

One-Letter Automata: How to Reduce k Tapes to One

Hristo Ganchev and Stoyan Mihov and Klaus U. Schulz

August 19, 2003

Abstract

The class of n -dimensional regular relations has various closure properties that are interesting for practical applications. From a computational point of view, each closure operation may be realized with a corresponding construction for k -tape finite state automata. While the constructions for union, Kleene-star and (coordinate-wise) concatenation are simple, specific and non-trivial algorithms are needed for relational operations such as composition, projection, and cartesian product. Here we show that all these operations for k -tape automata can be represented and computed using standard operations on conventional one-tape finite state automata plus some trivial rules for tape manipulation. As a key notion we introduce the concept of a one-letter k -tape automaton, which yields a bridge between k -tape and one-tape automata. We achieve a general and efficient implementational framework for n -tape automata.

1 Introduction

Multi-tape finite state automata and especially 2-tape automata have been widely used in many areas of computer science such as Natural Language Processing [5, 8, 14] and Speech Processing [9, 11]. They provide an uniform, clear and computationally efficient framework for dictionary representation [3, 7] and realization of rewrite rules [1, 6, 4], as well as text tokenization, lexicon tagging, part-of-speech disambiguation, indexing, filtering and many other text processing tasks [5, 8, 13, 14]. The properties of k -tape finite state automata differ significantly from the corresponding properties of 1-tape automata. For example, for $k \geq 2$ the class of relations recognized by k -tape automata is not closed under intersection and complement. Moreover there is no general determinization procedure for k -tape automata. On the other side the class of relations recognized by k -tape finite state automata is closed under a number of useful relational operations like composition, cartesian product, projection, inverse etc. This latter property makes k -tape automata interesting for many practical applications.

There exist a number of implementations for k -tape finite state automata [5, 10, 12]. Most of them are implementing the 2-tape case only. While it is straightforward to realize constructions for k -tape automata that yield union, Kleene-star and concatenation of the recognized relations, the computation of relational operations such as composition, projection and cartesian product is a complex task. This makes the use of the k -tape automata framework tedious and difficult.

We introduce an approach for presenting all relevant operations for n -tape automata using standard operations for classical 1-tape automata plus some straightforward operations for adding, deleting and permuting tapes. In this way we obtain a transparent, general and efficient framework for implementing k -tape automata.

The main idea is to consider a restricted form of k -tape automata where all transition labels have exactly one non-empty component representing a single letter. The set of all k -tuples of this form represents the basis of the monoid of k -tuples of words together with the coordinate-wise concatenation. We call this kind of automata *one-letter automata*. Treating the basis elements as symbols of a derived alphabet, one-letter automata can be considered as conventional 1-tape automata. This gives rise to a correspondence where standard operations for 1-tape automata may be used to replace complex operations for k -tape automata.

The paper is structured as follows. Section 2 provides some formal background. In Section 3 we introduce one-letter k -tape automata. We show that classical algorithms for union, concatenation and Kleene-star over one-letter automata (considered as 1-tape automata) are correct if the result is interpreted as a k -tape automaton. Section 4 is central. A condition is given that guarantees that the intersection of two k -dimensional regular relations is again regular. For k -tape one-letter automata of a specific form that reflects this condition, any classical algorithm for intersecting the associated 1-tape automata can be used for computing the intersection of the regular relations recognized by the automata. Section 5 shows how to implement tape permutations for one-letter automata. Using tape permutations, the inverse relation to a given k -dimensional regular relation can be realized. In a similar way, Section 6 treats tape insertion, tape deletion and projection operations for k -dimensional regular relations. Section 7 shows how to reduce the computation of composition and cartesian product of regular relations to intersections of the kind discussed in Section 4 plus tape insertion and projection. In Section 8 we add some final remarks. We comment on problems that may arise when using k -tape automata and on possible solutions.

2 Formal Background

We assume that the reader is familiar with standard notions from automata theory (see, e.g., [2, 13]). In the sequel, with Σ we denote a finite set of symbols called the *alphabet*, ϵ denotes the empty word, and $\Sigma^\epsilon := \Sigma \cup \{\epsilon\}$. The length of a word $w \in \Sigma^*$ is written $|w|$. If $L_1, L_2 \subseteq \Sigma^*$ are languages, then

$$L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

denotes their *concatenation*. Here $w_1 \cdot w_2$ is the usual concatenation of words. Recall that $\langle \Sigma^*, \cdot, \epsilon \rangle$ is the free monoid with set of generators Σ .

If $v = \langle v_1, \dots, v_k \rangle$ and $w = \langle w_1, \dots, w_k \rangle$ are two k -tuples of words, then

$$v \odot w := \langle v_1 \cdot w_1, \dots, v_k \cdot w_k \rangle$$

denotes the *coordinate-wise concatenation*. With $\hat{\epsilon}$ we denote the k -tuple $\langle \epsilon, \dots, \epsilon \rangle$. The tuple $\langle (\Sigma^*)^k, \odot, \hat{\epsilon} \rangle$ is a monoid that can be described as the k -fold cartesian product of the free monoid $\langle \Sigma^*, \cdot, \epsilon \rangle$. As set of generators we consider

$$\hat{\Sigma}_k := \{ \langle \epsilon, \dots, \underset{i}{a}, \dots, \epsilon \rangle \mid 1 \leq i \leq k, a \in \Sigma \}.$$

Note that the latter monoid is not free, due to obvious commutation rules for generators. For relations $R \subseteq (\Sigma^*)^k$ we define

$$\begin{aligned} R^0 &:= \{ \hat{\epsilon} \}, \\ R^{i+1} &:= R^i \odot R, \\ R^* &:= \bigcup_{i=0}^{\infty} R^i \quad (\text{Kleene-star}). \end{aligned}$$

Let $k \geq 2$ and $1 \leq i \leq k$. The relation

$$R \ominus (i) := \{\langle w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_k \rangle \mid \exists v \in \Sigma^* : \langle w_1, \dots, w_{i-1}, v, w_{i+1}, \dots, w_k \rangle \in R\}$$

is called the *projection of R to the set of coordinates $\{1, \dots, i-1, i+1, \dots, k\}$* . If $R_1, R_2 \subseteq (\Sigma^*)^k$ are two relations of the same arity, then

$$R_1 \odot R_2 := \{v \odot w \mid v \in R_1, w \in R_2\}$$

denotes the *coordinate-wise concatenation*. If $R_1 \subseteq \Sigma^{*k}$ and $R_2 \subseteq \Sigma^{*l}$ are two relations, then

$$R_1 \times R_2 := \{\langle w_1, \dots, w_{k+l} \rangle \mid \langle w_1, \dots, w_k \rangle \in R_1, \langle w_{k+1}, \dots, w_{k+l} \rangle \in R_2\}$$

is the *cartesian product* of R_1 and R_2 and

$$R_1 \circ R_2 := \{\langle w_1, \dots, w_{k+l-2} \rangle \mid \exists w : \langle w_1, \dots, w_{k-1}, w \rangle \in R_1, \langle w, w_k, \dots, w_{k+l-2} \rangle \in R_2\}$$

is the *composition* of R_1 and R_2 . Further well-known operations for relations are union, intersection, and inversion ($k = 2$).

