Closure Under Reversal of Languages over Infinite Alphabets

Daniel Genkin[1], Michael Kaminski[2](✉), and Liat Peterfreund[2]

[1] Department of Computer and Information Science, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104, USA

[2] Department of Computer Science, Technion – Israel Institute of Technology, 32000 Haifa, Israel

kaminski@cs.technion.ac.il

Abstract. It is shown that languages definable by weak pebble automata are not closed under reversal. For the proof, we establish a kind of periodicity of an automaton's computation over a specific set of words. The periodicity is partly due to the finiteness of the automaton description and partly due to the word's structure. Using such a periodicity we can find a word such that during the automaton's run on it there are two different, yet indistinguishable, configurations. This enables us to remove a part of that word without affecting acceptance. Choosing an appropriate language leads us to the desired result.

Keywords: Infinite alphabets · Weak pebble automata
Closure properties · Reversal

1 Introduction

While automata for words over finite alphabets are well-understood, a broad research activity began very recently on automata for words over infinite alphabets. Note, that for infinite alphabets, states alone are not sufficient, because an automaton should be able to check equality of input symbols. This can be done by (dynamically) marking a set of symbols of a fixed finite cardinality and allowing equality tests with these symbols.

Finite-Memory Automata (FMA) [2,3] keep marked symbols in a finite number of registers and Pebble Automata (PA) [5,6] keep marked symbols under a finite number of pebbles. Both are very restrictive models intended for recognizing an analog of regular languages over finite alphabets. The class of languages recognizable by FMA and PA enjoys many of the properties of regular languages. Languages recognizable by FMA are closed under standard language operations: intersection, union, concatenation, and iteration (Kleene star), whereas languages recognizable by PA are closed under all boolean operations (i.e., union, intersection and complementation), but are not closed under iteration. However, the emptiness problem for FMA is decidable, whereas it is decidable for weak 2-PA only.

This paper deals with weak PA which, as mentioned above, are finite state automata equipped with a finite number k of pebbles, numbered from 1 to k. Each pebble can serve as the head of the automaton or point at a position in the input word. The pebbles are placed on the input word in the stack discipline: the first pebble placed is the last to be lifted. The first pebble placed is pebble 1. One pebble can only point at one position and the most recently placed pebble serves as the head of the automaton. The automaton moves from one state to another depending on the symbol under the head pebble, the equality tests among symbols under the pebbles, and the equality tests among the pebbles' positions.

There are two main variants of PA, weak and strong and we focus on weak PA (wPA). We show that, unlike languages accepted by strong PA, languages accepted by wPA are not closed under reversal. In this paper we deal with two languages - $L_\subseteq$ and $L_\supseteq$. Both languages consist of words of a special form $\boldsymbol{u}\$\boldsymbol{v}$, where \$ is a special symbol (the separator) not occurring in $\boldsymbol{u}$ or $\boldsymbol{v}$ and the symbols in each of these words are pairwise different. In $L_\subseteq$, each symbol occurring in $\boldsymbol{u}$ also occurs in $\boldsymbol{v}$ and, in $L_\supseteq$, each symbol occurring in $\boldsymbol{v}$ also occurs in $\boldsymbol{u}$. Thus, $L_\subseteq$ and $L_\supseteq$ are the reversals of each other. We show that the language $L_\subseteq$ is accepted by wPA, whereas $L_\supseteq$ is not. For this, for each automaton accepting all words of $L_\supseteq$, we construct a special word $\boldsymbol{u}\$\boldsymbol{v} \not\in L_\supseteq$, yet still accepted by the automaton. In that word a prefix of $\boldsymbol{u}$ is "properly" spread in $\boldsymbol{v}$ and its construction is based on a kind of periodicity of the sequence of the states in the run of PA. In particular, the head pebble behaves periodically when it sees symbols different from those under the other pebbles.

The paper is organized as follows. In Sect. 2 we present the definition of wPA and state the separation theorem whose proof, for the case of 2-wPA, is presented in Sect. 3.[1] In Sect. 4 we show how to modify the languages $L_\subseteq$ and $L_\supseteq$ to languages whose words do not contain a distinguished separator symbol. We conclude the paper with a short remark that concerns the technique applied for the proof of the separation theorem.

2 Weak Pebble Automata

As introduced in [5,6], Pebble Automata over infinite alphabets are finite state machines equipped with a finite set of numbered pebbles. The computation of an automaton on an input word starts when the lowest numbered pebble is located at the leftmost position of the input and acts as the head of the automaton. During the computation, an automaton can place (respectively, lift) a pebble on (respectively, from) the input. It can also move the pebble that acts as the head of the automaton. That pebble is the highest numbered pebble present on the input, whereas the other pebbles serve as pointers at the input symbols. The use of the pebbles is restricted by the stack discipline (pebble i can only be placed

[1] The proof of the general case can be found in [7], and, hopefully, will also appear elsewhere.

when pebble $i-1$ is present on the input word and pebble i can only be lifted when pebble $i+1$ is not present on the input word).

