end this chapter by considering the universe problem for the deterministic integer weighted finite automata.

Lemma 3.24. *Let $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, A, \sigma, q_0)$, be a* DFA($\mathbb{Z}$) *and $P \subseteq Q$ be the set of the states to which there is a path from q_0. Then $L(\mathcal{A}^\gamma) = A^*$ if and only if, for all $p \in P$ and $a \in A$, there is an edge from p reading a and all these edges have weight 0 in γ.*

Proof. Assume that $L(\mathcal{A}^\gamma) = A^*$ and assume on the contrary that for some $p \in P$ and $a \in A$, there is no edge $\langle p, a, q, z \rangle$. Since $\mathcal{A}$ is deterministic, there is only one path in $\mathcal{A}$ for every word in A^+. Let w be a word of a path which ends in p according to $\mathcal{A}$. Then there is no path for wa in $\mathcal{A}$, which is a contradiction. The latter condition follows similarly, since if there is an edge $\langle p, a, q, z\rangle$, where $z \neq 0$, then wa is not in $L(\mathcal{A}^\gamma)$.

The other direction is trivial. □

As a corollary we get

Theorem 3.25. *The universe problem is decidable for deterministic integer weighted finite automata.*

3.6 Summary

In this section we present a table where we summarize the results of the integer weighted finite automata proved in this chapter. In the following table we denote that closure by + and nonclosure by −.

	FA($\mathbb{Z}$)	unimodal FA($\mathbb{Z}$)	(unimodal) DFA($\mathbb{Z}$)
Union	+ 3.7	+ 3.10	− 3.22
Intersection	− 3.11	− 3.12	− 3.21
Concatenation	− 3.13	− 3.14	− 3.21
Star	− 3.16	− 3.16	− 3.21
Complement mod ε	− 3.17	− 3.17	− 3.22
Shuffle	− 3.20	− 3.20	− 3.21
Image(h)	+ (*) 3.19	+ (*) 3.19	− 3.23
Image(h^{-1})	+ 3.19	+ (*) 3.19	+ 3.23
Universe problem	undecidable	undecidable	decidable

Here h is a morphism, and it assumed to be nonerasing in the cases denoted by (*). Note that we actually proved that the universe problem is undecidable for 4-state unimodal FA($\mathbb{Z}$), and for FA($\mathbb{Z}$) over binary alphabet.

4 Finite substitutions

In this chapter we present two undecidability results concerning the inclusion and the equivalence problem for finite ε–free substitutions. In these undecidability reductions we shall use the undecidability of the universe problem for the integer weighted finite automaton over a binary alphabet, proved in the previous chapter. The proofs presented are from Halava and Harju [8].

4.1 A special inclusion problem

In this section we prove the undecidability of a special inclusion problem. Let A and B be alphabets, and $h\colon A^* \to B^*$ be a morphism. Then h is said to be *periodic nonerasing* morphism, if there exists a word $w \in B^+$ such that $h(u) \in w^+$ for all $u \in A^+$.

Theorem 4.1. *It is undecidable for periodic nonerasing morphisms $h\colon \{b, 0, 1, c\}^* \to \{a, b\}^*$ and ε–free finite substitutions $\chi : \{b, 0, 1, c\}^* \to 2^{\{a,b\}^*}$ whether*

$$h(u) \in \chi(u) \quad \textit{for all} \quad u \in b\,\{0, 1\}^*\,c.$$

Proof. Let $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, \{0, 1\}, \sigma, q_1)$, be an FA($\mathbb{Z}$). We shall define h and χ in a way that $h(u) \in \chi(u)$ for all $u \in b\{0, 1\}^*c$ if and only if $L(\mathcal{A}^\gamma) = \{0, 1\}^*$.

