Quantitative Semantics on Jumping Automata

Ishai Salgado

Quantitative Semantics on Jumping Automata

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Ishai Salgado

Submitted to the Senate
of the Technion — Israel Institute of Technology
Sivan 5784 Haifa June 2024

This research was carried out under the supervision of Dr. Shaull Almagor, in the Faculty of Computer Science.

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract

Jumping automata are finite automata that read their input in a non-sequential manner, by allowing a reading head to ”jump“ between positions of the input, consuming a permutation of the input word. We argue that allowing the head to jump should incur some cost. To this end, we propose three quantitative semantics for jumping automata, whereby the jumps of the head in an accepting run define the cost of the run. The three semantics correspond to different interpretations of jumps; the *absolute distance* semantics counts the distance the head jumps, the *reversal* semantics counts the number of times the head changes direction, and the *Hamming* semantics measures the number of letter-swaps the run makes.

We define and study several problems regarding these semantics. The *membership problem* determines given a jumping automaton whether a word w is bounded by some given number k. We show that the membership problem is NP-complete under the three semantics.

We also study the *boundedness problem*: given a jumping automaton, decide whether its (quantitative) language is bounded by k. We establish the decidability and give lower bounds for this problem under several variants.

Furthermore, several relations between the boundedness problems of the semantics are established: if an automaton is bounded under the Hamming semantics, it is also shown to be bounded under the reversal semantics. Similarly, absolute distance boundedness implies Hamming boundedness. We give examples showing that no other similar implications hold.

Chapter 1

Introduction

Traditional automata read their input sequentially as is the case for most state-based computational models. In contrast, a Jumping Finite Automaton (introduced in [MZ12]) may read its input in a non-sequential manner, jumping from letter to letter, as long as every letter is read exactly once. A jumping automaton is relevant in cases where the order of the input does not matter. One such example is when the input represents available resources, and we only wish to reason about their quantity. From a more language-theoretic perspective, this amounts to looking at the commutative closure of the languages, a.k.a. their Parikh image. Several works have studied the algorithmic properties and expressive power of these automata [FPS15, FS17, Vor18, FHY21, LPS14].

In [MZ12], the model of JFA (jumping finite automaton) and its motivation are presented. Various closure properties of JFA languages are studied, e.g. JFA languages are shown to be closed under complement, intersection, union, permutation and shuffle. [FS17] further studies the JFA model, introducing a variant of regular-like expressions, called alphabetical shuffle expressions that characterize JFA languages. Using these expressions it is proved that JFA are closed under iterated shuffle. Moreover, a number of complexity problems are studied, e.g. the membership problem for JFAs is shown to be NP-hard.

While JFAs are an attractive and simple model, they present a shortcoming when thought of as a model for systems, namely that the abstraction of the order may be too coarse. More precisely, the movement of the head can be thought of as a physical process of accessing the input storage of the JFA. Then, for some cases the movement should be cheaper than in others, e.g. when the head moves sequentially. The ability to jump around is physically more difficult so it should not come "for free".

In our work we present three *quantitative semantics* which attempt to quantify the cost of jumping. For our purposes we treat a JFA as a function from words to costs, capturing how expensive it is to accept a given word with respect to the head jumps. We wish to explore the properties of such semantics, their differences and other traits.

We briefly present the three different semantics: Consider a JFA $\mathcal{A}$ and a word w, and let ρ be an accepting run of $\mathcal{A}$ on w. The run ρ in our case specifies the sequence of states and indices visited in w. We first define the cost of individual runs.

1. In the *Hamming* ($\mathcal{A}_{\text{HAM}}$) semantics, we look at the word w' induced by ρ, i.e, the word obtained when reading w in the order ρ reads it. The cost of ρ is the number of letters where w' differs from w.

2. In the *Absolute Distance* ($\mathcal{A}_{\text{ABS}}$) semantics the cost of ρ is the sum of the lengths of jumps it makes.

3. In the *Reversal* ($\mathcal{A}_{\text{REV}}$) semantics the cost of ρ is the number of times the head "turns" while reading w (i.e. it changes from moving left to right to moving right to left or vice versa).

We then define the cost of the word w according to each semantics (which we denote by $\mathcal{A}_{\text{SEM}}(w)$) where SEM $\in \{\text{HAM}, \text{ABS}, \text{REV}\}$, by taking the run that minimizes the cost.

Thus, we lift JFAs from a Boolean automata model to the rich setting of quantitative models [DKV09]. Unlike other quantitative automata, however, the semantics in this setting arise naturally from the model, without an external domain. Moreover, the definitions are naturally motivated by different types of memory access, as we now demonstrate. First, consider a system whose memory is laid out in an array (i.e., a tape), with a reading head that can move along the tape. Moving the head requires some energy, and therefore the total energy spent reading the input corresponds to the ABS semantics. Next, consider a system whose memory is a spinning disk (or a sliding tape), so that the head stays in place and the movement is of the memory medium. Then, it is cheap to continue spinning in the same direction, and the main cost is in changing the arm direction. Then, the REV semantics best captures the cost. Finally, consider a system that reads its input sequentially, but is allowed to edit its input by replacing one letter with another, such that at the end the obtained word is a permutation of the original word. This is akin to *edit-distance automata* [Moh03] under a restriction of maintaining the amount of resources. Then, the minimal edits required correspond to the HAM semantics.

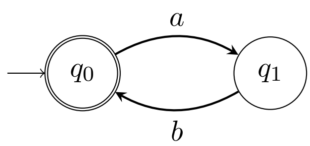

Figure 1.1: The Jumping Finite Automaton $\mathcal{A}$

Example 1.0.1. In order to illustrate the differences between the semantics defined above, consider the JFA $\mathcal{A}$, depicted in Figure 1.1. $\mathcal{A}$ accepts every word where the number of instances of a is equal to the number of instances of b, as every such word has a permutation in $(ab)^*$. But the word $w = a^3b^3$ has different costs depending on the semantincs used:

- In the Hamming semantics, $\mathcal{A}_{\mathrm{HAM}}(w) = 2$ as there is an accepting run where the only letters changed are the letters in indices 2 and 5. It is not hard to see that there is no better run.
- In the Absolute Distance semantics, $\mathcal{A}_{\mathrm{ABS}}(w) = 8$. An optimal order of indices to be read is 1,4,2,5,3,6, which has three jumps of cost two (1 to 4, 2 to 5 and 3 to 6. In Chapter 4, the cost of a jump is defined to be one less than the distance jumped over) and two jumps of cost 1 (4 to 2 and 5 to 3).
- In the Reversal semantics, $\mathcal{A}_{\mathrm{REV}}(w) = 4$ by the same sequence of indices above, as the head performs four "turns", two from right to left (at indices 4 and 5) and two from left to right (at indices 2 and 3).

1.1 Related Work

Jumping Automata were introduced in [MZ12]. We remark that [MZ12] contains some erroneous proofs (e.g., closure under intersection and complement, also pointed in [FS17]). The works in [FPS15, FS17] establish several expressiveness results on jumping automata, as well as some complexity results. In [Vor18] many additional closure properties are established. An extension of jumping automata with a two-way tape was studied in [FHY21], and jumping automata over infinite words were studied by the first author in [AY23].

When viewed as the commutative image of a language, jumping automata are closely related to Parikh Automata [KR03, CFM11, CFM12], which read their input and accept if a certain Parikh image relating to the run belongs to a given semilinear set (indeed, we utilize the Parikh Automata in our proofs). Another related model is that of symmetric transducers - automata equipped with outputs, such that permutations in the input correspond to permutations in the output. These were studied in [Alm20] in a jumping flavour, and in [NA21] in a quantitative k-window flavour.

More broadly, quantitative semantics have received much attention in the past two decades, with many motivations and different flavors of technicalities. For more information, the reader should refer to [Bok21, DKV09] and the references therein.

1.2 Contribution and Organization

Our contribution consists of the introduction of the three jumping semantics, and the study of decision problems pertaining to them (defined in Chapter 3). Our main focus is the boundedness problem: given a JFA $\mathcal{A}$, decide whether the function described by it under each of the semantics is bounded by some constant k. We establish the decidability for all the semantics, and consider the complexity of some fragments.

This work is organized as follows: the preliminaries and definitions are given in Chapter 2 and Chapter 3. Then, each of Chapters 4 to 6 studies one of the semantics, and follows the same structure: we initially establish that the membership problem for the semantics is NP-complete. Then we characterize the set of words whose cost is at most k using a construction of an NFA. These constructions differ according to the semantics, and involve some nice tricks with automata, but are technically not hard to understand. We note that these constructions are preceded by crucial observations regarding the semantics, which allow us to establish their correctness. Next, in Chapter 7 we give a complete picture of the interplay between the different semantics (using some of the results established beforehand). Finally, in Chapter 8 we discuss some exciting open problems.

Chapter 2

Preliminaries

Consider a finite alphabet Σ. For every $n \in \{1, \ldots, n\}$ we denote by Σ^n the set of words of length n over Σ. For $w \in \Sigma^n$ we denote its letters by $w = w_1 \cdots w_n$, and its length by $|w| = n$. Σ^* is the language of all words over Σ of any length.

2.1 Permutations

Given $n \in \mathbb{N}$, the *permutation group* S_n is the set of bijections (*permutations*) from $\{1, ..., n\}$ to itself. The *identity* permutation is denoted by id and is defined by $\text{id}(i) = i$ for every $i \in \{1, ..., n\}$.

S_n forms a group with the function-composition operation. Given a word $w = w_1 \cdots w_n \in \Sigma^n$ and a permutation $\pi \in S_n$, we define $\pi(w) = (w_{\pi(1)}, ..., w_{\pi(n)})$.

We say that a word y is a *permutation* of x, and we write $x \sim y$ if and only if there exists a permutation $\pi \in S_{|x|}$ such that $\pi(x) = y$.

2.2 Nondeterministic Finite Automata

A *nondeterministic finite automaton* (NFA) is a 5-tuple $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a nondeterministic transition function, $Q_0 \subseteq Q$ is a set of initial states, and $F \subseteq Q$ is a set of accepting states.

For a word $w = w_1 w_2 \cdots w_n \in \Sigma^*$, we define the *run* of $\mathcal{A}$ on w to be a sequence $\rho = q_0, q_1, \ldots, q_n$ where $q_0 \in Q_0$ and $q_{i+1} \in \delta(q_i, w_{i+1})$ for every $0 \leq i < |w|$. We say that ρ is *accepting* if $q_n \in F$. A word w is *accepted* by $\mathcal{A}$ if there exists an accepting run of $\mathcal{A}$ on w. The *language* of $\mathcal{A}$, denoted by $\mathfrak{L}(\mathcal{A})$, is the set of words accepted by $\mathcal{A}$.

An NFA $\mathcal{A}$ is *universal* if $\mathfrak{L}(\mathcal{A}) = \Sigma^*$. The *universality problem* for NFAs is the question of whether $\mathcal{A}$ is universal. This problem is well known to be PSPACE-complete [MS72].

2.3 Jumping Automata

Consider an NFA $\mathcal{A}$. We see $\mathcal{A}$ as a *jumping finite automaton* (JFA) by letting it "jump" over letters. Equivalently, it can be seen as an NFA that reads a (nondeterministically chosen) permutation of the input word.

Formally, consider an NFA $\mathcal{A}$. A word $w \in \Sigma^*$ is accepted by the *jumping finite automaton* (JFA) $\mathcal{A}$ if there is a permutation π such that $\pi(w)$ is accepted by $\mathcal{A}$ as an NFA. The *jumping language* of $\mathcal{A}$ is defined, in a similar way to NFAs, to be all the words accepted by $\mathcal{A}$. Equivalently, it is:

$$\mathfrak{J}(\mathcal{A}) = \{w \in \Sigma^* \mid \exists u \in \Sigma^*. w \sim u \wedge u \in \mathfrak{L}(\mathcal{A})\},$$

Since our aim is to reason about the manner with which the head of a JFA jumps, we introduce a notion to track the head along a run. Consider a word w of length n and an NFA $\mathcal{A}$. A *jump sequence* is a vector $\mathbf{a} = (a_0\ a_1\ a_2\ \ldots\ a_n\ a_{n+1})$ where $a_0 = 0$, $a_{n+1} = n + 1$ and $(a_1\ a_2\ \ldots\ a_n) \in S_n$. We denote by J_n the set of all jump sequences of size $n + 2$.