A transition depends on the current state, equality type of the symbols under the placed pebbles and equality among the pebbles' positions. The transition relation specifies change of state, the movement of the head and, possibly, whether the head pebble is lifted or a new pebble is placed.

Definition 1. *A deterministic one-way*[2] *k-wPA over an infinite alphabet Σ, is a tuple $\mathfrak{A} = \langle S, s_0, F, T\rangle$ whose components are as follows.*

- *S is a finite set of states,*
- *$s_0 \in S$ is the initial state,*
- *$F \subseteq S$ is a set of accepting states,*
- *T is a finite set of transitions of the form $\alpha \to \beta$, where*
  - *α is of the form (i, σ, P, V, s) or (i, P, V, s), $i \in \{1, \ldots, k\}$, $\sigma \in \Sigma$, $P, V \subseteq \{1, \ldots, i-1\}$, and $s \in S$, and*
  - *β is of the form $(p, \texttt{action})$, where $p \in S$ and $\texttt{action} \in \{\texttt{move}, \texttt{place}, \texttt{lift}\}$,*

  such that $\alpha \to \beta$ and $\alpha \to \beta'$ imply $\beta = \beta'$.

For a word $\boldsymbol{w} \in \Sigma^*$, a configuration of $\mathfrak{A}$ on $\boldsymbol{w}$ is of the form $\gamma = [i, s, \theta]$, where $i \in \{1, \ldots, k\}$, $s \in S$, and $\theta : \{1, \ldots, i\} \to \{1, \ldots, |\boldsymbol{w}|\}$ indicates the pebble's positions on the input word $\boldsymbol{w}$. That is, $\theta(j)$ is the position of pebble j. In what follows, we identify θ with the i-tuple $(\theta(1), \ldots, \theta(i))$. Thus, i can be recovered from θ, but it is convenient to include it into a configuration explicitly.

The *initial configuration* is $\gamma_0 = [1, s_0, (1)]$. That is, the run starts in the initial state s_0 with pebble 1 placed at the beginning of the input word. An *accepting configuration* is of the form $[i, s, \theta]$, where $s \in F$.

Let $\boldsymbol{w} = w_1 \cdots w_n \in \Sigma^+$. A transition $(i, \sigma, P, V, s) \to \beta$ *applies* to a configuration $\gamma = [j, s', \theta]$ if

(1) $i = j$ and $s' = s$,
(2) $P = \{h < i : \theta(h) = \theta(i)\}$,
(3) $V = \{h < i : w_{\theta(h)} = w_{\theta(i)}\}$, and
(4) $w_{\theta(i)} = \sigma$.

In the above definition, P is the set of pebbles placed at the same position as the head pebble, V is the set of pebbles placed above the same symbol as the head pebble, and the current symbol under the head pebble is σ.

A transition $(i, P, V, s) \to \beta$ applies to a configuration $\gamma = [j, s', \theta]$, if the above conditions (1)–(3) are satisfied and no transition of the form $(i, \sigma, P, V, s) \to \beta$ applies to γ.

The transition relation $\vdash_{\boldsymbol{w}}$ on the set of all configurations is defined as follows:[3] $[i, s, \theta] \vdash [i', s', \theta']$ if and only if there is a transition $\alpha \to (p, \texttt{action})$ that applies to $[i, s, \theta]$ such that $s' = p$ and the following holds.

[2] It has been shown in [8] that alternating non-deterministic and deterministic one-way wPA have the same expressive power.

[3] We omit the subscript $\boldsymbol{w}$ of $\vdash$, if it is clear from the context.

- For all $j < i$, $\theta'(j) = \theta(j)$,
- if `action` is `move`, then $i' = i$ and $\theta'(i) = \theta(i) + 1$,
- if `action` is `place`, then $i' = i + 1$ and $\theta'(i + 1) = \theta'(i) = \theta(i)$,[4] and
- if `action` is `lift`, then $i' = i - 1$ and θ' is the restriction of θ on $\{1, \ldots, i-1\}$.

The *language* $L(\mathfrak{A})$ of $\mathfrak{A}$ consists of all words $\boldsymbol{w}$ such that $\gamma_0 \vdash_{\boldsymbol{w}}^{*} \gamma$ for an accepting configuration γ.

Remark 1. Note that the accepted languages are quite symmetric: they contain only finitely many "distinguished" symbols explicitly mentioned in the automaton transitions and are invariant under any permutation of all other symbols of the infinite alphabet Σ, cf. [3, Proposition 2].

Remark 2. It follows from the definition that wPA languages do not contain the empty word ϵ, but the languages we deal with in this paper do not contain ϵ either.

