Jumping Automata Must Pay*

Shaull Almagor
Technion, Israel
shaull@technion.ac.il †

Ishai Salgado
Technion, Israel
ishaisalgado@campus.technion.ac.il

Jumping automata are finite automata that read their input in a non-sequential manner, by allowing a reading head to "jump" between positions on 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 distance* measures the number of letter-swaps the run makes.

We study these measures, with the main focus being the *boundedness problem*: given a jumping automaton, decide whether its (quantitative) languages is bounded by some given number k. We establish the decidability and complexity for this problem under several variants.

1 Introduction

Traditional automata read their input sequentially. Indeed, this is the case for most state-based computational models. In some settings, however, we wish to abstract away the order of the input letters. For example, 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 languages, a.k.a. their *Parikh image*. To capture this notion in a computation model, *Jumping Automata (*JFAs*)* were introduced in [19]. A jumping automaton may read its input in a non-sequential manner, jumping from letter to letter, as long as every letter is read exactly once. Several works have studied the algorithmic properties and expressive power of these automata [11, 12, 21, 10, 17, 4].

While JFAs are an attractive and simple model, they present a shortcoming when thought of as 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, sequential access is the most basic form of access and can be considered "cheap", but allowing the head to jump around is physically more difficult and therefore should incur a cost.

To address this, we present three *quantitative semantics* for JFAs, whereby a JFA represents a function from words to costs, which captures how expensive it is to accept a given word with respect to the head jumps. The different semantics capture different aspects of the cost of jumps, as follows.

Consider a JFA $\mathscr{A}$ and a word w, and let ρ be an accepting run of $\mathscr{A}$ on w. The run ρ specifies the sequence of states and indices visited in w. We first define the cost of individual runs.

- In the *Absolute Distance* semantics (ABS), the cost of ρ is the sum of the lengths of jumps it makes.
- In the *Reversal* semantics (REV), the cost of ρ is the number of times the reading head changes its direction (i.e., moving from right to left or from left to right).

*The full version can be found at the authors' homepages.

†This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 989/22). We also thank Michaël Cadilhac for fruitful discussions about this submission.

A. Achilleos and A. Francalanza (Eds.): Fifteenth International Symposium on Games, Automata, Logics, and Formal Verification (GandALF 2024).
EPTCS 409, 2024, pp. 19–34, doi:10.4204/EPTCS.409.6

- In the *Hamming* semantics (HAM), we consider the word w' induced by ρ, i.e., the order of letters that ρ reads. Then, the cost of ρ is the number of letters where w' differs from w.

We then define the cost of the word w in $\mathcal{A}$ according to each semantics, by taking the run that minimizes the cost.

Thus, we lift JFAs from a Boolean automata model to the rich setting of *quantitative* models [5, 3, 8, 9]. 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[1], and the main cost is in reversing the direction, which requires stopping and reversing a motor. 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* [20, 13] under a restriction of maintaining the amount of resources. Then, the minimal edits required correspond to the HAM semantics.

Example 1. *Consider a (standard)* NFA $\mathcal{A}$ *for the language given by the regular expression* $(ab)^*$ *(the concrete automaton chosen is irrelevant, see Remark 6). As a* JFA*,* $\mathcal{A}$ *accepts a word* w *if and only if* w *has an equal number of a's and b's. To illustrate the different semantics, consider the words* $w_1 = abababaab$ *and* $w_2 = ababbaba$*, obtained from* $(ab)^4$ *by flipping the third ab pair (*w_1*) and the third and fourth pairs (*w_2*). As we define in Section 3, we think of runs of the* JFA $\mathcal{A}$ *as if the input is given between end markers at indices* 0 *and* $n+1$*, and the jumping run must start at* 0 *and end in* $n+1$*.*

- *In the* ABS *semantics, the cost of* w_1*, denoted* $\mathcal{A}_{\text{ABS}}(w_1)$*, is* 2*: the indices read by the head are* $0,1,2,3,4,6,5,7,8,9$*, so there are two jumps of cost* 1*: from* 4 *to* 6 *and from* 5 *to* 7*. Similarly, we have* $\mathcal{A}_{\text{ABS}}(w_2) = 4$*, e.g., by the sequence* $0,1,2,3,4,6,5,8,7,9$*, which has two jumps of cost* 1 *(*4 *to* 6 *and* 7 *to* 9*), and one jump of cost* 2 *(*5 *to* 8*). (formally, we need to prove that there is no better run, but this is not hard to see).*
- *In the* REV *semantics, we have* $\mathcal{A}_{\text{REV}}(w_1) = 2$ *by the same sequence of indices as above, as the head performs two "turns", one at index* 6 *(from* $\rightarrow$ *to* $\leftarrow$*) and one at* 5 *(from* $\leftarrow$ *to* $\rightarrow$*). Here, however, we also have* $\mathcal{A}_{\text{REV}}(w_2) = 2$*, using the sequence* $0,1,2,3,4,6,8,7,5,9$*, whose turning points are* 8 *and* 5*.*
- *In the* HAM *semantics we have* $\mathcal{A}_{\text{HAM}}(w_1) = 2$ *and* $\mathcal{A}_{\text{HAM}}(w_2) = 4$*, since we must change the letters in all the flipped pairs for the words to be accepted.*

Example 2. *Consider now an* NFA $\mathcal{B}$ *for the language given by the regular expression* a^*b^**. Note that as a* JFA*,* $\mathcal{B}$ *accepts* $\{a,b\}^*$*, since every word can be reordered to the form* a^*b^**.*

Observe that in the REV *semantics, for every word* w *we have* $\mathcal{B}_{\text{REV}}(w) \leq 2$*, since at the worst case* $\mathcal{B}$ *makes one left-to-right pass to read all the a's, then a right-to-left pass to read all the b's, and then jump 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 costs can become unbounded. Indeed, in order to accept words of the form* b^na^n*, 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, again incurring a high cost.*

[1] We assume that the head does not return back to the start by continuing the spin, but rather reaches some end.

