Fundamenta Informaticae 72 (2006) 1–17
IOS Press

Interpreted Trajectories

Michael Domaratzki*
Jodrey School of Computer Science, Acadia University
Wolfville, NS Canada B4P 2R6
mike.domaratzki@acadiau.ca

Grzegorz Rozenberg†
Leiden Institute of Advanced Computer Science, LIACS
University of Leiden, Niels Bohr Weg 1
2333 CA Leiden, The Netherlands
rozenber@liacs.nl

Kai Salomaa*‡
School of Computing, Queen's University
Kingston, ON Canada K7L 3N6
ksalomaa@cs.queensu.ca

Abstract. We introduce generalized trajectories where the individual symbols are interpreted as operations performed on the operand words. The various previously considered trajectory-based operations can all be expressed in this formalism. It is shown that the generalized operations can simulate Turing machine computations. We consider the equivalence problem and a notion of unambiguity that is sufficient to make equivalence decidable for regular sets of trajectories under nonincreasing interpretations.

Keywords: Formal languages, decidability, trajectories, regularity, unambiguity

*Research supported in part by the Natural Sciences and Engineering Research Council of Canada.
†Also works: Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309-0430, USA
‡Address for correspondence: School of Computing, Queen's University, Kingston, ON Canada K7L 3N6

1. Introduction

The shuffle along trajectories operation was introduced in the paper by Mateescu, Rozenberg, and A. Salomaa [24]. For other earlier work on this topic see also [14, 21, 22, 25]. Recently there has been much renewed interest in trajectory based operations. A corresponding deletion operation has been studied by Domaratzki [2] and Kari and Sosík [19]. Semantic extensions of these operations where the shuffle operation can "see" symbols appearing in the operand words have been introduced by Domaratzki [3]. Examples of other recent work include [4, 5, 7, 8, 9, 10, 16, 17, 18]. The theory of trajectories is surveyed by Mateescu [23] and more recently by Domaratzki [6].

Shuffle or deletion along a set of trajectories T is an operation that combines two words w_1 and w_2 (or sets of words) as guided by some word in T. In the original definition of shuffle along trajectories the result does not depend on symbols occurring in w_1 and w_2, and in deletion along trajectories the operation just verifies that the deleted symbols match. The extensions introduced in [3] allow the trajectory to impose restrictions on the symbols to be inserted or deleted.

Here we consider generalized trajectories where the trajectory symbols are interpreted as operations to be performed on the operand words. In the general case, this allows the trajectories to encode computations of a Turing machine tape, or of a pushdown if the interpretation is restricted to be nonincreasing with respect to one of the arguments. The model is not capable to directly encode notions like the state of the machine, but these can be simulated indirectly in the definition of the set of "legal" trajectories. The class of languages that can be built from regular languages using a regular set of trajectories will consist of all the regular, all the context-free, or all the recursively enumerable languages depending on whether the interpretation is, respectively, nonincreasing, nonincreasing with respect to one of the arguments, or arbitrary.

Two sets of trajectories are said to be equivalent (under a given interpretation) if they define the same shuffle operation. Equivalence turns out to be undecidable even for regular sets of trajectories under nonincreasing interpretations. We consider a notion of unambiguity that is sufficient to make equivalence of regular sets of trajectories decidable. Unambiguity is closely related to determinism of sets of trajectories [24].

2. Definitions

Words that define controlled shuffle operations are called trajectories. Unless otherwise mentioned, the symbol Ξ is used to denote a finite *trajectory alphabet*.

Let Σ be an alphabet. A (Ξ, Σ)-*interpretation* I consists of three partial mappings

$$\tau^I, \pi_j^I : \Xi \times (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) \longrightarrow \Sigma^*, \quad j = 1, 2,$$

such that τ^I, π_1^I, and π_2^I are defined on the same subset of $\Xi \times (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\})$. This set is called the *domain of* I, $\mathrm{dom}(I)$.

The set of all (Ξ, Σ)-interpretations is $\mathcal{I}(\Xi, \Sigma)$. A pair (T, I), $T \subseteq \Xi^*$, $I \in \mathcal{I}(\Xi, \Sigma)$, is called an *interpreted set of trajectories.*

Next we define the I-shuffle, $w_1 \sqcup\!\sqcup_t^I w_2 \in \Sigma^* \cup \{\emptyset\}$, of words $w_1, w_2 \in \Sigma^*$ along a trajectory $t \in \Xi^*$ according to interpretation $I \in \mathcal{I}(\Xi, \Sigma)$. The case where $w_1 \sqcup\!\sqcup_t^I w_2 = \emptyset$ indicates that the operation is not defined.

Let $t = \gamma t'$, $\gamma \in \Xi$, $t' \in \Xi^*$, and $w_i = a_i w_i'$, $a_i \in \Sigma$, $w_i' \in \Sigma^*$, $i = 1, 2$. Then

$$w_1 \sqcup\!\sqcup_t^I w_2 = \begin{cases} \tau^I(\gamma, a_1, a_2)(\pi_1^I(\gamma, a_1, a_2) w_1' \sqcup\!\sqcup_{t'}^I \pi_2^I(\gamma, a_1, a_2) w_2'), & \text{if } (\gamma, a_1, a_2) \in \mathrm{dom}(I), \\ \emptyset, & \text{otherwise.} \end{cases}$$

For the cases where one or both of w_1 and w_2 is ε we set

$$a_1 w_1' \sqcup\!\sqcup_{\gamma t'}^I \varepsilon = \begin{cases} \tau^I(\gamma, a_1, \varepsilon)(\pi_1^I(\gamma, a_1, \varepsilon) w_1' \sqcup\!\sqcup_{t'}^I \pi_2^I(\gamma, a_1, \varepsilon)), & \text{if } (\gamma, a_1, \varepsilon) \in \mathrm{dom}(I), \\ \emptyset, & \text{otherwise,} \end{cases}$$

and, symmetrically,

$$\varepsilon \sqcup\!\sqcup_{\gamma t'}^I a_2 w_2' = \begin{cases} \tau^I(\gamma, \varepsilon, a_2)(\pi_1^I(\gamma, \varepsilon, a_2) \sqcup\!\sqcup_{t'}^I \pi_2^I(\gamma, \varepsilon, a_2) w_2'), & \text{if } (\gamma, \varepsilon, a_2) \in \mathrm{dom}(I), \\ \emptyset, & \text{otherwise,} \end{cases}$$

and

$$\varepsilon \sqcup\!\sqcup_{\gamma t'}^I \varepsilon = \begin{cases} \tau^I(\gamma, \varepsilon, \varepsilon)(\pi_1^I(\gamma, \varepsilon, \varepsilon) \sqcup\!\sqcup_{t'}^I \pi_2^I(\gamma, \varepsilon, \varepsilon)), & \text{if } (\gamma, \varepsilon, \varepsilon) \in \mathrm{dom}(I), \\ \emptyset, & \text{otherwise.} \end{cases}$$

Finally we set

$$w_1 \sqcup\!\sqcup_\varepsilon^I w_2 = \begin{cases} \varepsilon, & \text{if } w_1 = w_2 = \varepsilon, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Note that for a given interpretation I, τ^I and π_j^I are partial functions that are defined on $\mathrm{dom}(I)$. If the computation of an I-shuffle encounters a tuple $(\gamma, a_1, a_2) \notin \mathrm{dom}(I)$, for $\gamma \in \Xi$, $a_i \in \Sigma \cup \{\varepsilon\}$, $i = 1, 2$, this is an error-condition which means that the corresponding I-shuffle is not defined. According to the above definition, the I-operations corresponding to $(\gamma, a_1, a_2) \in \mathrm{dom}(I)$, $a_j = \varepsilon$, $j \in \{1, 2\}$, can be applied only when the jth argument is the empty word. This means that the evaluation of I-shuffle is uniquely determined by the arguments and the trajectory.