Intuitively, a jump sequence $\mathbf{a} = (a_0\ a_1\ a_2\ \ldots\ a_n\ a_{n+1})$ represents the order in which a JFA visits a given word of length n. First it visits the letter at index a_1, then the letter at index a_2 and so on. To capture this, we define $w_{\mathbf{a}} = w_{a_1} w_{a_2} \cdots w_{a_n}$. Observe that jump sequences enforce that the head starts at position 0 and ends at position $n + 1$, which can be thought as left and right markers, as is common in e.g., two-way automata.

An alternative view of jumping automata is via *Parikh Automata* (PA) [CFM12], [KR03]. The standard definition of PA is an automaton whose acceptance condition includes a semilinear set over the transitions. To simplify things, and to avoid defining unnecessary concepts (e.g., semilinear sets), for our purposes, a PA is a pair $(\mathcal{A}, \mathcal{C})$ where $\mathcal{A}$ is an NFA over alphabet Σ, and $\mathcal{C}$ is a JFA over Σ. Then, the PA $(\mathcal{A}, \mathcal{C})$ accepts a word w if $w \in \mathfrak{L}(\mathcal{A}) \bigcap \mathfrak{J}(\mathcal{C})$. Note that when $\mathfrak{L}(\mathcal{A}) = \Sigma^*$, then the PA coincides with $\mathfrak{J}(\mathcal{C})$. Our usage of PA is to obtain the decidability of certain problems. Specifically, from [KR03] we have that emptiness of PA is decidable.

Chapter 3

The Quantitative Semantics

3.1 Introduction

In this chapter we present and demonstrate the three quantitative semantics for JFAs. We then define the relevant decision problems. For the remainder of the chapter fix a JFA $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$.

3.2 The Semantics

3.2.1 The Absolute Distance Semantics

In the absolute distance semantics, the cost of a run (given as a jump sequence) is the sum of the sizes of the jump made by the head. Since we want to think of a sequential run as a run with 0 jumps, we measure a jump over k cells as distance $k-1$ (either to the left or to the right). This is captured as follows.

For $k \in \mathbb{Z} \setminus \{0\}$ we define $\|k\| = |k| - 1$. Consider a word $w \in \Sigma^*$ with length n, and let $\mathbf{a} = (a_0, a_1, a_2, \ldots, a_n, a_{n+1})$ be a jump sequence, then we lift the notation above and write $\|\mathbf{a}\| = \Sigma_{i=1}^{n+1} \|a_i - a_{i-1}\|$.

Definition 3.2.1. For a word $w \in \Sigma^*$ with length n we define

$$\mathcal{A}_{\text{ABS}}(w) = \min\{\|\mathbf{a}\| \mid \mathbf{a} \textit{ is a jump sequence, and } w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})\}$$

If the set above is empty, we define $\mathcal{A}_{\text{ABS}}(w) = \infty$.

3.2.2 The Reversal Semantics

In the reversal semantics, the cost of a run is the number of times the head changes direction in the corresponding jump sequence. Consider a word $w \in \Sigma^*$ with length n, and let $\mathbf{a} = (a_0\ a_1\ a_2\ \ldots\ a_n\ a_{n+1})$ be a jump sequence, we define

$$\#_{\text{REV}}(\mathbf{a}) = |\{i \in \{1, \ldots, n\} \mid (a_i > a_{i-1} \wedge a_i > a_{i+1}) \vee (a_i < a_{i-1} \wedge a_i < a_{i+1})\}|$$

Definition 3.2.2. For a word $w \in \Sigma^*$ with length n we define

$$\mathcal{A}_{\text{REV}}(w) = \min\{\#_{\text{REV}}(\mathbf{a}) \mid \mathbf{a} \textit{ is a jump sequence}, \textit{ and } w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})\}$$

If the set above is empty, we define $\mathcal{A}_{\text{REV}}(w) = \infty$.

3.2.3 The Hamming Semantics

In the Hamming semantics, the cost of a word is the minimal number of coordinates that need to be changed in order for the obtained word to be accepted by $\mathcal{A}$ (sequentially, as an NFA), so that the changed word is a permutation of w.

Definition 3.2.3. The Hamming distance between words of the same length $x, y \in \Sigma^n$ such that $x \sim y$ is:

$$d_{\text{HAM}}(x, y) = |\{i \mid x_i \neq y_i\}|$$

Definition 3.2.4. For a word $w \in \Sigma^*$ we define:

$$\mathcal{A}_{\text{HAM}}(w) = \min\{d_H(w, w') \mid w' \in L(\mathcal{A}), w' \sim w\}$$

If the set above is empty, we define $\mathcal{A}_{\text{HAM}}(w) = \infty$.

Remark. Note that the definitions of the three semantics are independent of the NFA, and only refer to its language. We can therefore refer to the cost of a word in a language according to each semantics, rather than the cost of a word in a concrete automaton.

3.3 Quantitative Decision Problems

Let $\text{SEM} \in \{\text{HAM}, \text{ABS}, \text{REV}\}$.

We say that $\mathcal{A}_{\text{SEM}}$ is *k-bounded* if for all $w \in \mathfrak{J}(\mathcal{A})$, $\mathcal{A}_{\text{SEM}}(w) \leq k$. We define a similar term for the universal variant: $\mathcal{A}_{\text{SEM}}$ is *universally k-bounded* if for all $w \in \Sigma^*$, $\mathcal{A}_{\text{SEM}}(w) \leq k$. Additionally, we say that $\mathcal{A}_{\text{SEM}}$ is *bounded* (or *universally bounded*) if there exists a $k \in \mathbb{N}$ such that $\mathcal{A}_{\text{SEM}}(w) \leq k$ for all $w \in \mathfrak{J}(\mathcal{A})$ (or all $w \in \Sigma^*$, respectively).

In the remainder of the work we focus on quantitative variants of the standard Boolean decision problems pertaining to the jump semantics. Specifically, we consider the following problems for each semantics.

1. Membership: Given a JFA $\mathcal{A}$, $k \in \mathbb{N}$ and a word w, decide whether $\mathcal{A}_{\text{SEM}}(w) \leq k$.
2. k-Boundedness (fixed k): Given a JFA $\mathcal{A}$, decide whether $\forall w \in \mathfrak{J}(\mathcal{A})\ \mathcal{A}_{\text{SEM}}(w) \leq k$.
3. Param-Boundedness: Given a JFA $\mathcal{A}$ and $k \in \mathbb{N}$, decide whether $\forall w \in \mathfrak{J}(\mathcal{A})\ \mathcal{A}_{\text{SEM}}(w) \leq k$.

We also pay special attention to the universal setting, in which case we refer to the last two problems as Univ-k-Boundedness and Univ-Param-Boundedness. For example, in Univ-Param-Boundedness we are given a JFA $\mathcal{A}$ and $k \in \mathbb{N}$ and the problem is to decide whether $\mathcal{A}_{\text{SEM}}(w) \leq k$ for all words $w \in \Sigma^*$.

The boundedness problems can be thought of as quantitative variants of Boolean universality (i.e., is the language equal to Σ^*). Observe that the problems above are not fully specified, as the encoding of k (binary or unary) when it is part of the input may affect the complexity. We remark on this when it is relevant.

Note that Boolean emptiness problem variants are absent from the list above. Indeed, a natural quantitative variant would be: is there a word w such that $\mathcal{A}_{\text{SEM}}(w) \leq k$. This, however, is identical to Boolean emptiness, since if there exists a word w such that $\mathcal{A}_{\text{SEM}}(w) \leq k$ then there exists a word w' such that $\mathcal{A}_{\text{SEM}}(w') = 0$. Conversely, if there does not exist such w, then $\mathcal{A}$ is empty when seen as an NFA. We therefore do not consider this problem. Another problem to consider is boundedness when k is existentially quantified. We elaborate on this problem in Chapter 8.

We observe that there is a simple reduction from k-Boundedness to Param-Boundedness: given a JFA $\mathcal{A}$, output $(\mathcal{A}, k)$. We will use this reduction throughout our work.

Example 3.3.1. As an example observe again the JFA $\mathcal{A}$ which appears in Figure 1.1. We have shown that $\mathcal{A}$ is not 1-bounded regarding the three measures. Moreover, the set of words $\{a^n b^n \mid n \in \mathbb{N}\}$ shows that $\mathcal{A}$ isn't k-bounded for any k and for any of the three measures.

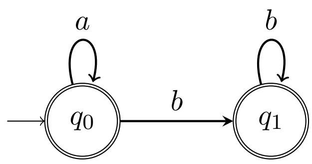

Figure 3.1: The Jumping Finite Automaton $\mathcal{B}$

Example 3.3.2. Consider now the NFA $\mathcal{B}$ from Figure 3.1 which accepts the language a^*b^*. When seen as a JFA, $\mathcal{B}$ accepts all the words in $\{a, b\}^*$, since every such word can be written as a sequence of a's followed by a sequence of b's. Observe that in the REV semantics, for every word w we have $\mathcal{B}_{\text{REV}}(w) \leq 2$, since at the worst case $\mathcal{B}$ reads all the a's in one left to right pass, then all of the b's in one right to left pass, and then jumps to the right end marker, and thus it has two turning points. In particular, $\mathcal{B}_{\text{REV}}$ is bounded. However, in the ABS and HAM semantics, the cost can become unbounded as seen by words of the form $b^n a^n$. Indeed, in the ABS semantics the head must first jump over all the b^n, incurring a high cost, and for the HAM semantics, all the letters must be changed, also incurring a high cost.

Chapter 4

The Absolute Distance Semantics

4.1 The Membership Problem for ABS

The first semantics we investigate is the Absolute Distance semantics, and we start by showing that its membership problem is NP-complete. This is based on bounding the distance with which a word can be accepted.

Lemma 4.1.1. *Let $\mathcal{A}$ be an NFA, and $w \in \mathfrak{J}(\mathcal{A})$ with length n. Then $\mathcal{A}_{ABS}(w) \leq n^2$.*

Proof. Since $w \in \mathfrak{J}(\mathcal{A})$, there exists a jump sequence $\mathbf{a} = (a_0\ a_1\ \ldots\ a_n\ a_{n+1})$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$ and $\mathcal{A}_{\text{ABS}}(w) = \|\mathbf{a}\|$. Observe that $|a_i - a_{i-1}| \leq n$ for all $i \in \{1, \ldots, n+1\}$, since there is no jump from 0 to $n+1$ (since the head starts at $a_0 = 0$ and ends at $a_n = n+1$ and must go through all the indices between). The following concludes the proof:

$$\|\mathbf{a}\| = \sum_{i=1}^{n+1} \|a_i - a_{i-1}\| = \sum_{i=1}^{n+1} (|a_i - a_{i-1}|) - 1 \leq \sum_{i=1}^{n+1} (n-1) = (n+1)(n-1) < n^2$$

We can now prove the complexity bound for the membership problem in the Absolute Distance semantics, as follows:

Theorem 4.1. *The problem of deciding, given a JFA $\mathcal{A}$, $w \in \Sigma^*$ and $k \in \mathbb{N}$, whether $\mathcal{A}_{ABS}(w) \leq k$, is* NP*-complete.*

Proof. First we establish membership in NP. If $k > n^2$ we can set $k = n^2$ as per Lemma 4.1.1. So we can assume $k \leq n^2$. Then, it is sufficient to nondeterministically guess a jump sequence $\mathbf{a} \in J_n$ and to check that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$ and that $\|\mathbf{a}\| \leq k$. Both conditions are easily checked in polynomial time, since k is polynomially bounded.