Related work Jumping automata were introduced in [19]. We remark that [19] contains some erroneous proofs (e.g., closure under intersection and complement, also pointed out in [12]). The works in [11, 12] establish several expressiveness results on jumping automata, as well as some complexity results. In [21] many additional closure properties are established. An extension of jumping automata with a two-way tape was studied in [10], and jumping automata over infinite words were studied by the first author in [4].

When viewed as the commutative image of a language, jumping automata are closely related to Parikh Automata [16, 7, 6, 14], 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 latter 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 [2] in a jumping-flavour, and in [1] 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. We refer the reader to [5, 9] and the references therein.

Contribution and paper organization Our contribution consists of the introduction of the three jumping semantics, and the study of decision problems pertaining to them (defined in Section 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 bounded by some constant k. We establish the decidability of this problem for all the semantics, and consider the complexity of some fragments. More precisely, we consider several variants of boundedness, depending on whether the bound is a fixed constant, or an input to the problem, and on whether the jumping language of $\mathcal{A}$ is universal. Our complexity results are summarized in Table 1.

Our paper is organized as follows: the preliminaries and definitions are given in Sections 2 and 3. Then, each of Sections 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 Section 7 we give a complete picture of the interplay between the different semantics (using some of the results established beforehand). Finally, in Section 8 we discuss some exciting open problems. Due to lack of space, some proofs appear in the full version.

	k-BND	PARAM-BND	UNIV-k-BND	UNIV-PARAM-BND	
				unary	binary
ABS	Decidable PSPACE-h	Decidable PSPACE-h	PSPACE-c	EXPSPACE PSPACE-h	2-EXPSPACE PSPACE-h
REV	Decidable PSPACE-h	Decidable PSPACE-h	PSPACE-c	EXPSPACE PSPACE-h	2-EXPSPACE PSPACE-h
HAM	Decidable PSPACE-h	Decidable PSPACE-h	PSPACE-c	PSPACE-c	EXPSPACE PSPACE-h

Table 1: Complexity results of the various boundedness problems for the three semantics. The complexity of membership is NP-complete for all the semantics. The "Decidable" entries depend on the complexity of the containment problem for Parikh Automata.

2 Preliminaries and Definitions

For a finite alphabet Σ we denote by Σ^* the set of finite words over Σ. For $w \in \Sigma^*$ we denote its letters by $w = w_1 \cdots w_n$, and its length by $|w| = n$. In the following, when discussing sets of numbers, we define $\min \emptyset = \infty$

Automata A *nondeterministic finite automaton (*NFA*)* is a 5-tuple $\mathscr{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ 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 $\alpha \subseteq Q$ is a set of accepting states. A *run* of $\mathscr{A}$ on a word $w = w_1 w_2 \ldots w_n$ is a sequence $\rho = q_0, q_1, \ldots, q_n$ such that $q_0 \in Q_0$ and for every $0 \leq i < n$ it holds that $q_{i+1} \in \delta(q_i, w_{i+1})$. The run ρ is *accepting* if $q_n \in \alpha$. A word w is *accepted* by $\mathscr{A}$ if there exists an accepting run of $\mathscr{A}$ on w. The *language* of $\mathscr{A}$, denoted $\mathfrak{L}(\mathscr{A})$, is the set of words accepted by $\mathscr{A}$.

Permutations Let $n \in \mathbb{N}$. The *permutation group* S_n is the set of bijections (i.e. *permutations*) from $\{1,...,n\}$ to itself. S_n forms a group with the function-composition operation and the identity permutation as a neutral element. Given a word $w = w_1 \cdots w_n$ and a permutation $\pi \in S_n$, we define $\pi(w) = w_{\pi(1)} \cdots w_{\pi(n)}$. For example, if $w = abcd$ and $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$ then $\pi(w) = cdba$. We usually denote permutations in *one-line form*, e.g., π is represented as $(3,4,2,1)$. We say that a word y is a *permutation* of x, and we write $x \sim y$ if there exists a permutation $\pi \in S_{|x|}$ such that $\pi(x) = y$.

Jumping Automata A jumping automaton is syntactically identical to an NFA, with the semantic difference that it has a reading head that can "jump" between indices of the input word. An equivalent view is that a jumping automaton reads a (nondeterministically chosen) permutation of the input word.

Formally, consider an NFA $\mathscr{A}$. We view $\mathscr{A}$ as a *jumping finite automaton* (JFA) by defining its *jumping language* $\mathfrak{J}(\mathscr{A}) = \{w \in \Sigma^* \mid \exists u \in \Sigma^*.\ w \sim u \wedge u \in \mathfrak{L}(\mathscr{A})\}$.

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 a JFA $\mathscr{A}$. 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 $\mathbf{a}$ 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 of 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)* [16, 6]. 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 $(\mathscr{A}, \mathscr{C})$ where $\mathscr{A}$ is an NFA over alphabet Σ, and $\mathscr{C}$ is a JFA over Σ. Then, the PA $(\mathscr{A}, \mathscr{C})$ accepts a word w if $w \in \mathfrak{L}(\mathscr{A}) \cap \mathfrak{J}(\mathscr{C})$. Note that when $\mathfrak{L}(\mathscr{A}) = \Sigma^*$, then the PA coincides with $\mathfrak{J}(\mathscr{C})$. Our usage of PA is to obtain the decidability of certain problems. Specifically, from [16] we have that emptiness of PA is decidable.

3 Quantitative Semantics for JFAs

In this section we present and demonstrate the three quantitative semantics for JFAs. We then define the relevant decision problems, and lay down some general outlines to solving them, which are used in later sections. For the remainder of the section fix a JFA $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$.

3.1 The Semantics

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 jumps 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 letters as distance $k-1$ (either to the left or to the right). This is captured as follows.