Definition 2.1 *The class of k -dimensional regular relations over the alphabet Σ is recursively defined in the following way:*

- \emptyset and $\{v\}$ for all $v \in \hat{\Sigma}_k$ are k -dimensional regular relations.
- If R_1, R_2 and R are k -dimensional regular relations, then so are
 - $R_1 \odot R_2$,
 - $R_1 \cup R_2$,
 - R^* .
- There are no other k -dimensional regular relations.

Note 2.2 *The class of k -dimensional regular relations over a given alphabet Σ is closed under union, Kleene-star, coordinate-wise concatenation, composition, projection, and cartesian product. For $k \geq 2$ the class of regular relations is not closed under intersection, difference and complement. Obviously, every 1-dimensional regular relation is a regular language over the alphabet Σ . Hence, for $k = 1$ we obtain closure under intersection, difference and complement.*

Definition 2.3 *Let k be a positive integer. A k -tape automaton is a six-tuple $A = \langle k, \Sigma, S, F, s_0, E \rangle$, where Σ is an alphabet, S is a finite set of states, $F \subseteq S$ is a set of final states, $s_0 \in S$ is the initial state and $E \subseteq S \times (\Sigma^\epsilon)^k \times S$ is a finite set of transitions. A sequence $s_0, a_1, s_1, \dots, s_{n-1}, a_n, s_n$, where s_0 is the initial state, $s_i \in S$ and $a_i \in (\Sigma^\epsilon)^k$ for $i = 1, \dots, n$, is a path for A iff $\langle s_{i-1}, a_i, s_i \rangle \in E$ for $1 \leq i < n$. The k -tape automaton A recognizes $v \in (\Sigma^*)^k$ iff there exists a path $s_0, a_1, s_1, \dots, s_{n-1}, a_n, s_n$ for A such that $s_n \in F$ and $v = a_1 \odot a_2 \dots a_{n-1} \odot a_n$. With $R(A)$ we denote the set of all tuples in $(\Sigma^*)^k$ recognized by A , i.e., $R(A) := \{v \in (\Sigma^*)^k \mid A \text{ recognizes } v\}$.*

For a given k -tape automaton $A = \langle k, \Sigma, S, F, s_0, E \rangle$ the *generalized transition relation* $E^* \subseteq S \times (\Sigma^*)^k \times S$ is recursively defined as follows:

1. $\langle s, \langle \epsilon, \dots, \epsilon \rangle, s \rangle \in E^*$ for all $s \in S$,
2. if $\langle s_1, v, s' \rangle \in E^*$ and $\langle s', a, s_2 \rangle \in E$, then $\langle s_1, v \odot a, s_2 \rangle \in E^*$, for all $v \in (\Sigma^*)^k$, $a \in (\Sigma^\epsilon)^k$, $s_1, s', s_2 \in S$.

Clearly, if A is a k -tape automaton, then $R(A) = \{v \in (\Sigma^*)^k \mid \exists f \in F : \langle s_0, v, f \rangle \in E^*\}$.

Note 2.4 *By a well-known generalization of Kleene's Theorem (see [6]), for each k -tape automaton A the set $R(A)$ is a k -dimensional regular relation, and for every k -dimensional regular relation R' , there exists a k -tape automaton A' such that $R(A') = R'$.*

3 Basic Operations for one-letter automata

In this section we introduce the concept of a one-letter automaton. One-letter automata represent a special form of k -tape automata that can be naturally interpreted as one-tape automata over the alphabet $\hat{\Sigma}_k$. We show that basic operations such as union, concatenation, and Kleene-star for one-letter automata can be realized using the corresponding standard constructions for conventional one-tape automata.

Definition 3.1 A k -tape finite state automaton $A = \langle k, \Sigma, S, F, s_0, E \rangle$ is a one-letter automaton iff all transitions $e \in E$ are of the form

$$e = \langle s, \langle \epsilon, \dots, \underset{\uparrow_{i-1}}{\epsilon}, \underset{\uparrow_i}{a}, \underset{\uparrow_{i+1}}{\epsilon}, \dots, \underset{\uparrow_k}{\epsilon} \rangle, s' \rangle$$

for some $1 \leq i \leq k$ and $a \in \Sigma$.

Proposition 3.2 For every k -tape automaton A we may effectively construct a k -tape one-letter automaton A' such that $R(A') = R(A)$.

Proof. First we can apply the classical ϵ -removal procedure in order to construct an $\hat{\epsilon}$ free k -tape automaton, which leaves the recognized relation unchanged. Let $\hat{A} = \langle k, \Sigma, S, F, s_0, E \rangle$ be an $\hat{\epsilon}$ -free k -tape automaton such that $R(A) = R(\hat{A})$. Then we construct $A' = \langle k, \Sigma, S', F, s_0, E' \rangle$ using the following algorithm:

$S' = S, E' = \emptyset$

FOR $s \in S$ DO:

FOR $\langle s, \langle a_1, a_2, \dots, a_k \rangle, s' \rangle \in E$ DO

LET $I = \{i \in \mathbb{N} \mid a_i \neq \epsilon\}$ ($I = \{i_1, \dots, i_t\}$);

LET $S'' = \{s_{i_1}, \dots, s_{i_{t-1}}\}$, SUCH THAT $S'' \cap S' = \emptyset$;

$S' = S' \cup S''$;

$E' = E' \cup \{ \langle s_{i_j}, \langle \epsilon, \dots, \epsilon, \underset{\uparrow_j}{a_{i_j}}, \epsilon, \dots, \epsilon \rangle, s_{i_{j+1}} \rangle \mid 0 \leq j \leq t-1, s_{i_0} = s \text{ and } s_{i_t} =$

$s' \rangle$;

END;

END.

Informally speaking, we split each transition with label $\langle a_1, a_2, \dots, a_k \rangle$ with $t > 1$ non-empty coordinates into t subtransitions, introducing $t-1$ new intermediate states. \square

Corollary 3.3 If $R \subseteq (\Sigma^*)^k$ is a k -dimensional regular relation, then there exists a k -tape one-letter automaton A such that $R(A) = R$.

Each k -tape one-letter automaton A over the alphabet Σ can be considered as a one-tape automaton (denoted by \hat{A}) over the alphabet $\hat{\Sigma}_k$. Conversely, every ϵ -free one-tape automaton over the alphabet $\hat{\Sigma}_k$ can be considered as a k -tape automaton over Σ . Formally, this correspondence can be described using two mappings.