Next, we observe the following. To each configuration $\gamma = [i, s, \theta]$ of a deterministic one-way wPA corresponds the vector $\varphi^{\gamma} = (P_1, \ldots, P_i)$, where

$$P_j = \{h < j : \theta(h) = \theta(j)\}.$$

That is, P_j is the set of pebbles placed before pebble j which are at the same position as pebble j in configuration γ.[5]

If $\gamma \vdash \gamma'$, then $\varphi^{\gamma'}$ can be computed from φ^{γ}, according to the automaton transitions. Namely, if $\varphi^{\gamma} = (P_1, \ldots, P_i)$ and the transition applied to γ is $\alpha \to (p, \texttt{action})$, then

- if `action` is `move`, then $\varphi^{\gamma'} = (P_1, \ldots, P_{i-1}, \emptyset)$,
- if `action` is `lift`, then $\varphi^{\gamma'} = (P_1, \ldots, P_{i-1})$, and
- if `action` is `place`, then $\varphi^{\gamma'} = (P_1, \ldots, P_i, P_i \cup \{i\})$.

We can extend the set of states from S to

$$S \times \bigcup_{i=1}^{k} \{(P_1, \ldots, P_i) : P_j \subseteq \{1, \ldots, j-1\},\ j = 1, \ldots, i\}$$

capturing in such a way the pebbles' positions by the state. This allows us to remove the P component from the left hand side of transitions. That is, we may assume that the left hand side of a transition is of the form (i, σ, V, s) or (i, V, s).

Finally, by adding some extra states and modifying the transitions appropriately, we can normalize the k-wPA behavior such that for each $i \in \{2, \ldots, k\}$ it acts as follows, cf. [6].

[4] That is, pebble $i + 1$ is placed at the position of pebble i, whereas in the strong PA model this pebble is placed at the beginning of the input word, i.e., at the leftmost position.

[5] By definition, $P_1 = \emptyset$ and, therefore, is redundant.

– A pebble is never lifted, but falls down when moving from the right end of the input. Thus, action `lift` is redundant.
– Only pebble 1 can enter a final state and only after it falls down from the right end of the input. In such a case, the accepting configuration consists of the corresponding accepting state only.
– Immediately after pebble i moves without falling down, pebble $i+1$ is placed.
– Immediately after pebble i falls down, pebble $i-1$ moves.

In what follows, we denote the set of letters occurring in a word $\boldsymbol{u}$ by $[\boldsymbol{u}]$. That is, if $\boldsymbol{u} = u_1 \cdots u_n$, then $[\boldsymbol{u}] = \{u_1, \ldots, u_n\}$.

Example 1 (Cf. [4, Example 3.1]). This example deals with the language $L_{\texttt{diff}}$ consisting of all words in which every symbol from Σ occurs at most one time:

$$L_{\texttt{diff}} = \{\sigma_1 \cdots \sigma_n : n \geq 1, \sigma_i \neq \$, \text{ for each } i = 1, \ldots, n, \text{ and } \sigma_i \neq \sigma_j, \text{ whenever } i \neq j\}.$$

This language is accepted by a 2-wPA that acts as follows. Pebble 1 advances through the input from left to right. At each step it verifies that the symbol under it is not \$, and then pebble 2 scans the suffix to the right of the position of pebble 1 to verify that the input symbol under pebble 1 differs from all symbols in that suffix.

Example 2. The language

$$L_{\texttt{diff\$diff}} = \{\boldsymbol{u}\$\boldsymbol{v} : \boldsymbol{u}, \boldsymbol{v} \in L_{\texttt{diff}}\}$$

is accepted by a 2-wPA that first, using the automaton from Example 1 scans $\boldsymbol{u}$ and then, using the same automaton, scans $\boldsymbol{v}$.

Example 3. The language

$$L_{\subseteq} = \{\boldsymbol{u}\$\boldsymbol{v} : \boldsymbol{u}, \boldsymbol{v} \in L_{\texttt{diff}} \text{ and } [\boldsymbol{u}] \subseteq [\boldsymbol{v}]\}$$

is accepted by a 2-wPA that acts as follows. Pebble 1 advances through the input to the separator \$. After each move of pebble 1 on $\boldsymbol{u}$, pebble 2 moves to \$ and then scans the suffix $\boldsymbol{v}$ of the input to find the symbol under pebble 1. Verifying that both $\boldsymbol{u}$ and $\boldsymbol{v}$ are in $L_{\texttt{diff}}$ can be done by the automaton from Example 2.

Theorem 1. *The language*

$$L_{\supseteq} = \{\boldsymbol{u}\$\boldsymbol{v} : \boldsymbol{u}, \boldsymbol{v} \in L_{\texttt{diff}} \text{ and } [\boldsymbol{v}] \subseteq [\boldsymbol{u}]\}$$

is not accepted by wPA.

The proof of Theorem 1 for the case of 2-wPA (see footnote 1) is presented in the next section.

Since $L_{\supseteq}$ is the reversal of $L_{\subseteq}$, by Example 3 and Theorem 1, the languages accepted by wPA are not closed under reversal.

3 Proof of Theorem 1

As we have already mentioned above, the proof is restricted to the case of 2-wPA only. For the proof of the general case, see [7, Sects. 6 and 7].[6]

For the rest of this paper, $\mathfrak{A} = \langle S, s_0, F, T \rangle$ is a 2-wPA and the positions of pebble 1 in runs of $\mathfrak{A}$ will be denoted by p_1, possibly primed.