For $k \in \mathbb{Z}$, define $[\![k]\!] = |k| - 1$. Consider a word $w \in \Sigma^*$ with $|w| = n$, 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 $[\![\mathbf{a}]\!] = \sum_{i=1}^{n+1} [\![a_i - a_{i-1}]\!]$.

Definition 3 (Absolute-Distance Semantics)**.** *For a word $w \in \Sigma^*$ with $|w| = n$ we define*

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

(recall that $\min \emptyset = \infty$ *by definition).*

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 $|w| = n$, 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 4 (Reversal Semantics)**.** *For a word $w \in \Sigma^*$ with $|w| = n$ we define*

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

The Hamming Semantics In the Hamming measure, the cost of a word is the minimal number of coordinates of w 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.

Consider two words $x, y \in \Sigma^*$ with $|x| = |y| = n$ such that $x \sim y$, we define the *Hamming Distance* between x and y as $d_H(x, y) = |\{i \mid x_i \neq y_i\}|$.

Definition 5 (Hamming Semantics)**.** *For a word $w \in \Sigma^*$ we define*

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

Remark 6. *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.*

3.2 Quantitative Decision Problems

In the remainder of the paper we focus on quantitative variants of the standard Boolean decision problems pertaining to the jumping semantics. Specifically, we consider the following problems for each semantics $\text{SEM} \in \{\text{ABS}, \text{HAM}, \text{REV}\}$.

- MEMBERSHIP: Given a JFA $\mathscr{A}$, $k \in \mathbb{N}$ and a word w, decide whether $\mathscr{A}_{\text{SEM}}(w) \leq k$.
- k-BND (for a fixed k): Given a JFA $\mathscr{A}$, decide whether $\forall w \in \mathfrak{J}(\mathscr{A})\ \mathscr{A}_{\text{SEM}}(w) \leq k$.
- PARAM-BND: Given a JFA $\mathscr{A}$ and $k \in \mathbb{N}$, decide whether $\forall w \in \mathfrak{J}(\mathscr{A})\ \mathscr{A}_{\text{SEM}}(w) \leq k$.

We also pay special attention to the setting where $\mathfrak{J}(\mathscr{A}) = \Sigma^*$, in which case we refer to these problems as UNIV-k-BND and UNIV-PARAM-BND. For example, in UNIV-PARAM-BND we are given a JFA $\mathscr{A}$ and $k \in \mathbb{N}$ and the problem is to decide whether $\mathscr{A}_{\text{SEM}}(w) \leq k$ 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 effect the complexity. We remark on this when it is relevant. Note that the emptiness problem is absent from the list above. Indeed, a natural quantitative variant would be: is there a word w such that $\mathscr{A}_{\text{SEM}}(w) \leq k$. This, however, is identical to Boolean emptiness, since $\mathfrak{L}(\mathscr{A}) \neq \emptyset$ if and only if there exists w such that $\mathscr{A}_{\text{SEM}}(w) = 0$. We therefore do not consider this problem. Another problem to consider is boundedness when k is existentially quantified. We elaborate on this problem in Section 8.

4 The Absolute-Distance Semantics

The first semantics we investigate is ABS, and we start by showing that (the decision version of) computing its value for a given word is NP-complete. This is based on bounding the distance with which a word can be accepted.

Lemma 7. *Consider a JFA $\mathscr{A}$ and $w \in \mathfrak{J}(\mathscr{A})$ with $|w| = n$, then $\mathscr{A}_{\text{ABS}}(w) < n^2$.*

Proof. Since $w \in \mathfrak{J}(\mathscr{A})$, there exists a jump sequence $\mathbf{a} = (a_0, a_1, a_2, \ldots, a_n, a_{n+1})$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})$. Therefore, $\mathscr{A}_{\text{ABS}}(w) \leq [\![\mathbf{a}]\!]$. 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 $a_0 = 0$ and $a_n = n+1$). 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 computing the absolute distance, as follows.

Theorem 8 (Absolute-Distance MEMBERSHIP is NP-complete)**.** *The problem of deciding, given $\mathscr{A}$, w and $k \in \mathbb{N}$, whether $\mathscr{A}_{\text{ABS}}(w) \leq k$, is NP-complete.*

Proof. In order to establish membership in NP, note that by Lemma 7, we can assume $k \leq n^2$, as otherwise we can set $k = n^2$. Then, it is sufficient to nondeterministically guess a jump sequence $\mathbf{a}$ and to check that $w_{a_1} \cdots w_{a_n} \in \mathfrak{L}(\mathscr{A})$ 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 [12] that deciding whether $w \in \mathfrak{J}(\mathscr{A})$ is NP-hard. We reduce this problem by outputting, given $\mathscr{A}$ and w, the same $\mathscr{A}$ and w with the bound $k = n^2$. The reduction is correct by Lemma 7 and the fact that if $w \notin \mathfrak{J}(\mathscr{A})$ then $\mathscr{A}_{\text{ABS}}(w) = \infty$. □

4.1 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 an 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 $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$.

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 $\mathbf{a} = (a_0, a_1, a_2, \dots, a_n, a_{n+1})$. Thus, the jumping head points to the letter at index a_j. Concurrently, imagine a "sequential" head (reading from left to right), which points to the j^{th} letter in w. Note that these two heads start and finish reading the word at the same indices $a_0 = 0$ and $a_{n+1} = n+1$. 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 the heads. The following lemma formalizes this idea.

Lemma 9. *Consider a jump sequence* $\mathbf{a} = (a_0, a_1, a_2, \dots, a_n, a_{n+1})$. *For every* $1 \le j \le n$ *it holds that* $[\![\mathbf{a}]\!] \ge |a_j - j|$.