Hardness in NP follows by reduction from (Boolean) membership in JFA: it is shown in [FS17] that deciding whether $w \in \mathfrak{J}(\mathcal{A})$ is NP-hard. We reduce this problem to ours by outputting, given $\mathcal{A}$ and w, the same $\mathcal{A}$ and w with the bound $k = n^2$. The reduction's correctness follows from the fact that if $w \in \mathfrak{J}(\mathcal{A})$ then by Lemma 4.1.1, $\mathcal{A}_{\text{ABS}}(w) \leq n^2$, and if $w \notin \mathfrak{J}(\mathcal{A})$ then $\mathcal{A}_{\text{ABS}}(w) = \infty > n^2$. ∎

4.2 Decidability of Boundedness Problems for ABS

We now turn our attention to the boundedness problems. Consider a JFA $\mathcal{A}$ and $k \in \mathbb{N}$. Intuitively, our approach is to construct and NFA $\mathcal{B}$ that simulates, while reading a word $w \in \Sigma^*$, every jump sequence of $\mathcal{A}$ on w whose absolute distance is at most k. The crux of the proof is to show that we can indeed bound the size of $\mathcal{B}$ as a function of k. At a glance, the main idea here is to claim that since the absolute distance is bounded by k, then $\mathcal{A}$ cannot make large jumps, nor many small jumps. Then, if we track a sequential head going from left to right, then the jumping head must always be within a bounded distance from it. We now turn to the formal arguments. Fix a JFA $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$.

In order to understand the next lemma, imagine $\mathcal{A}$'s jumping head while taking the j^{th} step in a run on w according to a jump sequence $\mathbf{a} = (a_0\ a_1\ a_2\ \ldots\ a_n\ a_{n+1})$. Thus the jumping head currently points to the letter at index a_j. Concurrently, imagine a "sequential" head, which points to the j^{th} letter in w, which is the letter that would be read if we were reading w in a sequential manner. Note that these two heads start and finish reading the word at the same indices $a_0 = 0$ and $a_{n+1} = n+1$. So, it stands to reason that if at any step while reading w the distance between these two heads is large, the cost of reading w according to $\mathbf{a}$ would also be large, as there would need to be jumps that bridge the gaps between them. The following lemma formalizes this idea:

Lemma 4.2.1. *Let $\mathbf{a} = (a_0\ a_1\ a_2\ \ldots\ a_n\ a_{n+1})$ be a jump sequence. For every $1 \leq j \leq n$ it holds that $\|\mathbf{a}\| \geq |a_j - j|$.*

Proof. Let $1 \leq j \leq n+1$. First, assume that $a_j \geq j$. We'll look at the sum $\sum_{i=1}^{j} \|a_i - a_{i-1}\| \leq \|\mathbf{a}\|$. From the definition of $\|\cdot\|$ we have $\sum_{i=1}^{j} \|a_i - a_{i-1}\| = \sum_{i=1}^{j} |a_i - a_{i-1}| - j$, and we conclude that in this case $\|\mathbf{a}\| \geq |a_j - j|$ by the following:

$$\sum_{i=1}^{j} |a_i - a_{i-1}| - j \underset{\substack{\text{triangle}\\\text{inequality}}}{\geq} \left|\sum_{i=1}^{j} a_i - a_{i-1}\right| - j \underset{\substack{\text{telescopic}\\\text{sum}}}{=} |a_j - a_0| - j = a_j - j = |a_j - j|$$

Now assume that $a_j < j$. The proof in this case is similar but instead of looking at the sum $\sum_{i=1}^{j} \|a_i - a_{i-1}\|$, we look at the sum $\sum_{i=j+1}^{n+1} \|a_i - a_{i-1}\| \leq \|\mathbf{a}\|$: From the definition of $\|\cdot\|$, $\sum_{i=j+1}^{n+1} \|a_i - a_{i-1}\| = \sum_{i=j+1}^{n+1} |a_i - a_{i-1}| - (n+1-(j+1)+1)$. Then,

$$\sum_{i=j+1}^{n+1} |a_i - a_{i-1}| - (n+1-(j+1)+1) \underset{\substack{\text{triangle}\\\text{inequality}}}{\geq} \left|\sum_{i=j+1}^{n+1} a_i - a_{i-1}\right| - (n+1-(j+1)+1)$$

$$\underset{\substack{\text{telescopic}\\\text{sum}}}{=} |a_{n+1} - a_j| - (n+1-j) = |n+1-a_j| - (n+1-j) = j - a_j = |j - a_j|$$

From Lemma 4.2.1 we get that in order for a word w to attain a small cost, it must be accepted with a jump sequence that stays close to the sequential head. More precisely:

Corollary 4.2. *Let $k \in \mathbb{N}$ and consider a word w such that $\mathcal{A}_{ABS}(w) \leq k$, then there exists a jumping sequence $\mathbf{a} \in J_n$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$ and for all $1 \leq j \leq n$ we have $|a_j - j| \leq k$.*

We now turn to the construction of an NFA that recognizes the words whose cost is at most k.

Lemma 4.2.2. *Let $k \in \mathbb{N}$. We can effectively construct an NFA $\mathcal{B}$ such that $\mathfrak{L}(\mathcal{B}) = \{w \in \Sigma^* \mid \mathcal{A}_{ABS}(w) \leq k\}$.*

Proof. Let $k \in \mathbb{N}$. Intuitively, $\mathcal{B}$ works as follows: it remembers in its states a window of size $2k+1$ centered around the current letter (recall that as an NFA, $\mathcal{B}$ reads its input sequentially). The window is constructed by nondeterministically guessing (and then verifying) the next k letters, and remembering the last k letters.

$\mathcal{B}$ then nondeterministically simulates a jumping sequence of $\mathcal{A}$ on the given word, with the property that the jumping head stays within distance k from the sequential head. This is done by marking for each letter in the window whether it has already been read in the jumping sequence, and nondeterministically guessing the next letter to read, while keeping track of the current jumping head location, as well as the total cost incurred so far. After reading a letter, the window is shifted by one to the right. If at any point the window is shifted so that a letter that has not been read by the jumping head shifts out of the $2k+1$ scope, the run rejects. The correctness of the construction follows from Corollary 4.2. We now turn to the formal details. Let $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0.F \rangle$. We define $\mathcal{B} = \langle \Sigma, Q', \delta', Q'_0, F' \rangle$ as follows.

The state space of $\mathcal{B}$ is $Q' = Q \times (\Sigma \times \{?, \checkmark\})^{-k,\dots,k} \times \{-k, \dots, k\} \times \{0, \dots, k\}$. We denote a state of $\mathcal{B}$ as (q, f, j, c) where $q \in Q$ is a state of $\mathcal{A}$, $f : \{-k, \dots, k\} \to \Sigma \times \{?, \checkmark\}$ represents a window of size $2k+1$ around the sequential head (the sequential head is always at index 0), where $\checkmark$ marks letters that have already been read by $\mathcal{A}$ (and ? marks the others), j represents the index of the head of $\mathcal{A}$ relative to the sequential head, and c represents the cost incurred thus far in the run. We refer to the components of f as $f(j) = (f(j)_1, f(j)_2)$ with $f(j)_1 \in \Sigma$ and $f(j)_2 \in \{?, \checkmark\}$.

The initial states of $\mathcal{B}$ are $Q'_0 = \{(q, f, j, j-1) \mid q \in Q_0 \wedge j > 0 \wedge (f(i)_2 = \checkmark \iff i \leq 0)\}$. That is, all states where the state of $\mathcal{A}$ is initial, the location of the jumping head is some $j > 0$ incurring a cost of $j-1$ (i.e, the initial jump $\mathcal{A}$ makes), and the window is guessed so that everything left of the first letter is marked as already-read (to simulate the fact that $\mathcal{A}$ cannot jump to the left of the first letter).

The transitions of $\mathcal{B}$ are defined as follows. Consider a state (q, f, j, c) and letter $\sigma \in \Sigma$, then $(q', f', j', c') \in \delta'((q, f, j, c), \sigma)$ if and only if the following hold:

- $f(1)_1 = \sigma$. That is, we verify that the next letter in the guessed window is indeed correct.
- $f(-k)_2 = \checkmark$. That is, the leftmost cell has been read. Otherwise by Corollary 4.2 the cost of continuing the run must be greater than k.
- $f(j)_2 \neq \checkmark$ and $f'(j-1) = \checkmark$ (if $j > -k$). That is, the current letter has not been previously read, and will be marked as read from now on (note that the index j before the transition corresponds to index $j-1$ after).
- $q' = \delta(q, f(j)_1)$, i.e. the state of $\mathcal{A}$ is updated according to the current letter.
- $c' = c + |j' + 1 - j| - 1$, since j' represents the index in the shifted window, so in the "pre-shifted" tape this is actually index $j'+1$. We demonstrate this in Figure 4.1. Also, $c' < k$ by the definition of Q.
- $f'(i) = f(i+1)$ for $i < k$. That is, the window is shifted and the index $f'(k)$ is nondeterministically guessed (note that the guess could potentially be $\checkmark$, but such a guess cannot lead to an accepting state).

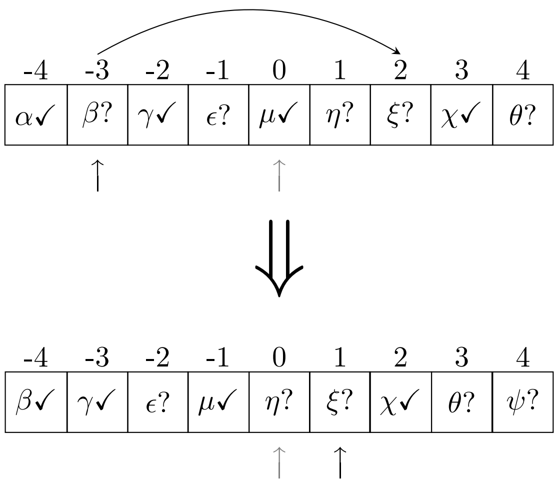

Figure 4.1: A single transition in the construction of Lemma 4.2.2.

In Figure 4.1, The semi-transparent arrow signifies the sequential head, while the opaque arrow is the "imaginary" jumping head. Here, the head jumps from -3 to 2, incurring a cost of 4, but in the indexing after the transition ξ is at index 1, thus the expression given for c' in the construction. Note that the letter being read must be μ, and that α must be checked, otherwise the run has failed.

Finally, the accepting states of $\mathcal{B}$ are $F' = \{(q, f, 1, c) \mid q \in F \wedge c \leq k \wedge f(j)_2 = ?$ for all $j > 0\}$. That is, the state of $\mathcal{A}$ is accepting, the overall cost is at most k, the location of the jumping head matches the sequential head (intuitively, location $n+1$), and no letter beyond the end of the tape has been used.

It is easy to verify that $\mathcal{B}$ indeed guesses a jump sequence and a corresponding run of $\mathcal{A}$ on the given word, provided that the jumping head stays within distance k of the sequential head. By Corollary 4.2, this restriction is complete, in the sense that if $\mathcal{A}_{\text{ABS}}(w) \leq k$ then there is a suitable jump sequence under this restriction with which w is accepted. ■

We can now easily conclude the decidability of the boundedness problems for the ABS semantics. The proof makes use of the decidability of emptiness for Parikh Automata [KR03].

Theorem 4.3. *The following problems are decidable for the ABS semantics: k-Boundedness, Param-Boundedness, Univ-k-Boundedness and Univ-Param-Boundedness.*

Proof. Consider a JFA $\mathcal{A}$ and $k \in \mathbb{N}$ (k is either fixed or given as input, which does not affect decidability), and let $\mathcal{B}$ be the NFA constructed as per Lemma 4.2.2. In order to decide Univ-k-Boundedness and Univ-Param-Boundedness, observe that $\mathcal{A}_{\text{ABS}}(w) \leq k$ for every word $w \in \Sigma^*$ if and only if $\mathfrak{L}(\mathcal{B}) = \Sigma^*$. Since the latter is decidable for NFA, we have decidability.