As usual we shall identify a singleton set $\{u\}$ with its sole member u. Let $s = |Q| + 1$, say $Q = \{q_1, \ldots, q_{s-1}\}$, where q_1 is the initial state of $\mathcal{A}$. Further, let

$$r = |\min\{z \mid \langle q, x, p, z\rangle \in \sigma\}|,$$

i.e. r is the absolute value of the minimal weight of an edge in $\mathcal{A}^\gamma$. Define

$$w = a^s b, \qquad N = \bigcup_{k=1}^{s-1} a^k b, \quad \text{and} \quad T_x = \bigcup_{\langle q_k, x, q_j, z\rangle \in \sigma} a^{s-k} b w^{z+r} a^j. \tag{4.1}$$

Here T_x encodes the rules of $\mathcal{A}^\gamma$. Define then

$$\chi(b) = a, \quad \chi(c) = Nw, \quad \chi(x) = T_x,$$

for both $x \in \{0, 1\}$, and

$$h(b) = w, \quad h(c) = w, \quad h(x) = w^{r+1}.$$

Let $L = b\{0, 1\}^*c$. Clearly, for all $u \in \{0, 1\}^*$, $h(buc) = w \cdot w^{|u|(r+1)} \cdot w$, and we shall prove that $w^{|u|(r+1)+2} \in \chi(buc)$ for $u \in \{0, 1\}^*$ if and only if $u \in L(\mathcal{A}^\gamma)$.

Let $u = x_1 \dots x_m$, where $x_i \in \{0,1\}$ for $1 \le i \le m$ and let $v \in \chi(buc)$. By the definition of χ we obtain that

$$v = a \cdot a^{s-k_1} b w^{z_1+r} a^{j_1} \cdot a^{s-k_2} b w^{z_2+r} a^{j_2} \cdots a^{s-k_m} b w^{z_m+r} a^{j_m} \cdot a^{\ell} b w,$$

for a sequence

$$\langle q_{k_i}, x_i, q_{j_i}, z_i \rangle, \quad 1 \le i \le m. \tag{4.2}$$

Assume that $v \in w^+$. Then necessarily $k_1 = 1$, $k_{i+1} = j_i$, for $1 \le i \le m-1$, and $\ell = s - j_m$. This means that the sequence (4.2) is a path of u in $\mathcal{A}^\gamma$ starting from the initial state q_1 and ending in the state q_{j_m}. It follows that

$$v = w^{1+\sum_{i=1}^{m}(z_i+r+1)+1} = w^{\sum_{i=1}^{m} z_i + m(r+1)+2}.$$

Now $v = w^{|u|(r+1)+2}$ if and only if $\sum_{i=1}^{m} z_i = 0$, which means that the sequence $\langle q_{k_i}, x_i, q_{j_i}, z_i \rangle$, $1 \le i \le m$, is an accepting path of u in $\mathcal{A}^\gamma$.

The undecidability claim of the theorem follows from the undecidability of the universe problem of the FA($\mathbb{Z}$) $\mathcal{A}^\gamma$, see Corollary 3.8. □

4.2 The equivalence problem

We shall now prove that the equivalence problem of finite substitutions is undecidable for the regular language $b\{0,1\}^* c$.

The proof we present modifies the proof of Lisovik [17]. In the original proof Lisovik used the undecidability of the reliability of the *defense systems*, cf. [16]. We shall consider these systems and the chain of the undecidability reductions proving the undecidability in Section 4.3.

We shall use the undecidability of the universe problem for finite integer weighted automata over an binary alphabet. Our chain of undecidability reductions is shorter than the original chain of Lisovik, and the model of an automaton we use is simple, and the undecidability proofs for it are elementary.

Note that the undecidability of the equivalence problem is not trivial, since it is known that there are rather general decidable special cases. For example, for finite prefix substitutions the equivalence problem on regular language is known to be decidable, see Karhumäki [14]. This follows from the fact that the monoid of all prefix codes is free and from the pumping property of regular languages. Also note that the question asking, whether the equivalence problem of finite substitutions on regular language is decidable, was a well-known and studied problem until Lisovik solved it in 1997, for example see Culik and Karhumäki [3].

There also exist some other undecidability results on ε–free finite substitutions, see especially Turakainen [23]. We shall state these results of Turakainen at the end of this section. For decidability and undecidability results of the equivalence problem for other mappings, see Karhumäki [13].

Theorem 4.2. *The equivalence problem for ε-free finite substitutions on the regular language $b\{0,1\}^*c$ is undecidable.*

Proof. Let $\mathcal{A}^\gamma$, where $\mathcal{A} = (Q, \{0,1\}, \sigma, q_1)$ be an FA($\mathbb{Z}$). We shall define two finite substitutions $\varphi, \xi : \{b, 0, 1, c\}^* \to \{a, b\}^*$ such that φ and ξ are equivalent on $b\{0,1\}^*c$ if and only if $L(\mathcal{A}^\gamma) = \{0,1\}^*$.