In contrast to ordinary shuffle on trajectories, the numbers of symbol occurrences in w_1, w_2, and t do not in any direct way tell us whether or not $w_1 \sqcup\!\sqcup_t^I w_2$ is defined. However, the value of $w_1 \sqcup\!\sqcup_t^I w_2$ can always be straightforwardly determined by following the recursive definition (or "computation") for $|t|$ steps.

Example 2.1. We illustrate how various previously considered trajectory-based operations can be expressed in this framework. For each interpretation I we list only the defined values of $\tau^I(\gamma, a_1, a_2)$, $\pi_1^I(\gamma, a_1, a_2)$, $\pi_2^I(\gamma, a_1, a_2)$, $\gamma \in \Xi$, $a_i \in \Sigma \cup \{\varepsilon\}$, $i = 1, 2$. In this way the description implicitly defines also $\mathrm{dom}(I)$.

(i) With $\Xi = \{0, 1\}$ the following interpretation I_{CST} yields the classical shuffle along trajectories definition [24]:

- $\tau^{I_{CST}}(0, a_1, a_2) = a_1$, $\pi_1^{I_{CST}}(0, a_1, a_2) = \varepsilon$, $\pi_2^{I_{CST}}(0, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{CST}}(1, a_1, a_2) = a_2$, $\pi_1^{I_{CST}}(1, a_1, a_2) = a_1$, $\pi_2^{I_{CST}}(1, a_1, a_2) = \varepsilon$ for all $a_1 \in \Sigma \cup \{\varepsilon\}$, $a_2 \in \Sigma$.

(ii) Deletion along trajectories (see [2] for the definition) uses trajectory alphabet $\Xi = \{i, d\}$. The following interpretation I_{CDT} yields this definition:

- $\tau^{I_{CDT}}(i, a_1, a_2) = a_1$, $\pi_1^{I_{CDT}}(i, a_1, a_2) = \varepsilon$, $\pi_2^{I_{CDT}}(i, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{CDT}}(d, a_1, a_1) = \varepsilon$, $\pi_j^{I_{CDT}}(d, a_1, a_1) = \varepsilon$, $j = 1, 2$, for all $a_1 \in \Sigma$.

(iii) The semantic shuffle on trajectories (SST) [3] uses a trajectory alphabet $\Xi = \{0, 1, \sigma, {c \atop 0}, {c \atop 1}, {c \atop \sigma} \mid c \in \Sigma\}$ that depends on the alphabet Σ of the operand words. The interpretation I_{SST} is given by the following:

- $\tau^{I_{SST}}(0, a_1, a_2) = a_1$, $\pi_1^{I_{SST}}(0, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SST}}(0, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SST}}(1, a_1, a_2) = a_2$, $\pi_1^{I_{SST}}(1, a_1, a_2) = a_1$, $\pi_2^{I_{SST}}(1, a_1, a_2) = \varepsilon$ for all $a_1 \in \Sigma \cup \{\varepsilon\}$, $a_2 \in \Sigma$.
- $\tau^{I_{SST}}(\sigma, a_1, a_1) = a_1$, $\pi_j^{I_{SST}}(\sigma, a_1, a_1) = \varepsilon$, $j = 1, 2$, for all $a_1 \in \Sigma$.
- $\tau^{I_{SST}}({a_1 \atop 0}, a_1, a_2) = a_1$, $\pi_1^{I_{SST}}({a_1 \atop 0}, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SST}}({a_1 \atop 0}, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SST}}({a_2 \atop 1}, a_1, a_2) = a_2$, $\pi_1^{I_{SST}}({a_2 \atop 1}, a_1, a_2) = a_1$, $\pi_2^{I_{SST}}({a_2 \atop 1}, a_1, a_2) = \varepsilon$ for all $a_1 \in \Sigma \cup \{\varepsilon\}$, $a_2 \in \Sigma$.
- $\tau^{I_{SST}}({a_1 \atop \sigma}, a_1, a_1) = a_1$, $\pi_j^{I_{SST}}({a_1 \atop \sigma}, a_1, a_1) = \varepsilon$, $j = 1, 2$, for all $a_1 \in \Sigma$.

(iv) The semantic deletion on trajectories (SDT) [3] uses a trajectory alphabet $\Xi = \{i, d, \sigma, {c \atop i}, {c \atop d}, {c \atop \sigma} \mid c \in \Sigma\}$. We leave as an exercise for the reader the construction of an interpretation $I_{SDT} \in \mathcal{I}(\Xi, \Sigma)$ that models this operation.

(v) Substitution on trajectories [16, 17, 18] uses trajectory alphabet $\Xi = \{0, 1\}$. The operation is given by the interpretation I_{SU}:

- $\tau^{I_{SU}}(0, a_1, a_2) = a_1$, $\pi_1^{I_{SU}}(0, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SU}}(0, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SU}}(1, a_1, a_2) = a_2$, $\pi_1^{I_{SU}}(1, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SU}}(1, a_1, a_2) = \varepsilon$ for all $a_1, a_2 \in \Sigma$, $a_1 \neq a_2$.

(vi) Splicing on routes [22] can be simulated by the following interpretation I_{SR}. Here $\Xi = \{0, \bar{0}, 1, \bar{1}\}$.

- $\tau^{I_{SR}}(0, a_1, a_2) = a_1$, $\pi_1^{I_{SR}}(0, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SR}}(0, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SR}}(\bar{0}, a_1, a_2) = \varepsilon$, $\pi_1^{I_{SR}}(\bar{0}, a_1, a_2) = \varepsilon$, $\pi_2^{I_{SR}}(0, a_1, a_2) = a_2$ for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SR}}(1, a_1, a_2) = a_2$, $\pi_1^{I_{SR}}(1, a_1, a_2) = a_1$, $\pi_2^{I_{SR}}(1, a_1, a_2) = \varepsilon$ for all $a_2 \in \Sigma$, $a_1 \in \Sigma \cup \{\varepsilon\}$.
- $\tau^{I_{SR}}(\bar{1}, a_1, a_2) = \varepsilon$, $\pi_1^{I_{SR}}(\bar{1}, a_1, a_2) = a_1$, $\pi_2^{I_{SR}}(\bar{1}, a_1, a_2) = \varepsilon$ for all $a_2 \in \Sigma$, $a_1 \in \Sigma \cup \{\varepsilon\}$.

□

To conclude this section we introduce the following notation. The I-shuffle, $I \in \mathcal{I}(\Xi, \Sigma)$, of words $w_1, w_2 \in \Sigma^*$, is extended in the natural way for sets of trajectories $T \subseteq \Xi^*$,

$$w_1 \sqcup\!\sqcup_T^I w_2 = \bigcup_{t \in T} w_1 \sqcup\!\sqcup_t^I w_2.$$

This is further extended for languages L_1, L_2 over Σ:

$$L_1 \sqcup\!\sqcup_T^I L_2 = \bigcup_{w_i \in L_i, i=1,2} w_1 \sqcup\!\sqcup_T^I w_2.$$

3. Regular Representations

Let $I \in \mathcal{I}(\Xi, \Sigma)$. We say that a language $L \subseteq \Sigma^*$ has a *regular I-representation* if $L = R_1 \sqcup\!\sqcup_T^I R_2$ for some regular set of trajectories $T \subseteq \Xi^*$ and regular languages $R_i \subseteq \Sigma^*$, $i = 1, 2$. If $\mathcal{C} \subseteq \mathcal{I}(\Xi, \Sigma)$, we say that L has a regular $\mathcal{C}$-representation if L has a regular I-representation for some $I \in \mathcal{C}$.

Next we consider restricted classes of (Ξ, Σ)-interpretations and study the question which languages have regular $\mathcal{C}$-representations for different classes $\mathcal{C}$.