Similarly, in order to decide k-Boundedness and Param-Boundedness, observe that $\mathcal{A}_{\text{ABS}}(w) \leq k$ for every word $w \in \mathfrak{J}(\mathcal{A})$ if and only if $\mathfrak{J}(\mathcal{A}) \subseteq \mathfrak{L}(\mathcal{B})$. We can decide whether the latter holds by constructing the PA $(\overline{\mathcal{B}}, \mathcal{A})$ where $\overline{\mathcal{B}}$ is an NFA for the complement of $\mathfrak{L}(\mathcal{B})$, and checking emptiness. Since emptiness for PA is decidable [KR03], we conclude decidability. ■

With further scrutiny, we see that the size of $\mathcal{B}$ constructed as per Lemma 4.2.2 is polynomial in the size of $\mathcal{A}$ and single exponential in k. Thus, Univ-k-Boundedness is in fact decidable in PSPACE.

4.3 PSPACE-Hardness of boundedness for ABS

In the following, we complement the decidability result of Theorem 4.3 by showing that already Univ-k-Boundedness is PSPACE-hard, for every $k \in \mathbb{N}$.

We first observe that the absolute distance of every word is even. In fact, this true for every jump sequence.

Lemma 4.3.1. *Consider a jump sequence $\mathbf{a} = (a_0\ a_1\ \ldots\ a_n\ a_{n+1})$, then $\|\mathbf{a}\|$ is even.*

Proof. Observe that the parity of $|a_i - a_{i-1}|$ is the same as the parity of $a_i - a_{i-1}$. It follows that the parity of $\|\mathbf{a}\| = \sum_{i=1}^{n+1} \|a_i - a_{i_1}\| = \sum_{i=1}^{n+1} |a_i - a_{i-1}| - 1$ is the same as that of:

$$\sum_{i=1}^{n+1} |a_i - a_{i-1}| - 1 = (\sum_{i=1}^{n+1} a_1 - a_{i-1}) - (n+1) = n + 1 - (n+1) = 0$$

and is therefore even (the penultimate equality is due to telescopic sum). ■

We are now ready to prove the hardness of UNIV-k-BOUNDEDNESS. Observe that for a word $w \in \Sigma^*$ we have that $\mathcal{A}_{\mathrm{ABS}}(w) = 0$ if and only if $w \in \mathfrak{L}(\mathcal{A})$ (indeed, a cost of 0 implies that that an accepting jumping sequence is the sequential run $0, 1, \ldots, |w|+1$. In particular, we have that $\mathcal{A}_{\mathrm{ABS}}$ is 0-bounded if and only if $\mathfrak{L}(\mathcal{A}) = \Sigma^*$. Since the universality problem for NFAs is PSPACE-complete, this readily proves that UNIV-0-BOUNDEDNESS is PSPACE-hard. Note, however, that this does *not* imply that UNIV-k-BOUNDEDNESS is also PSPACE-hard for other values of k, and that the same argument fails for $k > 0$. We therefore need a slightly more elaborate reduction.

Lemma 4.3.2. *For ABS the UNIV-k-BOUNDEDNESS and k-BOUNDEDNESS are PSPACE-hard for every $k \in \mathbb{N}$.*

Proof. In order for the reduction to work both for UNIV-k-BOUNDEDNESS and k-BOUNDEDNESS, we start with an initial transformation of the given NFA $\mathcal{A}$ to an NFA $\mathcal{A}'$ as follows. Given $\mathcal{A}$, we introduce a fresh symbol \$ to the alphabet, and modify it so that reading \$ from every state can either stay at the same state, or transition to a new accepting state $q_{\$}$. From $q_{\$}$ no letters can be read. $\mathcal{A}'$ then satisfies the following property: if $\mathfrak{L}(\mathcal{A}) = \Sigma^*$ then $\mathfrak{L}(\mathcal{A}') = (\Sigma \bigcup \{\$\})^*$, and if $\mathfrak{L}(\mathcal{A}) \neq \Sigma^*$ then there exists a word $x \notin \mathfrak{L}(\mathcal{A}')$ such that $x \in \mathfrak{J}(\mathcal{A}')$. Indeed, for every (non-empty) word $x \notin \mathfrak{L}(\mathcal{A})$ we have that $\$x \notin \mathfrak{L}(\mathcal{A}')$ but $x\$ \in \mathfrak{L}(\mathcal{A}')$. We henceforth identify $\mathcal{A}$ with $\mathcal{A}'$, and use this property in the proof.

By Lemma 4.3.1, we can assume without loss of generality that k is even. Indeed, if there exists $m \in \mathbb{N}$ such that $\mathcal{A}_{\mathrm{ABS}}(w) \leq 2m+1$ for every $w \in \Sigma^*$, then by Lemma 4.3.1, we also have $\mathcal{A}_{\mathrm{ABS}}(w) \leq 2m$. Therefore, we can assume $k = 2m$ for some $m \in \mathbb{N}$.

We reduce the universality problem for NFAs to the UNIV-$2m$-BOUNDEDNESS problem. Consider an NFA $\mathcal{A} = \langle Q, \Sigma, \delta, Q_0, F \rangle$. We first check whether $\epsilon \in \mathfrak{L}(\mathcal{A})$ (i.e, we check whether $Q_0 \cap F \neq \emptyset$). If $\epsilon \notin \mathfrak{L}(\mathcal{A})$, we output some fixed unbounded automaton $\mathcal{B}$ (e.g., as in Example 3.3.1). Observe that since $\epsilon \notin \mathfrak{L}(\mathcal{A})$ then $\mathcal{A}$ is not universal, preserving the reduction correctness in this case. We assume from now on that $\epsilon \in \mathfrak{L}(\mathcal{A})$.

Now let $\heartsuit \notin \Sigma$ be a fresh symbol. Intuitively, we obtain from $\mathcal{A}$ an NFA $\mathcal{B}$ over the alphabet $\Sigma \cup \{\heartsuit\}$ such that $w \in \mathfrak{L}(\mathcal{B})$ if and only if the following hold:

1. Either w does not contain m occurrences of $\heartsuit$, or

2. w contains exactly m occurrences of $\heartsuit$, but does not start with $\heartsuit$, and $w|_{\Sigma} \in \mathfrak{L}(\mathcal{A})$ (where $w|_{\Sigma}$ is obtained from w by removing all occurrences of $\heartsuit$), or

3. $w = \heartsuit^m$ (think of this as an exception to Condition 2 where $w|_{\Sigma} = \epsilon$).

Constructing $\mathcal{B}$ from $\mathcal{A}$ is relatively straightforward by taking $m+1$ copies of $\mathcal{A}$ to track the number of $\heartsuit$s in the word. In particular, the reduction is in polynomial time.

We claim that $\mathfrak{L}(\mathcal{A}) = \Sigma^*$ if and only if $\mathcal{B}_{\mathrm{ABS}}$ is $2m$-bounded. For the first direction, assume $\mathfrak{L}(\mathcal{A}) = \Sigma^*$. Let $w \in (\Sigma \cup \{\heartsuit\})^*$.

1. If $w \in \mathfrak{L}(\mathcal{B})$ then $\mathcal{B}_{\text{ABS}}(w) = 0 \leq 2m$.

2. If $w \notin \mathfrak{L}(\mathcal{B})$ then w starts with $\heartsuit$ but has exactly m occurrences of $\heartsuit$ and at least one letter which is not $\heartsuit$. Let j be the first index which does not contain $\heartsuit$. Observe that $j \leq m+1$ since there are at most m consecutive $\heartsuit$s at the start of w. The following jump sequence $\mathbf{a}$ causes $\mathcal{B}$ to accept $w_{\mathbf{a}}$:

$$\mathbf{a} = (0\ j\ 1\ 2\ \ldots\ j-1\ j+1\ j+2\ \ldots\ |w|\ |w|+1)$$

   Indeed, $w_{\mathbf{a}}$ does not start with $\heartsuit$, has exactly m occurrences of $\heartsuit$, and $w|_{\Sigma} \in \mathfrak{L}(\mathcal{A}) = \Sigma^*$, so Condition 2 holds. Finally, note that $\|\mathbf{a}\| \leq 2m$ (since the only non-zero jumps are 0 to j, j to 1, and $j-1$ to $j+1$).

For the converse, assume $\mathfrak{L}(\mathcal{A}) \neq \Sigma^*$. Let $x \notin \mathfrak{L}(\mathcal{A})$. Since $\epsilon \in \mathfrak{L}(\mathcal{A})$ by the treatment of this case above, $x \neq \epsilon$. We also can assume by the identification of $\mathcal{A}$ with $\mathcal{A}'$ that $x \in \mathfrak{J}(\mathcal{A})$. Now consider $w = \heartsuit^m x$. We claim that $\mathcal{B}_{\text{ABS}}(w) > 2m$. Indeed, let $\mathbf{a}$ be a jump sequence such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{B})$ (if there isn't a jump sequence like that, then $\mathcal{B}_{\text{ABS}}(w) = \infty$ and we are done). Then, $a_1 \geq m+1$, contributing a cost of at least m to $\mathbf{a}$. It can easily be seen by induction over m that in order for $\mathbf{a}$ to cover the entries $1, \ldots, m$ starting at position $m+1$ and ending at position $m+2$, it requires cost of at least another m (moreover, if the ending position is greater than $m+2$, then the overall cost is already greater than $2m$). Then, however, in order for w_a to be accepted by $\mathcal{B}$, it must hold that $w_a|_{\Sigma} \neq x$, so $\mathbf{a}$ is not the identity starting from $m+1$. It therefore has an additional cost of at least 1. Thus, $\|\mathbf{a}\| > 2m$. In particular, $\mathcal{B}_{\text{ABS}}(w) > 2m$, so $\mathcal{B}$ is not $2m$-bounded. Note that $w \in \mathfrak{J}(\mathcal{B})$, since $x \in \mathfrak{J}(\mathcal{A})$. Thus, we are done both in the universal and non-universal settings. ■

Lemma 4.3.2 shows hardness for fixed k, and in particular when k is part of the input. Thus UNIV-PARAM-BOUNDEDNESS and PARAM-BOUNDEDNESS are also PSPACE-hard, and UNIV-k-BOUNDEDNESS is PSPACE-complete.

Chapter 5

The Reversal Semantics

5.1 The Membership Problem for REV

We now study the reversal semantics. Recall from Definition 3.2.2 that for a JFA $\mathcal{A}$ and a word w, the cost $\mathcal{A}_{\text{REV}}(w)$ is the minimal number of times the jumping head changes "direction" in a jump sequence for which w is accepted.

Consider a word w with $|w| = n$ and a jump sequence $\mathbf{a} = (a_0\ a_1\ \dots\ a_n\ a_{n+1})$. We say that an index $1 \leq i \leq n$ is a *turning index* if $a_i > a_{i-1}$ and $a_i > a_{i+1}$ (i.e., a right-to-left turn) or if $a_i < a_{i-1}$ and $a_i < a_{i+1}$ (i.e., a left-to-right turn). We denote by $\texttt{Turn}(\mathbf{a})$ the set of turning indices of $\mathbf{a}$.

Example 5.1.1. For example, consider the jump sequence $(\overset{a_0}{0}\ \overset{a_1}{2}\ \overset{a_2}{3}\ \overset{a_3}{5}\ \overset{a_4}{7}\ \overset{a_5}{4}\ \overset{a_6}{1}\ \overset{a_7}{6}\ \overset{a_8}{8})$, then $\texttt{Turn}(\mathbf{a}) = \{4, 6\}$.

Note that the cost of w is then $\mathcal{A}_{\text{REV}}(w) = \min\{\texttt{Turn}(\mathbf{a}) \mid w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})\}$. Viewed in this manner, we have that $\mathcal{A}_{\text{REV}}(w) \leq |w|$, and computing $\texttt{Turn}(\mathbf{a})$ can be done in polynomial time. The reader would recall that in order to prove Theorem 4.1 we used the fact that k can be bounded in polynomial time in the size of w. Thus, identically we have the following.