Proof. Let $1 \le j \le n+1$. First, assume that $a_j \ge j$ and consider the sum $\sum_{i=1}^{j} [\![a_i - a_{i-1}]\!] \le [\![\mathbf{a}]\!]$. From the definition of $[\![\cdot]\!]$ we have $\sum_{i=1}^{j} [\![a_i - a_{i-1}]\!] = \left(\sum_{i=1}^{j} |a_i - a_{i-1}|\right) - j$, and we conclude that in this case $[\![\mathbf{a}]\!] \ge |a_j - j|$ by the following:

$$\left(\sum_{i=1}^{j} |a_i - a_{i-1}|\right) - j \underset{\text{triangle inequality}}{\ge} \left|\sum_{i=1}^{j} a_i - a_{i-1}\right| - j \underset{\text{telescopic sum}}{=} |a_j - a_0| - j \underset{a_0=0}{=} a_j - j \underset{a_j \ge j}{=} |a_j - j|$$

The direction $a_j < j$ is proved by looking at the sum of the *last* j elements: assume $a_j < j$, and consider the sum $\sum_{i=j+1}^{n+1} [\![a_i - a_{i-1}]\!] \le [\![\mathbf{a}]\!]$. From the definition of $[\![\cdot]\!]$ we have

$$\sum_{i=j+1}^{n+1} [\![a_i - a_{i-1}]\!] = \left(\sum_{i=j+1}^{n+1} |a_i - a_{i-1}|\right) - (n+1-(j+1)+1) = \left(\sum_{i=j+1}^{n+1} |a_i - a_{i-1}|\right) - (n+1-j)$$

Similarly to the previous case, from the triangle inequality we have

$$\left(\sum_{i=j+1}^{n+1} |a_i - a_{i-1}|\right) - (n+1-j) \ge |a_{n+1} - a_j| - (n+1-j) = n+1-a_j-(n+1-j) = j - a_j = |a_j - j|$$

where we use the fact that $a_{n+1} = n+1 > a_j$, and our assumption that $a_j < j$. This again concludes that $[\![\mathbf{a}]\!] \ge |a_j - j|$. □

From Lemma 9 we get that in order for a word w to attain a small cost, it must be accepted with a jumping sequence that stays close to the sequential head. More precisely:

Corollary 10. *Let* $k \in \mathbb{N}$ *and consider a word* w *such that* $\mathcal{A}_{\text{ABS}}(w) \le k$, *then there exists a jumping sequence* $\mathbf{a} = (a_0, a_1, a_2, \dots, a_n, a_{n+1})$ *such that* $w_{\mathbf{a}} \in \mathfrak{L}(\mathcal{A})$ *and for all* $1 \le j \le n$ *we have* $|a_j - j| \le k$.

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

Lemma 11. *Let $k \in \mathbb{N}$. We can effectively construct an* NFA $\mathscr{B}$ *such that* $\mathfrak{L}(\mathscr{B}) = \{w \in \Sigma^* | \mathscr{A}_{\text{ABS}}(w) \leq k\}$.

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

$\mathscr{B}$ then nondeterministically simulates a jumping sequence of $\mathscr{A}$ 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. Similarly, if the word ends but the guessed run tried to read a letter beyond the length of the word, the run rejects. The correctness of the construction follows from Corollary 10. We now turn to the formal details. Recall that $\mathscr{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$. We define $\mathscr{B} = \langle \Sigma, Q', \delta', Q'_0, \beta \rangle$ as follows.

The state space of $\mathscr{B}$ is $Q' = Q \times (\Sigma \times \{?, \checkmark\})^{-k,\dots,k} \times \{-k, \dots, k\} \times \{0, \dots, k\}$. We denote a state of $\mathscr{B}$ as (q, f, j, c) where $q \in Q$ is a state of $\mathscr{A}$, $f : \{-k, \dots, k\} \to \Sigma \times \{?, \checkmark\}$ represents a window of size $2k+1$ around the sequential head, where $\checkmark$ marks letters that have already been read by $\mathscr{A}$ (and ? marks the others), j represents the index of the head of $\mathscr{A}$ 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 $\mathscr{B}$ 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 $\mathscr{A}$ is initial, the location of the jumping head is some $j > 0$ incurring a cost of $j-1$ (i.e., the initial jump $\mathscr{A}$ makes), and the window is guessed so that everything left of the first letter is marked as already-read (to simulate the fact that $\mathscr{A}$ cannot jump to the left of the first letter).

The transitions of $\mathscr{B}$ are defined as follows. Consider a state (q, f, j, c) and a letter $\sigma \in \Sigma$, then $(q', f', j', c') \in \delta'((q, f, j, c), \sigma)$ if and only if the following hold (see Fig. 1 for an illustration):

- $f(1)_1 = \sigma$. That is, we verify that the next letter in the guessed window is indeed correct.
- $f(-k)_2 = \checkmark$. That is, the leftmost letter has been read. Otherwise by Corollary 10 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 read from now on (note that index j before the transition corresponds to index $j-1$ after).
- $q' = \delta(q, f(j)_1)$, i.e. the state of $\mathscr{A}$ 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 Fig. 1. Also, $c' \leq k$ 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[2].

Finally, the accepting states of $\mathscr{B}$ are $\beta = \{(q, f, 1, c) \mid q \in \alpha \wedge f(j)_2 = ? \text{ for all } j > 0\}$. That is, the state of $\mathscr{A}$ 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.

[2]The guess could potentially be $\checkmark$, but this is clearly useless.

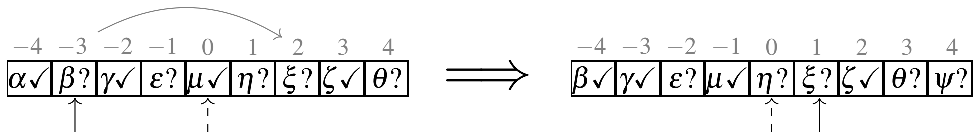

Figure 1: A single transition in the construction of Lemma 11. The dashed arrow signifies the sequential head, the full arrow is the "imaginary" jumping head. Here, the head jumps from -3 to 2, incurring a cost of 4, but in the indexing after the transition ξ is at index 1, thus the expression given 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.