Definition 3.4 The mapping $\hat{\cdot}$ maps every k -tape one-letter automaton $A = \langle k, \Sigma, S, F, s_0, E \rangle$ to the ϵ -free one-tape automaton $\hat{A} := \langle \hat{\Sigma}_k, S, F, s_0, E \rangle$. The mapping $\check{\cdot}$ maps a given ϵ -free one-tape automaton $A' = \langle \hat{\Sigma}_k, S, F, s_0, E \rangle$ to the k -tape one-letter automaton $\check{A}' := \langle k, \Sigma, S, F, s_0, E \rangle$.

Obviously, the mappings $\hat{\cdot}$ and $\check{\cdot}$ are inverse. From a computational point of view, the mappings merely represent a conceptual shift where we use another alphabet for looking at transitions labels. States and transitions are not changed.

Definition 3.5 *The mapping*

$$\phi : \hat{\Sigma}_k^* \rightarrow \Sigma^{*k} : a_1 \cdots a_n \mapsto a_1 \odot \cdots \odot a_n, \epsilon \mapsto \hat{\epsilon}$$

is called the natural homomorphism between the free monoid $(\hat{\Sigma}_k^*, \cdot, \epsilon)$ and the monoid $(\Sigma^{*k}, \odot, \hat{\epsilon})$.

It is trivial to check that ϕ is in fact a homomorphism. We have the following connection between the mappings $\hat{\cdot}$, $\check{\cdot}$ and ϕ .

Lemma 3.6 *Let $A = \langle k, \Sigma, S, F, s_0, E \rangle$ be a k -tape one-letter automaton. Then*

1. $\check{\hat{A}} = A$.
2. $R(A) = \phi(L(\hat{A}))$.

Furthermore, if A' is an ϵ -free one-tape automaton over $\hat{\Sigma}_k$, then $\hat{A}' = A'$.

Thus we obtain the following commutative diagram:

$$\begin{array}{ccc} A & \xrightarrow{\check{\cdot}} & \hat{A} \\ R \downarrow & & \downarrow L \\ R(A) & \xleftarrow{\phi} & L(\hat{A}) \end{array}$$

We get the following proposition as a direct consequence of Lemma 3.6 and the homomorphic properties of the mapping ϕ .

Proposition 3.7 *Let A_1 and A_2 be two k -tape one-letter automata. Then we have the following:*

1. $R(A_1) \cup R(A_2) = \phi(L(\hat{A}_1) \cup L(\hat{A}_2))$.
2. $R(A_1) \odot R(A_2) = \phi(L(\hat{A}_1) \cdot L(\hat{A}_2))$.
3. $R(A_1)^* = \phi(L(\hat{A}_1)^*)$.

Algorithmic constructions. From Part 1 of Proposition 3.7 we see the following. Let A_1 and A_2 be two k -tape one-letter automata. Then, to construct a one-letter automaton A such that $R(A) = R(A_1) \cup R(A_2)$ we may interpret A_i as a one-tape automaton \hat{A}_i ($i = 1, 2$). We use any union-construction for one-tape automata, yielding an automaton A' such that $L(A') = L(\hat{A}_1) \cup L(\hat{A}_2)$. Removing ϵ -transitions and interpreting the resulting automaton A'' as a k -tape automaton $A := \check{A}''$ we receive a one-letter automaton such that $R(A) = R(A_1) \cup R(A_2)$. Similarly Parts 2 and 3 show that “classical” algorithms for closing conventional one-tape automata under concatenation and Kleene-star can be directly applied to k -tape one-letter automata, yielding algorithms for closing k -tape one-letter automata under concatenation and Kleene-star.

4 Intersection of one-letter automata

It is well-known that the intersection of two k -dimensional regular relations is not necessarily a regular relation. For example, the relations $R_1 = \{\langle a^n b^k, c^n \rangle \mid n, k \in \mathbb{N}\}$ and $R_2 = \{\langle a^s b^n, c^n \rangle \mid s, n \in \mathbb{N}\}$ are regular, but $R_1 \cap R_2 = \{\langle a^n b^n, c^n \rangle \mid n \in \mathbb{N}\}$ is not regular since its first projection is not a regular language. We now introduce a condition that guarantees that the classical construction for intersecting one-tape automata is correct if used for k -tape one-letter automata. As a corollary we obtain a condition for the regularity of the intersection of two k -dimensional regular relations. This observation will be used later for explicit constructions that yield composition and cartesian product of one-letter automata. A few preparations are needed.

Definition 4.1 Let $v = b_1 \dots b_n$ be an arbitrary word over the alphabet Ξ , i.e., $v \in \Xi^*$. We say that the word v' is obtained from v by adding the letter b iff $v' = b_1 \dots b_j b b_{j+1} \dots b_n$ for some $0 \leq j \leq n$. In this case we also say that v is obtained from v' by deleting the symbol b .

Proposition 4.2 Let $v = a_1 \dots a_n \in \hat{\Sigma}_k^*$ and $\phi(v) = a_1 \odot a_2 \odot \dots \odot a_n = \langle w_1, \dots, w_k \rangle$. Let also $a = \langle \epsilon, \dots, \underset{i}{b}, \dots, \epsilon \rangle \in \hat{\Sigma}_k$. Then, if v' is obtained from v by adding the letter a , then $\phi(v') = \langle w_1, \dots, w_{i-1}, w'_i, w_{i+1}, \dots, w_k \rangle$ and w'_i is obtained from w_i by adding the letter b .

Definition 4.3 For a regular relation $R \subseteq (\Sigma^*)^k$ the coordinate i ($1 \leq i \leq k$) is inessential iff for all $\langle w_1, \dots, w_k \rangle \in R$ and any $v \in \Sigma^*$ we have

$$\langle w_1, \dots, w_{i-1}, v, w_{i+1}, \dots, w_k \rangle \in R.$$

Analogously, if A is a k -tape automaton such that $R(A) = R$ we say that tape i of A is inessential. Otherwise we call coordinate (tape) i essential.

Definition 4.4 Let A be a k -tape one-letter automaton and assume that each coordinate in the set $I \subseteq \{1, \dots, k\}$ is inessential for $R(A)$. Then A is in normal form w.r.t. I iff for any tape $i \in I$ we have:

1. $\forall s \in S, \forall a \in \Sigma : \langle s, \langle \epsilon, \dots, \underset{i}{a}, \dots, \epsilon \rangle, s \rangle \in E$,
2. $\forall s, s' \in S, \forall a \in \Sigma : (s' \neq s) \Rightarrow \langle s, \langle \underset{1}{\epsilon}, \dots, \underset{i}{a}, \dots, \underset{k}{\epsilon} \rangle, s' \rangle \notin E$.

Proposition 4.5 For any k -tape automaton A and any given set I of inessential coordinates of $R(A)$ we may effectively construct a k -tape one-letter automaton A' in normal form w.r.t. I such that $R(A') = R(A)$.