Theorem 5.1. *The problem of deciding, given $\mathcal{A}$ and k, whether $\mathcal{A}_{REV}(w) \leq k$ is NP-complete.*

Remark. For every jump sequence $\mathbf{a}$ we have that $|\texttt{Turn}(\mathbf{a})|$ is even, since the head starts at position 0 and ends at $n+1$, where after an odd number of turning points the direction is right-to-left, and hence cannot reach $n+1$.

5.2 Decidability of Boundedness Problems for REV

We begin by characterizing the words accepted using at most k reversals as a shuffle of subwords and reversed-subwords, as follows.

Definition 5.2.1. Let $x, y \in \Sigma^*$ be words, we define their *shuffle* to be the set of words obtained by interleaving parts of x and parts of y. Formally:

$$u \sqcup\!\sqcup v = \{x_1 \cdot y_1 \cdot x_2 \cdot y_2 \dots x_n \cdot y_n \mid \forall i\ x_i, y_i \in \Sigma^* \wedge u = x_1 x_2 \dots x_n \wedge\ v = y_1 y_2 \dots y_n\}$$

Example 5.2.2. If $x = aab$ and $y = cd$ then $x \sqcup\!\sqcup y$ contains the words $aabcd$, $acabd$, $caadb$, among others (the colors reflect which word each subword originated from). Note that the subwords may be empty, e.g., $caadb$, can be seen as starting with ϵ as a subword of x.

It is easy to see that $\sqcup\!\sqcup$ is a associative operation, so it can be extended to any finite number of words.

As we will soon see, intuitively, if $\mathcal{A}_{\text{REV}}(w) \leq k$, then w can be decomposed to a shuffle of at most $k+1$ subwords of itself, where all the even ones are reversed (representing the left-reading subwords). These subwords are the defined as follows.

Definition 5.2.3. Consider a word $w \in \Sigma^*$ and a jump sequence $\mathbf{a} \in J_{|w|}$. Write $\texttt{Turn}(\mathbf{a}) = \{i_1, i_2, \dots, i_l\}$ where $i_1 < i_2 < \dots < i_l$ and set $i_0 = 0$ and $i_{l+1} = n+1$. Then, for every $1 \leq j \leq l+1$, the *j-th turning subword of w with regard to* $\mathbf{a}$ is $s_j = w_{a_{i_{j-1}}} w_{a_{i_{j-1}+1}} \cdots w_{a_{i_j}-1}$.

Example 5.2.4. If $w = abcd$ and $\mathbf{a} = (\overset{a_0}{0}\ \overset{a_1}{2}\ \overset{a_2}{4}\ \overset{a_3}{1}\ \overset{a_4}{3}\ \overset{a_5}{5})$ then $\texttt{Turn}(\mathbf{a}) = \{2, 3\}$, $w_{\mathbf{a}} = bdac$, $s_1 = b$, $s_2 = d$ and $s_3 = ac$. In the case of sequential reading (e.g., when $\mathbf{a} = (0\ 1\ 2\ 3\ 4\ 5)$), $s_1 = abcd$ is the only turning subword of w.

Lemma 5.2.5. *Let $k \in \mathbb{N}$. Consider an NFA $\mathcal{A}$ and a word $w \in \Sigma^*$. Then $\mathcal{A}_{REV}(w) \leq k$ iff there exist words $s_1, s_2, \dots, s_{k+1} \in \Sigma^*$ such that the following hold.*

1. $s_1 s_2 \dots s_{k+1} \in \mathfrak{L}(\mathcal{A})$.
2. $w \in s_1 \sqcup\!\sqcup {s_2}^R \sqcup\!\sqcup s_3 \sqcup\!\sqcup {s_4}^R \sqcup\!\sqcup \dots \sqcup\!\sqcup s_{k+1}$ *(where ${s_i}^R$ is the reverse of s_i).*

Proof. For the first direction, assume $\mathcal{A}_{\text{REV}}(w) \leq k$, so there exists a jump sequence $\mathbf{a}$ such that $|\texttt{Turn}(\mathbf{a})| \leq k$ and $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$. Let $s_1, s_2, \dots, s_{l+1}$ be the turning subwords of w with regard to $\mathbf{a}$. If $l+1 < k+1$ we define $s_{l+2}, \dots, s_{k+1}$ to be ϵ. To avoid cumbersome indexing, we assume $l+1 = k+1$ in the following.

It is easy to see that conditions 1 and 2 hold for $s_1, s_2, \dots, s_{k+1}$. Indeed, by definition we have $s_1 s_2 \cdots s_{k+1} \in \mathfrak{L}(\mathcal{A})$, so condition 1 holds. For condition 2, observe that for every $1 \leq i \leq k+1$, if i is odd, then s_i consists of an ascending sequence of letters, and if i is even then s_i is a descending sequence. Since the s_i form a partition of the letters of w, we can conclude that $w \in s_1 \sqcup\!\sqcup {s_2}^R \sqcup\!\sqcup s_3 \sqcup\!\sqcup {s_4}^R \sqcup\!\sqcup \dots \sqcup\!\sqcup s_{k+1}$ (by shuffling the letters of these words to form exactly the sequence of indices $1, \dots, |w|$).

For the converse, consider words $s_1, s_2, \dots, s_{k+1}$ such that conditions 1 and 2 hold. By condition 2, we see that the word $s_1 s_2 \cdots s_{k+1}$ is a permutation of w, and moreover

- from the way w is obtained in $s_1 \sqcup\!\sqcup\, {s_2}^R \sqcup\!\sqcup\, s_3 \sqcup\!\sqcup\, {s_4}^R \sqcup\!\sqcup \ldots \sqcup\!\sqcup\, s_{k+1}$ we can extract a jump sequence $\mathbf{a}$ such that $w_{\mathbf{a}} = s_1 s_2 \cdots s_{k+1}$ and such that the turning subwords of $\mathbf{a}$ are exactly $s_1, {s_2}^R, \ldots, s_{k+1}$. Indeed, this follows from the same observation as above - for odd i we have that s_i is an increasing sequence of indices, and for even i it is decreasing. In particular, $|\mathtt{Turn}(\mathbf{a})| \leq k$, so $\mathcal{A}_{\mathrm{REV}}(w) \leq k$. ■

Using the characterization in Lemma 5.2.5, we can now construct a corresponding NFA, by intuitively guessing the shuffle decomposition and running copies of $\mathcal{A}$ and its reverse in parallel.

Lemma 5.2.6. *Let $k \in \mathbb{N}$ and consider a JFA $\mathcal{A}$. We can effectively construct an NFA $\mathcal{B}$ such that $\mathfrak{L}(\mathcal{B}) = \{w \in \Sigma^* \mid \mathcal{A}_{REV}(w) \leq k\}$.*

Proof. The overall plan is to construct $\mathcal{B}$ so that it captures the condition in Lemma 5.2.5. Intuitively, $\mathcal{B}$ keeps track of $k+1$ coordinates, each corresponding to a turning subword (that are nondeterministically constructed). The odd coordinates simulate the behavior of $\mathcal{A}$, whereas the even ones simulate the reverse of $\mathcal{A}$. In addition, $\mathcal{B}$ checks (using its initial and accepting states) that the runs on the subwords can be correctly concatenated. We proceed with the precise details.

Denote $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$. We construct $\mathcal{B} = \langle \Sigma, Q', \delta', Q'_0, F' \rangle$ as follows. $Q' = Q^{k+1}$, and the initial and final states are:

$$Q'_0 = \{(q_1, q_2, \ldots, q_{k+1}) \mid q_1 \in Q_0 \wedge q_i = q_{i+1} \text{ for all even } i\}$$
$$F' = \{(q_1, q_2, \ldots, q_{k+1}) \mid q_{k+1} \in F \wedge q_i = q_{i+1} \text{ for all odd } i\}$$

For the transiction function, we have that $(q'_0, q'_1, \ldots, q'_k) \in \delta'((q_0, q_1, \ldots, q_k), \sigma)$ if and only if there exists a single $1 \leq j \leq k+1$ such that $q'_j \in \delta(q_j, \sigma)$ if j is odd, and $q_j \in \delta(q'_j, \sigma)$ if j is even (this represents the reverse words in Lemma 5.2.5). In addition, for every $i \neq j$, it holds that $q'_i = q_i$.

We turn to show the correctness of $\mathcal{B}$. Consider an accepting run ρ of $\mathcal{B}$ on some word. Then ρ starts at state $(q_1, q_2, \ldots, q_{k+1}) \in Q'_0$ and ends at state $(s_1, s_2, \ldots, s_{k+1}) \in F'$. By the definition of δ', we can split ρ according to which component "progresses" in each transition, so that ρ can be written as a shuffle of run $\rho^1, \ldots, \rho^{k+1}$ where ρ^i leads from q_i to s_i in $\mathcal{A}$ if i is odd, and ρ^i leads from q_i to s_i in the *reverse* of $\mathcal{A}$ if i is even. The latter is equivalent to $(\rho^i)^R$ (i.e., the reverse run of ρ^i) leading from s_i to q_i in $\mathcal{A}$ if i is even.

We now observe that these runs can be concatenated as follows: Recall that $q_1 \in Q_0$ (by the definition of Q'_0). Then, ρ^1 leads from q_1 to s_1 in $\mathcal{A}$. By the definition of F we have $s_1 = s_2$, and $(\rho^2)^R$ leads from s_2 to q_2 in $\mathcal{A}$. Therefore, $\rho^1(\rho^2)^R$ leads from q_1 to q_2 in $\mathcal{A}$. Continuing in the same fashion, we have $q_2 = q_3$, and ρ^3 leading from q_3 to s_3, and so on up to s_{k+1}.

Thus, we have that $\rho^1(\rho^2)^R\rho^3\cdots(\rho^k)^R\rho^{k+1}$ is an accepting run of $\mathcal{A}$.

By identifying each accepting run ρ^i with the subword it induces (and reversing the subwords for even i), we have that $w \in \mathfrak{L}(\mathcal{B})$ if and only if there are words $s_1, \ldots, s_{k+1}$ such that the two conditions in Lemma 5.2.5 are satisfied. ∎

The proof of Lemma 9 shows that the size of $\mathcal{B}$ is polynomial in the size of $\mathcal{A}$ and single-exponential in k, giving us PSPACE membership for Univ-k-Boundedness. We can also conclude decidability for the rest of the boundedness problems using the same techniques as in the ABS case.

Theorem 5.2. *The following problems are decidable for the REV semantics: k-Boundedness, Param-Boundedness, Univ-k-Boundedness and Univ-Param-Boundedness.*

5.3 PSPACE-Hardness of Boundedness for REV

Following a similar scheme to the Absolute Distance Semantics of Chapter 4, observe that for a word $w \in \Sigma^*$ we have that $\mathcal{A}_{\text{REV}}(w) = 0$ if and only if $w \in \mathfrak{L}(\mathcal{A})$, which implies that Univ-0-Boundedness is PSPACE-hard. Yet again, the challenge is to prove hardness of Univ-k-Boundedness for all values of k.

Theorem 5.3. *For REV, Univ-k-Boundedness is PSPACE-complete for every $k \in \mathbb{N}$.*

Proof. By Remark 2 we can assume without loss of generality that k is even, and we denote $k = 2m$. We reduce the universality problem for NFAs to the Univ-$2m$-boundedness problem. Consider an NFA $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F\rangle$, and let $\heartsuit, \spadesuit \notin \Sigma$ be fresh symbols. We first check whether $\epsilon \in \mathfrak{L}(\mathcal{A})$. If $\epsilon \notin \mathfrak{L}(\mathcal{A})$, then $\mathfrak{L}(\mathcal{A}) \neq \Sigma^*$ and we output some fixed unbounded automaton $\mathcal{B}$ (e.g., as in Example 3.3.1).