It is easy to verify that $\mathscr{B}$ indeed guesses a jump sequence and a corresponding run of $\mathscr{A}$ on the given word, provided that the jumping head stays within distance k of the sequential head. By Corollary 10, this restriction is complete, in the sense that if $\mathscr{A}_{\text{ABS}}(w) \leq k$ then there is a suitable jump sequence under this restriction with which w is accepted. ☐

We can now readily conclude the decidability of the boundedness problems for the ABS semantics. The proof (in the full version) makes use of the decidability of emptiness for Parikh Automata [16].

Theorem 12. *The following problems are decidable for the* ABS *semantics:* k-BND, PARAM-BND, UNIV-k-BND *and* UNIV-PARAM-BND.

With further scrutiny, we see that the size of $\mathscr{B}$ constructed as per Lemma 11 is polynomial in the size of $\mathscr{A}$ and single-exponential in k. Thus, UNIV-k-BND is in fact decidable in PSPACE, whereas UNIV-PARAM-BND is in EXPSPACE and 2-EXPSPACE for k given in unary and binary, respectively. For the non-universal problems we do not supply upper complexity bounds, as these depend on the decidability for PA containment, for which we only derive decidability from [16].

4.2 PSPACE-Hardness of Boundedness for ABS

In the following, we complement the decidability result of Theorem 12 by showing that already UNIV-k-BND is PSPACE-hard, for every $k \in \mathbb{N}$.

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

Lemma 13. *Consider a jumping 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 that 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

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

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

We say that $\mathscr{A}_{\text{ABS}}$ is *k-bounded* if $\mathscr{A}_{\text{ABS}}(w) \leq k$ for all $w \in \Sigma^*$. We are now ready to prove the hardness of UNIV-k-BND. Observe that for a word $w \in \Sigma^*$ we have that $\mathscr{A}_{\text{ABS}}(w) = 0$ if and only if $w \in \mathfrak{L}(\mathscr{A})$ (indeed, a cost of 0 implies that an accepting jump sequence is the sequential run $0, 1, \ldots, |w|+1$). In particular, we have that $\mathscr{A}_{\text{ABS}}$ is 0-bounded if and only if $\mathfrak{L}(\mathscr{A}) = \Sigma^*$. Since the universality problem for NFAs is PSPACE-complete, this readily proves that UNIV-0-BND is PSPACE-hard. Note, however, that this does *not* imply that UNIV-k-BND 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 14. *For* ABS *the* UNIV-k-BND *and* k-BND *problems are PSPACE-hard for every* $k \in \mathbb{N}$.

Proof. We sketch the proof for UNIV-k-BND. The case of k-BND requires slightly more effort and is delegated to the full version. By Lemma 13, we can assume without loss of generality that k is even. Indeed, if there exists $m \in \mathbb{N}$ such that $\mathscr{A}_{\text{ABS}}(w) \leq 2m+1$ for every $w \in \Sigma^*$, then by Lemma 13 we also have $\mathscr{A}_{\text{ABS}}(w) \leq 2m$. Therefore, we assume $k = 2m$ for some $m \in \mathbb{N}$.

We reduce the universality problem for NFAs to the UNIV-$2m$-BND problem. Consider an NFA $\mathscr{A} = \langle Q, \Sigma, \delta, Q_0, \alpha \rangle$, and let $\heartsuit \notin \Sigma$ be a fresh symbol. Intuitively, we obtain from $\mathscr{A}$ an NFA $\mathscr{B}$ over the alphabet $\Sigma \cup \{\heartsuit\}$ such that $w \in \mathfrak{L}(\mathscr{B})$ if and only if the following hold:

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

We then have the following: if $\mathfrak{L}(\mathscr{A}) = \Sigma^*$, then for every $w \in (\Sigma \cup \{\heartsuit\})^*$ if $w \in \mathfrak{L}(\mathscr{B})$ then $\mathscr{B}_{\text{ABS}}(w) = 0 \leq 2m$, and if $w \notin \mathfrak{L}(\mathscr{B})$ then w starts with $\heartsuit$ but has exactly m occurrences of $\heartsuit$. Thus, jumping to the first occurrence of a letter in Σ incurs a cost of at most m, and reading the skipped $\heartsuit$ symbols raises the cost to at most $2m$. From there, w can be read consecutively and be accepted since $w|_\Sigma \in \mathfrak{L}(\mathscr{A})$. So again $\mathscr{B}_{\text{ABS}}(w) \leq 2m$, and $\mathscr{B}$ is $2m$-bounded.

Conversely, if $\mathfrak{L}(\mathscr{A}) \neq \Sigma^*$, take $x \notin \mathfrak{L}(\mathscr{A})$ such that $x \neq \varepsilon$ (see the full version for details regarding this assumption), and consider the word $w = \heartsuit^m x$. We then have $w \notin \mathfrak{L}(\mathscr{B})$, and moreover – in order to accept w (if at all possible), $\mathscr{B}$ first needs to jump over the initial $\heartsuit^m$, guaranteeing a cost of at least $2m$ (m for the jump and another m to later read the $\heartsuit^m$ prefix), and needs at least one more jump to accept x, since $x \notin \mathfrak{L}(\mathscr{A})$. Thus, $\mathscr{B}_{\text{ABS}}(w) > 2m$, so $\mathscr{B}$ is not $2m$-bounded. The precise construction and correctness are given in the full version. ☐

Lemma 14 shows hardness for fixed k, and in particular when k is part of the input. Thus, UNIV-PARAM-BND and PARAM-BND are also PSPACE-hard, and UNIV-k-BND is PSPACE-complete. Also, UNIV-PARAM-BND is in EXPSPACE and 2-EXPSPACE for k given in unary and binary, respectively.