As in the proof of Theorem 4.1, let $s = |Q| + 1$, and $Q = \{q_1, \ldots, q_{s-1}\}$, where q_1 is the initial state of $\mathcal{A}$, and let

$$r = |\min\{z \mid \langle q, x, p, z\rangle \in \sigma\}|.$$

We define w, N and T_x, for $x \in \{0,1\}$, as in (4.1). Further, for both $x \in \{0,1\}$, define

$$C_x = \bigcup_{\langle q_k, x, q_j, z\rangle \in \sigma} a^{s-k} b w^{z+r+1}.$$

Finally, we define the substitutions ξ and φ as follows:

$$\xi(b) = a \cup wN,$$

$$\xi(c) = (\varepsilon \cup N)w,$$

$$\xi(x) = w^{r+1} \cup (\varepsilon \cup N)(T_x \cup C_x N),$$

for $x \in \{0,1\}$, and

$$\varphi(b) = \xi(b) \cup w, \quad \varphi(c) = \xi(c), \quad \varphi(x) = \xi(x).$$

Let

$$L = b\{0,1\}^*c.$$

Clearly $\xi(v) \subseteq \varphi(v)$ for all $v \in L$, since $\xi(x) \subseteq \varphi(x)$ for all letters $x \in \{b, 0, 1, c\}$. Therefore to prove the claim we need to show that $\varphi(x) \subseteq \xi(x)$ for all $x \in L$ if and only if $L(\mathcal{A}^\gamma) = \{0,1\}^*$.

Assume first that $L(\mathcal{A}^\gamma) \neq \{0,1\}^*$. Let $u \in \{0,1\}^*$ be such that $u \notin L(\mathcal{A}^\gamma)$, and assume that h and χ are defined as in the proof of Theorem 4.1. Clearly $h(buc) \in \varphi(buc)$ and $\chi(buc) \subseteq \xi(buc)$, and by the proof of Theorem 4.1 and the form of N, $h(buc) \in \xi(buc)$ if and only if $u \in L(\mathcal{A}^\gamma)$. By our assumption $h(buc) = w^{|u|(r+1)+2} \notin \xi(buc)$. It follows from $L(\mathcal{A}^\gamma) \neq \{0,1\}^*$ that $\varphi(x) \not\subseteq \xi(x)$ for some $x \in b\{0,1\}^*c$.

Assume next that $L(\mathcal{A}^\gamma) = \{0,1\}^*$ and let $x = x_0 \ldots x_{n+1} \in L$, and $u = u_0 \ldots u_{n+1}$, where $x_i \in \{b, 0, 1, c\}$ and $u_i \in \varphi(x_i)$ for all $0 \le i \le n+1$. Note that necessarily $x_0 = b$ and $x_{n+1} = c$. We have to show that there exist words $v_i \in \xi(x_i)$ for all $0 \le i \le n+1$ such that $v_0 \ldots v_{n+1} = u$.

First, we note that the only difference in the images of ξ and φ is in the image of b, and $\xi(b) \setminus \varphi(b) = w$. If $u_0 \ne w$, then we have a trivial solution $u_i = v_i$ for all $0 \le i \le n+1$. Therefore we assume that $u_0 = w$.

We shall use parentheses to illustrate the factorizations by φ and ξ to u_i's and v_i's, respectively. We divide the considerations into three cases:

(i) If $n = 0$, then $x = bc$ and we have two subcases:

(1) If $u_1 = w \in \varphi(c)$, then $u_0 u_1 = (w)(w) = (a)(a^{s-1}bw) \in \xi(x)$.