We construct a word $\boldsymbol{w} \in L_{\supseteq}$ such that the run of $\mathfrak{A}$ on its prefix is periodic. That is, the sequence of states in the run on the prefix is periodic. Using the periodicity we can shrink this prefix without affecting acceptance. It should be emphasized that periodicity alone is not sufficient for deleting a pattern from the input. This is because each move of pebble 1 depends not only on the prefix up to its position, but on the whole input word, see also the note in the end of this section.

We start with examining the run of pebble 2.

Proposition 1. *There exists a positive integer ℓ_2 such that for all $\boldsymbol{w} \in \Sigma^+$, $\boldsymbol{w} = w_1 \cdots w_n$, the following holds. If*

$$[2, s_{j_1}, (p_1, j_1)] \vdash [2, s_{j_1+1}, (p_1, j_1+1)] \vdash \cdots \vdash [2, s_{j_2}, (p_1, j_2)], \tag{1}$$

where $w_j \neq w_{p_1}$ for all $j_1 \leq j \leq j_2$, then the sequence of states $s_{j_1+\ell_2}, \ldots, s_{j_2}$, is periodic with period ℓ_2.[7]

Proof. Let j_1 and j_2 satisfy the prerequisites of the proposition. The transitions applied to the configurations in (1) are of the form $(2, \emptyset, s) \rightarrow (\mathtt{move}, s')$ for some $s, s' \in S$, because the moves of pebble 2 do not depend on w_{p_1}.

Since $\mathfrak{A}$ is deterministic, for some positive integers $m_{s_{j_1}}, \ell_{s_{j_1}} \leq |S|$, after $m_{s_{j_1}}$ steps from w_{j_1}, pebble 2 becomes periodic with a period $\ell_{s_{j_1}}$. Thus, the proposition holds for $\ell_2 = |S|!$, because $m_{s_{j_1}} \leq |S| \leq |S|!$ and $\ell_{s_{j_1}} \leq |S|$ implies that $\ell_{s_{j_1}}$ divides $|S|!$.

Corollary 1. *Let $\boldsymbol{z}' = \boldsymbol{x}\boldsymbol{y}'$ and $\boldsymbol{z}'' = \boldsymbol{x}\boldsymbol{y}''$, $\boldsymbol{x} = x_1 \cdots x_n$, where*

$$[\boldsymbol{x}] \cap ([\boldsymbol{y}'] \cup [\boldsymbol{y}'']) = \emptyset,$$

$$|\boldsymbol{y}'|, |\boldsymbol{y}''| \geq \ell_2,$$

and

$$|\boldsymbol{y}''| \equiv_{\ell_2} |\boldsymbol{y}'|.$$[8]

If

$$[2, s, (p, |\boldsymbol{x}|)] \vdash_{\boldsymbol{z}'} [2, t, (p, |\boldsymbol{x}\boldsymbol{y}'|)], \tag{2}$$

then

$$[2, s, (p, |\boldsymbol{x}|)] \vdash_{\boldsymbol{z}''} [2, t, (p, |\boldsymbol{x}\boldsymbol{y}''|)]. \tag{3}$$

[6] Note that in [7] the pebbles are placed in the reversed order, i.e., the computation start with pebble k and pebble i is placed *after* pebble $i+1$, $i = 1, \ldots, k-1$.

[7] Recall that we identify θ with the tuple of its values and, by the observation in the previous section, we omit the P-component of transitions.

[8] As usual, $\equiv_{\ell_2}$ is the congruence modulo l_2.

Proof. Let $\boldsymbol{y}' = \boldsymbol{y}'_1\boldsymbol{y}'_2$ and $\boldsymbol{y}'' = \boldsymbol{y}''_1\boldsymbol{y}''_2$, where

$$\boldsymbol{y}'_1 = \boldsymbol{y}''_1 = \ell_2. \tag{4}$$

It follows from (2) that for some state s_1

$$[2, s, (p, |\boldsymbol{x}|)] \vdash_{\boldsymbol{z}'} [2, s_1, (p, |\boldsymbol{x}\boldsymbol{y}'_1|)] \tag{5}$$

and

$$[2, s_1, (p, |\boldsymbol{x}\boldsymbol{y}'_1|)] \vdash_{\boldsymbol{z}'} [2, t, (p, |\boldsymbol{x}\boldsymbol{y}'_1\boldsymbol{y}'_2|)] = [2, t, (p, |\boldsymbol{x}\boldsymbol{y}'|)]. \tag{6}$$