5 The Reversal Semantics

We now study the reversal semantics. Recall from Definition 4 that for a JFA $\mathscr{A}$ and a word w, the cost $\mathscr{A}_{\text{REV}}(w)$ 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 $\mathbf{a} = (a_0, a_1, a_2, \ldots, a_n, a_{n+1})$. 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 $\mathbf{a}$.

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 $\mathscr{A}_{\text{REV}}(w) = \min\{|\texttt{Turn}(\mathbf{a})| \mid w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})\}$. Viewed in this manner, we have that $\mathscr{A}_{\text{REV}}(w) \leq |w|$, and computing $\texttt{Turn}(\mathbf{a})$ can be done in polynomial time. Thus, analogously to Theorem 8 we have the following.

Theorem 15 (Reversal MEMBERSHIP is NP-complete)**.** *The problem of deciding, given $\mathscr{A}$ and k, whether $\mathscr{A}_{\text{REV}}(w) \leq k$ is NP-complete.*

Remark 16. *For every jump sequence* **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* $n+1$.

5.1 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. 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:

$$x \sqcup\!\sqcup y = \{s_1 \cdot t_1 \cdot s_2 \cdot t_2 \cdots s_k \cdot t_k \mid \forall i\ s_i, t_i \in \Sigma^* \wedge x = s_1 \cdots s_k \wedge y = t_1 \cdots t_k\}$$

For example, 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 an associative operation, so it can be extended to any finite number of words.

The following lemma states that, intuitively, if $\mathscr{A}_{\text{REV}}(w) \leq k$, 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). See the full version for the proof.

Lemma 17. *Let $k \in \mathbb{N}$. Consider an* NFA $\mathscr{A}$ *and a word $w \in \Sigma^*$. Then $\mathscr{A}_{\text{REV}}(w) \leq k$ if and only if there exist words $s_1, s_2, \ldots, s_{k+1} \in \Sigma^*$ such that the following hold.*

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

Using the characterization in Lemma 17, we can now construct a corresponding NFA, by intuitively guessing the shuffle decomposition and running copies of $\mathscr{A}$ and its reverse in parallel. See the full version for the proof.

Lemma 18. *Let $k \in \mathbb{N}$ and consider a* JFA $\mathscr{A}$*. We can effectively construct an* NFA $\mathscr{B}$ *such that* $\mathfrak{L}(\mathscr{B}) = \{w \in \Sigma^* \mid \mathscr{A}_{\text{REV}}(w) \leq k\}$.

The proof of Lemma 18 shows that the size of $\mathscr{B}$ is polynomial in the size of $\mathscr{A}$ and single-exponential in k, giving us PSPACE membership for UNIV-k-BND.

5.2 PSPACE-Hardness of Boundedness for REV

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

Theorem 19. *For* REV, UNIV-k-BND *is PSPACE-complete for every $k \in \mathbb{N}$.*

Proof. Membership in PSPACE follows from Lemma 18 and the discussion thereafter. For hardness, we follow the same flow as the proof of Lemma 14, but naturally the reduction itself is different. Specifically, we construct an NFA that must read an expression of the form $(\heartsuit\spadesuit)^m$ before its input. This allows us to shuffle the input to the form $\spadesuit^m\heartsuit^m$, which causes many reversals (see the full version). □

As in Section 4.2, it follows that UNIV-PARAM-BND, k-BND and PARAM-BND are also PSPACE-hard.

6 The Hamming Semantics

Recall from Definition 5 that for a JFA $\mathscr{A}$ and word w, the cost $\mathscr{A}_{\text{HAM}}(w)$ is the minimal Hamming distance between w and w' where $w' \sim w$ and $w' \in \mathfrak{L}(\mathscr{A})$.

Remark 20 (An alternative interpretation of the Hamming Semantics)**.** *We can think of a jumping automaton as accepting a permutation w' of the input word w. As such, a natural candidate for a quantitative measure is the "distance" of the permutation used to obtain w' from the identity (i.e. from w). The standard definition for such a distance is the number of transpositions of two indices required to move from one permutation to the other, namely the distance in the Cayley graph for the transpositions generators of S_n. It is easy to show that in fact, the Hamming distance coincides with this definition.*

In the full version we establish the complexity of computing the Hamming measure of a given word.

Theorem 21 (Hamming MEMBERSHIP is NP-complete)**.** *The problem of deciding, given $\mathscr{A}$ and $k \in \mathbb{N}$, whether $\mathscr{A}_{\text{HAM}}(w) \leq k$ is NP-complete.*

Similarly to Sections 4.1 and 5.1, in order to establish the decidability of UNIV-PARAM-BND, we start by constructing an NFA that accepts exactly the words for which $\mathscr{A}_{\text{HAM}}(w) \leq k$.

Lemma 22. *Let $k \in \mathbb{N}$. We can effectively construct an* NFA $\mathscr{B}$ *with* $\mathfrak{L}(\mathscr{B}) = \{w \in \Sigma^* \mid \mathscr{A}_{\text{HAM}}(w) \leq k\}$.

Proof. Let $k \in \mathbb{N}$. Intuitively, $\mathscr{B}$ works as follows: while reading a word w sequentially, it simulates the run of $\mathscr{A}$, but allows $\mathscr{A}$ to intuitively "swap" the current letter with a (nondeterministically chosen) different one (e.g., the current letter may be a but the run of $\mathscr{A}$ can be simulated on either a or b). Then, $\mathscr{B}$ 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. $\mathscr{B}$ 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. See the full version for a detailed construction and proof. ☐

An analogous proof to Theorem 12 gives us the following.

Theorem 23. *The following problems are decidable for the* HAM *semantics:* k-BND, PARAM-BND, UNIV-k-BND, *and* UNIV-PARAM-BND.