An interpretation $I \in \mathcal{I}(\Xi, \Sigma)$ is said to be *1-preserving* if for all $a_1, a_2 \in \Sigma \cup \{\varepsilon\}$, $\gamma \in \Xi$ such that $(\gamma, a_1, a_2) \in \mathrm{dom}(I)$,

(i) $\pi_1^I(\gamma, a_1, a_2) \in \{a_1, \varepsilon\}$, and

(ii) if $a_1 \in \Sigma$ and $\pi_1^I(\gamma, a_1, a_2) = \varepsilon$, then $\tau^I(\gamma, a_1, a_2) = a_1$.

If I is 1-preserving then for all $t \in \Xi^*$ and $w_1, w_2 \in \Sigma^*$ such that $w_1 \sqcup\!\sqcup_t^I w_2$ is defined, the word w_1 occurs as a scattered subword of $w_1 \sqcup\!\sqcup_t^I w_2$. *2-preserving* interpretations are defined completely analogously. The interpretations given in Example 2.1 for the classical shuffle on trajectories operation, I_{CST}, and for the SST operation, I_{SST}, are both 1-preserving and 2-preserving.

An interpretation $I \in \mathcal{I}(\Xi, \Sigma)$ is said to be *1-nonincreasing* if for all $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{\varepsilon\}$, $\gamma \in \Xi$,

$$\text{if } (\gamma, a_1, a_2) \in \mathrm{dom}(I), \text{ then } \pi_1^I(\gamma, a_1, a_2) \in \Sigma \cup \{\varepsilon\}$$

and

$$\text{if } (\gamma, \varepsilon, a_2) \in \mathrm{dom}(I), \text{ then } \pi_1^I(\gamma, \varepsilon, a_2) = \varepsilon. \tag{1}$$

Intuitively, the above means just that the function π_1^I never increases the length of the first argument word. All results in the following would hold if the second condition (1) would be weakened to say $\pi_1^I(\gamma, \varepsilon, a_2) \in \Sigma \cup \{\varepsilon\}$ (and then it could be combined with the first condition). However, then π_1^I would not, strictly speaking, be nonincreasing.

Again, *2-nonincreasing* interpretations are defined symmetrically, and *nonincreasing interpretations* are both 1-nonincreasing and 2-nonincreasing. Any j-preserving interpretation is j-nonincreasing, $j = 1, 2$.

The class of languages that have regular $\mathcal{C}$-representations, as can be expected, depends essentially on whether $\mathcal{C}$ consists of nonincreasing or j-nonincreasing, $j \in \{1, 2\}$, interpretations.

The below example illustrates how any context-free language can be represented by the I-shuffle of regular languages on a regular set of trajectories using a 1-preserving interpretation.

Example 3.1. Let $M = (Q, \Omega, \Gamma, \delta, q_0, Z_0, Q_F)$ be a pushdown automaton [15, 26] where Q is the set of states, Ω is the input alphabet, Γ is a binary pushdown alphabet, δ defines the transitions of M as a mapping that associates to an element of $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ a subset of $Q \times \Gamma^{\leq 2}$, q_0 is the start state, Z_0 is the initial stack symbol and Q_F is the set of final states. The language recognized by M by final state and empty stack is denoted $L(M)$.

Above we have restricted the pushdown alphabet to be binary and restricted the automaton to always write on the stack a word of length 0, 1 or 2 to replace the current topmost stack symbol. Since the number of states in Q is not limited, any context-free language can be recognized by a pushdown automaton with the above properties. The restrictions allow us to use in the regular I-representation of $L(M)$ a (Ξ, Σ)-interpretation I where the alphabets Ξ and Σ depend only on Ω. If the alphabets would be allowed to depend on M (i.e., Ξ or Σ could explicitly contain the states of M), a somewhat simpler construction would be possible.

Choose

$$\Xi = \{0, 1\} \cup (\Omega \cup \{\varepsilon\}) \times \Gamma \times \Gamma^{\leq 2} \quad (0, 1 \notin \Omega \cup \Sigma)$$

and,

$$\Sigma = \Omega \cup \Gamma.$$

Let $f : Q \times Q \longrightarrow \{0, 1\}^*$ be a bijective mapping. We define

$$R_1 = \Omega^*, \quad R_2 = \{Z_0\}, \text{ and}$$

$$\begin{aligned} T \;\; = \;\; & \{f(p_1, p_2)(c_1, d_1, e_1)f(p_2, p_3)(c_2, d_2, e_2) \cdots f(p_{n-1}, p_n)(c_{n-1}, d_{n-1}, e_{n-1}) \mid \\ & n \geq 1, p_i \in Q, i = 1, \ldots n, p_1 = q_0, p_n \in Q_F, c_j \in \Omega \cup \{\varepsilon\}, d_j \in \Gamma, e_j \in \Gamma^{\leq 2}, \\ & (p_{j+1}, e_j) \in \delta(p_j, c_j, d_j), j = 1, \ldots, n-1\}. \end{aligned}$$

We define a 1-preserving (Ξ, Σ)-interpretation I such that

$$R_1 \sqcup\!\sqcup_T^I R_2 = L(M). \tag{2}$$

Intuitively, each word $t \in T$ encodes a sequence of transitions of M and the interpretation I performs the corresponding operations on the argument words when the first argument is viewed as the input for M and the second argument as the stack contents. As usual, below we list the values of τ^I, π_j^I, $j = 1, 2$, for tuples in $\mathrm{dom}(I)$, all other values are undefined.

(i) Let $z \in \{0, 1\}$, $a_1, a_2 \in \Sigma \cup \{\varepsilon\}$ $(= \Omega \cup \Gamma \cup \{\varepsilon\})$. Then

$$\tau^I(z, a_1, a_2) = \varepsilon, \ \pi_j^I(z, a_1, a_2) = a_j, \ j = 1, 2.$$

(ii) Let $c \in \Omega$, $d \in \Gamma$, $e \in \Gamma^{\leq 2}$. Then

$$\tau^I((c, d, e), c, d) = c, \ \pi_1^I((c, d, e), c, d) = \varepsilon, \ \pi_2^I((c, d, e), c, d) = e.$$

(iii) Let $d \in \Gamma$, $e \in \Gamma^{\leq 2}$, $a \in \Omega \cup \{\varepsilon\}$. Then

$$\tau^I((\varepsilon, d, e), a, d) = \varepsilon, \ \pi_1^I((\varepsilon, d, e), a, d) = a, \ \pi_2^I((\varepsilon, d, e), a, d) = e.$$

The rules (i) just erase elements of $\{0,1\}$ from the trajectory and do not affect the argument words. Elements $\{0,1\}$ are used to encode pairs of states and to enforce, by way of the definition of T, that together with the following triple in $(\Omega \cup \{\varepsilon\}) \times \Gamma \times \Gamma^{\leq 2}$ the pair of states encodes a correct transition of M. Rules (ii) use a trajectory symbol $(c,d,e) \in \Omega \times \Gamma \times \Gamma^{\leq 2}$, and I verifies that the next symbol in the first argument word is c, τ^I writes c to the output, π_1^I erases c from the first argument, and π_2^I replaces in the second argument, which represents the stack, the symbol d with $e \in \Gamma^{\leq 2}$. Similarly rules (iii) simulate an ε-transition of M when the first argument is viewed as the input and the second argument as the stack. Here the next input symbol $a \in \Omega \cup \{\varepsilon\}$ can be arbitrary and it is not changed. The case $a = \varepsilon$ corresponds to the situation that the operations have reached the end of the first argument word.