(2) If $u_1 \in Nw \subseteq \varphi(c)$, then for some $1 \le k \le s-1$,

$$u_0 u_1 = (w)(a^{s-k}bw) = (wa^{s-k}b)(w) \in \xi(x).$$

(ii) If $n \ge 1$ and $u_1 \ne w^{r+1}$, then we show that there is a factorization such that $u_i = v_i$ for $2 \le i \le n+1$ and $u_0 u_1 = v_0 v_1$. For this, there are four subcases:

(1) If $u_1 \in T_{x_1}$, then $u_1 = a^{s-k}bw^{z+r}a^j$ for some $\langle q_k, x_1, q_j, z \rangle$, and

$$u_0 u_1 = (w)(a^{s-k}bw^{z+r}a^j) = (a)(a^{s-1}ba^{s-k}bw^{z+r}a^j) = v_0 v_1 \in a \cdot NT_{x_1},$$

where $a \in \xi(b) = \xi(x_0)$ and $NT_{x_1} \subseteq \xi(x_1)$ as required.

(2) If $u_1 \in NT_{x_1}$, then $u_1 = a^{s-\ell}ba^{s-k}bw^{z+r}a^j$ for some $\langle q_k, x_1, q_j, z \rangle$ and $1 \le \ell \le s-1$,

$$u_0 u_1 = (w)(a^{s-\ell}ba^{s-k}bw^{z+r}a^j) = (wa^{s-\ell}b)(a^{s-k}bw^{z+r}a^j) = v_0 v_1 \in wN \cdot T_{x_1},$$

where $wN \subseteq \xi(b) = \xi(x_0)$ and $T_{x_1} \subseteq \xi(x_1)$ as required.

(3) If $u_1 \in C_{x_1}N$, then $u_1 = a^{s-k}bw^{z+r+1}a^{s-\ell}b$ for some $\langle q_k, x_1, q_j, z \rangle$ and $1 \le \ell \le s-1$, and

$$\begin{aligned} u_0 u_1 &= (w)(a^{s-k}bw^{z+r+1}a^{s-\ell}b) \\ &= (a)(a^{s-1}ba^{s-k}bw^{z+r+1}a^{s-\ell}b) = v_0 v_1 \in a \cdot NC_{x_1}N, \end{aligned}$$

where $a \in \xi(b) = \xi(x_0)$ and $NC_{x_1}N \subseteq \xi(x_1)$ as required.

(4) If $u_1 \in NC_{x_1}N$, then $u_1 = a^{s-\ell}ba^{s-k}bw^{z+r+1}a^{s-t}b$ for some $\langle q_k, x_1, q_j, z \rangle$ and $1 \le \ell, t \le s-1$, and

$$\begin{aligned} u_0 u_1 &= (w)(a^{s-\ell}ba^{s-k}bw^{z+r+1}a^{s-t}b) \\ &= (wa^{s-\ell}b)(a^{s-k}bw^{z+r+1}a^{s-t}b) = v_0 v_1 \in wN \cdot C_{x_1}N, \end{aligned}$$

where $wN \subseteq \xi(b) = \xi(x_0)$ and $C_{x_1}N \subseteq \xi(x_1)$ as required.

(iii) If $n \geq 1$ and $u_1 = w^{r+1}$ then we need the fact that $L(\mathcal{A}^\gamma) = \{0,1\}^*$. Let $t = \min\{i \mid i \geq 1, u_i \neq w^{r+1}\}$. Note that $t \leq n+1$, since $x_{n+1} = c$ and $w^{r+1} \notin \varphi(c)$. Now $u_0 u_1 \dots u_{t-1} = w(w^{r+1}) \cdots (w^{r+1}) = w^{(t-1)(r+1)+1}$.

By our assumption, for the sequence $x' = x_1 \dots x_{t-1} \in \{0,1\}^*$, $\mathcal{A}^\gamma$ has edges

$$\left\langle q_{j_{i-1}}, x_i, q_{j_i}, z_i \right\rangle, \quad i = \{1, \dots, t-1\},$$

where $1 \leq j_i \leq s-1$ and $j_0 = 1$, and

$$\sum_{i=1}^{t-1} z_i = 0, \tag{4.3}$$

since $x' \in L(\mathcal{A}^\gamma) = \{0,1\}^*$. Therefore there exist words

$$v'_i = a^{s-j_{i-1}} b w^{z_i + r} a^{j_i} \in T_{x_i},$$

for $1 \leq i \leq t-1$. By setting $v_0 = a$, we get

$$\begin{aligned} v_0 v'_1 \dots v'_{t-1} &= a a^{s-1} b w^{z_1+r} a^{j_1} a^{s-j_1} b w^{z_2+r} a^{j_2} \dots a^{s-j_{t-2}} b w^{z_{t-1}+r} a^{j_{t-1}} \\ &= w w^{z_1+r} w w^{z_2+r} w \dots w w^{z_{t-1}+r} a^{j_{t-1}}. \end{aligned}$$