We note that the size of $\mathscr{B}$ constructed in Lemma 22 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-PARAM-BND and UNIV-k-BND are in PSPACE.

For a lower bound, we remark that similarly to Section 4.2, it is not hard to prove that UNIV-k-BND is also PSPACE-hard for every k, using relatively similar tricks. However, since UNIV-PARAM-BND is already PSPACE-complete, then UNIV-k-BND is somewhat redundant. We therefore make do with the trivial lower bound whereby we reduce universality of NFA to UNIV-0-BND.

Theorem 24. *For* HAM*, the* UNIV-PARAM-BND *problem is PSPACE-complete for k encoded in unary and fixed alphabet Σ.*

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 $\mathscr{A}$, if $\mathscr{A}_{\text{ABS}}$ is bounded, then so is $\mathscr{A}_{\text{HAM}}$, and if $\mathscr{A}_{\text{HAM}}$ is bounded, then so is $\mathscr{A}_{\text{REV}}$. We complete the picture by showing that these are the only relationships – we give examples for the remaining cases (see Table 2).

Lemma 25. *Consider a* JFA $\mathscr{A}$. *If* $\mathscr{A}_{\text{ABS}}$ *is bounded, then* $\mathscr{A}_{\text{HAM}}$ *is bounded.*

Proof. Consider a word $w \in \Sigma^*$, we show that if $\mathscr{A}_{\text{ABS}}(w) \leq k$ for some $k \in \mathbb{N}$ then $\mathscr{A}_{\text{HAM}}(w) \leq (2k+1)(k+1)$. Assume $\mathscr{A}_{\text{ABS}}(w) \leq k$, then there exists a jump sequence $\mathbf{a} = (a_0, \ldots, a_{n+1})$ such that $[\![\mathbf{a}]\!] \leq k$ and $w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})$. 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 9, 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, $\mathscr{A}$ 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 $a_i < i$, then 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 therefor 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 $\mathscr{A}_{\text{HAM}}(w) = |\{i \mid w_{a_i} \neq w_i\}| \leq |\{i \mid a_i \neq i\}| \leq (2k+1)(k+1)$ □

Lemma 26. *Consider a* JFA $\mathscr{A}$. *If* $\mathscr{A}_{\text{HAM}}$ *is bounded, then* $\mathscr{A}_{\text{REV}}$ *is bounded.*

Proof. Consider a word $w \in \Sigma^*$, we show that if $\mathscr{A}_{\text{HAM}}(w) \leq k$ for some $k \in \mathbb{N}$ then $\mathscr{A}_{\text{REV}}(w) \leq 3k$. Assume $\mathscr{A}_{\text{HAM}}(w) \leq k$, then there exists a jump sequence $\mathbf{a} = (a_0, \ldots, a_{n+1})$ such that $w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})$ 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 $a_i = i$ (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 $\mathbf{a}' = (a'_0, \ldots, a'_{n+1})$ obtained from $\mathbf{a}$ by composing it with 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 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[3] in the latter set). □

Combining Lemmas 25 and 26, we have the following.

Corollary 27. *If* ABS *is bounded, then so is* REV.

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 6). The examples are summarized in Table 2, and are proved 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 2: Examples for every possible combination of bounded/unbounded semantics. The languages are given by regular expressions (e.g., $(a+b)^*a$ is the languages of words that end with a.)

Example 28. *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 29. *The language $(a+b)^*a$ is bounded in the* HAM *and* REV *semantics, but unbounded in* ABS. *Indeed, let $\mathscr{A}$ be an* NFA *such that $\mathfrak{L}(\mathscr{A}) = (a+b)^*a$ and consider a word $w \in \mathfrak{J}(\mathscr{A})$, then w has at least one occurrence of a at some index i. Then, for the jumping sequence $\mathbf{a} = (0,1,2,\dots,i-1,n,i+1,\dots n-1,i,n+1)$ we have that $w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})$. 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 $\mathscr{A}_{\text{HAM}}(w) \leq 2$ and $\mathscr{A}_{\text{REV}}(w) \leq 2$.*

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 9, it follows that $\mathscr{A}_{\text{ABS}}(w) \geq n-1$, and by increasing n, we have that $\mathscr{A}_{\text{ABS}}$ is unbounded.*

Example 30. *The language a^*b^* is bounded in the* REV *semantics, but unbounded in* HAM *and* ABS. *Indeed, let $\mathscr{A}$ be an* NFA *such that $\mathfrak{L}(\mathscr{A}) = a^*b^*$ and consider a word $w \in \mathfrak{J}(\mathscr{A})$, and denote by $i_1 < i_2 \dots < i_k$ the indices of a's in w in increasing order, and by $j_1 > j_2 > \dots > j_{n-k}$ the indices of b's in decreasing order. Then, for the jumping sequence $\mathbf{a} = (i_1,\dots,i_k,j_1,\dots j_{n-k},n+1)$ we have that $w_{\mathbf{a}} \in \mathfrak{L}(\mathscr{A})$, and $\mathscr{A}_{\text{REV}}(w) \leq 2$ (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 $\mathfrak{L}(\mathscr{A})$ is $w' = a^n b^n$, and $d_H(w,w') = n$, so $\mathscr{A}_{\text{HAM}}$ is unbounded. By Lemma 25 it follows that $\mathscr{A}_{\text{ABS}}$ is also unbounded.*

[3] A slightly finer analysis shows that this is in fact at most $2k$, but we are only concerned with boundedness.

Example 31. *The language* $(ab)^*$ *is unbounded in all the semantics. Indeed, let* $\mathscr{A}$ *be an* NFA *such that* $\mathfrak{L}(\mathscr{A}) = (ab)^*$*, then by Lemma 26 and Corollary 27 it suffices to show that* $\mathscr{A}_{\text{REV}}$ *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* $i \leq 2n$ *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* $\mathscr{A}_{\text{REV}}(w) = 2n$*, and* $\mathscr{A}_{\text{REV}}$ *is unbounded.*

8 Discussion and Future Work

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

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 $\mathscr{A}_{\text{SEM}}$ 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-BND 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 classes. For example, in the HAM semantics, using a parametric k we can construct a labelled VASS. But the latter do not admit decidable properties for the corresponding boundedness problem.