Proof. Let $A = \langle k, \Sigma, S, F, s_0, E \rangle$. Without loss of generality we can assume that A is in one-letter form (Proposition 3.2). To construct $A' = \langle k, \Sigma, S, F, s_0, E' \rangle$ we use the following algorithm:

```

E' = E
FOR s ∈ S DO
  FOR i ∈ I DO
    FOR a ∈ Σ DO
      IF ((⟨s, ⟨ε, ..., ε,  $\underset{i}{a}$ , ε, ..., ε⟩, s'⟩ ∈ E') & (s' ≠ s)) THEN
        E' = E' \ {⟨s, ⟨ε, ..., ε,  $\underset{i}{a}$ , ε, ..., ε⟩, s'⟩};
        E' = E' ∪ ⟨s, ⟨ε, ..., ε,  $\underset{i}{a}$ , ε, ..., ε⟩, s⟩;
      END;
    END;
  END;
END.
```

The algorithm does not change any transition on an essential tape. Transitions between distinct states that affect an inessential tape in I are erased. For each state we add loops with all symbols from the alphabet for the inessential tapes in I . The correctness of the above algorithm follows from the fact that for any inessential tape $i \in I$ we have $\langle w_1, \dots, w_i, \dots, w_n \rangle \in R(A)$ iff $\langle w_1, \dots, \epsilon, \dots, w_n \rangle \in R(A)$. \square

Corollary 4.6 *Let $R \subseteq (\Sigma^*)^k$ be a regular relation with a set I of inessential coordinates. Then there exists a k -tape one-letter automaton A in normal form w.r.t. I such that $R(A) = R$.*

The following property of k -tape automata in normal form will be useful when proving Lemma 4.8.

Proposition 4.7 *Let $A = \langle k, \Sigma, S, F, s_0, E \rangle$ be a k -tape one-letter automaton in normal form w.r.t. the set of inessential coordinates I . Let $i_0 \in I$ and let $v = a_1 \dots a_n \in L(\hat{A})$. Then for any $a = \langle \epsilon, \dots, \underset{i_0}{\uparrow} b, \dots, \epsilon \rangle \in \hat{\Sigma}_k$ and any word $v' \in \hat{\Sigma}_k^*$ obtained from v by adding a we have $v' \in L(\hat{A})$.*

Proof. The condition for the automaton A to be in normal form w.r.t. I yields that for all $s \in S$ the transition $\langle s, a, s \rangle$ is in E , which proves the proposition. \square

Now we are ready to formulate and prove the following sufficient condition for the regularity of the intersection of two regular relations. With K we denote the set of coordinates $\{1, \dots, k\}$.

Lemma 4.8 *For $i = 1, 2$, let A_i be a k -tape one-letter automaton, let $I_i \subseteq K$ denote a given set of inessential coordinates for A_i . Let A_i be in normal form w.r.t. I_i ($i = 1, 2$). Assume that $|K \setminus (I_1 \cup I_2)| \leq 1$, which means that there exists at most one common essential tape for A_1 and A_2 . Then $R(A_1) \cap R(A_2)$ is a regular k -dimensional relation. Moreover $R(A_1) \cap R(A_2) = \phi(L(\hat{A}_1) \cap L(\hat{A}_2))$.*

Proof. It is obvious that $\phi(L(\hat{A}_1) \cap L(\hat{A}_2)) \subseteq R(A_1) \cap R(A_2)$, because if $a_1 \dots a_n \in L(\hat{A}_1) \cap L(\hat{A}_2)$, then by Lemma 3.6 we have $a_1 \odot \dots \odot a_n \in R(A_1) \cap R(A_2)$. We give a detailed proof for the other direction, showing that

$$R(A_1) \cap R(A_2) \subseteq \phi(L(\hat{A}_1) \cap L(\hat{A}_2)).$$

For the proof the reader should keep in mind that the transition labels of the automata A_i ($i = 1, 2$) are elements of $\hat{\Sigma}_k$, which means that the sum of the lengths of the words representing the components is exactly 1.

Let $\langle w_1, w_2, \dots, w_k \rangle \in R(A_1) \cap R(A_2)$. Let $j_0 \in K$ be a coordinate such that for each $j_0 \neq j \in K$ we have $j \in I_1$ or $j \in I_2$. Let $E_1 = K \setminus I_1$. Recall that for $i \in E_1$, $i \neq j_0$ always $i \in I_2$ is an inessential tape for A_2 . Then by the definition of inessential tapes the tuples $\langle w'_1, \dots, w'_k \rangle$ and $\langle w''_1, \dots, w''_k \rangle$, where

$$w'_i = \begin{cases} \epsilon, & \text{if } i \in I_1 \\ w_i, & \text{if } i \in E_1 \end{cases} \quad w''_i = \begin{cases} \epsilon, & \text{if } i \in E_1 \text{ and } i \neq j_0 \\ w_i, & \text{otherwise} \end{cases}$$

respectively are in $R(A_1)$ and $R(A_2)$. Then there are words $v' = a'_1 \dots a'_n \in L(\hat{A}_1)$ and $v'' = a''_1 \dots a''_m \in L(\hat{A}_2)$ such that $\phi(v') = \langle w'_1, \dots, w'_k \rangle$ and $\phi(v'') = \langle w''_1, \dots, w''_k \rangle$. Note that $n = \sum_{i=1}^k |w'_i|$ and $m = \sum_{i=1}^k |w''_i|$. Furthermore, $w_{j_0} = w'_{j_0} = w''_{j_0}$. Let $l = |w_{j_0}|$.

We now construct a word $a_1 a_2 \dots a_r \in L(\hat{A}_1) \cap L(\hat{A}_2)$ such that $a_1 \odot a_2 \odot \dots \odot a_r = \langle w_1, \dots, w_k \rangle$, which imposes that $r = n + m - l$. Each letter a_i is obtained copying a suitable letter from one of the sequences $a'_1 \dots a'_n$ and $a''_1 \dots a''_m$. In order to control the selection, we use the pair of indices t'_i, t''_i ($0 \leq i < n + m - l$), which can be considered as pointers to the two sequences. The definition of t'_i, t''_i and a_i proceeds inductively in the following way. Let $t'_0 = t''_0 := 1$. Assume that t'_i and t''_i are defined for some $0 \leq i < n + m - l$. We show how to define a_{i+1} and the indices t'_{i+1} and t''_{i+1} . We distinguish four cases:

1. if $t'_i = n + 1$ and $t''_i = m + 1$ we stop; else

2. if $a'_{t'_i} = \langle \epsilon, \dots, b, \dots, \epsilon \rangle$ for some $j \neq j_0$, then $a_{i+1} := a'_{t'_i}$, $t'_{i+1} := t'_i + 1$, $t''_{i+1} := t''_i$,
3. if $a'_{t'_i} = \langle \epsilon, \dots, b, \dots, \epsilon \rangle$ or $t'_i = n + 1$, and $a''_{t''_i} = \langle \epsilon, \dots, c, \dots, \epsilon \rangle$ for some $j \neq j_0$, then $a_{i+1} := a''_{t''_i}$, $t'_{i+1} := t'_i$, and $t''_{i+1} := t''_i + 1$.
4. if $a'_{t'_i} = a''_{t''_i} = \langle \epsilon, \dots, b, \dots, \epsilon \rangle$ for some $b \in \Sigma$, then $a_{i+1} := a'_{t'_i}$, $t'_{i+1} := t'_i + 1$ and $t''_{i+1} := t''_i + 1$.