It follows from (5) that

$$[2, s, (p, |\boldsymbol{x}|)] \vdash_{\boldsymbol{z}''} [2, s_1, (p, |\boldsymbol{x}\boldsymbol{y}''_1|)], \tag{7}$$

because the moves of the automaton do not depend on x_p – the symbol under pebble 1, and it follows from (6) that

$$[2, s_1, (p, |\boldsymbol{x}\boldsymbol{y}''_1|)] \vdash_{\boldsymbol{z}''} [2, t, (p, |\boldsymbol{x}\boldsymbol{y}''_1\boldsymbol{y}''_2|)] = [2, t, (p, |\boldsymbol{x}\boldsymbol{y}''|)], \tag{8}$$

because, by Proposition 1, (4) implies that the automaton is periodic with period ℓ_2 from state s_1 and

$$|\boldsymbol{y}'_2| = |\boldsymbol{y}'| - |\boldsymbol{y}'_1| \equiv_{\ell_2} |\boldsymbol{y}''| - |\boldsymbol{y}''_1| = |\boldsymbol{y}''_2|.$$

Combining (7) and (8), we obtain (3).

Corollary 2. *Let* $\boldsymbol{w}, \boldsymbol{w}' \in L_{\mathtt{diff\$diff}}$, $\boldsymbol{w} = \boldsymbol{u}'\boldsymbol{v}\$\boldsymbol{x}$ *and* $\boldsymbol{w}' = \boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}\$\boldsymbol{x}$ *be such that* $|\boldsymbol{u}''| \equiv_{\ell_2} 0$ *and* $|\boldsymbol{v}| \geq \ell_2$*. If*

$$[1, s_0, (1)] \vdash^*_{\boldsymbol{w}} [1, t, (|\boldsymbol{u}'|)],$$

then

$$[1, s_0, (1)] \vdash^*_{\boldsymbol{w}'} [1, t, (|\boldsymbol{u}'|)].$$

Proof. It suffices to show that for any $i < |\boldsymbol{u}'|$

$$[1, s, (i)] \vdash^*_{\boldsymbol{w}} [1, t, (i+1)] \tag{9}$$

implies

$$[1, s, (i)] \vdash^*_{\boldsymbol{w}'} [1, t, (i+1)] \tag{10}$$

from which the corollary follows by a straightforward induction on the length of $\boldsymbol{u}'$.

We break the automaton run (9) into three parts:

$$[1, s, (i)] \vdash_{\boldsymbol{w}} [2, s_1, (i, i)] \vdash^*_{\boldsymbol{w}} [2, s_2, (i, |\boldsymbol{u}'|)], \tag{11}$$

$$[2, s_2, (i, |\boldsymbol{u}'|)] \vdash^*_{\boldsymbol{w}} [2, s_3, (i, |\boldsymbol{u}'\boldsymbol{v}\$|)], \tag{12}$$

and

$$[2, s_3, (i, |\boldsymbol{u}'\boldsymbol{v}\$|)] \vdash^*_{\boldsymbol{w}} [2, s_4, (i, |\boldsymbol{w}|)] \vdash_{\boldsymbol{w}} [1, s_5, (i)] \vdash_{\boldsymbol{w}} [1, t, (i+1)], \tag{13}$$

where the automaton enters configuration $[1, s_5, (i)]$ after pebble 2 falls down from the right end of the input entering state s_5.

From (11), by the same moves of the automaton,

$$[1, s, (i)] \vdash_{\boldsymbol{w}'} [2, s_1, (i, i)] \vdash^*_{\boldsymbol{w}'} [2, s_2, (i, |\boldsymbol{u}'|)] \tag{14}$$

and from (12), by Corollary 1 with p, $\boldsymbol{x}$, $\boldsymbol{y}'$, and $\boldsymbol{y}''$ being i, $\boldsymbol{u}'$, $\boldsymbol{v}$, and $\boldsymbol{u}''\boldsymbol{v}$, respectively,

$$[2, s_2, (i, |\boldsymbol{u}'|)] \vdash^*_{\boldsymbol{w}'} [2, s_3, (i, |\boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}\$|)]. \tag{15}$$

Finally, from (13), by the same moves of the automaton,

$$[2, s_3, (i, |\boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}\$|)] \vdash^*_{\boldsymbol{w}'} [2, s_4, (i, |\boldsymbol{w}|)] \vdash_{\boldsymbol{w}'} [1, s_5, (i)] \vdash_{\boldsymbol{w}'} [1, t, (i+1)], \tag{16}$$

because both $\boldsymbol{w}$ and $\boldsymbol{w}'$ have the same suffix $\boldsymbol{x}$ and in both runs pebble 1 is placed above the same symbol.

Combining (14)–(16), we obtain (10).