By (4.3)

$$v_0 v'_1 \dots v'_{t-1} = w^{(t-1)r+(t-1)} a^{j_{t-1}} = w^{(t-1)(r+1)} a^{j_{t-1}}, \tag{4.4}$$

and therefore $u_0 u_1 \dots u_{t-1} = v'_0 v'_1 \dots v'_{t-1} a^{s-j_{t-1}} b$. We may now choose $v_i = v'_i$ for $1 \leq i \leq t-2$.

We have two cases depending on t. First if $t = n+1$, then we have two subcases. Recall that $x_{n+1} = c$.

(1) If $u_{n+1} = w$, then we can choose $v_n = v_{t-1} = v'_{t-1}$ and $v_{n+1} = a^{s-j_{t-1}} b w \in Nw \in \xi(c)$. We get that $u = v$ by (4.4).

(2) If $u_{n+1} = a^{s-k} b w \in Nw$ for some $1 \leq k \leq s-1$, then we set

$$v_{t-1} = v_n = a^{s-j_{t-2}} b w^{z_{t-1}+r+1} a^{s-k} b \in C_{x_n} N \quad \text{and} \quad v_{n+1} = w.$$

Again this yields that $u = v$ by (4.4).

The second case is that $t \leq n$. In this case $x_t \in \{0,1\}$. Then we choose $v_i = u_i$ for $t+1 \leq i \leq n+1$ and there are four subcases for v_{t-1} and v_t depending on u_t:

(1) If $u_t = a^{s-k} b w^{z+r} a^j \in T_{x_t}$ for some $\langle q_k, x, q_j, z \rangle$, then we set

$$v_{t-1} = v'_{t-1} \text{ and } v_t = a^{s-j_{t-1}} b a^{s-k} b w^{z+r} a^j \in NT_{x_t},$$

to obtain $u = v$.

(2) If $u_t = a^{s-\ell}ba^{s-k}bw^{z+r}a^j \in NT_{x_t}$ for some $\langle q_k, x, q_j, z\rangle$ and $1 \leq \ell \leq s-1$, then we set

$$v_{t-1} = a^{s-j_{t-2}}bw^{z_{t-1}+r}a^{j_{t-1}}a^{s-j_{t-1}}ba^{s-\ell}b = a^{s-j_{t-2}}bw^{z_{t-1}+r+1}a^{s-\ell}b \in C_{x_t}N$$

and

$$v_t = a^{s-k}bw^{z+r}a^j \in T_{x_t},$$

to obtain $u = v$.

(3) If $u_t = a^{s-k}bw^{z+r+1}a^{s-\ell}b \in C_{x_t}N$ for some $\langle q_k, x, q_j, z\rangle$ and $1 \leq \ell \leq s-1$, then we set

$$v_{t-1} = v'_{t-1} \text{ and } v_t = a^{s-j_{t-1}}ba^{s-k}bw^{z+r+1}a^{s-\ell}b \in NC_{x_t}N,$$

to obtain $u = v$.

(4) If $u_t = a^{s-\ell}ba^{s-k}bw^{z+r+1}a^{s-h}b \in NC_{x_t}N$ for some $\langle q_k, x, q_j, z\rangle$ and $1 \leq \ell, h \leq s-1$, then we set

$$v_{t-1} = a^{s-j_{t-2}}bw^{z_{t-1}+r}a^{j_{t-1}}a^{s-j_{t-1}}ba^{s-\ell} = a^{s-j_{t-2}}bw^{z_{t-1}+r+1}a^{s-\ell}b \in C_{x_t}N$$

and

$$v_t = a^{s-k}bw^{z+r+1}a^{s-h}b \in C_{x_t}N,$$

to obtain $u = v$.

We have proved that if $L(\mathcal{A}^\gamma) = \{0,1\}^*$ then $\varphi(L) \subseteq \xi(L)$. The undecidability now follows by the undecidability of the universe problem for FA($\mathbb{Z}$) over a binary alphabet, Corollary 3.8.