From an intuitive point of view, the definition yields a kind of zig-zag construction. We always proceed in one sequence until we come to a transition that affects coordinate j_0 . At this point we continue using the other sequence. Once we have in both sequences a transition that affects j_0 , we enlarge both indices. From $w'_{j_0} = w''_{j_0} = w_{j_0}$ it follows immediately that the recursive definition stops exactly when $i + 1 = n + m - l$. In fact the subsequences of $a'_1 \dots a'_n$ and $a''_1 \dots a''_m$ from which w_{j_0} is obtained must be identical.

Using induction on $0 \leq i \leq n + m - l$ we now prove that the word $a_1 \dots a_i$ is obtained from $a'_1 \dots a'_{t'_i-1}$ by adding letters in $\hat{\Sigma}_k$ which have a non- ϵ symbol in an inessential coordinate for $R(A_1)$. The base of the induction is obvious. Let the statement be true for some $0 \leq i < n + m - l$. We prove it for $i + 1$:

The word $a_1 \dots a_i a_{i+1}$ is obtained from $a_1 \dots a_i$ by adding the letter $a_{i+1} = \langle \epsilon, \dots, b, \dots, \epsilon \rangle$,

and according to the induction hypothesis $a_1 \dots a_i$ is obtained from $a'_1, \dots, a'_{t'_i-1}$ by adding letters with a non- ϵ symbol in a coordinate in I_1 . If $j \in E_1$ (Cases 2 and 4), then $a_{i+1} = a'_{t'_i}$, $t'_{i+1} = t'_i + 1$ and $t'_{i+1} - 1 = t'_i$, hence $a_1 \dots a_i a_{i+1}$ is obtained from $a'_1, \dots, a'_{t'_i-1} a'_{t'_{i+1}-1}$ by adding letters satisfying the above condition. On the other side, if $j \in I_1$ (Case 3) we have $a_{i+1} = a''_{t''_i}$ and $t'_{i+1} := t'_i$, which means that $a'_1 \dots a'_{t'_{i+1}-1} = a'_1 \dots a'_{t'_i-1}$ and a_{i+1} is a letter satisfying the condition. Thus $a_1 \dots a_i a_{i+1}$ is obtained from $a'_1 \dots a'_{t'_{i+1}-1}$ adding letters which have non- ϵ symbol on an inessential tape for A_1 , which means that the statement is true for $i + 1$.

Analogously we prove for $0 \leq i \leq n + m - l$ that $a_1 \dots a_i$ is obtained from $a''_1 \dots a''_{t''_i-1}$ by adding letters in $\hat{\Sigma}_k$ which have a non- ϵ symbol in an inessential coordinate for $R(A_2)$.

By Proposition 4.7, $a_1 \dots a_{n+m-l} \in L(\hat{A}_1)$ and $a_1 \dots a_{n+m-l} \in L(\hat{A}_2)$. From Proposition 4.2 we obtain that $a_1 \odot \dots \odot a_{n+m-l} = \langle u_1, \dots, u_k \rangle$, where $u_i = w'_i$ if $i \in E_1$ and $u_i = w''_i$ otherwise. But now remembering the definition of w'_i and w''_i we obtain that $a_1 \odot \dots \odot a_{n+m-l} = \langle w_1, \dots, w_k \rangle$, which we had to prove. \square

Corollary 4.9 *If $R_1 \subseteq (\Sigma)^k$ and $R_2 \subseteq (\Sigma)^k$ are two k -dimensional regular relations with at most one common essential coordinate i ($1 \leq i \leq k$), then $R_1 \cap R_2$ is a k -dimensional regular relation.*

Algorithmic construction. From Lemma 4.8 we see the following. Let A_1 and A_2 be two k -tape one-letter automata with at most one common essential tape i . Assume that both automata are in normal form w.r.t. the sets of inessential tapes. Then the relation $R(A_1) \cap R(A_2)$ is recognized by any ϵ -free 1-tape automaton A' accepting $L(\hat{A}_1) \cap L(\hat{A}_2)$, treating A' as a k -tape one-letter automaton $A = \hat{A}'$.

5 Tape permutation and inversion for one-letter automata

In the sequel, let S_k denote the symmetric group of k elements.

Definition 5.1 Let $R \subseteq (\Sigma^*)^k$ be a regular relation, let $\sigma \in S_k$. The permutation of coordinates induced by σ , $\sigma(R)$, is defined as

$$\sigma(R) := \{ \langle w_{\sigma^{-1}(1)}, \dots, \underset{\sigma(i)}{\uparrow} w_i, \dots, w_{\sigma^{-1}(k)} \rangle \mid \langle w_1, \dots, w_k \rangle \in R \}.$$

Proposition 5.2 For a given k -tape one-letter automaton $A = \langle k, \Sigma, S, F, s_0, E \rangle$, let $\sigma(A) := \langle k, \Sigma, S, F, s_0, \sigma(E) \rangle$ where

$$\sigma(E) := \{ \langle s, \langle \epsilon, \dots, \underset{\sigma(i)}{\uparrow} a_i, \dots, \epsilon \rangle, s' \rangle \mid \langle s, \langle \epsilon, \dots, \underset{i}{\uparrow} a_i, \dots, \epsilon \rangle, s' \rangle \in E \}.$$

Then $R(\sigma(A)) = \sigma(R(A))$.

Proof. Using induction over the construction of E^* and $\sigma(E)^*$ we prove that for all $s' \in S$ and $\langle w_1, \dots, w_k \rangle \in (\Sigma^*)^k$ we have

$$\begin{aligned} & \langle s_0, \langle w_1, \dots, w_k \rangle, s' \rangle \in E^* \\ \Leftrightarrow & \langle s_0, \langle w_{\sigma^{-1}(1)}, \dots, \underset{\sigma(i)}{\uparrow} w_i, \dots, w_{\sigma^{-1}(k)} \rangle, s' \rangle \in \sigma(E)^*. \end{aligned}$$

“ \Rightarrow ”. The base of the induction is obvious since $\langle s_0, \langle \epsilon, \dots, \epsilon \rangle, s_0 \rangle \in E^* \cap \sigma(E)^*$. Now suppose that there are transitions

$$\begin{aligned} & \langle s_0, \langle w_1, \dots, w_{i-1}, w'_i, w_{i+1}, \dots, w_k \rangle, s \rangle \in E^* \\ & \langle s, \langle \epsilon, \dots, \underset{i}{\uparrow} a_i, \dots, \epsilon \rangle, s' \rangle \in E. \end{aligned}$$

Then, by induction hypothesis, $\langle s_0, \langle w_{\sigma^{-1}(1)}, \dots, \underset{\sigma(i)}{\uparrow} w'_i, \dots, w_{\sigma^{-1}(k)} \rangle, s \rangle \in \sigma(E)^*$. The definition of $\sigma(E)$ shows that $\langle s, \langle \epsilon, \dots, \underset{\sigma(i)}{\uparrow} a_i, \dots, \epsilon \rangle, s' \rangle \in \sigma(E)$. Hence

$$\langle s_0, \langle w_{\sigma^{-1}(1)}, \dots, \underset{\sigma(i)}{\uparrow} w'_i a_i, \dots, w_{\sigma^{-1}(k)} \rangle, s' \rangle \in \sigma(E)^*,$$

which we had to prove.