Definition 2. *Let ℓ be a positive integer and let $\boldsymbol{u}, \boldsymbol{v} \in L_{diff}$, $\boldsymbol{u} = u_1 \cdots u_m$ and $\boldsymbol{v} = v_1 \cdots v_n$, be such that $[\boldsymbol{u}] \subseteq [\boldsymbol{v}]$: $u_i = v_{j_i}$, $i = 1, \ldots, m$. We say that $\boldsymbol{u}$ is ℓ-*spread *in $\boldsymbol{v}$, if for all $i = 1, \ldots, m$, $j_i > j_{i-1}$ and $j_i \equiv_\ell j_{i-1}$, where $j_0 = 0$.*

Proposition 2. *Let $\boldsymbol{w} = \boldsymbol{u}\boldsymbol{v}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$, where $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$, and let $1 < p'_1 < p''_1 \leq |\boldsymbol{u}|$. If*

$$[2, s, (p'_1, |\boldsymbol{u}\boldsymbol{v}\$|)] \vdash^* [1, t, (p'_1)], \tag{17}$$

then

$$[2, s, (p''_1, |\boldsymbol{u}\boldsymbol{v}\$|)] \vdash^* [1, t, (p''_1)].[9] \tag{18}$$

Proof. Let

- $\boldsymbol{u} = u_1 \cdots u_m$ and $\boldsymbol{x} = x_1 \cdots x_n$, and
- $u_{p'_1} = x_{j'}$ and $u_{p''_1} = x_{j''}$.

Then $j' < j''$ and it follows from (17) that for some states t' and t'',

$$[2, s, (p'_1, |\boldsymbol{u}\boldsymbol{v}\$|)] \vdash^* [2, t', (p'_1, |\boldsymbol{u}\boldsymbol{v}\$| + j')] \vdash [2, t'', (p'_1, |\boldsymbol{u}\boldsymbol{v}\$| + j' + 1)] \tag{19}$$

and

$$[2, t'', (p'_1, |\boldsymbol{u}\boldsymbol{v}\$| + j' + 1)] \vdash^* [1, t, (p'_1)]. \tag{20}$$

[9] The automaton enters configurations $[1, t, (p'_1)]$ and $[1, t, (p''_1)]$ after pebble 2 falls down from the right end of the input entering state t.

Since $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$, both j' and j'' are divisible by ℓ_2. Thus, it follows from (19), by Proposition 1, that

$$[2, s, (p_1'', |\boldsymbol{uv}\$|)] \vdash^* [2, t', (p_1'', |\boldsymbol{uv}\$| + j'')] \vdash [2, t'', (p_1'', |\boldsymbol{uv}\$| + j'' + 1)], \quad (21)$$

because $x_{j'} = u_{p_1'}$ and $x_{j''} = u_{p_1''}$.

Finally, since

$$(|\boldsymbol{w}| - (|\boldsymbol{uv}\$| + j' + 1)) - (|\boldsymbol{w}| - (|\boldsymbol{uv}\$| + j'' + 1)) = j'' - j' \equiv_{\ell_2} 0,$$

it follows from (20), by Proposition 1, that

$$[2, t'', (p_1'', |\boldsymbol{uv}\$| + j'' + 1)] \vdash^* [1, t, (p_1'')]. \quad (22)$$

Combining (21) and (22), we obtain (18).

Corollary 3. *Let $\boldsymbol{w} = \boldsymbol{uv}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$ be such that $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$ and $|\boldsymbol{v}| \geq \ell_2$, and let $p_1' < p_1'' \leq |\boldsymbol{u}|$ be equivalent modulo ℓ_2. If*

$$[2, s, (p_1', p_1')] \vdash^* [2, t, (p_1' + 1, p_1' + 1)],$$

then

$$[2, s, (p_1'', p_1'')] \vdash^* [2, t, (p_1'' + 1, p_1'' + 1)].$$

Proof. Let s_1, s_2, and s_3 be the states such that

$$\begin{aligned}[2, s, (p_1', p_1')] \vdash^* [2, s_1, (p_1', |\boldsymbol{uv}\$|)] &\vdash^* [1, s_2, (p_1')] \\ &\vdash [1, s_3, (p_1' + 1)] \vdash [2, t, (p_1' + 1, p_1' + 1)].\end{aligned}$$

Then, by Propositions 1 and 2,

$$\begin{aligned}[2, s, (p_1'', p_1'')] \vdash^* [2, s_1, (p_1'', |\boldsymbol{uv}\$|)] &\vdash^* [1, s_2, (p_1'')] \\ &\vdash [1, s_3, (p_1'' + 1)] \vdash [2, t, (p_1'' + 1, p_1'' + 1)].\end{aligned}$$

Proposition 3 below shows that the behavior of pebble 1 on words $\boldsymbol{uv}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$ such that $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$ is also periodic.

Proposition 3. *For each $\boldsymbol{w} = \boldsymbol{uv}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$ such that $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$ and $|\boldsymbol{v}| \geq \ell_2$, there exist positive integers $m_{\boldsymbol{w}}$ and $\ell_{\boldsymbol{w}}$ for which the following holds. If*

$$[2, s_{j_1}, (p_1, p_1)] \vdash^* [2, s_{j_2}, (p_1 + 1, p_1 + 1)] \vdash^* \cdots \vdash^* [2, s_{j_{|\boldsymbol{u}|-p_1}}, (|\boldsymbol{u}|, |\boldsymbol{u}|)],$$

then the sequence of states $s_{p_1+m_{\boldsymbol{w}}}, \ldots, s_{|\boldsymbol{u}|-p_1}$ is periodic with period $\ell_{\boldsymbol{w}}$.