From the above description it is clear that for any $t \in T$, $w_1 \in \Omega^*$ and $w_2 \in \Gamma^*$, if $w_1 \sqcup\!\sqcup_t^I w_2$ is defined, the evaluation of this operation directly simulates the particular computation of M on input w_1 and remaining stack contents w_2 that is given by the sequence of transitions encoded by t. In this case $w_1 \sqcup\!\sqcup_t^I w_2 = w_1$. On the other hand, for any $w_1 \in L(M)$ there exists $t \in T$ that corresponds to a sequence of accepting computation steps of M and consequently $w_1 \sqcup\!\sqcup_t^I Z_0 = w_1$. It follows that (2) holds. □

As mentioned above, the alphabets Ξ and Σ used to construct the (Ξ,Σ)-interpretation I in Example 3.1 depend only on the input alphabet Ω of M. For easier readability the construction defines $\Sigma = \Omega \cup \Gamma$ but if Ω has at least two symbols, nothing prevents us from using elements of Ω as the stack symbols, and consequently Σ needs only two symbols. If Ω is binary, the construction gives a trajectory alphabet Ξ of size 44. The size of Ξ could be reduced[1] but it is not clear whether Ξ could have e.g. just two symbols.

By recalling the standard construction used to simulate a Turing machine tape by two pushdowns, it is easy to modify the construction of Example 3.1 to allow general I-shuffles (where I can be increasing with respect to both arguments) to simulate computations of a single-tape Turing machine M.

Let Z_1 and Z_2 be symbols used as endmarkers. A particular computation of M on input w can be simulated by a shuffle $(wZ_1) \sqcup\!\sqcup_t^I Z_2$ where, as in Example 3.1, t encodes the sequence of transitions used in the computation of M. The computation can first read the input word w, output w using the function τ^I and at the same time store a copy of w in reverse order to the second pushdown using π_2^I. After reaching the endmarker Z_1, the computation uses only ε as values for τ^I, (the symbols "pushed back" to the first argument can be marked to distinguish them from the original "input" symbols) and the two pushdowns then simulate the actual computation of M on w. Without loss of generality, the computation of M can be made to erase the tape contents before accepting, and this guarantees that if t gives a sequence of transitions that accepts w, the operations used to evaluate $(wZ_1) \sqcup\!\sqcup_t^I Z_2$ consume both argument words.

With the above construction, $(wZ_1) \sqcup\!\sqcup_t^I Z_2 = w$ if the sequence of transitions given by t accepts w, and it is undefined otherwise. By choosing T to be the set of all valid sequences of transitions (that begin in the start state, end in an accepting state, and where the initial state of each transition equals to the new state of the previous transition), we get $\Omega^* Z_1 \sqcup\!\sqcup_T^I Z_2 = L(M)$, where Ω is the input alphabet of M.

Since any language having a regular I-representation, $I \in \mathcal{I}(\Xi,\Sigma)$, is clearly recursively enumerable, we have:

[1] An immediate observation is that f could use a unary encoding instead of a binary one.

Theorem 3.1. Let Σ be an alphabet. There exists a trajectory alphabet Ξ such that the class of languages with regular I-representations, $I \in \mathcal{I}(\Xi, \Sigma)$, consists of all the recursively enumerable languages over Σ.

Recall that for given $t \in \Xi^*$ and $w_1, w_2 \in \Sigma^*$ the value of $w_1 \sqcup\!\sqcup_t^I w_2$ can be straightforwardly computed. On the other hand, the existence of universal Turing machines gives the following result.

Corollary 3.1. There exist alphabets Ξ and Σ, a regular language $T \subseteq \Xi^*$, and an interpretation $I \in \mathcal{I}(\Xi, \Sigma)$ such that the following problem is undecidable: Given words $w_1, w_2 \in \Sigma^*$, determine whether or not $w_1 \sqcup\!\sqcup_T^I w_2$ is empty.

We contrast the result of Theorem 3.1 by showing that for any nonincreasing interpretation I and a regular set of trajectories T the operation $\sqcup\!\sqcup_T^I$ preserves regularity. The result extends the previously known closure properties of the classical trajectory based operations [24, 2, 3, 19]. Note that the interpretations $I_{CST}, I_{CDT}, I_{SST}, I_{SDT}$ given in Example 2.1 to simulate the classical trajectory operations are all nonincreasing.

Lemma 3.1. Let I be a nonincreasing (Ξ, Σ)-interpretation and $T \subseteq \Xi^*$ a regular set of trajectories. For any regular languages $R_i \subseteq \Sigma^*$, $i = 1, 2$, the language $R_1 \sqcup\!\sqcup_T^I R_2$ is regular.

Proof. Let $A_j = (\Sigma, Q_j, q_{j,0}, Q_{j,F}, \delta_j)$ be a DFA recognizing R_j, $j = 1, 2$, and let $B = (\Xi, P, p_0, P_F, \mu)$ be a DFA recognizing T. We assume that there are no incoming transitions to any of the start states $q_{j,0}$, $j = 1, 2$, or p_0.

Recall that a generalized NFA is a finite automaton where transitions are labeled by arbitrary words. It is well known that any language recognized by a generalized NFA is regular [27]. We define a generalized NFA C that recognizes $R_1 \sqcup\!\sqcup_T^I R_2$. Denote

$$Q'_j = Q_j \times (\Sigma \cup \{\sharp, \flat\}).$$

Choose

$$C = (\Sigma,\ P \times Q'_1 \times Q'_2,\ (p_0, (q_{1,0}, \sharp), (q_{2,0}, \sharp)),\ (P_F \times (Q_{1,F} \times \{\flat\}) \times (Q_{2,F} \times \{\flat\})) \cup X,\ \rho),$$

where

$$X = \begin{cases} \{(p_0, (q_{1,0}, \sharp), (q_{2,0}, \sharp))\}, & \text{if } p_0 \in P_F, q_{j,0} \in Q_{j,F}, j = 1, 2, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{3}$$

and the transitions of ρ are defined as follows.

Let $(p, (q_1, x_1), (q_2, x_2)), (p', (q'_1, x'_1), (q'_2, x'_2)) \in P \times Q'_1 \times Q'_2$ and $(\gamma, a_1, a_2) \in \mathrm{dom}(I)$, $\gamma \in \Xi$, $a_i \in \Sigma \cup \{\varepsilon\}$, $i = 1, 2$. Then ρ has a transition from $(p, (q_1, x_1), (q_2, x_2))$ to $(p', (q'_1, x'_1), (q'_2, x'_2))$ labeled by $\tau^I(\gamma, a_1, a_2)$ always when the following conditions hold:

(i) $\mu(p, \gamma) = p'$,

(ii) if $\pi_j^I(\gamma, a_1, a_2) = \varepsilon$:

  (a) if $a_j \in \Sigma$, $j \in \{1, 2\}$:

    i. if $x_j = \sharp$, then $\delta_j(q_j, a_j) = q'_j$ and $x'_j = \sharp$,

ii. if $x_j = \flat$, the transition is not defined for $a_j \in \Sigma$,

iii. if $x_j = c \in \Sigma$, then $a_j = c$, $q'_j = q_j$ and $x'_j = \sharp$,

(b) if $a_j = \varepsilon$, $j \in \{1, 2\}$:

i. if $x_j = \sharp$, then $q'_j = q_j$ and $x'_j = \flat$, ($\flat$ indicates a guess that the jth input is finished),

ii. if $x_j = \flat$, then $q'_j = q_j$ and $x'_j = \flat$,

iii. if $x_j = c \in \Sigma$, then the transition is not defined.

(iii) if $\pi^I_j(\gamma, a_1, a_2) = c \in \Sigma$:

(a) if $a_j \in \Sigma$, $j \in \{1, 2\}$:

i. if $x_j = \sharp$, then $\delta_j(q_j, a_j) = q'_j$ and $x'_j = c$, $j \in \{1, 2\}$,

ii. if $x_j = \flat$, then the transition is not defined with $a_j \in \Sigma$,

iii. if $x_j \in \Sigma$, then $x_j = a_j$, $x'_j = c$ and $q'_j = q_j$, $j \in \{1, 2\}$.