$\square$

In the previous proof we actually proved that it is undecidable, whether the inclusion

$$\varphi(x) \subseteq \xi(x),$$

holds for all word $x \in \Sigma^*$. This also follows from the statement of Theorem 4.2.

As we mentioned, there are also some other undecidability results concerning finite substitutions. We shall state two propositions proved by Turakainen in [23].

Note that, for a a finite substitution $\varphi\colon \Sigma^* \to 2^{\Delta^*}$, the inverse of φ is defined by

$$\varphi^{-1}(w) = \{v \in \Delta^* \mid w \in \varphi(v)\}$$

for all $w \in \Sigma^*$.

Proposition 4.3. *The following are undecidable for the class of all ε–free finite substitutions $\varphi\colon \{a,b\}^* \to 2^{\{a,b,c\}^*}$:*

(i) *Does the equality $\varphi_1(w) \cap \{ac,bc\}^* = \varphi_2(w) \cap \{ac,bc\}^*$ hold for all w in $\{a,b\}^*$?*

(ii) *Does the equality $\varphi_1(w) \cap \{ac,bc\}^* = \varphi_2(w)$ hold for all w in $\{a,b\}^*$?*

(iii) *Does the equality $\varphi_1^{-1}(w) = \varphi_2^{-1}(w)$ hold for all w in $\{ac,bc\}^*$?*

Let S be a set. The cardinality of S is denoted by $\operatorname{card} S$

Proposition 4.4. *It is undecidable, given an ε–free finite substitution φ and two morphisms h and g, whether $\varphi(w) = gh^{-1}(w)$ for all w in $\{ac,bc\}^*$. It is also undecidable whether* $\operatorname{card}\varphi(w) = \operatorname{card} gh^{-1}(w)$ *for all w in $\{ac,bc\}^*$.*

4.3 Defense systems

In this section we shall consider the so called defense systems, DS for short, defined by Lisovik in [16]. We summarize the chain of the undecidability reduction for the original proof of L. Lisovik for Theorem 4.2.

A defense system is supposed to defense infinitely many *nodes* numbered by the integers $\mathbb{Z}$.

A DS is a triple $\mathcal{D} = (K, H, \Gamma)$, where K is a set of *lines*,

$$K = \{i \mid 1 \le i \le s, \quad i \in \mathbb{N}\},$$

for a given $s \in \mathbb{N}$, H is a set of *instructions* and Γ is a set of *attacking symbols.*

Each node can be defended by any of the lines from K. In other words, each node can be defended by s different lines. The initial situation in our case is that only the node 0 is defended by the line 1, and the other nodes do not have defense at all.

The *attacker* is supposed to "send" symbols from the set Γ to the defending system, which on the other hand uses some instruction defined for these symbols. This means that the attacks can be thought as words from Γ^*, and the behaviour of the DS is defined by the instructions under this attack word.

Each rule of the set H is of the form (k, a, j, z, p), where $1 \le k, j \le s$, $a \in \Gamma$, $z \in \{-1, 0, 1\}$ and p is a real number, $0 \le p \le 1$. Such a rule means

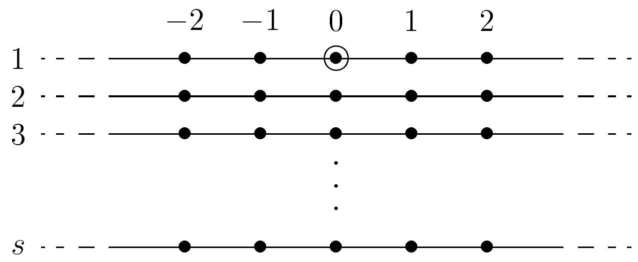

Figure 8: A picture illustrating a defense system in the initial configuration defending the node 0 by the line 1.

that when an attacking symbol a is sent, defense of the node i by the line k can move with probability p to defend the node $i+z$ by the line j. We shall denote this probability also by $p^z_{a,k,j}$. Naturally we set, for all $a \in \Gamma$,

$$\sum_{j=1}^{s} \sum_{z=-1}^{1} p^z_{a,k,j} = 1$$

for each k.