“ \Leftarrow ”. Follows analogously to “ \Rightarrow ”. □

Corollary 5.3 Let $R \subseteq (\Sigma^*)^k$ be a k -dimensional regular relation and $\sigma \in S_k$. Then also $\sigma(R)$ is a k -dimensional regular relation.

Algorithmic construction. From Proposition 5.2 we see the following. Let A be a 2-tape one-letter automaton. If σ denotes the transposition $(1, 2)$, then automaton $\sigma(A)$ defined as above recognizes the relation $R(A)^{-1}$.

6 Tape insertion, tape deletion and projection

Definition 6.1 Let $R \subseteq (\Sigma^*)^k$ be a k -dimensional regular relation. We define the insertion of an inessential coordinate at position i (denoted $R \oplus (i)$) as

$$R \oplus (i) := \{ \langle w_1, \dots, w_{i-1}, v, w_i, \dots, w_k \rangle \mid \langle w_1, \dots, w_{i-1}, w_i, \dots, w_k \rangle \in R, v \in \Sigma^* \}.$$

Proposition 6.2 Let $A = \langle k, \Sigma, S, F, s_0, E \rangle$ be a k -tape one-letter automaton. Let $A' := \langle k+1, \Sigma, S, F, s_0, E' \rangle$ where

$$E' := \begin{aligned} & \{ \langle s, \langle \epsilon, \dots, \underset{\uparrow}{a}, \dots, \epsilon, \underset{\uparrow}{\epsilon} \rangle, s' \rangle \mid \langle s, \langle \epsilon, \dots, \underset{\uparrow}{a}, \dots, \underset{\uparrow}{\epsilon} \rangle, s' \rangle \in E \} \\ & \cup \{ \langle s, \langle \epsilon, \dots, \epsilon, \underset{\uparrow}{a} \rangle, s \rangle \mid s \in S, a \in \Sigma \}. \end{aligned}$$

Then $R(A') = R(A) \oplus (k+1)$.

Proof. First, using induction on the construction of E'^* we prove that for all $s \in S$ and $\langle w_1, \dots, w_k, w_{k+1} \rangle \in (\Sigma^*)^{k+1}$ we have

$$\langle s_0, \langle w_1, \dots, w_k, w_{k+1} \rangle, s' \rangle \in E'^* \Leftrightarrow \langle s_0, \langle w_1, \dots, w_k, \epsilon \rangle, s' \rangle \in E'^*.$$

“ \Rightarrow ”. The base of the induction is obvious. Assume there are transitions

$$\begin{aligned} & \langle s_0, \langle w_1, \dots, w_{i-1}, w'_i, w_{i+1}, \dots, w_k, w_{k+1} \rangle, s \rangle \in E'^* \\ & \langle s, \langle \epsilon, \dots, \underset{\uparrow}{a}, \dots, \epsilon \rangle, s' \rangle \in E'. \end{aligned}$$

First assume that $i \leq k$. By induction hypothesis,

$$\langle s_0, \langle w_1, \dots, w_{i-1}, w'_i, w_{i+1}, \dots, w_k, \epsilon \rangle, s \rangle \in E'^*.$$

Using the definition of E'^* we obtain

$$\langle s_0, \langle w_1, \dots, w_{i-1}, w'_i a, w_{i+1}, \dots, w_k, \epsilon \rangle, s' \rangle \in E'^*.$$

If $i = k+1$, then $s = s'$. We may directly use the induction hypothesis to obtain $\langle s_0, \langle w_1, \dots, w_k, \epsilon \rangle, s' \rangle \in E'^*$.

“ \Leftarrow ”. Let $\langle s_0, \langle w_1, \dots, w_k, \epsilon \rangle, s' \rangle \in E'^*$. Let $w_{k+1} = v_1 \dots v_n \in \Sigma^*$ where $v_i \in \Sigma$. The definition of E' shows that for all v_i ($1 \leq i \leq n$) there exists a transition $\langle s', \langle \epsilon, \dots, \epsilon, \underset{\uparrow}{v_i} \rangle, s' \rangle \in E'$.

Hence $\langle s_0, \langle w_1, \dots, w_k, w_{k+1} \rangle, s' \rangle \in E'^*$.

To finish the proof observe that the definition of E' yields

$$\langle s_0, \langle w_1, \dots, w_k \rangle, s' \rangle \in E'^* \Leftrightarrow \langle s_0, \langle w_1, \dots, w_k, \epsilon \rangle, s' \rangle \in E'^*.$$

□

Corollary 6.3 If $R \subseteq (\Sigma^*)^k$ is a regular relation, then $R \oplus (i)$ is a $(k+1)$ -dimensional regular relation.

Proof. The corollary directly follows from Proposition 6.2 and Proposition 5.2 having in mind that $R \oplus (i) = \sigma(R \oplus (k+1))$ where σ is the cyclic permutation $(i, i+1, \dots, k, k+1) \in S_{k+1}$. □

It is well-known that the projection of a k -dimensional regular relation is again a regular relation. The following propositions show how to obtain a $(k-1)$ -tape one-letter automaton representing the relation $R \ominus (i)$ (cf. Section 2) directly from a k -tape one-letter automaton representing the relation R .

Proposition 6.4 Let $A = \langle k, \Sigma, S, F, s_0, E \rangle$ be a k -tape one-letter automaton. Let $A' := \langle k-1, \Sigma, S, F, s_0, E' \rangle$ be the $(k-1)$ -tape automaton where for $i \leq k-1$ we have

$$\langle s, \langle \epsilon, \dots, \underset{\uparrow}{a}, \dots, \underset{\uparrow}{\epsilon} \rangle, s' \rangle \in E' \Leftrightarrow \langle s, \langle \underset{\uparrow}{\epsilon}, \dots, \underset{\uparrow}{a}, \dots, \underset{\uparrow}{\epsilon} \rangle, s' \rangle \in E$$

and furthermore

$$\langle s, \langle \epsilon, \dots, \underset{k-1}{\uparrow} \epsilon \rangle, s' \rangle \in E' \Leftrightarrow \exists a_k \in \Sigma_k : \langle s, \langle \epsilon, \dots, \underset{k-1}{\uparrow} \epsilon, \underset{k}{\uparrow} a_k \rangle, s' \rangle \in E.$$

Then $R(A') = R(A) \ominus (k)$.