(b) if $a_j = \varepsilon$, $j \in \{1, 2\}$:

i. if $x_j = \sharp$, then $q'_j = q_j$ and $x'_j = \flat$,

ii. if $x_j = \flat$, then $q'_j = q_j$ and $x'_j = \flat$,

iii. if $x_j \in \Sigma$, then the transition is not defined.

Note that since I is nonincreasing, $\pi^I_j(\gamma, a_1, a_2)$ is in $\Sigma \cup \{\varepsilon\}$ when it is defined.

In the simulation of A_j by C, a state $(q_j, \sharp)$ represents that A_j has reached state q_j, and a state (q_j, c), $c \in \Sigma$ represents the situation that the computation of A_j after reading the original current symbol of the jth argument word has reached state q_j and the current symbol has been replaced by c in the previous operation π^I_j. A state $(q_j, \flat)$ represents a situation where (C has guessed that) the jth argument word has been completely processed.

The computation of C nondeterministically factorizes the input into a sequence of subwords of the form $\tau^I(\gamma, a_1, a_2)$, $(\gamma, a_1, a_2) \in \mathrm{dom}(I)$. For each subword $\tau^I(\gamma, a_1, a_2)$, C makes a transition where according to (i) the first component of the states directly simulates the operation of B, thus verifying that the guessed sequence of trajectory symbols is in T. The rules (ii) and (iii) then simulate the computation of A_1 and A_2 on the argument words that may be modified by the operations π^I_j, $j = 1, 2$.

In rules (ii) (a), the π^I_j operation consumes the current symbol from the jth argument. If the current state is $(q_j, \sharp)$, the symbol a_j has not been modified by previous applications of π^I_j and the $(j+1)$th component of the state of C (i.e., the state of A_j) is modified directly using δ_j.

If the current state is $(q_j, \flat)$, the computation of C is not successful. Occurrence of $\flat$ indicates that the computation has simulated an I-operation with ε from the jth argument and it cannot now read $a_j \in \Sigma$.

If the current state is (q_j, c), this means that the current symbol been modified by a previous application of π^I_j and the modified symbol is stored in the state (as c). In the state, the component q_j indicates the state that A_j reached after reading the original symbol in this position and A_j continues the computation in state q_j.

The rules (ii) (b) are the same as above, except now we simulate an operation of I with $(\gamma, \varepsilon, a_2) \in \mathrm{dom}(I)$. If $x_j = \sharp$, C guesses that the jth argument is finished and marks this by setting $x'_j = \flat$. If $x_j = \flat$, C has already previously guessed that the jth argument is finished. The case $x_j = c \in \Sigma$ is an error situation.

The rules (iii) replace the current symbol a_j of the jth argument word by c. In (iii) (a), if $x_j = \sharp$, a_j is "from" the original argument word and hence the state reached after this symbol is obtained according to the transition function δ_j. The new symbol c is stored as x'_j. If $x_j = \flat$, C has previously guessed that the jth argument is finished and this is an error. If $x_j \in \Sigma$, the current symbol has been modified before. The newly modified symbol is again stored as x'_j and the operation does not change the state of A_j at this position.

Finally, the rules (iii) (b) represent the same situation as in (iii) (a), except that C simulates an I-operation that is performed at the end of the jth input, that is $a_j = \varepsilon$.

As explained above, the computation of C verifies that the input word can be written in a form

$$\tau^I(\gamma_1, a_{1,1}, a_{1,2}) \cdot \tau^I(\gamma_2, a_{2,1}, a_{2,2}) \cdot \ldots \cdot \tau^I(\gamma_m, a_{m,1}, a_{m,2}), \quad m \geq 0, \tag{4}$$

where $t = \gamma_1 \cdots \gamma_m \in T$ and (4) is the sequence of outputs produced by applying the operation $\sqcup\!\sqcup_t^I$ to some words $w_1 \in R_1$ and $w_2 \in R_2$. The inclusion of X in the set of final states of C according to (3) guarantees that C accepts ε if ε is in T as well as in both R_1 and R_2. Naturally it is possible that C accepts ε also using a nonempty trajectory if in (4) $\tau^I(\gamma_i, a_{i,1}, a_{i,2}) = \varepsilon$, $i = 1, \ldots, m$.

Above we have verified that C nondeterministically checks that the input is of the form $w_1 \sqcup\!\sqcup_t^I w_2$, $t \in T$, $w_i \in R_i$, $i = 1, 2$, and, consequently, $L(C) = R_1 \sqcup\!\sqcup_T^I R_2$. □

Lemma 3.2. Let I be a 1-nonincreasing (or, respectively, 2-nonincreasing) (Ξ, Σ)-interpretation and $T \subseteq \Xi^*$ be a regular set of trajectories. For any regular languages $R_i \subseteq \Sigma^*$, $i = 1, 2$, the language $R_1 \sqcup\!\sqcup_T^I R_2$ is context-free.

Proof. We only sketch the idea of the proof and leave the details of the construction for the reader. A pushdown automaton to recognize $R_1 \sqcup\!\sqcup_T^I R_2$ can be obtained from the construction used in the proof of Lemma 3.1 by using the stack to remember the prefixes that π_2^I adds to the second argument (or that π_1^I adds to the first argument when I is 2-nonincreasing). □

Let $\mathcal{UI}$ denote the class of unrestricted interpretations, $\mathcal{N}_j$ the class of j-nonincreasing interpretations, $j = 1, 2$, and $\mathcal{N}$ the class of nonincreasing interpretations.

Now by Theorem 3.1, Lemmas 3.1 and 3.2, and Example 3.1 we get the following characterizations. If we consider languages over a fixed alphabet Σ, in (i) and (ii) of Theorem 3.2 the trajectory alphabet needed for the representations is also fixed.

Theorem 3.2. (i) The class of languages that have regular $\mathcal{UI}$-representations consists of all the recursively enumerable languages.

(ii) The class of languages that have regular $\mathcal{N}_j$-representations consists of all the context-free languages, $j = 1, 2$.

(iii) The class of languages that have regular $\mathcal{N}$-representations consists of the regular languages.

4. Equivalence Problem

It is known that two sets of trajectories T_1 and T_2 as considered in [24] define the same shuffle operation if and only if $T_1 = T_2$. The same does not hold for the SST operation considered in [3] but there it is shown that we can algorithmically decide whether or not two regular sets of trajectories define the same SST operation.

The situation is more problematic for sets of interpreted trajectories. Example 3.1 directly implies that equivalence of regular sets of trajectories under 1-nonincreasing (or 2-nonincreasing) interpretations is undecidable. It turns out that also equivalence of regular sets of trajectories under a nonincreasing interpretation is undecidable.

We say that trajectories $T_1, T_2 \subseteq \Xi^*$ are *I-equivalent*, where $I \in \mathcal{I}(\Xi, \Sigma)$, if for all $w_1, w_2 \in \Sigma^*$,

$$w_1 \sqcup\!\sqcup_{T_1}^I w_2 = w_1 \sqcup\!\sqcup_{T_2}^I w_2.$$

It follows that for any languages $L_1, L_2 \subseteq \Sigma^*$, $L_1 \sqcup\!\sqcup_{T_1}^I L_2 = L_1 \sqcup\!\sqcup_{T_2}^I L_2$ when T_1 and T_2 are I-equivalent.