Otherwise, we obtain from $\mathcal{A}$ an NFA $\mathcal{B}$ over the alphabet $\Sigma \bigcup \{\heartsuit, \spadesuit\}$ such that $w \in \mathfrak{L}(\mathcal{B})$ if and only if the following hold:

1. Either w does not contain exactly m occurrences of $\heartsuit$ and of $\spadesuit$, or
2. $w = (\heartsuit\spadesuit)^m x$ where $x \in \mathfrak{L}(\mathcal{A})$ (in particular $x \in \Sigma^*$).

Constructing $\mathcal{B}$ from $\mathcal{A}$ is straightforward as the union of two components: one that accepts words that satisfy condition 1 (using $2m+1$ states) and one for condition 2, which prepends to $\mathcal{A}$ a component with $2m$ states accepting $(\heartsuit\spadesuit)^m$. In particular, the reduction is in polynomial time.

We then have the following: if $\mathfrak{L}(\mathcal{A}) = \Sigma^*$, then for every $w \in (\Sigma \bigcup \{\heartsuit, \spadesuit\})^*$, if w satisfies condition 1, then $\mathcal{B}_{\text{REV}}(w) = 0$. Otherwise, w has exactly m occurrences of $\heartsuit$ and of $\spadesuit$. Denote the indices of $\heartsuit$ by $i_1 < i_2 < \ldots < i_m$ and of $\spadesuit$ by $j_1 < j_2 < \ldots < j_m$.

Also denote by $t_1 < t_2 < \ldots < t_r$ the remaining indices of w. Then consider the jump sequence

$$\mathbf{a} = (0\ i_1\ j_1\ i_2\ j_2\ \ldots\ i_m\ j_m\ t_1\ t_2\ \ldots\ t_r\ n+1)$$

We claim that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{B})$ by condition 2. Indeed, w starts with $(\heartsuit\spadesuit)^m$, followed by letters in Σ composing a word x. Since $x \in \mathfrak{L}(\mathcal{A}) = \Sigma^*$, we have that condition 2 holds. In addition, observe that $t_1 < t_2 < \ldots < t_r < n+1$, then $\texttt{Turn}(\mathbf{a}) \subseteq \{i_1, j_1, \ldots, i_m, j_m, t_0\}$, and in particular $|\texttt{Turn}(\mathbf{a})| \leq 2m+1$.

Moreover, by Remark 2 we know that $|\texttt{Turn}(\mathbf{a})|$ is even, so in fact $|\texttt{Turn}(\mathbf{a})| \leq 2m = k$. We conclude that $\mathcal{B}_{\text{REV}}(w) \leq k$, so $\mathcal{B}$ is k-bounded.

Conversely, if $\mathfrak{L}(\mathcal{A}) \neq \Sigma^*$, take $x \notin \mathfrak{L}(\mathcal{A})$ such that $x \neq \epsilon$ (which exists since we checked above that $\epsilon \in \mathfrak{L}(\mathcal{A})$). Consider the word $w \in \spadesuit^m\heartsuit^m x$, then have $w \notin \mathfrak{L}(\mathcal{B})$. We claim that $\mathcal{B}_{\text{REV}}(w) > 2m$. Indeed, if there exists $\mathbf{a}$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{B})$, then since w has exactly m occurrences of $\spadesuit$ and of $\heartsuit$, it must be accepted by condition 2. By the structure of w, the jump sequence $\mathbf{a}$ needs to permute $\spadesuit^m\heartsuit^m$ into $(\heartsuit\spadesuit)^m$. Intuitively, this means that the head must jump "back and forth" for $2m$ steps. More precisely, for every $i \in \{1, \ldots, |w|\}$ it holds that

$$a_i \in \begin{cases} \{m+1, \ldots, 2m\}, & i \leq 2m \text{ is odd} \\ \{1, \ldots, m\}, & i \leq 2m \text{ is even} \\ \{2m+1, \ldots, |w|\}, & i > 2m \end{cases}$$

In particular, $\{1, \ldots, 2m\} \subseteq \texttt{Turn}(\mathbf{a})$. Observe that the remaining suffix of $w_{\mathbf{a}}$ starting at $2m+1$ cannot be x, since $x \notin \mathfrak{L}(\mathcal{A})$, so $\mathbf{a}$ is not the identity starting from $2m+1$. It therefore has an additional reversal cost of at least 1. Thus, $|\texttt{Turn}(\mathbf{a})| > 2m$. In particular, $\mathcal{B}_{\text{REV}}(w) > 2m$, so $\mathcal{B}$ is not $2m$-bounded, and we are done. ■

As in Section 4.3, it follows that k-Boundedness, Univ-Param-Boundedness and Param-Boundedness are also PSPACE-hard.

Chapter 6

The Hamming Semantics

6.1 The Membership Problem for HAM

Recall from Definition 3.2.4 that for a JFA $\mathcal{A}$ and word w, the cost $\mathcal{A}_{\text{HAM}}(w)$ is the minimal Hamming distance between w and w' where $w' \sim w$ and $w' \in \mathfrak{L}(\mathcal{A})$.

We establish the complexity of the Membership problem for HAM.

Theorem 6.1. *The problem of deciding, given $\mathcal{A}$ and $k \in \mathbb{N}$, whether $\mathcal{A}_{HAM}(w) \leq k$ is NP-complete.*

Proof. By definition we have that $\mathcal{A}_{\text{HAM}}(w) \leq |w|$ for every word $w \in \mathfrak{J}(\mathcal{A})$. Thus, in order to decide whether $\mathcal{A}_{\text{HAM}}(w) \leq k$ we can nondeterministically guess a permutation $w' \sim w$ and verify that $w' \in \mathfrak{L}(\mathcal{A})$ and that $d_H(w, w') \leq k$. Both conditions are computable in polynomial time. Therefore, the problem is in NP.

Hardness follows (similarly to the proof of Theorem 4.1) by reduction from membership in JFA, noting that $w \in \mathfrak{J}(\mathcal{A})$ if and only if $\mathcal{A}_{\text{HAM}}(w) \leq |w|$. ■

6.2 Decidability of Boundedness Problems for HAM

Similarly to 4.2 Sections and 5.2, in order to establish the decidability of UNIV-PARAM-BOUNDEDNESS, we start by constructing an NFA that accepts the words w for which $\mathcal{A}_{\text{HAM}}(w) \leq k$.

Lemma 6.2.1. *Let $k \in \mathbb{N}$. We can effectively construct an NFA $\mathcal{B}$ with $\mathfrak{L}(\mathcal{B}) = \{w \in \Sigma^* \mid \mathcal{A}_{HAM}(w) \leq k\}$.*

Proof. Let $k \in \mathbb{N}$. Intuitively, $\mathcal{B}$ works as follows: while reading a word w sequentially, it simulates the run of $\mathcal{A}$, but allows $\mathcal{A}$ to intuitively "swap" the current letter with a (nondeterministically chosen) different one (e.g., the current letter may be a but the run of $\mathcal{A}$ can be simulated on either a or b). Then, $\mathcal{B}$ keeps track of the swaps made by counting for each letter a how many times it was swapped by another letter, and how many times another letter was swapped to it. This is done by keeping a counter ranging from $-k$ to k, counting the difference between the number of occurrences of each letter in the simulated word versus the actual word. We refer to this value as the *balance* of the letter. $\mathcal{B}$ also keeps track of the total number of swaps. Then, a run is accepting if at the end of the simulation, the total amount of swaps does not exceed k, and if all the letters end up with 0 balance.

We now turn to the formal details. Recall that $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, F \rangle$. We define $\mathcal{B} = \langle \Sigma, Q', \delta', Q'_0, F' \rangle$. The state space of $\mathcal{B}$ is $Q' = Q \times \{-k, \ldots, k\}^\Sigma \times \{0, \ldots, k\}$. We denote a state of $\mathcal{B}$ by (q, f, c) where $q \in Q$ is the current state of $\mathcal{A}$, $f : \Sigma \to \{-k, \ldots, k\}$ describes for each letter its balance and $c \in \{0, \ldots, k\}$ is the total number of swaps thus far.

The initial states of $\mathcal{B}$ are $Q'_0 = \{(q, f, 0) \mid q \in Q_0 \wedge f(\sigma) = 0 \text{ for all } \sigma \in \Sigma\}$. That is, we start in an initial state of $\mathcal{A}$ with balance and total cost of 0. The transition function is defines as follows. Consider a state (q, f, c) and a letter $\sigma \in \Sigma$, then $(q', f', c') \in \delta'((q, f, c), \sigma)$ if and only if either $q' \in \delta(q, \sigma)$ and $f' = f$ and $c' = c$, or there exists $\tau \in \Sigma$, $\tau \neq \sigma$ such that $q' \in \delta(q, \tau)$, $c' = c' + 1$, $f'(\sigma) = f(\sigma) - 1$, and $f'(\tau) = f(\tau) + 1$. That is, in each transition we either read the current letter σ, or swap for a letter τ and update the balances accordingly.

Finally, the accepting states of $\mathcal{B}$ are $F' = \{(q, f, c) \mid q \in F \wedge f(\sigma) = 0 \text{ for all } \sigma \in \Sigma\}$.

In order to establish correctness, we observe that every run of $\mathcal{B}$ on a word w induces a word w' (with the nondeterministically guessed letters) such that along the run the components f and c of the states track the swaps made between w and w'. In particular, c keeps track of the number of total swaps, and $\sum_{\sigma \in \Sigma} f(\sigma) = 0$. Moreover, for every word σ, the value $f(\sigma)$ is exactly the number of times σ was read in w' minus the number of times σ was read in w.

Since $\mathcal{B}$ accepts a word if $f \equiv 0$ at the last state, it follows that $\mathcal{B}$ accepts if and only if $w' \sim w$, and the run of $\mathcal{A}$ on w' is accepting. Finally, since c is bounded by k and is increased upon each swap, then limiting the image of f to values in $\{-k, \ldots, k\}$ does not pose a restriction, as they cannot go beyond these bounds without c going beyond the bound k as well. ■

An analogous proof to Theorem 4.3 gives us the following.

Theorem 6.2. *The following problems are decidable for the HAM semantics: k-BOUNDEDNESS, UNIV-k-BOUNDEDNESS and UNIV-PARAM-BOUNDEDNESS.*

We note that the size of $\mathcal{B}$ constructed in Lemma 6.2.1 is polynomial in k and single-exponential in $|\Sigma|$, and therefore when Σ is fixed and k is either fixed or given in unary, both UNIV-k-BOUNDEDNESS and UNIV-PARAM-BOUNDEDNESS are in PSPACE.

6.3 PSPACE-Hardness of Boundedness for HAM

Following a similar scheme to the Absolute Distance and Reversal Semantics of Chapters 4 and 5, observe that for a word $w \in \Sigma^*$ we have that $\mathcal{A}_{\text{HAM}}(w) = 0$ if and only if $w \in \mathfrak{L}(\mathcal{A})$, which implies that Univ-0-boundedness is PSPACE-hard. Also it is not hard to prove using similar tricks that Univ-k-Boundedness is PSPACE-hard. But in the HAM semantics case, since Univ-Param-Boundedness is already PSPACE-complete, then Univ-k-Boundedness is somewhat redundant. We therefore make do with the trivial lower bound whereby we reduce universality of NFA to Univ-0-Boundedness.

Theorem 6.3. *For HAM, the Univ-Param-Boundedness problem is PSPACE-complete for k coded in unary and fixed alphabet Σ.*

Chapter 7

Interplay Between the Semantics

Having established some decidability results, we now turn our attention to the interplay between the different semantics, in the context of boundedness. We show that for a given JFA $\mathcal{A}$, if $\mathcal{A}_{\text{ABS}}$ is bounded, then so is $\mathcal{A}_{\text{HAM}}$, and if $\mathcal{A}_{\text{HAM}}$ is bounded, then so is $\mathcal{A}_{\text{REV}}$. We complete the picture by showing that these are the only relationships - we give examples for the remaining cases.