Note 6.5 *The resulting automaton A' is not necessarily a one-letter automaton because A' may have some $\hat{\epsilon}$ -transitions. It could be transformed into a one-letter automaton using a standard ϵ -removal procedure.*

Proof of Proposition 6.4. It is sufficient to prove that for all $\langle w_1, \dots, w_{k-1}, w_k \rangle \in (\Sigma^*)^k$ and $s' \in S$ we have

$$\langle s_0, \langle w_1, \dots, w_{k-1}, w_k \rangle, s' \rangle \in E^* \Leftrightarrow \langle s_0, \langle w_1, \dots, w_{k-1} \rangle, s' \rangle \in E'^*.$$

Again we use an induction on the construction of E^* and E'^* .

“ \Rightarrow ”. The base is trivial since $\langle s_0, \langle \epsilon, \dots, \underset{k}{\uparrow} \epsilon \rangle, s_0 \rangle \in E^*$ and $\langle s_0, \langle \epsilon, \dots, \underset{k-1}{\uparrow} \epsilon \rangle, s_0 \rangle \in E'^*$.

Let $\langle s_0, \langle w_1, \dots, w'_j, \dots, w_k \rangle, s \rangle \in E^*$ and $\langle s, \langle \epsilon, \dots, \underset{j}{\uparrow} a, \dots, \epsilon \rangle, s' \rangle \in E$ for some $1 \leq j \leq k$.

First assume that $j < k$. The induction hypothesis yields $\langle s_0, \langle w_1, \dots, w'_j, \dots, w_{k-1} \rangle, s \rangle \in E'^*$. Since $\langle s, \langle \epsilon, \dots, \underset{j}{\uparrow} a, \dots, \underset{k-1}{\uparrow} \epsilon \rangle, s' \rangle \in E'$ we have $\langle s_0, \langle w_1, \dots, w'_j a, \dots, w_{k-1} \rangle, s' \rangle \in E'^*$.

If $j = k$, then the induction hypothesis yields $\langle s_0, \langle w_1, \dots, w_{k-1} \rangle, s \rangle \in E'^*$. We have $\langle s, \langle \epsilon, \dots, \underset{k-1}{\uparrow} \epsilon \rangle, s' \rangle \in E'$, hence $\langle s_0, \langle w_1, \dots, w_{k-1} \rangle, s' \rangle \in E'^*$.

“ \Leftarrow ”. Similarly as “ \Rightarrow ”. □

Corollary 6.6 *If $R \subseteq (\Sigma^*)^k$ is a regular relation, then $R \ominus (i)$ is $(k-1)$ -dimensional regular relation.*

Proof. The corollary follows directly from $R \ominus (i) = (\sigma^{-1}(R)) \ominus (k)$, where σ is the cyclic permutation $(i, i+1, \dots, k) \in S_k$ and Proposition 5.2. □

Algorithmic construction. The constructions given in Proposition 6.4 and 5.2 together with an obvious $\hat{\epsilon}$ -elimination show how to obtain a one-letter $(k-1)$ -tape automaton A' for the projection $R \ominus (i)$, given a one-letter k -tape automaton A recognizing R .

7 Composition and cartesian product of regular relations

We now show how to construct composition and cartesian product (cf. Section 2) of regular relations via automata constructions for standard 1-tape automata.

Lemma 7.1 *Let $R_1 \subseteq (\Sigma^*)^{n_1}$ and $R_2 \subseteq (\Sigma^*)^{n_2}$ be regular relations. Then the composition $R_1 \circ R_2$ is a $(n_1 + n_2 - 2)$ -dimensional regular relation.*

Proof. Using Corollary 6.3 we see that the relations

$$\begin{aligned} R'_1 &:= (\dots((R_1 \oplus (n_1 + 1)) \oplus (n_1 + 2)) \oplus \dots) \oplus (n_1 + n_2 - 1) \\ R'_2 &:= (\dots((R_2 \oplus (1)) \oplus (2)) \oplus \dots) \oplus (n_1 - 1) \end{aligned}$$

are $(n_1 + n_2 - 1)$ -dimensional regular relations. Using the definition of \oplus we see that the essential coordinates for R'_1 are in the set $E_1 = \{1, 2, \dots, n_1\}$ and those of R'_2 are in the set

$E_2 = \{n_1, n_1+1, \dots, n_1+n_2-1\}$. Therefore R'_1 and R'_2 have at most one common essential coordinate, namely n_1 . Corollary 4.9 shows that $R = R'_1 \cap R'_2$ is a (n_1+n_2-1) -dimensional regular relation. Since coordinates in E_1 (resp. E_2) are inessential for R'_2 (resp. R'_1) we obtain

$$\begin{aligned} & \langle w'_1, \dots, w'_{n_1-1}, w, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R'_1 \cap R'_2 \\ \Leftrightarrow & \langle w'_1, \dots, w'_{n_1-1}, w \rangle \in R_1 \quad \& \quad \langle w, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R_2. \end{aligned}$$

Using the definition of \ominus and Corollary 4.6 we obtain that $R \ominus (n_1)$ is a (n_1+n_2-2) -dimensional regular relation such that

$$\begin{aligned} & \langle w'_1, \dots, w'_{n_1-1}, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R \ominus (n_1) \\ \Leftrightarrow & \exists w \in \Sigma^* : \langle w'_1, \dots, w'_{n_1-1}, \underset{\uparrow}{w}, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R. \end{aligned}$$

Combining both equivalences we obtain

$$\begin{aligned} & \langle w'_1, \dots, w'_{n_1-1}, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R \ominus (n_1) \\ \Leftrightarrow & \exists w \in \Sigma^* : \langle w'_1, \dots, w'_{n_1-1}, w \rangle \in R_1 \quad \& \quad \langle w, w''_{n_1+1}, \dots, w''_{n_1+n_2-1} \rangle \in R_2, \end{aligned}$$

i.e. $R \ominus (n_1) = R_1 \circ R_2$. □

Lemma 7.2 *Let $R_1 \subseteq (\Sigma^*)^{n_1}$ and $R_2 \subseteq (\Sigma^*)^{n_2}$ be regular relations. Then the cartesian product $R_1 \times R_2$ is a (n_1+n_2) -dimensional regular relation over Σ .*

Proof. Similarly as in Lemma 7.1 we construct the (n_1+n_2) -dimensional regular relations

$$\begin{aligned} R'_1 & := (\dots((R_1 \oplus (n_1+1)) \oplus (n_1+2)) \oplus \dots) \oplus (n_1+n_2) \\ R'_2 & := (\dots((R_2 \oplus (1)) \oplus (2)) \oplus \dots) \oplus (n_1). \end{aligned}$$