We establish the undecidability result by using a reduction from the equivalence problem for finite transducers. By a finite transducer we mean a tuple

$$M = (\Omega, Q, q_0, Q_F, E), \tag{5}$$

where Ω is the input and the output alphabet, Q is the finite set of states, $q_0 \in Q$ is the start state, $Q_F \subseteq Q$ is the set of final states and E is the set of transitions,

$$E \subseteq (Q \times \Omega \times \{\varepsilon\} \times Q) \cup (Q \times \{\varepsilon\} \times \Omega \times Q). \tag{6}$$

The transduction realized by M is denoted $\Delta(M)$ ($\subseteq \Omega^* \times \Omega^*$). It is known that any rational relation (with same input and output alphabet) can be realized by a transducer with transitions as in (6) [1].

Theorem 4.1. Let $\Sigma = \{a, b\}$ and $\Xi = \{0, 1, a_1, a_2, b_1, b_2\}$. Given a nonincreasing (Ξ, Σ)-interpretation I and regular sets of trajectories T_1, T_2, it is undecidable whether or not T_1 and T_2 are I-equivalent.

Proof. Let $I \in \mathcal{I}(\Xi, \Sigma)$ be defined as follows:

(i) Let $z \in \{0, 1\}$ and $x_j \in \Sigma \cup \{\varepsilon\}$, $j = 1, 2$. Then

$$\tau^I(z, x_1, x_2) = \varepsilon,\ \pi_j^I(z, x_1, x_2) = x_j, \quad j = 1, 2.$$

(ii) Let z be either symbol a or b, and $x \in \Sigma \cup \{\varepsilon\}$. Then

(a) $\tau^I(z_1, z_1, x) = \varepsilon$, $\pi_1^I(z_1, z_1, x) = \varepsilon$, $\pi_2^I(z_1, z_1, x) = x$.
(b) $\tau^I(z_2, x, z_2) = \varepsilon$, $\pi_1^I(z_2, x, z_2) = x$, $\pi_2^I(z_2, x, z_2) = \varepsilon$.

(In all cases not listed above, the I-operations are not defined.) Since τ^I does not produce any output, for all $w_1, w_2 \in \Sigma^*$ and $t \in \Xi^*$,

$$w_1 \sqcup\!\sqcup_t^I w_2 \text{ is defined if and only if } w_1 \sqcup\!\sqcup_t^I w_2 = \varepsilon. \tag{7}$$

Consider an arbitrary finite transducer with input and output alphabet Σ,

$$M = (\Sigma, Q, q_0, Q_F, E),$$

where the transitions are of the form (6). Let $f : Q \times Q \longrightarrow \{0, 1\}^*$ be an injective mapping. When $q_1, q_2 \in Q$ and $z \in \{a_1, a_2, b_1, b_2\}$, we say that the word $f(q_1, q_2)z$ *encodes an E-transition* if the following holds:

- If $z = x_1$, then $(q_1, x, \varepsilon, q_2) \in E$, $x \in \{a, b\}$.
- If $z = x_2$, then $(q_1, \varepsilon, x, q_2) \in E$, $x \in \{a, b\}$.

We define $T \subseteq \Xi^*$ as follows:

$$\begin{aligned} T \;=\; & \{f(p_1, p_2)z_1 f(p_2, p_3)z_2 \cdot \ldots \cdot f(p_n, p_{n+1})z_n \mid (n = 0 \text{ and } q_0 \in Q_F) \text{ or} \\ & n \geq 1, p_i \in Q, i = 1, \ldots, n+1, p_1 = q_0, p_{n+1} \in Q_F, z_j \in \{a_1, a_2, b_1, b_2\}, \\ & \text{and } f(p_j, p_{j+1})z_j \text{ encodes an } E\text{-transition}, j = 1, \ldots, n\}. \end{aligned}$$

For any $t \in T$ and $w_1, w_2 \in \Sigma^*$, the evaluation of $w_1 ⧢_t^I w_2$ simulates a computation of M with input w_1 and output w_2. According to the definition of T, the trajectory t encodes a sequence of E-transitions leading from the start state to a final state. The rules (i) of the interpretation I erase the encodings of pairs of states from t (that is, all symbols 0 and 1 are processed without changing the arguments w_1 and w_2). The purpose of the states is just to guarantee that the sequence of symbols z_j, (z is a or b, $j \in \{1, 2\}$), represent input/output actions that can occur in a computation of M. According to rules (ii), an occurrence of a trajectory symbol z_1 causes that z is read from w_1 and an occurrence of a symbol z_2 causes that z is read from w_2.

The above means that for all $w_1, w_2 \in \Sigma^*$,

$$w_1 ⧢_t^I w_2 \text{ is defined if and only if } (w_1, w_2) \in \Delta(M). \tag{8}$$

Note that this holds also for $w_1 = w_2 = \varepsilon$ since $\varepsilon \in T$ if and only if $q_0 \in Q_F$.

Since equivalence of finite transducers is undecidable [1], (7) and (8) imply that I-equivalence of regular sets of trajectories ($\subseteq \Xi^*$) is undecidable. □

4.1. Unambiguity and determinism

Next we consider special cases where equivalence can be decided.

Let $I \in \mathcal{I}(\Xi, \Sigma)$. We say that a set of trajectories $T \subseteq \Xi^*$ is *I-unambiguous* if for all $w_1, w_2, w_3 \in \Sigma^*$ there exists at most one $t \in T$ such that

$$w_3 = w_1 ⧢_t^I w_2.$$

The above definition of unambiguity closely resembles the definition of deterministic sets of trajectories [24] but the two notions do not coincide if Σ is a singleton alphabet. Let I_{CST} be the interpretation used in Example 2.1 to represent the classical shuffle on trajectories operation. Recall that $T \subseteq \{0, 1\}^*$ is said to be *deterministic* [24] if for all $w_1, w_2 \in \Sigma^*$, the cardinality of the set $w_1 ⧢_T^{I_{CST}} w_2$ is at most one. If $T \subseteq \{0, 1\}^*$ is I_{CST}-unambiguous then T is always deterministic in the above sense. This follows by observing that if $w_1 ⧢_T^{I_{CST}} w_2$ has cardinality at least two, then there exist $t_1, t_2 \in T$, $t_1 \neq t_2$ such that $a^{|w_1|} ⧢_{t_1}^{I_{CST}} a^{|w_2|} = a^{|w_1|} ⧢_{t_2}^{I_{CST}} a^{|w_2|}$ where a is any symbol of Σ.

The converse implication holds if Σ has at least two symbols. We note that if $w_1 ⧢_{t_1}^{I_{CST}} w_2 = w_1 ⧢_{t_2}^{I_{CST}} w_2$, for $w_1, w_2 \in \Sigma^*$, $t_1, t_2 \in T \subseteq \{0, 1\}^*$, $t_1 \neq t_2$, then $a^{|w_1|} ⧢_T^{I_{CST}} b^{|w_2|}$, where $a, b \in \Sigma$, $a \neq b$, contains at least two words. On the other hand, if Σ is a singleton alphabet, then any set of trajectories $T \subseteq \{0, 1\}^*$ is deterministic in the sense of [24].

When considering general interpreted sets of trajectories, a significant drawback is, as observed below, that I-unambiguity is undecidable for regular sets of trajectories, even if I is nonincreasing.

Lemma 4.1. Let Σ be an alphabet with at least two symbols. Given trajectory alphabet Ξ, a nonincreasing interpretation $I \in \mathcal{I}(\Xi, \Sigma)$ and a regular set of trajectories $T \subseteq \Xi^*$ it is undecidable whether or not T is I-unambiguous.

Proof. Let
$$\{(u_1, v_1), \ldots, (u_m, v_m) \mid m \geq 1, u_i, v_i \in \Sigma^+, i = 1, \ldots, m\} \tag{9}$$
be an arbitrary instance of the Post correspondence problem (PCP) [15, 12].