Note that the underlying system in the defense system is nondeterministic and therefore the model of the defense system we have defined is also called *nondeterministic DS*, NDS for short.

An NDS can also be viewed as a countable Markov system.

To simplify the notations we denote each configuration of an NDS by an integer. If the node i is defended by the line j, we denote this configuration by the integer $i \cdot s + (j-1)$. Recall that the initial configuration is that the node 0 is defended by the line 1, which is represented as the integer 0.

Let $w \in \Gamma^*$. We shall denote the probability that the NDS is in a configuration $k \in \mathbb{Z}$ in response to a finite sequence of attacking signals w by $p_w(k)$.

Let $\mathcal{D} = (K, H, \Gamma)$ be a defense system. $\mathcal{D}$ is called *unreliable* if, for some $w \in \Gamma^*$, after an attacking sequence w the probability that the node 0 is defended by some line is 0, i.e. $p_w(j) = 0$ for all $0 \leq j \leq s-1$. The word w here is called *critical*. If there is no critical words $w \in \Gamma^*$ for $\mathcal{D}$, then $\mathcal{D}$ is called *reliable*.

Note that the problem of reliability of the NDS $\mathcal{D} = (K, H, \Gamma)$ is equivalent to the universe problem for the integer weighted finite automaton

$$\mathcal{A} = (K, \Gamma, \sigma, 0),$$

where the edges in σ are constructed from the instructions in H by selecting only those instructions with positive probability and then omitting the probabilities.

Actually, we are interested in the case when Γ is fixed to be a binary alphabet $\{0, 1\}$. The next result is from Lisovik [16].

Proposition 4.5. *The unreliability of nondeterministic defense system is undecidable, i.e. it is undecidable for NDS $\mathcal{D} = (K, H, \{0, 1\})$ whether there exists $w \in \{0, 1\}^*$ such that $p_w(j) = 0$ for all $0 \leq j \leq s - 1$.*

This propositions also yields, that the universe problem is undecidable for integer weighted finite automata. But our proof for this presented in Chapter 3 is much simpler, and the number of lines in the proof of the previous proposition is very large.

The proof of the previous proposition uses the undecidability of the inclusion problem of the Z–*transducers*, a special type of a *finite transducer*. A finite transducer is a finite automaton with output, i.e. each transition has labels from $A^* \times B^*$, where B is the output alphabet. In a Z–transducer the input alphabet is binary, say $\{0, 1\}$, and the output alphabet is unary, say $\{c\}$. Further, the labels in transitions are from $\{0, 1\} \times \{c, cc\}$ and the transducer has only one initial and one final state. A Z–transducer is deterministic, if the underlying automaton is deterministic, and complete, if the underlying automaton is complete.

For a Z–tranducer C, let

$$O(C) = \left\{(u, v) \mid u \in \{0, 1\}^+,\ v \in \{c\}^+,\ \text{and } v \text{ is a output of } u\right\}.$$

Lemma 4.6. *It is undecidable for deterministic Z–transducer C and nondeterministic and complete Z–transducer D whether $O(C) \subseteq O(D)$*

The previous lemma is a corollary from [15]. It is a corollary of the next proposition.

Proposition 4.7. *It is undecidable for two finite transducers C and D with unary output alphabet whether*

1) $O(C) \subseteq O(D)$,

2) $O(C) = O(D)$.

For the proof of the previous proposition, see also Ibarra [10] and for further result, see Turakainen [23].

Lisovik's proof of the previous propositions in [15] uses the undecidability of the PCP, which, as we already mentioned, follows from the halting problem for Turing machines.

We compare the two different chains of the undecidability reductions for the the equivalence problem of two ε–free finite substitutions on regular language in Figure 9 below.

$$
\begin{tikzcd}
 & \text{Halting problem} \arrow[dl, "{[20]}"'] \arrow[dr, "\text{Theorem 2.3}"] & \\
\text{PCP} \arrow[d, "{[15]}"] & & \text{Prefix PCP} \arrow[d, "\text{Theorem 3.2}"] \\
\text{Inclusion for FT with unary output} \arrow[d, "{[15]}"] & & \text{Universe problem for FA}(\mathbb{Z}) \arrow[dd, "\text{Corollary 3.8}"] \\
\text{Inclusion for Z–transducers} \arrow[d, "{[16]}"] & & \\
\text{Reliability of the NDS} \arrow[dr, "{[17]}"] & & \text{Universe problem for binary FA}(\mathbb{Z}) \arrow[dl, "\text{Theorem 4.2}"] \\
 & \text{Equivalence of the } \varepsilon\text{-free finite substitutions on regular language} &
\end{tikzcd}
$$

Figure 9: Two different proof chains for the undecidability of the equivalence problem of ε–free finite substitutions on the regular language $b\{0,1\}^*c$.