Proof. Let t_i, $i = 1, \ldots, |\boldsymbol{u}| - p_1$, be such that

$$[2, s_{j_i}, (p_1 + i, p_1 + i)] \vdash^* [2, t_i, (p_1 + i, |u\boldsymbol{v}\$|)].$$

That is, t_i is the state in which pebble 2 arrives at \$, when pebble 1 is placed above the $(p_1 + i)$th symbol of $\boldsymbol{u}$.

Let $m_{\boldsymbol{w}} = \ell_2|S| + 1$. Since there are ℓ_2 equivalence classes modulo ℓ_2 and the number of different states in the sequence is bounded by $|S|$, there are two indices j_1 and j_2,

$$p_1 \le j_1 < j_2 \le p_1 + m_{\boldsymbol{w}}$$

such that $j_1 \equiv_{\ell_2} j_2$ and $t_{j_1} = t_{j_2}$.

We put $\ell_{\boldsymbol{w}} = j_2 - j_1$. It follows from Corollary 3 by a straightforward induction on $i = 0, 1, \dots$ (with $s = t_{j_1+i}$, $t = t_{j_1+i+1}$, $p_1' = j_1 + i$ and $p_1'' = j_2 + i$) that

$$[2, t_{j_1+i}, (j_1 + i, j_1 + i)] \vdash^* [2, t_{j_1+i+1}, (j_1 + i + 1, j_1 + i + 1)]$$

implies

$$[2, t_{j_2+i}, (j_2 + i, j_2 + i)] \vdash^* [2, t_{j_2+i+1}, (j_2 + i + 1, j_2 + i + 1)].$$

Thus, the proposition follows from the equality $t_{j_1} = t_{j_2}$.

Corollary 4. *There exist a positive integer ℓ_1 such that the following holds. Let $\boldsymbol{w} = \boldsymbol{u}\boldsymbol{v}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$, where $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$ and $|\boldsymbol{v}| \ge \ell_2$. If*

$$[2, s_{j_1}, (p_1, p_1)] \vdash^* [2, s_{j_2}, (p_1 + 1, p_1 + 1)] \vdash^* \cdots \vdash^* [2, s_{j_{|\boldsymbol{u}|-p_1}}, (|\boldsymbol{u}|, |\boldsymbol{u}|)],$$

then the sequence of states $s_{p_1+\ell_1}, \dots, s_{|\boldsymbol{u}|-p_1}$ is periodic with period ℓ_1.

Proof. It follows from the proof of Proposition 3 that for each $\boldsymbol{w} = \boldsymbol{u}\boldsymbol{v}\$\boldsymbol{x} \in L_{\mathtt{diff\$diff}}$ such that $\boldsymbol{u}$ is ℓ_2-spread in $\boldsymbol{x}$, $m_{\boldsymbol{w}}, \ell_{\boldsymbol{w}} \le \ell_2|S| + 1$. Thus, we can put $\ell_1 = (\ell_2|S| + 1)!$, because the latter is divisible by both $m_{\boldsymbol{w}}$ and $\ell_{\boldsymbol{w}}$ for all above $\boldsymbol{w}$.

At last, we have arrived at the proof of Theorem 1.

Proof (of Theorem 1). Assume to the contrary that $L(\mathfrak{A}) = L_{\supseteq}$. Let

$$\boldsymbol{w}' = \boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}\$\boldsymbol{x} \in L_{\supseteq} \cap L_{\subseteq},^{10}$$

where $\boldsymbol{u}'$ is ℓ_2-spread in $\boldsymbol{x}$, $|\boldsymbol{v}| \ge \ell_2$, and $|\boldsymbol{u}'| = |\boldsymbol{u}''| = \ell_1$; and let $\boldsymbol{w} = \boldsymbol{u}'\boldsymbol{v}\$\boldsymbol{x}$.

Since $\ell_1 \equiv_{\ell_2} 0$, by Corollary 2,

$$[1, s_0, (1)] \vdash^*_{\boldsymbol{w}} [1, t, (|\boldsymbol{u}'|)]$$

implies

$$[1, s_0, (1)] \vdash^*_{\boldsymbol{w}'} [1, t, (|\boldsymbol{u}'|)]$$

and, since $|\boldsymbol{u}'| = |\boldsymbol{u}''| = \ell_1$, by Corollary 4 with $p_1 = 1$,

$$[1, s_0, (1)] \vdash^*_{\boldsymbol{w}'} [1, t, (|\boldsymbol{u}'\boldsymbol{u}''|)].$$

In addition, the runs of $\mathfrak{A}$ from state t on the (same) suffix $\boldsymbol{v}\$\boldsymbol{x}$ of $\boldsymbol{w}$ and $\boldsymbol{w}'$ are the same. In particular, they terminate in the same state. However, $\boldsymbol{w}'$ belongs to $L_{\supseteq}$, whereas $\boldsymbol{w}$ does not.