We choose as the trajectory alphabet

$$\Xi = \{(i, 1, x_i) \mid 1 \leq i \leq m, 1 \leq x_i \leq |u_i|\} \cup \{(i, 2, y_i) \mid 1 \leq i \leq m, 1 \leq y_i \leq |v_i|\}$$

Let $h : \{1, \ldots, m\} \longrightarrow \Sigma^*$ be an injective mapping such that $|h(i)| = |h(j)|$ for all $i, j \in \{1, \ldots, m\}$. We define a (Ξ, Σ)-interpretation I by the following conditions:

(i) $\tau^I((i, 1, 1), b, b) = h(i)$, $\pi_j^I((i, 1, 1), b, b) = \varepsilon$, $j = 1, 2$, if $b \in \Sigma$ is the first symbol of u_i, $i \in \{1, \ldots, m\}$.

(ii) $\tau^I((i, 1, x), b, b) = \varepsilon$, $\pi_j^I((i, 1, x), b, b) = \varepsilon$, $j = 1, 2$, if $2 \leq x \leq |u_i|$ and $b \in \Sigma$ is the xth symbol of u_i, $i \in \{1, \ldots, m\}$.

(iii) $\tau^I((i, 2, 1), b, b) = h(i)$, $\pi_j^I((i, 2, 1), b, b) = \varepsilon$, $j = 1, 2$, if $b \in \Sigma$ is the first symbol of v_i, $i \in \{1, \ldots, m\}$.

(iv) $\tau^I((i, 2, x), b, b) = \varepsilon$, $\pi_j^I((i, 2, x), b, b) = \varepsilon$, $j = 1, 2$, if $2 \leq x \leq |v_i|$ and $b \in \Sigma$ is the xth symbol of v_i, $i \in \{1, \ldots, m\}$.

Let $f : \{1, \ldots, m\} \longrightarrow \Xi^*$ be the morphism defined by the condition

$$f(i) = (i, 1, 1) \cdots (i, 1, |u_i|), \;\; 1 \leq i \leq m,$$

and let $g : \{1, \ldots, m\} \longrightarrow \Xi^*$ be the morphism defined by the condition

$$g(i) = (i, 2, 1) \cdots (i, 2, |v_i|), \;\; 1 \leq i \leq m.$$

We define
$$T = f(\{1, \ldots, m\}^+) \cup g(\{1, \ldots, m\}^+).$$

Assume that for some $w_1, w_2 \in \Sigma^*$ and $t_1, t_2 \in T$, $t_1 \neq t_2$, both operations $w_1 \sqcup\!\sqcup_{t_j}^I w_2$, $j = 1, 2$, are defined and
$$w_1 \sqcup\!\sqcup_{t_1}^I w_2 = w_1 \sqcup\!\sqcup_{t_2}^I w_2. \tag{10}$$
For any $x \in \{1, \ldots, m\}^+$, any shuffle along $f(x)$ (respectively, $g(x)$) that is defined outputs the h-encoding of x. Thus if $t_1 \neq t_2$ and (10) holds, the only possibility is that $t_1 = f(x)$, $x \in \{1, \ldots, m\}^+$) and $t_2 = g(y)$, $y \in \{1, \ldots, m\}^+$), or vice versa. Now (10) implies that $x = y$ gives a solution to the PCP instance (9).

Conversely, any solution $i_1, \ldots, i_k$ to the PCP instance (9) gives distinct trajectories $f(i_1 \cdots i_k)$, $g(i_1 \cdots i_k) \in T$ such that (10) holds when $w_1 = w_2$ is the word $u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$. $\square$

It is known that Post correspondence problem remains undecidable when the size of the instances (that is, m in (9)) is bounded to be at most nine [12, 13]. However, the trajectory alphabet used in the proof of Lemma 4.1 depends both on the size of the PCP instance and the lengths of the words appearing in the instance.

Due to Lemma 4.1, we consider below also the more restrictive notion of determinism that is associated with an interpretation rather than with individual sets of trajectories.

For words w_1 and w_2, we denote $w_1 \not\sim_{\text{pref}} w_2$ if w_1 is not a prefix of w_2 and w_2 is not a prefix of w_1.

We say that a (Ξ, Σ)-interpretation I is *deterministic* if

(i) for any $a_1, a_2 \in \Sigma \cup \{\varepsilon\}$, and any $\gamma_1, \gamma_2 \in \Xi$, $\gamma_1 \neq \gamma_2$, such that $(\gamma_j, a_1, a_2) \in \text{dom}(I)$, $j = 1, 2$,

$$\tau^I(\gamma_1, a_1, a_2) \not\sim_{\text{pref}} \tau^I(\gamma_2, a_1, a_2),$$

(ii) for any $(\gamma, a_1, a_2) \in \text{dom}(I)$, $\gamma \in \Xi$, $a_1, a_2 \in \Sigma \cup \{\varepsilon\}$,

$$\tau^I(\gamma, a_1, a_2) \neq \varepsilon.$$

Let $I \in \mathcal{I}(\Xi, \Sigma)$ be deterministic and let $w_1, w_2 \in \Sigma^*$ be arbitrary. Consider trajectories $t_1, t_2 \in \Xi^*$, $t_1 \neq t_2$, such that $w_1 ш^I_{t_i} w_2$ is defined, $i = 1, 2$. If we can write $t_1 = t\gamma_1 t_1'$ and $t_2 = t\gamma_2 t_2'$, $t, t_1', t_2' \in \Xi^*$, $\gamma_1, \gamma_2 \in \Xi$, $\gamma_1 \neq \gamma_2$, then condition (i) implies that

$$w_1 ш^I_{t_1} w_2 \neq w_1 ш^I_{t_2} w_2. \tag{11}$$

If t_1 is a proper prefix of t_2, again (11) holds by condition (ii).

Proposition 4.1. If $I \in \mathcal{I}(\Xi, \Sigma)$ is a deterministic interpretation, then any set of trajectories $T \subseteq \Xi^*$ is I-unambiguous.

For the positive decidability result we can compare sets of trajectories with different interpretations. Let $I_1, I_2 \in \mathcal{I}(\Xi, \Sigma)$ and $T_1, T_2 \subseteq \Xi^*$. We say that the interpreted sets of trajectories (T_1, I_1) and (T_2, I_2) are *equivalent* if for all $w_1, w_2 \in \Sigma^*$,

$$w_1 ш^I_{T_1} w_2 = w_1 ш^I_{T_2} w_2.$$

The above implies that $L_1 ш^I_{T_1} L_2 = L_1 ш^I_{T_2} L_2$ for any $L_1, L_2 \subseteq \Sigma^*$.

Theorem 4.2. Let $I_1, I_2 \in \mathcal{I}(\Xi, \Sigma)$ be nonincreasing interpretations. For given regular sets of trajectories T_1 and T_2 ($\subseteq \Xi^*$) such that T_i is I_i-unambiguous, $i = 1, 2$, it is decidable whether or not the interpreted sets of trajectories (T_1, I_1) and (T_2, I_2) are equivalent.

Proof. Since T_j is regular and I_j nonincreasing, a nondeterministic 3-tape automaton M_j [11, 20] can recognize 3-tuples of the form $(w_1, w_2, w_1 ш^{I_j}_t w_2)$, $w_1, w_2 \in \Sigma^*$, $t \in T_j$, $j \in \{1, 2\}$.

Depending on the next symbols a_1, a_2 read from the first two tapes, M_j guesses $\gamma \in \Xi$ such that $(\gamma, a_1, a_2) \in \text{dom}(I)$. Then it reads $\tau^I(\gamma, a_1, a_2)$ from the third tape, and the tape heads on the first two tapes remain stationary or are advanced one step depending on the values of $\pi^I_i(\gamma, a_1, a_2)$, $i = 1, 2$. In the former case the symbol $\pi^I_i(\gamma, a_1, a_2)$ is stored in the state of M_j. Using the finite state memory, M_j

can verify that the sequence of guessed trajectory symbols is in the regular language T_j. Except for the choice of the next symbol $\gamma \in \Xi$, the operation of M_j can be made to be completely deterministic.