References

[1] J. Autebert, J. Berstel, and L. Boasson, *Context-Free languages and Pushdown Automata*, Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), vol. 1, Springer-Verlag, 1997.

[2] B. Baker and R. Book, *Reversal-bounded multipushdown machines*, J. Comput. System Sci. **8** (1974), 315–332.

[3] K. Culik II and J. Karhumäki, *Decision problems solved with the help of the Ehrenfeucht's conjecture*, Bulletin of the EATCS, no. 27, 1985, pp. 30–35.

[4] A. Ehrenfeucht, J. Karhumäki, and G. Rozenberg, *The (generalized) Post Correspondence Problem with lists consisting of two words is decidable*, Theoret. Comput. Sci. **21** (1982), 119–144.

[5] S. Eilenberg, *Automata, languages and machines*, vol. A, Academic Press, 1974.

[6] S. A. Greibach, *An infinite hierarchy of context-free languages*, J. Assoc. Comput. Mach. **16** (1969), 91–106.

[7] V. Halava and T. Harju, *Undecidability in integer weighted finite automata*, Tech. Report 158, Turku Centre for Computer Science, January 1998.

[8] ———, *Undecidability of the equivalence of finite substitutions on regular language*, Tech. Report 160, Turku Centre for Computer Science, February 1998.

[9] M. A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley, 1978.

[10] O. H. Ibarra, *The unsolvability of the equivalence problem for ϵ-free NGSM's with unary input (output) alphabet and applications*, SIAM J. of Comput. **7** (1978), no. 4, 524–532.

[11] ———, *Restricted one-counter machines with undecidable universe problems*, Math. Systems Theory **13** (1979), 181–186.

[12] M. Jantzen, *Confluent String Rewriting*, Springer-Verlag, 1988.

[13] J. Karhumäki, *The equivalence of mappings on languages*, Trends, techniques, and problems in theoretical computer science (Smolenice, 1986), Lecture Notes in Comput. Sci., vol. 281, Springer, 1987, pp. 26–38.

[14] J. Karhumäki, *Equations over finite sets of words and equivalence problems in automata theory*, Theoret. Comput. Sci. **108** (1993), no. 1, 103–118.

[15] L. P. Lisovik, *Minimal undecidable identity problem for finite-automaton mappings*, Cybernetics **19** (1983), no. 2, 160–165.

[16] ______, *An undecidable problem for countable Markov chains*, Cybernetics **27** (1991), no. 2, 163–169.

[17] ______, *Nondeterministic systems and finite substitutions on regular language*, Bulletin of the EATCS, no. 63, 1997, pp. 156–160.

[18] Y. Matiyasevich and G. Sénizergues, *Decision problems for semi-Thue systems with a few rules*, Proceedings, 11^{th} Annual IEEE Symposium on Logic in Computer Science (New Brunswick, New Jersey), IEEE Computer Society Press, 27–30 July 1996, pp. 523–531.

[19] V. Mitrana and R. Stiebe, *The accepting power of finite automata over groups*, New Trends in Formal Language (G. Păun and A. Salomaa, eds.), Lecture Notes in Comput. Sci., vol. 1218, Springer-Verlag, 1997, pp. 39–48.

[20] E. Post, *A variant of a recursively unsolvable problem*, Bulletin of Amer. Math. Soc. **52** (1946), 264–268.

[21] G. Rozenberg and A. Salomaa, *Cornerstones of Undecidability*, Prentice Hall, 1994.

[22] A. Salomaa, *Formal languages*, Academic Press, 1973.

[23] P. Turakainen, *The undecidability of some equivalence problems concerning ngsm's and finite substitutions*, Theoret. Comput. Sci. **174** (1997), no. 1-2, 269–274.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**