Theorem 7.1. *Consider a JFA $\mathcal{A}$. If $\mathcal{A}_{HAM}$ is bounded, then $\mathcal{A}_{REV}$ is bounded.*

Proof. Consider a word $w \in \Sigma^*$, we show that if $\mathcal{A}_{\text{HAM}}(w) \leq k$ for some $k \in \mathbb{N}$ then $\mathcal{A}_{\text{REV}}(w) \leq 3k$. Assume $\mathcal{A}_{\text{HAM}}(w) \leq k$, then there exists a jump sequence $\mathbf{a} = (a_0 \ \dots \ a_{n+1})$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$ and $w_{\mathbf{a}}$ differs from w in at most k indices.

We claim that we can assume without loss of generality that for every index i such that $w_{a_i} = w_i$ we have that $a_i = i$ (i.e., i is a *fixed point*). Intuitively - there is no point swapping identical letters. Indeed, assume that this is not the case, and further assume that $\mathbf{a}$ has the minimal number of fixed points among such jump sequences. Thus, there exists some j for which $a_j \neq j$ but $w_{a_j} = w_j$ Let m be such that $a_m = j$, and consider the jump sequence $\mathbf{a}' = (a'_0, \dots, a'_{n+1})$ obtained from $\mathbf{a}$ by composing the swap $(a_j \ a_m)$. Then, for every $i \notin \{j, m\}$ we have that $a'_i = a_i$. In addition, $a'_j = a_m = j$ as well as $a'_m = a_j$. In particular, $\mathbf{a}'$ has more fixed points than $\mathbf{a}$ (exactly those of $\mathbf{a}$ and j). However, we claim that $w_{\mathbf{a}} = w_{\mathbf{a}'}$. Indeed, the only potentially problematic coordinates are a_j and a_m. For j we have $w_{a_j} = w_j = w_{a'_j}$ and for m we have $w_{a'_m} = w_{a_j} = w_j = w_{a_m}$. This is a contradiction to $\mathbf{a}$ having a minimal number of fixed points, so we conclude that no such coordinate $a_j \neq j$ exists.

Next, observe that $\texttt{Turn}(\mathbf{a}) \subseteq \{i \mid a_i \neq i \vee a_{i+1} \neq i+1 \vee a_{i-1} \neq i-1\}$. Indeed, if $a_{i-1} = i-1$, $a_i = i$ and $a_{i+1} = i+1$ then clearly i is not a turning index. By the property established above, we have that $w_{a_i} = w_i$, if and only if $a_i = i$. It follows that $\texttt{Turn}(\mathbf{a}) \subseteq \{i \mid w_{a_i} \neq w_i \vee w_{a_{i+1}} \neq w_{i+1} \vee w_{a_{i-1}} \neq w_{i-1}\}$, so $|\texttt{Turn}(\mathbf{a})| \leq 3k$ (since each index where $w_{\mathbf{a}} \neq w$ is counted at most 3 times in the latter set). ■

Theorem 7.2. *Consider a JFA $\mathcal{A}$. If $\mathcal{A}_{ABS}$ is bounded, then $\mathcal{A}_{HAM}$ is bounded.*

Proof. Consider a word $w \in \Sigma^*$, we show that if $\mathcal{A}_{\text{ABS}}(w) \leq k$ for some $k \in \mathbb{N}$ that $\mathcal{A}_{\text{HAM}}(w) \leq (2k+1)(k+1)$. Assume $\mathcal{A}_{\text{ABS}}(w) \leq k$, then there exists a jump sequence $\mathbf{a} = (a_0 \ \ldots \ a_{n+1})$ such that $\|\mathbf{a}\| \leq k$ and $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$. In the following we show that $a_i = i$ for all but $(2k+1)(k+1)$ indices, i.e., $|i \mid a_i \neq i| \leq (2k+1)(k+1)$.

It is convenient to think of the jumping head moving according to $\mathbf{a}$ in tandem with a sequential head moving from left to right. Recall that by Lemma 4.2.1, for every index i we have that $i - k \leq a_i \leq i + k$, i.e. the jumping head stays within distance k from the sequential head.

Consider an index i such that $a_i \neq i$ (if there is no such index, we are done). We claim that within at most $2k$ steps, $\mathcal{A}$ performs a jump of cost at least 1 according to $\mathbf{a}$. More precisely, there exists $i + 1 \leq j \leq i + 2k$ such that $|a_j - a_{j-1}| > 1$. To show this we split to two cases:

- If $a_i > i$, then there exists some $m \leq i$ such that m has not yet been visited according to $\mathbf{a}$ (i.e., by step i). Index m must be visited by a_i within at most k steps (otherwise it becomes outside the $i - k, i + k$ window around the sequential head), and since $a_i > i$, it must perform a "left jump" of size at least 2 (otherwise it always remains to the right of the sequential reading head).

- If $a_i < i$, the there exists some $m \geq i$ such that m has already been visited by step i according to $\mathbf{a}$. Therefore, within at most $2k$ steps, the jumping head must skip at least over this position (think of m as a hurdle coming toward the jumping head, which must stay within distance k of the sequential head and therefore has to skip over it). Such a jump incurs a cost of at least 1.

Now, let $B = \{i \mid a_i \neq i\}$ and assume by way of contradiction that $|B| > (2k+1)(k+1)$. By the above, for every $i \in B$, within $2k$ steps the run incurs a cost of at least 1. While some of these intervals of $2k$ steps may overlap, we can still find at least $k + 1$ such disjoint segments (indeed, every $i \in B$ can cause an overlap with at most $2k$ other indices). More precisely, there are $i_1 < i_2 < \ldots < i_{k+1}$ in B such that $i_j > i_{j-1} + 2k$ for all j, and therefore each of the costs incurred within $2k$ steps of visiting i_j is independent of the others. This, however, implies that $\|\mathbf{a}\| \geq k + 1$, which is a contradiction, so $|B| \leq (2k+1)(k+1)$.

It now follows that $\mathcal{A}_{\text{HAM}}(w) = |\{i \mid w_{a_i} \neq w_i\}| \leq |\{i \mid a_i \neq i\}| \leq (2k+1)(k+1)$. ■

Combining Lemmas 7.1 and 7.2 we have the following.

Corollary 7.3. *Consider a JFA $\mathcal{A}$. If $\mathcal{A}_{ABS}$ is bounded, then $\mathcal{A}_{REV}$ is bounded.*

We proceed to show that no other implication holds with regard to boundedness, by demonstrating languages for each possible choice of bounded/unbounded semantics (c.f. Remark 1). The examples are summarized in Table 7.1, and are below.

ABS	HAM	REV	Language
Bounded	Bounded	Bounded	$(a+b)^*$
Unbounded	Bounded	Bounded	$(a+b)^*a$
Unbounded	Unbounded	Bounded	a^*b^*
Unbounded	Unbounded	Unbounded	$(ab)^*$

Table 7.1: Examples for every possible combination of bounded/unbounded semantics. The languages are given by regular expressions (e.g., $(a+b)^*a$ is the language of words that end with a).

Example 7.0.1. The language $(a+b)^*$ is bounded in all semantics. This is trivial, since every word is accepted, and in particular has cost 0 in all semantics.

Example 7.0.2. The language $(a+b)^*a$ is bounded in HAM and REV semantics, but unbounded in ABS. Indeed, let $\mathcal{A}$ be an NFA such that $\mathfrak{L}(\mathcal{A}) = (a+b)^*a$ and consider a word $w \in \mathfrak{J}(\mathcal{A})$, then w has at least one occurrence of a at some index i. Then, for the jumping sequence $\mathbf{a} = (0, 1, 2, \ldots, i-1, n, i+1, \ldots, n-1, i, n+1)$ we have that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$. Observe that $d_H(w_{\mathbf{a}}, w) \leq 2$ (since $w_{\mathbf{a}}$ differs from w only in indices i and n), and $\texttt{Turn}(\mathbf{a}) \subseteq \{i, n\}$, so $\mathcal{A}_{\text{HAM}} \leq 2$ and $\mathcal{A}_{\text{REV}} \leq 2$.

For ABS, however, consider the word ab^n for every $n \in \mathbb{N}$. Since the letter a must be read last, then in any jumping sequence accepting the word, there is a point where the jumping head is at index n and the sequential head is at position 1. By Lemma 4.2.1, it follows that $\mathcal{A}_{\text{ABS}}(w) \geq n-1$, and by increasing n, we have that $\mathcal{A}_{\text{ABS}}$ is unbounded.

Example 7.0.3. The language a^*b^* is bounded in the REV semantics, but unbounded in HAM and ABS. Indeed let $\mathcal{A}$ be an NFA such that $\mathfrak{L}(\mathcal{A}) = a^*b^*$ and consider a word $w \in \mathfrak{J}(\mathcal{A})$, and denote by $i_1 < i_2 < \ldots < i_k$ the indices of a's in w in increasing order, and by $j_1 < j_2 < \ldots < j_{n-k}$ the indices of b's in decreasing order. Then, for the jumping sequence $\mathbf{a} = (i_1 \ \ldots \ i_k \ j_1 \ \ldots \ j_{n-k} \ n+1)$ we have that $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$, and $\mathcal{A}_{\text{REV}} \leq 2$ (since the jumping head goes right reading all the a's, then left reading all the b's, then jumps to $n+1$).

For HAM, consider the word $w = b^n a^n$ for every $n \in \mathbb{N}$. The only permutation of w that is accepted in $\mathfrak{L}(\mathcal{A})$ is $w' = a^n b^n$, and $d_H(w, w') = 2n$ so $\mathcal{A}_{\text{HAM}}$ is unbounded. By Lemma 7.2 it follows that $\mathcal{A}_{\text{ABS}}$ is also unbounded.

Example 7.0.4. The language $(ab)^*$ is unbounded in all the semantics. Indeed, let $\mathcal{A}$ be an NFA such that $\mathfrak{L}(\mathcal{A}) = (ab)^*$, then by Lemma 7.1 and Corollary 7.3 it suffices to show that $\mathcal{A}_{\text{REV}}$ is unbounded.

Consider the word $w = b^n a^n$ for every $n \in \mathbb{N}$, and let $\mathbf{a} = (a_0 \ a_1 \ \ldots \ a_{2n} \ a_{2n+1})$ such that $w_{\mathbf{a}} \in (ab)^*$, then for every odd i we have $a_i \in \{n+1, \ldots, 2n\}$ and for every even $i \leq 2n$ we have $a_i \in \{1, \ldots, n\}$. In particular, every index $1 \leq i \leq 2n$ is a turning point, so $\mathcal{A}_{\text{REV}}(w) = 2n$ and $\mathcal{A}_{\text{REV}}$ is unbounded.

Chapter 8

Conclusion and open questions

8.1 Conclusion

Quantitative semantics are often defined by externally adding some quantities (e.g., weights) to a finite-state model, usually with the intention of explicitly reasoning about some unbounded domain. It is rare and pleasing when quantitative semantics arise naturally from a Boolean model. In this work, we studied three such semantics: the Absolute Distance semantic, the Reversal semantic and the Hamming semantic.

We established decidability for some boundedness problems variants for these semantics, and gave lower bounds for some fragments (see Chapters 4-6).

Curiously, despite the semantics being intuitively unrelated, it turns out that they give rise to interesting interplay (see Chapter 7).