The crucial observation is that, since T_j is I_j-unambiguous, for any 3-tuple of words

$$(w_1, w_2, w_3) \tag{12}$$

there exists at most one trajectory $t \in T_j$ such $w_3 = w_1 \sqcup\!\sqcup_t^I w_2$. Thus, if M_j accepts (12), M_j has exactly one accepting computation on (12).

Now (T_1, I_1) and (T_2, I_2) are equivalent if and only if M_1, M_2 are multiplicity equivalent. The result follows since multiplicity equivalence of nondeterministic multitape automata is decidable [11]. □

By Proposition 4.1 we get a positive decidability result for deterministic interpretations. This result has the advantage that, in contrast to I-unambiguity, determinism of interpretations can be easily verified.

Corollary 4.1. Given deterministic nonincreasing interpretations $I_j \in \mathcal{I}(\Xi, \Sigma)$, and regular sets of trajectories $T_j \subseteq \Xi^*$, $j = 1, 2$, it is decidable whether or not (T_1, I_1) and (T_2, I_2) are equivalent.

If $\Xi = \{\sigma\}$ consists of only one symbol, for any nonincreasing interpretation $I \in \mathcal{I}(\Xi, \Sigma)$, all sets of trajectories $T \subseteq \Xi^*$ are "almost" I-unambiguous in the following sense. An equality

$$w_1 \sqcup\!\sqcup_{t_1}^I w_2 = w_1 \sqcup\!\sqcup_{t_2}^I w_2 \tag{13}$$

with $t_1 \neq t_2$ can hold only if

$$(\sigma, \varepsilon, \varepsilon) \in \mathrm{dom}(I) \text{ and } \tau^I(\sigma, \varepsilon, \varepsilon) = \varepsilon. \tag{14}$$

If (14) holds, then always when $w_1 \sqcup\!\sqcup_{t_1}^I w_2$ is defined, the equation (13) holds for all t_2 such that t_1 is a prefix of t_2.

Thus when Ξ consists of only one symbol, in the proof of Theorem 4.2 the multitape automata M_j can be made to accept after reaching the end of the input tapes provided that the sequence of trajectory symbols used is a prefix of some word in the regular language T_j. The latter condition can be easily remembered by the finite state-memory of M_j. This means that for any 3-tuple of words as in (12), M_j has at most one accepting computation.

Corollary 4.2. Let Ξ be a singleton alphabet. For nonincreasing interpretations $I_1, I_2 \in \mathcal{I}(\Xi, \Sigma)$ and regular sets of trajectories $T_1, T_2 \subseteq \Xi^*$ it is decidable whether or not (T_1, I_1) and (T_2, I_2) are equivalent.

In the construction of Theorem 4.1 the trajectory alphabet Ξ needs only five symbols if f uses a unary encoding for pairs of states of the transducer. We do not know whether equivalence of regular sets of trajectories under nonincreasing interpretations remains undecidable when Ξ has cardinality between 2 and 4.

References

[1] J. Berstel: *Transductions and Context-Free Languages.* Teubner, Stuttgart, 1979.

[2] M. Domaratzki: Deletion along trajectories. *Theoretical Computer Science*, 320 (2004), 293–313.

[3] M. Domaratzki: Semantic shuffle on and deletion along trajectories. In *Proc. 8th International Conference on Developments on Language Theory,* DLT'04, LNCS 3340, Springer-Verlag, Berlin, 2004, 163–174.

[4] M. Domaratzki: Trajectory-based codes. *Acta Informatica,* 40 (2004), 491–527.

[5] M. Domaratzki: Trajectory-based embedding relations. *Fundamenta Informatica,* 59 (2004), 349–363.

[6] M. Domaratzki: More words on trajectories. *Bulletin of the EATCS*, 86 (June 2005), 107–145.

[7] M. Domaratzki, A. Mateescu, K. Salomaa, S. Yu: Deletion along trajectories and commutative closure. In *Proc. of 4th International Conference on Combinatorics on Words* (T. Harju, J. Karhumäki, eds.), 2003, 309–319.

[8] M. Domaratzki, K. Salomaa: Decidability of trajectory-based equations. In *Mathematical Foundations of Computer Science,* MFCS'04 (J. Fiala, V. Koubek, J. Kratochvíl, eds.), LNCS 3153, Springer-Verlag, Berlin, 2004, 723–734. Full version to appear in *Theoretical Computer Science.*

[9] M. Domaratzki, K. Salomaa: Codes defined by multiple sets of trajectories. In *Proc. 11th International Conference on Automata and Formal Languages*, AFL'05 (Z. Ésik, Z. Fülöp, eds.), 2005, 97–111.

[10] M. Domaratzki, K. Salomaa: Restricted sets of trajectories and decidability of shuffle decompositions. *International Journal of Foundations of Computer Science,* 16 (2005), 897–912.

[11] T. Harju, J. Karhumäki: The equivalence problem of multitape finite automata. *Theoretical Computer Science,* 78 (1991), 347–355.

[12] T. Harju, J. Karhumäki: Morphisms. In [26], 439–510.

[13] T. Harju, J. Karhumäki, D. Krob: Remarks on generalized Post correspondence problem. In *13th Annual Symposium on Theoretical Aspects of Computer Science,* STACS'96, LNCS 1046, Springer-Verlag, Berlin, 1996, 39–48.

[14] T. Harju, A. Mateescu, A. Salomaa: Shuffle on trajectories: The Schützenberger product and related operations. In *Mathematical Foundations of Computer Science*, MFCS'98 (L. Brim, J. Gruska, J. Zlatuska, eds.), LNCS 1450, Springer-Verlag, Berlin, 1998, 503–511.

[15] J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[16] L. Kari, S. Konstantinidis, P. Sosík: Substitutions on trajectories. In *Theory is Forever* (J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg, eds.), LNCS 3113, Springer-Verlag, Berlin, 2004, 145–158.

[17] L. Kari, S. Konstantinidis, P. Sosík: Substitutions, trajectories and noisy channels. In *Implementation and Application of Automata,* CIAA'04 (M. Domaratzki, A. Okhotin, K. Salomaa, S. Yu, eds.), LNCS 3317, Springer-Verlag, Berlin, 2005, 202–212.

[18] L. Kari, S. Konstantinidis, P. Sosík: Operations on trajectories with applications to coding and bioinformatics. *International Journal of Foundations of Computer Science,* 16 (2005), 531–546.

[19] L. Kari, P. Sosík: Aspects of shuffle and deletion on trajectories. *Theoretical Computer Science,* 332 (2005), 47–61.

[20] E. Kinber: The inclusion problem for some classes of deterministic multitape automata. *Theoretical Computer Science,* 26 (1983), 1–24.

[21] A. Mateescu: CD grammar systems and trajectories. *Acta Cybernetica*, 12 (1997), 141–157.

[22] A. Mateescu: Splicing on routes: a framework of DNA computation. In *Unconventional Models of Computation* (C. Calude, J. Casti, M. Dinneen, eds.), Springer-Verlag, Singapore, 1998, 273–285.

[23] A. Mateescu: Words on trajectories. *Bulletin of the EATCS,* 65 (1998), 118–135.

[24] A. Mateescu, G. Rozenberg, A. Salomaa: Shuffle on trajectories: Syntactic constraints. *Theoretical Computer Science,* 197 (1998), 1–56.

[25] A. Mateescu, K. Salomaa, S. Yu: On fairness of many dimensional trajectories. *J. Automata, Languages and Combinatorics,* 5 (2000), 145–157.

[26] G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages,* Vol. I–III. Springer-Verlag, Berlin, 1997.

[27] S. Yu: Regular languages. In [26], 41–110.