We remark on one fragment that can be shown to be decidable: consider a setting where the jumps are restricted to swapping disjoint pairs of adjacent letters, each incurring a cost of 1. Then, the JFA can be translated to a weighted automaton, whose boundedness problem is decidable by [15, 18]. We remark that the latter decidability is a very involved result. This suggests (but by no means proves) that boundedness may be a difficult problem.

References

[1] Antonio Abu Nassar & Shaull Almagor (2022): *Simulation by Rounds of Letter-To-Letter Transducers*. In: *30th EACSL Annual Conference on Computer Science Logic*, doi:`10.4230/LIPIcs.CSL.2022.3`.

[2] Shaull Almagor (2020): *Process Symmetry in Probabilistic Transducers*. In: *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, doi:`10.4230/LIPIcs.FSTTCS.2020.35`.

[3] Shaull Almagor & Orna Kupferman (2011): *Max and sum semantics for alternating weighted automata*. In: *International Symposium on Automated Technology for Verification and Analysis*, Springer, pp. 13–27, doi:`10.1007/978-3-642-24372-1_2`.

[4] Shaull Almagor & Omer Yizhaq (2023): *Jumping Automata over Infinite Words*. In: *International Conference on Developments in Language Theory*, Springer, pp. 9–22, doi:`10.1007/978-3-031-33264-7_2`.

[5] Udi Boker (2021): *Quantitative vs. weighted automata*. In: *Reachability Problems: 15th International Conference, RP 2021, Liverpool, UK, October 25–27, 2021, Proceedings 15*, Springer, pp. 3–18, doi:10.1007/978-3-030-89716-1_1.

[6] Michaël Cadilhac, Alain Finkel & Pierre McKenzie (2012): *Affine Parikh automata*. *RAIRO-Theoretical Informatics and Applications* 46(4), pp. 511–545, doi:10.1051/ita/2012013.

[7] Michaël Cadilhac, Alain Finkel & Pierre McKenzie (2012): *Bounded parikh automata*. *International Journal of Foundations of Computer Science* 23(08), pp. 1691–1709, doi:10.1142/S0129054112400709.

[8] Krishnendu Chatterjee, Laurent Doyen & Thomas A Henzinger (2010): *Quantitative languages*. *ACM Transactions on Computational Logic (TOCL)* 11(4), pp. 1–38, doi:10.1007/978-3-540-87531-4_28.

[9] Manfred Droste, Werner Kuich & Heiko Vogler (2009): *Handbook of weighted automata*. Springer Science & Business Media, doi:10.1007/978-3-642-01492-5.

[10] Szilárd Zsolt Fazekas, Kaito Hoshi & Akihiro Yamamura (2021): *Two-way deterministic automata with jumping mode*. *Theoretical Computer Science* 864, pp. 92–102, doi:10.1016/j.tcs.2021.02.030.

[11] Henning Fernau, Meenakshi Paramasivan & Markus L Schmid (2015): *Jumping finite automata: characterizations and complexity*. In: *International Conference on Implementation and Application of Automata*, Springer, pp. 89–101, doi:10.1007/978-3-319-22360-5_8.

[12] Henning Fernau, Meenakshi Paramasivan, Markus L Schmid & Vojtěch Vorel (2017): *Characterization and complexity results on jumping finite automata*. *Theoretical Computer Science* 679, pp. 31–52, doi:10.1016/j.tcs.2016.07.006.

[13] Dana Fisman, Joshua Grogin & Gera Weiss (2023): *A Normalized Edit Distance on Infinite Words*. In: *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, Schloss-Dagstuhl-Leibniz Zentrum für Informatik, doi:10.4230/LIPIcs.CSL.2023.20.

[14] Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen & Martin Zimmermann (2022): *Parikh Automata over Infinite Words*. In: *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, doi:10.4230/LIPIcs.FSTTCS.2022.40.

[15] Kosaburo Hashiguchi (1982): *Limitedness theorem on finite automata with distance functions*. *Journal of computer and system sciences* 24(2), pp. 233–244, doi:10.1016/0022-0000(82)90051-4.

[16] Felix Klaedtke & Harald Rueß (2003): *Monadic second-order logics with cardinalities*. In: *Automata, Languages and Programming: 30th International Colloquium, ICALP 2003 Eindhoven, The Netherlands, June 30–July 4, 2003 Proceedings 30*, Springer, pp. 681–696, doi:10.1007/3-540-45061-0_54.

[17] Giovanna J Lavado, Giovanni Pighizzini & Shinnosuke Seki (2014): *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*, Springer, pp. 294–305, doi:10.1007/978-3-319-09704-6_26.

[18] Hing Leung & Viktor Podolskiy (2004): *The limitedness problem on distance automata: Hashiguchi's method revisited*. *Theoretical Computer Science* 310(1-3), pp. 147–158, doi:10.1016/S0304-3975(03)00377-3.

[19] Alexander Meduna & Petr Zemek (2012): *Jumping finite automata*. *International Journal of Foundations of Computer Science* 23(07), pp. 1555–1578, doi:10.1142/S0129054112500244.

[20] Mehryar Mohri (2002): *Edit-distance of weighted automata*. In: *International Conference on Implementation and Application of Automata*, Springer, pp. 1–23, doi:10.1007/3-540-44977-9_1.

[21] Vojtěch Vorel (2018): *On basic properties of jumping finite automata*. *International Journal of Foundations of Computer Science* 29(01), pp. 1–15, doi:10.1142/S0129054118500016.