The coordinates in $\{1, 2, \dots, n_1\}$ are inessential for R'_2 and those in $\{n_1+1, \dots, n_1+n_2\}$ are inessential for R'_1 . Therefore R'_1 and R'_2 have no common essential coordinate and, by Corollary 4.9, $R := R'_1 \cap R'_2$ is a (n_1+n_2) -dimensional regular relation. Using the definition of inessential coordinates and the definition of \oplus we obtain

$$\begin{aligned} & \langle w'_1, \dots, w'_{n_1}, w''_{n_1+1}, \dots, w''_{n_1+n_2} \rangle \in R \\ \Leftrightarrow & \langle w'_1, \dots, w'_{n_1} \rangle \in R_1 \quad \& \quad \langle w''_{n_1+1}, \dots, w''_{n_1+n_2} \rangle \in R_2, \end{aligned}$$

which shows that $R = R_1 \times R_2$. □

Algorithmic construction. The constructions described in the above proofs show how to obtain one-letter automata for the composition $R_1 \circ R_2$ and for the cartesian product $R_1 \times R_2$ of the regular relations $R_1 \subseteq (\Sigma^*)^{n_1}$ and $R_2 \subseteq (\Sigma^*)^{n_2}$, given one-letter automata A_i for R_i ($i = 1, 2$). In more detail, in order to construct an automaton for $R_1 \circ R_2$ we

1. add $n_2 - 1$ final inessential tapes to A_1 and $n_1 - 1$ initial inessential tapes to A_2 in the way described above (note that the resulting automata are in normal form w.r.t. the new tapes),
2. intersect the resulting automata as conventional one-tape automata over the alphabet $\hat{\Sigma}_{n_1+n_2-1}$, obtaining A ,
3. remove the n_1 -th tape from A and apply an ϵ -removal, thus obtaining A' , which is the desired automaton.

In order to construct an automaton for $R_1 \times R_2$ we

1. add n_2 final inessential tapes to A_1 and n_1 initial inessential tapes to A_2 in the way described above,

2. intersect the resulting automata as normal one-tape automata over the alphabet $\hat{\Sigma}_{n_1+n_2}$, obtaining A , which is the desired automaton.

At the end we will discuss the problem of how to represent identity relations as regular relations. First observe that the automaton $A := \langle 2, \Sigma, S, F, s_0, E \rangle$ where $\Sigma := \{a_1, \dots, a_n\}$, $S := \{s_0, s_1, \dots, s_n\}$, $F := \{s_0\}$ and

$$E := \{\langle s_0, \langle a_i, \epsilon \rangle s_i \mid 1 \leq i \leq n \rangle \cup \{\langle s_i, \langle \epsilon, a_i \rangle s_0 \mid 1 \leq i \leq n \rangle\}$$

accepts $R(A) = \{\langle v, v \rangle \mid v \in \Sigma^*\}$. The latter relation we denote with Id_Σ .

Proposition 7.3 *Let R_1 be 1-dimensional regular relation, i.e., a regular language. Then the set $Id_{R_1} := \{\langle v, v \rangle \mid v \in R_1\}$ is a regular relation. Moreover $Id_{R_1} = (R_1 \oplus (2)) \cap Id_\Sigma$.*

8 Conclusion

We introduced the concept of a one-letter k -tape automaton and showed that one-letter automata can be considered as conventional 1-tape automata over an enlarged alphabet. Using this correspondence, standard constructions for union, concatenation, and Kleene-star for 1-tape automata can be directly used for one-letter automata. Furthermore we have seen that the usual relational operations for k -tape automata can be traced back to the intersection of 1-tape automata plus straightforward operations for adding, permuting and erasing tapes.

We have implemented the presented approach for implementation of transducer (2-tape automata) representing rewrite rules. Using it we have successfully realized Bulgarian hyphenation and tokenization.

Still, in real applications the use of one-letter automata comes with some specific problems, in particular in situations where the composition algorithm is heavily used. In the resulting automata we sometimes find a large number of paths that are equivalent if permutation rules for generators are taken into account. For example, we might find three paths with label sequences

$$\begin{aligned} &\langle a, \epsilon \rangle, \langle a, \epsilon \rangle, \langle \epsilon, b \rangle \\ &\langle \epsilon, b \rangle, \langle a, \epsilon \rangle, \langle a, \epsilon \rangle \\ &\langle a, \epsilon \rangle, \langle \epsilon, b \rangle, \langle a, \epsilon \rangle, \end{aligned}$$

all representing the tuple $\langle aa, b \rangle$. In the worst case this may lead to an exponential blow-up of the number of states, compared to the classical construction for n -tape automaton.

We currently study techniques to get rid of superfluous paths. In many cases, equivalences of the above form can be recognized and used for eliminating states and transitions. The extension and refinement of these methods is one central point of current and future work.

References

- [1] Dale Gerdemann and Gertjan van Noord, Transducers from Rewrite Rules with Back-references. EACL 99, Bergen Norway, 1999.
- [2] John Hopcroft and Jeffrey Ullman, Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, MA, 1979.
- [3] Lauri Karttunen, Constructing Lexical Transducers, In The Proceedings of the 15th International Conference on Computational Linguistics. Coling 94, I, pages 406-411. Kyoto, Japan, August 5-9, 1994.

- [4] Lauri Karttunen, The Replace Operator, Finite-State language processing, 117-147, E. Roche, Y. Schabes (Eds.), MIT Press, 1997.
- [5] L. Karttunen, J.-P. Chanod, G. Grefenstette, A Schiller, Regular Expressions for language Engineering, Journal of Natural Language Engineering vol 2 no 4 (1997) pp 307-330, Cambridge University Press, 1997.
- [6] Ronald Kaplan and Martin Kay, Regular Models of Phonological Rule Systems, Computational Linguistics, 20(3):331-379, 1994.
- [7] Stoyan Mihov and Denis Maurel, Direct Construction of Minimal Acyclic Subsequential Transducers, Implementation and Application of Automata, S. Yu, A. Pun (Eds.), LNCS 2088, Springer 2001.
- [8] M. Mohri, On Some Applications of Finite-State Automata Theory to Natural Language Processing, *Natural Language Engineering*, Vol. 2(1), 1-20, 1996.
- [9] Mehryar Mohri, Finite-State Transducers in Language and Speech Processing. Computational Linguistics, 23(2), 1997.
- [10] Mehryar Mohri, Fernando Pereira, Michael Riley, A Rational Design for a Weighted Finite-State Transducer Library. In Derick Wood and Sheng Yu, editors, Proceedings of the Second International Workshop on Implementing Automata (WIA '97). volume 1436 of Lecture Notes in Computer Science, pages 144-158. Springer-Verlag, Berlin-NY, September 1998.
- [11] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, 16(1):69-88, 2002.
- [12] Gertjan van Noord. FSA Utilities: A Toolbox to Manipulate Finite-state Automata. In: Darrell Raymond, Derick Wood and Sheng Yu (eds), Automata Implementation. Lecture Notes in Computer Science 1260, Springer Verlag, 1997.
- [13] Emmanuel Roche and Yves Schabes, Deterministic Part-Of-Speech Tagging with Finite State Transducers, Computational Linguistics, 21(2), 227-253, June 1995.
- [14] Emmanuel Roche and Yves Schabes, Introduction, Finite-State language processing, E. Roche, Y. Schabes (Eds.), MIT Press, 1997.