8.2 An Open question

We argue that boundedness is a fundamental decision problem for the semantics we introduce, as it measures whether one can make do with a certain budget for jumping. An open question left in this research is *existentially-quantified boundedness*: whether there *exists* some bound k for which $\mathcal{A}_{\text{SEM}}$ is k-bounded. This problem seems technically challenging, as in order to establish its decidability, we would need to upper-bound the minimal k for which the automaton is k-bounded, if it exists. The difficulty arises from two fronts: first, standard methods for showing such bounds involve some pumping argument. However, the presence of permutations makes existing techniques inapplicable. We expect that a new toolbox is needed to give such arguments. Second, the constructions we present for Univ-Param-Boundedness in the various semantics seem like the natural approach to take. Therefore, a sensible direction for the existential case is to analyze these constructions with a parametric k. The systems obtained this way, however, do not fall into (generally) decidable cases. For example, in the HAM semantics, using a parameter k we can construct a labelled VASS. But the latter do not admit decidable properties for the corresponding boundedness problem. We remark that it is conceptually possible that existential-boundedness is decidable without the bound being constructive. This, however, seems somewhat unlikely, and we do not have any reasonable techniques to tackle this problem in a non-constructive manner.

Bibliography

[Alm20] Shaull Almagor. Process symmetry in probabilistic transducers. In *Foundations of Software Technology and Theoretical Computer Science*, 2020.

[AY23] Shaull Almagor and Omer Yizhaq. Jumping automata over infinite words. In Frank Drewes and Mikhail Volkov, editors, *Developments in Language Theory*, pages 9–22, Cham, 2023. Springer Nature Switzerland.

[Bok21] Udi Boker. Quantitative vs. weighted automata. In Paul C. Bell, Patrick Totzke, and Igor Potapov, editors, *Reachability Problems*, pages 3–18, Cham, 2021. Springer International Publishing.

[CFM11] Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Bounded parikh automata. *Int. J. Found. Comput. Sci.*, 23:1691–1710, 2011.

[CFM12] Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Affine parikh automata. *RAIRO Theor. Informatics Appl.*, 46:511–545, 2012.

[DKV09] Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata.* Springer Publishing Company, Incorporated, 1st edition, 2009.

[FHY21] Szilárd Zsolt Fazekas, Kaito Hoshi, and Akihiro Yamamura. Two-way deterministic automata with jumping mode. *Theoretical Computer Science*, 864:92–102, 2021.

[FPS15] Henning Fernau, Meenakshi Paramasivan, and Markus L Schmid. Jumping finite automata: characterizations and complexity. In *Implementation and Application of Automata: 20th International Conference, CIAA 2015, Umeå, Sweden, August 18-21, 2015, Proceedings 20*, pages 89–101. Springer, 2015.

[FS17] Paramasivan Fernau and Vorel Schmid. Characterization and complexity results on jumping finite automata. *Theoretical Computer Science*, 679:31–52, 2017.

[KR03] Felix Klaedtke and Harald Ruess. Monadic second-order logics with cardinalities. In *International Colloquium on Automata, Languages and Programming*, 2003.

[LPS14] Giovanna J Lavado, Giovanni Pighizzini, and Shinnosuke Seki. Operational state complexity under parikh equivalence. In *Descriptional Complexity of Formal Systems: 16th International Workshop, DCFS 2014, Turku, Finland, August 5-8, 2014. Proceedings 16*, pages 294–305. Springer, 2014.

[Moh03] Mehryar Mohri. Edit-distance of weighted automata. In Jean-Marc Champarnaud and Denis Maurel, editors, *Implementation and Application of Automata*, pages 1–23, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[MS72] Albert R Meyer and Larry J Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT*, volume 72, pages 125–129, 1972.

[MZ12] Alexander Meduna and Petr Zemek. Jumping finite automata. *International Journal of Foundations of Computer Science*, 23(07):1555–1578, 2012.

[NA21] Antonio Abu Nassar and Shaull Almagor. Simulation by rounds of letter-to-letter transducers. *Log. Methods Comput. Sci.*, 19, 2021.

[Vor18] Vojtěch Vorel. On basic properties of jumping finite automata. *International Journal of Foundations of Computer Science*, 29(01):1–15, 2018.

בעייה פתוחה

אנו טוענים שבעיית החסם הוא בעיית הכרעה מהותית עבור הסמנטיקות שאנו מציגים, שכן היא מודדת האם אפשר להסתפק בתקציב מסוים לצורך קבלת מילה ע"י אוטומט קופץ. שאלה פתוחה שנותרה במחקר זה היא בעיית קיום החסם: האם קיים k כך ששפת האוטומט הקופץ חסומה ע"י k. הבעיה הזו נראית מאתגרת במיוחד, שכן כדי לבסס את הכריעות שלה, נצטרך לחסום מלמעלה את ה-k המינימלי שעבורו האוטומט חסום ע"י k, אם הוא קיים. הקושי נובע משתי סיבות: ראשית, שיטות סטנדרטיות להצגת חסמים כאלה כרוכות בטיעון ניפוח. עם זאת, הנוכחות של תמורות גורמות לטכניקות קיימות להיות לא ישימות. אנו מצפים שיש צורך בארגז כלים חדש כדי להוכיח טיעונים כאלה. בנוסף, הבניות שאנו מציגים עבור בעיית החסם בסמנטיקות השונות נראות כמו הגישה הטבעית לנקוט. לכן, כיוון הגיוני לבעיית הקיום הוא לנתח את הבניות הללו עם k פרמטרי. המערכות שמתקבלות בדרך זו, עם זאת, לא מסתכמים (בדרך כלל) למקרים הניתנים להכרעה.

- בסמנטיקת הסיבוב העלות של ρ היא מספר הפעמים שהראש "מסתובב" בזמן קריאת w (כלומר הוא משתנה מתנועה משמאל לימין לתנועה מימין לשמאל או להיפך).

לאחר מכן אנו מגדירים את העלות של המילה w לפי כל סמנטיקה על ידי מציאת הריצה שממזערת את העלות.

לפיכך, אנו מתייחסים לאוטומטים קופצים כמודלים כמותיים ולא כמודלים בוליאניים. אולם בניגוד לאוטומטים כמותיים אחרים, הסמנטיקות שהגדרנו נובעות באופן טבעי מהמודל. יתרה מכך, ההגדרות נובעות באופן טבעי על ידי סוגים שונים של גישות לזיכרון, כפי שאנו מדגימים כעת. ראשית, נסתכל על מערכת שהזיכרון שלה מונח במערך (לדוגמא, קלטת), עם ראש קורא שיכול לנוע לאורך הקלטת. הזזת הראש דורשת אנרגיה, ולכן סך האנרגיה המושקעת בקריאת הקלט תואמת לסמנטיקת הערך המוחלט. לעומת זאת, במערכת שהזיכרון שלה הוא דיסק מסתובב, הראש נשאר במקומו והדיסק מבצע את התנועה. לכן, זול להמשיך להסתובב באותו כיוון, והעלות העיקרית היא בהיפוך הכיוון, המצריך עצירה והיפוך מנוע. סמנטיקת ההיפוך תופסת בצורה הטובה ביותר את עלות התנועה במקרה הזה. לבסוף, נסתכל על מערכת שקוראת את הקלט שלה באופן סדרתי, אבל מותר לערוך את הקלט שלה על ידי החלפת אות אחת באחרת, כך שבסוף המילה המתקבלת היא תמורה של המילה המקורית. העריכות המינימליות הנדרשות מתאימות לסמנטיקת האמינג.

שאלות כמותיות

אנו מגדירים וחוקרים מספר בעיות בנוגע לסמנטיקה זו. בעיית השייכות קובעת בהינתן אוטומט קופץ אם מילה w מוגבלת במספר נתון כלשהו k. אנו מראים שבעיית השייכות היא NP שלמה תחת שלוש הסמנטיקות.

אנו חוקרים גם את בעיית החסם: בהינתן אוטומט קופץ, מחליטים האם ערכי כל המילים בשפה (הכמותית) שלו חסומים ע"י k. אנו מראים כי בעיה זו כריעה בשלוש הסמטיקות ונותנים חסמים תחתונים למספר ווריאציות של הבעיה.

על מנת להוכיח את הכריעות של בעיית החסם תחת אחת מהסמטיקות, אנו בונים אוטומט סופי לא דטרמיניסטי $\mathcal{B}$ (אשר קורא את קלטו באופן סדרתי) המקבל את שפת כל המילים שמחירן קטן מ-k. לצורך בניית $\mathcal{B}$ אנו נעזרים בתכונות מסויימות של הסמנטיקה אותה אנחנו חוקרים. לדוגמא, בסמנטיקת הערך המוחלט ניתן לחסום ע"י k את המרחק בין הראש הסדרתי לראש הקופץ בכל צעד במהלך קריאת הקלט.

במחקרנו אנו מציגים גם מספר קשרים בין בעיות החסם של הסמנטיקות: אנו מראים כי אם אוטומט הינו חסום תחת סמנטיקת האמינג, הוא חסום גם תחת סמנטיקת ההיפוך. באופן דומה, חסימות של סמנטיקת הערך המוחלט גוררת חסימות תחת סמנטיקת המינג. אנו מציגים גם דוגמאות המראות שאין גרירות דומות אחרות.

תקציר

אוטומטים קופצים

אוטומטים סופיים לא-דטרמיניסטיים בדרך כלל קוראים את הקלט שלהם באופן רציף כפי שקורה ברוב המודלים החישוביים מבוססי מצבים. לעומת זאת, אוטומט סופי קופץ עשוי לקרוא את הקלט שלו באופן לא רציף, לקפוץ מאות לאות, כל עוד כל אות נקראת פעם אחת בדיוק. אוטומט קופץ רלוונטי במקרים שבהם אין חשיבות לסדר הקלט. דוגמה אחת כזו היא כאשר הקלט מייצג את המשאבים הזמינים, ואנו רוצים רק לחקור לגבי הכמות שלהם. מנקודת מבט תיאורטית יותר של שפות פורמליות, זה מסתכם בהסתכלות על הסגור הקומוטטיבי של השפות, המכונה גם תמונת הפריק שלהן.

בעוד אוטומטים קופצים הם מודל אטרקטיבי ופשוט, הם מציגים חיסרון כשחושבים עליהם כמודל למערכות. זאת מכיוון שניתן לחשוב על תנועת הראש כעל תהליך פיזי של גישה לאחסון הקלט של האוטומט הקופץ. לכן, במקרים מסוימים התנועה צריכה להיות זולה יותר מאשר במקרים אחרים. למשל, כאשר הראש זז בצורה סדרתית במהלך קריאת הקלט (ללא קפיצות). היכולת לקפוץ מסביב קשה יותר פיזית ולכן היא לא צריכה לבוא "חינם".

סמנטיקות כמותיות

בעבודתנו אנו מציגים שלוש סמנטיקות כמותיות המנסות לכמת את עלות הקפיצה. למטרותינו אנו מתייחסים לאוטומטים קופצים כאל פונקציה ממילים לעלויות, וכך אנו תופסים כמה יקר לקבל מילה נתונה ביחס לקפיצות הראש. אנו רוצים לחקור את המאפיינים של סמנטיקות אלו, את ההבדלים ביניהן ותכונות אחרות.

אנו מציגים בקצרה את שלוש הסמנטיקות השונות: נסתכל על אוטומט קופץ $\mathcal{A}$ ומילה w, ותהי ρ ריצה מקבלת של w ב-$\mathcal{A}$. הריצה ρ במקרה שלנו מציינת את רצף המצבים והאינדקסים שבהם ביקרנו ב-w. תחילה נגדיר את העלות של ריצה בודדת.

- בסמנטיקת האמינג, אנו מסתכלים על המילה w' המושרה על ידי ρ, כלומר, המילה המתקבלת בקריאת w בסדר ρ קוראת אותה. העלות של ρ היא מספר האותיות שבהן w' שונה מ-w.
- בסמנטיקת הערך המוחלט העלות של ρ היא סכום אורכי הקפיצות שהיא עושה.

המחקר בוצע בהנחייתו של ד"ר שאול אלמגור, בפקולטה למדעי המחשב.

מחבר חיבור זה מצהיר כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו', נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

סמנטיקות כמותיות לאוטומטים קופצים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

ישי סלגדו

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
סיון התשפ"ד   חיפה   יוני 2024

סמנטיקות כמותיות לאוטומטים קופצים

ישי סלגדו