Note that periodicity of pebble 1 alone (Corollary 4) without periodicity of pebble 2 (Corollary 2) is not sufficient for deleting the pattern $\boldsymbol{u}''$ from $\boldsymbol{w}'$. This is because pebble 1 has to arrive at position $|\boldsymbol{u}'|$ in the same state in the runs of $\mathfrak{A}$ on $\boldsymbol{w}$ and $\boldsymbol{w}'$ and these runs depend on the whole inputs.

[10] Thus, both $\boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}$ and $\boldsymbol{x}$ are in $L_{\mathtt{diff}}$ and $[\boldsymbol{u}'\boldsymbol{u}''\boldsymbol{v}] = [\boldsymbol{x}]$.

4 Removing the Distinguished Separator Symbol \$

In this section we show how to modify the languages $L_{\subseteq}$ and $L_{\supseteq}$ to languages whose words do not contain a distinguished separator symbol.

Let

$$L'_{\subseteq} = \{\sigma \boldsymbol{u} \sigma \boldsymbol{v} : \sigma \boldsymbol{u}, \sigma \boldsymbol{v} \in L_{\mathtt{diff}} \text{ and } [\boldsymbol{u}] \subseteq [\boldsymbol{v}]\}$$

and

$$L'_{\supseteq} = \{\sigma \boldsymbol{u} \sigma \boldsymbol{v} : \sigma \boldsymbol{u}, \sigma \boldsymbol{v} \in L_{\mathtt{diff}} \text{ and } [\boldsymbol{v}] \subseteq [\boldsymbol{u}]\}.$$

Then $L'_{\supseteq}$ is the reversal of $L'_{\subseteq}$.

The language $L'_{\subseteq}$ is accepted by a 3-wPA that acts as follows. Pebble 1 at the leftmost position is used to distinguish the second σ of the input word. Pebble 2 advances through the input to the second σ. After each move of pebble 2 on $\boldsymbol{u}$, pebble 3 moves to the second σ and then scans the suffix $\boldsymbol{v}$ of the input to find the symbol under pebble 2. At the end of the computation, pebble 1 moves to the end of the input to accept. Verifying that both $\boldsymbol{u}$ and $\boldsymbol{v}$ are in $L_{\mathtt{diff}}$ can be done by the automaton from Example 2, cf. Example 3.

It can be readily seen that each automaton accepting $L'_{\supseteq}$ modifies to an automaton accepting $L_{\supseteq}$. Thus, it follows from Theorem 1 that $L'_{\supseteq}$ is not accepted by wPA.

Alternatively, similarly to the proof of Theorem 1, one can show that $L'_{\supseteq}$ is not accepted by wPA that is normalized as follows.

- A pebble is never lifted, but falls down when moving from the right end of the input.
- Pebble 1 never leaves the leftmost position.
- Only pebble 2 can enter a final state and only after it falls down from the right end of the input.
- Immediately after pebble i moves without falling down, pebble $i + 1$ is placed.
- Immediately after pebble i falls down, pebble $i - 1$ moves.

In such a way, transitions of the form

$$(i, \$, V, s) \to \beta$$

in the proof of Theorem 1 are replaced with transitions of the form

$$(i + 1, \{1\} \cup \{j + 1 : j \in V\}, s) \to \beta.$$

5 Concluding Remark

It seems that the "shrinking" technique applied for the proof of Theorem 1 is quite appropriate for dealing with computations over infinite alphabets. For example, shrinking the input (by totally different tools) was used in [1] for proving decidability of languages accepted by certain variants of FMA.

References

1. Genkin, D., Kaminski, M., Peterfreund, L.: A note on the emptiness problem for alternating finite-memory automata. Theoret. Comput. Sci. **526**, 97–107 (2014)
2. Kaminski, M., Francez, N.: Finite-memory automata. In: Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science, Los Alamitos, CA, pp. 683–688. IEEE Computer Society Press (1990)
3. Kaminski, M., Francez, N.: Finite-memory automata. Theoret. Comput. Sci. **134**, 329–363 (1994)
4. Kaminski, M., Tan, T.: A note on two-pebble automata over infinite alphabets. Fundam. Inform. **98**, 1–12 (2010)
5. Neven, F., Schwentick, T., Vianu, V.: Towards regular languages over infinite alphabets. In: Sgall, J., Pultr, A., Kolman, P. (eds.) MFCS 2001. LNCS, vol. 2136, pp. 560–572. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44683-4_49
6. Neven, F., Schwentick, T., Vianu, V.: Finite state machines for strings over infinite alphabets. ACM Trans. Comput. Log. **5**, 403–435 (2004)
7. Peterfreund, L.: Closure under reversal of languages over infinite alphabets: a case study. Master's thesis, Department of Computer Science, Technion - Israel Institute of Technology (2015). http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2015/MSC/MSC-2015-20
8. Tan, T.: On pebble automata for data languages with decidable emptiness problem. J. Comput. Syst. Sci. **76**, 778–791 (2010)